kaspersky

Kaspersky LogAnalyzer

© 2026 AO Kaspersky Lab

Contents

Kaspersky LogAnalyzer Help
About LogAnalyzer
- Distribution kit
- Hardware and software requirements
- LogAnalyzer interface
- Compatibility with other applications

Program architecture
- Core
- Collector
- Storage
- Basic entities
  - About tenants
  - About events
  - About assets
  - About resources
  - About services
  - About agents

Program licensing
- About the End User License Agreement
- About the license
- About the License Certificate
- About the license key
- About the key file
- About the license code
- Data provision in Kaspersky LogAnalyzer
- Adding a license key to the program web interface
- Viewing information about an added license key in the program web interface
- Removing a license key in the program web interface

Administrator's guide
- Installing and uninstalling LogAnalyzer
  - Program installation requirements
    - Upgrading from Oracle Linux 8.x to Oracle Linux 9.x
  - Ports used by LogAnalyzer during installation
  - Downloading CA certificates
  - Reissuing internal CA certificates
  - Modifying the self-signed web console certificate
  - Synchronizing time on servers
  - About the inventory file
    - LogAnalyzer settings in the inventory file
  - Installation on a single server
    - Preparing the single.inventory.yml inventory file
    - Installing the program on a single server
  - Distributed installation

Preparing the test machine
Preparing the target machine
Preparing the distributed.inventory.yml inventory file
Installing the program in a distributed configuration
Distributed installation with LogAnalyzer Core in a Kubernetes cluster
Additional requirements for deploying KUMA Core in Kubernetes
Installing LogAnalyzer in a Kubernetes cluster
Preparing the test machine
Preparing the target machine
Preparing the k0s.inventory.yml inventory file
Installing the program in a high availability configuration
Migrating the LogAnalyzer Core to a new Kubernetes cluster
LogAnalyzer Core availability under various scenarios
Managing Kubernetes and access to LogAnalyzer
Time zone in a Kubernetes cluster
Distributed installation with LogAnalyzer Core in a Raft cluster
Preparing the test machine
Preparing the target machine
Preparing the distributed.inventory.yml file for installing LogAnalyzer with multiple LogAnalyzer Core services in a Raft cluster
Installing LogAnalyzer in a Raft cluster
Expanding the Raft cluster
Supported scenarios for installing and upgrading from 3.4.x to 4.x with the Core installed on a Raft cluster
Backing up LogAnalyzer
Modifying the configuration of LogAnalyzer
Upgrading older versions of LogAnalyzer
Troubleshooting update errors
Uninstalling LogAnalyzer
Working with tenants
Selecting a tenant
Tenant affiliation rules
Managing users
User roles
Configuring the security policy for logging in to the LogAnalyzer web interface
Creating a user
Editing a user account
Special considerations involved in changing the set of spaces assigned to a user
LogAnalyzer services
Services tools
Getting service identifier
Stopping, starting, checking status of the service
Restarting the service
Deleting the service
Partitions window
Searching for related events
Service resource sets
Creating a storage

- ClickHouse cluster structure
- ClickHouse cluster node settings
- Cold storage of events
  - Removing cold storage disks
  - Disconnecting, archiving, and connecting partitions
- Creating a set of resources for a storage
- Creating a storage service in the LogAnalyzer web interface
- Installing a storage in the LogAnalyzer network infrastructure

Creating an event router
- Starting the Event Router Installation Wizard
  - Step 1. General settings of the event router
  - Step 2. Routing
  - Step 3. Setup validation
- Installing the event router on the server

Creating a collector
- Starting the Collector Installation Wizard
  - Step 1. Connect event sources
  - Step 2. Transport
  - Step 3. Event parsing
  - Step 4. Filtering events
  - Step 5. Event aggregation
  - Step 6. Event enrichment
  - Step 7. Routing
  - Step 8. Setup validation
- Installing a collector in the LogAnalyzer network infrastructure.
- Validating collector installation
- Ensuring uninterrupted collector operation
  - Event stream control using rsyslog
  - Event stream control using nginx

Predefined collectors

Creating an agent
- Creating a set of resources for an agent
- Managing connections for an agent
- Creating an agent service in the LogAnalyzer web interface
- Automatically created agents
- Installing an agent in a LogAnalyzer network infrastructure
  - Installing a LogAnalyzer agent on Linux devices
  - Installing a LogAnalyzer agent on Windows devices
- Update agents
- Exporting events from isolated network segments to LogAnalyzer
  - Diode agent configuration file
  - Description of secret fields
  - Installing Linux Agent in an isolated network segment
  - Installing Windows Agent in an isolated network segment
- Exporting events from Windows machines to LogAnalyzer
- Importing a file with a list of servers for WMI agents

Configuring event sources

Configuring receipt of Auditd events
Configuring the LogAnalyzer collector for receiving Auditd events
Configuring the event source server using the rsyslog service
Configuring receipt of KATA/EDR events
Configuring the export of KATA/EDR events to LogAnalyzer
Creating a LogAnalyzer collector for receiving KATA/EDR events
Installing a LogAnalyzer collector for receiving KATA/EDR events
Configuring the export of Kaspersky Security Center events to the LogAnalyzer SIEM system
Configuring export of Kaspersky Security Center events in CEF format
Configuring a LogAnalyzer collector for collecting Kaspersky Security Center events
Installing a LogAnalyzer collector for obtaining Kaspersky Security Center events
Configuring receiving Kaspersky Security Center event from MS SQL
Creating an account in the MS SQL database
Configuring the SQL Server Browser service
Creating a secret in LogAnalyzer
Configuring a connector
Configuring a LogAnalyzer collector for receiving Kaspersky Security Center events from an MS SQL database
Installing a LogAnalyzer collector for receiving Kaspersky Security Center events from the MS SQL database
Configuring the receipt of events from Windows devices using LogAnalyzer Agent (WEC)
Configuring audit of events from Windows devices
Configuring an audit policy on a Windows device
Configuring an audit using a group policy
Configuring centralized receipt of events from Windows devices using the Windows Event Collector service
Configuring data transfer from the event source server
Configuring the Windows Event Collector service
Granting permissions to view Windows events
Granting permissions to log on as a service
Configuring a LogAnalyzer collector for receiving events from Windows devices
Installing a LogAnalyzer collector for receiving events from Windows devices
Configuring the export of of events from Windows devices to LogAnalyzer using LogAnalyzer Agent (WEC)
Configuring the receipt of events from Windows devices using LogAnalyzer Agent (WMI)
Managing audit settings for managing LogAnalyzer.
Configuring an audit using a local policy
Configuring an audit using a group policy
Configuring data transfer from the event source server
Granting permissions to view Windows events
Granting permissions to log on as a service
Configuring receipt of DNS server events using the ETW agent
Configuration on the Windows side
Configuring receipt of PostgreSQL events
Installing the pgAudit plugin
Configuring a Syslog server to send events
Configuring receipt of IVK Kolchuga-K events
Configuring the sending of IVK Kolchuga-K events to LogAnalyzer.
Configuring receipt of CryptoPro NGate events
Configuring the export of CryptoPro NGate events to LogAnalyzer.
Configuring receipt of Ideco UTM events

Configuring the export of Ideco UTM events to LogAnalyzer.
Configuring receipt of KWTS events
Configuring the export of KWTS events to LogAnalyzer.
Configuring receipt of KLMS events
Configuring the export of KLMS events to LogAnalyzer
Configuring receipt of KSMG events
Configuring the export of KSMG events to LogAnalyzer
Configuring the receipt of KICS for Networks events
Creating a KICS for Networks connector for sending events to LogAnalyzer
Configuring the export of KICS for Networks events to LogAnalyzer
Creating a LogAnalyzer collector to receive KICS for Networks events
Configuring receipt of PT NAD events
Configuring the export of PT NAD events to LogAnalyzer.
Configuring receipt of events using the MariaDB Audit Plugin
Configuring the MariaDB Audit Plugin to send MySQL events
Configuring the MariaDB Audit Plugin to send MariaDB Events
Configuring a Syslog server to send events
Configuring receipt of Apache Cassandra events
Configuring Apache Cassandra event logging in LogAnalyzer.
Configuring receipt of FreeIPA events
Configuring the export of FreeIPA events to LogAnalyzer.
Configuring receipt of VipNet TIAS events
Configuring the export of ViPNet TIAS events to LogAnalyzer.
Configuring receipt of Nextcloud events
Configuring audit of Nextcloud events
Configuring a Syslog server to send Nextcloud events
Configuring receipt of Snort events
Configuring logging of Snort events
Configuring receipt of Suricata events
Configuring audit of Suricata events
Configuring receipt of FreeRADIUS events
Configuring audit of FreeRADIUS events
Configuring a Syslog server to send FreeRADIUS events
Configuring receipt of VMware vCenter events
Configuring the connection to VMware vCenter
Configuring receipt of zVirt events
Configuring export of zVirt events
Configuring receipt of Zeek IDS events
Conversion of the Zeek IDS event log format
Configuring Windows event reception using Kaspersky Endpoint Security for Windows
Configuring Linux event reception using Kaspersky Endpoint Security for Linux
Configuring receipt of Codemaster Mirada events
Configuring audit of the Codemaster Mirada system
Configuring receipt of Postfix events
Configuring Postfix to send events
Configuring a LogAnalyzer collector for receiving and processing Postfix events
Configuring receipt of CommuniGate Pro events

- Configuring CommuniGate Pro to send events
- Configuring a LogAnalyzer collector for receiving and processing CommuniGate Pro events
- Configuring receipt of Yandex Cloud events
- Configuring audit of Yandex Cloud events
- Configuring export of Yandex Cloud events
- Configuring receipt of Microsoft 365 events
- Configuring receipt of the Kontinent encryption system events
- Preparing to receive Kontinent events
- Configuring the kuma-kont utility
- Starting and managing the kuma-kont service
- Settings in the kuma-kont-config.yaml configuration file
- Configuring receipt of VK WorkSpace Mail events
- Configuring the export of VK WorkSpace Mail events

Monitoring event sources

- Source status
- List of event sources
- Monitoring policies

Integration with other solutions

- Integration with Kaspersky CyberTrace
- Integrating CyberTrace indicator search
- Configuring CyberTrace to receive and process requests
- Creating event Enrichment rules
- Integrating CyberTrace interface
- Configuring event enrichment and retroscan
- Integration with Kaspersky Threat Intelligence Portal
- Initializing the Kaspersky Threat Intelligence Portal integration
- Requesting information from Kaspersky Threat Intelligence Portal
- Viewing information from Kaspersky Threat Intelligence Portal
- Updating information from Kaspersky Threat Intelligence Portal
- Authentication using domain accounts
- Enabling and disabling domain authentication
- Configuring connection between KUMA and FreeIPA
- Configuring a connection between LogAnalyzer and Active Directory
- Configuring a connection between LogAnalyzer and Active Directory Federation Services
- Configuring connection in Active Directory Federation Services
- Troubleshooting the Access denied error
- Integration with NeuroDAT SIEM IM
- Importing asset information from RedCheck
- Configuring receipt of Sendmail events
- Configuring Sendmail logging
- Configuring export of Sendmail events

Managing LogAnalyzer

- Logging in to the application web interface
- Service monitoring
- Viewing LogAnalyzer metrics
- Managing LogAnalyzer tasks
- Viewing the tasks table

Configuring the display of the tasks table
Viewing task run results
Restarting a task
Connecting to an SMTP server
LogAnalyzer notifications
LogAnalyzer logs
Working with geographic data
Geodata format
Converting geographic data from MaxMind to IP2Location
Importing and exporting geographic data
Default mapping of geographic data
User guide
LogAnalyzer resources
Operations with resources
Creating, renaming, moving, and deleting resource folders
Creating, duplicating, moving, editing, and deleting resources
Bulk deletion of resources
Updating resources
Configuring a custom source using Kaspersky Update Utility
Exporting resources
Importing resources
Resource search
Tag management
Resource usage tracing
Resource versioning
Comparing resource versions
Restoring a resource version
Configuring the retention period for resource versions
Destinations
Destination, internal type
Destination, storage type
Destination, eventRouter type
Normalizers
Event parsing settings
Extended event schema
Adding extended event schema fields
Editing extended event schema fields
Importing and exporting extended event schema fields
Deleting extended event schema fields
Using extended event schema fields in normalizers
Enrichment in the normalizer
Conditions for forwarding data to an extra normalizer
Supported event sources
Aggregation rules
Enrichment rules
Filters
Proxies

- Dictionaries
- Connectors
  - Viewing connector settings
  - Adding a connector
  - Connector settings
    - Connector, internal type
    - Connector, tcp type
    - Connector, udp type
    - Connector, nats-jetstream type
    - Connector, kafka type
    - Connector, http type
    - Connector, sql type
    - Connector, file type
    - Connector, 1c-log type
    - Connector, 1c-xml type
    - Connector, diode type
    - Connector, ftp type
    - Connector, nfs type
    - Connector, wmi type
    - Connector, wec type
    - Connector, etw type
    - Connector, snmp type
    - Connector, snmp-trap type
      - Configuring the source of SNMP trap messages for Windows
        - Configuring and starting the SNMP and SNMP trap services
        - Configuring the Event to Trap Translator service
    - Connector, kata/edr type
    - Connector, vmware type
    - Connector, elastic type
    - Connector, office365 type
    - Connector, dfi type
  - Predefined connectors
- Secrets

Analytics
- Working with events
  - Filtering and searching events
    - Selecting Storage
    - Generating an SQL query using a builder
    - Manually creating an SQL query
    - Saving query history
    - Managing saved search queries
    - Filtering events by period
    - Grouping events
    - Displaying names instead of IDs
    - Presets
    - Saving and selecting events filter configuration
    - Deleting event filter configurations

Supported ClickHouse functions
Viewing event detail areas
Exporting events
Configuring the table of events
Refreshing events table
Getting events table statistics
Generating an SQL query using LogAnalyzer SQL functions
Using the 'enrich' function set in SQL queries
Using the 'lookup' function set in SQL queries
Event route tracing
Categorization of events
Granular access to events
Dashboards
Creating a dashboard
Selecting a dashboard
Selecting a dashboard as the default dashboard
Managing dashboard settings
Deleting a dashboard
Enabling and disabling the TV mode
Predefined dashboards
Universal dashboards
Exporting dashboards
Importing dashboards
Reports
Report template
Creating report template
Configuring report schedule
Editing report template
Copying report template
Deleting report template
Exporting or importing a report template
Universal report templates
Generated reports
Viewing reports
Generating reports
Saving reports
Updating reports
Deleting reports
Widgets
Basics of managing widgets
Special considerations for displaying data in widgets
Creating a widget
Editing a widget
Deleting a widget
Events widget
Contacting Technical Support
REST API

- Creating a token
- Configuring permissions to access the API
- Authorizing API requests
- Standard error
- REST API v3 operations

Appendices

- Commands for components manual starting and installing
- Checking the integrity of LogAnalyzer files
- Normalized event data model
- Configuring the data model of a normalized event from KATA EDR
- User account data model
- LogAnalyzer audit events
  - Event fields with general information
  - User was successfully signed in or failed to sign in
  - User login changed
  - User role was changed
  - Other data of the user was changed
  - User logged out
  - User password was changed
  - User was created
  - User role was assigned
  - User role was revoked
  - The user has edited the set of fields settings to define sources
  - User access token was changed
  - Changed the set of spaces to differentiate access to events
  - Service was created
  - Service was deleted
  - Service was reloaded
  - Service was restarted
  - Service was started
  - Service was paired
  - Service status was changed
  - Partition index was deleted by user
  - Storage partition was deleted automatically due to expiration
  - Storage partition was deleted automatically or moved due to exceeding the storage capacity.
  - Resource was added
  - Resource was deleted
  - Resource was updated
  - Settings were updated
  - Tenant was created
  - Tenant was enabled
  - Tenant was disabled
  - Other tenant data was changed
  - Updated data retention policy after changing drives or the operation ended with an error
  - The dictionary was successfully updated on the service or operation was unsuccessful
- Sending test events to LogAnalyzer
- Time format

- Mapping fields of predefined normalizers
- Deprecated resources
- Generating events for testing a normalizer

Glossary

- Aggregation
- Cluster
- Collector
- Connector
- Dashboard
- Enrichment
- Event
- Filter
- LogAnalyzer web interface
- Network port
- Normalization
- Normalizer
- Parsing
- Raw event
- Report
- Role
- SELinux (Security-Enhanced Linux)
- SIEM
- STARTTLS
- UserPrincipalName

Information about third-party code

Trademark notices

The MITRE Corporation Terms of Use

List of limitations

Kaspersky LogAnalyzer Help

Hardware and software requirements

- Hardware and software requirements

Getting started

- Application architecture
- Installation and removal
- Managing tenants
- Monitoring event sources

Managing the LogAnalyzer web interface

- Administrator's guide
- User guide

Additional features

- Interaction with external systems via API

Licensing

- LogAnalyzer licensing

Contacting Technical Support

- How to get technical support

About LogAnalyzer

LogAnalyzer (hereinafter also referred to as the "application") is an integrated software solution that combines the following functionality:

- Receiving, processing, and storing information security events of Kaspersky applications
- Searching in received events
- Building dashboards and generating reports on received events.

LogAnalyzer is available in the following versions:

- Basic (supplied as part of other solutions)
- Commercial (supplied as a standalone solution).

The application is built on a microservice architecture. This means you can create and flexibly configure microservices (hereinafter also referred to as "services") for your particular infrastructure.

The update functionality (including anti-virus signature updates and code base updates) may not be available in the application in the territory of the USA.

Distribution kit

The distribution kit includes the following files:

- kuma-ansible-installer-<build number>.tar.gz, used to install LogAnalyzer without using Kubernetes. You can perform a distributed installation with the LogAnalyzer Core in a Raft-based cluster.
- kuma-ansible-installer-k8s-<build number>.tar.gz, used for installing LogAnalyzer in a high-availability configuration based on Kubernetes.
- Files containing information about the version (release notes) in Russian and English

Hardware and software requirements

Recommended hardware

This section lists the hardware requirements for processing an incoming event stream in LogAnalyzer at various Events per Second (EPS) rates.

The table below lists the hardware and software requirements for installing the LogAnalyzer components, assuming that the ClickHouse cluster only accepts INSERT queries. Hardware requirements for SELECT queries are calculated separately for the particular database usage profile of the customer.

Recommended hardware for ClickHouse cluster storage

When designing the storage cluster, the EPS, the average event size, and the storage depth are not the only things you must take into account. You must also be mindful of special considerations that are presented in this article for your convenience in the following sections:

- Disk subsystem
- Network interface
- RAM
- Usage profile

Disk subsystem

Insert queries (write access) constitute the majority of the load that LogAnalyzer places on ClickHouse:

- Events arrive in the database in an almost constant stream.
- Inserts are small and frequent by ClickHouse standards.
- ClickHouse has high write amplification because of the constant merging of parts of partitions in the background.
- A partition part under 1 GB in size is stored as a single file. When the part reaches 1 GB, wide column format is applied, wherein each column of the table is represented by two files, .dat and .mrk. In this case, overwriting a part to merge it with another part requires writing to 384 files + fsync for each, which is a lot of IOPS.

Of course, ClickHouse handles search queries at the same time, but there are much fewer of those than write operations. Considering the load profile, we recommend organizing the disk subsystem in the following way:

1. Use DAS (Direct Attached Storage) with NVME or SAS/SATA interfaces, avoiding SAN/NAS solutions.
2. If the nodes of the ClickHouse cluster are deployed in a virtual environment, the disk array must be directly mounted to the /opt/kaspersky/kuma directory. Avoid virtual disks.
3. Ideally, SSD (SATA/NVME) should be used, if possible. Any server-grade SSDs will outperform HDDs (even 15k RPM). For SSDs, use RAID-10 or RAID-0; for RAID-0, make sure that replication in ClickHouse is used.
4. If SSDs cannot be used, use 15k-RPM SAS HDDs in RAID-10.
5. A hybrid option is possible: storing hot data (for example, for the last month) on SSDs, and cold data on HDDs. This arrangement makes sure the SSD arrays always handle the write operations.
6. A software RAID array (madm) will always outperform a hardware RAID controller and offer more flexibility. When using RAID-0 or RAID-10, it is important that the stripe size be 1 MB. In hardware RAID controllers, the default stripe size is often 64 KB, and the maximum supported value is 256–512 KB. For RAID-10, near-layout is optimal for writing, and far-layout is optimal for reading. You must keep this in mind when using a hybrid SSD + HDD configuration.
7. We recommend EXT4 as the file system. If, for any reason, this file system cannot be used, you may use the XFS file system instead. We recommend enabling the 'noatime' mount option for either of these file systems.

Network interface

1. On each node of the cluster, a network interface with a bandwidth of at least 10 Gbps must be installed and appropriately switched. Considering write amplification and replication, 25 Gbps is ideal.

2. If necessary, replication processes can use a separate network interface that does not handle insert and search traffic.

RAM

ClickHouse recommends keeping the RAM-to-stored-data ratio at 1:100.
But how much RAM is needed if each node is supposed to store 50 TB, given that the event storage depth is often dictated by regulatory requirements, while search queries are not performed to the full storage depth? Thus, the recommendation can be interpreted as follows: if the average query depth involves scanning partitions with a total volume of 10 TB, then you need 100 GB of RAM. This is a general recommendation for broad analytical queries.
In the general case, more free RAM on the server improves the performance of search queries for the most recent events because the contents of the partition files will stay in operating system's page cache, so file content will be read from RAM, and not from disk.

Usage profile

The hardware requirements are designed for the database to consume a stream of a certain EPS when the database is at rest (no search queries are being executed, only inserts). At the KUMA deployment stage, it is not always possible to accurately answer the following questions:

1. What search and analytic queries will be performed in the system?

2. What is the intensity, concurrency, and depth of search queries?

3. What is the actual performance of the cluster nodes?

Thus, it is perfectly normal for a LogAnalyzer ClickHouse cluster to evolve over time. If an increase of EPS or intensity/resource consumption of search queries makes the cluster unable to cope with the load, you can add more shards (horizontal scaling) or improve the disk subsystems/CPU/RAM on the cluster nodes.

The configuration of the equipment must be chosen based on the system load profile. You can use the "All-in-one" configuration for an event stream of under 10,000 EPS and when using graphical panels supplied with the system.

LogAnalyzer supports Intel and AMD CPUs with SSE 4.2 and AVX instruction set support.

	Up to 3,000 EPS	Up to 10,000 EPS	Up to 20,000 EPS	Up to 50,000 EPS
LogAnalyzer Core requirements	–	–	One device. Device characteristics: • At least 12 threads or vCPUs. • At least 12 GB of RAM. • At least 500 GB in the /opt directory. • Data storage type: SSD. • Data transfer rate: at least 53 Mbps.	One device. Device characteristics: • At least 16 threads or vCPUs. • At least 24 GB of RAM. • At least 1 TB in the /opt directory. • Data storage type: SSD. • Data transfer rate: at least 134 Mbps.
Collector requirements	–	–	One device. Device characteristics: • At least 16 threads or vCPUs. • At least 24 GB of RAM. • At least 500 GB in the /opt directory. • Data storage type: SSD. • Data transfer rate: at least 53 Mbps.	Two devices. Characteristics of each device: • At least 16 threads or vCPUs. • At least 24 GB of RAM. • At least 500 GB in the /opt directory. • Data storage type: SSD. • Data transfer rate: at least 134 Mbps.
Event router requirements	–	CPU: 3 cores RAM: 240 MB.	CPU: 6 cores RAM: 250 MB.	CPU: 9 cores RAM: 7000 MB.
Keeper requirements	–	–	Three devices. Characteristics of each device: • At least 6 threads or vCPUs. • At least 12 GB of RAM. • At least 100 GB in the /opt directory. • Data storage type: SSD. • Data transfer rate: at least 53 Mbps.	Three devices. Characteristics of each device: • At least 6 threads or vCPUs. • At least 12 GB of RAM. • At least 100 GB in the /opt directory. • Data storage type: SSD. • Data transfer rate: at least 134 Mbps.
Storage requirements	–	–	Two devices. Characteristics of each device: • At least 24 threads or vCPUs. • At least 64 GB of RAM. • At least 16.5 TB in the /opt directory. • Data storage type: SSD*. The recommended transfer rate between ClickHouse nodes is at least 10 Gbps if the data stream is equal to or exceeds 20,000 EPS.	Four devices. Characteristics of each device: • At least 24 threads or vCPUs. • At least 64 GB of RAM. • At least 41.3 TB in the /opt directory. • Data storage type: SSD*. The recommended transfer rate between ClickHouse nodes is at least 10 Gbps if the data stream is equal to or exceeds 20,000 EPS.

	Up to 3,000 EPS	Up to 10,000 EPS	Up to 20,000 EPS	Up to 50,000 EPS
Operating systems	• Ubuntu 22.04 LTS, 24.04 LTS. • Oracle Linux 8.6, 8.7, 8.10, 9.2, 9.3, 9.4, 9.5. • Astra Linux Special Edition RUSB.10015-01 (2021-1126SE17 update 1.7.1) • Astra Linux Special Edition RUSB.10015-01 (2022-1011SE17MD update 1.7.2.UU.1) • Astra Linux Special Edition RUSB.10015-01 (2022-1110SE17 update 1.7.3) Core version 5.15.0.33 or higher is required. • Astra Linux Special Edition RUSB.10015-01 (2023-0630SE17MD, update 1.7.4.UU.1) • Astra Linux Special Edition RUSB.10015-01 (2024-0212SE17MD, update 1.7.5.UU.1) • Astra Linux Special Edition RUSB.10015-01 (2024-0830SE17, update 1.7.6) • Astra Linux Special Edition RUSB.10015-01 (2025-0319SE17 update 1.7.7) • Astra Linux Special Edition RUSB.10015-01 (2025-0114SE18MD, update 1.8.1.UU.2). • RED OS 7.3.4, 8 • MosOS 15.5			
TLS ciphersuites	TLS versions 1.2 and 1.3 are supported. Integration with a server that does not support the TLS versions and ciphersuites that LogAnalyzer requires is impossible. Supported TLS 1.2 ciphersuites: • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 Supported TLS 1.3 ciphersuites: • TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256			

Recommended configurations for different load levels

The table below describes the configurations and recommended resources for LogAnalyzer components depending on the EPS level, including storage requirements and sizing recommendations.

Recommended hardware configurations and resource consumption by LogAnalyzer depending on EPS

Setting	Up to 3,000 EPS	Up to 10,000 EPS	Up to 20,000 EPS	Up to 50,000 EPS
Installation type	Installation on a single server	Installation on a single server	Distributed installation on multiple servers	Distributed installation on multiple servers
Number of servers and their purpose	One server that hosts all system components.	One server that hosts all system components.	One server for the LogAnalyzer Core. One server for the collector Three dedicated servers with the keeper role. Two servers for the Storage.	One server for the LogAnalyzer Core. Two servers for the Collector. Three dedicated servers with the keeper role. Four servers for the storage.

Setting	Up to 3,000 EPS	Up to 10,000 EPS	Up to 20,000 EPS	Up to 50,000 EPS
Server hardware	At least 16 threads or 16 vCPUs. At least 32 GB of RAM. At least 500 GB in the /opt directory. Data storage type: SSD. Data transfer rate: at least 100 Mbps.	At least 24 threads or 24 vCPUs. At least 64 GB of RAM. At least 500 GB in the /opt directory. Data storage type: SSD. Data transfer rate: at least 100 Mbps.	The LogAnalyzer Core, collector, and keeper require at least 16 threads or 16 vCPUs and at least 64 GB of RAM per server. The storage requires fast SSDs, high IOPS, at least 1 Gbps of network bandwidth.	Same specifications as for EPS 20,000, but with twice the number of collectors and storage servers. Higher requirements for network connectivity between all nodes: at least 1 Gbps of network bandwidth is recommended.
Storage configuration (ClickHouse)	One shard, one replica. Disk buffer enabled.	One shard, one replica. Disk buffer enabled.	Two shards, two replicas in each. If high availability is not required, one server with one replica per shard may be used.	Two shards, two replicas in each. If high availability is not required, two servers with one replica per shard may be used.

Additional recommendations for specifying and sizing the configuration:

- If users actively engage with the system, including frequent generation of reports, using dashboards, and running custom SQL queries, you need to adjust the sizing of both the LogAnalyzer Core and the storage because such activity substantially increases the load on both components.
- When choosing the disk subsystem for the storage (ClickHouse), only SSD drives may be used. An exception is possible if you do not plan to run aggregation SQL queries with a depth of more than 24 hours. In this case, you may use HDDs with the following minimum specifications: 15,000 RPM, SAS interface, RAID-10.
- When placing components in HDFS, you may use HDD regardless of the queries that you run.
- Events are exported via temporary files that are saved in the /opt/kaspersky/kuma/core/tmp/ directory. These files are automatically deleted after 10 days. If you plan to export a large amount of events, you must allocate additional space.

Working in virtual environments

The following virtual environments are supported for installing LogAnalyzer:

- VMware 6.5 or later
- Hyper-V for Windows Server 2012 R2 or later
- QEMU-KVM 4.2 or later
- "Brest" virtualization software RDTSP.10001-02

Working in cloud environments

KUMA can work in a cloud infrastructure. The system can be installed on virtual machines following the IaaS (infrastructure-as-a-service) model.

For a cloud infrastructure, we recommend using the single-server configuration for 3000 EPS and 10,000 EPS. Virtual machines must satisfy the hardware and software requirements of a regular installation.

When choosing the disk subsystem of the server, use the "number of input/output operations (IOPS)" parameter as the reference. The recommended minimum value is 1000 IOPS.

Resource recommendations for the Collector component

Consider that for event processing efficiency, the CPU core count is more important than the clock rate. For example, eight CPU cores with a medium clock rate can process events more efficiently than four CPU cores with a high clock rate.

Consider also that the amount of RAM utilized by the collector depends on configured enrichment methods (DNS, accounts, assets, enrichment with data from Kaspersky CyberTrace) and whether aggregation is used (RAM consumption is influenced by the data aggregation window setting, the number of fields used for aggregation of data, volume of data in fields being aggregated). The usage of computation resources by LogAnalyzer depends on the type of events being parsed and the efficiency of the normalizer.

Kaspersky recommendations for storage servers

You must use high-speed protocols, such as Fibre Channel or iSCSI 10G for the connection of the data storage system to storage servers. We do not recommend using application-level protocols such as NFS or SMB to connect data storage systems.

On ClickHouse cluster servers, we recommend using the ext4 file system.

If you are using RAID arrays, we recommend using RAID 0 for high performance, or RAID 10 for high performance and high availability.

To ensure high availability and performance of the data storage subsystem, we recommend making sure that all ClickHouse nodes are deployed strictly on different disk arrays.

If you are using a virtualized infrastructure to host system components, we recommend deploying ClickHouse cluster nodes on different hypervisors. You must prevent any two virtual machines with ClickHouse from running on the same hypervisor.

For high-load LogAnalyzer installations, we recommend installing ClickHouse on physical servers.

Requirements for agent devices

You must install agents on network infrastructure devices that will send data to the LogAnalyzer collector. Device requirements are listed in the following table.

	Windows devices	Linux devices
CPU	Single-core, 1.4 GHz or higher	Single-core, 1.4 GHz or higher
RAM	512 MB	512 MB

	Windows devices	Linux devices
Free disk space	1 GB	1 GB
Operating systems	• Microsoft® Windows® 2012 Microsoft Windows 2012 support is limited. • Microsoft Windows Server 2012 R2 Microsoft Windows 2012 R2 support is limited. • Microsoft Windows Server 2016 • Microsoft Windows Server 2019 • Microsoft Windows Server 2022 • Microsoft Windows Server 2025 • Microsoft Windows 10 • Microsoft Windows 11	• Oracle® Linux 8.6, 8.7, 9.2, 9.4, 9.5. • Alma Linux 9.5 • Astra Linux Special Edition RUSB.10015-01 (2021-1126SE17 update 1.7.1) • Astra Linux Special Edition RUSB.10015-01 (2022-1011SE17MD update 1.7.2.UU.1) • Astra Linux Special Edition RUSB.10015-01 (2022-1110SE17 update 1.7.3) • Astra Linux Special Edition RUSB.10015-01 (2023-0630SE17MD, update 1.7.4.UU.1) • Astra Linux Special Edition RUSB.10015-01 (2025-0319SE17 update 1.7.7) • Astra Linux Special Edition RUSB.10015-01 (2025-0114SE18MD, update 1.8.1.UU.2). • Astra Linux 1.8 x86-64 with glibc 2.36 (Debian GLIBC 2.36-9+deb12u7+ci202405171200+astra5). • Astra Linux 1.7 x86-64 with glibc 2.28 (Debian GLIBC 2.28-10+deb10u4+ci202407161455+astra12). • CentOS Linux release 8.2.2004 (Core) with glibc 2.28 (GNU libc). • CentOS Linux release 7.7.1908 (Core) with glibc 2.17 (GNU libc). • Debian 12 c with glibc 2.36 (Debian GLIBC 2.36-9+deb12u7). • Red Hat Enterprise Linux 9.5 (Plow) with glibc 2.34 (GNU libc). • Red OS 8. • Rocky Linux 9.3 (Blue Onyx) with glibc 2.34 (GNU libc) • Rocky Linux 9.5. • SuSE 15.6. • Ubuntu 24.04.2 LTS with glibc 2.39 (Ubuntu GLIBC 2.39-0ubuntu8.4) • Ubuntu 22.04.4 LTS with glibc 2.35 (Ubuntu GLIBC 2.35-0ubuntu3.8) • MosOS 15.5.

Hardware requirements for installing KUMA agents with WEC, ETW, and WMI transports

This functionality is only available in the commercial version of LogAnalyzer.

LogAnalyzer agents installed on Windows devices can use various types of transports to receive events: WEC (Windows Event Collector), ETW (Event Tracing for Windows), and WMI (Windows Management Instrumentation). The performance of the agents and their impact on the system depend on the limitations imposed by Windows and on the amount and type of events processed (see the table below).

Requirements for agents by transport type

Transport type	Maximum performance (EPS)	Limitations imposed by Windows	Limitations imposed by LogAnalyzer	Recommended agent configuration	CPU and RAM usage
WEC	Up to 2500 events per second	When writing to system logs (Application, System, Hardware Events), Windows limits the rate to approximately 2500 events per second. When writing to multiple logs at the same time, performance is further reduced: up to 1100 EPS with 2 logs, up to 800 EPS with 3 logs, and up to 500 EPS with 4 logs.	The agent can process events in the range of 1500–2500 EPS without lag.	CPU: 1 core with 2 threads or 2 vCPUs. RAM: 4 GB.	Average CPU load by the agent: up to 0.18 (equivalent to approximately 0.28% of the total system performance). RAM usage: up to 209 MB.
ETW	Up to 700 events per second	Only one system log is supported: **Microsoft → Windows → DNS-Server**. The Windows operating system does not allow writing more than 700 events per second to this log.	No limitations on the LogAnalyzer side have been identified, however, when EPS exceeds 700, the agent does not receive data due to Windows restrictions.	CPU: 1 core with 2 threads or 2 vCPUs. RAM: 4 GB.	Average CPU load by the agent: up to 0.267 (equivalent to approximately 0.42% of the total system performance). RAM usage: up to 186 MB.
WMI	Up to 5000 events per second	You can circumvent Windows restrictions by connecting multiple devices. Each additional device increases the total event volume that can be processed by 1000–1500 EPS.	Upon reaching 5000 EPS, the agent starts processing events with a lag.	CPU: 1 core with 2 threads or 2 vCPUs. RAM: 4 GB.	Average CPU load by the agent: up to 0.61 (equivalent to approximately 0.91% of the total system performance). RAM usage: up to 311 MB.

Hardware requirements for KUMA agents on Linux devices depending on EPS

LogAnalyzer agents installed on Linux devices can use various types of transports. The hardware requirements for LogAnalyzer agents depend on the type of transport and the EPS (see the table below).

Requirements for KUMA agents on Linux devices depending on EPS

Transport type	10,000 EPS	20,000 EPS	60,000 EPS	100,000 EPS
HTTP	CPU: 3 cores RAM: 160 MB.	CPU: 5 cores RAM: 160 MB.	CPU: 6 cores RAM: 260 MB.	CPU: 6 cores RAM: 300 MB.

Requirements for client devices for managing the LogAnalyzer web interface

CPU:

- Intel® Core™ i3, 8th generation, 32-bit
- AMD Ryzen 3, 32-bit

RAM: 8 GB

Supported browsers:

- Google™ Chrome™ 110 or later
- Mozilla™ Firefox™ 115 or later

Device requirements for installing LogAnalyzer on Kubernetes

Minimum configuration of a Kubernetes cluster for deployment of a high-availability LogAnalyzer configuration:

- 1 load balancer node (not part of the cluster)
- 3 controller nodes
- 2 worker nodes

The minimum hardware requirements for devices for installing LogAnalyzer on Kubernetes are listed in the table below.

Minimum hardware requirements for installing LogAnalyzer on Kubernetes

	Balancer	Controller	Worker node
CPU	1 core with 2 threads or 2 vCPUs.	1 core with 2 threads or 2 vCPUs.	12 threads or 12 vCPUs.
RAM	At least 2 GB	At least 2 GB	At least 24 GB
Free disk space	At least 30 GB	At least 30 GB	At least 1 TB in the /opt directory. At least 32 GB in the /var/lib directory.
Network bandwidth	10 Gbps	10 Gbps	10 Gbps

LogAnalyzer interface

The application is managed using the web interface.

The window of the application web interface contains the following:

- Sections in the left part of the application web interface window
- Tabs in the upper part of the application web interface window for some sections of the application
- The workspace in the lower part of the application web interface window

The workspace displays the information that you choose to view in the sections and on the tabs of the application web interface window. It also contains controls that you can use to configure the display of the information.

While managing the application web interface, you can use shortcut keys to perform the following actions:

- In all sections: close the window that opens in the right side pane—**Esc**.
- In the **Events** section:
  - Switch between events in the right side pane—↑ and ↓.
  - Start a search (when focused on the query field)—**Ctrl/Command**+**Enter**.
  - Save a search query—**Ctrl/Command**+**S**.

Compatibility with other applications

Kaspersky Endpoint Security for Linux

If LogAnalyzer components and the Kaspersky Endpoint Security for Linux application are installed on the same server, the report.db directory may grow very large and even take up the entire drive space. In addition, Kaspersky Endpoint Security for Linux scans all LogAnalyzer files by default, including service files, which may affect performance. To avoid these problems:

- Upgrade Kaspersky Endpoint Security for Linux to version 12.0 or later.
- We do not recommend enabling the network components of Kaspersky Endpoint Security for Linux.
- Add the following directories to general exclusions and to on-demand scan exclusions:

  a. On the LogAnalyzer Core server:

  - /opt/kaspersky/kuma/metrics/ — directory with data of the Metrics service.

  b. On the storage server:

  - /opt/kaspersky/kuma/clickhouse/ — the ClickHouse directory.
  - /opt/kaspersky/kuma/storage/<storage ID>/buffers/ — directory with storage buffers.

  c. On the collector server:

  - /opt/kaspersky/kuma/collector/<collector ID>/buffers — directory with buffers.
  - /opt/kaspersky/kuma/collector/<collector>/data/ — directory with dictionaries.

  d. Directories with logs for each service.

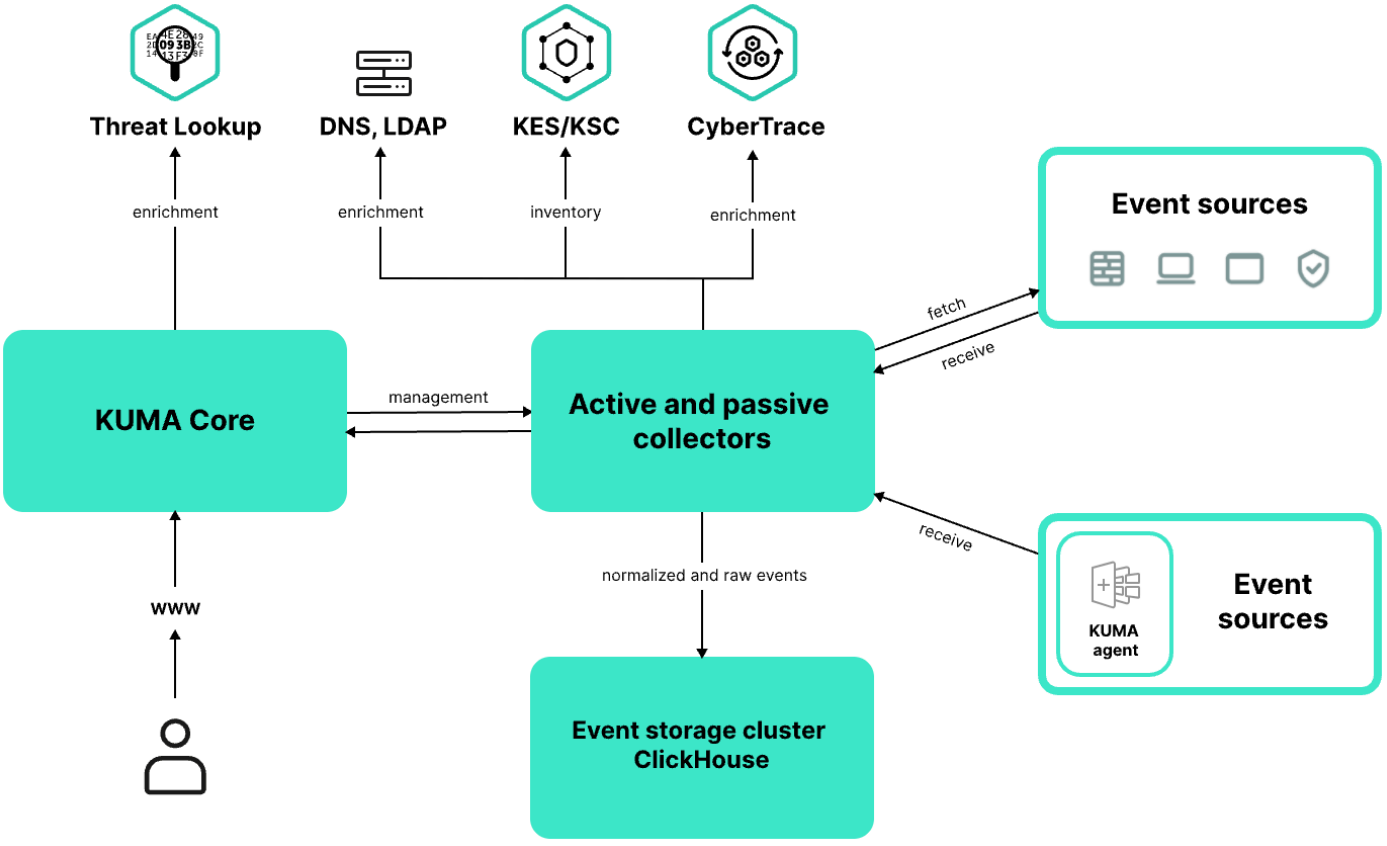
For more details on scan exclusions, please refer to the Kaspersky Endpoint Security for Linux Online Help.

Program architecture

The standard installation of the application includes the following components:

- The *Core* that includes a graphical user interface for monitoring and managing the settings of system components.
- *Agents* that are used to forward raw events from servers and workstations to LogAnalyzer destinations.
- One or more *Collectors* that receive messages from event sources and parse, normalize, and, if necessary, filter and/or aggregate them.
- *Event routers* that receive events from collectors and apply the configured filters to route the events to the configured destinations. In this way, these services balance the load on the network links.
- *Storage*, which contains normalized events.

Events are transmitted between components over optionally encrypted, reliable transport protocols. You can configure load balancing to distribute load between service instances, and you can also enable automatic switchover to a backup component if the primary component becomes unavailable. If all components are unavailable, events are saved to the hard disk buffer to be sent later. The size of the buffer in the file system for temporary storage of events can be changed.



LogAnalyzer architecture

Core

The *Core* is the central component of LogAnalyzer that serves as the foundation upon which all other services and components are built. The Core provides a graphical user interface that is intended for everyday use as well as for configuring the system as a whole.

The Core allows you to:

- Create and configure services (or components) of the application, as well as integrate the necessary software into the system.
- Manage services and user accounts of the application in a centralized way.
- Visualize application performance statistics.

Collector

A *collector* is an application component that receives messages from event sources, processes these messages, and sends them to a storage and/or third-party services.

For each collector, one connector and one normalizer must be configured. You can also configure any number of additional normalizers, filters, enrichment rules, and aggregation rules. For the collector to send normalized events to other services, you must add destinations.

The collector iterates over the following steps:

1. **Receive messages from event sources**

   To receive messages, you must configure an active or passive connector. A passive connector only listens for messages from an event source, while an active connector actively polls an event source, such as a database management system.

   Connectors can have different types. The choice of connector type depends on the transport protocol used for messaging. For example, if your event source sends messages over TCP, you must install a connector of the TCP type.

   The application has the following connector types available:

   - tcp
   - udp
   - nats-jetstream
   - kafka
   - kata/edr
   - http
   - sql

- file
- 1c-xml
- 1c-log
- diode
- ftp
- nfs
- vmware
- wmi
- wec
- snmp-trap
- elastic
- etw

2. **Parse and normalize events**

   Events received by the connector are processed using a normalizer and normalization rules set by the user. The choice of normalizer depends on the format of messages coming from the event source. For example, if your event source sends messages in the CEF format, you must select a normalizer of the CEF type.

   The following normalizers are available in the application:

   - JSON
   - CEF
   - Regexp
   - Syslog (as per RFC3164 and RFC5424)
   - CSV
   - Key-value
   - XML
   - SQL
   - IPFIX (v10)

3. **Filter normalized events**

   You can configure filters to identify events that satisfy certain conditions and only send such events for processing.

4. **Enrich and convert normalized events**

   Enrichment rules let you add supplementary information from internal and external sources to the events. The application can use the following enrichment sources:

- constants
- Dictionaries (available only in the commercial version of LogAnalyzer)
- cybertrace (available only in the commercial version of LogAnalyzer)
- cybertrace-http (available only in the commercial version of LogAnalyzer)
- dns (available only in the commercial version of LogAnalyzer)
- events
- templates
- timezone data
- geographic data
- DLL Hijacking scan

Mutation rules let you convert event field contents in accordance with the defined criteria. The application offers the following conversion methods:

- lower: convert all characters to lower case.
- upper: convert all characters to upper case.
- regexp: extract a substring using RE2 regular expressions.
- substring—gets a substring based on the specified numbers of the start and end positions.
- replace: replace some text with a string.
- trim: delete the specified characters.
- append: add characters to the end of the field value.
- prepend: adds characters to the beginning of the field value.

5. **Aggregate normalized events**

   You can configure aggregation rules to avoid sending many events of the same kind to the storage. Configuring aggregation rules lets you combine several events into one event. This helps you reduce the load on the services responsible for further event processing, conserves storage space and the license quota for events per second (EPS). For example, you can aggregate into one event all events involving network connections made using the same protocol (transport and application layers) between two IP addresses and received during a specified time interval.

6. **Send out normalized events**

   Having passed through all processing steps, the event is sent to the configured destinations.

Storage

A LogAnalyzer *storage* is used to store normalized events and ensure that LogAnalyzer can quickly and reliably access these events to extract analytical data. Access speed and high availability are made possible by the ClickHouse technology. This means that a *storage* is a ClickHouse cluster associated with a LogAnalyzer storage service. ClickHouse clusters can be supplemented with cold storage disks.

When choosing a ClickHouse cluster configuration, consider the specific event storage requirements of your organization. For more information, please refer to the ClickHouse documentation.

You can create *spaces* in repositories. Spaces let you structure the data in the cluster and, for example, store events of a certain type together.

Basic entities

This section introduces the entities that LogAnalyzer works with.

About tenants

LogAnalyzer has a multitenancy mode in which one instance of the LogAnalyzer application installed in the infrastructure of the main organization (main tenant) enables the isolation of branches (tenants) so as to allow them to receive and process their own events without affecting each other.

The system is managed centrally through the shared web interface, however, the tenants operate independently of each other and have access only to their own resources, services, and settings. Events of tenants are stored separately.

A user can have access to multiple tenants at the same time. You can also select the tenants whose data you want to be displayed in sections of the LogAnalyzer web interface.

Two tenants are created by default in LogAnalyzer:

- The Main tenant contains resources and services that belong to the main tenant. Only the general administrator has access to these resources.

- The Shared tenant is where the general administrator can place resources, asset categories, and monitoring policies that are available to users of all tenants. Access to the Shared tenant can be **restricted** for individual users.

  If in the settings of a user, the **Hide shared resources** check box is selected, that user cannot gain access to the Shared folder belonging to the Shared tenant in the LogAnalyzer web interface in the **Resources** → **<resource type>** section. This means that the user cannot view, edit, or otherwise use shared resources. The user is also unable to export shared resources and resource sets that incorporate resources from the Shared tenant, either through the web interface or through the REST API.

  If any of the services available to the user use shared resources, the names of these resources are displayed in the service settings, but user cannot view or modify the resources. The content of active lists is available to the user even if the resource of this active list is shared.

  The limitation does not apply to shared asset categories. Shared resources are also always available to users with the general administrator role.

About events

Events are information security events registered on the monitored elements of the corporate IT infrastructure. For example, events include login attempts, interactions with a database, and information sent by sensors. Each individual event may appear meaningless, but taken together, they paint a bigger picture of network activity that can help you identify security threats. This is the core functionality of LogAnalyzer.

LogAnalyzer receives events from logs and restructures them by bringing data from heterogeneous sources to a uniform format (this process is called normalization). The events are then filtered, aggregated, and sent to the storage service to be stored.

Throughout their life cycle, events undergo conversions and may be named differently. The following is an outline of the life cycle of a typical event:

The first steps are carried out in a collector.

1. Raw event. The original message from an event source received at a LogAnalyzer connector is called a *raw event*. This is an unprocessed message and it cannot be used yet by LogAnalyzer. To make it usable, it must be normalized to fit the LogAnalyzer data model. This happens at the next stage.

2. Normalized event. A normalizer transforms the data of the raw event to make it fit the LogAnalyzer data model. After this conversion, the original message becomes a *normalized event* and can be used by LogAnalyzer for analysis. From here on, only normalized events are used in LogAnalyzer. Raw events are no longer used, but they can be kept as a part of normalized events inside the `Raw` field.

   The application has the following normalizers:

   - JSON
   - CEF
   - Regexp
   - Syslog (as per RFC3164 and RFC5424)
   - CSV/TSV
   - Key-value
   - XML
   - SQL

   At this point, normalized events can already be used for analysis.

3. Event destination. After an event is processed by the collector, it is ready to be forwarded to LogAnalyzer storage.

The following event types are distinguished:

1. Base event. An event that has been normalized.

2. Aggregated event. When dealing with a large number of similar events, you can "merge" them into a single event to save processing time and resources. These act as base events and are processed in the same way, but In addition to all of the parameters of the parent events (the events that have been "merged"), an aggregated event has a counter that tells how many parent events it represents. Aggregated events also store the time when the first and last parent events were received.

3. Audit event. Audit events are created when certain security-related actions are completed in LogAnalyzer. These events are used to ensure system integrity. They are automatically placed in a separate storage space and stored for at least 365 days.

4. Monitoring event. These events are used to track changes in the amount of data received by LogAnalyzer.

About assets

Assets are network devices registered in LogAnalyzer. Assets generate network traffic when they send and receive data. LogAnalyzer can be configured to track this activity and create base events with a clear indication of where the traffic is coming from and where it is going. The event can contain source and destination IP addresses, as well as DNS names. If you register an asset with certain parameters (for example, a specific IP address), this asset is linked to all events that mention these parameters (IP address in this example).

About resources

Resources are LogAnalyzer components that contain parameters for implementing various functions: for example, establishing a connection with a given web address or converting data according to certain rules. These modular components are assembled into resource sets for services, which in turn are used to create LogAnalyzer services.

About services

Services are the main components of LogAnalyzer that handle events: receiving, processing, analyzing, and storing the events. Each service consists of two parts that work together:

- One part of the service is created in the LogAnalyzer web interface based on a resource set for services.
- The second part of the service is installed in the network infrastructure where the LogAnalyzer system is deployed as one of its components. The server part of a service can consist of multiple instances: for example, services of the same agent or storage can be installed on multiple devices at once.

The two parts of a service are linked to each other by the service ID.

About agents

LogAnalyzer *agents* are services that forward raw events from servers and workstations to LogAnalyzer destinations.

The following types of agents are provided:

- wmi agents are used to receive data from remote Windows devices using Windows Management Instrumentation. These are installed on Windows devices.
- wec agents are used to receive Windows logs from a local device using Windows Event Collector. They are installed to Windows assets.
- tcp agents receive data over TCP. These are installed on Linux and Windows devices.
- udp agents receive data over UDP. They are installed to Linux and Windows assets.
- nats-jetstream—used for NATS communications. They are installed to Linux and Windows assets.
- kafka agents are used for kafka communications. They are installed to Linux and Windows assets.

- http agents are used for communication over HTTP. They are installed to Linux and Windows assets.
- file agents get data from a file. They are installed to Linux and Windows assets.
- ftp agents receive data over the File Transfer Protocol. They are installed to Linux and Windows assets.
- nfs agents receive data over the Network File System protocol. They are installed to Linux and Windows assets.
- snmp agents receive data using the Simple Network Management Protocol. They are installed to Linux and Windows assets.
- diode agents are used together with data diodes to receive events from isolated network segments. They are installed to Linux and Windows assets.
- etw agents receive Event Tracing for Windows data. They are installed to Windows assets.

Program licensing

This section covers the main aspects of application licensing.

About the End User License Agreement

The *End User License agreement* is a legal agreement between you and AO Kaspersky Lab that specifies the conditions under which you can use the application.

Read the terms of the End User License Agreement carefully before using the application for the first time.

You can familiarize yourself with the terms of the End User License Agreement in the following ways:

- Go to the directory with the extracted installer and read the ./roles/kuma/files/LICENSE text file.
- Go to the directory with the extracted installer and run the following command to display the text of the End User License Agreement:

  ./roles/kuma/files/kuma license --show
- On a host with any LogAnalyzer component installed (such as Core, collector, storage), run the following command to display the text of the End User License Agreement:

  /opt/kaspersky/kuma/kuma license --show
- On devices included in the kuma_storage, kuma_collector, kuma_correlator, or kuma_core groups in the inventory file, open the LICENSE file located in the /opt/kaspersky/kuma directory.

  On a host in the kuma_core group, you can view the End User License Agreement only if a non-cluster installation is selected.
- On the Windows agent, run the following command to display the text of the End User License Agreement:

  .\kuma.exe license --show
- On the Linux agent, go to the directory with the 'kuma' executable file and run the following command to display the text of the End User License Agreement:

  ./kuma license --show

You accept the terms of the End User License Agreement by confirming your acceptance of the End User License Agreement during the application installation. If you do not accept the terms of the End User License Agreement, you must cease the installation of the application and must not use the application.

About the license

A *License* is a time-limited right to use the application, granted under the terms of the End User License Agreement.

A license entitles you to the following kinds of services:

- Use of the application in accordance with the terms of the End User License Agreement
- Getting technical support

The scope of services and the duration of usage depend on the type of license under which the application was activated.

A license is provided when the program is purchased. LogAnalyzer behavior in case no license is available:

- After the license expires, LogAnalyzer keeps working, but with limited functionality: collectors continue to receive and process events for 7 days, after which they restart and stop receiving new events. Old events remain available. Creating and editing resources, creating and running services also becomes impossible.
- If the license key is removed, LogAnalyzer collectors stop receiving and processing new events immediately. Old events remain available. Creating and editing resources, creating and running services also becomes impossible.

To continue using LogAnalyzer with its full functionality, you need to renew your license.

We recommend that you renew your license no later than its expiration date to ensure maximum protection against cyberthreats.

About the License Certificate

A *License Certificate* Is a document that is provided to you along with a key file or activation code.

The License Certificate contains the following information about the license being granted:

- License ID or order number
- Information about the user who is granted the license
- Information about the application that can be activated under the provided license
- Restriction on the number of licensing units (for example, the number of events that can be processed per second)
- Start date of the license term
- License expiration date or license term
- License type

About the license key

A *license key* is a sequence of bits that you can apply to activate and then use the application in accordance with the terms of the End User License Agreement. License keys are generated by Kaspersky specialists.

You can add a license key to the application by applying a *key file*. The license key is displayed in the application interface as a unique alphanumeric sequence after you add it to the application.

The license key may be blocked by Kaspersky in case the terms of the License Agreement have been violated. If the license key has been blocked, you need to add another one if you want to use the application.

A license key may be active or reserve.

An *active license key* is the license key currently used by the application. An active license key can be added for a trial or commercial license. The application cannot have more than one active license key.

A *reserve license key* is the license key that entitles the user to use the application but is not currently in use. The additional license key automatically becomes active when the license associated with the current active license key expires. An additional license key can be added only if an active license key has already been added.

A license key for a trial license can be added as an active license key. A license key for a trial license cannot be added as an additional license key.

About the key file

The *key file* is a file named license.key provided to you by Kaspersky. The key file is used to add a license key that activates the application.

You receive a key file at the email address that you provided after purchasing LogAnalyzer.

You do not need to connect to Kaspersky activation servers in order to activate the application with a key file.

If the key file has been accidentally deleted, you can restore it. You may need a key file, for example, to register with Kaspersky CompanyAccount.

To restore the key file, you need to do one of the following:

- Contact the license seller.
- Get a key file on the Kaspersky website based on the available activation code.

About the license code

A *license code* is a unique sequence of twenty Latin letters and numerals that allows you to activate the application. Kaspersky sends you a license code to the email address that you provided after purchasing LogAnalyzer.

Activation with a license code from the Core server requires constant internet access. To activate with a license code, you need a connection to Kaspersky activation servers:

https://activation-v2.kaspersky.com:443

In the case of a closed infrastructure, you can specify a proxy server.

If the license code was accidentally deleted, you can get it again by contacting the license vendor.

When the license code is deleted from LogAnalyzer, the LogAnalyzer collectors immediately stop receiving and processing new events. Old events remain available. Creating and editing resources, creating and running services also becomes impossible.

The web interface of the application displays settings depending on the functionality covered by the license.

If you want to use a license code to activate LogAnalyzer, in the **Settings** → **License** section, in the **Activation type** drop-down list, select **Activate by code**.

If you have activated the application using a key file and are adding a new license code, activation with the license code is performed seamlessly if the parameters of the license code fully match the parameters of the key file. If the parameters are different, the services are restarted.

LogAnalyzer generates an audit event after adding a license key, deleting a license key, or a license expiring.

When switching from a key file to a license code, the old license key is deleted automatically. Before renewing your license, make sure that you have access to the old key file.

Data provision in Kaspersky LogAnalyzer

Data provided to third parties

LogAnalyzer functionality does not involve automatic provision of user data to third parties.

Locally processed data

Kaspersky LogAnalyzer (hereinafter also referred to as "LogAnalyzer" or the "application") is an integrated software solution that combines the following functionality:

- Receiving, processing, and storing information security events
- Searching in received events
- Generating reports and dashboards for monitoring the state of the customer's infrastructure
- Monitoring event sources

To perform its primary functions, LogAnalyzer may receive, store and process the following information:

- Information contained in events from configured sources.

  In the collector, an event source is configured; events are generated and then transmitted to other services of the application. In some cases, the first point at which the event is received may be the agent service, which transmits events from the event source to the collector. Additionally, you can configure the saving of the address or host name of the event aggregator server.

- Information needed to integrate LogAnalyzer with other applications (Kaspersky Threat Lookup, Kaspersky CyberTrace).

  It can include certificates, tokens, URLs or credentials for establishing an encrypted connection with the other application, or other data necessary for the basic functionality of LogAnalyzer, for example, an email address. The user enters this data in the LogAnalyzer web interface.

- Information about sources from which event receipt is configured.

  It can include the source name, host name, IP address, the monitoring policy assigned to the source. The monitoring policy specifies the email address of the person responsible, to whom a notification will be sent if the policy is violated.

- User accounts: name, username, email address. The user can view their profile data in the LogAnalyzer web interface.

- User profile settings:

  - User role in LogAnalyzer. Assigned roles will be displayed.

  - Localization language, notification settings, display of non-printable characters.

    The user enters this data in the LogAnalyzer web interface.

  - Default dashboard, TV mode flag for the dashboard, SQL query for default events, default preset.

    The user specifies these settings in the corresponding sections of the LogAnalyzer web interface.

- Data for domain authentication of users in LogAnalyzer:

  - Active Directory: root DN for searching access groups in the Active Directory directory service, URL of the domain controller, certificate (the root public key that the AD certificate is signed with), full path to the access group of users in AD (distinguished name).

  - Active Directory Federation Services: trusted party ID (LogAnalyzer ID in AD FS), URI for getting Connect metadata, URL for redirection from AD FS, and the AD FS server certificate.

  - FreeIPA: Base DN, URL, certificate (the public root key that was used to signed the FreeIPA certificate), custom integration credentials, connection credentials.

- Audit events

  LogAnalyzer automatically records audit events.

- KUMA event log

  The user can enable extended logging in the LogAnalyzer console. Log entries are stored on the user's device, no data is transmitted automatically.

- Information about the user accepting the terms and conditions of legal agreements with Kaspersky.

- Information about current licenses in LogAnalyzer.

- Any information that the user enters in the LogAnalyzer interface.

The information listed above can find its way into LogAnalyzer in the following ways:

- The user enters this data in the LogAnalyzer web interface.

- LogAnalyzer services (agent or collector) receive data if the user has configured a connection to event sources.

- Through the LogAnalyzer REST API.

The listed information is stored in the LogAnalyzer database (ClickHouse, SQLite). Passwords are stored in an encrypted form (the hash of the password is stored).

All of the information listed above can be transmitted to Kaspersky only in dump files, trace files, or log files of LogAnalyzer components, including log files created by the installer and utilities.

Dump files, trace files, and log files of LogAnalyzer components may contain personal and confidential information. Dump files, trace files, and log files are stored on the device in unencrypted form. Dump files, trace files, and log files are not automatically submitted to Kaspersky, but the administrator can manually submit this information to Kaspersky at the request of Technical Support to help troubleshoot LogAnalyzer problems.

Kaspersky uses the collected data in anonymized form and only for general statistical purposes. Summary statistics is generated from the received raw data automatically and does not contain any personal or other confidential information. When new data accumulates, older data is erased (once a year). Summary statistics is stored indefinitely.

Kaspersky protects all received data in accordance with applicable law and Kaspersky policies. Data is transmitted over secure communication channels.

Adding a license key to the program web interface

You can add an application license key in the LogAnalyzer web interface. If you use a key file to activate the application, two license keys, active and reserve, may be present. If you use an activation code, there is only one license key. In LogAnalyzer, a group of an active and a reserve license keys or one activation code is called a *license group*. Accordingly, you can add a license group using two key files or using an activation code.

If multiple license groups are added to an instance of LogAnalyzer, the active license keys of all license groups are automatically merged. This creates a *merged license*. License keys of such a license are merged in accordance with the following rules:

- The **Available EPS** values of all active license keys are added together. The resulting EPS value is specified in the merged license and becomes available to all tenants.

  If any license key included in the merged license expires, the EPS limit value of that license key is automatically subtracted from the total EPS limit of the merged license. Accordingly, when a new active license key is added to the merged license, the EPS limit value of that license key is automatically added to the total EPS limit.

- The set of modules available in tenant licenses must be the same. If a license contains a module that is not available in the other licenses, this module is not available in the merged license. Accordingly, such a module is not available to any of the tenants.

  The same rule applies when adding new license keys to the merged license. If a module is available in the new license key, but is not available in the merged license, this module does not become available under the merged license. If a module that is available in the merged license is not available in the new license key, this module will become unavailable under the merged license. When adding a license key with a mismatched set of modules, LogAnalyzer displays a warning and adds such a license key only after your confirmation.

- A merged license can include up to 50 active keys.

Only users with the Administrator role can add a license key.

Adding a new license group

To add a new license group:

1. Open the LogAnalyzer web interface and select **Settings** → **Other** → **License**.

   This opens a window with a list of added license groups.

2. Click **Add license key**.

3. This opens a window; in that window, enter a name for the new license group.

4. Select the method for adding the license key:

   - Using an activation code.

     1. Enter the activation code.

     2. If your company uses a proxy server to connect to the internet, enter proxy server information. An internet connection is required to activate the application.

   - Using a key file.

     1. In the **Active license key** section, click the **Upload** button, then select the key file (file with the KEY extension).

     2. In the **Reserve license key** section, add the reserve license key in the same way.

5. Click the **Activate** button.

The license group is added to the list, and the application is activated.

Adding a reserve license key

To add a reserve license key to a license group:

1. Open the LogAnalyzer web interface and select **Settings** → **Other** → **License**.

   This opens a window with a list of added license groups.

2. Click the name of the license group to which you want to add the reserve license key.

3. This opens a window; in that window, in the **Reserve license key** section, click the **Upload** button, then select the key file (file with the KEY extension).

4. Click the **Activate** button.

Replacing the active license key

If you want to add an active license key instead of the key that has already been added, delete the current license key first. Then add the new license key.

Viewing information about an added license key in the program web interface

In the LogAnalyzer web interface, you can view information about the added license key. Information about the license key is displayed under **Settings** → **License**.

Only users with the Administrator role can view license information.

The **License** tab window displays the following information about added license keys:

- **Expires on**—date when the license key expires.
- **Days remaining**—number of days before the license is expired.
- **EPS available**—number of events processed per second supported by the license.
- **Current current per day**—current average number of unique events processed over 24 hours, calculated using the sliding window method. Every 15 seconds, LogAnalyzer counts the events processed by the collector (after filtration and aggregation at the output of the collector) to get the average EPS per hour, and then per day.
- **License key**—unique alphanumeric sequence.
- **Company**—name of the company that purchased the license.
- **Client name**—name of client who purchased the license.
- **Modules**—modules available for the license.

Additional settings for SMART I and SMART II licenses

For SMART I and SMART II licenses, the following settings are available in addition to the above:

- **EPS current per day** is the current average amount of events processed by LogAnalyzer per day.
- **EPS current per hour** is the current average amount of events processed by LogAnalyzer per hour.

If values of these two settings are exceeded, the collector with the maximum EPS value stops receiving new events for one hour. After one hour, the collector resumes operation. Multiple collectors may be paused at the same time. A notification about the maximum EPS allowed by the license being exceeded is sent to the user with the General Administrator role. The limitations apply only to SMART I and SMART II licenses.

Removing a license key in the program web interface

In LogAnalyzer, you can remove an added license key from the program (for example, if you need to replace the current license key with a different key). After the license key is removed, collectors immediately stop receiving and processing events. Old events remain available. Creating and editing resources, creating and running services also becomes impossible. This functionality will be re-activated the next time you add a license key.

Only users with the administrator role can delete license keys.

To delete an added license key:

1. Open the LogAnalyzer web interface and select **Settings** → **License**.
   This opens the window with the terms and conditions of the LogAnalyzer license.
2. Click the icon on the license key that you want to delete.
   A confirmation window opens.
3. Confirm deletion of the license key.

The license key will be removed from the application.

Administrator's guide

This section provides information about installing and configuring the LogAnalyzer SIEM system.

Installing and uninstalling LogAnalyzer

Warning! Installing the application in a high-availability configuration, including installation in a Kubernetes or Raft cluster, violates the certification conditions of the application.

To install KUMA, you need the distribution kit:

- **kuma-ansible-installer-<build number>.tar.gz** contains all necessary files for installing KUMA without Kubernetes. You can perform a distributed installation of LogAnalyzer Core in a cluster based on the Raft algorithm (hereinafter referred to as the "Raft cluster").

- **kuma-ansible-installer-<build number>-environment.tar.gz** contains components for installing the library for connecting to the Oracle database, Chromium, as well as components for automating the configuration and deployment of LogAnalyzer.

- **kuma-ansible-installer-k8s-<build number>.tar.gz** (not included in the certified version of the product) contains all the necessary files for installing KUMA in a high-availability configuration based on Kubernetes.

To complete the installation, you need the install.sh installer file and an inventory file that describes your infrastructure. You can create an inventory file based on a template. Each distribution contains an install.sh installer file and the following inventory file templates:

- single.inventory.yml.template

- distributed.inventory.yml.template

- expand.inventory.yml.template

- k0s.inventory.yml.template

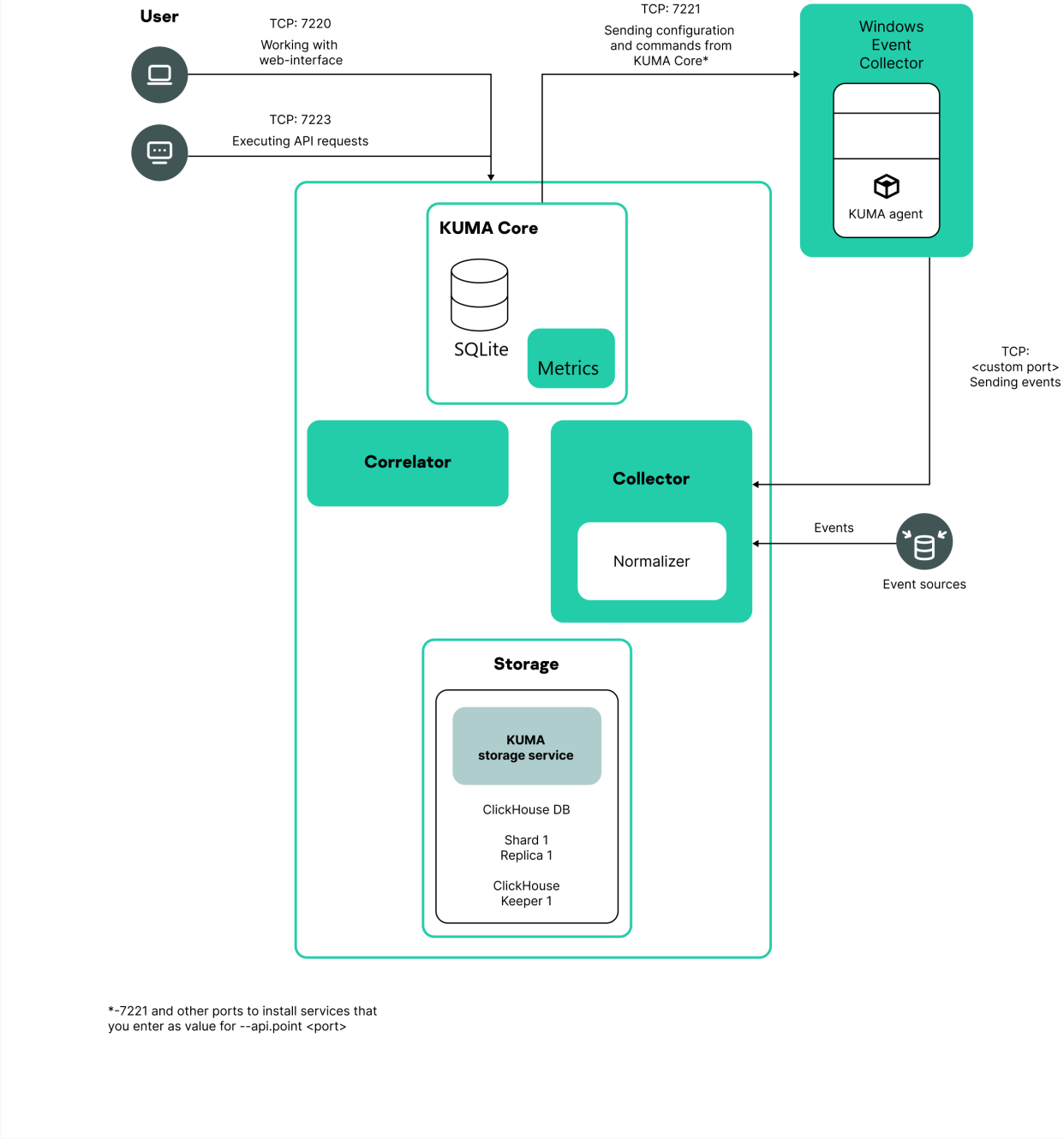
LogAnalyzer keeps its files in the /opt directory, so we recommend making /opt a separate partition and allocating 16 GB for the operating system and the remainder of the disk space for the /opt partition.

LogAnalyzer is installed in the same way on all hosts using the installer and your prepared inventory file in which you describe your configuration. We recommend taking time to think through the setup before you proceed.

The following installation options are available:

- Installation on a single server

  Single-server installation diagram



Installation on a single server

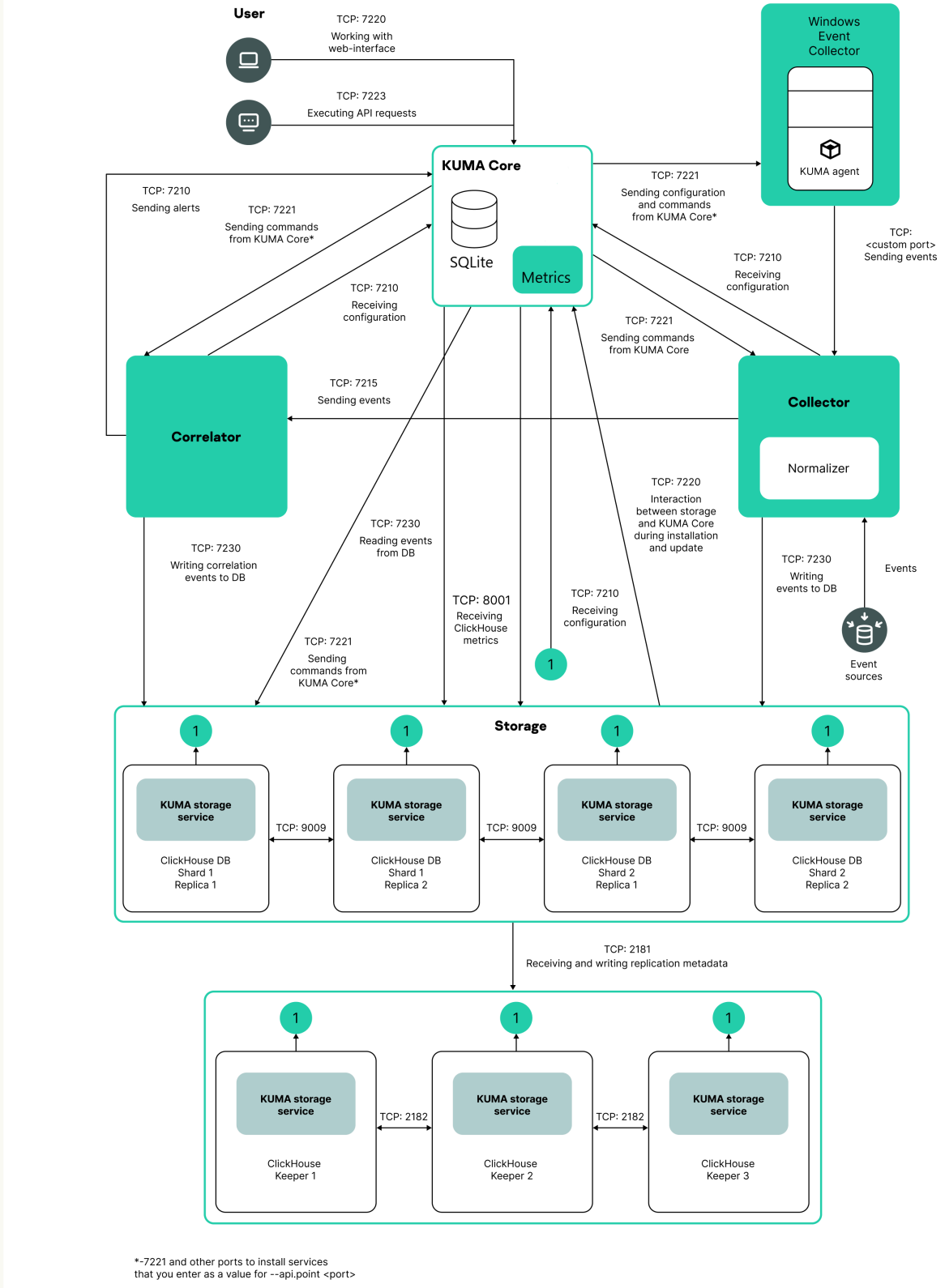
Example inventory file for installation on a single server

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: true
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma1.example.com:
    kuma_collector:
      hosts:
        kuma1.example.com
    kuma_correlator:
      hosts:
        kuma1.example.com
    kuma_storage:
      hosts:
        kuma1.example.com:
          shard: 1
          replica: 1
          keeper: 1
```

You can install all LogAnalyzer components on the same server: specify the same server in the single.inventory.yml inventory file for all components. An "all-in-one" installation can handle a small stream of events, up to 10,000 EPS. If you plan to use many dashboard layouts and process a lot of search queries, a single server might not be sufficient. In that case, we recommend the distributed installation.

- Distributed installation

Distributed Installation diagram



Distributed installation diagram

Example inventory file for distributed installation

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
          ip: 0.0.0.0
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ip: 0.0.0.0
    kuma_storage:
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server4.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server5.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 1
        kuma-storage-cluster1-server6.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 2
        kuma-storage-cluster1-server7.example.com:
          ip: 0.0.0.0
```

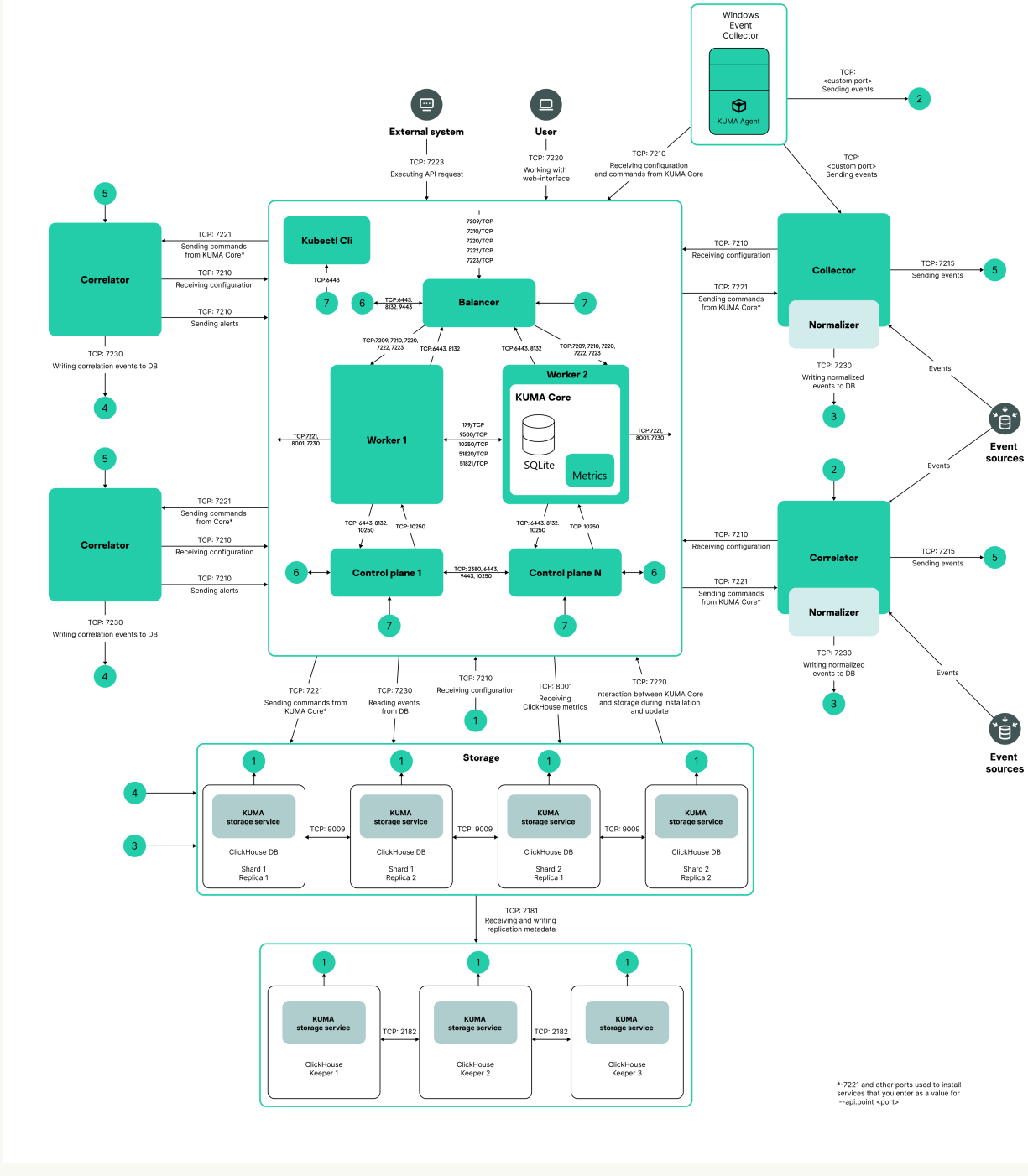
```
        shard: 0
        replica: 0
        keeper: 3
```

You can install LogAnalyzer services on different servers; you can describe the configuration for a distributed installation in the distributed.inventory.yml inventory file.

- Installation in a high availability configuration

  - Distributed installation with LogAnalyzer Core in a Kubernetes cluster

    Distributed installation with LogAnalyzer Core in a Kubernetes cluster



Distributed installation in a high availability configuration

Example inventory file for distributed installation with LogAnalyzer Core in a Kubernetes cluster

```
all:
  vars:
    deploy_to_k8s: true
    need_transfer: true
    generate_etc_hosts: false
    airgap: true
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ip: 0.0.0.0
        kuma-collector-2.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ip: 0.0.0.0
        kuma-correlator-2.example.com:
          ip: 0.0.0.0
    kuma_storage:
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server4.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server5.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 1
        kuma-storage-cluster1-server6.example.com:
          ip: 0.0.0.0
          shard: 0
```

```
          replica: 0
          keeper: 2
        kuma-storage-cluster1-server7.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 3
kuma_k0s:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_lb:
      hosts:
        kuma_lb.example.com:
          kuma_managed_lb: true
    kuma_control_plane_master:
      hosts:
        kuma_cpm.example.com:
          ansible_host: 10.0.1.10
    kuma_control_plane_master_worker:
    kuma_control_plane:
      hosts:
        kuma_cp1.example.com:
          ansible_host: 10.0.1.11
        kuma_cp2.example.com:
          ansible_host: 10.0.1.12
    kuma_control_plane_worker:
    kuma_worker:
      hosts:
        kuma-core-1.example.com:
          ansible_host: 10.0.1.13
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
        kuma_worker2.example.com:
          ansible_host: 10.0.1.14
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
```

You can install the LogAnalyzer Core on a Kubernetes cluster for high availability. Describe the configuration in the k0s.inventory.yml inventory file.

- Distributed installation with LogAnalyzer Core in a Raft cluster

  Distributed installation with LogAnalyzer Core in a Raft cluster

  Example distributed.inventory.yml inventory file for initial installation

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
          ip: 0.0.0.0
          raft_node_addr: ""
    kuma_core_peers:
      hosts:
        kuma-core-2.example.com:
          ip: 0.0.0.0
        kuma-core-3.example.com:
          ip: 0.0.0.0
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ip: 0.0.0.0
    kuma_storage:
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server4.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server5.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 1
```

```
        kuma-storage-cluster1-server6.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 2
        kuma-storage-cluster1-server7.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 3
```

Example inventory file for expanding the cluster using expand.sh

```
kuma:
  vars:
    no_firewall_actions: false
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core_peers:
      hosts:
        kuma-core-4.example.com:
          ip: 0.0.0.0
        kuma-core-5.example.com:
          ip: 0.0.0.0
```

You can install multiple LogAnalyzer Cores in a Raft cluster for data replication and high availability in a distributed system. Describe the configuration in the distributed.inventory.yml inventory file.

Program installation requirements

General application installation requirements

You can install the application on the following operating systems:

- Oracle Linux
- Astra Linux
- Ubuntu
- RED OS
- MosOS

  Supported operating system versions are listed in the Hardware and software requirements section.

  Supported configurations are Server and Server with GUI support. In the Server with GUI support configuration, you do not need to install additional packages for reporting.

  RED OS 8 is supported without high availability (HA). When using RED OS 8 in the Server with GUI support configuration, you need to install the iscsi-initiator-utils package, and then run the following commands:

  ```
  systemctl enable iscsid
  systemctl start iscsid
  ```

Before deploying the application, make sure the following conditions are met:

- Servers on which you want to install the components satisfy the hardware and software requirements.
- The ports to be used by the installed instance of LogAnalyzer are available.
- LogAnalyzer components are addressed using the fully qualified domain name (FQDN) of the host in the hostname.example.com format. Before you install the application, make sure that the correct host FQDN is returned in the `Static hostname` field. To do so, run the following command:

  ```
  hostnamectl status
  ```

- The server where the installer is run does not have the name `localhost` or `localhost.<domain>`.
- The name of the server on which you are installing LogAnalyzer Core does not start with a numeral.
- Time synchronization over Network Time Protocol (NTP) is configured on all servers with LogAnalyzer services.

Installation requirements for Oracle Linux, Astra Linux, Ubuntu, RED OS, and MosOS devices

	Oracle Linux	Astra Linux	Ubuntu	RED OS	MosOS
Python version	3.6 to 3.11. Hosts with Python earlier than 3.6 (inclusive) cannot be combined in the same instance with Python 3.10 and later hosts.	3.6 to 3.11. Hosts with Python earlier than 3.6 (inclusive) cannot be combined in the same instance with Python 3.10 and later hosts.	3.6 to 3.11. Hosts with Python earlier than 3.6 (inclusive) cannot be combined in the same instance with Python 3.10 and later hosts.	3.6 to 3.11. Hosts with Python earlier than 3.6 (inclusive) cannot be combined in the same instance with Python 3.10 and later hosts.	3.6 to 3.11. Hosts with Python earlier than 3.6 (inclusive) cannot be combined in the same instance with Python 3.10 and later hosts.
SELinux module	Disabled	Disabled	Disabled	Disabled	Disabled
Package manager	pip3	pip3	pip3	pip3	pip3

Basic packages	• netaddr
	• firewalld
	More on upgrading from Oracle Linux 8.x to Oracle Linux 9.x.
	To install the packages, run the following commands:
	`pip3 install netaddr`
	`yum install firewalld`
		• python3-apt
		• curl
		• libcurl4
		To install the packages, run the following command:
		`apt install python3-apt curl libcurl4`
			• python3-apt
			• curl
			• libcurl4
			To install the packages, run the following command:
			`apt install python3-apt curl libcurl4`
				• netaddr
				• firewalld
				To install the packages, run the following commands:
				`pip3 install netaddr`
				`dnf install firewalld`
					• python3-rpm
					To install the package, run the following commands:
					`zypper refresh`
					`zypper install python3-rpm`

	Oracle Linux	Astra Linux	Ubuntu	RED OS	MosOS
Dependent packages	No value.	• netaddr • python3-cffi-backend To install the packages, run the following command: `apt install python3-netaddr python3-cffi-backend` If you plan to query Oracle DB databases from LogAnalyzer, you need to install the Astra Linux libaio1 package.	• netaddr • python3-cffi-backend To install the packages, run the following command: `apt install python3-netaddr python3-cffi-backend`	No value.	No value.

	Oracle Linux	Astra Linux	Ubuntu	RED OS	MosOS

Packages that must be installed on a device with the LogAnalyzer Core for correct generation and downloading of reports	• nss
	• gtk2
	• atk
	• libnss3.so
	• libatk-1.0.so.0
	• libxkbcommon
	• libdrm
	• at-spi2-atk
	• mesa-libgbm
	• alsa-lib
	• cups-libs
	• libXcomposite
	• libXdamage
	• libXrandr
	
	To install the packages, run the following command:
	`yum install nss gtk2 atk libnss3.so libatk-1.0.so.0 libxkbcommon libdrm at-spi2-atk mesa-libgbm alsa-lib cups-libs libXcomposite libXdamage libXrandr`
		• libgtk2.0.0
		• libnss3
		• libatk-adaptor
		• libdrm-common
		• libgbm1
		• libxkbcommon0
		• libasound2
		
		To install the packages, run the following command:
		`apt install libgtk2.0.0 libnss3 libatk-adaptor libdrm-common libgbm1 libxkbcommon0 libasound2`
			• libatk1.0-0
			• libgtk2.0-0
			• libatk-bridge2.0-0
			• libcups2
			• libxcomposite-dev
			• libxdamage1
			• libxrandr2
			• libgbm-dev
			• libxkbcommon-x11-0
			• libpangocairo-1.0-0
			• libasound2 (for Ubuntu 22.04 LTS) or libasound2t64 (for Ubuntu 24.04 LTS).
			
			You can install these packages using the following command (for Ubuntu 22.04 LTS):
			`apt install libatk1.0-0 libgtk2.0-0 libatk-bridge2.0-0 libcups2 libxcomposite-dev libxdamage1 libxrandr2 libgbm-dev libxkbcommon-x11-0 libpangocairo-1.0-0 libasound2`
			
			For Ubuntu 24.04 LTS, run the following command:
			`apt install libatk1.0-0 libgtk2.0-0 libatk-bridge2.0-0 libcups2 libxcomposite-dev libxdamage1 libxrandr2 libgbm-dev libxkbcommon-x11-0 libpangocairo-1.0-0 libasound2t64`
				• nss
				• gtk2
				• atk
				• libnss3.so
				• libatk-1.0.so.0
				• libxkbcommon
				• libdrm
				• at-spi2-atk
				• mesa-libgbm
				• alsa-lib
				• cups-libs
				• libXcomposite
				• libXdamage
				• libXrandr
				
				To install the packages, run the following command:
				`dnf install nss gtk2 atk libnss3.so libatk-1.0.so.0 libxkbcommon libdrm at-spi2-atk mesa-libgbm alsa-lib cups-libs libXcomposite libXdamage libXrandr`
					No value.

User permissions required to install the application	No value.	You need to assign the required permissions to the user that will be installing the application: `sudo pdpl-user -i 63 < user that will be installing the application >`	No value.	No value.	No value.

Upgrading from Oracle Linux 8.x to Oracle Linux 9.x

To upgrade from Oracle Linux 8.x to Oracle Linux 9.x:

1. Run the following commands to disable LogAnalyzer services on the hosts where the services are installed:
   - `sudo systemctl disable kuma-collector-<service ID>.service`
   - `sudo systemctl disable kuma-storage-<service ID>.service`
   - `sudo systemctl disable kuma-metrics-<metrics service ID>.service`
   - `sudo systemctl disable kuma-vmalert.service`
   - `sudo systemctl disable kuma-core-<KUMA Core service ID>.service`
2. Upgrade the OS on every host.
3. Run the following commands to enable the services on the hosts where the services are installed:
   - `sudo systemctl enable kuma-core-<KUMA Core service ID>.service`
   - `sudo systemctl enable kuma-storage-<service ID>.service`
   - `sudo systemctl enable kuma-collector-<service ID>.service`
   - `sudo systemctl enable kuma-metrics-<metrics service ID>.service`
   - `sudo systemctl enable kuma-vmalert.service`
4. Restart the hosts.

As a result, the upgrade is completed.

Ports used by LogAnalyzer during installation

For the application to run correctly, you need to ensure that the LogAnalyzer components are able to interact with other components and applications over the network using the protocols and ports specified during the installation of the LogAnalyzer components.

Before installing the Core on the device, make sure that the following ports are free:

- 7227: used by the Metrics service.
- 8880: used by VMalert.

The table below lists the default ports. The installer automatically opens the ports during LogAnalyzer installation

Network ports used for the interaction of LogAnalyzer components

Protocol	Port	Direction	Purpose of the connection
HTTPS	7222	From the LogAnalyzer client to the server with the LogAnalyzer Core component.	Reverse proxy to the CyberTrace system.
HTTPS	8123	Local requests from the storage service to the local node of the ClickHouse cluster.	Writing and getting normalized events in the ClickHouse cluster.
HTTPS	9009	Between replicas of the ClickHouse cluster.	Internal data communication between replicas of the ClickHouse cluster.
TCP	2181	From ClickHouse cluster nodes to the ClickHouse keeper replication coordination service.	Getting and writing replication metadata by replicas of ClickHouse servers.
TCP	2182	From one ClickHouse keeper replication coordination service to another.	Internal communication between replication coordination services to reach a quorum.
TCP	7210	From all LogAnalyzer components to the LogAnalyzer Core server.	Getting the LogAnalyzer configuration from the LogAnalyzer Core server.
TCP	7220	• From the LogAnalyzer client to the server with the LogAnalyzer Core component. • From storage hosts to the KUMA Core server during installation or upgrade.	• User access to the LogAnalyzer web interface. • Interaction between the storage hosts and the KUMA Core during installation or upgrade. You can close this port after completing the installation or upgrade.
TCP	7221 and other ports used for service installation as the --api.port <port> parameter value	From the LogAnalyzer Core to LogAnalyzer services.	Administration of services from the LogAnalyzer web interface.
TCP	7223	To the LogAnalyzer Core server.	Default port for API requests.
TCP	9000	• Outgoing and incoming connections between servers of the ClickHouse cluster. • From the local client.sh client to the local cluster node.	Port of the ClickHouse native protocol (also called *ClickHouse TCP*). Used by ClickHouse applications and processes, such as clickhouse-server, clickhouse-client, and native ClickHouse tools Used for inter-server communication for distributed queries. Also used for writing and getting data in the ClickHouse cluster.

Ports used by predefined OOTB resources

The installer automatically opens the ports during LogAnalyzer installation

Ports used by predefined OOTB resources:

- 7230/tcp
- 7231/tcp
- 7232/tcp
- 7233/tcp
- 7234/tcp
- 7235/tcp

- 5140/tcp
- 5140/udp
- 5141/tcp
- 5144/udp

LogAnalyzer Core traffic in a high availability configuration

The table below lists the initiator (source) of the connection and the destination. The port number on the initiator can be dynamic. Return traffic within the established connection must not be blocked.

LogAnalyzer Core traffic in a high availability configuration

Source	Destination	Destination port	Type
External LogAnalyzer services	Load balancer	7209	TCP
External LogAnalyzer services	Load balancer	7210	TCP
External LogAnalyzer services	Load balancer	7220	TCP
External LogAnalyzer services	Load balancer	7222	TCP
External LogAnalyzer services	Load balancer	7223	TCP
Worker node	Load balancer	6443	TCP
Worker node	Load balancer	8132	TCP
Control node	Load balancer	6443	TCP
Control node	Load balancer	8132	TCP
Control node	Load balancer	9443	TCP
Worker node	External LogAnalyzer services	Depending on the settings specified when creating the service.	TCP
Load balancer	Worker node	7209	TCP
Load balancer	Worker node	7210	TCP
Load balancer	Worker node	7220	TCP
Load balancer	Worker node	7222	TCP
Load balancer	Worker node	7223	TCP
External LogAnalyzer services	Worker node	7209	TCP
External LogAnalyzer services	Worker node	7210	TCP
External LogAnalyzer services	Worker node	7220	TCP
External LogAnalyzer services	Worker node	7222	TCP
External LogAnalyzer services	Worker node	7223	TCP
Worker node	Worker node	179	TCP
Worker node	Worker node	9500	TCP
Worker node	Worker node	10250	TCP
Worker node	Worker node	51820	UDP
Worker node	Worker node	51821	UDP
Control node	Worker node	10250	TCP
Load balancer	Control node	6443	TCP
Load balancer	Control node	8132	TCP
Load balancer	Control node	9443	TCP

Source	Destination	Destination port	Type
Worker node	Control node	6443	TCP
Worker node	Control node	8132	TCP
Worker node	Control node	10250	TCP
Control node	Control node	2380	TCP
Control node	Control node	6443	TCP
Control node	Control node	9443	TCP
Control node	Control node	10250	TCP
Cluster management console (CLI)	Load balancer	6443	TCP
Cluster management console (CLI)	Control node	6443	TCP

Downloading CA certificates

In the LogAnalyzer web interface, you can download the following CA certificates:

- REST API CA certificate.

  This certificate is used to authenticate the API server serving the LogAnalyzer public API. You can also use this certificate when importing data from MaxPatrol reports.

  You can also change this certificate if you want to use your company's certificate and key instead of the self-signed certificate of the web console.

  For an installation with the LogAnalyzer Core service in a Raft cluster without a balancer, you can download the CA certificate of the REST API to any node, but a secure connection is possible only with the core-1 node. The rest of the nodes return an error. If you want to directly log in with the same certificate to all nodes of the Raft cluster without a balancer, when generating a certificate, you can specify the FQDNs of all LogAnalyzer Core nodes in the Raft cluster in the Subject Alternative Names (SAN) field.

- Microservice CA certificate.

  This certificate is used for authentication when connecting log sources to passive collectors using TLS, but without specifying your own certificate.

To download a CA certificate:

1. Open the LogAnalyzer web interface.
2. In the lower left corner of the window, click the name of the user account, and in the menu, click the **REST API CA certificate** or **Microservice CA certificate** button, depending on the certificate that you want to download.

The certificate is saved to the download directory configured in your browser.

Reissuing internal CA certificates

The storage location of the self-signed CA certificate and the certificate reissue mechanism have been changed.
The certificate is stored in the database. The previous method of reissuing internal certificates by deleting certificates from the file system of the Core and restarting the Core is no longer allowed. The old method will cause the Core to fail to start. Until the certificate reissuing process is completed, new services may not be connected to the Core.
After reissuing the internal CA certificates in the **Settings** → **Other** → **General** → **Reissue internal CA certificates** section of the KUMA web interface, you must stop the services, delete the old certificates from service directories, and manually restart all services. Only users with the General Administrator role can reissue internal CA certificates.

The **Reissue internal CA certificates** option is available only to a user with the General Administrator role.

The process of reissuing certificates for an individual service remains the same: in the LogAnalyzer web interface, in the **Resources** → **Active services** section, select the service; in the context menu, select **Reset certificate**, and delete the old certificate from the service installation directory. LogAnalyzer automatically generates a new certificate. You do not need to restart running services, the new certificate is applied automatically. A stopped service must be restarted to have the certificate applied.

To reissue internal CA certificates:

1. In the LogAnalyzer web interface, go to the **Settings** → **Other** → **General** section, click the **Reissue internal CA certificates** button, and review the displayed warning. If you decide to proceed with reissuing certificates, click **Yes**.

   As a result, the CA certificates for LogAnalyzer services and the CA certificate for ClickHouse are reissued. Next, you must stop the services, delete old certificates from the service installation directories, restart the Core, and restart the stopped services to apply the reissued certificates.

2. Connect to the hosts where the collector, correlator, and event router services are deployed.

   a. Stop all services with the following command:

   ```
   sudo systemctl stop kuma-<collector/correlator/eventRouter/metrics>-<service ID>.service
   ```

   b. Delete the internal.cert and internal.key certificate files from the "/opt/kaspersky/kuma/<service type>/<service ID>/certificates" directories with the following command:

   ```
   sudo rm -f /opt/kaspersky/kuma/<service type>/<service ID>/certificates/internal.cert
   sudo rm -f /opt/kaspersky/kuma/<service type>/<service ID>/certificates/internal.key
   ```

3. Connect to the hosts where storage services are deployed.

   a. Stop all storage services.

   ```
   sudo systemctl stop kuma-<storage>-<service ID>.service
   ```

   b. Delete the internal.cert and internal.key certificate files from the "/opt/kaspersky/kuma/storage/<ID service>/certificates" directories.

   ```
   sudo rm -f /opt/kaspersky/kuma/storage/<service ID>/certificates/internal.cert
   sudo rm -f /opt/kaspersky/kuma/storage/<service ID>/certificates/internal.key
   ```

4. Delete all ClickHouse certificates from the "/opt/kaspersky/kuma/clickhouse/certificates" directory.

   ```
   sudo rm -f /opt/kaspersky/kuma/clickhouse/certificates/internal.cert
   sudo rm -f /opt/kaspersky/kuma/clickhouse/certificates/internal.key
   ```

5. Connect to the hosts where agent services are deployed.

   a. Stop the services of Windows agents and Linux agents.

   b. Delete the internal.cert and internal.key certificate files from working directories of the agents.

6. Start the LogAnalyzer Core to apply the new CA certificates.

   - For an "all-in-one" or distributed installation of LogAnalyzer, run the following command:

     ```
     sudo systemctl restart kuma-core-00000000-0000-0000-0000-000000000000.service
     ```

   - For LogAnalyzer with a Core in a Kubernetes cluster, to restart the LogAnalyzer Core, run the following commands on any controller:

     ```
     sudo k0s kubectl exec deployment/metrics -n kuma -- rm -rf /opt/kaspersky/kuma/metrics/certificates
     sudo k0s kubectl rollout restart deployment metrics -n kuma
     sudo k0s kubectl rollout restart deployment/core-deployment -n kuma
     ```

     The LogAnalyzer Core must be restarted using the command because restarting the LogAnalyzer Core in the LogAnalyzer interface affects only the LogAnalyzer Core container and not the entire pod.

7. Restart all services that were stopped during the procedure.

   ```
   sudo systemctl start kuma-<collector/correlator/eventRouter/storage/metrics>-<service ID>.service
   ```

Internal CA certificates are reissued and applied.

Modifying the self-signed web console certificate

You can use your company's certificate and key instead of the self-signed certificate of the web console. For example, if you want to replace self-signed CA Core certificate with a certificate issued by an enterprise CA, you must provide an external.cert and an unencrypted external.key in PEM format.

The following example shows how to replace a self-signed CA Core certificate with an enterprise certificate in PFX format. You can use the instructions as an example and adapt the steps according to your needs.

To replace the LogAnalyzer web console certificate with an external certificate:

1. If you are using a certificate and key in a PFX container, use OpenSSL to convert the PFX file to a certificate and encrypted key in PEM format:

   ```
   openssl pkcs12 -in kumaWebIssuedByCorporateCA.pfx -nokeys -out external.cert
   openssl pkcs12 -in kumaWebIssuedByCorporateCA.pfx -nocerts -nodes -out external.key
   ```

   When carrying out the command, you are required to specify the PFX key password (Enter Import Password).

   As a result, the external.cert certificate and the external.key in PEM format are returned.

2. If your external.key is not encrypted, make sure that the key has the PKCS#8 format. LogAnalyzer will not accept a key file that begins with the BEGIN RSA PRIVATE KEY tag. In this case, convert the file using the following command:

   ```
   openssl rsa -in rsa_private.key -outform PEM -out private.key
   ```

   The file produced by this command begins with the BEGIN PRIVATE KEY tag, and LogAnalyzer accepts such a key file.

3. In the LogAnalyzer web interface, go to the **Settings** → **Other** → **General** → **Core settings** section. Under **External TLS pair**, click **Upload certificate** and **Upload key** and upload the external.cert file and the unencrypted external.key file in PEM format.

4. Restart LogAnalyzer:

   ```
   systemctl restart kuma-core
   ```

5. Refresh the web page or restart the browser that you are using to manage the LogAnalyzer web interface.

Your company certificate and key have been replaced.

Synchronizing time on servers

To configure time synchronization on servers:

1. Install chrony:

   ```
   sudo apt install chrony
   ```

2. Configure the synchronization of system time with an NTP server:

   a. Make sure the virtual machine has Internet access.

      If access is available, go to step b.

      If the virtual machine does not have internet access, edit the `/etc/chrony.conf` file to replace `2.pool.ntp.org` with the name or IP address of your corporate NTP server.

   b. Start the system time synchronization service:

      ```
      sudo systemctl enable --now chronyd
      ```

   c. Wait a few seconds and run the following command:

      ```
      sudo timedatectl | grep 'System clock synchronized'
      ```

      If the system time is synchronized correctly, the output will contain "System clock synchronized: yes".

Synchronization is configured.

About the inventory file

You can install, update, or remove KUMA components by changing to the directory with the extracted kuma-ansible-installer and using the Ansible tool and a prepared inventory file. You can specify LogAnalyzer configuration settings in the inventory file; the installer then uses these settings when deploying, updating, and removing the application. The inventory file must conform to the YAML format.

You can create an inventory file based on the templates included in the distribution kit. The following templates are provided:

- single.inventory.yml.template can be used when installing LogAnalyzer on a single server. This template contains the minimum set of settings optimized for installation on a single device without using a Kubernetes cluster.
- distributed.inventory.yml.template can be used for the initial distributed installation of LogAnalyzer without using a Kubernetes cluster, for expanding an all-in-one installation to a distributed installation, and for updating LogAnalyzer.
- expand.inventory.yml.template can be used in some reconfiguration scenarios, such as adding collector servers, expanding an existing storage cluster, or adding a new storage cluster. If you use this inventory file to edit the configuration, the installer does not stop services in the entire infrastructure. If you reuse the inventory file, the installer can stop only services on hosts that are listed in the expand.inventory.yml inventory file.
- k0s.inventory.yml.template can be used to install or migrate LogAnalyzer to a Kubernetes cluster.

We recommend saving a backup copy of the inventory file that you used to install the application. You can use it to add components to the system or remove LogAnalyzer.

LogAnalyzer settings in the inventory file

The inventory file may include the following blocks:

- `all`
- `kuma`
- `kuma_k0s`

For each host, you must specify the FQDN in the <`host name`>.<`domain`> format and, if necessary, an IPv4 or IPv6 address. The LogAnalyzer Core domain name and its subdomains may not start with a numeral.

Example:

```
hosts:
  hostname.example.com:
    ip: 0.0.0.0
```

The 'all' block

In this block, you can specify the variables that apply to all hosts listed in the inventory file, including the implicitly specified localhost on which the installation is started. Variables can be overridden at the level of host groups or individual hosts.

Example of overriding variables in the inventory file

```
all:
  vars:
    ansible_connection: ssh
    deploy_to_k8s: False
    need_transfer: False
    airgap: True
    deploy_example_services: True
kuma:
  vars:
    ansible_become: true
    ansible_user: i.ivanov
    ansible_become_method: su
    ansible_ssh_private_key_file: ~/.ssh/id_rsa
  children:
    kuma_core:
      vars:
        ansible_user: p.petrov
        ansible_become_method: sudo
```

The table below lists all possible variables in the `vars` section and their descriptions.

List of possible variables in the vars section

Variable	Description
`ansible_connection`	Method used to connect to target machines. Possible values: • `ssh` to connect to remote hosts over SSH. • `local` to establish no connection with remote hosts.
`ansible_user`	User name used to connect to target machines and install components. If root login is blocked on the target machines, choose a user that has the right to establish SSH connections and elevate privileges using su or sudo.

Variable	Description
`ansible_become`	This variable specifies if you want to elevate the privileges of the user that is used to install LogAnalyzer components. Possible values: • You must specify `true` if `ansible_user` is not `root`. • `false`.
`ansible_become_method`	Method for elevating the privileges of the user that is used to install LogAnalyzer components. You must specify `su` or `sudo` if `ansible_user` is not `root`.
`ansible_ssh_private_key_file`	Path to the private key in the /<path>/.ssh/id_rsa format. You must specify this variable if you want to use a key file other than the default key file (~/.ssh/id_rsa).
`deploy_to_k8s`	This variable specifies whether you want to deploy LogAnalyzer components in a Kubernetes cluster. Possible values: • The `false` value is specified in the single.inventory.yml and distributed.inventory.yml templates. • The `true` value is specified in the k0s.inventory.yml template. If you do not specify this variable, it defaults to `false`.
`need_transfer`	This variable specifies whether you want to migrate LogAnalyzer Core to a new Kubernetes cluster. You need to specify this variable only if `deploy_to_k8s` is `true`. Possible values: • `false` means LogAnalyzer must be installed or updated on the Kubernetes cluster. • `true` means LogAnalyzer Core must be migrated to a new Kubernetes cluster. If you do not specify this variable, it defaults to `false`.
`no_firewall_actions`	This variable specifies whether the installer must perform the steps to configure the firewall on the hosts. Possible values: • `true` means that at startup, the installer does not perform the steps to configure the firewall on the hosts. • `false` means that at startup, the installer performs the steps to configure the firewall on the hosts. This is the value that is specified in all inventory file templates. If you do not specify this variable, it defaults to `false`.
`generate_etc_hosts`	This variable specifies whether the machines must be registered in the DNS zone of your organization. The installer automatically adds the IP addresses of the machines from the inventory file to the /etc/hosts files on the machines on which LogAnalyzer components are installed. The specified IP addresses must be unique. Possible values: • `false`. • `true`. If you do not specify this variable, it defaults to `false`. When using `generate_etc_hosts: true` in the inventory file for a high-availability configuration, you must specify the balancer IP address in the `ansible_host` variable for the balancer host in the `kuma_lb` section.
`deploy_example_services`	This variable specifies whether predefined services are created during the installation of LogAnalyzer. You need to specify this variable if you want to create demo services independently of the single/distributed/k0s inventory file. Possible values: • `false` means predefined services are not created when installing LogAnalyzer. This is the value that is specified in all inventory file templates. • `true` means predefined services are created when installing LogAnalyzer. If you do not specify this variable, it defaults to `false`.

Variable	Description
`low_resources`	This variable specifies whether KUMA is being installed in an environment with limited computational resources. This variable is not specified in any of the inventory file templates. Possible values: • `false` means LogAnalyzer is being installed for production use. In this case, the installer checks the requirements of the worker nodes (CPU, RAM, and free disk space) in accordance with the hardware and software requirements. If the requirements are not satisfied, the installation is stopped with an error message. • `true` means that LogAnalyzer is being installed in an environment with limited computational resources. In this case, the minimum size of the LogAnalyzer Core installation directory on the host is 4 GB. All other computational resource limitations are ignored. If you do not specify this variable, it defaults to `false`.
`core_disk_request`	This variable sets the size of the LogAnalyzer Core volume for an installation in a Kubernetes cluster. If this variable is not set, the following default values are used for the LogAnalyzer Core volume size during installation: • If `low_resources=false`, the default volume size is 512Gi. • If `low_resources=true`, the default volume size is 4Gi. The 'Gi' suffix is required.
`metrics_disk_request`	This variable sets the size of the Metrics service volume for an installation in a Kubernetes cluster. If this variable is not set, the following default values are used for the Metrics server volume size during installation: • If `low_resources=false`, the default volume size is 80Gi. • If `low_resources=true`, the default volume size is 4Gi. The 'Gi' suffix is required.
`metrics_transfer_job_timeout`	This variable sets the timeout in minutes for copying metrics from a LogAnalyzer 3.4 Core volume to a new metrics volume in LogAnalyzer 4.0. The default setting is 120m. The 'm' suffix is required.

The 'kuma' block

In this block, you can specify the settings of KUMA components deployed outside of the Kubernetes cluster. The `kuma` block can contain the following sections:

- `vars` contains variables that apply to all hosts specified in the `kuma` block.
- `children` contains groups of settings for components:
  - `kuma_core` contains LogAnalyzer Core settings.
    - The `ansible_port` variable lets you specify a non-standard SSH ports during installation. Must be specified for each host in the inventory file, even if the same ports are used on all hosts. Optional setting.
    - `raft_node_addr` is the address to be used for the cluster of nodes with the LogAnalyzer Core service. This address must be public and known to other nodes. After the LogAnalyzer Core is started for the first time with this address, the address cannot be changed anymore for this node. This value must be specified in the <FQDN>:<`port`> format. If this setting is not specified explicitly, <FQDN> defaults to the FQDN of the host on which the LogAnalyzer Core is deployed, and <`port`> defaults to 7209. The `raft_node_addr` setting must be specified together with the `raft_node_listen setting on each LogAnalyzer Core server with if DNS is configured in such a way that the server contains multiple FQDNs`. Optional setting.
    - `raft_node_listen` is the address at which the node listens for messages from its peers in the cluster. This setting must be specified in the following format <`IPv4 or IPv6 address of the LogAnalyzer Core server`>:<`port`>. This address can be changed. The `raft_node_listen` setting must be specified together with the `raft_node_addr setting on every LogAnalyzer Core server`. Optional setting.
  - `kuma_core_peers` contains settings of additional LogAnalyzer Cores. The installer prepares the servers specified in the `kuma_core_peers` group for the subsequent installation of additional LogAnalyzer Core services. You must specify an even number of servers in this group. If the `kuma_core_peers` group is missing from the inventory file or is empty, the normal installation or upgrade procedure with a single LogAnalyzer Core is performed. Only one LogAnalyzer Core service can be installed on each server in the `kuma_core_peers` group.

    Using the kuma_core_peers group for different installer use cases

Installer use case	The inventory file contains a kuma_core_peers group	The inventory file does not contain a kuma_core_peers group or the group is empty
Installation using the install.sh installer	Servers from the group are prepared for running the LogAnalyzer Core service. The kuma-core-00000000-0000-0000-0000-000000000000 service is installed on a server from the kuma_core group (displayed as core-1 in the web interface). On all servers from the kuma_core_peers group, the kuma-core-UID services are sequentially installed and started (displayed as core-<FQDN> in the web interface). These services are connected to the core-1 service. All additional services also connect to mongoDB running on the core-1 host, so the mongoDB is bound to all IP addresses of the core-1 server, and port 27017/tcp is allowed on the firewall.	A standard installation is performed with one core-1 service and one metrics service on a server from the kuma_core group.
Upgrading from version 3.4.x using the install.sh installer	Servers from the kuma_core_peers group are prepared for running the LogAnalyzer Core service. LogAnalyzer Core services are not installed on these servers. You can later create additional LogAnalyzer Core services in the web interface and install these on the prepared servers.	Normal upgrade is performed, the old metrics services are replaced by the new servers running on the server from the kuma_core group.
Expanding the installation using the expand.sh installer	You can create or expand a Raft cluster in an existing installation. To do this, you need to create an expand.inventory.yml inventory file, but fill in only the kuma_core_peers group. After running the expand.sh installer with the expand.inventory.yml inventory file, you need to manually create the LogAnalyzer Core services in the web interface and install them on the servers prepared by the installer. After expanding the Raft cluster, we recommend reflecting the changes in the kuma_core_peers group in the inventory file distributed.inventory.yml in case you need to upgrade LogAnalyzer.	A normal installation expansion is performed.

- `raft_node_addr` is the address to be used for the cluster of nodes with the LogAnalyzer Core service. This address must be public and known to other nodes. After the LogAnalyzer Core is started for the first time with this address, the address cannot be changed anymore for this node. This value must be specified in the <FQDN>:<`port`> format. If this setting is not specified explicitly, <FQDN> defaults to the FQDN of the host on which the LogAnalyzer Core is deployed, and <`port`> defaults to 7209. The `raft_node_addr` setting must be specified together with the `raft_node_listen setting on each LogAnalyzer Core server with if DNS is configured in such a way that the server contains multiple FQDNs`. Optional setting.

- `raft_node_listen` is the address at which the node listens for messages from its peers in the cluster. This setting must be specified in the following format <`IPv4 or IPv6 address of the LogAnalyzer Core server`>:<`port`>. This address can be changed. The `raft_node_listen` setting must be specified together with the `raft_node_addr setting on every LogAnalyzer Core server`. Optional setting.

- `kuma_collector` contains settings of LogAnalyzer collectors. You can specify multiple hosts.

  - The `ansible_port` variable lets you specify a non-standard SSH ports during installation. Must be specified for each host in the inventory file, even if the same ports are used on all hosts. Optional setting.

- `kuma_storage` contains settings of LogAnalyzer storage nodes. You can specify multiple hosts as well as shard, replica, and keeper IDs for hosts using the following settings:

  - The `ansible_port` variable lets you specify a non-standard SSH ports during installation. Must be specified for each host in the inventory file, even if the same ports are used on all hosts. Optional setting.

  - `shard` is the shard ID.

  - `replica` is the replica ID.

  - `keeper` is the keeper ID.

The specified shard, replica, and keeper IDs are used only if you are deploying demo services as part of a fresh LogAnalyzer installation. In other cases, the shard, replica, and keeper IDs that you specified in the LogAnalyzer web interface when creating a resource set for the storage are used.

kuma_k0s block

In this section, you can specify the settings of the Kubernetes cluster that ensures high availability of LogAnalyzer. This block is specified only in an inventory file based on k0s.inventory.yml.template.

For test and demo installations in environments with limited computational resources, you must also set `low_resources: true` in the `all` block. In this case, the minimum size of the LogAnalyzer Core installation directory is reduced to 4 GB and the limitations of other computational resources are ignored.

For each host in the `kuma_k0s` block, a unique FQDN and IP address must be specified in the `ansible_host` variable, except for the host in the `kuma_lb` section. For the host in the `kuma_lb` section, only the FQDN must be specified. Hosts must be unique within a group.

For a demo installation, you may combine controller with a worker node. Such a configuration does not provide high availability of the LogAnalyzer Core and is only intended for demonstrating the functionality or for testing the software environment.
The minimal configuration that ensures high availability is 3 controllers, 2 worker nodes, and 1 nginx load balancer. In production, we recommend using dedicated worker nodes and controllers. If a cluster controller is under workload and the pod with the LogAnalyzer Core is hosted on the controller, if the controller goes down, access to the LogAnalyzer Core will be completely lost.

The `kuma_k0s` block can contain the following sections:

- `vars` contains variables that apply to all hosts specified in the `kuma` block.
- `children` contains settings of the Kubernetes cluster that provides high availability of KUMA.

The following table lists possible variables in the `vars` section and their descriptions.

List of possible variables in the `vars` section

Variable group	Description	
`kuma_lb`	FQDN of the load balancer. You can install the nginx load balancer or a third-party TCP load balancer. If you are installing the nginx load balancer, you can set `kuma_managed_lb=true` to automatically configure the nginx load balancer when installing KUMA, open the necessary network ports on the nginx load balancer host (6443, 8132, 9443, 7209, 7210, 7220, 7222, 7223, 7226), and restart to apply the changes. If you are installing a third-party TCP load balancer, you must manually configure it before installing KUMA.	
`kuma_control_plane_master`	The host that acts as the primary controller of the cluster.	Groups for specifying the primary controller. You only need to specify a host for one group.
`kuma_control_plane_master_worker`	A host that combines the role of the primary controller and worker node in the cluster. For each cluster controller that is combined with a worker node, in the inventory file, you must specify `extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-ingress=true,node.longhorn.io/create-default-disk=true"`.	

Variable group	Description	
`kuma_control_plane`	Hosts that act as controllers in the cluster.	Groups for specifying secondary controllers.
`kuma_control_plane_worker`	Hosts that combine the roles of controller and worker node in the cluster. For each cluster controller that is combined with a worker node, in the inventory file, you must specify `extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-ingress=true,node.longhorn.io/create-default-disk=true"`.	
`kuma_worker`	Worker nodes of the cluster. For each cluster controller that is combined with a worker node, in the inventory file, you must specify `extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-ingress=true,node.longhorn.io/create-default-disk=true"`.	

`ipAutodetectionMethod`	If multiple network interfaces are being used on the worker nodes of the cluster at the same time, the `ipAutodetectionMethod` variable lets you specify a name mask of the network interface to be used for communication between the worker nodes in the cluster.
	For example, if you want to use only network interfaces named ethN (where N is the number of the network interface) for communication between worker nodes of the cluster, you can specify the variable as follows:
	`kuma_k0s:`
	`vars:`
	`ip_autodetection_method: "interface=eth.*"`
	This makes the cluster use a network interface with a name that matches the `eth.*` mask.
	If the network interface name on each worker node is the same, for example eth0, you can specify the variable without a mask:
	`kuma_k0s:`
	`vars:`
	`ip_autodetection_method: "interface=eth0"`
	For more information, please refer to the Calico Open Source documentation.	

Installation on a single server

To install LogAnalyzer components on a single server, follow these steps:

1. Ensure that the hardware, software, and installation requirements of LogAnalyzer are met.

2. Prepare the single.inventory.yml inventory file.

   Use the single.yml.template inventory file template from the distribution kit to create a single.inventory.yml inventory file and describe the network structure of application components in that file. The installer uses the single.inventory.yml file to deploy LogAnalyzer.

3. Install the application.

   Install the application and log in to the web interface using the default credentials.

If necessary, you can move application components to different servers to continue with a distributed configuration.

Preparing the single.inventory.yml inventory file

LogAnalyzer components can be installed, updated, and removed in the directory containing the extracted installer by using the Ansible® tool and the user-created YML *inventory file* containing a list of the hosts of LogAnalyzer components and other settings. If you want to install all LogAnalyzer components on the same server, you must specify the same host for all components in the inventory file.

To create an inventory file for installation on a single server:

1. Copy the archive with the `kuma-ansible-installer-<version>.tar.gz` installer to the server and extract it using the following command (about 2 GB of disk space is required):

   ```
   sudo tar -xpf kuma-ansible-installer-<version name>.tar.gz
   ```

2. Go to the LogAnalyzer installer directory by executing the following command:

   ```
   cd kuma-ansible-installer
   ```

3. Copy the single.inventory.yml.template and create an inventory file named single.inventory.yml:

   ```
   cp single.inventory.yml.template single.inventory.yml
   ```

4. Edit the settings in the single.inventory.yml inventory file.

   If you want predefined services to be created during the installation, set deploy_example_services to true.

   ```
   deploy_example_services: true
   ```

   The predefined services will appear only as a result of the initial installation of LogAnalyzer. If you are upgrading the system using the same inventory file, the predefined services are not re-created.

5. Replace all `kuma.example.com` strings in the inventory file with the name of the host on which you want to install LogAnalyzer components.

The inventory file is created. Now you can use it to install LogAnalyzer on a single server.

We recommend saving a backup copy of the inventory file that you used to install the application. You can use it to add components to the system or remove LogAnalyzer.

Example inventory file for installation on a single server

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: true
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma1.example.com:
    kuma_collector:
      hosts:
        kuma1.example.com
    kuma_correlator:
      hosts:
        kuma1.example.com
    kuma_storage:
      hosts:
        kuma1.example.com:
          shard: 1
          replica: 1
          keeper: 1
```

Installing the program on a single server

You can install all LogAnalyzer components on a single server using the Ansible tool and the single.inventory.yml inventory file.

To install LogAnalyzer on a single server:

1. Download the kuma-ansible-installer-<`build number`>.tar.gz LogAnalyzer distribution kit to the server and extract it. The archive is extracted into the kuma-ansibleinstaller directory.

2. Go to the directory with the unpacked installer.

3. Depending on the type of application activation that you plan to use, do one of the following:

   - If you want to activate the application with a key file, place the key file in the <`installer directory`>/roles/kuma/files/ directory.

     The key file must be named license.key.

     ```
     sudo cp <key file>.key <installer directory>/roles/kuma/files/license.key
     ```

   - If you want to activate the application with a license code, go to the next step of the instructions.

     > Activation with a license code is available only for LogAnalyzer 3.4 and later. For earlier versions of LogAnalyzer, you must activate the application with a key file.

4. Run the following command to start the component installation with your prepared single.inventory.yml inventory file:

   ```
   sudo ./install.sh single.inventory.yml
   ```

   If privilege elevation requires specifying a password, run the following command and specify the password when prompted:

   ```
   sudo ./install.sh single.inventory.yml --ask-become-pass
   ```

5. Accept the terms of the End User License Agreement.

   If you do not accept the terms of the End User License Agreement, the program will not be installed.

   Depending on the type of application activation, the installer produces one of the following results:

   - If you want to activate the application using a file and have placed the key file in <`installer directory`>/roles/kuma/files/, running the installer with the single.inventory.yml inventory file installs LogAnalyzer Core, all services specified in the inventory file, and OOTB resources. If example_services=true is set in the inventory file, demo services are installed.

   - If you want to activate the application with a license code or provide a license file later, running the installer with the "single.inventory.yml" inventory file installs only LogAnalyzer Core.

     To install the services, specify the license code on the command line. Then run the postinstall.sh installer with the "single.inventory.yml" inventory file.

     ```
     sudo ./postinstall.sh single.inventory.yml
     ```

     This creates the specified services. You can select the resources that you want to import from the repository.

6. After the installation is complete, log in to the LogAnalyzer web interface and enter the address of the LogAnalyzer web interface in the address bar of your browser, then enter your credentials on the login page.

   The address of the LogAnalyzer web interface is `https://<FQDN of the host where LogAnalyzer is installed>:7220`.

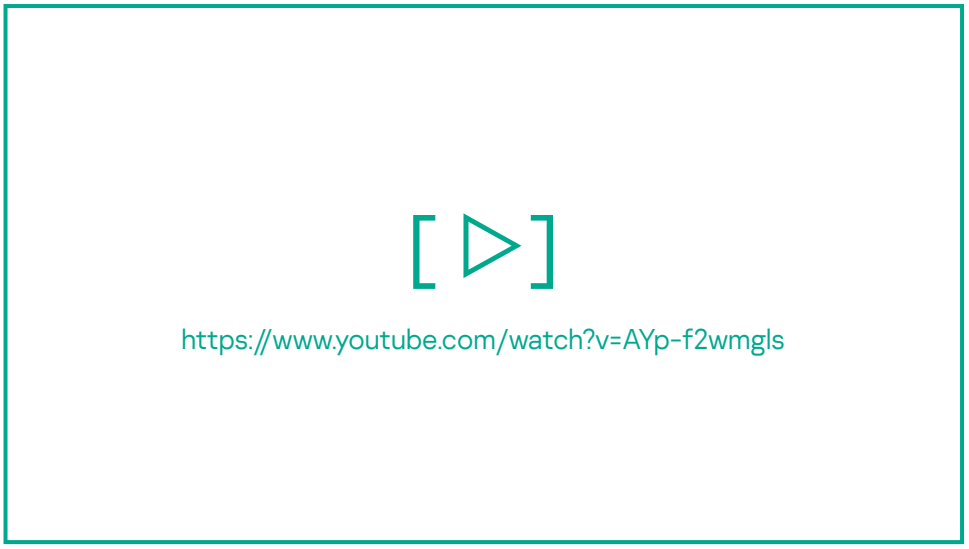
   Default login credentials:
   - login: `admin`
   - password: `mustB3Ch@ng3d!`

   After the first login, change the password of the admin account

All LogAnalyzer components are installed and you are logged in to the web interface.

We recommend backing up the inventory file that you used to install the program. You can use this inventory file to add components to the system or remove LogAnalyzer.

You can expand the installation to a distributed installation.



https://www.youtube.com/watch?v=AYp-f2wmgls

Distributed installation

The distributed installation of LogAnalyzer involves several steps:

1. Verifying that the hardware, software, and installation requirements for LogAnalyzer are satisfied.
2. Preparing the control machine.

   The control machine is used during the application installation process to extract and run the installer files.
3. Preparing the target machines.

   The application components are installed on the target machines.
4. Preparing the distributed.inventory.yml inventory file.

   Create an inventory file with a description of the network structure of application components. The installer uses this inventory file to deploy LogAnalyzer.
5. Installing the application.

   Install the application and log in to the web interface.
6. Creating services.

   Create the client part of the services in the LogAnalyzer web interface and install the server part of the services on the target machines.

   Make sure the LogAnalyzer installation is complete before you install LogAnalyzer services. We recommend installing services in the following order: storage, collectors, agents.

   When deploying several LogAnalyzer services on the same host, during installation, you must specify unique ports for each service using the `--api.port <port>` parameter.

If necessary, you use your company's certificate as the LogAnalyzer web console certificate.

https://www.youtube.com/watch?v=-cNRCxNzV2A

Preparing the test machine

To prepare the control machine for installing LogAnalyzer:

1. Ensure that hardware, software, and installation requirements of the application are met.

2. Generate an SSH key for authentication on the SSH servers of the target machines:

   ```
   sudo ssh-keygen -f /root/.ssh/id_rsa -N "" -C kuma-ansible-installer
   ```

   If SSH root access is blocked on the control machine, generate an SSH key for authentication on the SSH servers of the target machines for **a user from the sudo group**:

   > If the user does not have sudo rights, add the user to the sudo group:
   >
   > ```
   > usermod -aG sudo user
   > ```

   ```
   ssh-keygen -f /home/<name of the user from the sudo group>/.ssh/id_rsa -N "" -C kuma-ansible-installer
   ```

   As a result, the key is generated and saved in the user's home directory. You should specify the full path to the key in the inventory file in the value of the ansible_ssh_private_key_file parameter so that the key is available during installation.

3. Make sure that the control machine has network access to all the target machines by host name and copy the SSH key to each target machine:

   ```
   sudo ssh-copy-id -i /root/.ssh/id_rsa root@<host name of the target machine>
   ```

   If SSH root access is blocked on the test machine and you want to use the SSH key from the home directory of the sudo group user, make sure that the test machine has network access to all target machines by host name and copy the SSH key to each target machine using the following command:

   ```
   ssh-copy-id -i /home/<name of the user in the sudo group>/.ssh/id_rsa <name of the user in the sudo group>@<host name of the target machine>
   ```

4. Copy the archive with the `kuma-ansible-installer-<version>.tar.gz` installer to the test machine and unpack it using the following command (about 2 GB of disk space is required):

   ```
   sudo tar -xpf kuma-ansible-installer-<version>.tar.gz
   ```

The control machine is prepared for installing LogAnalyzer.

Preparing the target machine

To prepare the target machine for the installation of LogAnalyzer components:

1. Ensure that hardware, software, and installation requirements are met.
2. Specify the host name. We recommend specifying a FQDN. For example, kuma1.example.com.

   Do not change the LogAnalyzer host name after installation: this will make it impossible to verify the authenticity of certificates and will disrupt the network communication between the application components.
3. Register the target machine in your organization's DNS zone to allow host names to be resolved to IP addresses.

   If your organization does not use a DNS server, you can use the /etc/hosts file for name resolution. The content of the files can be automatically generated for each target machine when installing LogAnalyzer.
4. To get the hostname that you must specify when installing LogAnalyzer, run the following command and record the result:

   ```
   hostname -f
   ```

   The control machine must be able to access the target machine using this name.

The target machine is ready for the installation of LogAnalyzer components.

Preparing the distributed.inventory.yml inventory file

To create the distributed.inventory.yml inventory file:

1. Go to the LogAnalyzer installer directory by executing the following command:

   ```
   cd kuma-ansible-installer
   ```
2. Create an inventory file named distributed.inventory.yml by copying distributed.inventory.yml.template:

   ```
   cp distributed.inventory.yml.template distributed.inventory.yml
   ```
3. Edit the settings in the distributed.inventory.yml.

We recommend backing up the inventory file that you used to install the program. You can use it to add components to the system or remove LogAnalyzer.

Example inventory file for distributed installation

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
          ip: 0.0.0.0
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ip: 0.0.0.0
    kuma_storage:
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server4.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server5.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 1
        kuma-storage-cluster1-server6.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 2
        kuma-storage-cluster1-server7.example.com:
          ip: 0.0.0.0
```

```
          shard: 0
          replica: 0
          keeper: 3
```

Installing the program in a distributed configuration

LogAnalyzer is installed using the Ansible tool and a YML inventory file. The installation is performed from the control machine, and all of the LogAnalyzer components are installed on target machines.

To install LogAnalyzer:

1. On the control machine, go to the directory containing the extracted installer.

   ```
   cd kuma-ansible-installer
   ```

2. Depending on the type of application activation that you plan to use, do one of the following:

   - To activate the application using a key file, place the key file in the <installer directory>/roles/kuma/files/ directory.

     The key file must be named license.key.

     ```
     sudo cp <key file>.key <installer directory>/roles/kuma/files/license.key
     ```

   - To activate the application using an activation code, proceed to the next step of the instructions.

3. From the directory with the unpacked installer, start the installation of components using the prepared distributed.inventory.yml inventory file:

   ```
   sudo ./install.sh distributed.inventory.yml
   ```

   If privilege elevation on the target machine requires specifying a password, run the following command and specify the password when prompted:

   ```
   sudo ./install.sh distributed.inventory.yml --ask-become-pass
   ```

4. If you want to use a non-standard port for the SSH connection during installation, select one of the following options:

   - If the port is the same for a group of hosts, specify it in the inventory file in the 'vars' group.

     Example of a distributed.inventory.yml inventory file with the same non-standard port for a group of storage hosts

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
          ip: 0.0.0.0
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ip: 0.0.0.0
    kuma_storage:
      vars:
        ansible_port: 2222
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server4.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server5.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 1
        kuma-storage-cluster1-server6.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 2
```

```
        kuma-storage-cluster1-server7.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 3
```

- If a different non-standard port is used for the connection to each host, specify the port for each host in the inventory file, for example:

  Example of a distributed.inventory.yml inventory file with a different non-standard port for each storage host

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
          ip: 0.0.0.0
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ip: 0.0.0.0
    kuma_storage:
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ansible_port: 2222
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ansible_port: 2223
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ansible_port: 2224
          ip: 0.0.0.0
          shard: 2
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server4.example.com:
          ansible_port: 2225
          ip: 0.0.0.0
          shard: 2
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server5.example.com:
          ansible_port: 2226
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 1
        kuma-storage-cluster1-server6.example.com:
          ansible_port: 2227
```

```
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 2
        kuma-storage-cluster1-server7.example.com:
          ansible_port: 2228
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 3
```

5. Accept the terms of the End User License Agreement.

   If you do not accept the terms of the End User License Agreement, the program will not be installed.

   Depending on the type of application activation, the installer produces one of the following results:

   - If you are activating the application using a key file and have placed the key file in <installer directory>/roles/kuma/files/, running the installer with the distributed.inventory.yml inventory file installs LogAnalyzer Core, all services specified in the inventory file, and OOTB resources.

   - If you are activating the application with an activation code or plan to upload a key file later, running the installer with the "distributed.inventory.yml" inventory file installs only LogAnalyzer Core.

     To install services, enter the activation code on the command line. Then run the postinstall.sh installer with the distributed.inventory.yml inventory file.

     ```
     sudo ./postinstall.sh distributed.inventory.yml
     ```

     This creates the specified services. You can select the resources that you want to import from the repository.

6. After the installation is complete, log in to the LogAnalyzer web interface and enter the address of the LogAnalyzer web interface in the address bar of your browser, then enter your credentials on the login page.

   The address of the LogAnalyzer web interface is `https://<FQDN of the host where LogAnalyzer is installed>:7220`.

   Default login credentials:

   - user name: `admin`

   - password: `mustB3Ch@ng3d!`

   After the first login, change the password of the admin account

All LogAnalyzer components are installed and you are logged in to the web interface.

We recommend backing up the inventory file that you used to install the program. You can use this inventory file to add components to the system or remove LogAnalyzer.

Distributed installation with LogAnalyzer Core in a Kubernetes cluster

The high-availability configuration of LogAnalyzer involves deploying the LogAnalyzer Core on a Kubernetes cluster and using an external TCP traffic balancer.

To create a high-availability LogAnalyzer installation, use the installer from the kuma-ansible-installer-k8s-<build number>.tar.gz package and prepare the k0s.inventory.yml inventory file by specifying the configuration of the cluster. For a new installation in a high availability configuration, OOTB resources are always imported. You can also perform an installation with deployment of demo services. To do this, set "deploy_example_services: true" in the inventory file.

The LogAnalyzer Core can be placed in a Kubernetes cluster in the following ways:

- Install LogAnalyzer in a Kubernetes cluster.
- Migrate the Core of an existing LogAnalyzer installation to the Kubernetes cluster.

Minimum configuration

Kubernetes has 2 node roles:

- Controllers (control-plane). Nodes with this role manage the cluster, store metadata, and balance the workload.
- Workers (worker). Nodes with this role bear the workload by hosting KUMA processes.

To deploy LogAnalyzer in a high availability configuration, you need:

- 3 dedicated controllers
- 2 worker nodes
- 1 TCP balancer

You must not use the balancer as the control machine for running the LogAnalyzer installer.

To ensure the adequate performance of the LogAnalyzer Core in Kubernetes, you must allocate 3 dedicated nodes that have only the controller role. This will provide high availability for the Kubernetes cluster itself and will ensure that the workload (LogAnalyzer processes and other processes) cannot affect the tasks involved in managing the Kubernetes cluster. If you are using virtualization tools, make sure that the nodes are hosted on different physical servers and that these physical servers are not being used as worker nodes.

For a demo installation of LogAnalyzer, you may combine the controller and worker roles. However, if you are expanding an installation to a distributed installation, you must reinstall the entire Kubernetes cluster and allocate 3 dedicated nodes with the controller role and at least 2 nodes with the worker role. LogAnalyzer cannot be upgraded to later versions if any of the nodes combine the controller and worker roles.

Additional requirements for deploying KUMA Core in Kubernetes

If you plan to protect LogAnalyzer's network infrastructure using Kaspersky Endpoint Security for Linux, first install LogAnalyzer in the Kubernetes cluster and only then deploy Kaspersky Endpoint Security for Linux. When upgrading or uninstalling LogAnalyzer, you must first stop Kaspersky Endpoint Security for Linux using the following command:

```
systemctl stop kesl
```

When you install LogAnalyzer in a high-availability configuration, the following requirements must be met:

- General application installation requirements.
- The hosts that you plan to use for Kubernetes cluster nodes must not use IP addresses from the following Kubernetes ranges:
  - serviceCIDR: 10.96.0.0/12
  - podCIDR: 10.244.0.0/16

  Traffic to proxy servers must excluded for the IP addresses from these ranges.
- Each host must have a unique ID (/etc/machine-id).
- The firewalld or uwf firewall management tool must be installed and enabled on the hosts for adding rules to iptables.

- The nginx load balancer must be installed and configured (for details, please refer to the nginx load balancer documentation). You can install the nginx load balancer using one of the following commands:

  - `sudo yum install nginx` (for Oracle Linux)
  - `sudo apt install nginx-full` (for Astra Linux)
  - `sudo apt install nginx libnginx-mod-stream` (for Ubuntu)
  - `sudo yum install nginx nginx-all-modules` (for RED OS)

  If you want the nginx load balancer to be configured automatically during the LogAnalyzer installation, install the nginx load balancer and allow SSH access to it in the same way as for the Kubernetes cluster hosts.

  Example of an automatically created nginx configuration

The installer creates the /etc/nginx/kuma_nginx_lb.conf configuration file. An example of the file contents is shown below. The `upstream` sections are generated dynamically and contain the IP addresses of the Kubernetes cluster controllers (in the example, 10.0.0.2-4 in the `upstream kubeAPI_backend`, `upstream konnectivity_backend`, `controllerJoinAPI_backend` sections) and the IP addresses of the worker nodes (in the example 10.0.1.2-3), for which the inventory file contains the `"kaspersky.com/kuma-ingress=true"` value for the `extra_args` variable.

The "`include /etc/nginx/kuma_nginx_lb.conf;`" line must be added to the end of the /etc/nginx/nginx.conf file to apply the generated configuration file. If you have a large number of active services and users, you may need to increase the limit of open files in the nginx.conf settings.

Configuration file example:

```
# Ansible managed
#
# LB KUMA cluster
#
stream {
    server {
        listen          6443;
        proxy_pass      kubeAPI_backend;
    }
    server {
        listen          8132;
        proxy_pass      konnectivity_backend;
    }
    server {
        listen          9443;
        proxy_pass      controllerJoinAPI_backend;
    }
    server {
        listen          7209;
        proxy_pass      kuma-core-hierarchy_backend;
        proxy_timeout   86400s;
    }
    server {
        listen          7210;
        proxy_pass      kuma-core-services_backend;
        proxy_timeout   86400s;
    }
    server {
        listen          7220;
        proxy_pass      kuma-core-ui_backend;
        proxy_timeout   86400s;
    }
    server {
        listen          7222;
        proxy_pass      kuma-core-cybertrace_backend;
        proxy_timeout   86400s;
    }
    server {
        listen          7223;
        proxy_pass      kuma-core-rest_backend;
        proxy_timeout   86400s;
    }
    upstream kubeAPI_backend {
        server 10.0.0.2:6443;
```

```
        server 10.0.0.3:6443;
        server 10.0.0.4:6443;
    }
    upstream konnectivity_backend {
        server 10.0.0.2:8132;
        server 10.0.0.3:8132;
        server 10.0.0.4:8132;
    }
    upstream controllerJoinAPI_backend {
        server 10.0.0.2:9443;
        server 10.0.0.3:9443;
        server 10.0.0.4:9443;
    }
    upstream kuma-core-hierarchy_backend {
        server 10.0.1.2:7209;
        server 10.0.1.3:7209;
    }
    upstream kuma-core-services_backend {
        server 10.0.1.2:7210;
        server 10.0.1.3:7210;
    }
    upstream kuma-core-ui_backend {
        server 10.0.1.2:7220;
        server 10.0.1.3:7220;
    }
    upstream kuma-core-cybertrace_backend {
        server 10.0.1.2:7222;
        server 10.0.1.3:7222;
    }
    upstream kuma-core-rest_backend {
        server 10.0.1.2:7223;
        server 10.0.1.3:7223;
    }
 worker_rlimit_nofile 1000000;
 events {
worker_connections 20000;
}
 # worker_rlimit_nofile is the limit on the number of open files (RLIMIT_NOFILE)
for workers.This is used to raise the limit without restarting the main process.
# worker_connections  is the maximum number of connections that a worker can open
simultaneously.
```

- An access key from the device on which LogAnalyzer is installed must be added to the nginx load balancer server.
- On the nginx load balancer server, the SELinux module must be disabled in the operating system.
- The tar, systemctl packages are installed on the hosts.

During LogAnalyzer installation, the hosts are automatically checked to see if they meet the following hardware requirements:

- Number of CPU cores (threads) – 12 or more.
- RAM: 22,528 MB or more

- Free disk space in the /opt partition: 1000 GB or more.
- For an installation from scratch, the /var/lib partition must have at least 32 GB of free space. If the cluster already has been installed on this node, the size of the required free space is reduced by the size of the /var/lib/k0s directory.

If these conditions are not satisfied, the installation is stopped. For a demo installation, you can disable the check of these conditions by setting `low_resources: true` in the inventory file.

Additional requirements when installing on Astra Linux or Ubuntu operating systems.

The following packages must be installed on the machines intended for deploying a Kubernetes cluster:

- open-iscsi
- wireguard
- wireguard-tools

To install the packages, run the following command:

```
sudo apt install open-iscsi wireguard wireguard-tools
```

Additional requirements when installing on the Oracle Linux, RED OS, or Red Hat Enterprise Linux operating systems

The following packages must be installed on the machines intended for deploying the Kubernetes cluster:

- iscsi-initiator-utils
- wireguard-tools

Before installing the packages on Oracle Linux, you must add the EPEL repository as a source of packages using one of the following commands:

- `sudo yum install oracle-epel-release-el8` (for Oracle Linux 8)
- `sudo yum install oracle-epel-release-el9` (for Oracle Linux 9)

To install the packages, run the following command:

```
sudo yum install iscsi-initiator-utils wireguard-tools
```

Installing LogAnalyzer in a Kubernetes cluster

The distributed installation of LogAnalyzer involves several steps:

1. Verifying that the hardware, software, and installation requirements for LogAnalyzer are satisfied.
2. Preparing the control machine.

   The control machine is used during the application installation process to extract and run the installer files.

3. Preparing the target machines.

   The program components are installed on the target machines.

4. Preparing the k0s.inventory.yml inventory file.

   Create an inventory file with a description of the network structure of program components. The installer uses this inventory file to deploy LogAnalyzer.

5. Installing the program.

   Install the application and log in to the web interface.

6. Creating services.

   Create the client part of the services in the LogAnalyzer web interface and install the server part of the services on the target machines.

   Make sure the LogAnalyzer installation is complete before you install LogAnalyzer services. We recommend installing services in the following order: storage, collectors, agents.

   > When deploying several LogAnalyzer services on the same host, during installation, you must specify unique ports for each service using the `--api.port <port>` parameter.

If necessary, you use your company's certificate as the LogAnalyzer web console certificate.

Preparing the test machine

To prepare the control machine for installing LogAnalyzer:

1. Ensure that hardware, software, and installation requirements of the application are met.

2. Generate an SSH key for authentication on the SSH servers of the target machines:

   ```
   sudo ssh-keygen -f /root/.ssh/id_rsa -N "" -C kuma-ansible-installer
   ```

   If SSH root access is blocked on the control machine, generate an SSH key for authentication on the SSH servers of the target machines for **a user from the sudo group**:

   > If the user does not have sudo rights, add the user to the sudo group:
   >
   > ```
   > usermod -aG sudo user
   > ```

   ```
   ssh-keygen -f /home/<name of the user from the sudo group>/.ssh/id_rsa -N "" -C kuma-ansible-installer
   ```

   As a result, the key is generated and saved in the user's home directory. You should specify the full path to the key in the inventory file in the value of the ansible_ssh_private_key_file parameter so that the key is available during installation.

3. Make sure that the control machine has network access to all the target machines by host name and copy the SSH key to each target machine:

   ```
   sudo ssh-copy-id -i /root/.ssh/id_rsa root@<host name of the target machine>
   ```

   If SSH root access is blocked on the test machine and you want to use the SSH key from the home directory of the sudo group user, make sure that the test machine has network access to all target machines by host name and copy the SSH key to each target machine using the following command:

   ```
   ssh-copy-id -i /home/<name of the user in the sudo group>/.ssh/id_rsa <name of the user in the sudo group>@<host name of the target machine>
   ```

4. Copy the installer archive from the `kuma-ansible-installer-k8s-<version number>.tar.gz` package to the test machine and unpack it using the following command:

   ```
   sudo tar -xpf kuma-ansible-installer-k8s-<version number>.tar.gz
   ```

The control machine is prepared for installing LogAnalyzer.

Preparing the target machine

To prepare the target machine for the installation of LogAnalyzer components:

1. Ensure that hardware, software, and installation requirements are met.

2. Specify the host name. We recommend specifying a FQDN. For example, kuma1.example.com.

   Do not change the LogAnalyzer host name after installation: this will make it impossible to verify the authenticity of certificates and will disrupt the network communication between the application components.

3. Register the target machine in your organization's DNS zone to allow host names to be translated to IP addresses.

   The option of using the /etc/hosts file is not available when the Core is deployed in Kubernetes.

4. To get the hostname that you must specify when installing LogAnalyzer, run the following command and record the result:

   ```
   hostname -f
   ```

   The control machine must be able to access the target machine using this name.

The target machine is ready for the installation of LogAnalyzer components.

Preparing the k0s.inventory.yml inventory file

To create the k0s.inventory.yml inventory file:

1. Go to the LogAnalyzer installer directory by executing the following command:

   ```
   cd kuma-ansible-installer-ha
   ```

2. Copy the k0s.inventory.yml.template file to create the expand.inventory.yml inventory file:

   ```
   cp k0s.inventory.yml.template k0s.inventory.yml
   ```

3. Edit the inventory file settings in k0s.inventory.yml.

   Example inventory file for an installation with the LogAnalyzer Core in a Kubernetes cluster with 3 controllers, 2 worker nodes, and 1 balancer

```
all:
  vars:
    ansible_connection: ssh
    ansible_user: root
    deploy_to_k8s: true
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
    metrics_disk_request: 150Gi
    core_disk_request: 100Gi
kuma:
  children:
    kuma_collector:
      hosts:
        kuma-collector.example.com:
    kuma_correlator:
      hosts:
        kuma-correlator.example.com:
    kuma_storage:
      hosts:
        kuma-storage-cluster1.server1.example.com
        kuma-storage-cluster1.server2.example.com
        kuma-storage-cluster1.server3.example.com
        kuma-storage-cluster1.server4.example.com
        kuma-storage-cluster1.server5.example.com
        kuma-storage-cluster1.server6.example.com
        kuma-storage-cluster1.server7.example.com
kuma_k0s:
  children:
    kuma_lb:
      hosts:
        kuma-lb.example.com:
          kuma_managed_lb: true
    kuma_control_plane_master:
      hosts:
        kuma_cpm.example.com:
          ansible_host: 10.0.1.10
    kuma_control_plane_master_worker:
    kuma_control_plane:
      hosts:
        kuma_cp2.example.com:
          ansible_host: 10.0.1.11
        kuma_cp3.example.com:
          ansible_host: 10.0.1.12
    kuma_control_plane_worker:
    kuma_worker:
      hosts:
        kuma-w1.example.com:
          ansible_host: 10.0.2.11
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
        kuma-w2.example.com:
          ansible_host: 10.0.2.12
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
```

For such a configuration, specify the parameters as follows: need_transfer: false, deploy_example_services: false; in the kuma_storage section, list the servers for the storage cluster. After the installation is complete, you can use the LogAnalyzer web interface to assign the shard, replica and keeper roles to the servers specified in the inventory.

Example inventory file for migrating the Core from a distributed installation to a Kubernetes cluster to ensure high availability

```
all:
  vars:
    ansible_connection: ssh
    ansible_user: root
    deploy_to_k8s: true
    need_transfer: true
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
    metrics_disk_request: 150Gi
    core_disk_request: 100Gi
kuma:
  children:
    kuma_collector:
      hosts:
        kuma-collector.example.com:
    kuma_correlator:
      hosts:
        kuma-correlator.example.com:
    kuma_storage:
      hosts:
        kuma-storage-cluster1.server1.example.com
        kuma-storage-cluster1.server2.example.com
        kuma-storage-cluster1.server3.example.com
        kuma-storage-cluster1.server4.example.com
        kuma-storage-cluster1.server5.example.com
        kuma-storage-cluster1.server6.example.com
        kuma-storage-cluster1.server7.example.com
kuma_k0s:
  children:
    kuma_lb:
      hosts:
        kuma-lb.example.com:
          kuma_managed_lb: true
    kuma_control_plane_master:
      hosts:
        kuma_cpm.example.com:
          ansible_host: 10.0.1.10
    kuma_control_plane_master_worker:
    kuma_control_plane:
      hosts:
        kuma_cp2.example.com:
          ansible_host: 10.0.1.11
        kuma_cp3.example.com:
          ansible_host: 10.0.1.12
    kuma_control_plane_worker:
    kuma_worker:
      hosts:
        kuma-w1.example.com:
          ansible_host: 10.0.2.11
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
        kuma-w2.example.com:
          ansible_host: 10.0.2.12
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
```

The kuma_core, kuma_collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used in the distributed.inventory.yml file when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to subsequent versions, or when a new installation was performed. In the k0s.inventory.yml inventory file, set deploy_to_k8s: true, need_transfer: true, deploy_example_services: false.

We recommend backing up the inventory file that you used to install the program. You can use it to add components to the system or remove KUMA.

Installing the program in a high availability configuration

LogAnalyzer is installed using the Ansible tool and the k0s.inventory.yml inventory file. The installation is performed from the control machine, and all of the LogAnalyzer components are installed on target machines.

To install LogAnalyzer:

1. On the control machine, go to the directory containing the extracted installer.

   ```
   cd kuma-ansible-installer-ha
   ```

2. Depending on the type of application activation that you plan to use, do one of the following:

   - If you want to activate the application with a key file, place the key file in the <installer directory>/roles/kuma/files/ directory.

     The key file must be named license.key.

     ```
     sudo cp <key file>.key <installer directory>/roles/kuma/files/license.key
     ```

   - If you want to activate the application with a license code, go to the next step of the instructions.

3. From the folder with the unpacked installer, start the installation of components using the prepared distributed.inventory.yml inventory file:

   ```
   sudo ./install.sh k0s.inventory.yml
   ```

   If privilege elevation on the target machine requires specifying a password, run the following command and specify the password when prompted:

   ```
   sudo ./install.sh k0s.inventory.yml --ask-become-pass
   ```

4. If you want to use a non-standard port for the SSH connection during installation, select one of the following options:

   - If the port is the same for a group of hosts, specify the port in the inventory file in the 'vars' group.

     Example of a k0s.inventory.yml inventory file with the same non-standard port for all hosts in the kuma_control_plane section

```
all:
  vars:
    ansible_connection: ssh
    ansible_user: root
    deploy_to_k8s: true
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
    metrics_disk_request: 150Gi
    core_disk_request: 100Gi
kuma:
  children:
    kuma_collector:
      hosts:
        kuma-collector.example.com:
    kuma_correlator:
      hosts:
        kuma-correlator.example.com:
    kuma_storage:
      hosts:
        kuma-storage-cluster1.server1.example.com
        kuma-storage-cluster1.server2.example.com
        kuma-storage-cluster1.server3.example.com
        kuma-storage-cluster1.server4.example.com
        kuma-storage-cluster1.server5.example.com
        kuma-storage-cluster1.server6.example.com
        kuma-storage-cluster1.server7.example.com
kuma_k0s:
  children:
    kuma_lb:
      hosts:
        kuma-lb.example.com:
          kuma_managed_lb: true
    kuma_control_plane_master:
      hosts:
        kuma_cpm.example.com:
          ansible_host: 10.0.1.10
    kuma_control_plane_master_worker:
    kuma_control_plane:
      vars:
        ansible_port: 2222
      hosts:
        kuma_cp2.example.com:
          ansible_host: 10.0.1.11
        kuma_cp3.example.com:
          ansible_host: 10.0.1.12
    kuma_control_plane_worker:
    kuma_worker:
      hosts:
        kuma-w1.example.com:
          ansible_host: 10.0.2.11
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
        kuma-w2.example.com:
          ansible_host: 10.0.2.12
```

```
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
```

- If a different non-standard port is used for the connection to each host, specify the port for each host in the inventory file.

  Example of a k0s.inventory.yml inventory file with a different non-standard port for each hosts in the kuma_control_plane group

```
all:
  vars:
    ansible_connection: ssh
    ansible_user: root
    deploy_to_k8s: true
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
    metrics_disk_request: 150Gi
    core_disk_request: 100Gi
kuma:
  children:
    kuma_collector:
      hosts:
        kuma-collector.example.com:
    kuma_correlator:
      hosts:
        kuma-correlator.example.com:
    kuma_storage:
      hosts:
        kuma-storage-cluster1.server1.example.com
        kuma-storage-cluster1.server2.example.com
        kuma-storage-cluster1.server3.example.com
        kuma-storage-cluster1.server4.example.com
        kuma-storage-cluster1.server5.example.com
        kuma-storage-cluster1.server6.example.com
        kuma-storage-cluster1.server7.example.com
kuma_k0s:
  children:
    kuma_lb:
      hosts:
        kuma-lb.example.com:
          kuma_managed_lb: true
    kuma_control_plane_master:
      hosts:
        kuma_cpm.example.com:
          ansible_host: 10.0.1.10
    kuma_control_plane_master_worker:
    kuma_control_plane:
      hosts:
        kuma_cp2.example.com:
          ansible_port: 2222
          ansible_host: 10.0.1.11
        kuma_cp3.example.com:
          ansible_port: 2223
          ansible_host: 10.0.1.12
    kuma_control_plane_worker:
    kuma_worker:
      hosts:
        kuma-w1.example.com:
          ansible_host: 10.0.2.11
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
        kuma-w2.example.com:
          ansible_host: 10.0.2.12
```

```
          extra_args: "--labels=kaspersky.com/kuma-core=true,kaspersky.com/kuma-
ingress=true,node.longhorn.io/create-default-disk=true"
```

5. Accept the terms of the End User License Agreement.

   If you do not accept the terms of the End User License Agreement, the program will not be installed.

   Depending on the type of application activation, the installer produces one of the following results:

   - If you want to activate the application using a key file and have placed the key file in <installer directory>/roles/kuma/files/, running the installer with the k0s.inventory.yml inventory file installs LogAnalyzer Core, all services specified in the inventory file, and OOTB resources.

   - If you want to activate the application with a license code or provide a key file later, running the installer with the "k0s.inventory.yml" inventory file installs only LogAnalyzer Core.

     To install the services, specify the license code on the command line. Then run the postinstall.sh installer with the "k0s.inventory.yml" inventory file.

     ```
     sudo ./postinstall.sh k0s.inventory.yml
     ```

     This creates the specified services. You can select the resources that you want to import from the repository.

6. After the installation is complete, log in to the LogAnalyzer web interface and enter the address of the LogAnalyzer web interface in the address bar of your browser, then enter your credentials on the login page.

   The address of the LogAnalyzer web interface is `https://<FQDN of the nginx load balancer>:7220`.

   Default login credentials:

   - user name: `admin`

   - password: `mustB3Ch@ng3d!`

   After the first login, change the password of the admin account

All LogAnalyzer components are installed and you are logged in to the web interface.

We recommend backing up the inventory file that you used to install the program. You can use this inventory file to add components to the system or remove LogAnalyzer.

Migrating the LogAnalyzer Core to a new Kubernetes cluster

To migrate the LogAnalyzer Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the LogAnalyzer Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed LogAnalyzer Core on all hosts on which you want to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

Resolving the LogAnalyzer Core migration error

Migration of the LogAnalyzer Core from a host to a new Kubernetes cluster may be stopped due to a timeout at the `Deploy Core transfer job` step. In this case, the following error message is recorded in the log of core-transfer migration tasks:

```
cp: can't stat '/mnt/kuma-source/core/.lic': No such file or directory
```

To prevent this error, before you start migrating the LogAnalyzer Core:

1. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.
2. In the core-transfer-job.yaml.j2 file, find the following lines:

   ```
   cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

3. Edit these lines as follows, making sure you keep the indentation (number of space characters):

   ```
   cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

4. Save the changes to the file.

   You can then restart the distributed installation using the prepared k0s.inventory.yml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If you started migrating the LogAnalyzer Core from a host to a new Kubernetes cluster and the migration failed with an error, follow the steps below to fix the error.

To fix the error after attempting to migrate the LogAnalyzer Core from a host to a new Kubernetes cluster:

1. On any controller of the cluster, delete the Ingress object by running the following command:

   ```
   sudo k0s kubectl delete daemonset/ingress -n ingress
   ```

2. Check if a migration job exists in the cluster:

   ```
   sudo k0s kubectl get jobs -n kuma
   ```

3. If a migration job exists, delete it:

   ```
   sudo k0s kubectl delete job core-transfer -n kuma
   ```

4. Go to the console of a host from the kuma_core group.
5. Start the LogAnalyzer Core services by running the following commands:

   ```
   sudo systemctl start kuma-mongodb
   sudo systemctl start kuma-core-00000000-0000-0000-0000-000000000000
   ```

6. Make sure that the kuma-core-00000000-0000-0000-0000-000000000000 service has been successfully started:

   ```
   sudo systemctl status kuma-core-00000000-0000-0000-0000-000000000000
   ```

7. Make sure that the kuma_core group has access to the KUMA interface by host FQDN.

   Other hosts do not need to be running.

8. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.

9. In the core-transfer-job.yaml.j2 file, find the following lines:

```
cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
```

10. Edit these lines as follows, making sure you keep the indentation (number of space characters):

```
cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/ &&
```

11. Save the changes to the file.

You can then restart the distributed installation using the prepared k0s.inventory.yaml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If the component is not detected on the worker nodes, a clean installation of the LogAnalyzer Core is performed in the cluster without migrating resources to it. Existing components must be manually recreated with the new Core in the LogAnalyzer web interface.

For collectors and storages from the inventory file, certificates for communication with the Core inside the cluster will be reissued. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed is entered into the ConfigMap.

LogAnalyzer Core availability under various scenarios

LogAnalyzer Core availability in various scenarios:

- **The worker node on which the LogAnalyzer Core service is deployed fails or loses network connectivity.**

  Access to the LogAnalyzer web interface is lost. After 6 minutes, Kubernetes initiates migration of the Core bucket to an operational node of the cluster. After the deployment, which takes less than one minute, is completed, the LogAnalyzer web interface becomes available again at URLs based on the FQDN of the load balancer. To determine on which of the hosts the Core is running, run the following command in the terminal of one of the controllers:

  ```
  k0s kubectl get pod -n kuma -o wide
  ```

  When the malfunctioning worker node or access to it is restored, the Core bucket is not migrated from its current worker node. A restored node can participate in replication of a disk volume of the Core service.

- **A worker node that contains a replica of the LogAnalyzer Core disk, and which is not hosting the Core service at the moment, fails or loses network connectivity.**

  The LogAnalyzer web interface remains available at URLs based on the FQDN of the load balancer. The network storage creates a replica of the running Core disk volume on other running nodes. There is also no disruption of access to LogAnalyzer at URLs based on the FQDNs of currently operational nodes.

- **Loss of availability of one or more cluster controllers when quorum is maintained.**

  Worker nodes work normally. Access to LogAnalyzer is not disrupted. A failure of cluster controllers extensive enough to break quorum leads to the loss of control over the cluster.

  Correspondence of the number of machines in use to ensure high availability

Number of controllers when installing a cluster	Minimum number of controllers required for the operation of the cluster (quorum)	Admissible number of failed controllers
1	1	0
2	2	0
3	2	1
4	3	1
5	3	2
6	4	2
7	4	3
8	5	3
9	5	4

- **Simultaneous failure of all Kubernetes cluster controllers.**

  The cluster cannot be managed and therefore will have impaired performance.

- **Simultaneous loss of availability of all worker nodes of a cluster with replicas of the Core volume and the Core pod.**

  Access to the LogAnalyzer web interface is lost. If all replicas are lost, information will be lost.

Managing Kubernetes and access to LogAnalyzer

When installing LogAnalyzer in a high-availability configuration, the file named artifacts/k0s-kubeconfig.yml is created in the installer directory. This file contains the details required for connecting to the created Kubernetes cluster. The same file is created on the main controller in the home directory of the user set as ansible_user in the inventory file.

> To ensure that the Kubernetes cluster can be monitored and managed, the k0s-kubeconfig.yml file must be saved in a location available for the cluster administrators. Access to the file must be restricted.

Managing a Kubernetes cluster

To monitor and manage the cluster, you can use the k0s application that is installed on all cluster nodes during LogAnalyzer deployment. For example, you can use the following command to view the load on worker nodes:

```
k0s kubectl top nodes
```

Access to the LogAnalyzer Core

The URL of the LogAnalyzer Core is `https://<worker node FQDN>:<worker node port>`. Available ports: 7209, 7210, 7220, 7222, 7223. Port 7220 is the default port for connecting to the LogAnalyzer Core web interface. Any worker node whose `extra_args` parameter contains the value `kaspersky.com/kuma-ingress=true` can be used as an access point.

It is not possible to log in to the LogAnalyzer web interface on multiple worker nodes simultaneously using the same credentials. Only the most recently established connection remains active.

If you are using an external load balancer in the configuration of the high availability Kubernetes cluster, you must use the FQDN of the load balancer for access to LogAnalyzer Core ports.

Time zone in a Kubernetes cluster

The time zone within the Kubernetes cluster is always UTC+0, so the following time difference must be taken into account when dealing with data created by a high-availability LogAnalyzer Core:

- In audit events, the time zone in the `DeviceTimeZone` field is UTC+0.
- In generated reports, the difference between the report generation time and the browser's time will be displayed.
- In the dashboard, the user will find the difference between the time in the widget (the time of the user's browser is displayed) and the time in the exported widget data in the CSV file (the time of the Kubernetes cluster is displayed).

Distributed installation with LogAnalyzer Core in a Raft cluster

A distributed installation of LogAnalyzer with multiple LogAnalyzer Core services in a cluster helps provide high availability. One of the services becomes the leader; the leader is elected based on the Raft algorithm. Write operations are performed by the leader and broadcast to all servers in the cluster. Read operations are performed by each service from its local SQLite database. Thus, in addition to high availability, creating multiple LogAnalyzer Core services also takes care of horizontal scaling. The number of deployed LogAnalyzer Core services must be odd.

If necessary, you can:

- Perform a distributed installation with LogAnalyzer Core in a Raft cluster using the install.sh installer and the distributed.inventory.yml inventory file.

  In the distributed.inventory.yml inventory file, you must manually add the kuma_core_peers group and specify an even number of servers on which you want to install the additional LogAnalyzer Core services. Also, in the kuma_core group, you can specify the raft_node_addr parameter in the `<FQDN of a server from the kuma_core group on which the KUMA Core is installed>:<port>` format. This FQDN will be used to listen for signals from other nodes.

  After starting the install.sh installer with the prepared distributed.inventory.yml inventory file, none of the additional LogAnalyzer Core services are automatically installed on the servers. The installer only prepares the servers specified in the inventory file for the installation of services. After the installer finishes, only one LogAnalyzer Core service is automatically installed on the server specified in the kuma_core group. After the installer finishes, you need to manually create the LogAnalyzer Core services in the web interface and install the LogAnalyzer Core services with unique IDs one by one on each server specified in the kuma_core_peers group.

- Expand the cluster and add the necessary number of LogAnalyzer Core services using the expand.sh installer and the expand.inventory.yml inventory file.

  In the expand.inventory.yml inventory file, you must manually add the kuma_core_peers group and specify an even number of servers on which you want to install the additional LogAnalyzer Core services.

  After starting the expand.sh installer with the prepared expand.inventory.yml inventory file, none of the additional LogAnalyzer Core services are automatically installed on the servers. The installer only prepares the servers specified in the inventory file for the installation of services. After the installer finishes, only one LogAnalyzer Core service is automatically installed on the server specified in the kuma_core group. After the installer finishes, you need to manually create the LogAnalyzer Core services in the web interface and install the LogAnalyzer Core services with unique IDs one by one on each server specified in the kuma_core_peers group.

- Upgrade a distributed installation of LogAnalyzer 3.4 to a distributed installation with LogAnalyzer Core in a Raft cluster using the install.sh installer and the distributed.inventory.yml inventory file.

  In the distributed.inventory.yml inventory file, you must manually add the kuma_core_peers group and specify an even number of servers on which you want to install the additional LogAnalyzer Core services. Also, in the kuma_core group, you can specify the raft_node_addr parameter in the `<FQDN of a server from the kuma_core group on which the KUMA Core is installed>:<port>` format. This FQDN will be used to listen for signals from other nodes.

  After starting the installer with the prepared inventory file, none of the additional LogAnalyzer Core services are automatically installed on the servers. The installer only prepares the servers specified in the inventory file for the installation of services. After the installer finishes, only one LogAnalyzer Core service is automatically installed on the server specified in the kuma_core group. After the installer finishes, you need to manually create the LogAnalyzer Core services in the web interface and install the LogAnalyzer Core services with unique IDs one by one on each server specified in the kuma_core_peers group.

After installation, all LogAnalyzer Core services are automatically added to the Raft cluster.

LogAnalyzer Core logs are stored in the /opt/kaspersky/kuma/core/`<LogAnalyzer Core service ID>`/log/core directory.

Preparing the test machine

To prepare the control machine for installing LogAnalyzer:

1. Ensure that hardware, software, and installation requirements of the application are met.

2. Generate an SSH key for authentication on the SSH servers of the target machines:

   `sudo ssh-keygen -f /root/.ssh/id_rsa -N "" -C kuma-ansible-installer`

   If SSH root access is blocked on the control machine, generate an SSH key for authentication on the SSH servers of the target machines for **a user from the sudo group**:

   > If the user does not have sudo rights, add the user to the sudo group:
   >
   > `usermod -aG sudo user`

   `ssh-keygen -f /home/<name of the user from the sudo group>/.ssh/id_rsa -N "" -C kuma-ansible-installer`

   As a result, the key is generated and saved in the user's home directory. You should specify the full path to the key in the inventory file in the value of the ansible_ssh_private_key_file parameter so that the key is available during installation.

3. Make sure that the control machine has network access to all the target machines by host name and copy the SSH key to each target machine:

   `sudo ssh-copy-id -i /root/.ssh/id_rsa root@<host name of the target machine>`

   If SSH root access is blocked on the test machine and you want to use the SSH key from the home directory of the sudo group user, make sure that the test machine has network access to all target machines by host name and copy the SSH key to each target machine using the following command:

   `ssh-copy-id -i /home/<name of the user in the sudo group>/.ssh/id_rsa <name of the user in the sudo group>@<host name of the target machine>`

4. Copy the `kuma-ansible-installer-<version number> .tar.gz` installer archive to the test machine and unpack it using the following command:

   `sudo tar -xpf kuma-ansible-installer-<version name>.tar.gz`

The control machine is prepared for installing LogAnalyzer.

Preparing the target machine

To prepare the target machine for the installation of LogAnalyzer components:

1. Ensure that hardware, software, and installation requirements are met.

2. Specify the host name. We recommend specifying a FQDN. For example, kuma1.example.com.

   Do not change the LogAnalyzer host name after installation: this will make it impossible to verify the authenticity of certificates and will disrupt the network communication between the application components.

3. Register the target machine in your organization's DNS zone to allow host names to be resolved to IP addresses.

   If your organization does not use a DNS server, you can use the /etc/hosts file for name resolution. The content of the files can be automatically generated for each target machine when installing LogAnalyzer.

4. To get the hostname that you must specify when installing LogAnalyzer, run the following command and record the result:

   ```
   hostname -f
   ```

   The control machine must be able to access the target machine using this name.

The target machine is ready for the installation of LogAnalyzer components.

Preparing the distributed.inventory.yml file for installing LogAnalyzer with multiple LogAnalyzer Core services in a Raft cluster

LogAnalyzer components can be installed, updated, and removed in the directory containing the extracted installer by using the Ansible® tool and the user-created YML *inventory file* containing a list of the hosts of LogAnalyzer components and other settings.

To create a distributed.inventory.yml file for installing LogAnalyzer with multiple LogAnalyzer Core services on a Raft cluster:

1. Go to the LogAnalyzer installer directory by executing the following command:

   ```
   cd kuma-ansible-installer
   ```

2. Create an inventory file named distributed.inventory.yml by copying distributed.inventory.yml.template:

   ```
   cp distributed.inventory.yml.template distributed.inventory.yml
   ```

3. Edit the settings in the distributed.inventory.yml.

   Manually add the kuma-core-peers group to the inventory file and specify the servers for installing additional LogAnalyzer Core services. Together, the kuma-core and kuma-core-peers groups must list an odd number of servers.

   The kuma-core-00000000-0000-0000-0000-000000000000 service is installed on a server from the kuma_core group (displayed as core-1 in the web interface). On all servers from the kuma_core_peers group, the kuma-core-UID services are sequentially installed and started (displayed as core-<FQDN> in the web interface). These services are connected to the core-1 service. Only one LogAnalyzer Core service can be installed on each server.

We recommend saving a backup copy of the inventory file that you used to install the application. You can use it to add components to the system or remove LogAnalyzer.

Example distributed.inventory.yml file for installing LogAnalyzer with multiple LogAnalyzer Core services in a Raft cluster

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
          ip: 0.0.0.0
          raft_node_addr: ""
    kuma_core_peers:
      hosts:
        kuma-core-2.example.com:
          ip: 0.0.0.0
        kuma-core-3.example.com:
          ip: 0.0.0.0
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ip: 0.0.0.0
    kuma_storage:
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server4.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server5.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 1
```

```
        kuma-storage-cluster1-server6.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 2
        kuma-storage-cluster1-server7.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 3
```

Installing LogAnalyzer in a Raft cluster

LogAnalyzer is installed using the Ansible tool and a YML inventory file. The installation is performed from the control machine, and all of the LogAnalyzer components are installed on target machines.

To install LogAnalyzer:

1. On the control machine, go to the directory containing the extracted installer.

   ```
   cd kuma-ansible-installer
   ```

2. Depending on the type of application activation that you plan to use, do one of the following:

   - If you want to activate the application with a key file, place the key file in the <installer directory>/roles/kuma/files/ directory.

     The key file must be named license.key.

     ```
     sudo cp <key file>.key <installer directory>/roles/kuma/files/license.key
     ```

   - If you want to activate the application with a license code, go to the next step of the instructions.

3. From the directory with the extracted installer, start the installation of components using the prepared inventory file, distributed.inventory.yml:

   ```
   sudo ./install.sh distributed.inventory.yml
   ```

   If privilege elevation on the target machine requires specifying a password, run the following command and specify the password when prompted:

   ```
   sudo ./install.sh single.inventory.yml --ask-become-pass
   ```

4. If you want to use a non-standard port for the SSH connection during installation, select one of the following options:

   - If the port is the same for a group of hosts, specify it in the inventory file in the 'vars' group.

     Example of a distributed.inventory.yml inventory file for a high-availability installation with KUMA Core in a Raft cluster with the same non-standard port for all KUMA Core hosts

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
          ip: 0.0.0.0
          raft_node_addr: ""
          ansible_port: 2222
    kuma_core_peers:
      vars:
        ansible_port: 2222
      hosts:
        kuma-core-2.example.com:
          ip: 0.0.0.0
        kuma-core-3.example.com:
          ip: 0.0.0.0
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ip: 0.0.0.0
    kuma_storage:
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server4.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server5.example.com:
          ip: 0.0.0.0
```

```
        shard: 0
        replica: 0
        keeper: 1
      kuma-storage-cluster1-server6.example.com:
        ip: 0.0.0.0
        shard: 0
        replica: 0
        keeper: 2
      kuma-storage-cluster1-server7.example.com:
        ip: 0.0.0.0
        shard: 0
        replica: 0
        keeper: 3
```

- If a different non-standard port is used for the connection to each host, specify the port for each host in the inventory file.

  Example of a distributed.inventory.yml inventory file for a high-availability installation with KUMA Core in a Raft cluster with a different non-standard port for each KUMA Core host

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
          ip: 0.0.0.0
          raft_node_addr: ""
          ansible_port: 2222
    kuma_core_peers:
      vars:
        ansible_port: 2222
      hosts:
        kuma-core-2.example.com:
          ip: 0.0.0.0
        kuma-core-3.example.com:
          ip: 0.0.0.0
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ip: 0.0.0.0
    kuma_storage:
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server4.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server5.example.com:
          ip: 0.0.0.0
```

```
        shard: 0
        replica: 0
        keeper: 1
      kuma-storage-cluster1-server6.example.com:
        ip: 0.0.0.0
        shard: 0
        replica: 0
        keeper: 2
      kuma-storage-cluster1-server7.example.com:
        ip: 0.0.0.0
        shard: 0
        replica: 0
        keeper: 3
```

5. Accept the terms of the End User License Agreement.

   If you do not accept the terms of the End User License Agreement, the program will not be installed.

   Depending on the type of application activation, the installer produces one of the following results:

   - If you want to activate the license with a file and the key file is placed in the `<installer directory>`/roles/kuma/files/ directory, running the installer with the distributed.inventory.yml inventory file installs the LogAnalyzer Core on the server from the `kuma_core` section, additional LogAnalyzer Cores are installed on servers from the `kuma_core_peers` section, all services and OOTB resources specified in the inventory file are installed.

   - If you want to activate the application with a license code, or if you want to provide a key file later, running the installer with the distributed.inventory.yml inventory file only installs the LogAnalyzer Core on the server from the `kuma_core` section, and additional LogAnalyzer Cores on servers from the `kuma_core_peers` section.

     To install the services, specify the license code on the command line. Then run the postinstall.sh installer with the "distrtibuter.inventory.yml" inventory file.

     ```
     sudo ./postinstall.sh distributed.inventory.yml
     ```

     This creates the specified services. You can select the resources that you want to import from the repository.

6. After the installation is complete, log in to the LogAnalyzer web interface and enter the address of the LogAnalyzer web interface in the address bar of your browser, then enter your credentials on the login page.

   The address of the LogAnalyzer web interface is `https://<FQDN of the host where LogAnalyzer is installed>:7220`.

   Default login credentials:

   - user name: `admin`
   - password: `mustB3Ch@ng3d!`

   After the first login, change the password of the admin account

All LogAnalyzer components are installed and you are logged in to the web interface.

Subsequently, when installing LogAnalyzer services, you can specify multiple comma-separated URLs in the installation command to connect to the LogAnalyzer Core services in the Raft cluster.

If you are not using a load balancer in the Raft cluster, you can log in to a LogAnalyzer Core node directly. As you go from node to node, whenever you successfully authenticate on the next node, the session opened on the previous node is closed.

If you want to use domain authentication via AD FS to avoid having to authenticate on each node, you need to configure a balancer in the Raft cluster, specify the FQDN of the balancer in the AD FS settings, and then configure domain authentication via AD FS in LogAnalyzer by specifying the FQDN of the balancer in the settings.

All servers on which LogAnalyzer Core services are deployed must have the same time zone.

We recommend backing up the inventory file that you used to install the program. You can use this inventory file to add components to the system or remove LogAnalyzer.

Expanding the Raft cluster

Expanding the Raft cluster in LogAnalyzer involves the following steps:

1. Preparing servers for installing LogAnalyzer Core services

   The expand.sh installer, which you run on the control machine, prepares the necessary files on the target machines for the subsequent installation of additional LogAnalyzer Core services.

2. Creating and installing additional LogAnalyzer Core services

   You need to create additional LogAnalyzer Core services in the web interface and install the LogAnalyzer Core service on the servers specified in the kuma_core_peers group of the expand.inventory.yml inventory file (one server at a time).

3. Reflecting the changes in the distributed.inventory.yml inventory file

   We recommend updating the distributed.inventory.yml inventory file to reflect the changes made in case LogAnalyzer needs to be updated.

Preparing servers for installing LogAnalyzer Core services

To prepare servers for the subsequent installation of additional LogAnalyzer Core services:

1. On the control machine, go to the directory containing the extracted installer.

   ```
   cd kuma-ansible-installer
   ```

2. Depending on the type of application activation that you plan to use, do one of the following:

   - If you want to activate the application with a key file, place the key file in the <`installer directory`>/roles/kuma/files/ directory.

     The key file must be named license.key.

     ```
     sudo cp <key file>.key <installer directory>/roles/kuma/files/license.key
     ```

   - If you want to activate the application with a license code, go to the next step of the instructions.

3. Create an inventory file named expand.inventory.yml by copying the expand.inventory.yml.template file:

   cp expand.inventory.yml.template expand.inventory.yml

   Edit the settings in the expand.inventory.yml inventory file and specify the servers that you want to add in the 'kuma_core_peers' group. Together, the kuma_core and kuma_core_peers groups must contain an odd number of servers.

   Example of an expand.inventory.yml file for adding servers for additional LogAnalyzer Core services

   ```
   kuma:
     vars:
       no_firewall_actions: false
       ansible_connection: ssh
       ansible_user: root
     children:
       kuma_core_peers:
         hosts:
           kuma-core-4.example.com:
             ip: 0.0.0.0
           kuma-core-5.example.com:
             ip: 0.0.0.0
   ```

4. On the test machine, run the following command as root from the directory with the unpacked installer:

   ./expand.sh expand.inventory.yml

5. Accept the terms of the End User License Agreement.

   If you do not accept the terms of the End User License Agreement, the program will not be installed.

   Depending on the type of application activation, the installer produces one of the following results:

   - If you plan to activate the application using a key file, and the key file is placed in the <`installer directory`>/roles/kuma/files/ directory, the installer started with the expand.inventory.yml file prepares servers specified in the inventory file for installation of additional LogAnalyzer Core services.

   - If you want to activate the application with a license code or provide a key file later, running the installer with the "expand.inventory.yml" inventory file installs only LogAnalyzer Core.

Running this command on each target machine specified in the expand.inventory.yml inventory file creates files for creating and installing additional LogAnalyzer Core services.

Creating and installing additional LogAnalyzer Core services

LogAnalyzer Core services consist of a client part and a server part, therefore creating a service involves two steps:

1. Creating the client part of the LogAnalyzer Core service in the web interface.

   a. In the **Resources** → **Active services** section, click **Add** and in the drop-down list, select **Add Core service**.

   b. This opens the **Add Core service** window; in that window, specify the name of the service in the **Core service name** field and click **Add**. The added service appears in the **Services** list.

   c. Select the check box next to the added LogAnalyzer Core service and in the toolbar, click ⋮ to open a menu, and in the menu, select **Copy ID**. You will need the ID of the LogAnalyzer Core service to install the service on the server.

2. Creating the server part of the LogAnalyzer Core service.

   Run the following command on the target machine.

   ```
   sudo /opt/kaspersky/kuma/kuma core --raft.join <FQDN of the host from the kuma_core section where the first LogAnalyzer Core service is running>:7210 --id <LogAnalyzer Core service ID copied from the web interface> --install
   ```

   Repeat the steps to create the client and server parts of the LogAnalyzer Core service for each host in the kuma_core_peers group. Only one LogAnalyzer Core service can be installed on a host.

Reflecting the changes in the distributed.inventory.yml inventory file

Add the kuma_core_peers group to the distributed.inventory.yml inventory file and specify the servers for the additional LogAnalyzer Core services that you specified in the expand.inventory.yml inventory file so that the inventory file distributed.inventory.yml contains up-to-date information in case you need to update LogAnalyzer.

The Raft cluster is expanded. LogAnalyzer Core services are added to the Raft cluster and connected to core-1. When installing the resto of the LogAnalyzer services, you can specify multiple comma-separated URLs in the installation command to connect to the LogAnalyzer Core services in the Raft cluster.

Supported scenarios for installing and upgrading from 3.4.x to 4.x with the Core installed on a Raft cluster

Supported LogAnalyzer installation scenarios

Installation option for 3.4.x	Installation option for 4.x.x	Support and implementation
–	Single-server installation or distributed installation with one LogAnalyzer Core service	Performed automatically by the installer. The inventory file used is similar to the inventory file of version 3.4.x. The LogAnalyzer Core service on a host specified in the `kuma_core` section has the name 'core-1' and the ID 00000000-0000-0000-0000-000000000000.
–	Single-server installation with LogAnalyzer Core in a Raft cluster or distributed installation with LogAnalyzer Core in a Raft cluster	Performed automatically by the installer. The installer prepares hosts from the additional `kuma_core_peers` section, automatically creates additional LogAnalyzer Core services, and includes these in the Raft cluster by installing them on hosts from this section. The LogAnalyzer Core service on a host from the `kuma_core` section has the name 'core-1' and the ID 00000000-0000-0000-0000-000000000000. The LogAnalyzer Core services on hosts from the `kuma_core_peers` section have names of the 'core-<host_name_in_inventory_file>' form and unique IDs.

Installation option for 3.4.x	Installation option for 4.x.x	Support and implementation
Creating a Raft cluster in a 4.xx installation with one LogAnalyzer Core service		You can configure additional hosts using the expand.sh installer and the expand.inventory.yml inventory file with a `kuma_core_peers` section. After that, you need to create additional LogAnalyzer Core services in the LogAnalyzer web interface and use the installation command to install the LogAnalyzer Core services on the prepared hosts, one host at a time.
Single-server installation or distributed installation	Single-server installation with one LogAnalyzer Core service in a Raft cluster or distributed installation with one LogAnalyzer Core service in a Raft cluster	Performed automatically by the installer. The inventory file used is similar to the 3.4.X inventory. The name and ID of the LogAnalyzer Core service on a host from the `kuma_core` section are not changed.
Single-server installation or distributed installation	Single-server installation with multiple LogAnalyzer Core services in a Raft cluster or distributed installation with multiple LogAnalyzer Core services in a Raft cluster	Each LogAnalyzer Core service must be hosted on a separate host. The installer only prepares hosts from the additional `kuma_core_peers` section for the subsequent manual installation of additional LogAnalyzer Core services. After upgrading to version 4.x.x, you need to create additional LogAnalyzer Core services in the LogAnalyzer web interface, which are then automatically added to the cluster. If necessary, you can configure additional hosts after the upgrade using the expand.sh installer and the expand.inventory.yml inventory file, create LogAnalyzer Core services in the LogAnalyzer web interface, install LogAnalyzer Core services on additional hosts, and include services in the cluster. During the upgrade, the `kuma_core` and `kuma_core_peers` sections are checked for uniqueness of hosts, similar to installing from scratch. If you populated the `kuma_core_peers` section during the upgrade, the actions that the install.sh installer performs for it are similar to the actions performed by the individual launch of the expand.sh installer.
–	Installing a single LogAnalyzer Core service on a Kubernetes cluster	You can install the LogAnalyzer Core service as one instance with two volume replicas. You can use an inventory file similar to version 3.4.x. For high availability, you need at least two worker nodes.
Installation with LogAnalyzer Core in a Kubernetes cluster	Installing a single LogAnalyzer Core service on a Kubernetes cluster	The LogAnalyzer Core service stays as one instance with two volume replicas. The inventory file used is similar to the inventory file of version 3.4.x. Multiple LogAnalyzer Core services are not supported in 4.x.x.
Migrating a 4.x.x installation with one LogAnalyzer Core service to an installation with one LogAnalyzer Core service in a Kubernetes cluster		The procedure is similar to migration to a high-availablity installation in previous versions of LogAnalyzer. You can install the LogAnalyzer Core service as one instance with two volume replicas. The inventory file used is similar to the inventory file of version 3.4.x. For high availability, you need at least two worker nodes. The host of the LogAnalyzer Core service becomes one of the worker nodes of the cluster.

Backing up LogAnalyzer

LogAnalyzer allows you to back up the LogAnalyzer Core database and certificates. The backup functionality is intended for restoring LogAnalyzer. To migrate or copy resources, use the resource export and import functionality instead.

You can perform backup using the REST API.

You can also back up individual partitions using the archival procedure.

Special considerations for LogAnalyzer backup

- Data can only be restored from a backup copy created in the same version of LogAnalyzer.

  If you perform a backup in LogAnalyzer 4.0 or later, and then restore the backup to another device with LogAnalyzer 4.0 using the REST API (`system/backup → system/restore`), alert events are not restored. To restore alert events in ClickHouse, you need to install a storage with the same storage cluster ID, where in the *kuma.alert_events_local* table, event records can be matched to alerts by IDs. If the storage changes, alert events can be viewed provided that a backup copy is created and then the ClickHouse database is restored.

- Backing up collectors is not necessary, except collectors with an SQL connection. When restoring such collectors, you need to revert the ID to the original value.

Modifying the configuration of LogAnalyzer

The LogAnalyzer configuration can be modified in the following ways.

- Expanding an all-in-one installation to a distributed installation.

To expand an all-in-one installation to a distributed installation:

1. Create a backup copy of LogAnalyzer.

2. Remove the pre-installed correlator, collector, storage, and event router services from the server.

   a. In the LogAnalyzer web interface, under **Resources** → **Active services**, select a service and click **Copy ID**. On the server where the services were installed, run the service removal command:

   ```
   sudo /opt/kaspersky/kuma/kuma <collector/correlator/storage/eventRouter> --id
   <service ID copied from the KUMA web interface> --uninstall
   ```

   Repeat the removal command for each service.

   b. Then remove the services in the LogAnalyzer web interface:

As a result, only the LogAnalyzer Core remains on the initial installation server.

3. Prepare the distributed.inventory.yml inventory file and in that file, specify the initial all-in-one initial installation server in the `kuma_core` group.

   In this way, the LogAnalyzer Core remains on the original server, and you can deploy the other components on other servers. In the inventory file, specify the servers on which you want to install the LogAnalyzer components.

   Example inventory file for expanding an all-in-one installation to a distributed installation

```
all:
  vars:
    deploy_to_k8s: false
    need_transfer: false
    generate_etc_hosts: false
    deploy_example_services: false
    no_firewall_actions: false
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_core:
      hosts:
        kuma-core-1.example.com:
          ip: 0.0.0.0
    kuma_collector:
      hosts:
        kuma-collector-1.example.com:
          ip: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
          ip: 0.0.0.0
    kuma_storage:
      hosts:
        kuma-storage-cluster1-server1.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server2.example.com:
          ip: 0.0.0.0
          shard: 1
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server3.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 1
          keeper: 0
        kuma-storage-cluster1-server4.example.com:
          ip: 0.0.0.0
          shard: 2
          replica: 2
          keeper: 0
        kuma-storage-cluster1-server5.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 1
        kuma-storage-cluster1-server6.example.com:
          ip: 0.0.0.0
          shard: 0
          replica: 0
          keeper: 2
        kuma-storage-cluster1-server7.example.com:
          ip: 0.0.0.0
```

```
            shard: 0
            replica: 0
            keeper: 3
```

4. Create and install the storage, collector, correlator, event router, and agent services on other machines.

   a. After you specify the settings in all sections of the distributed.inventory.yml file, run the installer on the control machine.

      ```
      sudo ./install.sh distributed.inventory.yml
      ```

      This command creates files necessary to install the LogAnalyzer components (storage, collectors, correlators) on each target machine specified in distributed.inventory.yml.

   b. Create storage, collector, and correlator services.

The expansion of the installation is completed.

- **Adding servers for collectors to a distributed installation**.

The following instructions describe adding one or more servers to an existing infrastructure to then install collectors on these servers to balance the load. You can use these instructions as an example and adapt them according to your needs.

To add servers to a distributed installation:

1. Ensure that the target machines meet hardware, software, and installation requirements.

2. On the control machine, go to the directory with the extracted LogAnalyzer installer by running the following command:

   ```
   cd kuma-ansible-installer
   ```

3. Create an inventory file named expand.inventory.yml by copying the expand.inventory.yml.template file:

   ```
   cp expand.inventory.yml.template expand.inventory.yml
   ```

4. Edit the settings in the expand.inventory.yml inventory file and specify the servers that you want to add in the kuma_collector section.

   Example expand.inventory.yml inventory file for adding collector servers

   ```
   kuma:
     vars:
       ansible_connection: ssh
       ansible_user: root
     children:
       kuma_collector:
         kuma-additional-collector1.example.com
         kuma-additional-collector2.example.com
       kuma_correlator:
       kuma_storage:
         hosts:
   ```

5. On the test machine, run the following command as root from the directory with the unpacked installer:

   ./expand.sh expand.inventory.yml

   This command creates files for creating and installing the collector on each target machine specified in the expand.inventory.yml inventory file.

6. Create and install the collectors. A LogAnalyzer collector consists of a client part and a server part, therefore creating a collector involves two steps.

    1. Creating the client part of the collector, which includes a resource set and the collector service.

        To create a resource set for a collector, in the LogAnalyzer web interface, in the **Resources** → **Collectors** section, click **Add collector** and edit the settings. For more details, see Creating a collector.

        At the last step of the configuration wizard, after you click **Create and save**, a resource set for the collector is created and the collector service is automatically created. The command for installing the service on the server is also automatically generated and displayed on the screen. Copy the installation command and proceed to the next step.

    2. Creating the server part of the collector.

    a. On the target machine, run the command you copied at the previous step. The command looks as follows, but all parameters are filled in automatically.

        ```
        sudo /opt/kaspersky/kuma/kuma <storage> --core https://<KUMA Core server FQDN>:<port used by KUMA Core for internal communication (port 7210 by default)> --id <service ID copied from the KUMA web interface> --install
        ```

        The collector service is installed on the target machine. You can check the status of the service in the web interface under **Resources** → **Active services**.

    b. Run the same command on each target machine specified in the expand.inventory.yml inventory file.

7. Add the new servers to the distributed.inventory.yml inventory file so that it has up-to-date information in case you need to upgrade LogAnalyzer.

Servers are successfully added.

- Adding servers to an existing storage cluster

The following instructions describe adding multiple servers to an existing storage cluster. You can use these instructions as an example and adapt them to your requirements.

To add servers to an existing storage cluster:

1. Ensure that the target machines meet hardware, software, and installation requirements.

2. On the control machine, go to the directory with the extracted LogAnalyzer installer by running the following command:

   ```
   cd kuma-ansible-installer
   ```

3. Create an inventory file named expand.inventory.yml by copying the expand.inventory.yml.template file:

   ```
   cp expand.inventory.yml.template expand.inventory.yml
   ```

4. Edit the settings in the expand.inventory.yml inventory file and specify the servers that you want to add in the 'storage' section. In the following example, the 'storage' section specifies servers for installing two shards, each of which contains two replicas. In the expand.inventory.yml inventory file, you must only specify the FQDN; you will assign the roles of shards and replicas later in the LogAnalyzer web interface as you follow the steps of these instructions. You can adapt this example according to your needs.

   Example expand.inventory.yml inventory file for adding servers to an existing storage cluster

   ```
   kuma:
     vars:
       ansible_connection: ssh
       ansible_user: root
     children:
       kuma_collector:
       kuma_correlator:
       kuma_storage:
         hosts:
           kuma-storage-cluster1-server8.example.com:
           kuma-storage-cluster1-server9.example.com:
           kuma-storage-cluster1-server10.example.com:
           kuma-storage-cluster1-server11.example.com:
   ```

5. On the test machine, run the following command as root from the directory with the unpacked installer:

   ```
   ./expand.sh expand.inventory.yml
   ```

   ```
   Running this command on each target machine specified in the
   expand.inventory.yml inventory file creates files for creating and installing
   the storage.
   ```

6. You do not need to create a separate storage because you are adding servers to an existing storage cluster. Edit the storage settings of the existing cluster:

   a. In the **Resources** → **Storages** section, select an existing storage and open the storage for editing.

   b. In the **ClickHouse cluster nodes** section, click **Add nodes** and specify roles in the fields for the new node. The following example describes how to specify IDs to add two shards, containing two replicas each, to an existing cluster. You can adapt this example according to your needs.

      Example:

      ClickHouse cluster nodes

      <existing nodes>

      FQDN: kuma-storage-cluster1server8.example.com

      Shard ID: 1

      Replica ID: 1

      Keeper ID: 0

      FQDN: kuma-storage-cluster1server9.example.com

      Shard ID: 1

      Replica ID: 2

      Keeper ID: 0

      FQDN: kuma-storage-cluster1server9.example.com

      Shard ID: 2

      Replica ID: 1

      Keeper ID: 0

      FQDN: kuma-storage-cluster1server10.example.com

      Shard ID: 2

      Replica ID: 2

      Keeper ID: 0

   c. Save the storage settings.

      Now you can create storage services for each ClickHouse cluster node.

7. To create a storage service, in the LogAnalyzer web interface, in the **Resources** → **Active services** section, click **Add service**.

   This opens the **Choose a service** window; in that window, select the storage you edited at the previous step and click **Create service**. Do the same for each ClickHouse storage node you are adding.

   As a result, the number of created services must be the same as the number of nodes being added to the ClickHouse cluster, for example, four services for four nodes. The created storage services are displayed in the LogAnalyzer web interface in the **Resources** → **Active services** section.

8. Now storage services must be installed on each server by using the service ID.

   a. In the LogAnalyzer web interface, in the **Resources** → **Active services** section, select the storage service that you need and click **Copy ID**.

      The service ID is copied to the clipboard; you need it for running the service installation command.

   b. Compose and run the following command on the target machine:

      ```
      sudo /opt/kaspersky/kuma/kuma <storage> --core https://<KUMA Core server FQDN>:<port used by KUMA Core for internal communication (port 7210 by default)> --id <service ID copied from the KUMA web interface> --install
      ```

      The storage service is installed on the target machine. You can check the status of the service in the web interface under **Resources** → **Active services**.

   c. Run the storage service installation command on each target machine listed in the 'storage' section of the expand.inventory.yml inventory file, one machine at a time. On each machine, the unique service ID within the cluster must be specified in the installation command.

9. To apply changes to a running cluster, in the LogAnalyzer web interface, under **Resources** → **Active services**, select the check boxes next to all storage services in the cluster that you are expanding and click **Update configuration**. Changes are applied without stopping services.

10. Add the new servers to the distributed.inventory.yml inventory file so that it has up-to-date information in case you need to upgrade LogAnalyzer.

Servers are successfully added to a storage cluster.

- Adding another storage cluster.

The following instructions describe adding an extra storage cluster to an existing infrastructure. You can use these instructions as an example and adapt them to suit your needs.

To add a storage cluster:

1. Ensure that the target machines meet hardware, software, and installation requirements.

2. On the control machine, go to the directory with the extracted LogAnalyzer installer by running the following command:

   ```
   cd kuma-ansible-installer
   ```

3. Create an inventory file named expand.inventory.yml by copying the expand.inventory.yml.template file:

   ```
   cp expand.inventory.yml.template expand.inventory.yml
   ```

4. Edit the settings in the expand.inventory.yml inventory file and specify the servers that you want to add in the 'storage' section. In the following example, the 'storage' section specifies servers for installing three dedicated keepers and two shards, each of which contains two replicas. In the expand.inventory.yml inventory file, you must only specify the FQDN; you will assign the roles of keepers, shards, and replicas later in the LogAnalyzer web interface by following the steps of these instructions. You can adapt this example to suit your needs.

   Example expand.inventory.yml inventory file for adding a storage cluster

   ```
   kuma:
     vars:
       ansible_connection: ssh
       ansible_user: root
     children:
       kuma_collector:
       kuma_correlator:
       kuma_storage:
         hosts:
           kuma-storage-cluster2-server1.example.com
           kuma-storage-cluster2-server2.example.com
           kuma-storage-cluster2-server3.example.com
           kuma-storage-cluster2-server4.example.com
           kuma-storage-cluster2-server5.example.com
           kuma-storage-cluster2-server6.example.com
           kuma-storage-cluster2-server7.example.com
   ```

5. On the test machine, run the following command as root from the directory with the unpacked installer:

   ```
   ./expand.sh expand.inventory.yml
   ```

   ```
   This command creates files for creating and installing the storage on each
   target machine specified in the expand.inventory.yml inventory file.
   ```

6. Create and install the storage. For each storage cluster, you must create a separate storage, for example, three storages for three storage clusters. A storage consists of a client part and a server part, therefore creating a storage involves two steps.

    1. Creating the client part of the storage, which includes a resource set and the storage service.

        a. To create a resource set for a storage, in the LogAnalyzer web interface, under **Resources** → **Storages**, click **Add storage** and edit the settings. In the **ClickHouse cluster nodes** section, specify roles for each server that you are adding: keeper, shard, replica. For more details, see Creating a resource set for a storage.

        The created resource set for the storage is displayed in the **Resources** → **Storages** section. Now you can create storage services for each ClickHouse cluster node.

        b. To create a storage service, in the LogAnalyzer web interface, in the **Resources** → **Active services** section, click **Add service**.

        This opens the **Choose a service** window; in that window, select the resource set that you created for the storage at the previous step and click **Create service**. Do the same for each ClickHouse storage.

        As a result, the number of created services must be the same as the number of nodes in the ClickHouse cluster, for example, fifty services for fifty nodes. The created storage services are displayed in the LogAnalyzer web interface in the **Resources** → **Active services** section. Now you need to install storage services on each node of the ClickHouse cluster by using the service ID.

    2. Creating the server part of the storage.

    a. On the target machine, create the server part of the storage: in the LogAnalyzer web interface, in the **Resources** → **Active services** section, select a storage service and click **Copy ID**.

    The service ID is copied to the clipboard; you need it for running the service installation command.

    b. Compose and run the following command on the target machine:

    ```
    sudo /opt/kaspersky/kuma/kuma <storage> --core https://<KUMA Core server
    FQDN>:<port used by KUMA Core for internal communication (port 7210 by
    default)> --id <service ID copied from the KUMA web interface> --install
    ```

    The storage service is installed on the target machine. You can check the status of the service in the web interface under **Resources** → **Active services**.

    c. Run the storage service installation command on each target machine listed in the 'storage' section of the expand.inventory.yml inventory file, one machine at a time. On each machine, the unique service ID within the cluster must be specified in the installation command.

    d. Dedicated keepers are automatically started immediately after installation and are displayed in the **Resources** → **Active services** section with the green status. Services on other storage nodes may not start until services are installed for all nodes in that cluster. Up to that point, services can be displayed with the red status. This is normal behavior when creating a new storage cluster or adding nodes to an existing storage cluster. As soon as the service installation command is run on all nodes of the cluster, all services get the green status.

7. Add the new servers to the distributed.inventory.yml inventory file so that it has up-to-date information in case you need to upgrade LogAnalyzer.

The extra storage cluster is successfully added.

- Removing servers from a distributed installation.

To remove a server from a distributed installation:

1. Remove all services from the server that you want to remove from the distributed installation.

   a. Remove the server part of the service. Copy the service ID in the LogAnalyzer web interface and run the following command on the target machine:

   ```
   sudo /opt/kaspersky/kuma/kuma <collector/correlator/storage> --core
   https://<KUMA Core server FQDN>:<port used by KUMA Core for internal
   communication (port 7210 by default)> --id <service ID copied from the KUMA
   web interface> --install
   ```

   b. Remove the client part of the service in the LogAnalyzer web interface in the **Active services** → **Delete** section.

   The service is removed.

2. Repeat step 1 for each server that you want to remove from the infrastructure.

3. Remove the servers from the relevant sections of the distributed.inventory.yml inventory file to make sure the inventory file has up-to-date information in case you need to upgrade LogAnalyzer.

Servers are removed from the distributed installation.

- Removing a storage cluster from a distributed installation.

  To remove one or more storage clusters from a distributed installation:

  1. Remove the storage service on each cluster server that you want to remove from the distributed installation.

     a. Remove the server part of the storage service. Copy the service ID in the LogAnalyzer web interface and run the following command on the target machine:

     ```
     sudo /opt/kaspersky/kuma/kuma <storage> --id <service ID> --uninstall
     ```

     Repeat for each server.

     b. Remove the client part of the service in the LogAnalyzer web interface in the **Resources** → **Active services** → **Delete** section.

     The service is removed.

  2. Remove servers from the 'storage' section of the distributed.inventory.yml inventory file to make sure the inventory file has up-to-date information in case you need to upgrade LogAnalyzer or modify its configuration.

  The cluster is removed from the distributed installation.

- Migrating the LogAnalyzer Core to a new Kubernetes cluster

To migrate the LogAnalyzer Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the LogAnalyzer Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed LogAnalyzer Core on all hosts on which you want to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

Resolving the LogAnalyzer Core migration error

Migration of the LogAnalyzer Core from a host to a new Kubernetes cluster may be stopped due to a timeout at the `Deploy Core transfer job` step. In this case, the following error message is recorded in the log of core-transfer migration tasks:

```
cp: can't stat '/mnt/kuma-source/core/.lic': No such file or directory
```

To prevent this error, before you start migrating the LogAnalyzer Core:

1. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.
2. In the core-transfer-job.yaml.j2 file, find the following lines:

   ```
   cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

3. Edit these lines as follows, making sure you keep the indentation (number of space characters):

   ```
   cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

4. Save the changes to the file.

You can then restart the distributed installation using the prepared k0s.inventory.yml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If you started migrating the LogAnalyzer Core from a host to a new Kubernetes cluster and the migration failed with an error, follow the steps below to fix the error.

To fix the error after attempting to migrate the LogAnalyzer Core from a host to a new Kubernetes cluster:

1. On any controller of the cluster, delete the Ingress object by running the following command:

   ```
   sudo k0s kubectl delete daemonset/ingress -n ingress
   ```

2. Check if a migration job exists in the cluster:

   ```
   sudo k0s kubectl get jobs -n kuma
   ```

3. If a migration job exists, delete it:

   ```
   sudo k0s kubectl delete job core-transfer -n kuma
   ```

4. Go to the console of a host from the kuma_core group.
5. Start the LogAnalyzer Core services by running the following commands:

   ```
   sudo systemctl start kuma-mongodb
   sudo systemctl start kuma-core-00000000-0000-0000-0000-000000000000
   ```

6. Make sure that the kuma-core-00000000-0000-0000-0000-000000000000 service has been successfully started:

   ```
   sudo systemctl status kuma-core-00000000-0000-0000-0000-000000000000
   ```

7. Make sure that the kuma_core group has access to the KUMA interface by host FQDN.

   Other hosts do not need to be running.

8. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.

9. In the core-transfer-job.yaml.j2 file, find the following lines:

   ```
   cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

10. Edit these lines as follows, making sure you keep the indentation (number of space characters):

    ```
    cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
    cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/ &&
    ```

11. Save the changes to the file.

    You can then restart the distributed installation using the prepared k0s.inventory.yaml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If the component is not detected on the worker nodes, a clean installation of the LogAnalyzer Core is performed in the cluster without migrating resources to it. Existing components must be manually recreated with the new Core in the LogAnalyzer web interface.

For collectors and storages from the inventory file, certificates for communication with the Core inside the cluster will be reissued. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed is entered into the ConfigMap.

Upgrading older versions of LogAnalyzer

The upgrade procedure is the same for all hosts and involves using the installer and inventory file.

Version upgrade scheme:

2.0.x → 2.1.3 → 3.0.3 → 3.2.x → 3.4.x → 4.0

2.1.x → 2.1.3 → 3.0.3 → 3.2.x → 3.4.x → 4.0

2.1.3 → 3.0.3 → 3.2.x → 3.4.x → 4.0

3.0.x → 3.0.3 → 3.2.x → 3.4.x → 4.0

If you made manual changes to the unit files of LogAnalyzer services in the /usr/lib/systemd/system directory, these changes are lost when upgrading from a previous version of LogAnalyzer. If necessary, make these changes again after upgrading the application.

Upgrading from version 2.0.x to 2.1.3

To install LogAnalyzer version 2.1.3 over version 2.0.x, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Create a backup copy of the LogAnalyzer Core. If necessary, you will be able to recover from a backup copy for version 2.0.

   LogAnalyzer backups created in versions 2.0 and earlier cannot be restored in version 2.1.3. This means that you cannot install LogAnalyzer 2.1.3 and restore a LogAnalyzer 2.0 backup in it.

   Create a backup copy immediately after upgrading LogAnalyzer to version 2.1.3.

2. Make sure that all application installation requirements are met.

3. Make sure that MongoDB versions are compatible by running the following commands on the LogAnalyzer Core device:

   ```
   cd /opt/kaspersky/kuma/mongodb/bin/
   ./mongo
   use kuma
   db.adminCommand({getParameter: 1, featureCompatibilityVersion: 1})
   ```

   If the component version is different from 4.4, set the value to 4.4 using the following command:

   ```
   db.adminCommand({ setFeatureCompatibilityVersion: "4.4" })
   ```

4. During installation or upgrade, make sure that TCP port 7220 on the LogAnalyzer Core is accessible from the LogAnalyzer storage hosts.

5. If you have a keeper deployed on a separate device in the ClickHouse cluster, install the storage service on the same device before you start the upgrade:

   - Use the existing storage of the cluster to create a storage service for the keeper in the web interface.
   - Install the service on the device with the dedicated ClickHouse keeper.

6. In the inventory file, specify the same hosts that were used when installing LogAnalyzer version 2.0.X. Set the following settings to `false`:

   ```
   deploy_to_k8s false
   need_transfer false
   deploy_example_services false
   ```

   When the installer uses this inventory file, all KUMA components are upgraded to version 2.1.3. The available services and storage resources are also reconfigured on hosts from the kuma_storage group:

   - ClickHouse systemd services are deleted.
   - Certificates are deleted from the /opt/kaspersky/kuma/clickhouse/certificates directory.
   - The 'Shard ID', 'Replica ID', 'Keeper ID', and 'ClickHouse configuration override' fields are filled in for each node in the storage resource based on values from the inventory file and service configuration files on the host. Subsequently, you will manage the roles of each node in the LogAnalyzer web interface.
   - All existing configuration files from the /opt/kaspersky/kuma/clickhouse/cfg directory are deleted (subsequently, they will be generated by the storage service).
   - The value of the LimitNOFILE parameter ('Service' section) is changed from 64,000 to 500,000 in the kuma-storage systemd services.

7. If you use alert segmentation rules, prepare the data for migrating the existing rules and save. In the next step, you can use this data to re-create the rules. During the upgrade, alert segmentation rules are not migrated automatically.

8. To perform an upgrade, you need a valid password from the admin user. If you forgot the password of the admin user, contact Technical Support to reset the current password, then use the new password to perform the upgrade at the next step.

Upgrading LogAnalyzer

1. Depending on the KUMA deployment scheme that you are using, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the application.

- Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high availability configuration.

  If an inventory file is not available for the current version, use the provided inventory file template and edit it as necessary. To view a list of hosts and host roles in the current LogAnalyzer system, in the web interface, go to **Resources** → **Active services** section.

  The upgrade process mirrors the installation process.

  If you want to upgrade from a distributed installation to a distributed installation in a high availability configuration, you must first upgrade the distributed installation and then **migrate the Core to a Kubernetes cluster**.

To migrate the LogAnalyzer Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the LogAnalyzer Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed LogAnalyzer Core on all hosts on which you want to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

Resolving the LogAnalyzer Core migration error

Migration of the LogAnalyzer Core from a host to a new Kubernetes cluster may be stopped due to a timeout at the `Deploy Core transfer job` step. In this case, the following error message is recorded in the log of core-transfer migration tasks:

```
cp: can't stat '/mnt/kuma-source/core/.lic': No such file or directory
```

To prevent this error, before you start migrating the LogAnalyzer Core:

1. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.
2. In the core-transfer-job.yaml.j2 file, find the following lines:

   ```
   cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

3. Edit these lines as follows, making sure you keep the indentation (number of space characters):

   ```
   cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/
   &&
   ```

4. Save the changes to the file.

   You can then restart the distributed installation using the prepared k0s.inventory.yml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If you started migrating the LogAnalyzer Core from a host to a new Kubernetes cluster and the migration failed with an error, follow the steps below to fix the error.

To fix the error after attempting to migrate the LogAnalyzer Core from a host to a new Kubernetes cluster:

1. On any controller of the cluster, delete the Ingress object by running the following command:

   ```
   sudo k0s kubectl delete daemonset/ingress -n ingress
   ```

2. Check if a migration job exists in the cluster:

   ```
   sudo k0s kubectl get jobs -n kuma
   ```

3. If a migration job exists, delete it:

   ```
   sudo k0s kubectl delete job core-transfer -n kuma
   ```

4. Go to the console of a host from the kuma_core group.
5. Start the LogAnalyzer Core services by running the following commands:

   ```
   sudo systemctl start kuma-mongodb
   sudo systemctl start kuma-core-00000000-0000-0000-0000-000000000000
   ```

6. Make sure that the kuma-core-00000000-0000-0000-0000-000000000000 service has been successfully started:

   ```
   sudo systemctl status kuma-core-00000000-0000-0000-0000-000000000000
   ```

7. Make sure that the kuma_core group has access to the KUMA interface by host FQDN.

   Other hosts do not need to be running.

8. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.

9. In the core-transfer-job.yaml.j2 file, find the following lines:

   ```
   cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

10. Edit these lines as follows, making sure you keep the indentation (number of space characters):

    ```
    cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
    cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/
    &&
    ```

11. Save the changes to the file.

    You can then restart the distributed installation using the prepared k0s.inventory.yaml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If the component is not detected on the worker nodes, a clean installation of the LogAnalyzer Core is performed in the cluster without migrating resources to it. Existing components must be manually recreated with the new Core in the LogAnalyzer web interface.

For collectors and storages from the inventory file, certificates for communication with the Core inside the cluster will be reissued. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin

- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed is entered into the ConfigMap.

1. When upgrading on systems that contain large amounts of data and are operating with limited resources, the system may return the 'Wrong admin password' error message after you enter the administrator password. If you specify the correct password, LogAnalyzer may still return this error because of LogAnalyzer being unable to start the Core service due to a timeout error and resource limitations. If you enter the administrator password three times without waiting for the installation to complete, the update may end with a fatal error. Resolve the timeout error to proceed with the update.

The final stage of preparing LogAnalyzer for work

1. After upgrading LogAnalyzer, clear your browser cache.
2. Re-create the alert segmentation rules.
3. Manually upgrade the LogAnalyzer agents.

LogAnalyzer is successfully upgraded.

Upgrading from version 2.1.x to 2.1.3

To install LogAnalyzer version 2.1.3 over version 2.1.x, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Create a backup copy of the LogAnalyzer Core. If necessary, you will be able to recover from a backup copy for version 2.1.x.

   LogAnalyzer backups created in versions earlier than 2.1.3 cannot be restored in version 2.1.3. This means that you cannot install LogAnalyzer 2.1.3 and restore a LogAnalyzer 2.1.x backup in it.

   Create a backup copy immediately after upgrading LogAnalyzer to version 2.1.3.

2. Make sure that all application installation requirements are met.

3. During installation or upgrade, make sure that TCP port 7220 on the LogAnalyzer Core is accessible from the LogAnalyzer storage hosts.

4. To perform an upgrade, you need a valid password from the admin user. If you forgot the password of the admin user, contact Technical Support to reset the current password, then use the new password to perform the upgrade at the next step.

Upgrading LogAnalyzer

1. Depending on the KUMA deployment scheme that you are using, do one the following:

   - Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the application.

   - Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high availability configuration.

   If an inventory file is not available for the current version, use the provided inventory file template and edit it as necessary. To view a list of hosts and host roles in the current LogAnalyzer system, in the web interface, go to **Resources** → **Active services** section.

   The upgrade process mirrors the installation process.

   If you want to upgrade from a distributed installation to a distributed installation in a high availability configuration, you must first upgrade the distributed installation and then **migrate the Core to a Kubernetes cluster**.

To migrate the LogAnalyzer Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the LogAnalyzer Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed LogAnalyzer Core on all hosts on which you want to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

Resolving the LogAnalyzer Core migration error

Migration of the LogAnalyzer Core from a host to a new Kubernetes cluster may be stopped due to a timeout at the `Deploy Core transfer job` step. In this case, the following error message is recorded in the log of core-transfer migration tasks:

```
cp: can't stat '/mnt/kuma-source/core/.lic': No such file or directory
```

To prevent this error, before you start migrating the LogAnalyzer Core:

1. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.
2. In the core-transfer-job.yaml.j2 file, find the following lines:

   ```
   cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

3. Edit these lines as follows, making sure you keep the indentation (number of space characters):

   ```
   cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/
   &&
   ```

4. Save the changes to the file.

   You can then restart the distributed installation using the prepared k0s.inventory.yml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If you started migrating the LogAnalyzer Core from a host to a new Kubernetes cluster and the migration failed with an error, follow the steps below to fix the error.

To fix the error after attempting to migrate the LogAnalyzer Core from a host to a new Kubernetes cluster:

1. On any controller of the cluster, delete the Ingress object by running the following command:

   ```
   sudo k0s kubectl delete daemonset/ingress -n ingress
   ```

2. Check if a migration job exists in the cluster:

   ```
   sudo k0s kubectl get jobs -n kuma
   ```

3. If a migration job exists, delete it:

   ```
   sudo k0s kubectl delete job core-transfer -n kuma
   ```

4. Go to the console of a host from the kuma_core group.
5. Start the LogAnalyzer Core services by running the following commands:

   ```
   sudo systemctl start kuma-mongodb
   sudo systemctl start kuma-core-00000000-0000-0000-0000-000000000000
   ```

6. Make sure that the kuma-core-00000000-0000-0000-0000-000000000000 service has been successfully started:

   ```
   sudo systemctl status kuma-core-00000000-0000-0000-0000-000000000000
   ```

7. Make sure that the kuma_core group has access to the KUMA interface by host FQDN.

   Other hosts do not need to be running.

8. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.

9. In the core-transfer-job.yaml.j2 file, find the following lines:

   ```
   cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

10. Edit these lines as follows, making sure you keep the indentation (number of space characters):

    ```
    cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
    cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/
    &&
    ```

11. Save the changes to the file.

    You can then restart the distributed installation using the prepared k0s.inventory.yaml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If the component is not detected on the worker nodes, a clean installation of the LogAnalyzer Core is performed in the cluster without migrating resources to it. Existing components must be manually recreated with the new Core in the LogAnalyzer web interface.

For collectors and storages from the inventory file, certificates for communication with the Core inside the cluster will be reissued. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin

- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed is entered into the ConfigMap.

2. When upgrading on systems that contain large amounts of data and are operating with limited resources, the system may return the 'Wrong admin password' error message after you enter the administrator password. If you specify the correct password, LogAnalyzer may still return this error because of LogAnalyzer being unable to start the Core service due to a timeout error and resource limitations. If you enter the administrator password three times without waiting for the installation to complete, the update may end with a fatal error. Resolve the timeout error to proceed with the update.

The final stage of preparing LogAnalyzer for work

1. After upgrading LogAnalyzer, clear your browser cache.
2. Manually upgrade the LogAnalyzer agents.

LogAnalyzer is successfully upgraded.

Upgrading from version 2.1.3 to 3.0.3

To install LogAnalyzer version 3.0.3 over version 2.1.3, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Create a backup copy of the LogAnalyzer Core. If necessary, you can restore data from backup for version 3.0.3.

   LogAnalyzer backups created in versions 2.1.3 and earlier cannot be restored in version 3.0.3. This means that you cannot install LogAnalyzer 3.0.3 and restore a LogAnalyzer 2.1.3 backup in it.

   Create a backup copy immediately after upgrading LogAnalyzer to version 3.0.3.

2. Make sure that all application installation requirements are met.

3. During installation or upgrade, make sure that TCP port 7220 on the LogAnalyzer Core is accessible from the LogAnalyzer storage hosts.

Upgrading LogAnalyzer

Depending on the KUMA deployment scheme that you are using, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the application.

- Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high availability configuration.

  If an inventory file is not available for the current version, use the provided inventory file template and edit it as necessary. To view a list of hosts and host roles in the current LogAnalyzer system, in the web interface, go to **Resources** → **Active services** section.

The upgrade process mirrors the installation process.

If you want to upgrade from a distributed installation to a distributed installation in a high availability configuration, you must first upgrade the distributed installation and then **migrate the Core to a Kubernetes cluster**.

To migrate the LogAnalyzer Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the LogAnalyzer Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed LogAnalyzer Core on all hosts on which you want to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

Resolving the LogAnalyzer Core migration error

Migration of the LogAnalyzer Core from a host to a new Kubernetes cluster may be stopped due to a timeout at the `Deploy Core transfer job` step. In this case, the following error message is recorded in the log of core-transfer migration tasks:

```
cp: can't stat '/mnt/kuma-source/core/.lic': No such file or directory
```

To prevent this error, before you start migrating the LogAnalyzer Core:

1. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.
2. In the core-transfer-job.yaml.j2 file, find the following lines:

   ```
   cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

3. Edit these lines as follows, making sure you keep the indentation (number of space characters):

   ```
   cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

4. Save the changes to the file.

You can then restart the distributed installation using the prepared k0s.inventory.yml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If you started migrating the LogAnalyzer Core from a host to a new Kubernetes cluster and the migration failed with an error, follow the steps below to fix the error.

To fix the error after attempting to migrate the LogAnalyzer Core from a host to a new Kubernetes cluster:

1. On any controller of the cluster, delete the Ingress object by running the following command:

   ```
   sudo k0s kubectl delete daemonset/ingress -n ingress
   ```

2. Check if a migration job exists in the cluster:

   ```
   sudo k0s kubectl get jobs -n kuma
   ```

3. If a migration job exists, delete it:

   ```
   sudo k0s kubectl delete job core-transfer -n kuma
   ```

4. Go to the console of a host from the kuma_core group.
5. Start the LogAnalyzer Core services by running the following commands:

   ```
   sudo systemctl start kuma-mongodb
   sudo systemctl start kuma-core-00000000-0000-0000-0000-000000000000
   ```

6. Make sure that the kuma-core-00000000-0000-0000-0000-000000000000 service has been successfully started:

   ```
   sudo systemctl status kuma-core-00000000-0000-0000-0000-000000000000
   ```

7. Make sure that the kuma_core group has access to the KUMA interface by host FQDN.

   Other hosts do not need to be running.

8. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.

9. In the core-transfer-job.yaml.j2 file, find the following lines:

```
cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
```

10. Edit these lines as follows, making sure you keep the indentation (number of space characters):

```
cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/ &&
```

11. Save the changes to the file.

You can then restart the distributed installation using the prepared k0s.inventory.yaml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If the component is not detected on the worker nodes, a clean installation of the LogAnalyzer Core is performed in the cluster without migrating resources to it. Existing components must be manually recreated with the new Core in the LogAnalyzer web interface.

For collectors and storages from the inventory file, certificates for communication with the Core inside the cluster will be reissued. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed are entered into the ConfigMap.

The final stage of preparing LogAnalyzer for work

1. After upgrading LogAnalyzer, clear your browser cache.
2. Manually upgrade the LogAnalyzer agents.

LogAnalyzer is successfully upgraded.

Known limitations

1. Hierarchical structure is not supported in 3.0.2, therefore all LogAnalyzer hosts become standalone hosts when upgrading from version 2.1.3 to 3.0.2.
2. For existing users, after upgrading from 2.1.3 to 3.0.2, the universal dashboard layout is not refreshed.

   Possible solution: restart the Core service (kuma-core.service), and the data will be refreshed with the interval configured for the layout.

Upgrading from version 3.0.x to 3.0.3

To install LogAnalyzer version 3.0.3 over version 3.0.x, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Create a backup copy of the LogAnalyzer Core. If necessary, you will be able to restore data from backup for version 3.0.x.

   LogAnalyzer backups created in versions earlier than 3.0.3 cannot be restored in version 3.0.3. This means that you cannot install LogAnalyzer 3.0.3 and restore a LogAnalyzer 3.0.x backup in it.

   Create a backup copy immediately after upgrading LogAnalyzer to version 3.0.3.

2. Make sure that all application installation requirements are met.

3. During installation or upgrade, make sure that TCP port 7220 on the LogAnalyzer Core is accessible from the LogAnalyzer storage hosts.

Upgrading LogAnalyzer

Depending on the KUMA deployment scheme that you are using, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the application.

- Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high availability configuration.

  If an inventory file is not available for the current version, use the provided inventory file template and edit it as necessary. To view a list of hosts and host roles in the current LogAnalyzer system, in the web interface, go to **Resources** → **Active services** section.

The upgrade process mirrors the installation process.

If you want to upgrade from a distributed installation to a distributed installation in a high availability configuration, you must first upgrade the distributed installation and then **migrate the Core to a Kubernetes cluster**.

To migrate the LogAnalyzer Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the LogAnalyzer Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed LogAnalyzer Core on all hosts on which you want to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

Resolving the LogAnalyzer Core migration error

Migration of the LogAnalyzer Core from a host to a new Kubernetes cluster may be stopped due to a timeout at the `Deploy Core transfer job` step. In this case, the following error message is recorded in the log of core-transfer migration tasks:

```
cp: can't stat '/mnt/kuma-source/core/.lic': No such file or directory
```

To prevent this error, before you start migrating the LogAnalyzer Core:

1. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.
2. In the core-transfer-job.yaml.j2 file, find the following lines:

   ```
   cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

3. Edit these lines as follows, making sure you keep the indentation (number of space characters):

   ```
   cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

4. Save the changes to the file.

You can then restart the distributed installation using the prepared k0s.inventory.yml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If you started migrating the LogAnalyzer Core from a host to a new Kubernetes cluster and the migration failed with an error, follow the steps below to fix the error.

To fix the error after attempting to migrate the LogAnalyzer Core from a host to a new Kubernetes cluster:

1. On any controller of the cluster, delete the Ingress object by running the following command:

   ```
   sudo k0s kubectl delete daemonset/ingress -n ingress
   ```

2. Check if a migration job exists in the cluster:

   ```
   sudo k0s kubectl get jobs -n kuma
   ```

3. If a migration job exists, delete it:

   ```
   sudo k0s kubectl delete job core-transfer -n kuma
   ```

4. Go to the console of a host from the kuma_core group.
5. Start the LogAnalyzer Core services by running the following commands:

   ```
   sudo systemctl start kuma-mongodb
   sudo systemctl start kuma-core-00000000-0000-0000-0000-000000000000
   ```

6. Make sure that the kuma-core-00000000-0000-0000-0000-000000000000 service has been successfully started:

   ```
   sudo systemctl status kuma-core-00000000-0000-0000-0000-000000000000
   ```

7. Make sure that the kuma_core group has access to the KUMA interface by host FQDN.

   Other hosts do not need to be running.

8. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.

9. In the core-transfer-job.yaml.j2 file, find the following lines:

```
cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
```

10. Edit these lines as follows, making sure you keep the indentation (number of space characters):

```
cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/ &&
```

11. Save the changes to the file.

You can then restart the distributed installation using the prepared k0s.inventory.yaml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If the component is not detected on the worker nodes, a clean installation of the LogAnalyzer Core is performed in the cluster without migrating resources to it. Existing components must be manually recreated with the new Core in the LogAnalyzer web interface.

For collectors and storages from the inventory file, certificates for communication with the Core inside the cluster will be reissued. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed are entered into the ConfigMap.

The final stage of preparing LogAnalyzer for work

1. After upgrading LogAnalyzer, clear your browser cache.
2. Manually upgrade the LogAnalyzer agents.

LogAnalyzer is successfully upgraded.

Known limitations

For existing users, after upgrading from 3.0.x to 3.0.3, the universal dashboard layout is not refreshed.

Possible solution: restart the Core service (kuma-core.service), and the data refresh interval configured for the layout will be used.

Upgrading from version 3.0.3 to 3.2.x

To install LogAnalyzer version 3.2.x over version 3.0.3, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Create a backup copy of the LogAnalyzer Core. If necessary, you can restore data from backup for version 3.0.3.

   LogAnalyzer backups created in versions 3.0.3 and earlier cannot be restored in version 3.2.x. This means that you cannot install LogAnalyzer 3.2.x and restore a LogAnalyzer 3.0.3 backup in it.

   Create a backup copy immediately after upgrading LogAnalyzer to version 3.2.x.

2. Make sure that all application installation requirements are met.

3. Make sure that the host name of the LogAnalyzer Core does not start with a numeral. The upgrade to version 3.2.x cannot be completed successfully if the host name of the LogAnalyzer Core starts with a numeral. In such a case, you will need to take certain measures to successfully complete the upgrade. Contact Technical Support for additional instructions.

4. During installation or upgrade, make sure that TCP port 7220 on the LogAnalyzer Core is accessible from the LogAnalyzer storage hosts.

Upgrading LogAnalyzer

Depending on the KUMA deployment scheme that you are using, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the application.

- Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high availability configuration.

  If an inventory file is not available for the current version, use the provided inventory file template and edit it as necessary. To view a list of hosts and host roles in the current LogAnalyzer system, in the web interface, go to **Resources** → **Active services** section.

The upgrade process mirrors the installation process.

If you want to upgrade from a distributed installation to a distributed installation in a high availability configuration, you must first upgrade the distributed installation and then **migrate the Core to a Kubernetes cluster**. For subsequent upgrades, use the k0s.inventory.yml inventory file with need_transfer: false, because the LogAnalyzer Core already has been migrated to the Kubernetes cluster and you do not need to repeat this.

To migrate the LogAnalyzer Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the LogAnalyzer Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed LogAnalyzer Core on all hosts on which you want to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

Resolving the LogAnalyzer Core migration error

Migration of the LogAnalyzer Core from a host to a new Kubernetes cluster may be stopped due to a timeout at the `Deploy Core transfer job` step. In this case, the following error message is recorded in the log of core-transfer migration tasks:

```
cp: can't stat '/mnt/kuma-source/core/.lic': No such file or directory
```

To prevent this error, before you start migrating the LogAnalyzer Core:

1. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.
2. In the core-transfer-job.yaml.j2 file, find the following lines:

   ```
   cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

3. Edit these lines as follows, making sure you keep the indentation (number of space characters):

   ```
   cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

4. Save the changes to the file.

You can then restart the distributed installation using the prepared k0s.inventory.yml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If you started migrating the LogAnalyzer Core from a host to a new Kubernetes cluster and the migration failed with an error, follow the steps below to fix the error.

To fix the error after attempting to migrate the LogAnalyzer Core from a host to a new Kubernetes cluster:

1. On any controller of the cluster, delete the Ingress object by running the following command:

   ```
   sudo k0s kubectl delete daemonset/ingress -n ingress
   ```

2. Check if a migration job exists in the cluster:

   ```
   sudo k0s kubectl get jobs -n kuma
   ```

3. If a migration job exists, delete it:

   ```
   sudo k0s kubectl delete job core-transfer -n kuma
   ```

4. Go to the console of a host from the kuma_core group.
5. Start the LogAnalyzer Core services by running the following commands:

   ```
   sudo systemctl start kuma-mongodb
   sudo systemctl start kuma-core-00000000-0000-0000-0000-000000000000
   ```

6. Make sure that the kuma-core-00000000-0000-0000-0000-000000000000 service has been successfully started:

   ```
   sudo systemctl status kuma-core-00000000-0000-0000-0000-000000000000
   ```

7. Make sure that the kuma_core group has access to the KUMA interface by host FQDN.

   Other hosts do not need to be running.

8. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.

9. In the core-transfer-job.yaml.j2 file, find the following lines:

```
cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
```

10. Edit these lines as follows, making sure you keep the indentation (number of space characters):

```
cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/ &&
```

11. Save the changes to the file.

You can then restart the distributed installation using the prepared k0s.inventory.yaml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If the component is not detected on the worker nodes, a clean installation of the LogAnalyzer Core is performed in the cluster without migrating resources to it. Existing components must be manually recreated with the new Core in the LogAnalyzer web interface.

For collectors and storages from the inventory file, certificates for communication with the Core inside the cluster will be reissued. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed are entered into the ConfigMap.

The final stage of preparing LogAnalyzer for work

1. After upgrading LogAnalyzer, clear your browser cache.
2. If you are using agents, manually upgrade the LogAnalyzer agents.

LogAnalyzer is successfully upgraded.

Known limitations

1. For existing users, after upgrading from 3.0.3 to 3.2.x, the universal dashboard layout is not refreshed.

   Possible solution: restart the Core service (kuma-core.service), and the data refresh interval configured for the layout will be used.

2. If the old Core service, "kuma-core.service" is still displayed after the upgrade, run the following command after installation is complete:

   sudo systemctl reset-failed

   After running the command, the old service is no longer displayed, and the new service starts successfully.

Upgrading from version 3.2.0 or 3.2.1 to 3.4

To install LogAnalyzer version 3.4 over version 3.2.0 or 3.2.1, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Create a backup copy of the LogAnalyzer Core. If necessary, you will be able to restore data from backup for version 3.2.

   LogAnalyzer backups created in versions 3.2 and earlier cannot be restored in version 3.4. This means that you cannot install LogAnalyzer 3.4 and restore a LogAnalyzer 3.2.x backup in it.

   Create a backup copy immediately after upgrading LogAnalyzer to version 3.4.

2. Make sure that all application installation requirements are met.

3. Make sure that the host name of the LogAnalyzer Core does not start with a numeral. The upgrade to version 4.0 cannot be completed successfully if the host name of the LogAnalyzer Core begins with a numeral. If the hostname of the LogAnalyzer Core starts with a numeral, a number of steps are necessary to successfully complete the upgrade. Contact Technical Support for additional instructions.

4. During installation or upgrade, make sure that TCP port 7220 on the LogAnalyzer Core is accessible from the LogAnalyzer storage hosts.

5. For event sources that have a red status in the **Source status** → **List of event sources** section, check out how many there are and if they are up to date; for outdated event sources, do one of the following:

   - Remove the event sources that you no longer need.
   - Edit the settings of monitoring policies that are applied to the outdated event sources to bring them up to date with the current stream of events.
   - For monitoring policies applied to outdated event sources, disable notifications.

   After LogAnalyzer is upgraded, notifications are sent to the configured email addresses for all red-status event sources that have notifications about triggered policies configured.

Upgrading LogAnalyzer

Depending on the KUMA deployment scheme that you are using, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the application.
- Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high availability configuration.

  If an inventory file is not available for the current version, use the provided inventory file template and edit it as necessary. To view a list of hosts and host roles in the current LogAnalyzer system, in the web interface, go to **Resources** → **Active services** section.

The upgrade process mirrors the installation process.

If you want to upgrade from a distributed installation to a distributed installation in a high availability configuration, you must first upgrade the distributed installation and then **migrate the Core to a Kubernetes cluster**. For subsequent upgrades, use the k0s.inventory.yml inventory file with need_transfer: false, because the LogAnalyzer Core already has been migrated to the Kubernetes cluster and you do not need to repeat this.

To migrate the LogAnalyzer Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the LogAnalyzer Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed LogAnalyzer Core on all hosts on which you want to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

Resolving the LogAnalyzer Core migration error

Migration of the LogAnalyzer Core from a host to a new Kubernetes cluster may be stopped due to a timeout at the `Deploy Core transfer job` step. In this case, the following error message is recorded in the log of core-transfer migration tasks:

```
cp: can't stat '/mnt/kuma-source/core/.lic': No such file or directory
```

To prevent this error, before you start migrating the LogAnalyzer Core:

1. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.
2. In the core-transfer-job.yaml.j2 file, find the following lines:

   ```
   cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

3. Edit these lines as follows, making sure you keep the indentation (number of space characters):

   ```
   cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

4. Save the changes to the file.

You can then restart the distributed installation using the prepared k0s.inventory.yml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If you started migrating the LogAnalyzer Core from a host to a new Kubernetes cluster and the migration failed with an error, follow the steps below to fix the error.

To fix the error after attempting to migrate the LogAnalyzer Core from a host to a new Kubernetes cluster:

1. On any controller of the cluster, delete the Ingress object by running the following command:

   ```
   sudo k0s kubectl delete daemonset/ingress -n ingress
   ```

2. Check if a migration job exists in the cluster:

   ```
   sudo k0s kubectl get jobs -n kuma
   ```

3. If a migration job exists, delete it:

   ```
   sudo k0s kubectl delete job core-transfer -n kuma
   ```

4. Go to the console of a host from the kuma_core group.
5. Start the LogAnalyzer Core services by running the following commands:

   ```
   sudo systemctl start kuma-mongodb
   sudo systemctl start kuma-core-00000000-0000-0000-0000-000000000000
   ```

6. Make sure that the kuma-core-00000000-0000-0000-0000-000000000000 service has been successfully started:

   ```
   sudo systemctl status kuma-core-00000000-0000-0000-0000-000000000000
   ```

7. Make sure that the kuma_core group has access to the KUMA interface by host FQDN.

   Other hosts do not need to be running.

8. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.

9. In the core-transfer-job.yaml.j2 file, find the following lines:

```
cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
```

10. Edit these lines as follows, making sure you keep the indentation (number of space characters):

```
cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/ &&
```

11. Save the changes to the file.

You can then restart the distributed installation using the prepared k0s.inventory.yaml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If the component is not detected on the worker nodes, a clean installation of the LogAnalyzer Core is performed in the cluster without migrating resources to it. Existing components must be manually recreated with the new Core in the LogAnalyzer web interface.

For collectors and storages from the inventory file, certificates for communication with the Core inside the cluster will be reissued. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin
- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed are entered into the ConfigMap.

The final stage of preparing LogAnalyzer for work

1. After upgrading LogAnalyzer, clear your browser cache.
2. If you are using agents, manually upgrade the LogAnalyzer agents.

LogAnalyzer is successfully upgraded.

Known limitations

1. For existing users, after upgrading from 3.2.0 or 3.2.1 to 3.4, the universal dashboard layout is not refreshed.

   Possible solution: restart the Core service (kuma-core.service), and the data refresh interval configured for the layout will be used.

2. If the old Core service, "kuma-core.service" is still displayed after the upgrade, run the following command after installation is complete:

   sudo systemctl reset-failed

   After running the command, the old service is no longer displayed, and the new service starts successfully.

Upgrading from version 3.4.x to 4.0

To install LogAnalyzer version 4.0 over version 3.4.x, complete the preliminary steps and then perform the upgrade.

Preliminary steps

1. Create a backup copy of the LogAnalyzer Core. If necessary, you will be able to restore data from backup for version 3.4.x.

   LogAnalyzer backups created in versions 3.4.x and earlier cannot be restored in version 4.0. This means that you cannot install LogAnalyzer 4.0 and restore a LogAnalyzer 3.4.x backup in it.

   Create a backup copy immediately after upgrading LogAnalyzer to version 4.0.

2. Make sure that all application installation requirements are met.

3. Make sure that the host name of the LogAnalyzer Core does not start with a numeral. The upgrade to version 4.0 cannot be completed successfully if the host name of the LogAnalyzer Core begins with a numeral. If the hostname of the LogAnalyzer Core starts with a numeral, a number of steps are necessary to successfully complete the upgrade. Contact Technical Support for additional instructions.

4. During installation or upgrade, make sure that TCP port 7220 on the LogAnalyzer Core is accessible from the LogAnalyzer storage hosts.

5. For event sources that have a red status in the **Source status** → **List of event sources** section, check out how many there are and if they are up to date; for outdated event sources, do one of the following:

   - Remove the event sources that you no longer need.
   - Edit the settings of monitoring policies that are applied to the outdated event sources to bring them up to date with the current stream of events.
   - For monitoring policies applied to outdated event sources, disable notifications.

   After LogAnalyzer is upgraded, notifications are sent to the configured email addresses for all red-status event sources that have notifications about triggered policies configured.

6. For an installation with the LogAnalyzer Core in a Kubernetes cluster, run the script to determine the size of metrics data on the LogAnalyzer Core 3.4 volume; if necessary, set or modify the values of `metrics_transfer_job_timeout`, `metrics_disk_request`, `core_disk_request` parameters in the k0s.inventory.yml inventory file; and then proceed with the upgrade procedure.

Upgrading LogAnalyzer

Depending on the KUMA deployment scheme that you are using, do one the following:

- Use the prepared distributed.inventory.yml inventory file and follow the instructions for distributed installation of the application.
- Use the prepared k0s.inventory.yml inventory file and follow the instructions for distributed installation in a high-availability configuration with the LogAnalyzer Core in a Kubernetes cluster.

  If an inventory file is not available for the current version, use the provided inventory file template and fill in the corresponding settings. To view a list of hosts and host roles in the current LogAnalyzer system, in the web interface, go to **Resources** → **Active services** section.

The upgrade process mirrors the installation process.

You can change the installation type as follows:

- If you want to upgrade from a distributed installation to an installation with the LogAnalyzer Core in a Kubernetes cluster, you must first upgrade the distributed installation to the latest version and then **migrate the Core to a Kubernetes cluster**. For subsequent upgrades, use the k0s.inventory.yml inventory file with `need_transfer: false` because the LogAnalyzer Core already has been migrated to the Kubernetes cluster and you do not need to repeat this.

To migrate the LogAnalyzer Core to a new Kubernetes cluster:

1. Prepare the k0s.inventory.yml inventory file.

   The kuma_core, kuma_ collector, kuma_correlator, kuma_storage sections of your k0s.inventory.yml inventory file must contain the same hosts that were used when KUMA was upgraded from version 2.1.3 to version 3.0.3 and then to version 3.2, or when a new installation was performed. In the inventory file, set deploy_to_k8s: true, need_transfer: true. Set deploy_example_services: false.

2. Follow the steps for distributed installation using your prepared k0s.inventory.yml inventory file.

Migrating the LogAnalyzer Core to a new Kubernetes cluster

When the installer is started with an inventory file, the installer looks for an installed LogAnalyzer Core on all hosts on which you want to deploy worker nodes of the cluster. The found Core will be moved from the host to within the newly created Kubernetes cluster.

Resolving the LogAnalyzer Core migration error

Migration of the LogAnalyzer Core from a host to a new Kubernetes cluster may be stopped due to a timeout at the `Deploy Core transfer job` step. In this case, the following error message is recorded in the log of core-transfer migration tasks:

```
cp: can't stat '/mnt/kuma-source/core/.lic': No such file or directory
```

To prevent this error, before you start migrating the LogAnalyzer Core:

1. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.
2. In the core-transfer-job.yaml.j2 file, find the following lines:

   ```
   cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

3. Edit these lines as follows, making sure you keep the indentation (number of space characters):

   ```
   cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/
   &&
   ```

4. Save the changes to the file.

   You can then restart the distributed installation using the prepared k0s.inventory.yml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If you started migrating the LogAnalyzer Core from a host to a new Kubernetes cluster and the migration failed with an error, follow the steps below to fix the error.

To fix the error after attempting to migrate the LogAnalyzer Core from a host to a new Kubernetes cluster:

1. On any controller of the cluster, delete the Ingress object by running the following command:

   ```
   sudo k0s kubectl delete daemonset/ingress -n ingress
   ```

2. Check if a migration job exists in the cluster:

   ```
   sudo k0s kubectl get jobs -n kuma
   ```

3. If a migration job exists, delete it:

   ```
   sudo k0s kubectl delete job core-transfer -n kuma
   ```

4. Go to the console of a host from the kuma_core group.
5. Start the LogAnalyzer Core services by running the following commands:

   ```
   sudo systemctl start kuma-mongodb
   sudo systemctl start kuma-core-00000000-0000-0000-0000-000000000000
   ```

6. Make sure that the kuma-core-00000000-0000-0000-0000-000000000000 service has been successfully started:

   ```
   sudo systemctl status kuma-core-00000000-0000-0000-0000-000000000000
   ```

7. Make sure that the kuma_core group has access to the KUMA interface by host FQDN.

   Other hosts do not need to be running.

8. Go to the directory with the extracted installer and open the roles/k0s_prepare/templates/core-transfer-job.yaml.j2 file for editing.

9. In the core-transfer-job.yaml.j2 file, find the following lines:

   ```
   cp /mnt/kuma-source/core/.lic {{ core_k0s_home }}/ &&
   cp /mnt/kuma-source/core/.tenantsEPS {{ core_k0s_home }}/ &&
   ```

10. Edit these lines as follows, making sure you keep the indentation (number of space characters):

    ```
    cp /mnt/kuma-source/core/{{ core_uid }}/.lic {{ core_k0s_home }}/ &&
    cp /mnt/kuma-source/core/{{ core_uid }}/.tenantsEPS {{ core_k0s_home }}/
    &&
    ```

11. Save the changes to the file.

    You can then restart the distributed installation using the prepared k0s.inventory.yaml inventory file. The migration of the LogAnalyzer Core from a host to a new Kubernetes cluster will succeed.

If the component is not detected on the worker nodes, a clean installation of the LogAnalyzer Core is performed in the cluster without migrating resources to it. Existing components must be manually recreated with the new Core in the LogAnalyzer web interface.

For collectors and storages from the inventory file, certificates for communication with the Core inside the cluster will be reissued. This does not change the Core URL for components.

On the Core host, the installer does the following:

- Removes the following systemd services from the host: kuma-core, kuma-mongodb, kuma-victoria-metrics, kuma-vmalert, and kuma-grafana.
- Deletes the internal certificate of the Core.
- Deletes the certificate files of all other components and deletes their records from MongoDB.
- Deletes the following directories:
  - /opt/kaspersky/kuma/core/bin
  - /opt/kaspersky/kuma/core/certificates
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/log
  - /opt/kaspersky/kuma/core/<KUMA Core service ID>/logs
  - /opt/kaspersky/kuma/metrics/<Metrics service ID>/bin

- Migrates data from the Core and its dependencies to a network drive within the Kubernetes cluster.
- On the Core host, it moves the following directories:
  - /opt/kaspersky/kuma/core → /opt/kaspersky/kuma/core.moved
  - /opt/kaspersky/kuma/metrics → /opt/kaspersky/kuma/metrics.moved

  After you have verified that the Core was correctly migrated to the cluster, you can delete these directories.

  If you encounter problems with the migration, check the logs for records of the 'core-transfer' migration task in the 'kuma' namespace in the cluster (this task is available for 1 hour after the migration).

  If you need to perform migration again, you must restore the original names of the /opt/kaspersky/kuma/*.moved directories.

  If the /etc/hosts file on the Core host contained lines that were not related to addresses in the 127.X.X.X range, the contents of the /etc/hosts file from the Core host is entered into the coredns ConfigMap when the Core is migrated to the Kubernetes cluster. If the Core is not migrated, the contents of the /etc/hosts file from the host where the primary controller is deployed are entered into the ConfigMap.

- If you want to upgrade LogAnalyzer in a distributed installation to the latest version of LogAnalyzer with the Core in a Raft cluster, you must manually add a `kuma_core_peers` group to the distributed.inventory.yml inventory file and in this group, specify an even number of servers on which you plan to install additional LogAnalyzer Core services. Also, in the `kuma_core` and `kuma_core_peers` groups, you can specify the `raft_node_addr` setting in the `<FQDN of the server from the kuma_core group on which the KUMA Core is installed>:<port>` format, if the server has multiple FQDNs. The FQDN specified in the `raft_node_addr` setting will be used to listen for signals from other nodes.

  After starting the installer with the prepared inventory file, none of the additional LogAnalyzer Core services are automatically installed on the servers. The installer only prepares the servers specified in the inventory file for the installation of services. After the installer finishes, only one LogAnalyzer Core service is automatically installed on the server specified in the `kuma_core` group. After the installer finishes, you need to manually create the LogAnalyzer Core services in the web interface and install the LogAnalyzer Core services with unique IDs one by one on each server specified in the `kuma_core_peers` group.

  After the upgrade, all existing LogAnalyzer services are connected to the LogAnalyzer Core on the server from the `kuma_core` group. Subsequently, when installing new services, you can specify multiple comma-separated URLs in the installation command to connect services to LogAnalyzer Cores in the Raft cluster.

The final stage of preparing LogAnalyzer for work

1. After upgrading LogAnalyzer, clear your browser cache.
2. If you are using agents, manually upgrade the LogAnalyzer agents.
3. Migrate alert events from MongoDB to ClickHouse.
4. Specify the storage for new alert events in alert filling settings.

LogAnalyzer is successfully upgraded.

Known limitations

1. For existing users, upgrading from 3.4.x to 4.0 does not update the universal dashboard.

   Possible solution: restart the Core service (kuma-core.service) to apply the data refresh interval configured for the dashboard.

2. If the old Core service, "kuma-core.service" is still displayed after the upgrade, run the following command after installation is complete:

   ```
   sudo systemctl reset-failed
   ```

   After running the command, the old service is no longer displayed, and the new service starts successfully.

Troubleshooting update errors

When updating LogAnalyzer, you may encounter the following errors:

- **Timeout error**

  When upgrading from version 2.0.x on systems that contain large amounts of data and are operating with limited resources, the system may return the Wrong admin password error message after you enter the administrator password. If you specify the correct password, LogAnalyzer may still return an error because LogAnalyzer is unable to start the Core service due to resource limit and a timeout error. If you enter the administrator password three times without waiting for the installation to complete, the update may end with a fatal error.

  Follow these steps to resolve the timeout error and successfully complete the update:

  1. Open a separate second terminal and run the following command to verify that the command output contains the timeout error line:

     ```
     journalctl -u kuma-core | grep 'start operation timed out'
     ```

     Timeout error message:

     kuma-core.service: start operation timed out. Terminating.

  2. After you find the timeout error message, in the /usr/lib/systemd/system/kuma-core.service file, change the value of the `TimeoutSec` parameter from 300 to 0 to remove the timeout limit and temporarily prevent the error from recurring.

  3. After modifying the service file, run the following commands in sequence:

     ```
     systemctl daemon-reload
     service kuma-core restart
     ```

  4. After running the commands and successfully starting the service in the second terminal, enter the administrator password again in your first terminal where the installer is prompting you for the password.

     LogAnalyzer continues the installation. In resource-limited environments, installation may take up to an hour.

  5. After installation finishes successfully, in the /usr/lib/systemd/system/kuma-core.service file, set the `TimeoutSec` parameter back to 300.

  6. After modifying the service file, run the following commands in the second terminal:

     ```
     systemctl daemon-reload
     service kuma-core restart
     ```

  After you run these commands, the update will be succeed.

- **Invalid administrator password**

The admin user password is required to automatically populate the storage settings during an update. If you entered an incorrect admin user password nine times during the TASK [Prompt for admin password], the installer still performs the update, and the web interface is still available. However, the storage settings are not migrated, and the storages will show a red status.

To fix the error and make the repositories available for use, update the storage settings:

1. Go to the storage settings, manually fill in the fields of the ClickHouse cluster, and click **Save**.

2. Restart the storage service.

The storage service starts with the specified settings, and its status is green.

- DB::Exception error

  After upgrading LogAnalyzer, the storage may have the red status, and its logs may contain errors about suspicious strings.

  Example error:

  ```
  DB::Exception::Exception(std::__1::basic_string<char, std::__1::char_traits<char>,
  std::__1::allocator<char>> const&, int, bool) @ 0xda0553a in
  /opt/kaspersky/kuma/clickhouse/bin/clickhouse
  ```

  To restart ClickHouse, carry out the following command on the LogAnalyzer storage server:

  ```
  touch /opt/kaspersky/kuma/clickhouse/data/flags/force_restore_data && systemctl
  restart kuma-storage-<ID of the storage that encountered the error>
  ```

- Expiration of k0s cluster certificates

  ### Symptoms

  Controllers or worker nodes cannot connect; pods cannot be moved from one worker node to another.

  Logs of the k0scontroller and k0sworker services contain multiple records with the following substring:

  ```
  x509: certificate has expired or is not yet valid
  ```

  ### Cause

  Cluster service certificates are valid for 1 year from the time of creation. The k0s cluster used in the high-availability LogAnalyzer installation automatically rotates all the service certificates it needs, but the rotation is performed only at startup of the k0scontroller service. If k0scontroller services on cluster controllers run without a restart for more than 1 year, service certificates become invalid.

  ### How to fix

  To fix the error, restart the k0scontroller services one by one as root on each controller of the cluster. This reissues the certificates:

  ```
  systemctl restart k0scontroller
  ```

  To check the expiration dates of certificates on controllers, run the following commands as root:

  ```
  find /var/lib/k0s/pki/ -type f -name "*.crt" -print|egrep -v 'ca.crt$'|xargs -L 1 -
  t -i bash -c 'openssl x509 -noout -text -in {}|grep After'
  ```

  ```
  find /var/lib/k0s/pki/etcd -type f -name "*.crt" -print|egrep -v 'ca.crt$'|xargs -L
  1 -t -i bash -c 'openssl x509 -noout -text -in {}|grep After'
  ```

  You can find the names of certificate files and their expiration dates in the output of these commands.

Fix the errors to successfully complete the update.

Uninstalling LogAnalyzer

To uninstall LogAnalyzer, use the Ansible tool and the inventory file that you have prepared.

To uninstall LogAnalyzer:

1. On the control machine, go to the installer directory:

   ```
   cd kuma-ansible-installer
   ```

2. Execute the following command:

   ```
   sudo ./uninstall.sh <inventory file>
   ```

   LogAnalyzer is uninstalled and all of its data is deleted from the server.

The databases that were used by LogAnalyzer (for example, the ClickHouse storage database) and the data therein must be deleted separately.

Special considerations for removing a high-availability configuration of LogAnalyzer

Which components need to be removed depends on the value of the `deploy_to_k8s` parameter in the inventory file that you are using to remove LogAnalyzer:

- If the setting is `true`, the Kubernetes cluster created during the LogAnalyzer installation is deleted.
- If the setting is `false`, all LogAnalyzer components except for the Core are removed from the Kubernetes cluster. The cluster is not deleted.

In addition to the LogAnalyzer components installed outside the cluster, the following directories and files are deleted on the cluster nodes:

- /usr/bin/k0s
- /etc/k0s/
- /var/lib/k0s/
- /usr/libexec/k0s/
- ~/k0s/ (for the ansible_user)
- /opt/longhorn/
- /opt/cni/
- /opt/containerd

When a cluster is being deleted, error messages may appear, however, it does not interrupt the installer.

- You can ignore such messages for the **Delete KUMA transfer job** and **Delete KUMA pod** tasks.
- For the **Reset k0s** task (if an error message contains the following text: "`To ensure a full reset, a node reboot is recommended.`") and the Delete k0s Directories and files task (if an error message contains the following text: "`I/O error: '/var/lib/k0s/kubelet/plugins/kubernetes.io/csi/driver.longhorn.io/`"), we recommend restarting the relevant host and trying to remove LogAnalyzer again with the same inventory file.

After removing LogAnalyzer, restart the hosts on which the LogAnalyzer or Kubernetes components were installed.

Working with tenants

Access to tenants is regulated in the settings of users. The *general administrator* has access to the data of all tenants. Only a user with this role can create and delete tenants.

Tenants are displayed in the table in the **Settings** → **Access** → **Tenants** section of the LogAnalyzer web interface. You can sort the table by clicking on columns.

Available columns:

- **Name** is the name of the tenant.
- **EPS limit** is the EPS (events per second) quota allocated to the tenant out of the overall EPS quota determined by the license.
- **Description** is the description of the tenant.
- **Created**is the tenant creation date.
- **Updated** is the date and time when the tenant information was most recently modified.

To create a tenant:

1. In the LogAnalyzer web interface, in the **Settings** → **Access** → **Tenants**, click **Add**.

   The **Add tenant** window opens.

2. Specify the tenant name in the **Name** field. The name must contain 1 to 128 Unicode characters.
3. In the **EPS limit** field, specify the EPS quota for the tenant. This field is optional. The sum total of EPS quotas for all tenants must not exceed the EPS available under the license. However, LogAnalyzer does not check the values entered in this field.
4. If necessary, add a **Description** of the tenant. The description can contain no more than 256 Unicode characters.
5. Click **Create**.

The tenant is added. Press F5 to refresh the page. After refreshing the page, the created tenant is displayed in the web interface.

To delete a tenant:

1. In the **Settings** → **Access** → **Tenants** section of the LogAnalyzer web interface, select the relevant tenant by selecting the check box next to it, then in the toolbar, select **Delete**.

2. This opens the **Delete tenant?** window, which displays information about the tenant and prompts you to enter a code and confirm the deletion of the tenant. If you want to proceed with deleting the tenant, enter the code.

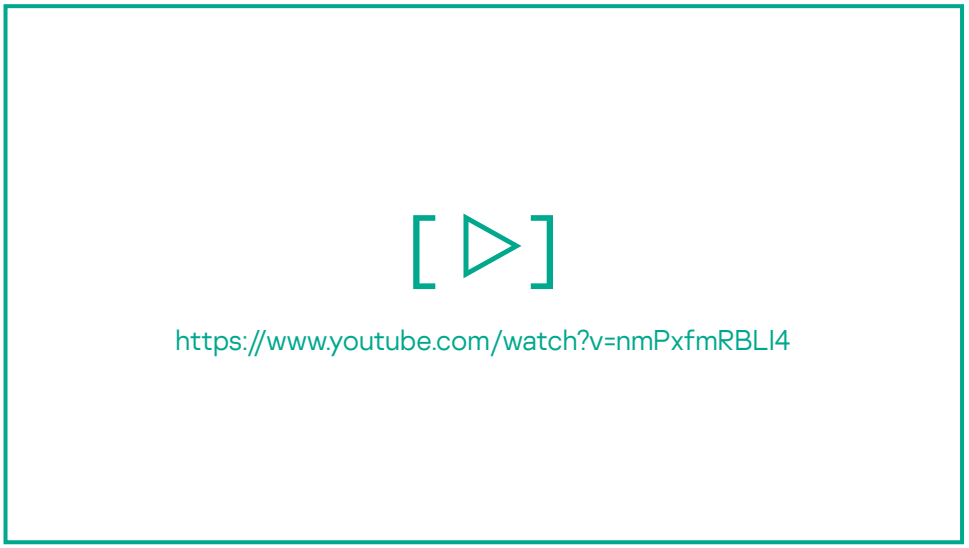
3. Click **OK**.

The tenant is deleted.

When you delete a tenant, the following happens:

- Services belonging to the tenant, with the exception of agents, are automatically stopped. In the section **Resources** → **Active services**, such services are displayed with a gray status.
- Reception and processing of events is terminated, and previously received events are deleted.
- All tenant resources are deleted.
- User accounts that have access only to the remote tenant are disabled—active sessions are terminated, and an error message is displayed when attempting to log in again.
- In the **Settings** → **Access** → **Domain authorization** section, under **Administration groups**, the role filters created for this tenant are deleted.

Agent services are not stopped automatically. If necessary, you can stop the Windows agent services manually in the **Start** → **Services** section; to stop the Linux agent services, you can press **`Ctrl`**+**`C`** in the terminal in which the agent was started.



https://www.youtube.com/watch?v=nmPxfmRBLI4

Selecting a tenant

If you have access to multiple tenants, LogAnalyzer lets you select the tenants for which you want data to be displayed in the LogAnalyzer web interface.

To select a tenant for displaying data:

1. In the LogAnalyzer web interface, click **Selected tenants**.

   The tenant selection area opens.

2. Select the check boxes next to the tenants whose data you want to see in sections of the LogAnalyzer web interface.

3. You must select at least one tenant. You can use the **Search** field to search for tenants.

4. Click the tenant selection area by clicking **Selected tenants**.

   Sections of the LogAnalyzer web interface will display only the data and analytics related to the selected tenants.

Your selection of tenants for data display will determine which tenants can be specified when creating resources, services, dashboards, report templates, widgets, and other LogAnalyzer settings that let you select a tenant.

Tenant affiliation rules

It is important to track which tenant owns specific objects created in LogAnalyzer because this determines who will have access to the objects and whether or not interaction with specific objects can be configured.

The tenant of an object (such as a service or resource) is determined by the user when the object is created. After the object is created, the tenant selected for that object cannot be changed. However, resources can be exported then imported into another tenant.

> Services that must be accommodated by the capacities of the main tenant can be deployed only by a user with the general administrator role.

Managing users

Multiple users can have access to LogAnalyzer. Users are assigned user roles, which affect the tasks the users can perform. The same user may have different roles with different tenants. However, you cannot assign roles to yourself, even if your user account has the General administrator role; the displayed list of roles cannot be edited.

You can create or edit user accounts in the **Settings** → **Users** section of the LogAnalyzer web interface. Users are also created automatically in the application if LogAnalyzer integration with Active Directory is enabled and the user is logging in to the LogAnalyzer web interface for the first time using their domain account.

> All internal users whose accounts were created in the LogAnalyzer web interface, except for the admin account, must change their password when they first log into the LogAnalyzer web interface. Subsequently, these users must regularly change their password in accordance with corporate policies, but at least once every three months.

User accounts are displayed in the **Users** window of the LogAnalyzer web interface on the **Active users** and **Inactive users** tabs, depending on the user account status, *Active* or *Inactive*. You can use the **Search** field to look for users. You can sort the tables by clicking the column header and selecting **Ascending** or **Descending**.

You can create, edit, or disable user accounts. When editing user accounts (your own or the accounts of others), you can generate an API token for them.

To disable a user account:

In the **Settings** → **Access** → **Users** → **Active users** section of the LogAnalyzer web interface, select the check box next to the relevant user and click **Disable**.

User roles

Functionality related to the integration of Kaspersky CyberTrace and Kaspersky Threat Lookup is available only in the commercial version of LogAnalyzer.

LogAnalyzer users may have the following roles:

- Users with the *General administrator* are responsible for managing the core systems of LogAnalyzer. For example, they install system components, perform maintenance, work with services, create backups, and add users to the system. These users have full access to LogAnalyzer.
- Users with the *Tenant administrator* are responsible for the functioning of LogAnalyzer systems belonging to specific tenants.
- Users with the *Tier 2 analyst* role are responsible for configuring the LogAnalyzer system to receive and process events of a specific tenant.
- Users with the *Tier 1 analyst* role are responsible for configuring the LogAnalyzer system to receive and process events of a specific tenant. Users with this role have fewer privileges than Tier 2 analysts.
- *Junior analyst*—this role is for users dealing with immediate security threats of a specific tenant. A user with this role can check resources of the Shared tenant through the REST API.
- *Read shared resources*—this role is intended for managing the Shared tenant. Users with this role have read access to shared resources.
- *Manage shared resources*—this role is intended for managing the Shared tenant. Users with this role have edit access to shared resources.

User permissions for the Reports section

The Reports section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
Create report template	yes	yes	yes	yes	no	no	no
View and edit templates and reports	yes	yes	yes	yes	no	no	no
Generate reports	yes	yes	yes	yes	no	no	no

View generated reports	yes	yes	yes	yes	yes	no	no
Modify generated reports	yes	yes	yes	yes	no	no	no
Export generated reports	yes	yes	yes	yes	yes	no	no
Delete templates and generated reports	yes	yes	yes	yes	no	no	no
Manage report generation settings	yes	yes	yes	only the settings of own reports	no	no	no
Duplicate report template	yes	yes	yes	yes	no	no	no
Open the generated report by email	yes	yes	yes	yes	yes	no	no

User permissions for the Dashboard section

The Dashboard section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View data on the dashboard and change dashboards	yes	yes	yes	yes	yes	no	no
View universal dashboard	yes	yes	yes	yes	yes	no	no
Add dashboards	yes	yes	yes	yes	no	no	no
Edit and rename dashboards	yes	yes	yes	yes	no	no	no
Delete dashboards	yes	yes	yes	yes	no	no	no
Enable and disable the TV mode	yes	yes	yes	yes	yes	no	no

User permissions for the Resources section

The Resources → Services & Resources → Services → Active services section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View the list of active services	yes	yes	yes	yes	yes	no	no
View and delete partitions in storage	yes	no	no	no	no	no	no
Download service logs	yes	yes	yes	yes	yes	no	no
Copy the service ID	yes	yes	yes	yes	yes	no	no
Refresh the table of active services	yes	yes	yes	yes	yes	no	no
Go to Events	yes	yes	yes	yes	yes	no	no
Create a set of resources for services	yes	yes	yes	no	no	no	no
Create a service under Resources → Services → Active services	yes	yes	no	no	no	no	no
Delete services	yes	yes	no	no	no	no	no
View services that have the gray status	yes	no	no	no	no	no	no
Restart services	yes	yes	no	no	no	no	no

Update the settings of services	yes	yes	yes	no	no	no	no
Reset certificates	yes	yes	no	no	no	no	no

User permissions for the Resources configuration and Services configuration sections

The Resources → Resources configuration and Configuring services and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View the list of resource and service configurations	yes	yes	yes	yes	no	yes	yes
Create a service configuration	yes	yes	yes	yes	no	no	no
Create a resource configuration	yes	yes	yes	yes	no	yes	no
Duplicate the configuration of a resource/service	yes	yes	yes	yes	no	yes	no
Edit the configuration of a resource/service	yes	yes	yes	own resource only	no	yes	no
Delete the configuration of a resource/service	yes	yes	yes	yes	no	yes	no
Import the configuration of a resource/service from a file	yes	yes	yes	no	no	yes	no
View the repository, import the resources from the repository	yes	yes	yes	no	no	no	no
Export the configuration of a resource/service	yes	yes	yes	yes	no	yes	yes
Create and edit a resource configuration in the Shared tenant	yes	no	no	no	no	yes	no
View a resource in the Shared tenant (including linking, export, duplication)	yes	no	no	no	no	yes	yes

User permissions for the List of event sources section

The Source status → List of event sources section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View sources of events	yes	yes	yes	yes	yes	no	no
Change sources of events	yes	yes	yes	no	no	no	no
Delete sources of events	yes	yes	yes	no	no	no	no

User permissions for the Monitoring policies section

The Source status → Monitoring policies section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View monitoring policies	yes	yes	yes	yes	yes	yes	yes
Create monitoring policies	yes	yes	yes	no	no	yes	no

Edit monitoring policies	yes	yes	yes	no	no	yes	no
Delete monitoring policies	yes	yes	yes	no	no	yes	no

User permissions for the Events section

The Events section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View the list of events	yes	yes	yes	yes	yes	no	no
Search events	yes	yes	yes	yes	yes	no	no
Open the details of events	yes	yes	yes	yes	yes	no	no
Open statistics	yes	yes	yes	yes	yes	no	no
Export events to a TSV file	yes	yes	yes	yes	yes	no	no
View filters (search query)	yes	yes	yes	yes	yes	yes	yes
Add (save) filter	yes	yes	yes	yes	yes	yes	no
Edit or delete someone else's filter folders	yes	yes	no	no	no	no	no
Delete own filters	yes	yes	yes	yes	yes	yes	no
Delete other users' filters	yes	yes	no	no	no	no	no
Run ktl enrichment (available only in the commercial version of LogAnalyzer)	yes	yes	yes	yes	no	no	no
Add an indicator to Internal TI CyberTrace (available only in the commercial version of LogAnalyzer)	yes	yes	yes	yes	no	no	no
Create presets	yes	yes	yes	yes	yes	yes	no
Delete presets	yes	yes	yes	yes	yes	yes	no
View and use presets	yes	yes	yes	yes	yes	yes	yes

User permissions for the Users and actions section

The Settings → Access → Users section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View the list of users	yes	no	no	no	no	no	no
Add a user	yes	no	no	no	no	no	no
Edit a user	yes	no	no	no	no	no	no
Generate token	yes	yes	yes	yes	yes	yes	yes
Change access rights for a token	yes	yes	yes	yes	yes	yes	yes
View the data of their own profile	yes	yes	yes	yes	yes	yes	yes
Edit the data of their own profile	yes	yes	yes	yes	yes	yes	yes

User permissions for the Tenants section

The Settings → Access → Tenants section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View the list of tenants	yes	no	no	no	no	no	no
Add tenants	yes	no	no	no	no	no	no
Change tenants	yes	no	no	no	no	no	no
Export tenants	yes	no	no	no	no	no	no

This section is available only to the general administrator.

User permissions for the Domain authorization section

The Settings → Access → Domain authorization section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View the Active Directory connection settings	yes	no	no	no	no	no	no
Edit the Active Directory connection settings	yes	no	no	no	no	no	no
Add filters based on roles for tenants	yes	no	no	no	no	no	no

This section is available only to the general administrator.

User permissions for the General section

The Settings → Other → General section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View the SMTP connection settings	yes	no	no	no	no	no	no
Edit the SMTP connection settings	yes	no	no	no	no	no	no

This section is available only to the general administrator.

User permissions for the License section

The Settings → Other → License section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View the list of added license keys	yes	no	no	no	no	no	no
Add license keys	yes	no	no	no	no	no	no
Delete license keys	yes	no	no	no	no	no	no

This section is available only to the general administrator.

User permissions for the Kaspersky CyberTrace section (available only in the commercial version of LogAnalyzer)

The Settings → Integrations → Kaspersky CyberTrace section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View the CyberTrace integration settings	yes	no	no	no	no	no	no
Edit the CyberTrace integration settings	yes	no	no	no	no	no	no

This section is available only to the general administrator.

User permissions for the Kaspersky Threat Lookup section (available only in the commercial version of LogAnalyzer)

The Settings → Integrations → Kaspersky Threat Lookup section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View the Threat Lookup integration settings	yes	no	no	no	no	no	no
Edit the Threat Lookup integration settings	yes	no	no	no	no	no	no

This section is available only to the general administrator.

User permissions for the Repository update section

The Settings → Other → Repository update section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View the parameters	yes	yes	yes	no	no	no	no
Edit the parameters	yes	no	no	no	no	no	no
Start the repository update task manually	yes	yes	yes	no	no	no	no

User permissions for the Access to spaces section

The Settings → Access → Access to spaces section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
Viewing the list of sets	yes	no	no	no	no	no	no
Adding, editing, deleting a space set	yes	no	no	no	no	no	no
Setting the default space set	yes	no	no	no	no	no	no
Assigning a space set to a user	yes	no	no	no	no	no	no

User permissions for the Tags section

The Settings → Other → Tags section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View the list of tags	yes	yes	yes	yes	no	no	no

User permissions for the Metrics section

The Metrics section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
Open metrics	yes	no	no	no	no	no	no

User permissions for the Task manager section

The Task manager section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
View a list of your own tasks	yes	yes	yes	yes	yes	no	no
Finish your own tasks	yes	yes	yes	yes	yes	no	no
Restart your own tasks	yes	yes	yes	yes	yes	no	no
View a list of all tasks	yes	yes	yes	yes	yes	no	no
Finish any task	yes	yes	no	no	no	no	no
Restart any task	yes	yes	yes	yes	yes	no	no

User permissions for the CyberTrace section (available only in the commercial version of LogAnalyzer)

The CyberTrace section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
Open the section	yes	no	no	no	no	no	no

This section is not displayed in the web interface unless CyberTrace integration is configured under **Settings** → **Integrations** → **Kaspersky CyberTrace**.

User permissions for the Access to tenant data section

The Access to tenant data section and actions in this section	General administrator	Tenant administrator	Tier 2 analyst	Tier 1 analyst	Junior analyst	Manage shared resources	Read shared resources
Access to tenants	yes	yes	yes	yes	yes	no	no
Shared tenant	yes	yes	yes	yes	yes	yes	yes
Main tenant	yes	yes	yes	yes	yes	no	no

Configuring the security policy for logging in to the LogAnalyzer web interface

To reduce the risk of unauthorized access to the LogAnalyzer web interface, you can configure rules for users logging in to the LogAnalyzer web interface. For example, configuring a security policy helps prevent an attacker from brute-forcing a password to log in to the LogAnalyzer web interface.

The security policy applies only to user accounts created in LogAnalyzer and does not affect domain accounts imported into LogAnalyzer when integrating with domain controllers. At the same time, the security policy affects the admin user account, which is used for the first login to the LogAnalyzer web interface, in a limited way. The only security policy setting that applies to this user account is the forced re-authentication requirement after a period of inactivity.

The security policy applies to all tenants and cannot be configured on a per-tenant basis.

When upgrading from a previous version of LogAnalyzer, the password requirements apply only to new passwords. Existing passwords are affected only by the **Password lifetime (days)**, other requirements do not apply to such passwords. For users whose passwords have not been changed for more than 90 days, LogAnalyzer suggests a password change after the upgrade.

> Managing security policy settings requires the General administrator role.

To configure the security policy for logging in to the LogAnalyzer web interface

1. Open the **Settings** → **Access** → **Security policy settings** section of the LogAnalyzer web interface.

2. Specify the following security settings:

   - **Maximum number of failed authentications**. If this value is exceeded, the user account is locked out for a duration specified in the **Blockage period after failed authentication attempts (mins)** setting. Enter a number from 0 to 10. The default setting is 0, meaning that this setting is disabled.

   - **Maximum number of authentication attempts per second per Core**. This setting helps prevent brute-forcing a password to log in to the LogAnalyzer web interface. Enter a number from 1 to 8. The default value is 3. If LogAnalyzer detects that the specified value is exceeded, further authentication attempts are blocked.

     Note that the value is specified per LogAnalyzer Core. If your LogAnalyzer instance has more than one LogAnalyzer Core deployed, the actual number of allowed authentication attempts per second is calculated as the specified value multiplied by the number of LogAnalyzer Cores.

   - **Blockage period after failed authentication attempts (mins)**. The duration for which the user account is locked out when the value of the **Maximum number of failed authentications** setting is exceeded. Enter a number from 0 to 1440. The default value is 15. If 0, the account status is changed to *Inactive*. A user with this status cannot log in to the LogAnalyzer web interface. The General administrator can change the user account status later. If the General administrator changes the status of a user account to *Active*, the user account is unlocked and the counter of failed login attempts is reset to 0.

   - **Prevent users from signing in after the password expiration**. This setting allows you to configure the behavior of the application after the expiration of the **Password lifetime (days)**.

     If this setting is enabled, the user is locked out of the LogAnalyzer web interface. The user account status changes to *Inactive*. A user with this status cannot log in to the LogAnalyzer web interface. The General administrator can change the user account status later. The user is locked out if the user has the **User must change password at next login** setting enabled.

     If this setting is disabled, the user with an expired password can log in to the LogAnalyzer web interface. LogAnalyzer then automatically prompts the user to change the password.

   - **Re-authentication is required after the user is inactive for (mins)**. If no mouse movement or key presses are registered for the specified time during a user's LogAnalyzer web interface session, LogAnalyzer requires the user to authenticate again. Enter a number from 1 to 1440. The default value is 15.

3. Specify the following user password settings:

- **Password lifetime (days)**. After the password lifetime expires, LogAnalyzer locks out or does not lock out the user account, depending on the **Prevent users from signing in after the password expiration** setting. Enter a number from 1 to 365. The default value is 90.

- **Minimum password length (symbols)**. Specify a number from 16 to 128. The default value is 16.

- **Time to notify the user before the password expiration (days)**. When the specified number of days remains before the password expires, the user is prompted in the LogAnalyzer web interface to change the password. Enter a number from 1 to 14. The default value is 14.

- **Character categories a password must include**. You can use this setting to configure requirements that apply to manually entered and automatically generated passwords:

  - The password must contain lower-case and upper-case letters of the Latin alphabet, numerals, and at least one of the special characters that you specified.

  - **Ignore repeated characters in the password**. If this check box is selected, two or more identical characters in a row are allowed in passwords.

4. Click **Save**.

Security policy is configured.

Creating a user

To create a user account:

1. In the LogAnalyzer web interface, open **Settings** → **Access** → **Users**.

2. Click the **Create new** button and set the parameters as described below.

   - Selecting the **Active** check box makes the user account active. The toggle switch is turned on by default.

     Only users with an *Active* account can log in to the LogAnalyzer web interface. The status of a user account can also change to *Inactive* automatically, for example, if the maximum allowed number of attempts to log in to the web interface of the application is exceeded and the **Blockage period after failed authentication attempts (mins)** is set to 0.

   - In the **Name** field, enter the user name. The length of the string must be 1 to 128 Unicode characters.

   - In the **Login** field, enter a unique user name for the user account. Must contain from 3 to 64 characters (only a–z, A–Z, 0–9, . \ - _).

   - In the **Email** field, enter the unique email address of the user. Must be a valid email address.

   - Select the **Receive email notifications** check box if you want the user to receive SMTP notifications from LogAnalyzer.

   - Select the **Notify about alerts in web interface** check box to display alert notifications in the LogAnalyzer web interface. Clear this check box if you want to disable notification display. This check box is selected by default.

   - Select the **Display non-printable characters** check box to display non-printing characters such as spaces, tab characters, and line breaks in the LogAnalyzer web interface. If the **Display non-printable characters** check box is selected, you can press **`Ctrl`**/**`Command`**+**`*`** to enable or disable the display of non-printing characters.

     Spaces and tab characters are displayed in all input fields (except the **Description** field), in normalizers, filters, and connector resources, and in SQL queries for searching events in the **Events** section. Spaces are displayed as dots. A tab character is displayed as a dash in normalizers, filters and connectors. In other fields, a tab character is displayed as one or two dots.

     Line break characters are displayed in all input fields that support multi-line input, such as the event search field.

   - In the **New password** field, enter a new password. You can enter a password or generate one by clicking the **Generate password** button. The password must satisfy the following requirements:

     - The length must satisfy the security policy.
     - At least one lowercase character.
     - At least one uppercase character.
     - At lease one numeral.
     - At least one special character from the list of special characters in the security policy.

- No more than two identical characters in a row.
- The password does not contain the login.

The user must change the password when logging in to the LogAnalyzer web interface for the first time and subsequently change the password at least once every three months. The General administrator can manage both of these settings in the security policy.

- In the **Confirm password** field, enter the password again for confirmation.
- If the **User must change password at next login** check box is selected, the next time the user attempts to log in to the LogAnalyzer web interface, the user must specify the old password and set a new password. After that, the user is allowed to log in to the LogAnalyzer web interface with the new password. By default, this check box is selected.

  This setting is available only for *Active* accounts.

  You can use this setting, for example, when creating a new user. You can assign a temporary password to the user and select this check box. After the user logs in with a temporary password, LogAnalyzer prompts the user to change the password.

  You can also make users reset their passwords in the **Settings** → **Access** → **Users** section by selecting users from the list and clicking the **Reset password at next login** button.
- Under **Access**, use the **Add role groups** buttons to assign roles to the user in specific tenants. A user can be assigned different roles in different tenants; multiple roles can be assigned within the same tenant.
- In the **Space sets** drop-down list, select check boxes next to one or more space sets to which you want to grant access. If the user does not have a space set selected, access is granted to the default space set.
- If the **Unlimited password lifetime** check box is selected, the **Password lifetime** setting does not apply to the user account.

  We recommend enabling this setting for a service account, restricting the rights of such an account, and setting a password of maximum complexity. Typically, a user account with an unlimited password lifetime is used for monitoring. When migrating users, you can manually enable this setting for those service accounts that were previously used to display the dashboard.

3. If necessary, use the **Generate token** button to generate an API token.

   Clicking this button displays the token creation window.

4. If necessary, configure the operations available to the user via the REST API by using the **API access rights** button.

5. Click **Save**.

The user account will be created and displayed in the **Users** table.

Editing a user account

To edit a user account:

1. In the LogAnalyzer web interface, open **Settings** → **Access** → **Users**.

2. Select the relevant user and in the user details area on the right, edit the settings as necessary.

   If you want to change your own user account, you can also click on the name of the user account in the lower-left corner of the window to open a menu and click the **Profile** button in the menu.

   - Selecting the **Active** check box makes the user account active. The toggle switch is turned on by default.

     Only users with an *Active* account can log in to the LogAnalyzer web interface. The status of a user account can also change to *Inactive* automatically, for example, if the maximum allowed number of attempts to log in to the web interface of the application is exceeded and the **Blockage period after failed authentication attempts (mins)** is set to 0.

   - In the **Name** field, edit the user name. The length of the string must be 1 to 128 Unicode characters.

   - In the **Login** field, enter a unique user name for the user account. Must contain from 3 to 64 characters (only a–z, A–Z, 0–9, . \ - _).

   - In the **Email** field, edit the unique email address of the user. The email address must be valid and unique.

   - Select the **Receive email notifications** check box if you want the user to receive SMTP notifications from LogAnalyzer.

   - Select the **Notify about alerts in web interface** check box to display alert notifications in the LogAnalyzer web interface. Clear this check box if you want to disable notification display. This check box is selected by default.

   - Select the **Display non-printable characters** check box to display non-printing characters such as spaces, tab characters, and line breaks in the LogAnalyzer web interface. If the **Display non-printable characters** check box is selected, you can press **`Ctrl/Command+*`** to enable and disable the display of non-printing characters.

     Spaces and tab characters are displayed in all input fields (except the **Description** field), in normalizers, filters, and connector resources, and in SQL queries for searching events in the **Events** section. Spaces are displayed as dots. A tab character is displayed as a dash in normalizers, filters and connectors. In other fields, a tab character is displayed as one or two dots.

     Line break characters are displayed in all input fields that support multi-line input, such as the event search field.

   - Under **Access**, use the **Add role groups** buttons to assign roles to the user in specific tenants. A user can be assigned different roles in different tenants; multiple roles can be assigned within the same tenant. For a domain user, the ability to change the main role (General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst) is blocked in the user card, while additional roles can be added or removed (Access to CII, Interaction with NCIRCC, Read shared resources), including management of additional role assignment to tenants.

   - In the **Space sets** drop-down list, select check boxes next to one or more space sets to which you want to grant access. If the user does not have a space set selected, access is granted to the default space set.

3. If you need to change the password, click the **Change password** button and fill in the fields described below in the opened window. When finished, click **OK**.

   - In the **Old password** field, enter the current password of the user account. The field is available if you change your account password.
   - In the **New password** field, enter a new password. You can enter a password or generate one by clicking the **Generate password** button. The password must satisfy the following requirements:
     - The length must satisfy the security policy.
     - At least one lowercase character.
     - At least one uppercase character.
     - At lease one numeral.
     - At least one special character from the list of special characters in the security policy.
     - No more than two identical characters in a row.
     - The password does not contain the login.
   - In the **Confirm password** field, enter the password again for confirmation.
   - If the **User must change password at next login** check box is selected, the next time the user attempts to log in to the LogAnalyzer web interface, the user must specify the old password and set a new password. After that, the user is allowed to log in to the LogAnalyzer web interface with the new password. This check box is selected by default.

     This setting is available only for *Active* accounts.

4. If necessary, use the **Generate token** button to generate an API token. Clicking this button displays the token creation window.

5. If necessary, configure the operations available to the user via the REST API by using the **API access rights** button.

6. If the **Unlimited password lifetime** check box is selected, the **Password lifetime** setting does not apply to the user account.

   We recommend enabling this setting for a service account, restricting the rights of such an account, and setting a password of maximum complexity. Typically, a user account with an unlimited password lifetime is used for monitoring. When migrating users, you can manually enable this setting for those service accounts that were previously used to display the dashboard.

7. Click **Save**.

The user account will be changed.

Special considerations involved in changing the set of spaces assigned to a user

A user is assigned one or more space sets; information about these is stored in the user account settings. To view spaces in a space set, hover over the space set in the user profile. If necessary, you can add or remove spaces from the space set in the **Settings** → **Access** → **Access to spaces** section.

If a space is deleted from the storage, the space is still displayed as part of the space set with the `deleted+UUID` designation. In this case, only the first 8 characters of the space's universal unique ID (UUID) are displayed. A deleted space is longer displayed as part of the space set after you save changes to that space set.

> A space is completely deleted from the space set if that space has no events. If there are some events in the space being deleted, LogAnalyzer displays the space as `deleted+UUID` even after saving changes to the space set.

LogAnalyzer services

Services are the main components of LogAnalyzer that help the system to manage events: services allow you to receive events from event sources and subsequently bring them to a common form that is convenient for storage and manual analysis. Each service consists of two parts that work together:

- One part of the service is created in the LogAnalyzer web interface based on a resource set for services.
- The second part of the service is installed in the network infrastructure where the LogAnalyzer system is deployed as one of its components. The server part of a service can consist of multiple instances: for example, services of the same agent or storage can be installed on multiple devices at once.

  > On the server side, LogAnalyzer services are located in the `/opt/kaspersky/kuma` directory.

  When you install LogAnalyzer in high availability mode, only the LogAnalyzer Core is installed in the cluster. Collectors and storages are hosted on hosts outside of the Kubernetes cluster.

Parts of services are connected to each other via the service ID.

Service types:

- Storages are used to save events.
- Collectors are used to receive events and convert them to LogAnalyzer format.
- Agents are used to receive events on remote devices and forward them to LogAnalyzer collectors.
- Event routers receive streams of events from collectors and then distribute the events to destinations in accordance with the configured filters.

In the LogAnalyzer web interface, services are displayed in the **Resources** → **Active services** section in table format. The table of services can be updated using the **Refresh** button and sorted by columns by clicking on the active headers. You can also configure the columns displayed in the table. To do so, click the gear button in the upper-right corner to display a drop-down list. In that drop-down list, select check boxes next to the names of the columns that you want to display in the table. You can leave any single column in the list to be displayed.

The maximum table size is not limited. If you want to select all services, scroll to the end of the table and select the **Select all** check box, which selects all available services in the table.

Table columns:

- **Status**—service status:
  - Green means the service is running and accessible from the Core server.
  - Red means the service is not running or is not accessible from the Core server.
  - Yellow is the status that applies to all services except the agent. The yellow status means that the service is running, but there are errors in the service log, or there are alerts for the service from Victoria Metrics. You can view the error message by hovering the mouse cursor over the status of the service in the **Active services** section.
  - Purple is the status that is applied to running services whose configuration file in the database has changed, but that have no other errors. If a service has an incorrect configuration file and has errors, for example, from Victoria Metrics, status of the service is yellow.
  - Gray means that if a deleted tenant had a running service that is still running, that service is displayed with a gray status on the Active services page. Services with the gray status are kept when you delete the tenant to let you copy the ID and remove services on your servers. Only the General administrator can delete services with the gray status. When a tenant is deleted, the services of that tenant are assigned to the Main tenant.
- **Type**—type of service: **agent**, **collector**, **storage**, **event router**.
- **Name**—name of the service. Clicking on the name of the service opens its settings.
- **Version**—service version.
- **Tenant**—the name of the tenant that owns the service.
- **FQDN**—fully qualified domain name of the service server.
- **IP address**—IP address of the server where the service is installed.
- **API port**—Remote Procedure Call port number.
- **Uptime**—Uptime of the service.
- **Created**—the date and time when the service was created.
- **UUID**—Unique ID of the service.

  By default, this column is not displayed in the table. You can add it to the table by clicking the gear icon in the upper-right part of the table of services and selecting the check box next to the name of the **UUID** column in the drop-down list.

You can sort data in the table in ascending and descending order, as well as by the **Status** parameter and by the service type in the **Type** column. To sort active services, right-click the context menu and select one or more statuses and a type.

You can use the buttons in the upper part of the **Services** window to perform the following group actions:

- **Add service**

  You can create new services based on existing service resource sets. We do not recommend creating services outside the main tenant without first carefully planning the inter-tenant interactions of various services and users.

- **Refresh**

  You can refresh the list of active services.

- **Update configuration**

  The **Update settings** button is not available if the LogAnalyzer Core service is among the services selected for group actions or if any of the selected services has the grey status. To make the **Update settings** button available for group actions, clear the check box from the LogAnalyzer Core service and services with the grey status.

- **Restart**

To perform an action with an individual service, right-click the service to display its context menu. The following actions are available:

- Reset certificate
- Delete
- Download log

  If you want to receive detailed information, enable the Debug mode in the service settings.

- Copy service ID

  You need this ID to install, restart, stop, or delete the service.

- Go to Events
- Go to active lists
- Go to context tables
- Go to partitions

To change a service, select a service under **Resources** → **Active services**. This opens a window with a set of resources based on which the service was created. You can edit the settings of the set of resources and save your changes. To apply the saved changes, restart the service.

If, when changing the settings of a collector resource set, you change or delete conversions in a normalizer connected to it, the edits will not be saved, and the normalizer itself may be corrupted. If you need to modify conversions in a normalizer that is already part of a service, the changes must be made directly to the normalizer under **Resources** → **Normalizers** in the web interface.

Services tools

This section describes the tools for working with services available in the **Resources** → **Active services** section of the LogAnalyzer web interface.

Getting service identifier

The service identifier is used to bind parts of the service residing within LogAnalyzer and installed in the network infrastructure into a single complex. An identifier is assigned to a service when it is created in LogAnalyzer, and is then used when installing the service to the server.

To get the identifier of a service:

1. Log in to the LogAnalyzer web interface and open **Resources** → **Active services**.
2. Select the check box next to the service whose ID you want to obtain, and click **Copy ID**.

   The identifier of the service will be copied to the clipboard. For instance, this ID can be used to install the service on a server.

Stopping, starting, checking status of the service

While managing LogAnalyzer, you may need to perform the following operations:

- Temporarily stop the service. For example, when restoring the Core from backup, or to edit service settings related to the operating system.
- Start the service.
- Check the status of the service.

The table below lists commands that may be useful while managing LogAnalyzer.

Commands for stopping, starting, and checking the status of a service

Service	Stop service	Start service	Check the status of the service
Core	`sudo systemctl stop kuma-core-< service ID >.service`	`sudo systemctl start kuma-core-< service ID >.service`	`sudo systemctl status kuma-core-< service ID >.service`
Services with an ID: • collector • storage	`sudo systemctl stop kuma-<collector/storage>-< service ID >.service`	`sudo systemctl start kuma-<collector/storage>-< service ID >.service`	`sudo systemctl status kuma-<collector/storage>-< service ID >.service`

Service	Stop service	Start service	Check the status of the service
Services without an ID: • kuma-grafana.service • kuma-mongodb.service • kuma-victoria-metrics.service • kuma-vmalert.service	`sudo systemctl stop <grafana/victoria-metrics/vmalert>.service`	`sudo systemctl start <grafana/victoria-metrics/vmalert>.service`	`sudo systemctl status <grafana/victoria-metrics/vmalert>.service`
Windows agents	To stop an agent service: 1. Copy the agent ID in the KUMA web interface. 2. Connect to the host on which you want to start the KUMA agent service. 3. Run PowerShell as an account with administrative rights. 4. Run the following command in PowerShell: `Stop-Service -Name "WindowsAgent-< agent ID >"`	To start an agent service: 1. Copy the agent ID in the KUMA web interface. 2. Connect to the host on which you want to start the KUMA agent service. 3. Run PowerShell as an account with administrative rights. 4. Run the following command in PowerShell: `Start-Service -Name "WindowsAgent-< agent ID >"`	To view the status of an agent service: 1. In Windows, go to the Start → Services menu, and in the list of services, double-click the relevant KUMA agent. 2. This opens a window; in that window, view the status of the agent in the Service status field.

Restarting the service

To restart the service:

1. Log in to the LogAnalyzer web interface and open **Resources** → **Active services**.
2. Select the check box next to the service and select the necessary option:
   - **Update configuration**—perform a hot update of a running service configuration. For example, you can change the field mapping settings or the destination point settings this way.
   - **Restart**—stop a service and start it again. This option is used to modify the port number or connector type.

     Special considerations involved in restarting LogAnalyzer agents:

     - LogAnalyzer Windows Agent can be restarted as described above only if it is running on a remote computer. If the service on the remote computer is inactive, you will receive an error when trying to restart from LogAnalyzer. In that case you must restart LogAnalyzer Windows Agent service on the remote Windows machine. For information on restarting Windows services, refer to the documentation specific to the operating system version of your remote Windows computer.
     - LogAnalyzer Agent for Linux stops when this option is used. To start the agent again, you must execute the command that was used to start it.
   - **Reset certificate**—remove certificates that the service uses for internal communication. This option may not be used to renew the Core certificate. To renew LogAnalyzer Core certificates, they must be reissued.

     Special considerations for deleting Windows agent certificates:

     - If the agent has the green status and you select **Reset certificate**, LogAnalyzer deletes the current certificate and creates a new one, the agent continues working with the new certificate.
     - If the agent has the red status and you select **Reset certificate**, LogAnalyzer generates an error that the agent is not running. In the %PROGRAMDATA%\Kaspersky Lab\KUMA\agent\<agent ID>\certificates folder, manually delete the internal.cert and internal.key files and start the agent manually. When the agent starts, a new certificate is created automatically.

     Special considerations for deleting Linux agent certificates:

     1. Regardless of the agent status, apply the **Reset certificate** option in the web interface to delete the certificate in the databases.
     2. In the agent installation folder **`/opt/kaspersky/agent/<Agent ID>/certificates`**, manually delete the internal.cert and internal.key files.
     3. Since the **Reset certificate** option stops the agent, to continue its operation, start the agent manually. When the agent starts, a new certificate is created automatically.

Deleting the service

Before deleting the service get its ID. The ID will be required to remove the service for the server.

To remove a service in the LogAnalyzer web interface:

1. Log in to the LogAnalyzer web interface and open **Resources** → **Active services**.
2. Select the check box next to the service you want to delete, and click **Delete**.

   A confirmation window opens.
3. Click **OK**.

The service is deleted from LogAnalyzer.

To remove a service from the server, run the following command:

```
sudo /opt/kaspersky/kuma/kuma <collector/correlator/storage/eventRouter/metrics> --id
<service ID> --uninstall
The service has been deleted from the server.
```

Partitions window

If the storage service was created and installed, you can view its partitions in the **Partitions** table.

*To open **Partitions** table:*

1. Log in to the LogAnalyzer web interface and open **Resources** → **Active services**.
2. Select the check box next to the relevant storage and click **Go to partitions**.

The **Partitions** table opens.

The table has the following columns:

- **Tenant**—the name of the tenant that owns the stored data.
- **Created**—partition creation date.
- **Space**—the name of the space.
- **Size**—the size of the space.
- **Events**—the number of stored events.
- **Transfer to cold storage**—the date when data will be migrated from the ClickHouse clusters to cold storage disks.
- **Expires**—the date when the partition expires. After this date, the partition and the events it contains are no longer available.

You can delete partitions.

To delete a partition:

1. Open the **Partitions** table (see above).
2. Open the ... drop-down list to the left from the required partition.
3. Select **Delete**.

   A confirmation window opens.
4. Click **OK**.

The partition has been deleted. Audit event partitions cannot be deleted.

Searching for related events

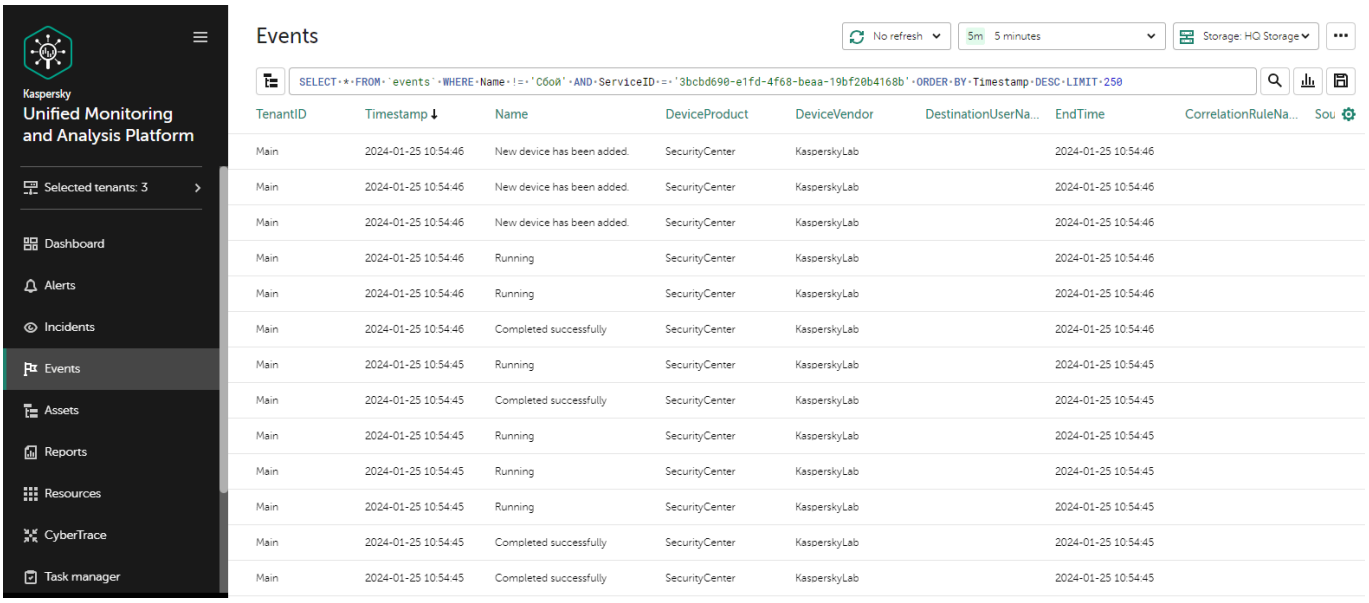
You can search for events processed by a certain collector.

To search for events related to a collector:

1. Log in to the LogAnalyzer web interface and open **Resources** → **Active services**.
2. Select the check box next to the relevant collector and click **Go to Events**.

   This opens a new browser tab with the LogAnalyzer **Events** section open.
3. To find events, click the icon.

   A table with events selected by the search expression `ServiceID = <ID of the selected service>` will be displayed.



Event search results

When searching for events, you may encounter the following shard unavailability error:

```
Code: 279. DB::NetException: All connection tries failed. Log: \\n\\nTimeout exceeded
while connecting to socket (host.example.com:port, connection timeout 1000 ms)\\nTimeout
exceeded while connecting to socket (host.example.com:port, connection timeout 1000
ms)\\nTimeout exceeded while connecting to socket (host.example.com:port, connection
timeout 1000 ms)\\n\\n: While executing Remote. (ALL_CONNECTION_TRIES_FAILED) (version
23.8.8.207)\\n\"}",
```

In this case, you need to override the ClickHouse configuration in storage settings.

To override the ClickHouse configuration:

1. In the LogAnalyzer web interface, select the **Resources** section.
2. In the **Services configuration** → **Storages** section, click the storage resource that you want to edit.

   This opens the **Edit storage** window.
3. In the **ClickHouse configuration override** field, specify the following settings:

   - To skip unavailable shards when searching, insert the following lines:

     ```
     <profiles>
             <default>
                     <skip_unavailable_shards>1</skip_unavailable_shards>
             </default>
     </profiles>
     ```

   - To set memory usage limits, specify `max_server_memory_usage` or `max_memory_usage_for_all_queries` parameters as recommended.
4. To apply the ClickHouse configuration, click **Save**.
5. Restart the storage services that depend on this resource.

   This resolves the shard unavailability error, and you can proceed to search for events processed by a particular collector.

Service resource sets

Service resource sets are a resource type, a LogAnalyzer component, a set of settings based on which the LogAnalyzer services are created and operate. Resource sets for services are collections of resources.

Any resources added to a resource set must be owned by the same tenant that owns the created resource set. An exception is the Shared tenant, whose owned resources can be used in the sets of resources of other tenants.

Resource sets for services are displayed in the **Resources** → **<type of the service resource set>** section of the LogAnalyzer web interface. Available types:

- Collectors
- Storages
- Agents

When you select the required type, a table opens with the available sets of resources for services of this type. The resource table contains the following columns:

- **Name**—the name of a resource set. Can be used for searching and sorting.
- **Updated**—the date and time of the last update of the resource set. Can be used for sorting.
- **Created by**—the name of the user who created the resource set.
- **Description**—the description of the resource set.

Creating a storage

A storage consists of two parts: one part is created inside the LogAnalyzer web interface, and the other part is installed on network infrastructure servers intended for storing events. The server part of a LogAnalyzer storage consists of ClickHouse nodes collected into a cluster. ClickHouse clusters can be supplemented with cold storage disks.

For each ClickHouse cluster, a separate storage must be installed.

Prior to storage creation, carefully plan the cluster structure and deploy the necessary network infrastructure. When choosing a ClickHouse cluster configuration, consider the specific event storage requirements of your organization.

We recommend using the ext4 file system.

A storage is created in several steps:

1. **Creating a resource set for a storage in the LogAnalyzer web interface**
2. **Creating a storage service in the LogAnalyzer web interface**
3. **Installing storage nodes in the network infrastructure**

When creating storage cluster nodes, verify the network connectivity of the system and open the ports used by the components.

If the storage settings are changed, the service must be restarted.

ClickHouse cluster structure

A ClickHouse *cluster* is a logical group of devices that possess all accumulated normalized LogAnalyzer events. It consists of one or more logical *shards*.

A *shard* is a logical group of devices that possess a specific **portion** of all normalized events accumulated in the cluster. It consists of one or more *replicas*. Increasing the number of shards lets you do the following:

- Accumulate more events by increasing the total number of servers and disk space.
- Absorb a larger **stream** of events by distributing the load associated with an influx of new events.
- Reduce the time taken to search for events by distributing search zones among multiple devices.

A *replica* is a device that is a member of a logical shard and possesses a single copy of that shard's data. If multiple replicas exist, it means multiple copies exist (the data is replicated). Increasing the number of replicas lets you do the following:

- Improve high availability.
- Distribute the total load related to data searches among multiple machines (although it's best to increase the number of shards for this purpose).

A *keeper* is a device that participates in **coordination** of data replication at the **whole** cluster level. At least one device per cluster must have this role. The recommended number of the devices with this role is 3. The number of devices involved in coordinating replication must be an **odd** number. The *keeper* and *replica* roles can be combined in one machine.

ClickHouse cluster node settings

Prior to storage creation, carefully plan the cluster structure and deploy the necessary network infrastructure. When choosing a ClickHouse cluster configuration, consider the specific event storage requirements of your organization.

When creating ClickHouse cluster nodes, verify the network connectivity of the system and open the ports used by the components.

For each node of the ClickHouse cluster, you need to specify the following settings:

- Fully qualified domain name (FQDN)—a unique address to access the node. Specify the entire FQDN, for example, `kuma-storage.example.com`.
- Shard, replica, and keeper IDs—the combination of these settings determines the position of the node in the ClickHouse cluster structure and the node role.

Node roles

The roles of the nodes depend on the specified settings:

- shard, replica, keeper—the node participates in the accumulation and search of normalized LogAnalyzer events and helps coordinate data replication at the cluster-wide level.
- shard, replica—the node participates in the accumulation and search of normalized LogAnalyzer events.
- keeper—the node **does not** accumulate normalized events, but helps coordinate data replication at the cluster-wide level. Dedicated keepers must be specified at the beginning of the list in the **Resources** → **Storages** → <Storage> → **Basic settings** → **ClickHouse cluster nodes** section.

ID requirements:

- If multiple shards are created in the same cluster, the shard IDs must be unique within this cluster.
- If multiple replicas are created in the same shard, the replica IDs must be unique within this shard.
- The keeper IDs must be unique within the cluster.

Example of ClickHouse cluster node IDs:

- shard 1, replica 1, keeper 1;
- shard 1, replica 2;
- shard 2, replica 1;
- shard 2, replica 2, keeper 3;
- shard 2, replica 3;
- keeper 2.

Cold storage of events

In LogAnalyzer, you can configure the migration of legacy data from a ClickHouse cluster to the cold storage. Cold storage can be implemented using the local disks mounted in the operating system or the Hadoop Distributed File System (HDFS). Cold storage is enabled when at least one cold storage disk is specified. If multiple storages are used, a cold storage disk or a HDFS disk must be mounted at the path specified in the storage configuration on each node with data. If a cold storage disk is not configured and the server runs out of disk space in hot storage, the storage service is stopped. If both hot storage and cold storage are configured, and space runs out on the cold storage disk, the KUMA storage service is stopped. We recommend avoiding such situations by adding custom event storage conditions in hot storage.

Cold storage disks can be added or removed. If you have added multiple cold storage disks, data is written to them in a round-robin manner. If data to be written to disk would take up more space than is available on that disk, this data and all subsequent data is written round-robin to the next cold storage disks. If you added only two cold storage disks, the data is written to the drive that has free space left.

After changing the cold storage settings, the storage service must be restarted. If the service does not start, the reason is specified in the storage log.

If the cold storage disk specified in the storage settings has become unavailable (for example, out of order), this may lead to errors in the operation of the storage service. In this case, recreate a disk with the same path (for local disks) or the same address (for HDFS disks) and then delete it from the storage settings.

Rules for moving the data to the cold storage disks

You can configure the storage conditions for events in hot storage of the ClickHouse cluster by setting a limit based on retention time or on maximum storage size. When cold storage is used, every 15 minutses and after each Core restart, KUMA checks if the specified storage conditions are satisfied:

1. LogAnalyzer gets the partitions for the storage being checked and groups the partitions by cold storage disks and spaces.

2. For each space, LogAnalyzer checks whether the specified storage condition is satisfied.

3. If the condition is satisfied (for example, if the space contains that exceed their retention time, or if the size of the storage has reached or exceeded limit specified in the condition), LogAnalyzer transfers all partitions with the oldest date to cold storage disks or deletes these partitions if no cold storage disk is configured or if it is configured incorrectly. This action is repeated while the configured storage condition remains satisfied in the space; for example, if after deleting partitions for a date, the storage size still exceeds the maximum size specified in the condition.

   LogAnalyzer generates an audit event when data transfer starts and ends, or when data is removed.

4. If retention time is configured in LogAnalyzer, whenever partitions are transferred to cold storage disks, it is checked whether the configured conditions are satisfied on the disk. If events are found on the disk that have been stored for longer than the **Event retention time**, which is counted from the moment the events were received in LogAnalyzer, the solution deletes these events or all partitions for the oldest date.

   LogAnalyzer generates an audit event when data is deleted.

If the ClickHouse cluster disks are 95% full, the biggest partitions are automatically moved to the cold storage disks. This can happen more often than once per hour.

During data transfer, the storage service remains operational, and its status stays green in the **Resources** → **Active services** section of the LogAnalyzer web interface. When you hover over the status icon, a message is displayed about the data transfer. When a cold storage disk is removed, the storage service has the yellow status.

Special considerations for storing and accessing events

- When using HDFS disks for cold storage, protect your data in one of the following ways:
  - Configure a separate physical interface in the VLAN, where only HDFS disks and the ClickHouse cluster are located.
  - Configure network segmentation and traffic filtering rules that exclude direct access to the HDFS disk or interception of traffic to the disk from ClickHouse.
- Events located in the ClickHouse cluster and on the cold storage disks are equally available in the LogAnalyzer web interface. For example, when searching for events.
- You can disable the storage of events or audit events on cold storage disks. To do so, specify the following in storage settings:
  - If you do not want to store events on cold storage disks, do one of the following:
    - If in the **Retention policy options** field, you have selected a gigabyte or percentage based storage condition, in the **Events storage time** field, specify `0`.
    - If in the **Retention policy options** field, you have selected a storage condition in days, in the **Event storage time** field, specify the same number of days as in the **Retention policy options**.
  - If you do not want to store audit events on cold storage disks, in the **Audit cold retention period** field, specify `0` (days).

Special considerations for using HDFS disks

- Before connecting HDFS disks, create directories for each node of the ClickHouse cluster on them in the following format: `<HDFS disk host>/<shard ID>/<replica ID>`. For example, if a cluster consists of two nodes containing two replicas of the same shard, the following directories must be created:
  - hdfs://hdfs-example-1:9000/clickhouse/1/1/
  - hdfs://hdfs-example-1:9000/clickhouse/1/2/

  Events from the ClickHouse cluster nodes are migrated to the directories with names containing the IDs of their shard and replica. If you change these node settings without creating a corresponding directory on the HDFS disk, events may be lost during migration.
- HDFS disks added to storage operate in the JBOD mode. This means that if one of the disks fails, access to the storage will be lost. When using HDFS, take high availability into account and configure RAID, as well as storage of data from different replicas on different devices.
- The speed of event recording to HDFS is usually lower than the speed of event recording to local disks. The speed of accessing events in HDFS, as a rule, is significantly lower than the speed of accessing events on local disks. When using local disks and HDFS disks at the same time, the data is written to them in turn.
- HDFS is used only as distributed file data storage of ClickHouse. Compression mechanisms of ClickHouse, not HDFS, are used to compress data.
- The ClickHouse server must have write access to the corresponding HDFS storage.

Removing cold storage disks

Before physically disconnecting cold storage disks, remove these disks from the storage settings.

To remove a disk from the storage settings:

- In the LogAnalyzer web interface, under **Resources** → **Storages**, select the relevant storage.

  This opens the storage.

- In the window, in the **Disks for cold storage** section, in the required disk's group of settings, click **Remove disk**.

  Data from removed disk is automatically migrated to other cold storage disks or, if there are no such disks, to the ClickHouse cluster. While data is being migrated, the status icon of the storage turns yellow and an hourglass icon is displayed. Audit events are generated when data transfer starts and ends.

- After event migration is complete, the disk is automatically removed from the storage settings. It can now be safely disconnected.

Removed disks can still contain events. If you want to delete them, you can manually delete the data partitions using the DROP PARTITION command.

If the cold storage disk specified in the storage settings has become unavailable (for example, out of order), this may lead to errors in the operation of the storage service. In this case, create a disk with the same path (for local disks) or the same address (for HDFS disks) and then delete it from the storage settings.

Detaching, archiving, and attaching partitions

If you want to optimize disk space and speed up queries in LogAnalyzer, you can detach data partitions in ClickHouse, archive partitions, or move partitions to a drive. If necessary, you can later reattach the partitions you need and perform data processing.

You can also use the created partition archives as backup copies of these partitions.

Detaching partitions

To detach partitions:

1. Determine the shard on all replicas of which you want to detach the partition.

2. Get the partition ID using the following command:

   ```
   sudo /opt/kaspersky/kuma/storage/<storage ID>/deps/clickhouse/bin/client.sh -d kuma --multiline --query "SELECT partition, name FROM system.parts;" |grep 20231130
   ```

   In this example, the command returns the partition ID for November 30, 2023.

3. One each replica of the shard, detach the partition using the following command and specifying the partition ID:

   ```
   sudo /opt/kaspersky/kuma/storage/<storage ID>/deps/clickhouse/bin/client.sh -d kuma --multiline --query "ALTER TABLE events_local_v2 DETACH PARTITION ID '<partition ID>'"
   ```

As a result, the partition is detached on all replicas of the shard. Now you can move the data directory to a drive or archive the partition.

Archiving partitions

To archive detached partitions:

1. Find the detached partition in disk subsystem of the server:

   ```
   sudo find /opt/kaspersky/kuma/clickhouse/data/ -name <ID of the detached partition>\*
   ```

2. Change to the 'detached' directory that contains the detached partition, and while in the 'detached' directory, perform the archival:

   ```
   sudo cd <path to the 'detached' directory containing the detached partition>
   ```

   ```
   sudo zip -9 -r detached.zip *
   ```

   For example:

   ```
   sudo cd /opt/kaspersky/kuma/clickhouse/data/store/d5b/d5bdd8d8-e1eb-4968-95bd-d8d8e1eb3968/detached/
   ```

   ```
   sudo zip -9 -r detached.zip *
   ```

The partition is archived.

Attaching partitions

To attach archived partitions to LogAnalyzer:

1. Increase the **Retention period** value.

   LogAnalyzer deletes data based on the date specified in the Timestamp field, which records the time when the event is received, and based on the **Retention period** value that you set for the storage.

   Before restoring archived data, make sure that the **Retention period** value overlaps the date in the Timestamp field. If this is not the case, the archived data will be deleted within 1 hour.

2. Place the archive partition in the 'detached' section of your storage and extract the archive:

   ```
   sudo unzip detached.zip -d <path to the 'detached' directory>
   ```

   For example:

   ```
   sudo unzip detached.zip -d /opt/kaspersky/kuma/clickhouse/data/store/d5b/d5bdd8d8-e1eb-4968-95bd-d8d8e1eb3968/detached/
   ```

3. Run the command to attach the partition:

   ```
   sudo /opt/kaspersky/kuma/storage/<storage ID>/deps/clickhouse/bin/client.sh -d kuma --multiline --query "ALTER TABLE events_local_v2 ATTACH PARTITION ID '<partition ID>'"
   ```

   ```
   Repeat the steps of extracting the archive and attaching the partition on each replica of the shard.
   ```

As a result, the archived partition is attached and its events are again available for search.

Creating a set of resources for a storage

In the LogAnalyzer web interface, a storage service is created based on the resource set for the storage.

To create a resource set for a storage in the LogAnalyzer web interface:

1. In the LogAnalyzer web interface, under **Resources** → **Storages**, click **Add storage**.

   This opens the **Create storage** window.

2. On the **Basic settings** tab, in the **Storage name** field, enter a unique name for the service you are creating. The name must contain 1 to 128 Unicode characters.

3. In the **Tenant** drop-down list, select the tenant that will own the storage.

4. In the **Tags** drop-down list, select the tags for the resource set that you are creating.

   The list includes all available tags created in the tenant of the resource and in the Shared tenant. You can find a tag in the list by typing its name in the field. If the tag you entered does not exist, you can press **Enter** or click **Add** to create it.

5. You can optionally add up to 256 Unicode characters describing the service in the **Description** field.

6. In the **Retention policy options** field, select an event storage condition in the ClickHouse cluster for the storage, which, when satisfied, will cause events to be transferred to cold storage disks or deleted if cold storage is not configured or is configured incorrectly. The condition is applied to the default space and to events from deleted spaces.

   By default, ClickHouse moves events to cold storage disks or deletes them if more than 97% of the storage is full. LogAnalyzer also applies an additional **365 days** storage condition when creating a storage. You can configure custom storage conditions for more stable performance of the storage.

   To set the storage condition, do one of the following:

   - If you want to limit the storage period for events, select **Days** from the drop-down list, and in the field, specify the maximum event storage period (in days) in the ClickHouse hot storage cluster.

     After the specified period, events are automatically transferred to cold storage disks or deleted from the ClickHouse cluster, starting with the partitions with the oldest date. The minimum value is 1. The default value is 365.

     For [OOTV] Storage, the default event storage period is 2 days. If you want to use this storage, you can change the event storage condition for it, if necessary.

   - If you want to limit the maximum storage size, select **GB** from the drop-down list, and in the field, specify the maximum storage size in gigabytes.

     When the storage reaches the specified size, events are automatically transferred to cold storage disks or deleted from the ClickHouse cluster, starting with the partitions with the oldest date. The minimum value and default value is 1.

   - If you want to limit the storage size to a percentage of disk space that is available to the storage, select **Percent** from the drop-down list and specify the maximum storage size as a percentage of the available disk space.

     > You can get information about available disk space from ClickHouse by sending the `'SELECT hostname(), name, total_space FROM system.disks'` query.

     When the storage reaches the specified percentage of disk space available to it, events are automatically transferred to cold storage disks or deleted from the ClickHouse cluster, starting with the partitions with the oldest date. Possible values: 1 to 95. The default value is 80. If you want to use percentages for all storage spaces, the sum total of percentages in the conditions of all spaces may not exceed 95, but we recommend specifying a limit of at most 90% for the entire storage or for individual spaces.

     We do not recommend specifying a value less than 5% because this increases the likelihood of data loss in the storage.

     > If the storage is organized as a cluster, the amount of occupied disk space is determined for each of the several servers separately. Stored data may be unevenly distributed among disks on the servers. LogAnalyzer pushes events to cold storage disks or deletes events when the maximum storage size is reached *on any individual server in the cluster*.

[The algorithm for determining the size of occupied storage space]

The size of occupied storage space is determined as follows:

1. Information about the state of LogAnalyzer disks and partitions is obtained from all available replicas.
2. If occupied space exceeds the configured values, LogAnalyzer takes action depending on the type of data storage: hot or cold.

   For hot storage, the following actions are performed for each available node and space:

   a. The total size of the partitions of this space on the replica is calculated.

   b. The maximum space size on the replica is calculated:

   - If an absolute value in GB is set for the storage size, the maximum space size on the replica is calculated as follows:

     `<maximum space size> = <configured threshold> / <number of replicas in the cluster>`

     For example, consider a cluster with three shards, each of which contains two replicas; a threshold of 6 GB is set, and the maximum size of data on a replica is 1 GB.

   - If a relative value is set for the storage size as a percentage, the maximum space on the replica is calculated as follows:

     `<maximum space size> = <space size on the replica> / <size of the replica disk> * 100%`

     > The size of the replica disk does not include the space reserved for `ext4`.

   If the specified threshold is exceeded, partitions on the host are deleted or moved to cold storage. Partitions are deleted on all shards.

   For cold storage, partitions are deleted for each available space in accordance with the event retention period (TTL) configured in LogAnalyzer.

7. If you want to use an additional storage condition, click **Add policy options** and specify an additional storage condition as described in step 6.

   The maximum number of conditions is two, and you can combine only conditions the following types:

   - Days and storage size in GB
   - Days and storage size as a percentage

   If you want to delete a storage condition, click the X icon next to this condition.

8. In the **Audit retention period** field, specify the period, in days, to store audit events. The minimum value and default value is `365`.

9. If cold storage is required, specify the event storage term:

   - **Event retention time** specifies the total LogAnalyzer event storage duration in days, counting from the moment when the event is received. When the specified period expires, events are automatically deleted from the cold storage disk. The default value is 0.

     The event retention time is calculated as the sum of the event retention time in the ClickHouse hot storage cluster until the condition specified in the **Retention policy options** setting is triggered, and the event retention time on the cold storage disk. After one of storage conditions is triggered, the data partition for the earliest date is moved to the cold storage disk, and there it remains until the event retention time in LogAnalyzer expires.

     Depending on the specified storage condition, the resulting retention time is as follows:

     - If you specified a storage condition in days, the **Event retention time** must be strictly greater than the number of days specified in the storage condition. You can calculate the cold retention period for events as the **Events storage time** minus the number of days specified in the **Retention policy options** setting.

       If you do not want to store events on the cold storage disk, you can specify the same number of days in the **Event retention time** field as in the storage condition.

     - If you specified the storage condition in terms of disk size (absolute or percentage), the minimum value of the **Event retention time** is 1. The cold storage duration for events is calculated as **Event retention time** minus the number of days from the receipt of the event to triggering of the condition and the disk partition filling up, but until the condition is triggered, calculating an exact duration is impossible. In this case, we recommend specifying a relatively large value for **Event retention time** to avoid events being deleted.

       If you do not want to store events on the cold storage disk, you can set **Event retention time** to 0.

   - **Audit cold retention period**—the number of days to store audit events. The minimum value is 0.

   The **Events storage time** and **Audit cold retention period** settings become available only after at least one cold storage disk has been added.

10. If you want to change ClickHouse settings, in the **ClickHouse configuration override** field, paste the lines with settings from the ClickHouse configuration XML file /opt/kaspersky/kuma/clickhouse/cfg/config.xml. Specifying the root elements <yandex>, </yandex> is not required. Settings passed in this field are used instead of the default settings.

    Example:

    <merge_tree>

    <parts_to_delay_insert>600</parts_to_delay_insert>

    <parts_to_throw_insert>1100</parts_to_throw_insert>

    </merge_tree>

11. Use the **Debug** toggle switch to specify whether resource logging must be enabled. If you want to only log errors for all KUMA components, disable debugging. If you want to get detailed information in the logs, enable debugging.

12. If necessary, in the **ClickHouse cluster nodes** section, add ClickHouse cluster nodes to the storage.

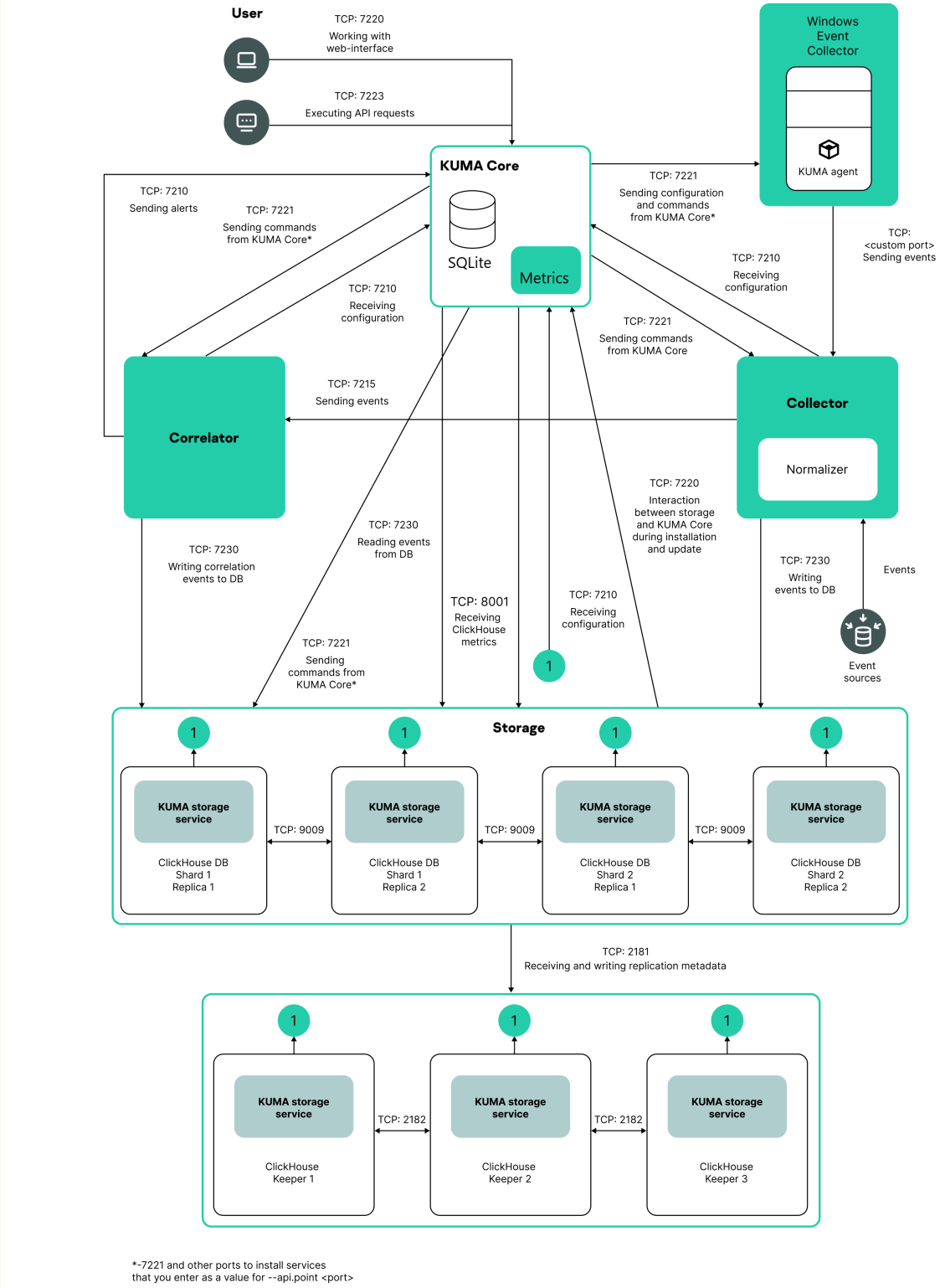
    There can be multiple nodes. You can add nodes by clicking the **Add node** button or remove nodes by clicking the X icon of the relevant node.

    Available settings:

    - In the **FQDN** field, enter the fully qualified domain name of the node that you want to add. For example, `kuma-storage-cluster1-server1.example.com`.

    - In the **Shard ID**, **Replica ID**, and **Keeper ID** fields, specify the role of the node in the ClickHouse cluster. The shard and keeper IDs must be unique within the cluster, the replica ID must be unique within the shard. The

following example shows how to populate the **ClickHouse cluster nodes** section for a storage with dedicated keepers in a **distributed installation**. You can adapt this example according to your needs.



Distributed Installation diagram

Example:

ClickHouse cluster nodes

FQDN: kuma-storage-cluster1-server1.example.com

Shard ID: 0

Replica ID: 0
Keeper ID: 1
FQDN: kuma-storage-cluster1server2.example.com
Shard ID: 0
Replica ID: 0
Keeper ID: 2
FQDN: kuma-storage-cluster1server3.example.com
Shard ID: 0
Replica ID: 0
Keeper ID: 3
FQDN: kuma-storage-cluster1server4.example.com
Shard ID: 1
Replica ID: 1
Keeper ID: 0
FQDN: kuma-storage-cluster1server5.example.com
Shard ID: 1
Replica ID: 2
Keeper ID: 0
FQDN: kuma-storage-cluster1server6.example.com
Shard ID: 2
Replica ID: 1
Keeper ID: 0
FQDN: kuma-storage-cluster1server7.example.com
Shard ID: 2
Replica ID: 2
Keeper ID: 0

13. If necessary, in the **Spaces** section, add spaces to the storage to distribute the stored events.

    There can be multiple spaces. You can add spaces by clicking the **Add space** button or remove nodes by clicking the X icon of the relevant space.

    Available settings:

    - In the **Name** field, specify a name for the space containing 1 to 128 Unicode characters.

    - In the **Retention policy options** field, select an event storage condition in the ClickHouse cluster for the space, which, when satisfied, will cause events to be transferred to cold storage disks or deleted if cold storage is not configured or is configured incorrectly. LogAnalyzer applies the **365 days** storage condition when a space is added.

      To set the storage condition for a space, do one of the following:

      - If you want to limit the storage period for events, select **Days** from the drop-down list, and in the field, specify the maximum event storage period (in days) in the ClickHouse hot storage cluster.

        After the specified period, events are automatically transferred to cold storage disks or deleted from the ClickHouse cluster, starting with the partitions with the oldest date. The minimum value is 1. The default value is 365.

      - If you want to limit the maximum storage space size, select **GB** from the drop-down list, and in the field, specify the maximum space size in gigabytes.

        When the space reaches the specified size, events are automatically transferred to cold storage disks or deleted from the ClickHouse cluster, starting with the partitions with the oldest date. The minimum value and default value is 1.

      - If you want to limit the space size to a percentage of disk space that is available to the storage (according to VictoriaMetrics), select **Percent** from the drop-down list, and in the field, specify the maximum space size as a percentage of the size of the disk available to the storage. In this case, the condition can also be triggered when the disk space available to the storage is decreased.

        When the space reaches the specified percentage of disk space available to the storage, events are automatically transferred to cold storage disks or deleted from the ClickHouse cluster, starting with the partitions with the oldest date. Possible values: 1 to 95. The default value is 80. If you want to use percentages for all storage spaces, the sum total of percentages in the conditions of all spaces may not exceed 95, but we recommend specifying a limit of at most 90% for the entire storage or for individual spaces.

        We do not recommend specifying small percentage values because this increases the probability of data loss in the storage.

      When using size as the storage condition, you must ensure that the total size of all spaces specified in the storage conditions does not exceed the physical size of the storage, otherwise an error will be displayed when starting the service.

      In storage conditions with a size limitation, use the same units of measure for all spaces of a storage (only gigabytes or only percentage values). Otherwise, if the condition is specified as a percentage for one space, and in gigabytes for another space, the storage may overflow due to mismatch of values, leading to data loss.

    - If you want to make a space inactive if it is outdated and no longer relevant, select the **Read only** check box.

      This prevents events from going into that space. To make the space active again, clear the **Read only** check box. This check box is cleared by default.

- If necessary, in the **Event retention time** field, specify the total LogAnalyzer event storage duration in days, counting from the moment when the event is received. When the specified period expires, events are automatically deleted from the cold storage disk. The default value is 0.

  The event retention time is calculated as the sum of the event retention time in the ClickHouse hot storage cluster until the condition specified in the **Retention policy options** setting is triggered, and the event retention time on the cold storage disk. After one of storage conditions is triggered, the data partition for the earliest date is moved to the cold storage disk, and there it remains until the event retention time in LogAnalyzer expires.

  Depending on the specified storage condition, the resulting retention time is as follows:

  - If you specified a storage condition in days, the **Event retention time** must be strictly greater than the number of days specified in the storage condition. The cold storage duration for events is calculated as the **Event storage time** minus the number of days specified in the **Retention policy options** setting.

    If you do not want to store events from this space on the cold storage disk, you can specify the same number of days in the **Event retention time** field as in the storage condition.

  - If you specified the storage condition in terms of disk size (absolute or percentage), the minimum value of the **Event retention time** is 1. The cold storage duration for events is calculated as **Event retention time** minus the number of days from the receipt of the event to triggering of the condition and the disk partition filling up, but until the condition is triggered, calculating an exact duration is impossible. In this case, we recommend specifying a relatively large value for **Event retention time** to avoid events being deleted.

    If you do not want to store events from this space on the cold storage disk, you can set **Event retention time** to 0.

  The **Event retention time** setting becomes available only after adding at least one cold storage disk.

- In the **Filter settings** section, you can specify conditions to identify events that will be put into this space. To create a new filter, in the **Filter** drop-down list, select an existing filter or **Create new**.

  Creating a filter in resources

To create a filter:

1. In the **Filter** drop-down list, select **Create new**.

2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.

3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

4. Under **Conditions**, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, fields of additional parameters for identifying the value to be passed to the filter may be displayed. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

   c. In the **Operator** drop-down list, select an operator.

Filter operators

- **=** means the left operand equals the right operand.
- **<** means the left operand is less than the right operand.
- **<=** means the left operand is less than or equal to the right operand.
- **>** means the left operand is greater than the right operand.
- **>=** means the left operand is greater than or equal to the right operand.
- **inSubnet** means the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains** means the left operand contains values of the right operand.
- **startsWith** means the left operand starts with one of the values of the right operand.
- **endsWith** means the left operand ends with one of the values of the right operand.
- **match** means the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit** checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
- **hasVulnerability** checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- The **inActiveList** operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary** checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory** means the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup** means the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- The **TIDetect** operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage.

- **inContextTable** checks for the presence of the entry in the specified context table.
- **intersect** checks for the presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. This check box does not apply to the **InSubnet**, **InActiveList**, **InCategory**, or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, from the **If** drop-down list, select **If not**.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

After the service is created, you can view and delete spaces in the storage resource settings.

There is no need to create a separate space for audit events. Events of this type (Type=4) are automatically placed in a separate Audit space with a storage term of at least 365 days. This space cannot be edited or deleted from the LogAnalyzer web interface.

14. If necessary, in the **Disks for cold storage** section, add to the storage the disks where you want to transfer events from the ClickHouse cluster for long-term storage.

    There can be multiple disks. You can add disks by clicking the **Add disk** button and remove them by clicking the **Remove disk** button.

    Available settings:

    - In the **FQDN** drop-down list, select the type of domain name of the disk you are connecting:
        - **Local**—for the disks mounted in the operating system as directories.
        - **HDFS**—for the disks of the Hadoop Distributed File System.
    - In the **Name** field, specify the disk name. The name must contain 1 to 128 Unicode characters.
    - If you select the **Local** domain name type for the disk, specify the absolute directory path of the mounted local disk in the **Path** field. The path must begin and end with a "/" character.

      > The directory of the mounted local disk must be owned by the LogAnalyzer user.

    - If you select **HDFS** domain name type for the disk, specify the path to HDFS in the **Host** field. For example, `hdfs://hdfs1:9000/clickhouse/`.

15. Go to the **Advanced settings** tab and fill in the following fields:

    - In the **Buffer size** field, enter the buffer size in bytes at which events must be sent to the database. The default value is 128 MB. No maximum value is configured. If the virtual machine has less free RAM than the specified **Buffer size**, KUMA sets the limit to 128 MB.

      If the expected traffic to the storage service is greater than 1 Gbps, we recommend changing the **Buffer size** setting (in bytes) to the expected value and setting the **Buffer flush interval** to 2.

      The value of the **Buffer size** setting can be determined empirically. If the **Insert QPS** metric of the storage is greater than 1 and you see that a queue of requests building up, follow the recommendations for calculating and adjusting the **Buffer size** setting in the Insert QPS section of the Viewing LogAnalyzer metrics article.

    - In the **Maximum buffer size** field, specify the maximum buffer size in bytes. The default value is 256 MB.

      By default, events are written to the database once per second. If the EPS rises and events are registered more often than once per second, the buffer size is automatically increased. The buffer is increased up to three times the **Buffer size** value, but cannot exceed the **Maximum buffer size** value. When the load goes back to normal, the buffer size stays the same; there is no automatic reduction of buffer size.

      If the **Maximum buffer size** is less than or equal to the **Buffer size**, the buffer size does not change.

      Restarting the storage service resets the increased buffer size to the value in the **Buffer size** field.

    - In the **Buffer flush interval** field, enter the time in seconds for which LogAnalyzer waits for the buffer to fill up. If the buffer is not full, but the specified time has passed, LogAnalyzer sends events to the database. The default value is 1 second.

    - In the **Required memory size** field, specify the minimum amount of free RAM in bytes. The default value is 128 MB.

      If the node on which the storage is installed has less free RAM than this setting, database insert request returns an error, and the events are recorded later. Limiting the memory size allows avoiding the situation when the memory fills up if, for any reason, the recording of events in the database is blocked and data begins to accumulate in memory.

- In the **Disk buffer size limit** field, enter a value in bytes. The disk buffer is used to temporarily store events that could not be sent for further processing or storage. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer. The default value is 10 GB.
- Use the **Disk buffer** toggle switch to enable or disable the disk buffer. By default, the disk buffer is enabled.
- Use the **Write to local database table** toggle switch to enable or disable writing to the local database table. Writing is disabled by default.

  If enabled, data is written only on the host on which the storage is located. We recommend using this functionality only if you have configured balancing on the collector at step **6. Routing**, in the **Advanced settings** section, the **URL selection policy** field is set to **Round robin**.

  If you disable writing, the data is distributed across the shards of the cluster.
- If necessary, use the **Debug** toggle switch to enable logging of service operations.
- You can use the **Create dump periodically** toggle switch at the request of Technical Support to generate resource (CPU, RAM, etc.) utilization reports in the form of dumps.
- In the **Dump settings** field, you can specify the settings to be used when creating dumps. For instructions on filling out this field, please contact Technical Support.

The set of resources for the storage is created and is displayed under **Resources** → **Storages**. Now you can create a storage service.

Creating a storage service in the LogAnalyzer web interface

When a resource set is created for a storage, you can proceed to create a storage service in LogAnalyzer.

To create a storage service in the LogAnalyzer web interface:

1. In the LogAnalyzer web interface, under **Resources** → **Active services**, click **Add service**.
2. In the opened **Choose a service** window, select the resource set that you just created for the storage and click **Create service**.

The storage service is created in the LogAnalyzer web interface and is displayed under **Resources** → **Active services**. Now storage services must be installed to each node of the ClickHouse cluster by using the service ID.

Installing a storage in the LogAnalyzer network infrastructure

For the storage service to work correctly, the host name of the server must match one of its IP addresses. Review the /etc/hosts file to make sure it matches. You can view the list of local IP addresses using the `ip a` command.

To create a storage:

1. Log in to the server where you want to install the service.
2. Execute the following command:

   `sudo /opt/kaspersky/kuma/kuma storage --core https://<LogAnalyzer Core server FQDN>:<port used by LogAnalyzer Core for internal communication (port 7210 by default)> --id <service ID copied from the LogAnalyzer web interface> --install`

   Example: `sudo /opt/kaspersky/kuma/kuma storage --core https://kuma.example.com:7210 --id XXXXX --install`

   > When deploying several LogAnalyzer services on the same host, during the installation process you must specify unique ports for each component using the `--api.port <port>` parameter. The following setting values are used by default: `--api.port 7221`.

3. Repeat steps 1–2 for each storage node.

   > Only one storage service can be installed on a host.

The storage is installed.

Creating an event router

> This functionality is only available in the commercial version of LogAnalyzer.

An event router is a service that allows you to receive streams of events from collectors and then distribute the events to specified destinations in accordance with the configured filters.

To have events from the collector sent to the event router, you must create an 'eventRouter' destination resource with the address of the event router and link the resource to the collectors that you want to send events to the event router.

The event router receives events on the API port, just like 'storage' and 'correlator' destinations.

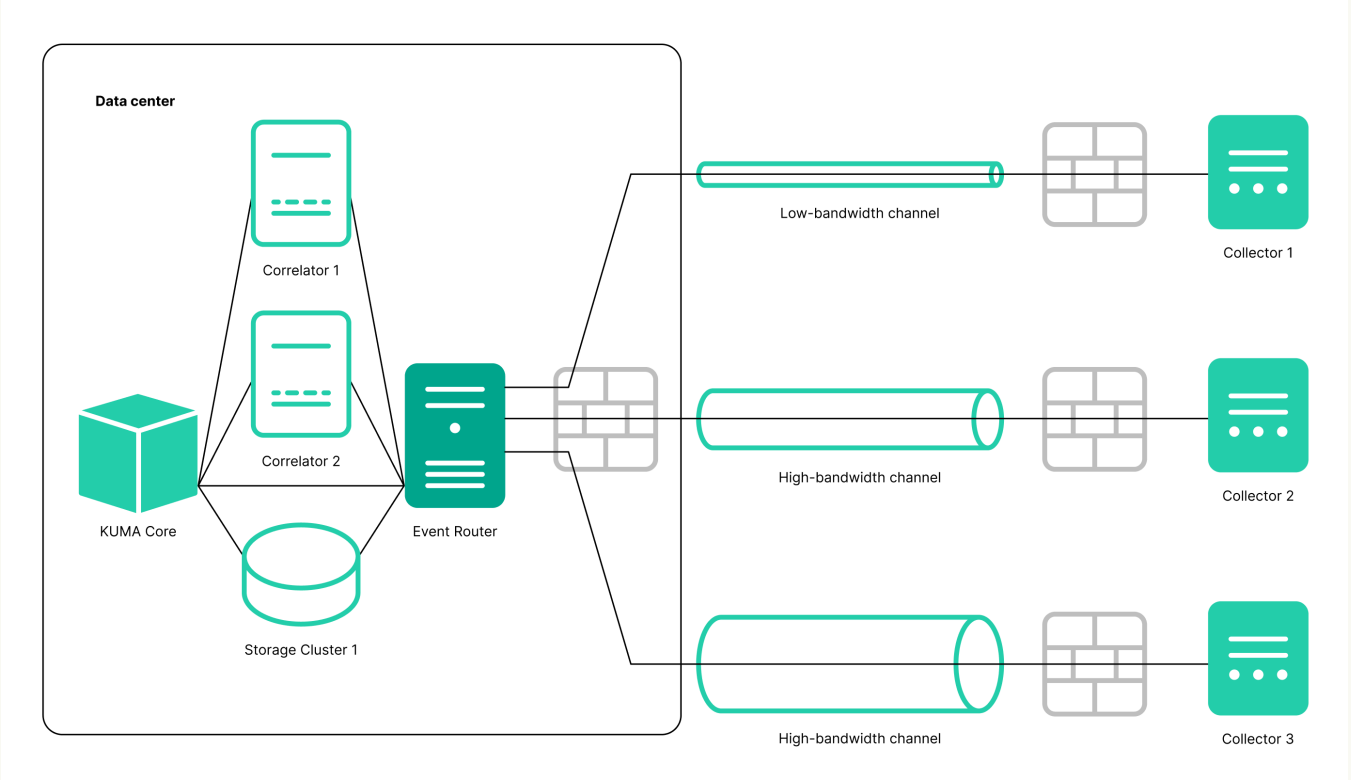
You can create a router in the **Resources** section.

Using an event router lets you reduce the utilization of links, which is important for low-bandwidth and busy links.

Possible use cases:

Collector — Router in the data center

The collector sends events to an event router in the data center, and the event router sends the events to the specified storage destination.



Preconditions:

- KUMA 3.2 collectors are configured at the branch offices.
- The data center has the capacity to install an event router.
- KUMA 3.2 is installed in the data center.

Steps:

1. In the data center:
   a. Create the **Event router** service.
   b. Create **storage** and **correlator** destination points and specify them in the Event router.
   c. In the Event router on the **Advanced settings** tab, configure a filter to send events to a storage. For example, "DeviceCustomString = correlator" or "DeviceCustomString = storage".
   d. Configure enrichment.
2. In the collectors at branch offices:
   a. Create a destination of the **eventRouter** type.
   b. Specify the URL of the event router in the data center of the branch office.
   c. If eventRouter replaces previously configured destinations, you can delete them.

Postcondition:

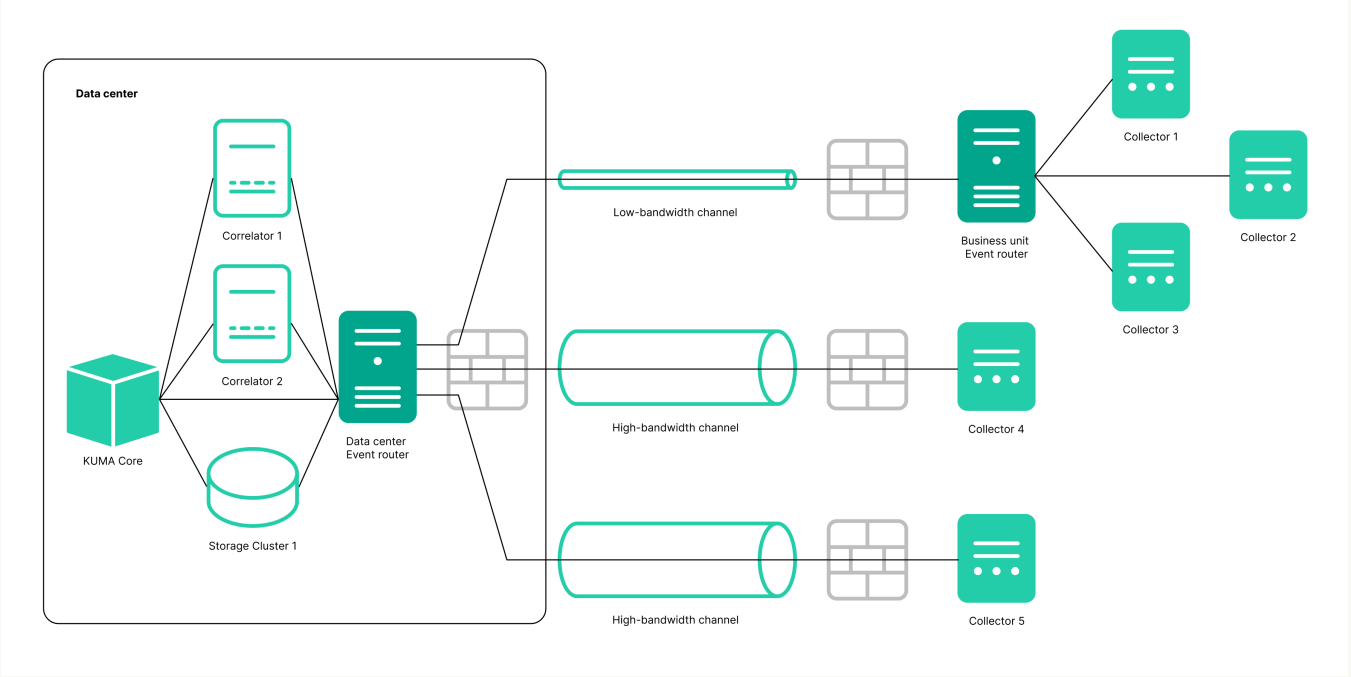
- Collectors at the branch offices are configured.
- The event router in the data center is configured.

Connections between branches and the data center have been optimized: you no longer need to configure the transmission of events to the storage at the data center in each collector.

Routing to the storage is performed within the data center.

[Cascade connection: Multiple collectors — Router at the branch office; Router at the branch office — Router in the data center]

Multiple collectors send events to the event router at the branch office, and the event router at the branch office sends events to the router in the data center, where events are then sent to the specified destinations, that is, correlators and storage.



Router cascade diagram

Preconditions:

- KUMA 3.2 collectors are configured at the branch offices.
- The data center has the capacity to install an event router.
- KUMA 3.2 is installed in the data center.

Steps:

1. In the data center:
   a. Create the **Event router** service.
   b. Create **storage** and **correlator** destination points and specify them in the Event router.
   c. In the Event router on the **Advanced settings** tab, configure a filter to send events to storage and/or correlator. For example, "DeviceCustomString = correlator" or "DeviceCustomString = storage".
2. At the branch office:
   a. Create the Event router service.
   b. Create a destination of the **eventRouter** type and specify the **URL** of the Event router in the data center.
3. In the collectors at branch offices:
   a. Create a destination of the **eventRouter** type and specify the **URL** of the Event router at the branch office.
   b. If eventRouter replaces previously configured destinations, you can delete them.

Postcondition:

1. Collectors at the branch offices are configured.
2. The event router in the data center and the event router at the branch office are configured.

The connections of branch offices with the data center are optimized: in each collector, you no longer need to configure events to be sent to the data center; it is enough to collect all events on the router and send it to the data center as one stream.

The event router must be installed on a Linux device. Only a user with the General Administrator role can create the service. You can create a service in any tenant; the tenant relation does not impose any restrictions.

You can install the event router service on a separate host or together with a collector. There is no separate section for the event router in the inventory file, so you can specify the host on which you want to install the event router in the `kuma_collector` section. Also, if you want to install the event router on a separate host, you need to specify this host in the inventory file in the `kuma_collector` section; in this way, the event router service will be upgraded as part of the next LogAnalyzer upgrade.

You can use the following metrics to get information about the service performance:

- IO
- Process
- OS

As with other resources, the following audit events are generated for the event router in LogAnalyzer:

- Resource was successfully added
- Resource was successfully updated
- Resource was successfully deleted

Installing an event router involves two steps:

- Create the event router service in the LogAnalyzer web interface using the Installation Wizard.
- Install the event router service on the server.

Starting the Event Router Installation Wizard

To start Event Router Installation Wizard:

1. In the LogAnalyzer web interface, in the **Resources** section, click **Event routers**.
2. This opens the **Event routers** window; in that window, click **Add**.

Follow the instructions of the installation wizard.

Step 1. General settings of the event router

This is a required step of the Installation Wizard. At this step, you specify the main settings of the event router: its name and the tenant that will own it.

To specify the general settings of the event router:

1. On the **Basic settings** tab, fill in the following fields:

   a. In the **Name** field, enter a unique name for the service you are creating. The name must contain 1 to 128 Unicode characters.

   b. In the **Tenant** drop-down list, select the tenant that will own the event router. An event router belonging to a tenant is organizational in nature and does not impose any restrictions.

   c. If necessary, specify the number of processes that the service can run concurrently in the **Handlers** field. By default, the number of handlers is the same as the number of vCPUs on the server where the service is installed.

   d. You can optionally add up to 4000 Unicode characters describing the service in the **Description** field.

2. On the **Advanced settings** tab, fill in the following fields:

   a. If necessary, use the **Debug** toggle switch to enable logging of service operations.

   b. You can use the **Create dump periodically** toggle switch at the request of Technical Support to generate resource (CPU, RAM, etc.) utilization reports in the form of dumps.

   c. In the **Dump settings** field, you can specify the settings to be used when creating dumps. For instructions on filling out this field, please contact Technical Support.

General settings of the event router are specified. Proceed to the next step of the Installation Wizard.

Step 2. Routing

This is a required step of the Installation Wizard. We recommend sending events to the storage destination. You can also select another event router as the destination.

To specify the settings of the destination to which you want the event router to send events received from collectors:

1. In the **Routing** step of the Installation Wizard, click **Add**.

2. This opens the **Create destination** window; in that window, specify the following settings:

   a. On the **Basic settings** tab, in the **Name** field, enter a unique name for the destination. The name must contain 1 to 128 Unicode characters.

   b. You can use the **State** toggle switch to enable or disable the service as needed.

   c. In the **Type** drop-down list, select the type of the destination. The following values are available:

      - nats-jetstream
      - tcp
      - http
      - kafka
      - file
      - storage
      - correlator
      - eventRouter

   d. On the **Advanced settings** tab, specify the values of parameters. The set of parameters that can be configured depends on the type of the destination selected on the **Basic settings** tab. For detailed information about parameters and their values, click the link for each type of destination in paragraph "c." of these instructions.

The created destination is displayed on the Installation Wizard tab. A destination resource can be removed from the resource set by selecting it and clicking **Delete** in the opened window.

Routing is configured. You can proceed to the next step of the Installation Wizard.

Step 3. Setup validation

This is the required, final step of the Installation Wizard.

To create an event router in the Installation Wizard:

1. Click **Create and save service**.

   The lower part of the window displays the command that you must use to install the router on the server.

   Example command:

   ```
   /opt/kaspersky/kuma/kuma eventrouter --core https://kuma-example:<port used for communication with the KUMA Core> --id <event router service ID> --api.port <port used for communication with the service> --install
   ```

   The port for communication with the LogAnalyzer Core, the service ID, and the port for communication with the service are added to the command automatically. You must also ensure the network connectivity of LogAnalyzer and open the ports used by application components, if necessary.

2. Close the Wizard by clicking **Save**.

The service is installed in the LogAnalyzer web interface. You can now proceed with installing the service in the LogAnalyzer network infrastructure.

Installing the event router on the server

To install the event router on the server:

1. Log in to the server where you want to install the event router service.

   If you want to install the event router service on a separate server, specify the host in the inventory file in the `kuma_correlator` group and install the router service on that host. When upgrading LogAnalyzer, the installer upgrades the event router installed on the host specified in the inventory file.

2. Create the /opt/kaspersky/kuma/ directory.

3. Copy the "kuma" file to the "/opt/kaspersky/kuma/" directory. The file is located inside the installer in the "/kuma-ansible-installer/roles/kuma/files/" directory.

4. Make sure the kuma file has sufficient rights to run. If the file is not executable, make it executable:

   ```
   sudo chmod +x /opt/kaspersky/kuma/kuma
   ```

5. Place the LICENSE file from the /kuma-ansible-installer/roles/kuma/files/ directory in the /opt/kaspersky/kuma/ directory and accept the terms and conditions of the End User License Agreement by running the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma license
   ```

6. Create the 'kuma' user:

   ```
   sudo useradd --system kuma && usermod -s /usr/bin/false kuma
   ```

7. Make the 'kuma' user the owner of the /opt/kaspersky/kuma directory and all files inside the directory:

   ```
   sudo chown -R kuma:kuma /opt/kaspersky/kuma/
   ```

8. Add the LogAnalyzer event router port to firewall exclusions.

   For the application to run correctly, ensure that the LogAnalyzer components are able to interact with other components and applications over the network using the protocols and ports specified during the installation of the LogAnalyzer components.

9. Execute the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma eventrouter --core https://<FQDN of the KUMA Core server>:<port used by KUMA Core server for internal communication (port 7210 by default)> --id <service ID copied from the KUMA web interface> --api.port <port used for communication with the installed component> --install
   ```

   Example:

   ```
   sudo /opt/kaspersky/kuma/kuma eventrouter --core https://kuma.example.com:7210 --id XXXX --api.port YYYY --install
   ```

The event router is installed on the server. You can use it to receive events from collectors and relay the events to specified destinations.

Creating a collector

A collector receives raw events from event sources, performs normalization, and sends processed events to their destinations. The maximum size of an event that can be processed by the KUMA collector is 4 MB.

If you are using the SMB license, and both the hourly average EPS and the daily average EPS allowed by the license is exceeded for a collector, the collector stops receiving events and is displayed with a red status and a notification about the EPS limit being exceeded. The user with the General Administrator role gets a notification about the EPS limit being exceeded and the collector being stopped. Every hour, the hourly average EPS value is recalculated and compared with the EPS limit in the license. If the hourly average is under the limit, the restrictions on the collector are lifted, and the collector resumes receiving and processing events.

Installing a collector involves two steps:

- Create the collector in the LogAnalyzer web interface using the Installation Wizard. In this step, you specify the general collector settings to be applied when installing the collector on the server.
- Install the collector on the network infrastructure server on which you want to receive events.

Actions in the LogAnalyzer web interface

The creation of a collector in the LogAnalyzer web interface is carried out by using the Installation Wizard. This Wizard combines the required resources into a resource set for a collector. Upon completion of the Wizard, the service itself is automatically created based on this resource set.

To create a collector in the LogAnalyzer web interface,

Start the Collector Installation Wizard:

- In the LogAnalyzer web interface, in the **Resources** section, click the **Add event source** button.
- In the LogAnalyzer web interface in the **Resources** → **Collectors** section click the **Add collector** button.

As a result of completing the steps of the Wizard, a collector service is created in the LogAnalyzer web interface.

A resource set for a collector includes the following resources:

- Connector
- Normalizer (at least one)
- Filters (if required)
- Aggregation rules (if required)
- Enrichment rules (if required)
- Destinations (normally one for sending events to a storage)

These resources can be prepared in advance, or you can create them while the Installation Wizard is running.

Actions on the LogAnalyzer Collector Server

When installing the collector on the server that you intend to use for receiving events, run the command displayed at the last step of the Installation Wizard. As part of the installation process, you must specify the ID automatically assigned to the service in the LogAnalyzer web interface, as well as the port used for communication.

Testing the installation

After creating a collector, you are advised to make sure that it is working correctly.

Starting the Collector Installation Wizard

A collector consists of two parts: one part is created inside the LogAnalyzer web interface, and the other part is installed on the network infrastructure server intended for receiving events. The Installation Wizard creates the first part of the collector.

To start the Collector Installation Wizard:

- In the LogAnalyzer web interface, in the **Resources** section, click **Add event source**.
- In the LogAnalyzer web interface in the **Resources** → **Collectors** section, click **Add collector**.

Follow the instructions of the Wizard.

Aside from the first and last steps of the Wizard, the steps of the Wizard can be performed in any order. You can switch between steps by using the **Next** and **Previous** buttons, as well as by clicking the names of the steps in the left side of the window.

After the wizard finishes, a resource set for a collector is created in the LogAnalyzer web interface under **Resources** → **Collectors**, and a collector service is added under **Resources** → **Active services**.

Step 1. Connect event sources

This is a required step of the Installation Wizard. At this step, you specify the main settings of the collector: its name and the tenant that will own it.

To specify the general settings of the collector:

1. On the **Basic settings** tab, fill in the following fields:

   a. In the **Collector name** field, enter a unique name for the service you are creating. The name must contain 1 to 128 Unicode characters.

   > When certain types of collectors are created, agents named `agent: <Collector name>, auto created` are also automatically created together with the collectors. If this type of agent was previously created and has not been deleted, you cannot create a collector named `<Collector name>`. If this is the case, you will have to either specify a different name for the collector or delete the previously created agent.

   b. In the **Tenant** drop-down list, select the tenant that will own the collector. The tenant selection determines what resources will be available when the collector is created.

   > If you return to this window from another subsequent step of the Installation Wizard and select another tenant, you will have to manually edit all the resources that you have added to the service. Only resources from the selected tenant and the Shared tenant can be added to the service.

   c. In the **Workers** field, if necessary, specify the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.

   d. You can optionally add up to 256 Unicode characters describing the service in the **Description** field.

2. On the **Advanced settings** tab, fill in the following fields:

   a. If necessary, use the **Debug** toggle switch to enable logging of service operations. Error messages of the collector service are logged even when debug mode is disabled. The log can be viewed on the machine where the collector is installed, in the /opt/kaspersky/kuma/collector/<collector ID>/log/collector directory.

   b. You can use the **Create dump periodically** toggle switch at the request of Technical Support to generate resource (CPU, RAM, etc.) utilization reports in the form of dumps.

   c. In the **Dump settings** field, you can specify the settings to be used when creating dumps. For instructions on filling out this field, please contact Technical Support.

General settings of the collector are specified. Proceed to the next step of the Installation Wizard.

Step 2. Transport

This is a required step of the Installation Wizard. On the **Transport** tab of the Installation Wizard, select or create a connector and in its settings, specify the source of events for the collector service.

To add an existing connector to a resource set,

select the name of the required connector from the **Connector** drop-down list.

The **Transport** tab of the Installation Wizard displays the settings of the selected connector. You can open the selected connector for editing in a new browser tab using the button.

To create a new connector:

1. Select **Create new** from the **Connector** drop-down list.

2. In the **Type** drop-down list, select the connector type and specify its settings on the **Basic settings** and **Advanced settings** tabs. The available settings depend on the selected type of connector:

   - tcp
   - udp
   - nats-jetstream
   - kafka
   - http
   - sql
   - file
   - 1c-log
   - 1c-xml
   - diode
   - ftp
   - nfs
   - wmi
   - wec
   - etw
   - snmp
   - snmp-trap
   - kata\edr
   - vmware
   - elastic

   When using the **tcp** or **udp** connector type at the normalization stage, IP addresses of the assets from which the events were received will be written in the DeviceAddress event field if it is empty.

   When using a **wmi**, **wec**, or **etw** connector, agents are automatically created for receiving Windows events.

   It is recommended to use the default encoding (UTF-8), and to apply other settings only if bit characters are received in the fields of events.

To configure LogAnalyzer collectors to listen on ports below 1000, the service of the relevant collector must be started with root privileges. To do this, after installing the collector, add the line `AmbientCapabilities = CAP_NET_BIND_SERVICE` to its systemd configuration file in the [Service] section.
The systemd file is located in the /usr/lib/systemd/system/kuma-collector-<`collector ID`>.service directory.

The connector is added to the resource set of the collector. The created connector is only available in this resource set and is not displayed in the web interface **Resources** → **Connectors section**.

Proceed to the next step of the Installation Wizard.

Step 3. Event parsing

This is a required step of the Installation Wizard. On the **Event parsing** tab of the Installation Wizard, click the **+ Add event parsing** button to open the **Basic event parsing** window, and in that window, in the **Normalizer** drop-down list, select or create a normalizer. In normalizer settings, define the rules for converting raw events into normalized events. You can add multiple event parsing rules to the normalizer to implement complex event processing logic. You can test the normalizer using test events.

When you create a new normalizer in the Installation Wizard, the default normalizer is saved in the resource set for the collector. You cannot use the created normalizer in other collectors. You can select the **Save normalizer** check box to create the normalizer as a separate resource. This will make the normalizer available for selection in other collectors of the tenant.

Adding a normalizer to a resource set

To add a normalizer to a resource set:

1. Click the **+ Add event parsing** button.

   This opens the **Basic event parsing** window with the normalizer settings. By default, the **Normalization scheme** tab is selected.

2. In the **Normalizer** drop-down list, select a normalizer. You can select a normalizer that belongs to the tenant of the collector or to the Shared tenant.

   The normalizer settings are displayed.

   If you want to edit the settings of the normalizer, in the **Normalizer** drop-down list, click the pencil icon next to the name of the normalizer to open the **Edit normalizer** window, and in that window, click the dark circle. This opens the **Basic event parsing** window with the normalizer settings. If you want to edit additional parsing settings, in the **Editing normalizer** window, hover over the dark circle and click the plus symbol that appears. This opens the **Additional event parsing** with settings of the additional parsing. For more details on configuring the additional parsing of events, see the **Creating the structure of event normalization rules** section below.

3. Click **OK**.

The normalizer is added and displayed as a dark circle on the **Event parsing** tab of the Installation Wizard. You can click the dark circle to view the settings of the normalizer.

To create a new normalizer in the collector:

1. Click the **+ Add event parsing** button.

   This opens the **Basic event parsing** window with the normalizer settings. By default, the **Normalization scheme** tab is selected.

2. If you want to create the normalizer as a separate resource, select the **Save normalizer** check box. If the check box is selected, the normalizer becomes available for selection in other collectors of the tenant. This check box is cleared by default.

3. In the **Name** field, enter a unique name for the normalizer. Maximum length of the name: 128 Unicode characters.

4. In the **Parsing method** drop-down list, select the type of events to receive. Depending on the selected value, you can use the predefined event field matching rules or define rules manually. When you select some parsing methods, additional settings may become available that you must specify.

   Available parsing methods:

   - **json**

     This parsing method is used to process JSON data where each object, including its nested objects, occupies a single line in a file.

     When processing files with hierarchically structured data, you can reference the fields of nested objects using the dot notation. For example, the `username` parameter from the string `"user": {"username": "system: node: example-01"}` can be accessed by using the `user.username` query.

     Files are processed line by line. Multi-line objects with nested structures may be normalized incorrectly.

     In complex normalization schemes where additional normalizers are used, all nested objects are processed at the first normalization level, except for cases when the extra normalization conditions are not specified and, therefore, the event being processed is passed to the extra normalizer in its entirety.

     You can use `\n` and `\r\n` as newline characters. Strings must be UTF-8 encoded.

     If you want to send the raw event for advanced normalization, at each nesting level in the **Advanced event parsing** window, in the **Use raw event** drop-down list, select **Yes**.

   - **cef**

     This parsing method is used to process CEF data.

     If you select this parsing method, you can use the predefined rules for converting events to the LogAnalyzer format by clicking **Apply default mapping**.

   - **regexp**

This parsing method is used to create custom rules for processing data in a format using regular expressions.

You must add a regular expression (RE2 syntax) with named capturing groups to the field under **Normalization**. The name of the capturing group and its value are considered the field and value of the raw event that can be converted to an event field in LogAnalyzer format.

To add event handling rules:

1. If necessary, copy an example of the data you want to process to the **Event examples** field. We recommend completing this step.

2. In the field under **Normalization**, add a regular expression with named capture groups following the RE2 syntax, for example `"(?P<name>regexp)"`. The regular expression added to the field under **Normalization** must exactly match the event. When designing the regular expression, we recommend using special characters that match the starting and ending positions of the text: `^`, `$`.

   You can add multiple regular expressions or remove regular expressions. To add a regular expression, click **Add regular expression**. To remove a regular expression, click the delete icon ✕ next to it.

3. Click the **Copy field names to the mapping table** button.

   Capture group names are displayed in the **KUMA field** column of the **Mapping** table. You can select the corresponding LogAnalyzer field in the column opposite each capturing group. If you followed the CEF format when naming the capturing groups, you can use automatic CEF mapping by selecting the **Use CEF syntax for normalization** check box.

Event handling rules are added.

- **syslog**

  This parsing method is used to process data in syslog format.

  If you select this parsing method, you can use the predefined rules for converting events to the LogAnalyzer format by clicking **Apply default mapping**.

  To parse events in rfc5424 format with a `structured-data` section, in the **Keep extra fields** drop-down list, select **Yes**. This makes the values from the `structured-data` section available in the `Extra` fields.

- **csv**

  This parsing method is used to create custom rules for processing CSV data.

  When choosing this parsing method, you must specify the separator of values in the string in the **Delimiter** field. Any single-byte ASCII character can be used as a delimiter for values in a string.

- **kv**

This parsing method is used to process data in key-value pair format. Available parsing method settings are listed in the table below.

Available parsing method settings

Setting	Description
Pair delimiter	The character used to separate key-value pairs. You can specify any single-character (1 byte) value. The specified value must not match the value specified in the **Value delimiter** field.
Value delimiter	The character used to separate a key from its value. You can specify any single-character (1 byte) value. The specified value must not match the value specified in the **Pair delimiter** field.

- xml

This parsing method is used to process XML data in which each object, including nested objects, occupies a single line in a file. Files are processed line by line.

If you want to send the raw event for advanced normalization, at each nesting level in the **Advanced event parsing** window, in the **Use raw event** drop-down list, select **Yes**.

If you select this parsing method, under **XML attributes**, you can specify the key XML attributes to be extracted from tags. If an XML structure has multiple XML attributes with different values in the same tag, you can identify the necessary value by specifying the key of the value in the **Source** column of the **Mapping** table.

To add key XML attributes:

1. Click **+ Add field**.

2. This opens a window; in that window, specify the path to the XML attribute.

You can add multiple XML attributes or remove XML attributes. To remove an individual XML attribute, click the delete icon ✕ next to it. To remove all XML attributes, click **Reset**.

If XML key attributes are not specified, then in the course of field mapping the unique path to the XML value will be represented by a sequence of tags.

Tag numbering

Starting with LogAnalyzer 2.1.3, you can use automatic tag numbering in XML events. This lets you parse an event with the identical tags or unnamed tags, such as `<Data>`.

As an example, we will number the tags of the `EventData` attribute of the **Microsoft Windows PowerShell event ID 800**.

```
<Event xmlns="http://schemas .microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Microsoft-Windows-ActiveDirectory_DomainService" Guid="
    {0e8478c5-3605-4e8c-8497-1e730c959516}" EventSourceName="NTDS" />
    <EventID Qualifiers="0000">0000</EventID>
    <Version>@</Version>
    <Level>4</Level>
    <Task>15</Task>
    <Opcode>0</Opcode>
    <Keywords >0x8080000000000000</Keywords>
    <TimeCreated SystemTime="2000-01-01T00:00:00.659495900Z" />
    <EventRecordID>55647</EventRecordID>
    <Correlation />
    <Execution ProcessID="1" ThreadID="1" />
    <Channel>service</Channel>
    <Computer>computer</Computer>
    <Security UserID="0000" />
  </System>
  <EventData>
    <Data>583</Data>
    <Data>36</Data>
    <Data>192.168.0.1:5084</Data>
    <Data>level</Data>
    <Data>name, lDAPDisplayName</Data>
    <Data />
    <Data>5545</Data>
    <Data>3</Data>
    <Data>0</Data>
    <Data>0</Data>
    <Data>0</Data>
    <Data>15</Data>
    <Data>none</Data>
  </EventData>
</Event>
```

To parse events with identical tags or unnamed tags, you need to configure tag numbering and data mapping for numbered tags with KUMA event fields.

LogAnalyzer 3.0.x supports using XML attributes and tag numbering at the same time in the same extra normalizer. If an XML attribute contains unnamed tags or identical tags, we recommend using tag numbering. If the XML attribute contains only named tags, we recommend using XML attributes.

To use XML attributes and tag numbering in extra normalizers, you must sequentially enable the **Use raw event** setting in each extra normalizer along the path that the event follows to the target extra normalizer, and in the target extra normalizer itself.

For an example of how tag numbering works, you can refer to the **MicrosoftProducts** normalizer. The **Use raw event** setting is enabled sequentially in both **AD FS** and **424** extra normalizers.

To set up the parsing of events with unnamed or identical tags:

1. Open an existing normalizer or create a new normalizer.

2. In the **Basic event parsing** window of the normalizer, in the **Parsing method** drop-down list, select **xml**.

3. In the **Tag numbering** field, click **+ Add field**.

4. In the displayed field, enter the full path to the tag to whose elements you want to assign a number, for example, `Event.EventData.Data`. The first tag gets number 0. If the tag is empty, for example, `<Data />`, it is also assigned a number.

5. To configure data mapping, under **Mapping**, click **+ Add row** and do the following:

   a. In the displayed row, in the **Source** field, enter the full path to the tag and the index of the tag. For example, for the Microsoft Windows PowerShell event ID 800 from the example above, the full paths to tags and tag indices are as follows:

      - `Event.EventData.Data.0`
      - `Event.EventData.Data.1`
      - `Event.EventData.Data.2` and so on.

   b. In the **KUMA field** drop-down list, select the field in the KUMA event that will receive the value from the numbered tag after parsing.

6. Save changes in one of the following ways:

   - If you created a new normalizer, click **Save**.
   - If you edited an existing normalizer, in the collector to which the normalizer is linked, click **Update configuration**.

Parsing is configured.

- **ipfix**

  This parsing method is used to process IPFIX data.

  If you select this parsing method, you can use the predefined rules for converting events to the LogAnalyzer format by clicking **Apply default mapping**. If the **ipfix** parsing method is selected for the main parsing, extra normalization is not available.

- **sql**

  The normalizer uses this parsing method to process data obtained by making a selection from the database.

  This parsing method becomes available only if at the Transport step of the Installation Wizard you specified a connector of the sql type.

- **normalizer**

This parsing method allows you to reuse a previously created normalizer as the primary or extra normalizer for processing events.

If you select this parsing method, in the **Normalizer** drop-down list, you need to select one of the previously created normalizers of the json, cef, regexp, syslog, csv, kv, xml, or sql type. Normalizers with the **normalizer** parsing method cannot be selected.

5. In the **Keep raw event** drop-down list, specify whether you want to store the original raw event in the re-created normalized event.

   - **Don't save**—do not save the raw event. This value is selected by default.
   - **Only errors**—save the raw event in the `Raw` field of the normalized event if errors occurred when parsing it. You can use this value to debug the service. In this case, an event having a non-empty `Raw` field will indicate problems.
   - **Always**—always save the raw event in the `Raw` field of the normalized event.

6. In the **Keep extra fields** drop-down list, choose whether you want to store the raw event fields in the normalized event if no mapping is configured for them (see step 8 of these instructions):

   - **No**. This value is selected by default.
   - **Yes**. The original event fields are saved in the `Extra` field of the normalized event.

   This field is not available if the **Parsing method** is set to **normalizer**.

7. If necessary, provide an example of the data you want to process to the **Event examples** field. We recommend completing this step.

   You can add multiple event examples. Every new example must be specified on a new line. At the same time, in the **Mapping** table, its parsing result will be displayed in the **Examples** column. This lets you check whether the fields of different types of incoming events are mapped correctly.

   Limitations involved in parsing event examples

   When parsing the examples of data specified in the **Event examples** field, the following parameters are not taken into account in the extra normalizer:

   - Enrichment configured in the root normalizer
   - Filters specified in the extra normalizer
   - The **Use raw event** setting
   - Conversions in the root normalizer
   - Conversions for the value of the **Field to pass into normalizer** parameter.

   This field is not available if the **Parsing method** is set to **normalizer**.

8. In the **Mapping** table, configure the mapping of raw event fields to LogAnalyzer event fields:

   a. Click **+ Add row**.

      You can add multiple table rows or delete table rows. To delete a table row, select the check box next to it and click the **Delete** button.

   b. In the **Source** column, specify the name of the raw event field that you want to map to the LogAnalyzer event field.

      For details about the field name format, refer to the Normalized event data model article. For a description of the mapping, refer to the Mapping fields of predefined normalizers article.

      If you want to create rules for modifying the fields of the original event before writing to the LogAnalyzer event fields, click the settings (⚙) icon next to the field name to open the **Conversion** window, and in that window, click **+ Add conversion**. You can reorder the rules or delete the rules. To reorder rules, use the reorder ⋮⋮ icons. To delete a rule, click the delete ✕ icon next to it.

      Available conversions

Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.

- **lower**—is used to make all characters of the value lowercase

- **upper**—is used to make all characters of the value uppercase

- **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.

- **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.

- **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:

  - **Replace chars** specifies the sequence of characters to be replaced.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, the new value is `soft-Windows-Sys`.

- **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:

  - **Expression** is the RE2 regular expression whose results you want to replace.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.
- Converting the IP address to the IPv4 format:
  - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.
  - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append**, **prepend**.

If at the Transport step of the Installation Wizard, you specified a connector of the file type, you can pass the name or path of the file being processed by the collector to the LogAnalyzer event field. To do this, in the **Source** column, specify one of the following values:

- `$kuma_fileSourceName` to pass the name of the file being processed by the collector in the LogAnalyzer event field.
- `$kuma_fileSourcePath` to pass the path to the file being processed by the collector in the LogAnalyzer event field.

When you use a **file** connector, the new variables in the normalizer will only work with destinations of the internal type.

c. In the **LogAnalyzer field** column, select a LogAnalyzer event field from the drop-down list. You can find the LogAnalyzer event field by typing its name. If the name of the LogAnalyzer event field begins with `DeviceCustom*` or `Flex*`, enter a unique custom label in the **Label** field, if necessary.

If you have loaded data into the **Event examples** field, the table will have an **Examples** column containing examples of values carried over from the raw event field to the LogAnalyzer event field.

The **Mapping** table is not available if the **Parsing method** is set to **normalizer**.

9. Click **OK**.

The normalizer is created and displayed as a dark circle on the **Event parsing** tab of the Installation Wizard. You can click the dark circle to view the settings of the normalizer. If you want to edit additional parsing settings, hover over the dark circle and click the plus symbol that appears. This opens the **Additional event parsing** with settings of the additional parsing. For more details on configuring the additional parsing of events, see the **Creating the structure of event normalization rules** section below.

When displaying a normalization scheme, the structure of nested normalizers is not displayed. To view this structure, you need to go to the normalizer card by clicking its name link. If you are using a nested normalizer with extra normalization, keep in mind that the structure of the output data may change. In that case, it is important to understand which field is passed to the other extra normalizer.

Enriching normalized events with additional data

You can create enrichment rules in the normalizer to add extra data to created normalized events. Enrichment rules are stored in the normalizer in which they were created.

The **Enrichment** tab is not available if on the **Normalization scheme** tab, the **Parsing method** is set to **normalizer**.

To add enrichment rules to the normalizer:

1. Select the main or additional normalization rule to open a window, and in that window, select the **Enrichment** tab.

2. Click the **+ Add enrichment** button.

   The enrichment rule settings section is displayed. You can add multiple enrichment rules or delete enrichment rules. To delete an enrichment rule, click the delete ✕ icon next to it.

3. In the **Source kind** drop-down list, select the type of the enrichment source. When you select some enrichment source types, additional settings may become available that you must specify.

   Available Enrichment rule source types:

   - **constant**

     This type of enrichment is used when a constant needs to be added to an event field. Available enrichment type settings are listed in the table below.

     Available enrichment type settings

Setting	Description
Constant	The value to be added to the event field. Maximum length of the value: 255 Unicode characters. If you leave this field blank, the existing event field value is removed.
Target field	The LogAnalyzer event field that you want to populate with the data.

     If you are using the event enrichment functions for extended schema fields of `String`, `Number`, or `Float` type with a constant, the constant is added to the field.

     If you are using the event enrichment functions for extended schema fields of `Array of strings`, `Array of numbers`, or `Array of floats` type with a constant, the constant is added to the elements of the array.

   - **dictionary**

     This type of enrichment is used if you need to add a value from the dictionary of the **Dictionary** type. Available enrichment type settings are listed in the table below.

     Available enrichment type settings

Setting	Description
Dictionary name	The dictionary from which the values are to be taken.
Key fields	Event fields whose values are to be used for selecting a dictionary entry. To add an event field, click **Add field**. You can add multiple event fields.

     If you are using event enrichment with the **dictionary** type selected as the **Source kind** setting, and an array field is specified in the **Key enrichment fields** setting, when an array is passed as the dictionary key, the array is serialized into a string in accordance with the rules of serializing a single value in the TSV format.

     Example: The **Key fields** setting of the **enrichment** uses the `SA.StringArrayOne` extended schema field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`. The following values are passed to the dictionary as the key: `['a','b','c']`.

     If the **Key enrichment fields** setting uses an extended schema array field and a regular event schema field, the field values are separated by the `|` character when the dictionary is queried.

     Example: The **Key enrichment fields** setting uses the `SA.StringArrayOne` extended schema field and the `Code` string field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`, and the `Code` string field contains the `myCode` sequence of characters. The following values are passed to the dictionary as the key: `['a','b','c']|myCode`.

   - **table**

This type of enrichment is used if you need to add a value from the dictionary of the **Table** type. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Dictionary name	The dictionary from which the values are to be taken.
Key fields	Event fields whose values are to be used for selecting a dictionary entry. To add an event field, click **Add field**. You can add multiple event fields.
Mapping	Event fields for data transfer: • **Dictionary field** specifies dictionary fields from which data is to be transmitted. The available fields depend on the selected dictionary resource. • **KUMA field** specifies event fields to which data is to be transmitted. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written there.

The first field in the table (**Dictionary field**) is taken as the key with which the fields selected from the event as key fields are matched (**KUMA field**). As the key in the **Dictionary field**, you must select an indicator of compromise by which the enrichment is to be performed, for example, IP address, URL, or hash. In the rule, you must select the event field that corresponds to the selected indicator of compromise in the dictionary field.

If you want to select multiple key fields, you can specify them using `|` as a separator (when specifying in the web interface or importing as a CSV file), for example, `<IP address>|<user name>`.

You can add new table rows or delete table rows. To add a new table row, click **Add new element**. To delete a row in the table, click the ✕ button.

- event

This type of enrichment is used when you need to write a value from another event field to the current event field. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Target field	The LogAnalyzer event field that you want to populate with the data.
Source field	The event field whose value is written to the target field.

Clicking opens the **Conversion** window, in which you can click **Add conversion** to create rules for modifying the source data before writing them to the LogAnalyzer event fields. You can reorder and delete created rules. To change the position of a rule, click next to it. To delete a rule, click next to it.

Available conversions

Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.

- **lower**—is used to make all characters of the value lowercase

- **upper**—is used to make all characters of the value uppercase

- **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.

- **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.

- **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:

  - **Replace chars** specifies the sequence of characters to be replaced.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, the new value is `soft-Windows-Sys`.

- **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:

  - **Expression** is the RE2 regular expression whose results you want to replace.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.
- Converting the IP address to the IPv4 format:
  - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.
  - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append**, **prepend**.

When using enrichment of events that have **eventevent** selected as the **Source kind** and the extended event schema fields are used as arguments, the following special considerations apply:

- If the source extended event schema field has the `Array of strings` type, and the target extended event schema field has the `String` type, the values are written to the target extended event schema field in TSV format.

  Example: the `SA.StringArray` extended event schema field contains values: `"string1"`, `"string2"`, `"string3"`. An event enrichment operation is performed. The result of the event enrichment operation is written to the `DeviceCustomString1` extended event schema field. The `DeviceCustomString1` extended event schema field contains values: `["string1", "string2", "string3"]`.
- If the source and target extended event schema fields have the `Array of strings` type, values of the source extended event schema field are added to the values of the target extended event

schema field, and the "," character is used as the delimiter.

Example: The `SA.StringArrayOne` field of the extended event scheme contains the `["string1", "string2", "string3"]` values, and the `SA.StringArrayTwo` field of the extended event scheme contains the `["string4", "string5", "string6"]` values. An event enrichment operation is performed. The result of the event enrichment operation is written to the `SA.StringArrayTwo` field of the extended event scheme. The `SA.StringArrayTwo` extended event schema field contains values: `["string4", "string5", "string6", "string1", "string2", "string3"]`.

- **template**

This type of enrichment is used when you need to write the result of processing Go templates into the event field. We recommend matching the value and the size of the field. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Template	The Go template. Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field from which the value must be passed to the script, for example, `{{.DestinationAddress}} attacked from {{.SourceAddress}}`.
Target field	The LogAnalyzer event field that you want to populate with the data.

If you are using enrichment of events that have **template** selected as the **Source kind**, and in which the target field has the `String` type, and the source field is an extended event schema field containing an array of strings, you can use one of the following examples for the template:

- `{{.SA.StringArrayOne}}`
- ```
  {{- range $index, $element := . SA.StringArrayOne -}}
  {{- if $index}}, {{end}}"{{$element}}"{{- end -}}
  ```

To convert the data in an array field in a template into the TSV format, use the `toString` function, for example:

```
template {{toString.SA.StringArray}}
```

4. In the **Target field** drop-down list, select the LogAnalyzer event field to which you want to write the data. You cannot select a value if in the **Source kind** drop-down list, you selected **table**.

5. If you want to enable details in the normalizer log, enable the **Debug** toggle switch. The toggle switch is turned off by default.

6. Click **OK**.

The event enrichment rules with additional data are added to the normalizer into the selected parsing rule.
```

Configuring parsing linked to IPv4 addresses

If at the Transport step of the Installation Wizard, you specified a connector of the udp, tcp, or http type, you can forward events from multiple IPv4 addresses from sources of different types to the same collector, and have the collector apply the specified normalizers. To do this, you need to specify several IPv4 addresses and select the normalizer that you want to apply to events coming from the specified IPv4 addresses. The following types of normalizers are available: **json**, **cef**, **regexp**, **syslog**, **csv**, **kv**, **xml**, **normalizer**.

If you select a connector type other than **udp**, **tcp**, or **http** in a collector with configured normalizers and linking to IPv4 addresses, the **Parsing settings** tab is no longer displayed, and only the first of the previously specified normalizers is specified on the **Event parsing** tab of the Installation Wizard. The **Parsing settings** tab is hidden immediately, and the changes are applied after the resource is saved. If you want to restore the previous settings, exit the Installation Wizard without saving.

For normalizers of the **Syslog** and **regexp** types, you can use of a chain of normalizers. In this case, you can specify additional normalization conditions depending on the value of the **DeviceProcessName** field. The difference from extra normalization is that you can specify shared normalizers.

To configure parsing with linking to IPv4 addresses:

1. Select the **Parsing settings** tab and click the **+ Event source** button.

   A group of parsing settings is displayed. You can add multiple parsings or delete parsings. To remove a parsing, click the delete ✕ icon next to it.

2. In the **IP address (-es)** field, enter the IPv4 address from which events will be sent. You can specify multiple IP addresses separated by commas. The length of the IPv4 address list is unlimited; however, we recommend limiting the number of IPv4 addresses to keep the load on the collector balanced. You must specify a value in this field if you want to apply multiple normalizers in one collector.

   > The IPv4 address must be unique for each normalizer. KUMA checks the uniqueness of IPv4 addresses, and if you specify the same IPv4 address for different normalizers, the `Field must be unique` message is displayed.

   If you want to send all events to the same normalizer without specifying IPv4 addresses, we recommend creating a separate collector. To improve performance, we recommend creating a separate collector with one normalizer if you want to apply the same normalizer to events from a large number of IPv4 addresses.

3. In the **Normalizer** drop-down list, create or select a normalizer. You can click the arrow icon next to the drop-down list to select the **Parsing flowcharts** tab.

   Normalization will be triggered if at the Transport step of the Installation Wizard, you specified a connector of the udp, tcp, or http type. For a **http** connector, you must specify the header of the event source. Taking into account the available connectors, the following normalizer types are available for automatic source recognition: **json**, **cef**, **regexp**, **syslog**, **csv**, **kv**, and **xml**.

4. If you selected a normalizer of the **Syslog** or **regexp** type, and you want to add a conditional normalization, click the **+ Add conditional normalization** button. Conditional normalization is available if in the **Mapping** table of the main normalizer, you configured the mapping of the source event field to the **DeviceProcessName** LogAnalyzer event field. Under **Condition**, in the **DeviceProcessName** field, specify a process name, and in the drop-down list, create or select a normalizer. You can specify multiple combinations of the **DeviceProcessName** LogAnalyzer event field and a normalizer. Normalization is performed until the first match.

Parsing with linking to IPv4 addresses is configured.

Creating a structure of event normalization rules

To implement complex event processing logic, you can add multiple event parsing rules to the normalizer. Events are switched between the parsing rules depending on the specified conditions. Events are processed sequentially in accordance with the creation order of the parsing rules. The event processing path is displayed as arrows.

To create an additional parsing rule:

1. Create a normalizer. For more information on creating normalizers, see the **Adding a normalizer to a resource set** section earlier in this article.

   The created normalizer is displayed as a dark circle on the **Event parsing** tab of the Installation Wizard.

2. Hover over the dark circle and click the plus sign that appears.

3. In the **Additional event parsing** window that opens, configure the additional event parsing rule:

   - **Extra normalization conditions** tab:

     If you want to send a raw event for extra normalization, in the **Use raw event** drop-down list, select **Yes**. The default value is **No**. We recommend passing a raw event to normalizers of **json** and **xml** types. If you want to send a raw event for extra normalization to the second, third, etc nesting levels, at each nesting level, in the **Use raw event** drop-down list, select **Yes**.

     To send only the events with a specific field to the additional normalizer, specify the field in the **Field to pass into normalizer** field.

     On this tab, you can define other conditions that must be satisfied for the event to be sent for additional parsing.

   - **Normalization scheme** tab:

     On this tab, you can configure event processing rules, similar to the main normalizer settings. The **Keep raw event** setting is not available. The **Event examples** field displays the values specified when the normalizer was created.

   - **Enrichment** tab:

     On this tab, you can configure enrichment rules for events. For more details on configuring enrichment rules, see the **Enriching normalized events with additional data** section earlier in this article. This tab is not available if on the **Normalization scheme** tab, the **Parsing method** is set to **normalizer**.

4. Click **OK**.

An additional parsing rule is added to the normalizer and displayed as a dark box. The dark box specifies the conditions that trigger the additional parsing rule.

You can do the following:

- You can chick the additional parsing rule to edit its settings.
- You can find an additional parsing rule by entering its name in the field in the upper part of the window.
- Create a new additional parsing rule. To do this, hover over the additional parsing rule and click on the plus icon that appears.
- Duplicate an additional parsing rule. To do this, hover over the additional parsing rule and click on the duplicate icon.

- Delete an additional parsing rule. To do this, hover over the additional parsing rule and click on the trash can icon.

- Change the position and order of additional normalizers. To do this, select the check box next to the extra normalizer and in the toolbar, click **Move**. This opens the **Move normalizer** window; in that window, select the required values in the **New parent** and **Place after** drop-down lists and click **Save**. If nothing is selected in the **Place after** drop-down list, the additional normalizer is moved to the beginning of the sequence of additional normalizers in the selected parent normalizer. In the lower part of the additional normalizer section, the number of additional normalizers associated with it is indicated.

  You can move multiple additional normalizers of the same level together. If you move an additional normalizer that has child normalizers, the whole chain is moved.

- Adjust the display of the structure of normalization rules in the toolbar using the **Collapse to default**, **Expand all**, **Compact view** buttons.

  By default, all additional normalizers at second level and below are collapsed. The main normalizer is the zero level, followed by the first level of extra normalizers, and the rest of the extra normalizers are collapsed. When you go to the normalization scheme, the scheme is displayed collapsed by default. For each item in the normalizer tree, you can collapse the child items using the button or expand them using the button. The buttons are displayed when you hover over the parent normalizer.

Proceed to the next step of the Installation Wizard.

Step 4. Filtering events

This is an optional step of the Installation Wizard. The **Event filtering** tab of the Installation Wizard allows you to select or create a filter whose settings specify the conditions for selecting events. You can add multiple filters to the collector. You can swap the filters by dragging them by the icon as well as delete them. Filters are combined by the AND operator.

When configuring filters, we recommend to adhere to the chosen normalization scheme. In filters, use only LogAnalyzer service fields and the fields that you specified in the normalizer in the **Mapping** and **Enrichment** sections. For example, if the DeviceAddress field is not used in normalization, avoid using the DeviceAddress field in a filter because such filtering will not work.

To add an existing filter to a collector resource set,

Click the **Add filter** button and select the required filter from the **Filter** drop-down menu.

To add a new filter to the collector resource set:

1. Click the **Add filter** button and select **Create new** from the **Filter** drop-down menu.

2. If you want to keep the filter as a separate resource, select the **Save filter** check box. This can be useful if you decide to reuse the same filter across different services. This check box is cleared by default.

3. If you selected the **Save filter** check box, enter a name for the created filter in the **Name** field. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** section, specify the conditions that must be met by the filtered events:

   - You can click the **Add condition** button to add filtering conditions. You can select two values (two operands, left and right) and assign the operation you want to perform with the selected values. The result of the operation is either True or False.

     - In the **op** drop-down list, select the function to be performed by the filter.

       In this drop-down list, you can select the **do not match case** check box if the operator should ignore the case of values. This check box is ignored if the **InSubnet**, **InActiveList**, **InCategory**, **InActiveDirectoryGroup** operators are selected. This check box is cleared by default.

Filter operators

- **=** means the left operand equals the right operand.
- **<** means the left operand is less than the right operand.
- **<=** means the left operand is less than or equal to the right operand.
- **>** means the left operand is greater than the right operand.
- **>=** means the left operand is greater than or equal to the right operand.
- **inSubnet** means the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains** means the left operand contains values of the right operand.
- **startsWith** means the left operand starts with one of the values of the right operand.
- **endsWith** means the left operand ends with one of the values of the right operand.
- **match** means the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit** checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
- **hasVulnerability** checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- The **inActiveList** operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary** checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory** means the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup** means the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- The **TIDetect** operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage.

- **inContextTable** checks for the presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

- In the **Left operand** and **Right operand** drop-down lists, select where the data to be filtered will come from. As a result of the selection, advanced settings are displayed, which you can use to precisely define the value to be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.
- You can use the **If** drop-down list to choose whether you need to create a negative filter condition.

Conditions can be deleted using the × button.

- The **Add group** button is used to add groups of conditions. The **AND** can be switched between **And**, **OR**, **NOT**.

  A condition group can be deleted using the × button.

- By clicking **Add filter**, you can add existing filters selected in the **Select filter** drop-down list to the conditions. You can click ↗ to navigate to a nested filter.

  A nested filter can be deleted using the × button.

The filter has been added.

Proceed to the next step of the Installation Wizard.

Step 5. Event aggregation

This is an optional step of the Installation Wizard. The **Event aggregation** tab of the Installation Wizard allows you to select or create aggregation rules whose settings specify the conditions for aggregating events of the same type. You can add multiple aggregation rules to the collector.

To add an existing aggregation rule to a set of collector resources,

click **Add aggregation rule** and select **Aggregation rule** in the drop-down list.

To add a new aggregation rule to a set of collector resources:

1. Click the **Add aggregation rule** button and select **Create new** from the **Aggregation rule** drop-down menu.
2. Enter the name of the newly created aggregation rule in the **Name** field. The name must contain 1 to 128 Unicode characters.
3. In the **Threshold** field, specify how many events must be accumulated before the aggregation rule triggers and the events are aggregated. The default value is `100`.
4. In the **Triggered rule lifetime** field, specify how long (in seconds) the collector must accumulate events to be aggregated. When this time expires, the aggregation rule is triggered and a new aggregation event is created. The default value is `60`.

5. In the **Identical fields** section, use the **Add field** button to select the fields that will be used to identify the same types of events. You can select the following service fields: SourceAssetID, DestinationAssetID, DeviceAssetID, SourceAccountID, DestinationAccountID. Asset IDs are added after event normalization, which means you can use them to aggregate events. The rest of the service fields are not used in aggregation. Selected events can be deleted using the buttons with a cross icon.

6. In the **Unique fields** section, you can click **Add field** to select the fields that will disqualify events from aggregation even if the events contain fields listed in the **Identical fields** section. Selected events can be deleted using the buttons with a cross icon.

7. In the **Sum fields** section, you can use the **Add field** button to select the fields whose values will be summed during the aggregation process. Selected events can be deleted using the buttons with a cross icon.

8. In the **Filter** section, you can specify the conditions to define events that will be processed by this resource. You can select an existing filter from the drop-down list or **create** a new filter.

   Creating a filter in resources

To create a filter:

1. In the **Filter** drop-down list, select **Create new**.

2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.

3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

4. Under **Conditions**, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, fields of additional parameters for identifying the value to be passed to the filter may be displayed. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

   c. In the **Operator** drop-down list, select an operator.

Filter operators

- **=** means the left operand equals the right operand.
- **<** means the left operand is less than the right operand.
- **<=** means the left operand is less than or equal to the right operand.
- **>** means the left operand is greater than the right operand.
- **>=** means the left operand is greater than or equal to the right operand.
- **inSubnet** means the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains** means the left operand contains values of the right operand.
- **startsWith** means the left operand starts with one of the values of the right operand.
- **endsWith** means the left operand ends with one of the values of the right operand.
- **match** means the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit** checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
- **hasVulnerability** checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- The **inActiveList** operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary** checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory** means the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup** means the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- The **TIDetect** operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage.

- **inContextTable** checks for the presence of the entry in the specified context table.
- **intersect** checks for the presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. This check box does not apply to the **InSubnet**, **InActiveList**, **InCategory**, or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, from the **If** drop-down list, select **If not**.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Aggregation rule added. You can delete it using the button.

Proceed to the next step of the Installation Wizard.

Step 6. Event enrichment

This is an optional step of the Installation Wizard. On the **Event enrichment** tab of the Installation Wizard, you can specify which data from which sources should be added to events processed by the collector. Events can be enriched with data obtained using enrichment rules.

Incoming events are first enriched with data from the sources that you select at this step of the wizard, and then the events are automatically enriched with asset information. For example, if you have dns enrichment configured, which leads to the IP address in the event being enriched with the corresponding DNS name of an asset, the event is enriched with this asset.

Rule-based enrichment

There can be more than one enrichment rule. You can add them by clicking the **Add enrichment** button and can remove them by clicking the button. You can use existing enrichment rules or create rules directly in the Installation Wizard.

To add an existing enrichment rule to a set of resources:

1. Click **Add enrichment**.

   This opens the enrichment rules settings block.

2. In the **Enrichment rule** drop-down list, select the relevant resource.

The enrichment rule is added to the resource set for the collector.

To create a new enrichment rule in a resource set:

1. Click **Add enrichment**.

   This opens the enrichment rules settings block.

2. In the **Enrichment rule** drop-down list, select **Create new**.

3. In the **Source kind** drop-down list, select the source of data for enrichment and define its corresponding settings:

- constant

  This type of enrichment is used when a constant needs to be added to an event field. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Constant	The value to be added to the event field. Maximum length of the value: 255 Unicode characters. If you leave this field blank, the existing event field value is removed.
Target field	The LogAnalyzer event field that you want to populate with the data.

  If you are using the event enrichment functions for extended schema fields of `String`, `Number`, or `Float` type with a constant, the constant is added to the field.

  If you are using the event enrichment functions for extended schema fields of `Array of strings`, `Array of numbers`, or `Array of floats` type with a constant, the constant is added to the elements of the array.

- dictionary (available only in the commercial version of LogAnalyzer)

  This type of enrichment is used if you need to add a value from the dictionary of the **Dictionary** type. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Dictionary name	The dictionary from which the values are to be taken.
Key fields	Event fields whose values are to be used for selecting a dictionary entry. To add an event field, click **Add field**. You can add multiple event fields.

  If you are using event enrichment with the **dictionary** type selected as the **Source kind** setting, and an array field is specified in the **Key enrichment fields** setting, when an array is passed as the dictionary key, the array is serialized into a string in accordance with the rules of serializing a single value in the TSV format.

  Example: The **Key fields** setting of the **enrichment** uses the `SA.StringArrayOne` extended schema field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`. The following values are passed to the dictionary as the key: `['a','b','c']`.

  If the **Key enrichment fields** setting uses an extended schema array field and a regular event schema field, the field values are separated by the `|` character when the dictionary is queried.

  Example: The **Key enrichment fields** setting uses the `SA.StringArrayOne` extended schema field and the `Code` string field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`, and the `Code` string field contains the `myCode` sequence of characters. The following values are passed to the dictionary as the key: `['a','b','c']|myCode`.

- event

This type of enrichment is used when you need to write a value from another event field to the current event field. Settings of this type of enrichment:

- In the **Target field** drop-down list, select the LogAnalyzer event field to which you want to write the data.
- In the **Source field** drop-down list, select the event field whose value will be written to the target field.
- Under **Conversion**, you can create rules for modifying the original data before it is written to the LogAnalyzer event fields. The conversion type can be selected from the drop-down list. You can use the **Add conversion** and **Delete** buttons to add or delete a conversion, respectively. The order of conversions is important.

  Available conversions

Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.

- **lower**—is used to make all characters of the value lowercase

- **upper**—is used to make all characters of the value uppercase

- **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.

- **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.

- **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:

  - **Replace chars** specifies the sequence of characters to be replaced.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, the new value is `soft-Windows-Sys`.

- **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:

  - **Expression** is the RE2 regular expression whose results you want to replace.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.
- Converting the IP address to the IPv4 format:
  - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.
  - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append**, **prepend**.

- template

This type of enrichment is used when you need to write the result of processing Go templates into the event field. We recommend matching the value and the size of the field. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Template	The Go template. Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field from which the value must be passed to the script, for example, `{{.DestinationAddress}} attacked from {{.SourceAddress}}`.
Target field	The LogAnalyzer event field that you want to populate with the data.

If you are using enrichment of events that have **template** selected as the **Source kind**, and in which the target field has the `String` type, and the source field is an extended event schema field containing an array of strings, you can use one of the following examples for the template:

- `{{.SA.StringArrayOne}}`
- ```
  {{- range $index, $element := . SA.StringArrayOne -}}
  {{- if $index}}, {{end}}"{{$element}}"{{- end -}}
  ```

To convert the data in an array field in a template into the TSV format, use the `toString` function, for example:

```
template {{toString.SA.StringArray}}
```

- dns (available only in the commercial version of LogAnalyzer)
```

This type of enrichment is used to send requests to a private network DNS server to convert IP addresses into domain names or vice versa. IP addresses are converted to DNS names only for private addresses: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 100.64.0.0/10.

Available settings:

- **URL**—in this field, you can specify the URL of a DNS server to which you want to send requests. You can use the **Add URL** button to specify multiple URLs.
- **RPS**—maximum number of requests sent to the server per second. The default value is `1,000`.
- **Workers** is the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.
- **Max tasks**—maximum number of simultaneously fulfilled requests. By default, this value is equal to the number of vCPUs of the LogAnalyzer Core server.
- **Cache TTL**—the lifetime (in seconds) of the values stored in the cache. The default value is `60` (seconds).
- **Cache disabled**—you can use this toggle switch to enable or disable caching. Caching is enabled by default.
- **Cache update disabled**—if cache update is enabled, then after the cache TTL expires, the collector sends repeat queries to the DNS server to update the information stored in the cache. This setting is available only when cache is enabled. By default, cache update is enabled.
- The **Recursive queries** setting is available starting with LogAnalyzer 3.4.1. You can use this toggle switch to make a LogAnalyzer collector send recursive queries to authoritative DNS servers for the purposes of enrichment. The default value is **Disabled**.

- **cybertrace** (available only in the commercial version of LogAnalyzer)

  For systems with a high load on the collector, we recommend using the cybertrace-http enrichment type. This type returns a structured response in JSON format and correctly distinguishes between strings and arrays of custom attributes, preventing parsing errors.

  When enriched through cybertrace, the collector gets an unstructured response. If the value of a custom attribute in CyberTrace contains square brackets, for example, **[str1**, **str2]**, LogAnalyzer treats it as an array, not a string.

  This type of enrichment is used to add information from CyberTrace data streams to event fields.

  Available settings:

  - **URL** (required)—in this field, you can specify the URL of a CyberTrace server to which you want to send requests. The default CyberTrace port is 9999.
  - **Number of connections**—maximum number of connections to the CyberTrace server that can be simultaneously established by LogAnalyzer. By default, this value is equal to the number of vCPUs of the LogAnalyzer Core server.
  - **RPS**—maximum number of requests sent to the server per second. The default value is `1,000`.
  - **Timeout**—amount of time to wait for a response from the CyberTrace server, in seconds. The default value is `30`.
  - **Maximum number of events in the enrichment queue**—maximum number of events stored in the enrichment queue for re-sending. The default value is 1,000,000.
  - **Mapping** (required)—this section of settings contains the mapping table for mapping LogAnalyzer event fields to CyberTrace indicator types. The **LogAnalyzer field** column shows the names of LogAnalyzer event fields, and the **CyberTrace indicator** column shows the types of CyberTrace indicators.

    Available types of CyberTrace indicators:

    - **ip**
    - **url**
    - **hash**

    In the mapping table, you must provide at least one string. You can use the **Add row** button to add a string, and can use the ✕ button to remove a string.

- **cybertrace-http** (available only in the commercial version of LogAnalyzer)

This is a new streaming event enrichment type in CyberTrace that allows you to send a large number of events with a single request to the CyberTrace API. Recommended for systems with a lot of events. The performance of cybertrace-http is superior to that of the cybertrace type.

Limitations:

- The cybertrace-http enrichment type cannot be used for retroscan in KUMA.
- If the cybertrace-http enrichment type is being used, detections are not saved in CyberTrace history in the Detections window.

Available settings:

- **Name** (required) is a unique name for the enrichment rule. Up to 128 Unicode characters.
- **Tenant** (required) is the name of the tenant in which you plan to work with CyberTrace.
- **URL** (required)—in this field, you can specify the URL of a CyberTrace server to which you want to send requests and the port that CyberTrace API is using. The default port is 443.
- **Secret** (required) is a drop-down list in which you can select the secret which stores the credentials for the connection.
- **Timeout**—amount of time to wait for a response from the CyberTrace server, in seconds. The default value is `30`.
- **Key fields** (required) is the list of event fields used for enriching events with data from CyberTrace.
- **Maximum number of events in the enrichment queue**—maximum number of events stored in the enrichment queue for re-sending. The default value is 1,000,000. After reaching 1 million events received from the CyberTrace server, events stop being enriched until the number of received events is reduced to less than 500,000.
- **Major version of CyberTrace** selects the CyberTrace version. Possible values: **<5**, **≥ 5.0**. If version **≥ 5.0** is selected, requests are executed in the General tenant in CyberTrace and use the new version of the API provided by Kaspersky CyberTrace 5.0 and later. If you select **<5**, requests use the Kaspersky CyberTrace 4.x API. The default value is **<5**.
- The **Debug** toggle switch enables resource logging. The toggle switch is turned off by default.
- **Tags** for resource search.

- **timezone**

This type of enrichment is used in collectors to assign a specific timezone to an event. Timezone information may be useful when searching for events that occurred at unusual times, such as nighttime.

When this type of enrichment is selected, the required timezone must be selected from the **Timezone** drop-down list.

Make sure that the required time zone is set on the server hosting the enrichment-utilizing service. For example, you can do this by using the `timedatectl list-timezones` command, which shows all time zones that are set on the server. For more details on setting time zones, please refer to your operating system documentation.

When an event is enriched, the time offset of the selected timezone relative to Coordinated Universal Time (UTC) is written to the DeviceTimeZone event field in the `+-hh:mm` format. For example, if you select the **Asia/Yekaterinburg** timezone, the value `+05:00` will be written to the DeviceTimeZone field. If the enriched event already has a value in the DeviceTimeZone field, it will be overwritten.

By default, if the timezone is not specified in the event being processed and enrichment rules by timezone are not configured, the event is assigned the timezone of the server hosting the service (collector) that processes the event. If the server time is changed, the service must be restarted.

Permissible time formats when enriching the DeviceTimeZone field

When processing incoming raw events in the collector, the following time formats can be automatically converted to the +-hh:mm format:

Time format in a processed event	Example
+-hh:mm	-07:00
+-hhmm	-0700
+-hh	-07

If the date format in the `DeviceTimeZone` field differs from the formats listed above, the collector server timezone is written to the field when an event is enriched with timezone information. You can create custom normalization rules for non-standard time formats.

- **geographic data**

This type of enrichment is used to add IP address geographic data to event fields. Learn more about linking IP addresses to geographic data.

When this type is selected, under **Mapping geographic data to event fields**, you must specify from which event field the IP address will be read, select the required attributes of geographic data, and define the event fields in which geographic data will be written:

1. In the **Event field with IP address** drop-down list, select the event field from which the IP address is read. Geographic data uploaded to LogAnalyzer is matched against this IP address.

   You can use the **Add event field with IP address** button to specify multiple event fields with IP addresses that require geographic data enrichment. You can delete event fields added in this way by clicking the **Delete event field with IP address** button.

   > When the `SourceAddress`, `DestinationAddress`, and `DeviceAddress` event fields are selected, the **Apply default mapping** button becomes available. You can use this button to add preconfigured mapping pairs of geographic data attributes and event fields.

2. For each event field you need to read the IP address from, select the type of geographic data and the event field to which the geographic data should be written.

   You can use the **Add geodata attribute** button to add field pairs for **Geodata attribute** – **Event field to write to**. You can also configure different types of geographic data for one IP address to be written to different event fields. To delete a field pair, click ×.

   - In the **Geodata attribute** field, select which geographic data corresponding to the read IP address should be written to the event. Available geographic data attributes: **Country**, **Region**, **City**, **Longitude**, **Latitude**.
   - In the **Event field to write to**, select the event field which the selected geographic data attribute must be written to.

   You can write identical geographic data attributes to different event fields. If you configure multiple geographic data attributes to be written to the same event field, the event will be enriched with the last mapping in the sequence.

- check DLL Hijacking

DLL Hijacking is an attack technique that involves delivering vulnerable legitimate software along with a malicious dynamic link library (DLL) to the target system. To detect such attacks, LogAnalyzer uses the AI module. The AI module analyzes the launch and runtime parameters of applications and identifies suspicious launches of legitimate software with malicious libraries.

To get a status, LogAnalyzer sends a request to KSN with information from event fields. In response to the request, KSN sends the analysis result.

You can configure DLL Hijacking attack detection in the collector.

To use an enrichment of the **check DLL Hijacking** type, your license must include the AI module. The General administrator must also accept the additional KSN license agreement in the **Kaspersky Security Network** section (**Settings** → **Integrations** → **AI services** → **Kaspersky Security Network**).

Available settings:

- **Workers** is the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.
- **Maximum cache size** is the maximum cache size for KSN responses, in megabytes. The default value is `128 MB`.

  Each KSN response has its own lifetime, during which it is retained in the cache. If LogAnalyzer sends a second request for a DLL file for which KSN has already issued a response, and the lifetime of that response has not yet expired, the request is not actually sent to KSN, but the response is taken from the cache.
- If the **Maximum number of events in enrichment queue** is exceeded, new events are not processed. The default value is `1,000,000`. `0` means that event processing stops if the queue contains any number of events.

- **Perform a repeat request**. KSN may return the **Unclassified** status. This status means that KSN has accepted the request and is currently processing it, but there is no status yet. Processing may take a long time, so the request must be re-sent. If this setting is enabled, specify additional settings:

  - If the **Create duplicates** setting is enabled, LogAnalyzer creates a duplicate event and uses it for repeated requests to KSN. In this case, the main event is routed in accordance with the workflow. A duplicate event is written to the database when one of the following conditions is satisfied:

    - A status other than **Unclassified** is received from KSN.
    - The disk buffer configured using the **Buffer size** setting is full.
    - The **Request count** is exceeded.

    If this setting is disabled, LogAnalyzer continues to send requests using the main event. The event is written to the database when one of the following conditions is satisfied:

    - A status other than **Unclassified** is received from KSN.
    - The disk buffer configured using the **Buffer size** setting is full.
    - The **Request count** is exceeded.

  - The **Request count** is the maximum number of requests to KSN. If, after the specified number of requests, a status other than **Unclassified** is not received, the event is written to the database with the **Unclassified** status. The default setting is `10`.

  - **Poll interval** is the interval between requests, in minutes. The default value is `1 minute`. The minimum value is `1 minute`.

  - **Buffer size** is the size of the disk buffer for events awaiting a repeat request, in megabytes. The default value is `10240 MB`. The minimum value is `10 MB`.

- **Proxy** specifies proxy server settings, if one is needed to connect to the KSN server. You can select one of the available proxy servers or create a new proxy server. By default, a proxy server is not used.

- The **Mapping** table contains a list of parameters that must be sent in a KSN request to get a status, and a list of the corresponding LogAnalyzer event fields. List of parameters in the request:

  - **dll hash (md5)**. Recommended field for mapping: OldFileHash.

  - **dll path**. Full path including the file name; recommended field for mapping: OldFilePath. The path may contain personal data, so LogAnalyzer sends this parameter to KSN in an anonymized form. For example:

    Path before anonymization:
    `C:\Windows\Device\HarddiskVolume1\Users\Alex\malware.dll`

    The same path in anonymized form: `device\harddiskvolume1\users\username`

  - **process hash (md5)**. Recommended field for mapping: FileHash.

  - **process path**. Full path including the file name. The path may contain personal data, so LogAnalyzer sends this parameter to KSN in an anonymized form. For example:

    Path before anonymization:
    `C:\Windows\Device\HarddiskVolume1\Users\Alex\malware.dll`

    The same path in anonymized form: `device\harddiskvolume1\users\username`

  - **dll name**. Recommended field for mapping: OldFileName.

  - **process name**. Recommended field for mapping: FileName.

  - **process signature**. Optional; recommended field for mapping: FileID.

  - **dll size**. Optional; recommended field for mapping: OldFileSize.

  For each required parameter, specify a corresponding LogAnalyzer event field in the table.

4. Use the **Debug** toggle switch to indicate whether or not to enable logging of service operations. Logging is disabled by default.

5. In the **Filter** section, you can specify conditions to identify events that will be processed by the enrichment rule resource. You can select an existing filter from the drop-down list or **create** a new filter.

   Creating a filter in resources

To create a filter:

1. In the **Filter** drop-down list, select **Create new**.

2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.

3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

4. Under **Conditions**, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, fields of additional parameters for identifying the value to be passed to the filter may be displayed. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

   c. In the **Operator** drop-down list, select an operator.

Filter operators

- **=** means the left operand equals the right operand.
- **<** means the left operand is less than the right operand.
- **<=** means the left operand is less than or equal to the right operand.
- **>** means the left operand is greater than the right operand.
- **>=** means the left operand is greater than or equal to the right operand.
- **inSubnet** means the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains** means the left operand contains values of the right operand.
- **startsWith** means the left operand starts with one of the values of the right operand.
- **endsWith** means the left operand ends with one of the values of the right operand.
- **match** means the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit** checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
- **hasVulnerability** checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- The **inActiveList** operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary** checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory** means the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup** means the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- The **TIDetect** operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage.

- **inContextTable** checks for the presence of the entry in the specified context table.
- **intersect** checks for the presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. This check box does not apply to the **InSubnet**, **InActiveList**, **InCategory**, or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, from the **If** drop-down list, select **If not**.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

The new enrichment rule was added to the resource set for the collector.

Step 7. Routing

This is an optional step of the Installation Wizard. On the **Routing** tab of the Installation Wizard, you can select or create destinations with settings indicating the forwarding destination of events processed by the collector. Typically, events from the collector are forwarded to a storage destination for retention, which also makes processed events available for viewing later. If necessary, events can be sent elsewhere, for example, to the event router. In that case, select the 'internal' connector at the Transport step. There can be more than one destination point.

To add an existing destination to a collector resource set:

1. In the **Routing** step of the Installation Wizard, click **Add**.

2. This opens the **Create destination** window; in that window, select the type of destination you want to add.

3. In the **Destination** drop-down list, select the necessary destination.

   The window name changes to **Edit destination**, and it displays the settings of the selected resource. To open the settings of a destination for editing in a new browser tab, click .

4. Click **Save**.

The selected destination is displayed on the Installation Wizard tab. A destination resource can be removed from the resource set by selecting it and clicking **Delete** in the opened window.

To add a new destination resource to a collector resource set:

1. In the **Routing** step of the Installation Wizard, click **Add**.
2. This opens the **Create destination** window; in that window, specify the following settings:
   a. On the **Basic settings** tab, in the **Name** field, enter a unique name for the destination. The name must contain 1 to 128 Unicode characters.
   b. You can use the **State** toggle switch to enable or disable the service as needed.
   c. In the **Type** drop-down list, select the type of the destination. The following values are available:
      - nats-jetstream
      - tcp
      - http
      - kafka
      - file
      - storage
      - eventRouter
   d. On the **Advanced settings** tab, specify the values of parameters. The set of parameters that can be configured depends on the type of the destination selected on the **Basic settings** tab. For detailed information about parameters and their values, click the link for each type of destination in paragraph "c." of these instructions.

The created destination is displayed on the Installation Wizard tab. A destination resource can be removed from the resource set by selecting it and clicking **Delete** in the opened window.

Proceed to the next step of the Installation Wizard.

Step 8. Setup validation

This is the required, final step of the Installation Wizard. At this step, LogAnalyzer creates a service resource set, and services are created automatically based on this set:

- The resource set for the collector is displayed under **Resources** → **Collectors**. It can be used to create new collector services. When this set of resources changes, all services that operate based on this set of resources will start using the new parameters after the services restart. To do so, you can use the **Save and restart services** and **Save and update service configurations** buttons.

  A resource set can be modified, copied, moved from one folder to another, deleted, imported, and exported, like other resources.

- Services are displayed in **Resources** → **Active services**. The services created using the Installation Wizard perform functions inside the LogAnalyzer program. To communicate with external parts of the network infrastructure, you need to install similar external services on the servers and assets intended for them. For example, an external collector service should be installed on a server intended as an events recipient, external storage services should be installed on servers that have a deployed ClickHouse service, and external agent services should be installed on the Windows assets that must both receive and forward Windows events.

To finish the Installation Wizard:

1. Click **Create and save service**.

   The **Setup validation** tab of the Installation Wizard displays a table of services created based on the resource set selected in the Installation Wizard. The lower part of the window shows examples of commands that you must use to install external equivalents of these services on their intended servers and assets.

   For example:

   /opt/kaspersky/kuma/kuma collector --core https://kuma-example:<port used for communication with the LogAnalyzer Core> --id <service ID> --api.port <port used for communication with the service> --install

   > The "kuma" file can be found inside the installer in the /kuma-ansible-installer/roles/kuma/files/ directory.

   The port for communication with the LogAnalyzer Core, the service ID, and the port for communication with the service are added to the command automatically. You should also ensure the network connectivity of the LogAnalyzer system and open the ports used by its components if necessary.

2. Close the Wizard by clicking **Save collector**.

The collector service is created in LogAnalyzer. Now the service must be installed on the server intended for receiving events.

> If a wmi, wec, or etw connector was selected for collectors, you must also install the automatically created LogAnalyzer agents.

Installing a collector in the LogAnalyzer network infrastructure.

A collector consists of two parts: one part is created in the LogAnalyzer web interface, and the other part is installed on the network infrastructure server intended for receiving events. The second part of the collector is installed in the network infrastructure.

To install a collector:

1. Log in to the server where you want to install the service.

2. Execute the following command:

   `sudo /opt/kaspersky/kuma/kuma collector --core https://<FQDN of the LogAnalyzer Core server>:<port used by LogAnalyzer Core for internal communication (port 7210 is used by default)> --id <service ID copied from the LogAnalyzer web interface> --api.port <port used for communication with the installed component>`

   Example: `sudo /opt/kaspersky/kuma/kuma collector --core https://test.kuma.com:7210 --id XXXX --api.port YYYY`

   If errors are detected as a result of the command execution, make sure that the settings are correct. For example, the availability of the required access level, network availability between the collector service and the Core, and the uniqueness of the selected API port. After fixing errors, continue installing the collector.

   If no errors are found, and the collector status in the LogAnalyzer web interface changes to *green*, stop the command execution and proceed to the next step.

   > The command can be copied at the last step of the installer wizard. The address and port of the LogAnalyzer Core server, the identifier of the collector to be installed, and the port that the collector uses for communication are automatically specified in the command.

   > When deploying several LogAnalyzer services on the same host, during the installation process you must specify unique ports for each component using the `--api.port <port>` parameter. The following setting values are used by default: `--api.port 7221`.

   > Before installation, ensure the network connectivity of LogAnalyzer components.

3. Run the command again by adding the `--install` key:

   `sudo /opt/kaspersky/kuma/kuma collector --core https://<FQDN of the LogAnalyzer Core server>:<port used by LogAnalyzer Core for internal communication (port 7210 is used by default)> --id <service ID copied from the LogAnalyzer web interface> --api.port <port used for communication with the installed component> --install`

   Example: `sudo /opt/kaspersky/kuma/kuma collector --core https://kuma.example.com:7210 --id XXXX --api.port YYYY --install`

4. Add LogAnalyzer collector port to firewall exclusions.

   > For the application to run correctly, ensure that the LogAnalyzer components are able to interact with other components and applications over the network using the protocols and ports specified during the installation of the LogAnalyzer components.

The collector is installed. You can use it to receive data from an event source and forward it for processing.

Validating collector installation

To verify that the collector is ready to receive events:

1. In the LogAnalyzer web interface, open **Resources** → **Active services**.
2. Make sure that the collector you installed has the green status.

   If the status of the collector is not green, view the log of this service on the machine where it is installed, in the /opt/kaspersky/kuma/collector/<collector ID>/log/collector directory. Errors are logged regardless of whether debug mode is enabled or disabled.

If the collector is installed correctly and you are sure that data is coming from the event source, the table should display events when you search for events associated with the collector.

*To check for normalization errors using the **Events** section of the LogAnalyzer web interface:*

1. Make sure that the Collector service is running.
2. Make sure that the event source is providing events to the LogAnalyzer.
3. Make sure that in the in the **Resources** section of the LogAnalyzer web interface, in the **Keep raw event** drop-down list for the **Normalizer** resource, you have selected **Only errors**.
4. In the **Events** section of LogAnalyzer, search for events with the following parameters:

   - `ServiceID = <ID of the collector to be checked>`
   - `Raw != ""`

   If any events are found with this search, it means that there are normalization errors and they should be investigated.

To check for normalization errors using the Grafana™ Dashboard:

1. Make sure that the Collector service is running.
2. Make sure that the event source is providing events to the LogAnalyzer.
3. Open the Metrics section and follow the LogAnalyzer Collectors link.
4. See if the Errors section of the Normalization widget displays any errors.

   If there are any errors, it means that there are normalization errors and they should be investigated.

For collectors that use WEC, WMI, or ETW connectors as the transport, make sure that a unique port is used for connecting to the agent. This port is specified in the **Transport** section of Collector Installation Wizard.

Ensuring uninterrupted collector operation

An uninterrupted event stream from the event source to LogAnalyzer is important for protecting the network infrastructure. Continuity can be ensured though automatic forwarding of the event stream to a larger number of collectors:

- On the LogAnalyzer side, two or more identical collectors must be installed.
- On the event source side, you must configure control of event streams between collectors using third-party server load management tools, such as rsyslog or nginx.

With this configuration of the collectors in place, no incoming events will be lost if the collector server is unavailable for any reason.

Please keep in mind that when the event stream switches between collectors, each collector will aggregate events separately.

If the KUMA collector fails to start, and its log includes the "panic: runtime error: slice bounds out of range [8:0]" error:

1. Stop the collector.

   ```
   sudo systemctl stop kuma-collector-<collector ID>
   ```

2. Delete the DNS enrichment cache files.

   ```
   sudo rm -rf /opt/kaspersky/kuma/collector/<collector ID>/cache/enrichment/DNS-*
   ```

3. Delete the event cache files (disk buffer). Run the command only if you can afford to jettison the events in the disk buffers of the collector.

   ```
   sudo rm -rf /opt/kaspersky/kuma/collector/<collector ID>/buffers/*
   ```

4. Start the collector service.

   ```
   sudo systemctl start kuma-collector-<collector ID>
   ```

Event stream control using rsyslog

To enable rsyslog event stream control on the event source server:

1. Create two or more identical collectors that you want to use to ensure uninterrupted reception of events.
2. Install rsyslog on the event source server (refer to the rsyslog documentation).

3. Add rules for forwarding the event stream between collectors to the configuration file /etc/rsyslog.conf:

```
*. * @@ <main collector server FQDN>: <port for incoming events>
$ActionExecOnlyWhenPreviousIsSuspended on
*. * @@ <backup collector server FQDN>: <port for incoming events>
$ActionExecOnlyWhenPreviousIsSuspended off
```

Example configuration file

Example configuration file specifying one primary and two backup collectors. The collectors are configured to receive events on TCP port 5140.

```
*.* @@kuma-collector-01.example.com:5140
$ActionExecOnlyWhenPreviousIsSuspended on
& @@kuma-collector-02.example.com:5140
& @@kuma-collector-03.example.com:5140
$ActionExecOnlyWhenPreviousIsSuspended off
```

4. Restart rsyslog by running the following command:

   `systemctl restart rsyslog`.

Event stream control is now enabled on the event source server.

Event stream control using nginx

To control the event stream using nginx, you need to create and configure an nginx server to receive events from the event source and then forward these to collectors.

To enable nginx event stream control on the event source server:

1. Create two or more identical collectors that you want to use to ensure uninterrupted reception of events.
2. Install nginx on the server intended for event stream control.
   - Installation command in Oracle Linux 8.6:

     `$sudo dnf install nginx`
   - Installation command in Ubuntu 20.4:

     `$sudo apt-get install nginx`

     When installing from sources, you must compile with the parameter `-with-stream` option:
     `$ sudo ./configure -with-stream -without-http_rewrite_module -without-http_gzip_module`

3. On the nginx server, add the stream module to the nginx.conf configuration file that contains the rules for forwarding the stream of events between collectors.

   Example stream module

   Example module in which event stream is distributed between the collectors kuma-collector-01.example.com and kuma-collector-02.example.com, which receive events via TCP on port 5140 and via UDP on port 5141. Balancing uses the nginx.example.com ngnix server.

   ```
   stream {
    upstream syslog_tcp {
   server kuma-collector-1.example.com:5140;
   server kuma-collector-2.example.com:5140;
   }
   upstream syslog_udp {
   server kuma-collector-1.example.com:5141;
   server kuma-collector-2.example.com:5141;
   }
    server {
   listen nginx.example.com:5140;
   proxy_pass syslog_tcp;
   }
   server {
   listen nginx.example.com:5141 udp;
   proxy_pass syslog_udp;
   proxy_responses 0;
   }
   }
    worker_rlimit_nofile 1000000;
   events {
   worker_connections 20000;
   }
   # worker_rlimit_nofile is the limit on the number of open files (RLIMIT_NOFILE) for workers. This is used to raise the limit without restarting the main process.
   # worker_connections is the maximum number of connections that a worker can open simultaneously.
   ```

   With a large number of active services and users, you may need to increase the limit of open files in the nginx.conf settings. For example:

   ```
   worker_rlimit_nofile 1000000;
   events {
   worker_connections 20000;
   }
   # worker_rlimit_nofile is the limit on the number of open files (RLIMIT_NOFILE) for workers. This is used to raise the limit without restarting the main process.
   # worker_connections is the maximum number of connections that a worker can open simultaneously.
   ```

4. Restart nginx by running the following command:

   ```
   systemctl restart nginx
   ```

5. On the event source server, forward events to the nginx server.

Event stream control is now enabled on the event source server.

Nginx Plus may be required to fine-tune balancing, but certain balancing methods, such as Round Robin and Least Connections, are available in the base version of ngnix.

For more details on configuring nginx, please refer to the nginx documentation.

Predefined collectors

The predefined collectors listed in the table below are included in the LogAnalyzer distribution kit.

Predefined collectors

Name	Description
[OOTB] CEF	Collects CEF events received over the TCP protocol.
[OOTB] KSC	Collects events from Kaspersky Security Center over the Syslog TCP protocol.
[OOTB] KSC SQL	Collects events from Kaspersky Security Center using an MS SQL database query.
[OOTB] Syslog	Collects events via the Syslog protocol.
[OOTB] Syslog-CEF	Collects CEF events that arrive over the UDP protocol and have a Syslog header.

Creating an agent

This functionality is only available in the commercial version of LogAnalyzer.

A LogAnalyzer agent consists of two parts: one part is created inside the LogAnalyzer web interface, and the second part is installed on a server or on an asset in the network infrastructure.

An agent is created in several steps:

1. **Creating a resource set for the agent in the LogAnalyzer web interface**
2. **Creating an agent service in the LogAnalyzer web interface**
3. **Installing the server portion of the agent to the asset that will send events**

LogAnalyzer agents for Windows devices can be created as follows:

- Manually by creating a set of agent resources and creating an agent service in the LogAnalyzer web interface.
- Automatically when creating a collector of the wmi, wec, or etw transport type. Although the set of resources and service of these agents are created in the Collector Installation Wizard, they must still be installed to the asset that will be used to forward an event. In a manually created agent, a connector of the etw type can be used in only one connection. If you configure multiple connections in a manually created etw agent, the status of the etw agent status is green, but events are not transmitted and an error is logged in the etw agent log.

Multiple agents can be installed on a device; the version of all such agents must be the same.

If an older version of the agent is already installed on the device where you want to create an agent, you must first stop the installed agent (remove the agent from a Windows device or restart the agent service on a Linux device), and then you can proceed to create a new agent. However, if the version of the installed agents is the same as the version of the agent that you want to create, stopping the agents is not necessary.

When creating and running an agent whose version is 3.0.1 or later, you must accept the terms and conditions of the End User License Agreement.

Creating a set of resources for an agent

In the LogAnalyzer web interface, an agent service is created based on the resource set for an agent that unites connectors and destinations. The agent type is determined by the connector that is used in the agent. The only exception is for agents with a destination of the diode type. These agents are considered to be diode agents.

To create a resource set for an agent in the LogAnalyzer web interface:

1. In the LogAnalyzer web interface, under **Resources** → **Agents**, click **Create new**.

   This opens the agent creation window. The left part of the window displays tabs with base settings of the agent and connections. The **Base settings** tab is active.

2. On the **Base settings** tab:

   - In the **Name** field, enter a unique name for the service you are creating. The name must contain 1 to 128 Unicode characters.
   - In the **Tenant** drop-down list, select the tenant that will own the storage.
   - If you want to enable logging of service operations, enable the **Debug** toggle switch.
   - If you want to view the names of services or addresses of hosts from which the event came, enable the **Track event route** toggle switch.

     The **Track event route** toggle switch is available if at least one **internal** destination is specified in the connections. By default, the toggle switch is **Disabled**.
   - You can optionally add up to 256 Unicode characters describing the service in the **Description** field.

3. Go to the tab of an existing connection or create a connection by clicking the **Add** button in the lower part of the left pane, then go to the tab of the newly created connection to edit its settings.

   By default, a connection named **Config #1** is created for a new agent. The name of the connection follows the **Config #<number>** pattern.

   You can create multiple connections for an agent. If necessary, you can manage connections:

   - Rename connections
   - Duplicate connections
   - Delete connections

4. If necessary, in the **Name of connection** field, rename the connection for your convenience when managing it, for example, to be able to figure out from which connection and from which agent events arrived. If you leave the field blank, a name is assigned following the **Config #<number>** pattern.

   The name can contain from 1 to 128 characters. The name can contain only letters and numerals and cannot contain special characters. Leading and trailing spaces are removed. When pasting a name into the field from the clipboard, if the text contains a newline, paragraph, or indentation, these characters are replaced with a space. You can reuse a name for multiple connections within the same agent.

   If you have enabled event route tracing, then when viewing event information, the **Events** section displays the name of the connection from which the event was received.

5. In the **Connector** group of settings, add a connector:

- If you want to select an existing connector, select it from the drop-down list.
- If you want to create a new connector, select **Create new** in the drop-down list and specify the following settings:
  - Specify the connector name in the **Name** field. The name must contain 1 to 128 Unicode characters.
  - In the **Type** drop-down list, select the connector type and specify its settings on the **Basic settings** and **Advanced settings** tabs. The available settings depend on the selected type of connector:
    - tcp
    - udp
    - nats-jetstream
    - kafka
    - http
    - file
    - ftp
    - nfs
    - wmi
    - wec
    - snmp
    - snmp-trap
    - etw
    - kata/edr
    - office365
    - dfi
    - vmware
    - 1c-log
    - 1c-xml

    When using the **tcp**, **udp**, **http**, **wec**, **wmi**, **etw** connector type, at the normalization stage, IP addresses of the assets from which the events were received are written to the DeviceAddress event field if it is empty.

> The ability to edit previously created wec, wmi, or etw connections in agents, collectors, and connectors is limited. You can change the connection type from **wec** to **wmi** or **etw** and back, but you cannot change the **wec**, **wmi**, or **etw** connection to any other connection type. When editing other connection types, you cannot select the **wec**, **wmi**, or **etw** types. You can create connections without any restrictions on the types of connectors.

> When adding an (existing or new) wmi, wec, or etw connector for an agent, the **TLS mode** and **Compression** settings are not displayed on the agent, but the values of these settings are stored in the agent's configuration. For a new connector, these settings are disabled by default.
> If TLS mode is enabled for an existing connector that is selected in the list, you cannot download the agent configuration file. In this case, to download the configuration file, you must go to the connector resource that is being used on the agent and disable TLS mode.

The connector is added to the selected connection of the agent's resource set. The created connector is only available in this resource set and is not displayed in the web interface **Resources → Connectors section**.

6. In the **Destinations** group of settings, add a destination.

   - If you want to select an existing destination, select it from the drop-down list.
   - If you want to create a new destination, select **Create new** in the drop-down list and specify the following settings:
     - Specify the destination name in the **Name** field. The name must contain 1 to 128 Unicode characters.
     - In the **Type** drop-down list, select the destination type and specify its settings on the **Basic settings** and **Advanced settings** tabs. The available settings depend on the selected type of destination:
       - **nats-jetstream**—used for NATS communications.
       - **tcp**—used for TCP communication.
       - **http**—used for HTTP communication.
       - **diode**—used to transmit events using a data diode.
       - **kafka**—used for Kafka communication.
       - **file**—used for writing to a file.
     - Enable or disable the **State** toggle switch to enable or disable the sending of events to the destination. This toggle switch is turned on by default.

   > The advanced settings for an agent destination (such as TLS mode and compression) must match the advanced destination settings for the collector that you want to link to the agent.

   There can be more than one destination point. You can add them by clicking the **Add destination** button and can remove them by clicking the ✕ button.

7. Repeat steps 3–5 for each agent connection that you want to create.

8. Click **Save**.

The resource set for the agent is created and displayed under **Resources** → **Agents**. Now you can create an agent service in LogAnalyzer.

Managing connections for an agent

You can manage the connections created for the agent by renaming, duplicating, or deleting them.

Renaming a connection

By default, the name of a new connection created in a resource set for an agent follows the **Connection <number>** pattern. For your convenience when managing connections, you can rename them, for example, to make it clear at a glance from which connection and from which agent an event was received.

To rename a connection:

1. In the LogAnalyzer web interface, in the **Resources** → **Agents** section, create a new agent or click an existing agent in the table.
2. If necessary, create a connection for the agent by clicking the **Add** button in the lower part of the left pane.
3. Go to the tab of the connection that you want to rename.
4. In the **Connection name** field, edit the name of the connection.

   The name can contain from 1 to 128 characters. The name can contain only letters and numerals and cannot contain special characters. Leading and trailing spaces are removed. When pasting a name into the field from the clipboard, if the text contains a newline, paragraph, or indentation, these characters are replaced with a space. You can reuse a name for multiple connections within the same agent.
5. Click **Save**.

The connection is renamed. If you have enabled event route tracing, then when viewing event information, the **Events** section displays the name of the connection from which the event was received.

Duplicating a connection

If you want to create a connection for an agent based on an existing connection, you can create a copy of the connection with identical settings.

To duplicate a connection:

1. In the LogAnalyzer web interface, in the **Resources** → **Agents** section, create a new agent or click an existing agent in the table.
2. If necessary, create a connection for the agent by clicking the **Add** button in the lower part of the left pane.
3. Go to the tab of the connection that you want to duplicate and click the **Duplicate** button.

A connection is created for the agent with the same settings as the original connection. The new connection is created with one of the following names:

- If the original connection name followed the **Connection <number>** template, the duplicated connection name also follows this template. The connection is named "Connection <number+1>", where <number> is the number of the last created connection whose name followed the same template.
- If the original connection had been renamed, the new connection gets the same name. You can reuse a name for multiple connections within the same agent.

You can edit the name of the new connection in the **Name of connection** field on the tab of the connection.

Removing a connection

If the agent has more than one connection, you can delete a connection.

To delete a connection:

1. In the LogAnalyzer web interface, in the **Resources** → **Agents** section, create a new agent or click an existing agent in the table.
2. If necessary, create a connection for the agent by clicking the **Add** button in the lower part of the left pane.
3. Go to the tab of the connection that you want to delete and click the **Delete** button.

You cannot restore a deleted connection, but you can recover the version of the agent in which this connection was previously saved through the change history.

Creating an agent service in the LogAnalyzer web interface

When a resource set is created for an agent, you can proceed to create an agent service in LogAnalyzer.

To create an agent service in the LogAnalyzer web interface:

1. In the LogAnalyzer web interface, under **Resources** → **Active services**, click **Add service**.
2. In the opened **Choose a service** window, select the resource set that was just created for the agent and click **Create service**.

The agent service is created in the LogAnalyzer web interface and is displayed under **Resources** → **Active services**. Now agent services must be installed to each asset from which you want to forward data to the collector. A service ID is used during installation.

Automatically created agents

When creating a collector with wec, wmi, or etw connectors, agents are automatically created for receiving Windows events.

Automatically created agents have the following special conditions:

- Automatically created agents can have only one connection.
- Automatically created agents are displayed under **Resources** → **Agents**, and `auto created` is indicated at the end of their name. Agents can be reviewed or deleted.
- The settings of automatically created agents are defined automatically based on the collector settings from the **Connect event sources** and **Transport** sections. You can change the settings only for a collector that has a created agent.
- The description of an automatically created agent is taken from the collector description in the **Connect event sources** section.
- Debugging of an automatically created agent is enabled and disabled in the **Connect event sources** section of the collector.
- When deleting a collector with an automatically created agent, you will be prompted to choose whether to delete the collector together with the agent or to just delete the collector. When deleting only the collector, the agent will become available for editing.
- When deleting automatically created agents, the type of collector changes to **http**, and the connection address is deleted from the **URL** field of the collector.
- If at least one Windows log name in wec or wmi connector is specified incorrectly, the agent will not receive events from any Windows log listed in the connector. At the same time the agent status will be green. Attempts to receive events will be repeated every 60 seconds, and error messages will be added to the service log.
- If in a connector of the etw type, the session name is specified incorrectly, the wrong provider is specified in the session, or an incorrect method is specified for sending events (to send events correctly, on the Windows Server side, you must specify "Real time" or "File and Real time" mode), events will not arrive from the agent, an error will be recorded in the agent log on Windows, and the status of the agent will be green. At the same time, no attempt will be made to get events every 60 seconds. If you modify session settings on the Windows side, you must restart the etw agent and/or the session for the changes to take effect.

In the LogAnalyzer interface, automatically created agents appear at the same time when the collector is created. However, they must still be installed on the asset that will be used to forward a message.

Installing an agent in a LogAnalyzer network infrastructure

When an agent service is created in LogAnalyzer, you can proceed to installation of the agent to the network infrastructure assets that will be used to forward data to a collector.

Multiple agents can be installed on a device; the version of all such agents must be the same.

Prior to installation, verify the network connectivity of the system and open the ports used by its components.

Installing a LogAnalyzer agent on Linux devices

LogAnalyzer agent installed on Linux devices stops when you close the terminal or restart the server. If you do not want to start LogAnalyzer agents manually, we recommend installing agents using an application that automatically starts applications whenever the server is restarted, for example, the Supervisor application. If you want to start LogAnalyzer agents automatically, specify the automatic start and restart settings in the LogAnalyzer configuration file. For more information on configuring automatic starting and restarting, see the official documentation of applications for automatically starting applications.

Example configuration in Supervisor

```
[program:agent_<name of LogAnalyzer agent>] command=sudo /opt/kaspersky/kuma/agent
agent --core https://<FQDN of LogAnalyzer Core server>:<port of LogAnalyzer Core
server>
autostart=true
autorestart=true
```

To install a LogAnalyzer agent on a Linux asset:

1. On the Linux device on which you want to install the LogAnalyzer agent, create directories for the LogAnalyzer configuration file and agents, for example:
   - /opt/kaspersky/kuma
   - /opt/kaspersky/agent
2. Place the LogAnalyzer configuration file in the directory created for it. You can find the LogAnalyzer configuration file inside the installer in the /kuma-ansible-installer/roles/kuma/files directory.

   Make sure the kuma file has sufficient rights to run.
3. Create the LogAnalyzer user:

   ```
   sudo useradd --system kuma && usermod -s /usr/bin/false kuma
   ```
4. Grant the LogAnalyzer user access to the directory with the LogAnalyzer configuration file and to all files within the directory:

   ```
   sudo chown -R kuma:kuma <path to the directory with the LogAnalyzer configuration
   file>
   ```

5. Install the LogAnalyzer agent:

   ```
   sudo /opt/kaspersky/kuma/agent agent --core https://<LogAnalyzer Core server FQDN>:<port used by LogAnalyzer Core for internal communication (port 7210 by default)> --id <service ID copied from the LogAnalyzer web interface> --wd <path to the directory that will contain the files of the installed LogAnalyzer agent. If this option is not specified, the files will be stored in the directory where the LogAnalyzer file is located> [--accept-eula]
   ```

   You can install two LogAnalyzer agents on the same Linux device. In this case, the LogAnalyzer agents will work in parallel. When installing the second LogAnalyzer agent, you need to specify a separate directory for it using the `--wd` option.

   To run the agent, you need to accept the End User License Agreement. You can add the `--accept-eula` option to the command to automatically accept the End User License Agreement during LogAnalyzer agent installation. This lets you perform the installation non-interactively. If you do not specify this option, you will need to accept or reject the License Agreement manually during the installation of the LogAnalyzer agent.

   Examples of installing the LogAnalyzer agent:

   - Installing the LogAnalyzer agent without automatically accepting the End User License Agreement:

     ```
     sudo /opt/kaspersky/kuma/agent agent --core https://kuma.example.com:7210 --id XXXX --wd /opt/kaspersky/kuma/agent/XXXX
     ```

   - Installing the LogAnalyzer agent with automatic acceptance of the End User License Agreement:

     ```
     sudo /opt/kaspersky/kuma/agent agent --core https://kuma.example.com:7210 --id XXXX --wd /opt/kaspersky/kuma/agent/XXXX --accept-eula
     ```

   > By using the `--accept-eula` option during the installation of the LogAnalyzer agent, you confirm that you agree with and accept the terms and conditions of the End User License Agreement.

6. If you chose LogAnalyzer installation with the automatic acceptance of the End User License Agreement and want to read the text of the End User License Agreement, or if the text of the End User License Agreement was not automatically provided to you during the installation process, run the following command:

   ```
   ./kuma license --show
   ```

The LogAnalyzer agent is installed on the Linux device.

You can configure a collector to receive data that the LogAnalyzer agent sends to LogAnalyzer.

Installing a LogAnalyzer agent on Windows devices

Before installing the LogAnalyzer agent on a Windows device, make sure that the user account under which the LogAnalyzer agent will run has the necessary rights.

Rights for a user account when installing a LogAnalyzer agent of any type

The server administrator must create a user account with the 'Log on as a service' permission on the Windows asset. This user account must be used to install the agent.

Rights for a user account when installing a LogAnalyzer agent of the wec type

The server administrator must create a user account with the 'Event Log Readers' and 'Log on as a service' permissions on the Windows asset. This user account must be used to install and start the agent.

If you plan to receive events from the Security log on a domain controller, grant access to the Security log to the the preset Network Service account in Windows. To do this, run the following command on the PowerShell command line of the domain controller device:

```
wevtutil sl security /ca:'O:BAG:SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-5-32-573)
(A;;0x1;;;S-1-5-20)'
```

> If you do not configure access to the Security log, but add the Security log in the wec connector settings, then the stream of events from the LogAnalyzer agent will not be received.

If you want to receive events from something other than a domain controller, it is sufficient to add the preset Network Service user account in Windows to the Event Log Readers group.

Rights for a user account when installing a LogAnalyzer agent of the wmi type

The server administrator must create a user account with the 'Event Log Readers' and 'Log on as a service' permissions on the Windows asset. This user account must be used to install the agent.

The 'Event Log Readers' and 'Log on as a service' permissions are required to run the agent under a domain account.

If you want to run the agent under a local account, you will need administrator rights as well as 'Event Log Readers' and 'Log on as a service' permissions.

If you want to perform remote collection of events (collection of events from devices connected to the device on which the LogAnalyzer agent is installed) and only read logs under a domain account, the 'Event Log Readers' permission is sufficient.

Rights for a user account when installing a LogAnalyzer agent of the etw type

The server administrator must create a user account with the 'Log on as a service' permission on the Windows asset. This user account must be used to install the agent.

To run the agent, the user account must have administrator rights or must be added to one of the following security groups:

- Domain admins
- Performance Log Users

Installing the LogAnalyzer agent

To install a LogAnalyzer agent on a Windows asset:

1. Copy the kuma.exe file to a folder on the Windows asset. `C:\Users\<User name>\Desktop\KUMA` folder is recommended for installation.

   The kuma.exe file is located inside the installer in the /kuma-ansible-installer/roles/kuma/files/ directory.

2. Start the Command Prompt on the Windows asset with Administrator privileges and locate the folder containing the kuma.exe file.

3. Execute the following command:

   `kuma agent --core https://<fully qualified domain name of the LogAnalyzer Core server>:<port used by the LogAnalyzer Core server for internal communications (port 7210 by default)> --id <ID of the agent service that was created in LogAnalyzer> --user <name of the user account used to run the agent, including the domain> --install [--accept-eula]`

   To run the agent, you need to accept the End User License Agreement. You can add the `--accept-eula` option to the command to automatically accept the End User License Agreement during agent installation. This lets you perform the installation non-interactively. If you do not specify this option, you will need to accept or reject the License Agreement manually during the installation of the LogAnalyzer agent.

   Examples:

   - Installing the LogAnalyzer agent without automatically accepting the End User License Agreement:

     `kuma agent --core https://kuma.example.com:7210 --id XXXXX --user domain\username --install`

   - Installing the LogAnalyzer agent with automatic acceptance of the End User License Agreement:

     `kuma agent --core https://kuma.example.com:7210 --id XXXXX --user domain\username --install --accept-eula`

   By using the `--accept-eula` option during the installation of the LogAnalyzer agent, you confirm that you agree with and accept the terms and conditions of the End User License Agreement.

   You can get help information by executing the `kuma help agent` command.

4. If you started the installation of the agent without automatically accepting the End User License Agreement, during the installation process, you will be prompted to read the text of the End User License Agreement and you will have the opportunity to accept or reject the agreement.

5. If you chose installation with the automatic acceptance of the End User License Agreement and want to read the text of the End User License Agreement, or if the text of the End User License Agreement was not automatically provided to you during the installation process, run the following command:

   `kuma.exe license --show`

   If you want to accept the End User License Agreement, run the following command and press `y`:

   `kuma.exe license`

6. Specify the password for the agent's user in one the following ways:

   - Manually.

   - From a file when installing the agent.

     To avoid entering the password manually, you can add the password to a .TXT file and place the file in the same directory as the executable file of the agent.

     In the command, specify the file containing the password as follows: `< name of the password file >.txt`. For example, `password.txt`.

     ```
     kuma agent --core https://kuma.example.com:7210 --id XXXXX --user domain\username --install < password.txt
     ```

     All data from the password.txt file is redirected to the command as the password, excluding end-of-line characters (\n\r, \n).

     Path for storing the agent folder after installation: C:\ProgramData\Kaspersky Lab\KUMA\agent\${agent_id}.

   If you did not accept the End User License Agreement (EULA) before installing the agent with the `< password.txt` flag, an error message is displayed saying that the agreement was not accepted.

   You can accept the EULA in the following ways:

   a. In the password file, you can specify the EULA acceptance flag. To achieve this, the file with the password must contain two lines: the first line containing "y", which serves to assert EULA acceptance, and the second line containing the password. For example:

      ```
      y
      <password>
      ```

      The agent installation command looks like this:

      ```
      kuma agent --core https://kuma.example.com:7210 --id XXXXX --user domain\username --install < password.txt
      ```

      In this case, you will not need to additionally accept the EULA when installing the agent.

      If the EULA has been accepted before running the command with the password file, the first line in the file is interpreted as the password and the installation is aborted with an `Access denied` error.

   b. You can accept the EULA and perform a silent installation at the same time by using the --accept-eula flag.

      The agent installation command looks like this:

      ```
      kuma agent --core https://kuma.example.com:7210 --id XXXXX --user domain\username --accept-eula --install < password.txt
      ```

      Make sure that only the password is specified in the file, on the first line, otherwise an error will occur.

The `C:\ProgramData\Kaspersky Lab\KUMA\agent\< agent ID >` folder is created and the LogAnalyzer agent service is installed in it. The agent forwards Windows events to LogAnalyzer, and you can set up a collector to receive them.

When the agent service is installed, it starts automatically. The service is also configured to restart in case of any failures. The agent can be restarted from the LogAnalyzer web interface, but only when the service is active. Otherwise, the service needs to be manually restarted on the Windows asset.

Removing a LogAnalyzer agent from Windows devices

To remove a LogAnalyzer agent from a Windows asset:

1. Start the Command Prompt on the Windows machine with Administrator privileges and locate the folder with kuma.exe file.
2. Run any of the commands below:
   - `kuma.exe agent --cfg <path to agent configuration file> --uninstall`
   - `kuma.exe agent --id <ID of agent service that was created in LogAnalyzer> --uninstall`

   The specified LogAnalyzer agent is removed from the Windows device. Windows events are no longer sent to LogAnalyzer.

When configuring services, you can check the configuration for errors before installation by running the agent with the following command:

```
kuma agent --core https://<fully qualified domain name of the LogAnalyzer Core server>:<port used by the LogAnalyzer Core server for internal communications (port 7210 by default)> --id <ID of the agent service that was created in LogAnalyzer> --user <name of the user account used to run the agent, including the domain>
```

Update agents

When updating LogAnalyzer versions, the WMI, WEC, and ETW agents installed on remote machines must also be updated.

To update the agent, use an administrator account and follow these steps:

1. In the LogAnalyzer web interface, in the **Resources** → **Active services** - **Agents** section, select the agent that you want to update and copy its ID.

   You need the ID to install the new agent with the same ID after removing the old agent.

2. In Windows, in the **Services** section, open the agent and click **Stop**.

3. On the command line, go to the folder where the agent is installed and run the command to remove the agent from the server.

   kuma.exe agent --id `<ID of agent service that was created in KUMA>` --uninstall

4. Place the new agent in the same folder.

5. On the command line, go to the folder with the new agent and from that folder, run the installation command using the agent ID from step 1.

   kuma agent --core https://<fullly qualified domain name of the KUMA Core server>:<port used by the KUMA Core server for internal communications (port 7210 by default)> --id <ID of the agent service that was created in KUMA> --user <name of the user account used to run the agent, including the domain> --install

6. When when installing the updated agent on a device for the first time, you must confirm the license. During the installation process, you are prompted to read the text of the license and then accept or reject the agreement. If this did not happen automatically, you can use the following command to view the text of the license:

   ```
   kuma.exe license --show
   ```

   If you want to accept the license agreement, run the command and press `y`:

   ```
   kuma.exe license
   ```

The agent is updated.

Exporting events from isolated network segments to LogAnalyzer

Data transfer scenario

Data diodes can be used to transfer events from isolated network segments to LogAnalyzer. Data transfer is organized as follows:

1. The LogAnalyzer agent that is with a **diode** destination installed on a standalone server receives events and moves them to a directory from which the data diode picks up the events.

   The agent accumulates events in a buffer until it overflows or for a user-defined period after the last write to disk. The events are then written to a file in the temporary directory of the agent. The file is moved to the directory processed by the data diode; its name is a combination of the file contents hash (SHA-256) and the file creation time.

2. The data diode moves files from the isolated server directory to the external server directory.

3. A LogAnalyzer collector with a **diode** connector installed on an external server reads and processes events from the files of the directory where the data diode places files.

   After all events are read from a file, it is automatically deleted. Before reading events, the contents of files are verified based on the hash in the file name. If the contents fail verification, the file is deleted.

In the described scenario, the LogAnalyzer components are responsible for moving events to a specific directory within the isolated segment and for receiving events from a specific directory in the external network segment. The data diode transfers files containing events from the directory of the isolated network segment to the directory of the external network segment.

For each data source within an isolated network segment, you must create its own LogAnalyzer collector and agent, and configure the data diode to work with separate directories.

Configuring LogAnalyzer components

Configuring LogAnalyzer components for transferring data from isolated network segments consists of the following steps:

1. Creating a collector service in the external network segment.

   At this step, you must create and install a collector to receive and process the files that the data diode will transfer from the isolated network segment. You can use the Collector Installation Wizard to create the collector and all the resources it requires.

   At the **Transport** step, you must select or create a connector of the **diode** type. In the connector, you must specify the directory to which the data diode will move files from the isolated network segment.

   The user "kuma" that runs the collector must have read/write/delete permissions in the directory to which the data diode moves data from the isolated network segment.

2. Creating a resource set for a LogAnalyzer agent.

   At this step, you must create a resource set for the LogAnalyzer agent that will receive events in an isolated network segment and prepare them for transferring to the data diode. The diode agent resource set has the following requirements:

   - The destination in the agent must have the **diode** type. In this resource, you must specify the directory from which the data diode will move files to the external network segment.
   - You cannot select connectors of the **sql** type for the diode agent.
   - TLS mode must be disabled in the connector of the diode agent.

3. Downloading the agent configuration file as JSON file.

   a. The set of agent resources from a diode-type destination must be downloaded as a JSON file.

   b. If secret resources were used in the agent resource set, you must manually add the secret data to the configuration file.

4. Installing the LogAnalyzer agent service in the isolated network segment.

   At this step, you must install the agent in an isolated network segment based on the agent configuration file that was created at the previous step. It can be installed to Linux and Windows devices.

Configuring a data diode

The data diode must be configured as follows:

- Data must be transferred atomically from the directory of the isolated server (where the LogAnalyzer agent places the data) to the directory of the external server (where the LogAnalyzer collector reads the data).
- The transferred files must be deleted from the isolated server.

For information on configuring the data diode, please refer to the documentation for the data diode used in your organization.

Special considerations

When working with isolated network segments, SQL is not supported.

In the scenario described above, the agent cannot be administered through the LogAnalyzer web interface because it resides in an isolated network segment. Such agents are not displayed in the list of active LogAnalyzer services.

Diode agent configuration file

A created set of agent resources with a diode-type destination can be downloaded as a configuration file. This file is used when installing the agent in an isolated network segment.

To download the configuration file:

In the LogAnalyzer web interface, under **Resources** → **Agents**, select the required set of agent resources with a diode destination and click **Download config**.

The agent settings configuration is downloaded as a JSON file based on the settings of your browser. Secrets used in the agent resource set are downloaded empty. Their IDs are specified in the file in the `secrets` section. To use a configuration file to install an agent in an isolated network segment, you must manually add secrets to the configuration file (for example, specify the URL and passwords used in the agent connector to receive events).

You must use an access control list (ACL) to configure permissions to access the file on the server where the agent will be installed. File read access must be available to the user account that will run the diode agent.

Below is an example of a diode agent configuration file with a kafka connector.

```
{
"config": {
"id": "<ID of the set of agent resources>",
"name": "<name of the set of agent resources>",
"proxyConfigs": [
{
"connector": {
"id": "<ID of the connector. This example shows a kafka-type connector, but other types of connectors can also be used in a diode agent. If a connector is created directly in the set of agent resources, the ID is not defined.>",
"name": "<name of the connector>",
"kind": "kafka",
"connections": [
{
"kind": "kafka",
"urls": [
"localhost:9093"
],
"host": "",
"port": "",
"secretID": "<ID of the secret>",
"clusterID": "",
"tlsMode": "",
"proxy": null,
"rps": 0,
"maxConns": 0,
"urlPolicy": "",
"version": "",
"identityColumn": "",
"identitySeed": "",
"pollInterval": 0,
"query": "",
"stateID": "",
"certificateSecretID": "",
"authMode": "pfx",
"secretTemplateKind": "",
"certSecretTemplateKind": ""
}
],
"topic": "<kafka topic name>",
"groupID": "<kafka group ID>",
"delimiter": "",
"bufferSize": 0,
"characterEncoding": "",
"query": "",
"pollInterval": 0,
"workers": 0,
"compression": "",
"debug": false,
"logs": [],
"defaultSecretID": "",
"snmpParameters": [
{
"name": "",
"oid": "",
"key": ""
}
],
"remoteLogs": null,
"defaultSecretTemplateKind": ""
},
"destinations": [
{
"id": "<ID of the destination. If the destination is created directly in the set of agent resources, the ID is not defined.>",
"name": "<destination name>",
"kind": "diode",
"connection": {
"kind": "file",
"urls": [
"<path to the directory where the destination should place events that the data diode will transmit from the isolated network segment>",
"<path to the temporary directory in which events are placed to prepare for data transmission by the diode>"
],
"host": "",
"port": "",
"secretID": "",
"clusterID": "",
"tlsMode": "",
"proxy": null,
"rps": 0,
```

```
"maxConns": 0,
"urlPolicy": "",
"version": "",
"identityColumn": "",
"identitySeed": "",
"pollInterval": 0,
"query": "",
"stateID": "",
"certificateSecretID": "",
"authMode": "",
"secretTemplateKind": "",
"certSecretTemplateKind": ""
},
"topic": "",
"bufferSize": 0,
"flushInterval": 0,
"diskBufferDisabled": false,
"diskBufferSizeLimit": 0,
"healthCheckPath": "",
"healthCheckTimeout": 0,
"healthCheckDisabled": false,
"timeout": 0,
"workers": 0,
"delimiter": "",
"debug": false,
"disabled": false,
"compression": "",
"filter": null,
"path": ""
}
]
}
],
"workers": 0,
"debug": false
},
"secrets": {
"<secret ID>": {
"pfx": "<encrypted pfx key>",
"pfxPassword": "<password to the encrypted pfx key. The 'changeit' value is exported from LogAnalyzer instead of the actual password. In the configuration file, you must manually specify the contents of secrets>"
}
},
"tenantID": "<ID of the tenant>"
}
```


Description of secret fields

Secret fields

Field name	Type	Description
`user`	string	User name
`password`	string	Password
`token`	string	Token
`urls`	array of strings	URL list
`publicKey`	string	Public key (used in PKI)
`privateKey`	string	Private key (used in PKI)
`pfx`	string containing the base64-encoded pfx file	Base64-encoded contents of the PFX file. In Linux, you can get the base64 encoding of a file by running the following command: `base64 -w0 src > dst`
`pfxPassword`	string	Password of the PFX
`securityLevel`	string	Used in snmp3. Possible values: `NoAuthNoPriv`, `AuthNoPriv`, `AuthPriv`

`community`	string	Used in snmp1
`authProtocol`	string	Used in snmp3. Possible values: `MD5`, `SHA`, `SHA224`, `SHA256`, `SHA384`, `SHA512`
`privacyProtocol`	string	Used in snmp3. Possible values: `DES`, `AES`
`privacyPassword`	string	Used in snmp3
`certificate`	string containing the base64-encoded pem file	Base64-encoded contents of the PEM file. In Linux, you can get the base64 encoding of a file by running the following command: `base64 -w0 src > dst`

Installing Linux Agent in an isolated network segment

To install a LogAnalyzer agent to a Linux device in an isolated network segment:

1. Place the following files on the Linux server in an isolated network segment that will be used by the agent to receive events and from which the data diode will move files to the external network segment:

   - Agent configuration file.

     You must use an access control list (ACL) to configure access permissions for the configuration file so that only the KUMA user will have file read access.

   - Executive file /opt/kaspersky/kuma/kuma (the "kuma" file can located in the installer in the /kuma-ansible-installer/roles/kuma/files/ directory).

2. Execute the following command:

   ```
   sudo ./agent agent --cfg <path to the agent configuration file> --wd <path to the directory for agent files. If this flag is not specified, the files will be stored in the directory where the kuma file is located>
   ```

The agent service is installed and running on the server in an isolated network segment. It receives events and relays them to the data diode so that they can be sent to an external network segment.

Installing Windows Agent in an isolated network segment

Before installing the LogAnalyzer agent on a Windows asset, the server administrator must create a user with the EventLogReaders and Log on as a service permissions on the Windows device. This user account must be used to start the agent.

To install a LogAnalyzer agent on a Windows device in an isolated network segment:

1. Place the following files on the Window server in an isolated network segment that will be used by the agent to receive events and from which the data diode will move files to the external network segment:

   - Agent configuration file.

     You must use an access control list (ACL) to configure access permissions for the configuration file so that the file can only be read by the user account that will run the agent.

   - Kuma.exe executable file. This file can be found inside the installer in the /kuma-ansible-installer/roles/kuma/files/ directory.

   We recommend using the `C:\Users\<user name>\Desktop\KUMA` folder.

2. Start the Command Prompt on the Windows asset with Administrator privileges and locate the folder containing the kuma.exe file.

3. Execute the following command:

   ```
   kuma.exe agent --cfg <path to the agent configuration file> --user <user name that will run the agent, including the domain> --install
   ```

   You can get installer Help information by running the following command:

   ```
   kuma.exe help agent
   ```

4. Enter the password of the user account used to run the agent.

The `C:\Program Files\Kaspersky Lab\KUMA\agent\<Agent ID>` folder is created in which the LogAnalyzer agent service is installed. The agent moves events to the folder so that they can be processed by the data diode.

When installing the agent, the agent configuration file is moved to the directory C:\Program Files\Kaspersky Lab\KUMA\agent\<agent ID specified in the configuration file>. The kuma.exe file is moved to the C:\Program Files\Kaspersky Lab\KUMA directory.

When installing an agent, its configuration file must not be located in the directory where the agent is installed.

When the agent service is installed, it starts automatically. The service is also configured to restart in case of any failures.

Removing a LogAnalyzer agent from Windows devices

To remove a LogAnalyzer agent from a Windows asset:

1. Start the Command Prompt on the Windows machine with Administrator privileges and locate the folder with kuma.exe file.
2. Run any of the commands below:
   - `kuma.exe agent --cfg <path to agent configuration file> --uninstall`
   - `kuma.exe agent --id <ID of agent service that was created in LogAnalyzer> --uninstall`

   The specified LogAnalyzer agent is removed from the Windows device. Windows events are no longer sent to LogAnalyzer.

When configuring services, you can check the configuration for errors before installation by running the agent with the following command:

```
kuma.exe agent --cfg <path to agent configuration file>
```

Exporting events from Windows machines to LogAnalyzer

To export events from Windows machines, LogAnalyzer uses a combination of an agent and a collector. Data transfer is organized as follows:

1. The LogAnalyzer agent installed on the machine obtains Windows events:
   - Using the WEC connector: the agent receives events arriving at the host under a subscription, as well as the server logs.
   - Using the WMI connector: the agent connects to remote servers specified in the configuration and receives events.
   - Using the ETW connector: the agent connect to the DNS server using the session name and provider specified in the connector settings, and receives events.
2. The agent sends events (without preprocessing) to the LogAnalyzer collector specified in the destination.

   You can configure the agent so that different logs are sent to different collectors.
3. The collector receives events from the agent, performs a full event processing cycle, and sends the processed events to the destination.

Receiving events from the WEC agent is recommended when using centralized gathering of events from Windows hosts using Windows Event Forwarding (WEF). The agent must be installed on the server that collects events; it acts as the Windows Event Collector (WEC). We do not recommend installing KUMA agents on every endpoint host from which you want to receive events.

The process of configuring the receipt of events using the WEC Agent is described in detail in the appendix: Configuring receipt of events from Windows devices using KUMA Agent (WEC).

For details about the Windows Event Forwarding technology, please refer to the official Microsoft documentation.

We recommend receiving events using the WMI agent in the following cases:

- If it is not possible to use the WEF technology to implement centralized gathering of events, and at the same time, installation of third-party software (for example, the KUMA agent) on the event source server is prohibited.
- If you need to obtain events from a small number of hosts — no more than 500 hosts per one KUMA agent.

The ETW agent is used only to retrieve events from Windows logs of DNS servers.

For connecting Windows logs as an event source, we recommend using the **Add event source** Wizard. When using a wizard to create a collector with WEC or WMI connectors, agents are automatically created for receiving Windows events. You can also manually create the resources necessary for collecting Windows events.

An agent and a collector for receiving Windows events are created and installed in several stages:

1. **Creating a set of resources for an agent.**

   Agent connector:

   When creating an agent, on the **Connection** tab, you must create or select a connector of the WEC, WMI, or ETW type.

   If at least one Windows log name in a WEC or WMI connector is specified incorrectly, the agent will receive events from all Windows logs listed in the connector, except the problematic log. At the same time the agent status will be green. Attempts to receive events will be repeated every 60 seconds, and error messages will be added to the service log.

   Agent destination:

   The type of agent destination depends on the data transfer method you use: nats-jetstream, tcp, http, diode, kafka, file.

   You must use the `\0` value as the destination separator.

   The advanced settings for the agent destination (such as separator, compression and TLS mode) must match the advanced destination settings for the collector connector that you want to link to the agent.

2. **Creating an agent service in the LogAnalyzer web interface.**

3. **Installing the LogAnalyzer agent on the Windows machine from which you want to receive Windows events.**

   Before installation, make sure that the system components have access to the network and open the necessary network ports:

   - Port 7210, TCP: from server with collectors to the Core.
   - Port 7210, TCP: from agent server to the Core.
   - The port configured in the **URL** field when the connector was created: from the agent server to the server with the collector.

4. **Creating and installing a LogAnalyzer collector.**

   When creating a set of collectors, at the **Transport** step, you must create or select a connector that the collector will use to receive events from the agent. Connector type must match the type of the agent destination.

   The advanced settings of the connector (such as delimiter, compression, and TLS mode) must match the advanced settings of the agent destination that you want to link to the agent.

Importing a file with a list of servers for WMI agents

If you have many servers, you can import a .TSV file with a list of event sources for WMI agents so that you do not have to manually fill in the **Server**, **Domain**, and **Log type** settings for the WMI agent for each server. In such a file, you can prepare a list of servers to which the WMI agent will connect and a list of logs that you want to receive from the specified servers.

Requirements for the file with a list of servers for WMI agents.

Acceptable file format: TSV.

If you change the recommended format to **All files** and upload a file that does not comply with the required format (for example, a JSON file, XML file, or some different format instead of TSV), such a file may be parsed incorrectly.

The maximum number of lines in a file with host settings is 500.

The following fields are required:

- **Server** is the name of the host from which the agent will receive logs. Acceptable format: hostname or IP address. Acceptable values: IPv4, IPv6, host name.
- **Domain** is the host ID, for communication with the host over the network. Acceptable format: fully qualified domain name or IP address. Acceptable values: IPv4, IPv6, FQDN.
- **Log type** is a list of logs and/or Windows subscriptions that the agent will receive. Acceptable format: array of strings. For example, you can specify Application, ForwardedEvents, or Security. If the file specifies a value that was not previously present in the list of logs, the log or subscription must be created that will also appear in the list of available for processing using the Windows agent.

  The maximum length of a log value is 1024 characters. You must specify at least one log or subscription.

After importing and filling in all the fields of the remote hosts, a check is performed when you save settings. If any errors are found in the fields, the settings are not saved, and the errors are highlighted. You can manually update the fields as necessary and save the settings.

To import a file with a list of sources:

1. In the **Resources** → **Agents** section of the LogAnalyzer interface, create a WMI agent or open the WMI agent settings for editing.
2. This opens a window; in that window, on the **Configuration resource #** tab, in the **Connector** section, specify the name of the connector, in the **Type** drop-down list, select **wmi** and click the **Import TSV** button.
3. This opens a window; in that window, select the prepared file with the list of sources in the .TSV format.

   LogAnalyzer parses the file and fills in under **Remote hosts**, fills the **Server**, **Domain**, **Log type** required fields.
4. If necessary, in the **Secret** drop-down list, select a secret of the **Credentials** type or create a secret of the **Credentials** type.

   When importing a list of hosts from a file, in the **Secret** drop-down list, **Default** remains selected, and the agent uses the default credentials when communicating with the server to retrieve logs.

5. If you want to cancel the import of data from the file, click **Cancel**. If you want to delete some data lines imported from a file, in the user interface, select the lines you want to delete and click **Delete**.

6. Save settings in one of the following ways:

   - For a newly created agent, click **Create**. The service is created in the LogAnalyzer web interface.
   - For an active agent opened for editing, click **Save**.

7. Apply the saved settings in one of the following ways:

   - For the newly created agent, install the agent service in the LogAnalyzer network infrastructure.
   - For an active agent service opened for editing, restart the agent service.

The file with a list of servers for WMI agents is imported.

Configuring event sources

This section provides information on configuring the receipt of events from various sources.

Configuring receipt of Auditd events

LogAnalyzer lets you monitor and audit Auditd events on Linux devices.

Before configuring event receiving, make sure to create a new LogAnalyzer collector for Auditd events.

Configuring the receipt of Auditd events involves the following steps:

1. **Configuring the LogAnalyzer collector for receiving Auditd events**
2. **Configuring the event source server**

   You can configure the event source server using the rsyslog or syslog-ng service.
3. **Verifying receipt of Auditd events by the LogAnalyzer collector**

   You can verify that the Auditd event source server is configured correctly by searching for related events in the LogAnalyzer web interface.

Configuring the LogAnalyzer collector for receiving Auditd events

At the **Transport** step, select the **TCP** or **UDP** connector type and move the **Auditd** toggle switch to the enabled position.

After creating a collector, in order to configure event receiving using rsyslog, you must install a collector on the network infrastructure server intended for receiving events.

For details on installing the LogAnalyzer collector, refer to the Installing collector in the network infrastructure section.

Configuring the event source server using the rsyslog service

The rsyslog service is used to transmit events from the server to the collector.

To configure transmission of events from the server to the collector:

1. Make sure that the rsyslog service is installed on the event source server. For this purpose, execute the following command:

   ```
   systemctl status rsyslog.service
   ```

   If the rsyslog service is not installed on the server, install it by executing the following command:

   ```
   yum install rsyslog
   ```

   ```
   systemctl enable rsyslog.service
   ```

   ```
   systemctl start rsyslog.service
   ```

2. Edit the audit.service configuration file, /etc/audit/auditd.conf, and change the values of the following parameters:

   ```
   name_format=NONE
   ```

   ```
   log_format=ENRICHED
   ```

   The `log_format` parameter determines the format of the records generated by the auditd service, and must be set to `ENRICHED` to have user information saved in events.

   After editing the settings, restart the auditd service:

   ```
   sudo systemctl restart auditd.service
   ```

3. In the /etc/rsyslog.d directory, create the audit.conf file with the following content, depending on your protocol:

   - To send events over TCP:

     ```
     $ModLoad imfile
     $InputFileName /var/log/audit/audit.log
     $InputFileTag tag_audit_log:
     $InputFileStateFile audit_log
     $InputFileSeverity info
     $InputFileFacility local6
     $InputRunFileMonitor
     *.* @@<IP address of the KUMA collector>:<port of the KUMA collector>
     For example:
     *.* @@192.1.3.4:5858
     ```

   - To send events over UDP:

     ```
     $ModLoad imfile
     $InputFileName /var/log/audit/audit.log
     $InputFileTag tag_audit_log:
     $InputFileStateFile audit_log
     $InputFileSeverity info
     $InputFileFacility local6
     $InputRunFileMonitor
     template(name="AuditFormat" type="string" string="<%PRI%>%TIMESTAMP:::date-rfc3339%
     %HOSTNAME% %syslogtag% %msg%\n")
     *.* @<IP address of the KUMA collector>:<port of the KUMA collector>
     For example:
     *.* @192.1.3.4:5858;AuditFormat
     ```

4. Save the changes to the audit.conf file.

5. Restart the rsyslog service by executing the following command:

   ```
   systemctl restart rsyslog.service
   ```

   The event source server is configured. Data about events is transmitted from the server to the LogAnalyzer collector.

Configuring receipt of KATA/EDR events

You can configure the receipt of Kaspersky Anti Targeted Attack Platform events in the LogAnalyzer SIEM system.

Before configuring event receipt, make sure to create a LogAnalyzer collector for the KATA/EDR events.

When creating a collector in the LogAnalyzer web interface, make sure that the port number and connector type match the port and protocol specified in the Kaspersky Anti Targeted Attack Platform event export settings in LogAnalyzer.

To receive Kaspersky Anti Targeted Attack Platform events using Syslog, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] KATA** normalizer.

Configuring the receipt of KATA/EDR events involves the following steps:

1. Configuring the forwarding of KATA/EDR events
2. Installing the LogAnalyzer collector in the network infrastructure.
3. Verifying receipt of KATA/EDR events in the LogAnalyzer collector

   You can verify that the KATA/EDR event source server is configured correctly by searching for related events in the LogAnalyzer web interface. Kaspersky Anti Targeted Attack Platform events are displayed as KATA in the table with search results.

Configuring the export of KATA/EDR events to LogAnalyzer

To configure export of events from Kaspersky Anti Targeted Attack Platform to LogAnalyzer:

1. In a browser on any computer with access to the Central Node server, enter the IP address of the server hosting the Central Node component.

   A window for entering Kaspersky Anti Targeted Attack Platform user credentials opens.
2. In the user credentials entry window, select the **Local administrator** check box and enter the Administrator credentials.
3. Go to the **Settings** → **SIEM system** section.
4. Specify the following settings:
   a. Select the **Activity log** and **Detections** check boxes.
   b. In the **Host/IP** field, enter the IP address or host name of the LogAnalyzer collector.
   c. In the **Port** field, specify the port number to connect to the LogAnalyzer collector.
   d. In the **Protocol** field, select **TCP** or **UDP** from the list.
   e. In the **Host ID** field, specify the server host ID to be indicated in the SIEM systems log as a detection source.
   f. In the **Alert frequency** field, enter the interval for sending messages: from 1 to 59 minutes.
   g. Enable TLS encryption, if necessary.
   h. Click **Apply**.

Export of Kaspersky Anti Targeted Attack Platform events to LogAnalyzer is configured.

Creating a LogAnalyzer collector for receiving KATA/EDR events

After configuring the event export settings, you must create a collector for Kaspersky Anti Targeted Attack Platform events in the LogAnalyzer web interface.

For details on creating a LogAnalyzer collector, refer to Creating a collector.

When creating a collector in the LogAnalyzer web interface, make sure that the port number matches the port specified at step 4c of Configuring export of Kaspersky Anti Targeted Attack Platform events to LogAnalyzer, and that the connector type corresponds to the type specified at step 4d.

To receive Kaspersky Anti Targeted Attack Platform events using Syslog, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] KATA** normalizer.

Installing a LogAnalyzer collector for receiving KATA/EDR events

After creating a collector, to configure receiving Kaspersky Anti Targeted Attack Platform events, install a new collector on the network infrastructure server intended for receiving events.

For details on installing the LogAnalyzer collector, refer to the Installing collector in the network infrastructure section.

Configuring the export of Kaspersky Security Center events to the LogAnalyzer SIEM system

LogAnalyzer allows you to receive and export events from the Kaspersky Security Center Administration Server to the LogAnalyzer SIEM system.

Configuring the export and receipt of Kaspersky Security Center events involves of the following steps:

1. Configuring the export of Kaspersky Security Center events.
2. Configuring the LogAnalyzer collector.
3. Installing the LogAnalyzer collector in the network infrastructure.
4. Verifying the receipt of Kaspersky Security Center events by LogAnalyzer.

   You can verify if the events from Kaspersky Security Center Administration Server were correctly exported to the LogAnalyzer SIEM system by using the LogAnalyzer web interface to search for related events.

To display Kaspersky Security Center events in the table, enter the following search expression:

```
SELECT * FROM `events` WHERE DeviceProduct = 'KSC' ORDER BY Timestamp DESC LIMIT 250
```

Configuring export of Kaspersky Security Center events in CEF format

Kaspersky Security Center allows you to configure the settings for exporting events in the CEF format to a SIEM system.

The function of exporting Kaspersky Security Center events in the CEF format to SIEM systems is available with Kaspersky Endpoint Security for Business Advanced license or above.

To configure export of events from Kaspersky Security Center Administration Server to the LogAnalyzer SIEM system:

1. In Kaspersky Security Center console tree, select the **Administration server** node.
2. In the workspace of the node, select the **Events** tab.
3. Click the **Configure notifications and event export** link and select **Configure export to SIEM system** from the drop-down list.

   The **Properties: Events** window opens. By default the **Events export** section is displayed.
4. In the **Events export** section, select the **Automatically export events to SIEM system database** check box.
5. In the **SIEM system** drop-down list select **ArcSight (CEF format)**.
6. In the corresponding fields, specify the address of the LogAnalyzer SIEM system server and the port for connecting to the server. Select **TCP/IP** as the protocol.

   You can click **Export archive** and specify the starting date from which pre-existing LogAnalyzer events are to be exported to the SIEM system database. By default, Kaspersky Security Center exports events starting from the current date.
7. Click **OK**.

As a result, the Kaspersky Security Center Administration Server automatically exports all events to the LogAnalyzer SIEM system.

Configuring a LogAnalyzer collector for collecting Kaspersky Security Center events

After configuring the export of events in the CEF format from Kaspersky Security Center Administration Server, configure the collector in the LogAnalyzer web interface.

To configure the LogAnalyzer Collector for Kaspersky Security Center events:

1. In the LogAnalyzer web interface, select **Resources** → **Collectors**.
2. In the list of collectors, find the collector with the **[OOTB] KSC** normalizer and open it for editing.

3. At the **Transport** step, in the **URL** field, specify the port to be used by the collector to receive Kaspersky Security Center events.

   The port must match the port of the LogAnalyzer SIEM system server.

4. At the **Event parsing** step, make sure that the **[OOTB] KSC** normalizer is selected.

5. At the **Routing** step, make sure that the following destinations are added to the collector resource set:

   - **Storage**. To send processed events to the storage.

   If a **Storage** destination has not been added, create one.

6. At the **Setup validation** tab, click **Create and save service**.

7. Copy the command for installing the LogAnalyzer collector that appears.

Installing a LogAnalyzer collector for obtaining Kaspersky Security Center events

After configuring the collector for collecting Kaspersky Security Center events in the CEF format, install the LogAnalyzer collector on the network infrastructure server intended for receiving events.

For details on installing the LogAnalyzer collector, refer to the Installing collector in the network infrastructure section.

Configuring receiving Kaspersky Security Center event from MS SQL

LogAnalyzer allows you to receive information about Kaspersky Security Center events from an MS SQL database.

Before configuring, make sure that you have created the LogAnalyzer collector for Kaspersky Security Center events from MS SQL.

When creating the collector in the LogAnalyzer web interface, at the **Transport** step, select the **[OOTB] KSC SQL** connector.

To receive Kaspersky Security Center events from the MS SQL database, at the **Event parsing** step, select the **[OOTB] KSC from SQL** normalizer.

Configuring event receiving consists of the following steps:

1. Creating an account in the MS SQL.
2. Configuring the SQL Server Browser service.
3. Creating a secret.

4. Configuring a connector.

5. Installing a collector in the network infrastructure.

6. Verifying receipt of events from MS SQL in the LogAnalyzer collector.

   You can verify that the receipt of events from MS SQL is configured correctly by searching for related events in the LogAnalyzer web interface.

Creating an account in the MS SQL database

To receive Kaspersky Security Center events from MS SQL, a user account is required that has the rights necessary to connect and work with the database.

To create an account for working with MS SQL:

1. Log in to the server with MS SQL for Kaspersky Security Center installed.

2. Using **SQL Server Management Studio**, connect to MS SQL using an account with administrator rights.

3. In the Object Explorer pane, expand the **Security** section.

4. Right-click the **Logins** folder and select **New Login** from the context menu.

   The **Login - New** window opens.

5. On the **General** tab, click the **Search** button next to the **Login name** field.

   The **Select User or Group** window opens.

6. In the **Enter the object name to select (examples)** field, specify the object name and click **OK**.

   The **Select User or Group** window closes.

7. In the **Login - New** window, on the **General** tab, select the **Windows authentication** option.

8. In the **Default database** field, select the Kaspersky Security Center database.

   The default Kaspersky Security Center database name is KAV.

9. On the **User Mapping** tab, configure the account permissions:

   a. In the **Users mapped to this login** section, select the Kaspersky Security Center database.

   b. In the **Database role membership for** section, select the check boxes next to the **db_datareader** and **public** permissions.

10. On the **Status** tab, configure the permissions for connecting the account to the database:

    - In the **Permission to connect to database engine** section, select **Grant**.

    - In the **Login** section, select **Enabled**.

11. Click **OK**.

    The **Login - New** window closes.

To check the account permissions:

1. Run **SQL Server Management Studio** using the created account.
2. Go to any MS SQL database table and make a selection based on the table.

Configuring the SQL Server Browser service

After creating an account in MS SQL, you must configure the SQL Server Browser service.

To configure the SQL Server Browser service:

1. Open **SQL Server Configuration Manager**.
2. In the left pane, select **SQL Server Services**.
   A list of services opens.
3. Open the **SQL Server Browser** service properties in one of the following ways:
   - Double-click the name of the **SQL Server Browser** service.
   - Right-click the name of the **SQL Server Browser** service and select **Properties** from the context menu.
4. In the **SQL Server Browser Properties** window that opens, select the **Service** tab.
5. In the **Start Mode** field, select **Automatic**.
6. Select the **Log On** tab and click the **Start** button.
   Automatic startup of the **SQL Server Browser** service is enabled.
7. Enable and configure the **TCP/IP** protocol by doing the following:
   a. In the left pane, expand the **SQL Server Network Configuration** section and select the **Protocols for <SQL Server name>** subsection.
   b. Right-click the **TCP/IP** protocol and select **Enable** from the context menu.
   c. In the **Warning** window that opens, click **OK**.
   d. Open the **TCP/IP** protocol properties in one of the following ways:
      - Double-click the **TCP/IP** protocol.
      - Right-click the **TCP/IP** protocol and select **Properties** from the context menu.
   e. Select the **IP Addresses** tab, and then in the **IPALL** section, specify port 1433 in the **TCP Port** field.
   f. Click **Apply** to save the changes.
   g. Click **OK** to close the window.

8. Restart the **SQL Server (<SQL Server name>)** service by doing the following:

   a. In the left pane, select **SQL Server Services**.

   b. In the service list on the right, right-click the **SQL Server (<SQL Server name>)** service and select **Restart** from the context menu.

9. In **Windows Defender Firewall with Advanced Security**, allow inbound connections on the server on the TCP port 1433.

Creating a secret in LogAnalyzer

After creating and configuring an account in MS SQL, you must add a secret in the LogAnalyzer web interface. This resource is used to store credentials for connecting to MS SQL.

To create a LogAnalyzer secret:

1. In the LogAnalyzer web interface, open **Resources** → **Secrets**.

   The list of available secrets will be displayed.

2. Click the **Add secret** button to create a new secret.

   The secret window is displayed.

3. Enter information about the secret:

   a. In the **Name** field, choose a name for the added secret.

   b. In the **Tenant** drop-down list, select the tenant that will own the created resource.

   c. In the **Type** drop-down list, select **urls**.

   d. In the **URL** field, specify a string of the following form:
   `sqlserver://[<domain>%5C]<username>:<password>@<server>:1433/<database_name>`

   where:

   - `domain` is a domain name.
   - `%5C` is the domain/user separator. Represents the "\" character in URL format.
   - `username` is the name of the created MS SQL account.
   - `password` is the password of the created MS SQL account.
   - `server` is the name or IP address of the server where the MS SQL database for Kaspersky Security Center is installed.
   - `database_name` is the name of the Kaspersky Security Center database. The default name is KAV.

   Example:
   `sqlserver://test.local%5Cuser:password123@10.0.0.1:1433/KAV`

   If the MS SQL database account password contains special characters (@ # $ % & * ! + = [ ] : ' , ? / \ ` ( ) ;), convert them to URL format.

4. Click **Save**.

For security reasons, the string specified in the URL field is hidden after the secret is saved.

Configuring a connector

To connect LogAnalyzer to an MS SQL database, you must configure the connector.

To configure a connector:

1. In the LogAnalyzer web interface, select **Resources** → **Connectors**.
2. In the list of connectors, find the **[OOTB] KSC SQL** connector and open it for editing.

   If a connector is not available for editing, copy it and open the connector copy for editing.

   If the **[OOTB] KSC SQL** connector is not available, contact your system administrator.
3. On the **Basic settings** tab, in the **URL** drop-down lists, select the secret created for connecting to the MS SQL database.
4. Click **Save**.

Configuring a LogAnalyzer collector for receiving Kaspersky Security Center events from an MS SQL database

After configuring the event export settings, you must create a collector in the LogAnalyzer web interface for Kaspersky Security Center events received from MS SQL.

For details on creating a LogAnalyzer collector, refer to Creating a collector.

When creating the collector in the LogAnalyzer web interface, at the **Transport** step, select the **[OOTB] KSC SQL** connector.

To receive Kaspersky Security Center events from MS SQL, at the **Event parsing** step, select the **[OOTB] KSC from SQL** normalizer.

Installing a LogAnalyzer collector for receiving Kaspersky Security Center events from the MS SQL database

After configuring the collector for receiving Kaspersky Security Center events from MS SQL, install the LogAnalyzer collector on the network infrastructure server where you intend to receive events.

For details on installing the LogAnalyzer collector, refer to the Installing collector in the network infrastructure section.

Configuring the receipt of events from Windows devices using LogAnalyzer Agent (WEC)

LogAnalyzer allows you to receive information about events from Windows devices using the WEC LogAnalyzer Agent.

Configuring event receiving consists of the following steps:

1. Configuring policies for receiving events from Windows devices.
2. Configuring centralized receipt of events using the Windows Event Collector service.
3. Granting permissions to view events.
4. Granting permissions to log on as a service.
5. Configuring the LogAnalyzer collector.
6. Installing the LogAnalyzer collector.
7. Forwarding events from Windows devices to LogAnalyzer.

Configuring audit of events from Windows devices

You can configure event audit on Windows devices for an individual device or for all devices in a domain.

This section describes how to configure an audit on an individual device and how to use a domain group policy to configure an audit.

Configuring an audit policy on a Windows device

To configure audit policies on a device:

1. Open the **Run** window by pressing the key combination **Win**+**R**.
2. In the opened window, type `secpol.msc` and click **OK**.

   The **Local security policy** window opens.
3. Select **Security Settings** → **Local policies** → **Audit policy**.
4. In the pane on the right, double-click to open the properties of the policy for which you want to enable an audit of successful and unsuccessful attempts.
5. In the **<Policy name> properties** window, on the **Local security setting** tab, select the **Success** and **Failure** check boxes to track successful and interrupted attempts.

   It is recommended to enable an audit of successful and unsuccessful attempts for the following policies:

   - Audit Logon
   - Audit Policy Change
   - Audit System Events
   - Audit Logon Events
   - Audit Account Management

Configuration of an audit policy on the device is complete.

Configuring an audit using a group policy

In addition to configuring an audit policy on an individual device, you can also configure an audit by using a domain group policy.

To configure an audit using a group policy:

1. Open the **Run** window by pressing the key combination **Win**+**R**.
2. In the opened window, type `gpedit.msc` and click **OK**.

   The **Local Group Policy Editor** window opens.
3. Select **Computer configuration** → **Windows configuration** → **Security settings** → **Local policies** → **Audit policy**.

4. In the pane on the right, double-click to open the properties of the policy for which you want to enable an audit of successful and unsuccessful attempts.

5. In the **<Policy name> properties** window, on the **Local security setting** tab, select the **Success** and **Failure** check boxes to track successful and interrupted attempts.

   It is recommended to enable an audit of successful and unsuccessful attempts for the following policies:

   - Audit Logon
   - Audit Policy Change
   - Audit System Events
   - Audit Logon Events
   - Audit Account Management

If you want to receive Windows logs from a large number of servers or if installation of LogAnalyzer agents on domain controllers is not allowed, it is recommended to configure Windows log redirection to individual servers that have the Windows Event Collector service configured.

The audit policy is now configured on the server or workstation.

Configuring centralized receipt of events from Windows devices using the Windows Event Collector service

The Windows Event Collector service allows you to centrally receive data about events on servers and workstations running Windows. You can use the Windows Event Collector service to subscribe to events that are registered on remote devices.

You can configure the following types of event subscriptions:

- **Source-initiated subscriptions**. Remote devices send event data to the Windows Event Collector server whose address is specified in the group policy. For details on the subscription configuration procedure, please refer to the Configuring data transfer from the event source server section.
- **Collector-initiated subscriptions**. The Windows Event Collector server connects to remote devices and independently gathers events from local logs. For details on the subscription configuration procedure, please refer to the Configuring the Windows Event Collector service section.

Configuring data transfer from the event source server

You can receive information about events on servers and workstations by configuring data transfer from remote devices to the Windows Event Collector server.

Preliminary steps

1. Verify that the Windows Remote Management service is configured on the event source server by running the following command in the PowerShell console:

   ```
   winrm get winrm/config
   ```

   If the Windows Remote Management service is not configured, initialize it by running the following command:

   ```
   winrm quickconfig
   ```

2. If the event source server is a domain controller, make the Windows logs available over the network by running the following command in PowerShell as an administrator:

   ```
   wevtutil set-log security /ca:'O:BAG:SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-5-32-573)(A;;0x1;;;S-1-5-20)'
   ```

   Verify access by running the following command:

   ```
   wevtutil get-log security
   ```

Configuring the firewall on the event source server

To enable the Windows Event Collector server to receive Windows log entries, inbound connection ports must be opened on the event source server.

To open ports for inbound connections:

1. On the event source server, open the **Run** window by pressing the key combination **Win**+**R**.

2. In the opened window, type `wf.msc` and click **OK**.

   The **Windows Defender Firewall with Advanced Security** window opens.

3. Go to the **Inbound Rules** section and click **New Rule** in the **Actions** pane.

   The **New Inbound Rule Wizard** opens.

4. At the **Rule type** step, select **Port**.

5. At the **Protocols and ports** step, select **TCP** as the protocol. In the **Specific local ports** field, indicate the relevant port numbers:

   - `5985` (for HTTP access)
   - `5986` (for HTTPS access)

   You can indicate one of the ports, or both.

6. At the **Action** step, select **Allow connection** (selected by default).

7. At the **Profile** step, clear the **Private** and **Public** check boxes.

8. At the **Name** step, specify a name for the new inbound connection rule and click **Done**.

Configuration of data transfer from the event source server is complete.

The Windows Event Collector server must have the permissions to read Windows logs on the event source server. These permissions can be assigned to both the Windows Event Collector server account and to a special user account. For details on granting permissions, please refer to the Granting user permissions to view the Windows Event Log.

Configuring the Windows Event Collector service

The Windows Event Collector server can independently connect to devices and gather data on events of any severity.

To configure the receipt of event data by the Windows Event Collector server:

1. On the event source server, open the **Run** window by pressing **Win**+**R**.
2. In the opened window, type `services.msc` and click **OK**.

   The **Services** window opens.
3. In the list of services, find and start the **Windows Event Collector** service.
4. Open the **Event Viewer** snap-in by doing the following:

   a. Open the **Run** window by pressing the key combination **Win**+**R**.

   b. In the opened window, type `eventvwr` and click **OK**.
5. Go to the **Subscriptions** section and click **Create Subscription** in the **Actions** pane.

6. In the opened **Subscription Properties** window, specify the name and description of the subscription, and define the following settings:

   a. In the **Destination log** field, select **ForwardedEvents** from the list.

   b. In the **Subscription type and source computers** section, click the **Select computers** button.

   c. In the opened **Computers** window, click the **Add domain computer** button.

   The **Select computer** window opens.

   d. In the **Enter the object names to select (examples)** field, list the names of the devices from which you want to receive event information. Click **OK**.

   e. In the **Computers** window, check the list of devices from which the Windows Event Collector server will gather event data and click **OK**.

   f. In the **Subscription properties** window, in the **Collected events** field, click the **Select events** button.

   g. In the opened **Request filter** window, specify how often and which data about events on devices you want to receive.

   h. If necessary, in the **<All event codes>** field, list the codes of the events whose information you want to receive or do not want to receive. Click **OK**.

7. If you want to use a special account to view event data, do the following:

   a. In the **Subscription properties** window, click the **Advanced** button.

   b. In the opened **Advanced subscription settings** window, in the user account settings, select **Specific user**.

   c. Click the **User and password** button and enter the account credentials of the selected user.

Configuration of the Event Collector Service is complete.

To verify that the configuration is correct and event data is being received by the Windows Event Collector server:

In the **Event Viewer** snap-in, go to **Event Viewer (Local)** → **Windows logs** → **ForwardedEvents**.

Granting permissions to view Windows events

You can grant permissions to view Windows events for a specific device or for all devices in a domain.

To grant permissions to view events on a specific device:

1. Open the **Run** window by pressing the key combination **Win**+**R**.

2. In the opened window, type `compmgmt.msc` and click **OK**.

   The **Computer Management** window opens.

3. Go to **Computer Management (local)** → **Local users and groups** → **Groups**.

4. In the pane on the right, select the **Event Log Readers** group and double-click to open the policy properties.

5. Click the **Add** button at the bottom of the **Properties: Event Log Readers** window.

   The **Select Users, Computers or Groups** window opens.

6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant permissions to view event data. Click **OK**.

To grant permissions to view events for all devices in a domain:

1. Log in to the domain controller with administrator privileges.
2. Open the **Run** window by pressing the key combination **Win**+**R**.
3. This opens a window; in that window, type `dsa.msc` and click **OK**.

   The **Active Directory Users and Computers** window opens.

4. Go to **Active Directory Users and Computers** → **<Domain name>** → **Builtin**.
5. In the pane on the right, select the **Event Log Readers** group and double-click to open the policy properties.

   In the **Properties: Event Log Readers** window, open the **Members** tab and click the **Add** button.

   The **Select Users, Computers or Groups** window opens.

6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant permissions to view event data. Click **OK**.

Granting permissions to log on as a service

You can grant permission to log on as a service to a specific device or to all devices in a domain. The "Log on as a service" permission allows you to start a process using an account that has been granted this permission.

To grant the "Log on as a service" permission to a device:

1. Open the **Run** window by pressing the key combination **Win**+**R**.
2. In the opened window, type `secpol.msc` and click **OK**.

   The **Local security policy** window opens.

3. Go to **Security settings** → **Local policies** → **User rights assignment**.
4. In the pane on the right, double-click to open the properties of the **Log on as a service** policy.
5. In the opened **Properties: Log on as a Service** window, click the **Add User or Group** button.

   The **Select Users or Groups** window opens.

6. In the **Enter the object names to select (examples)** field, list the names of the accounts or devices to which you want to grant the permission to log on as a service. Click **OK**.

Before granting the permission, make sure that the accounts or devices to which you want to grant the **Log on as a service** permission are not listed in the properties of the **Deny log on as a service** policy.

To grant the "Log on as a service" permission to devices in a domain:

1. Open the **Run** window by pressing the key combination **Win+R**.
2. In the opened window, type `gpedit.msc` and click **OK**.

   The **Local Group Policy Editor** window opens.
3. Select **Computer configuration** → **Windows configuration** → **Security settings** → **Local policies** → **User rights assignment**.
4. In the pane on the right, double-click to open the properties of the **Log on as a service** policy.
5. In the opened **Properties: Log on as a Service** window, click the **Add User or Group** button.

   The **Select Users or Groups** window opens.
6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant the permission to log on as a service. Click **OK**.

Before granting the permission, make sure that the accounts or devices to which you want to grant the **Log on as a service** permission are not listed in the properties of the **Deny log on as a service** policy.

Configuring a LogAnalyzer collector for receiving events from Windows devices

After you finish configuring the audit policy on devices, creating subscriptions to events and granting all the necessary permissions, you need to create a collector in the LogAnalyzer web interface for events from Windows devices.

For details on creating a LogAnalyzer collector, refer to Creating a collector.

To receive events from Windows devices, define the following collector settings in the LogAnalyzer Collector Installation Wizard:

1. At the **Transport** step, define the following settings:

   a. In the **Connector** window, select **Create**.

   b. In the **Type** field, select **http**.

   c. In the **Delimiter** field, select **\0**.
2. On the **Advanced settings** tab, in the **TLS mode** field, select **With verification**.
3. At the **Event parsing** step, click the **Add event parsing** button.
4. This opens the **Basic event parsing** window; in that window, in the **Normalizer** field, select **[OOTB] Microsoft Products** and click **OK**.

5. At the **Routing** step, add the following destinations:

   - **Storage**. To send processed events to the storage.

   If a **Storage** destination has not been added, create one.

6. At the **Setup validation** tab, click **Create and save service**.

7. Copy the command for installing the LogAnalyzer collector that appears.

Installing a LogAnalyzer collector for receiving events from Windows devices

After configuring the collector for receiving Windows events, install the LogAnalyzer collector on the server of the network infrastructure intended for receiving events.

For details on installing the LogAnalyzer collector, refer to the Installing collector in the network infrastructure section.

Configuring the export of of events from Windows devices to LogAnalyzer using LogAnalyzer Agent (WEC)

To complete the data forwarding configuration, you must create a WEC LogAnalyzer agent and then install it on the device from which you want to receive event information.

For more details on creating and installing a WEC LogAnalyzer Agent on Windows devices, please refer to the Forwarding events from Windows devices to LogAnalyzer section.

Configuring the receipt of events from Windows devices using LogAnalyzer Agent (WMI)

LogAnalyzer allows you to receive information about events from Windows devices using the WMI LogAnalyzer Agent.

Configuring event receiving consists of the following steps:

1. Configuring audit settings for managing LogAnalyzer.

2. Configuring data transfer from the event source server.

3. Granting permissions to view events.

4. Granting permissions to log on as a service.

5. Creating a LogAnalyzer collector.

   To receive Windows device events, in the LogAnalyzer Collector Installation Wizard, at the **Event parsing** step, in the **Normalizer** field, select **[OOTB] Microsoft Products**.

6. Installing the LogAnalyzer collector.

7. Forwarding events from Windows devices to LogAnalyzer.

   To complete the data forwarding configuration, you must create a WMI LogAnalyzer agent and then install it on the device from which you want to receive event information.

Managing audit settings for managing LogAnalyzer.

You can configure event audit on Windows devices both on a specific device using a local policy or on all devices in a domain using a group policy.

This section describes how to configure an audit on an individual device and how to use a domain group policy to configure an audit.

Configuring an audit using a local policy

To configure an audit using a local policy:

1. Open the **Run** window by pressing the key combination **Win+R**.

2. In the opened window, type `secpol.msc` and click **OK**.

   The **Local security policy** window opens.

3. Select **Security Settings** → **Local policies** → **Audit policy**.

4. In the pane on the right, double-click to open the properties of the policy for which you want to enable an audit of successful and unsuccessful attempts.

5. In the **<Policy name> properties** window, on the **Local security setting** tab, select the **Success** and **Failure** check boxes to track successful and interrupted attempts.

   It is recommended to enable an audit of successful and unsuccessful attempts for the following policies:

   - Audit Logon
   - Audit Policy Change
   - Audit System Events
   - Audit Logon Events
   - Audit Account Management

Configuration of an audit policy on the device is complete.

Configuring an audit using a group policy

In addition to configuring an audit on an individual device, you can also configure an audit by using a domain group policy.

To configure an audit using a group policy:

1. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.

2. In the opened window, type `gpedit.msc` and click **OK**.

   The **Local Group Policy Editor** window opens.

3. Select **Computer configuration** → **Windows configuration** → **Security settings** → **Local policies** → **Audit policy**.

4. In the pane on the right, double-click to open the properties of the policy for which you want to enable an audit of successful and unsuccessful attempts.

5. In the **<Policy name> properties** window, on the **Local security setting** tab, select the **Success** and **Failure** check boxes to track successful and interrupted attempts.

   It is recommended to enable an audit of successful and unsuccessful attempts for the following policies:

   - Audit Logon
   - Audit Policy Change
   - Audit System Events
   - Audit Logon Events
   - Audit Account Management

The audit policy is now configured on the server or workstation.

Configuring data transfer from the event source server

Preliminary steps

1. On the event source server, open the **Run** window by pressing the key combination **Win**+**R**.
2. In the opened window, type `services.msc` and click **OK**.

   The **Services** window opens.
3. In the list of services, find the following services:
   - Remote Procedure Call
   - RPC Endpoint Mapper
4. Check the **Status** column to confirm that these services have the **Running** status.

Configuring the firewall on the event source server

The Windows Management Instrumentation server can receive Windows log entries if ports are open for inbound connections on the event source server.

To open ports for inbound connections:

1. On the event source server, open the **Run** window by pressing the key combination **Win**+**R**.
2. In the opened window, type `wf.msc` and click **OK**.

   The **Windows Defender Firewall with Advanced Security** window opens.
3. In the **Windows Defender Firewall with Advanced Security** window, go to the **Inbound Rules** section and in the **Actions** pane, click **New Rule**.

   This opens the **New Inbound Rule Wizard**.
4. In the **New Inbound Rule Wizard,** at the **Rule Type** step, select **Port**.
5. At the **Protocols and ports** step, select **TCP** as the protocol. In the **Specific local ports** field, indicate the relevant port numbers:
   - `135`
   - `445`
   - `49152-65535`
6. At the **Action** step, select **Allow connection** (selected by default).
7. At the **Profile** step, clear the **Private** and **Public** check boxes.
8. At the **Name** step, specify a name for the new inbound connection rule and click **Done**.

Configuration of data transfer from the event source server is complete.

Granting permissions to view Windows events

You can grant permissions to view Windows events for a specific device or for all devices in a domain.

To grant permissions to view events on a specific device:

1. Open the **Run** window by pressing the key combination **Win**+**R**.
2. In the opened window, type `compmgmt.msc` and click **OK**.

   The **Computer Management** window opens.
3. Go to **Computer Management (local)** → **Local users and groups** → **Groups**.
4. In the pane on the right, select the **Event Log Readers** group and double-click to open the policy properties.
5. Click the **Add** button at the bottom of the **Properties: Event Log Readers** window.

   The **Select Users, Computers or Groups** window opens.
6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant permissions to view event data. Click **OK**.

To grant permissions to view events for all devices in a domain:

1. Log in to the domain controller with administrator privileges.
2. Open the **Run** window by pressing the key combination **Win**+**R**.
3. This opens a window; in that window, type `dsa.msc` and click **OK**.

   The **Active Directory Users and Computers** window opens.
4. In the **Active Directory Users and Computers** window, go to the **Active Directory Users and Computers** section → **<Domain name>** → **Builtin**.
5. In the pane on the right, select the **Event Log Readers** group and double-click to open the policy properties.

   In the **Properties: Event Log Readers** window, open the **Members** tab and click the **Add** button.

   The **Select Users, Computers or Groups** window opens.
6. In the **Select User, Computer, or Group** window, In the **Enter the object name to select (examples)** field, list the names of the users or devices to which you want to grant permissions to view event data. Click **OK**.

Granting permissions to log on as a service

You can grant permission to log on as a service to a specific device or to all devices in a domain. The "Log on as a service" permission allows you to start a process using an account that has been granted this permission.

Before granting the permission, make sure that the accounts or devices to which you want to grant the **Log on as a service** permission are not listed in the properties of the **Deny log on as a service** policy.

To grant the "Log on as a service" permission to a device:

1. Open the **Run** window by pressing the key combination **Win+R**.
2. In the opened window, type `secpol.msc` and click **OK**.

   The **Local security policy** window opens.
3. In the **Local Security Policy** window, go to the **Security Settings** → **Local Policies** → **User Rights Assignment** section.
4. In the pane on the right, double-click to open the properties of the **Log on as a service** policy.
5. This opens the **Properties: Log on as a Service** window; in that window, click **Add User or Group**.

   This opens the **Select Users or Groups** window.
6. In the **Enter the object names to select (examples)** field, list the names of the accounts or devices to which you want to grant the permission to log on as a service. Click **OK**.

To grant the "Log on as a service" permission to devices in a domain:

1. Open the **Run** window by pressing the key combination **Win+R**.
2. In the opened window, type `gpedit.msc` and click **OK**.

   The **Local Group Policy Editor** window opens.
3. Select **Computer configuration** → **Windows configuration** → **Security settings** → **Local policies** → **User rights assignment**.
4. In the pane on the right, double-click to open the properties of the **Log on as a service** policy.
5. This opens the **Properties: Log on as a Service** window; in that window, click **Add User or Group**.

   This opens the **Select Users or Groups** window.
6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant the permission to log on as a service. Click **OK**.

Configuring receipt of DNS server events using the ETW agent

The *Event Tracing for Windows* connector (hereinafter also referred to as the ETW connector) is a mechanism for logging events generated by applications and drivers on the DNS server. You can use the ETW connector to troubleshoot errors during development or to look for malicious activity.

The impact of the ETW connector on DNS server performance is insignificant. For example, a DNS server running on modern hardware and getting up to 100,000 queries per second (QPS) may experience a 5% performance drop while using the ETW connector. If the DNS server gets up to 50,000 requests per second, no performance drop is observed. We recommend monitoring DNS server performance when using the ETW connector, regardless of the number of requests per second.

By default, you can use the ETW connector on Windows Server 2016 or later. The ETW connector is also supported by Windows Server 2012 R2 if the update for event logging and change auditing is installed. The update is available on the Microsoft Support website.

The ETW connector consists of the following components:

- *Providers* are elements of the system that generate events and send them to the ETW connector. For example, Windows kernels or device drivers can act as providers. When working with code, developers must specify which events the providers must send to the ETW connector. An event may represent the execution of a function that the developer considers important, for example, a function that allows access to the Security Account Manager (SAM).

- *Consumers* are software systems that receive events generated by providers from the ETW connector, and use these events in some way. For example, LogAnalyzer can act as a consumer.

- *Controllers* are pieces of software that manage the interaction between providers and consumers. For example, the Logman or Wevtutil utilities can be controllers. Providers register with the controller to send events to consumers. The controller can enable or disable a provider. If a provider is disabled, it does not generate events.

Controllers use trace sessions for communication between providers and consumers. Trace sessions are also used for filtering data based on specified parameters because consumers may need different events.

Configuring DNS server event reception using the ETW connector involves the following steps:

1. Configuration on the Windows side.

2. Creating a LogAnalyzer collector.

   When creating a LogAnalyzer collector, follow these steps:

   a. At step 2 of the Collector Installation Wizard:

      1. In the **Type** drop-down list, select the **tcp** connector type. You can also specify the **http** connector type and other connector types with verification for secure transmission.

      2. In the **URL** field, enter the FQDN and port number on which the LogAnalyzer collector will listen for a connection from the LogAnalyzer agent. You can specify any unoccupied port number.

      3. In the **Delimiter** field, enter `\n`.

   b. At step 3 of the Collector Installation Wizard, in the **Normalizer** drop-down list, select a normalizer. We recommend selecting the predefined extended normalizer for Windows events, **[OOTB] Microsoft DNS ETW logs json**.

   c. At step 7 of the Collector Installation Wizard, add a **Storage** type destination for storing events.

   d. At step 8 of the Collector Installation Wizard, click **Create and save service**, and in the lower part of the window, copy the command for installing the LogAnalyzer collector on the server.

3. Installing the LogAnalyzer collector on the server.

   Do the following:

   a. Connect to the LogAnalyzer command line interface using a user account with root privileges.

   b. Install the LogAnalyzer collector by running the command that you copied at step 8 of the Collector Installation Wizard.

   c. If you want to add the LogAnalyzer collector port to the firewall exclusions and update the firewall settings, run the following commands:

      1. `firewall-cmd --add-port=< collector port number >/tcp --permanent`

      2. `firewall-cmd --reload`

   The LogAnalyzer collector is installed and the status of the LogAnalyzer collector service changes to green in the LogAnalyzer web interface.

4. Creating a LogAnalyzer agent.

   When creating a LogAnalyzer agent, follow these steps:

   a. Go to the **Connection 1** tab.

   b. Under **Connector**, in the **Connector** drop-down list, select **Create** and specify the following settings:

      1. In the **Type** drop-down list, select the **etw** connector type.

      2. In the **Session name** field, enter the provider name that you specified when you configured the reception of DNS server events using the ETW connector on the Windows side.

   c. Under **Destinations**, in the **Destination** drop-down list, select **Create** and specify the following settings:

      1. In the **Type** drop-down list, select the **tcp** destination type.

      2. In the **URL** field, enter the FQDN and port number on which the LogAnalyzer collector will listen for a connection from the LogAnalyzer agent. The value must match the value that you specified at step 2 of the Collector Installation Wizard.

   d. Go to the **Advanced settings** tab, and in the **Disk buffer size limit** limit field, enter `1073741824`.

5. Creating a LogAnalyzer agent service.

   You need to copy the ID of the created LogAnalyzer agent service. To do so, right-click next to the LogAnalyzer agent service and select **Copy ID** in the context menu.

6. Creating an account for the LogAnalyzer agent.

   Create a domain or local Windows user account for running the LogAnalyzer agent and reading the analytic log. You need to add the created user account to the **Performance Log Users** group and grant the **Log on service** permission (**Log on as a service**).

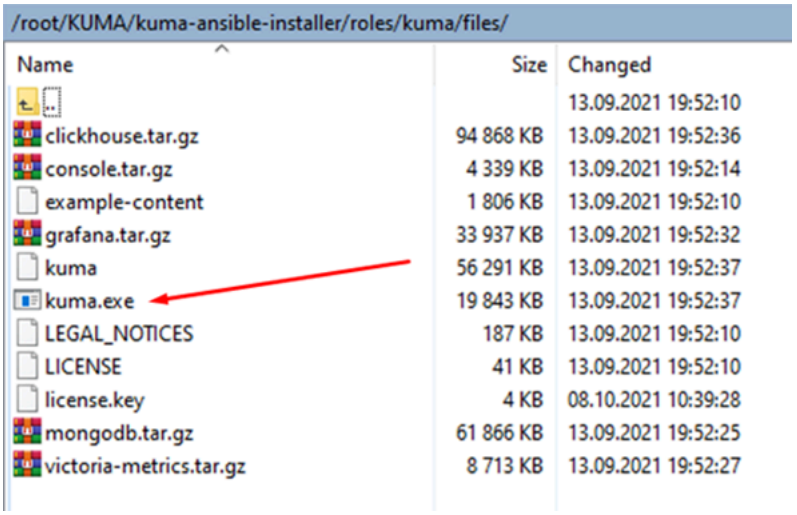
7. Installing a LogAnalyzer agent on a Windows server.

You need to install the LogAnalyzer agent on the Windows server that will be receiving events from the provider. To do so:

a. Add the FQDN of the LogAnalyzer Core server to the hosts file on the Windows server or to the DNS server.

b. Create the C:\Users\<user name>\Desktop\KUMA folder on the Windows server.

c. Copy the kuma.exe file from the LogAnalyzer installation package archive to the C:\Users\<user name>\Desktop\KUMA folder.



d. Run the command interpreter as administrator.

e. Change to the C:\Users\<user name>\Desktop\KUMA folder and run the following command:

```
C:\Users\<user name>\Desktop\KUMA>kuma.exe agent --core https://<DOMAIN-NAME-KUMA-CORE-Server>:7210 --id <LogAnalyzer agent service ID>
```

In the LogAnalyzer web interface, in the **Resources** → **Active services** section, make sure that the LogAnalyzer agent service is running and its status is now green, and then abort the command.

f. Start the LogAnalyzer Agent installation in one of the following ways:

- If you want to start the LogAnalyzer agent installation using a domain user account, run the following command:

  ```
  C:\Users\<user name>\Desktop\KUMA>kuma.exe agent --core https://<DOMAIN-NAME-KUMA-CORE-Server>:7210 --id <LogAnalyzer agent service ID> --user <domain>\<user account name for the LogAnalyzer agent> --install
  ```

- If you want to start the agent installation using a local user account, run the following command:

  ```
  C:\Users\<user name>\Desktop\KUMA>kuma.exe agent --core https://<DOMAIN-NAME-KUMA-CORE-Server>:7210 --id <LogAnalyzer agent service ID> --user <user account name for the LogAnalyzer agent> --install
  ```

You will need to enter the password of the LogAnalyzer agent user account.

The LogAnalyzer Windows Agent service <LogAnalyzer agent service ID> is installed on the Windows server. In the LogAnalyzer web interface, in the **Resources** → **Active services** section, if the LogAnalyzer agent service is not running and has the red status, you need to make sure that port 7210 is available, as well as the Windows collector port in the direction from the LogAnalyzer agent to the LogAnalyzer collector.

To uninstall the LogAnalyzer agent service on the Windows server, run the following command:

```
C:\Users\<user name>\Desktop\KUMA>kuma.exe agent --id <LogAnalyzer agent service ID> --uninstall
```

8. Verifying receipt of DNS server events in the LogAnalyzer collector.

   You can verify that you have correctly configured the reception of DNS server events using the ETW connector in the Searching for related events section of the LogAnalyzer web interface.

Configuration on the Windows side

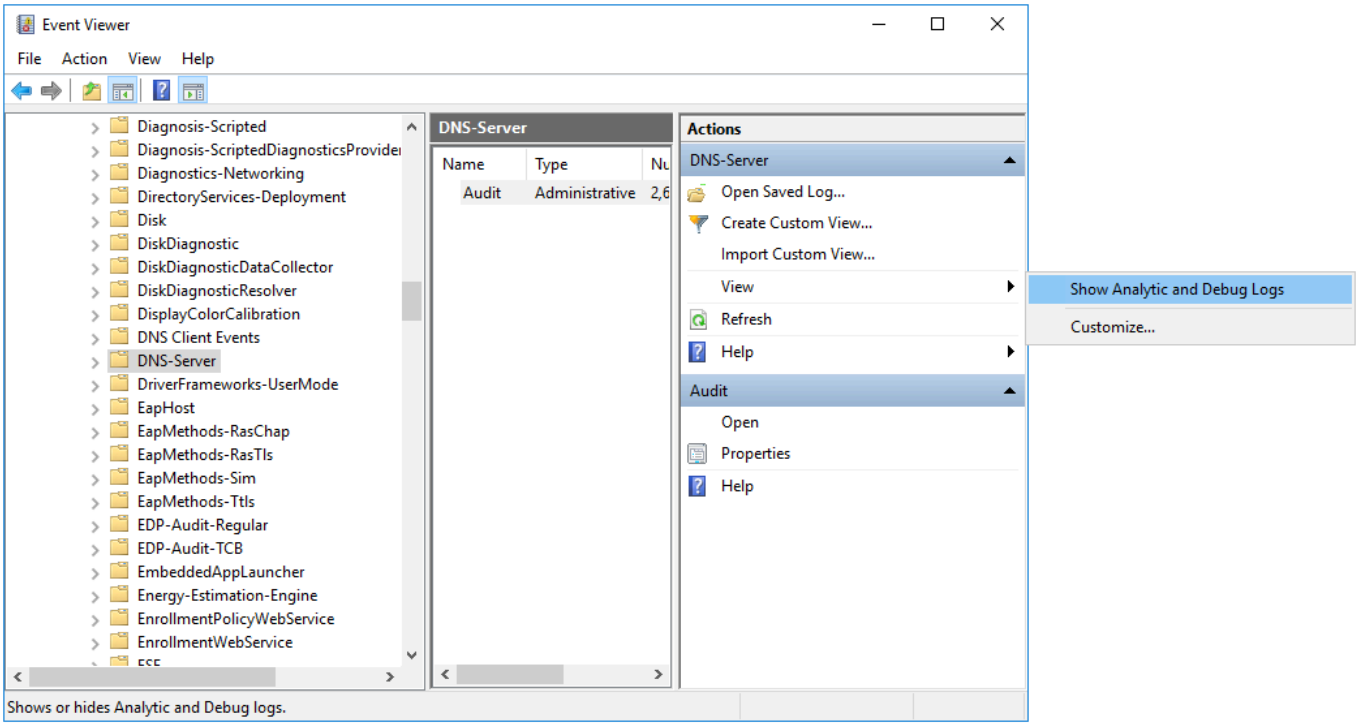
To configure the reception of DNS server events using the ETW connector on the Windows side:

1. Start the Event viewer by running the following command:

   ```
   eventvwr.msc
   ```

2. This opens a window; in that window, go to the **Applications and Services Logs** → **Microsoft** → **Windows** → **DNS-Server** folder.

3. Open the context menu of the **DNS-Server** folder and select **View** → **Show Analytic and Debug Logs**.

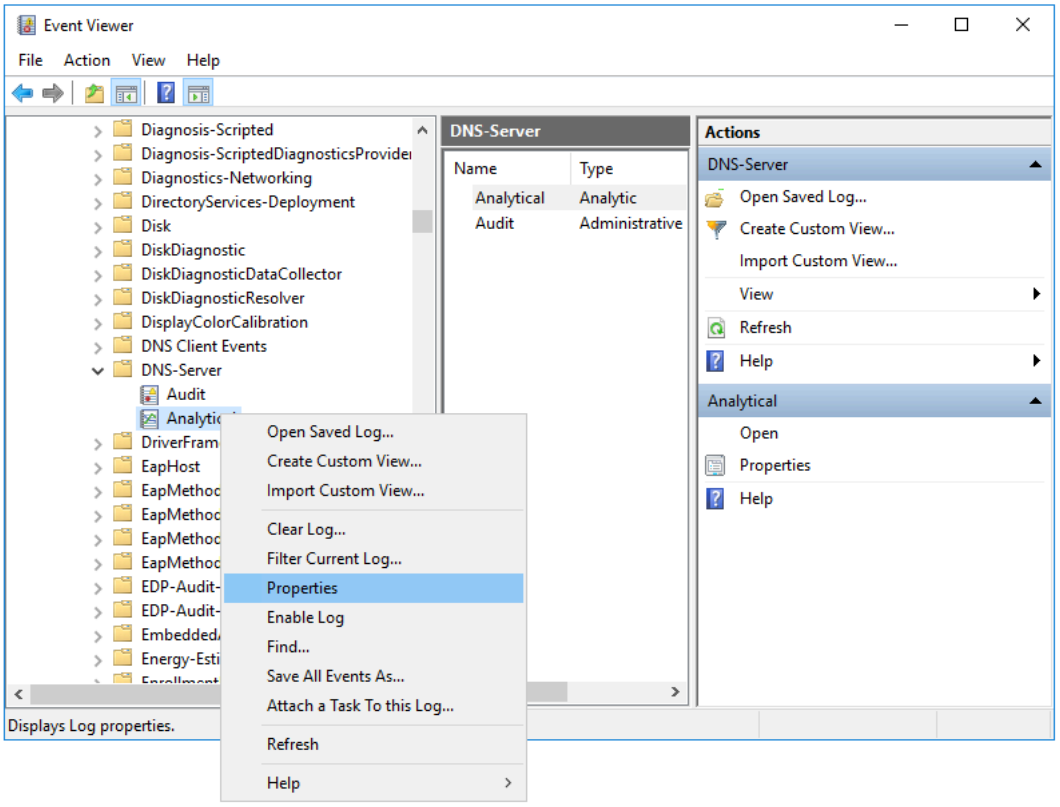


Displaying the analytical and debug logs

   The **Audit** debug log and **Analytical** log are displayed.

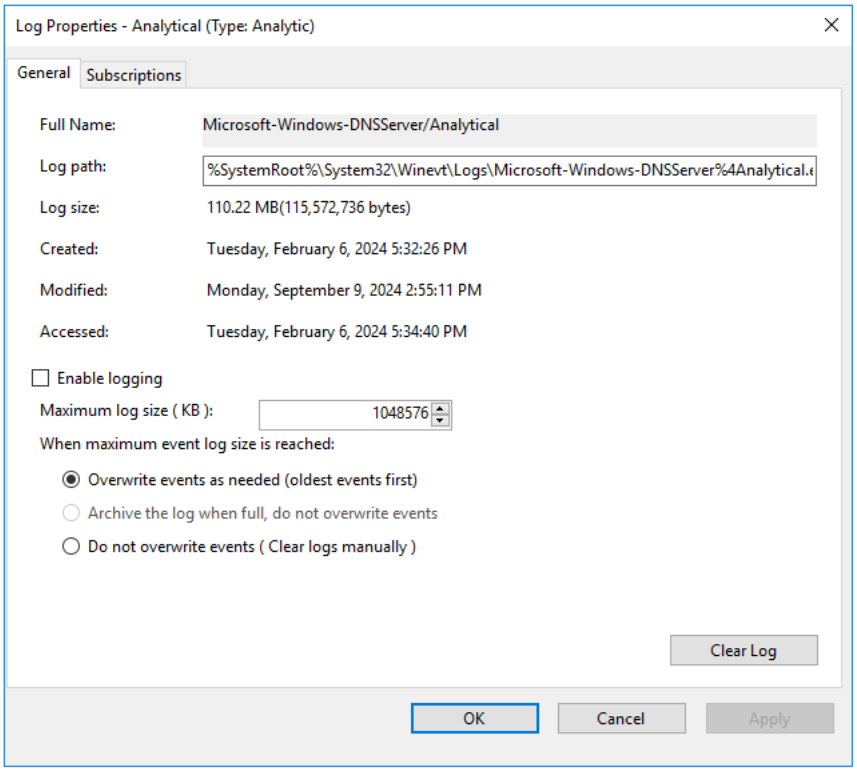
4. Configure the analytic log:

a. Open the context menu of the **Analytical** log and select **Properties**.



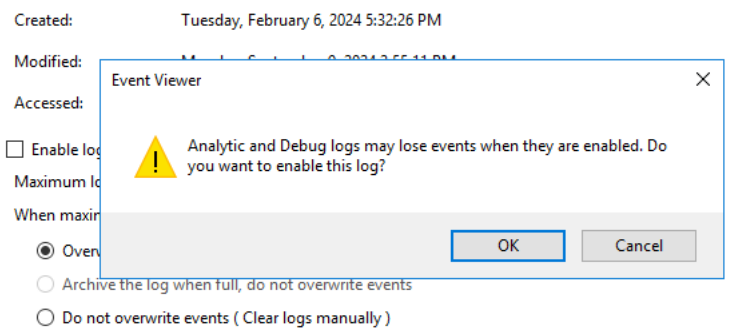
Go to the properties of the Analytical log

a. This opens a window; in that window, make sure that in the **Max Log Size (KB)** field, the value is `1048576`.



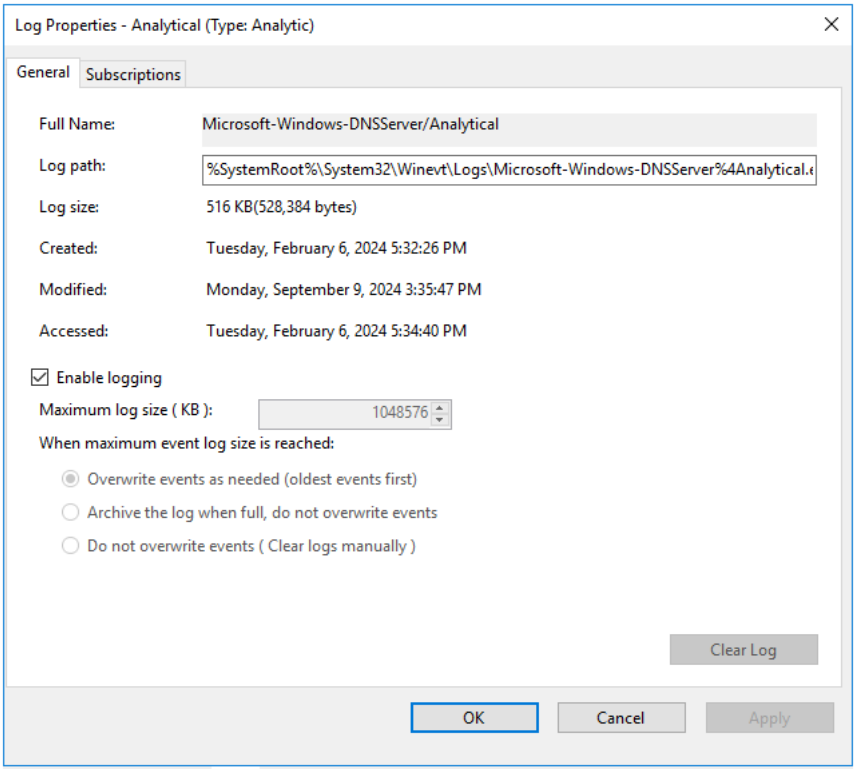
Analytical log properties

a. Select the **Enable logging** check box and in the confirmation window, click **OK**.



Confirmation window

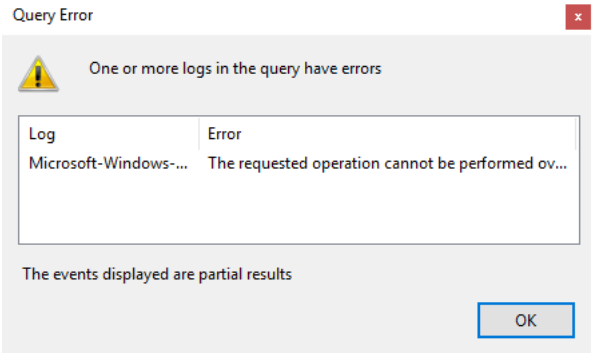
The analytic log must be configured as follows:



Reviewing the settings of the Analytical log for correctness

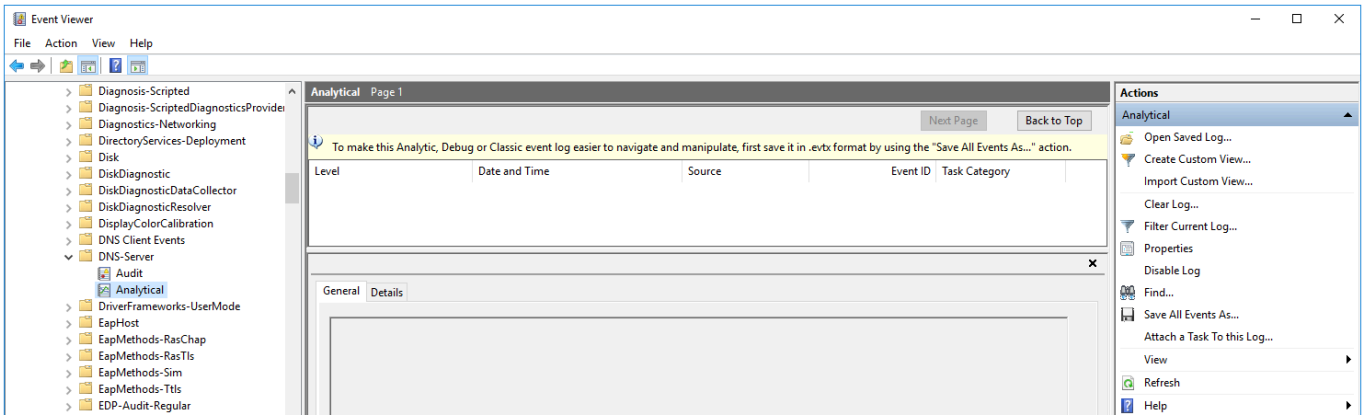
a. Click **Apply**, then click **OK**.

An error window is displayed.



Error window

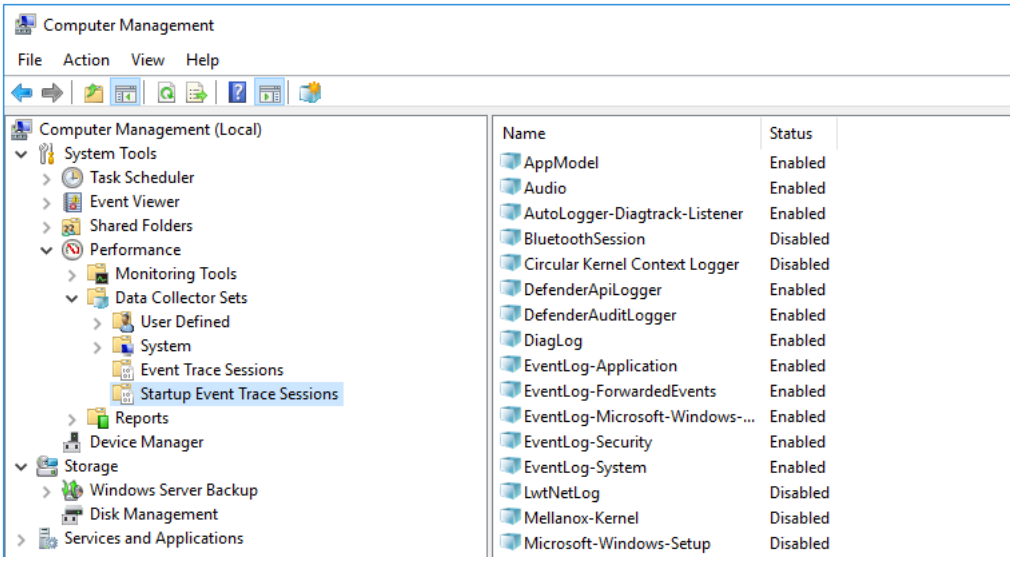
When analytic log rotation is enabled, events are not displayed. To view events, in the **Actions** pane, click **Stop logging**.



Stopping logging

5. Start Computer management as administrator.

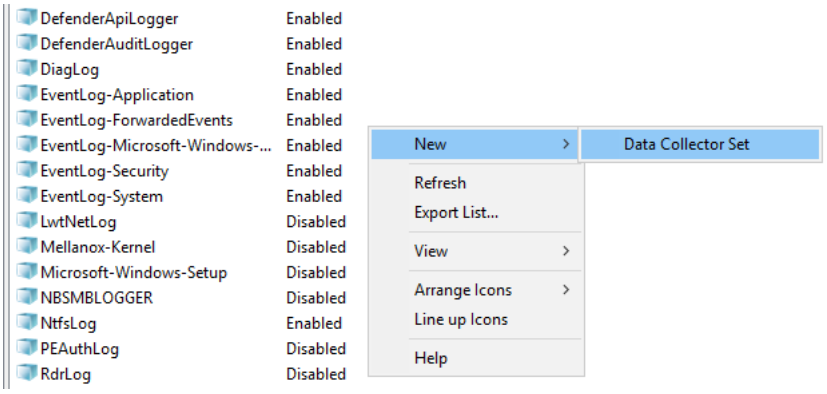
6. This opens a window; in that window, go to the **System Tools** → **Performance** → **Startup Event Trace Sessions** folder.



Managing the computer

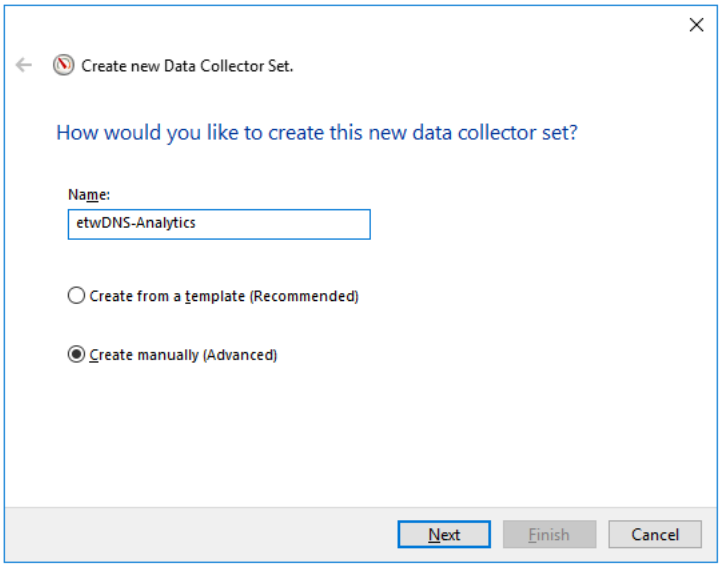
7. Create a provider:

a. Open the context menu of the **Startup Event Trace Sessions** folder and select **Create → Data Collector Set**.



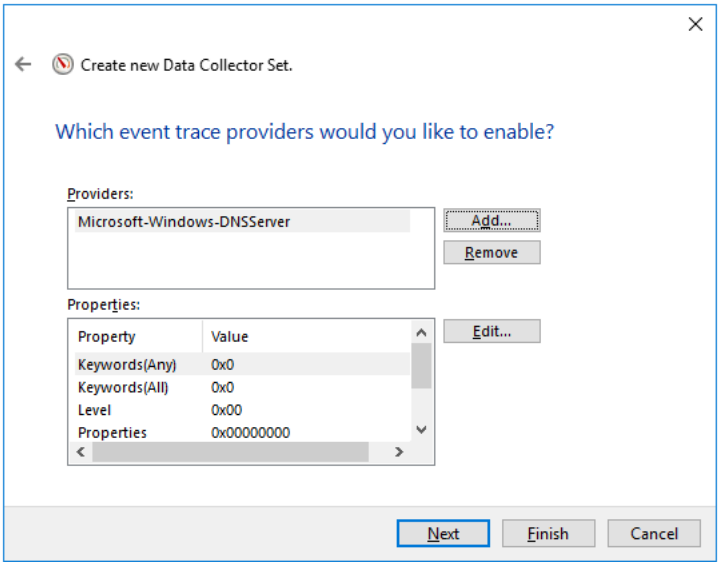
Creating a data collector group

a. This opens a window; in that window, enter the name of the provider and click **Next**.



Entering the provider name

a. Click **Add...** and in the displayed window, select the **Microsoft-Windows-DNSServer** provider.

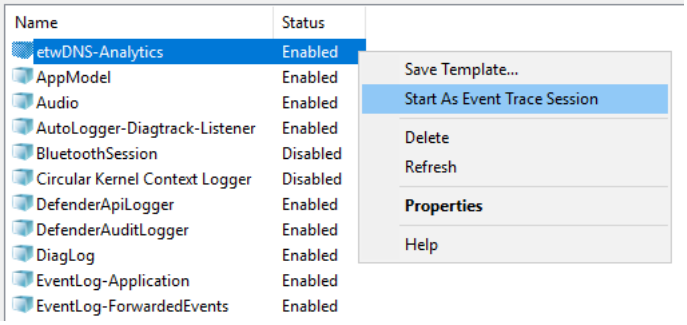


Selecting a provider

The LogAnalyzer agent with the ETW connector works only with System.Provider.Guid: {EB79061A-A566-4698-9119-3ED2807060E7} - Microsoft-Windows-DNSServer.

a. Click **Next** twice, then click **Finish**.

8. Open the context menu of the created provider and select **Start As Event Trace Session**.



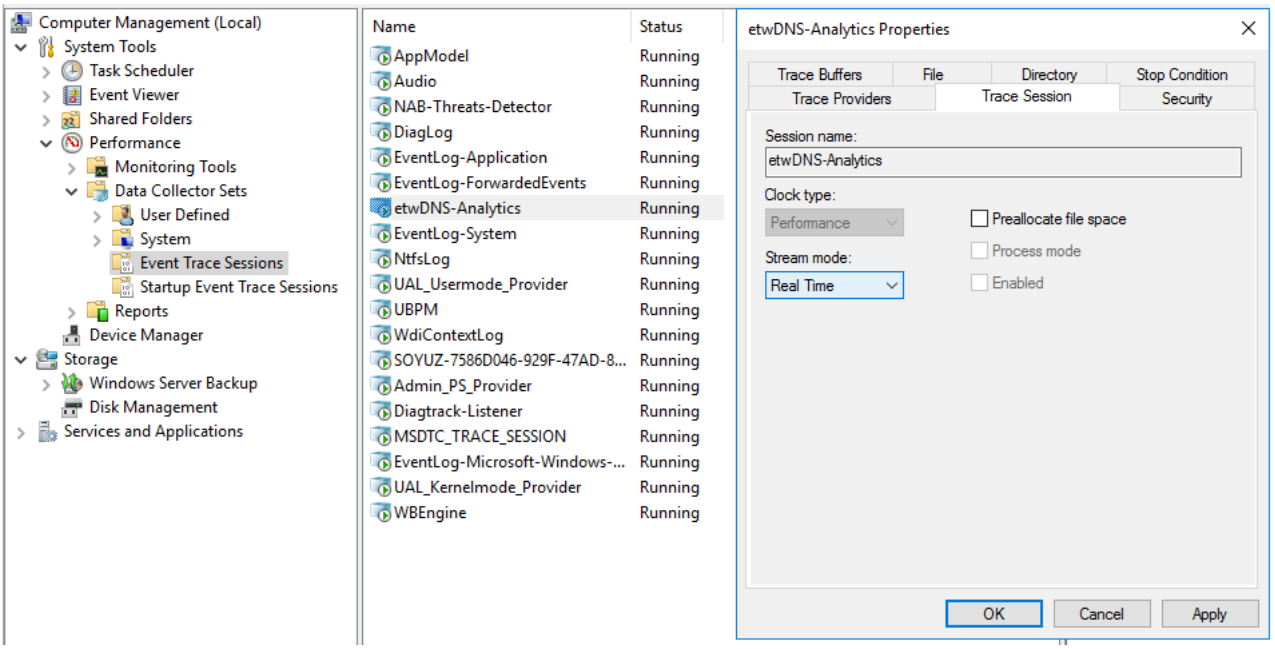
Starting the provider

9. Go to the **Event Trace Sessions** folder.

   Event trace sessions are displayed.

10. Open the context menu of the created event trace session and select **Properties**.

11. This opens a window; in that window, select the **Trace Sessions** tab and in the **Stream Mode** drop-down list, select **Real Time**.



Configuring stream mode

12. Click **Apply**, then click **OK**.

    DNS server event reception using the ETW connector is configured.

Configuring receipt of PostgreSQL events

LogAnalyzer lets you monitor and audit PostgreSQL events on Linux devices using rsyslog.

Events are audited using the pgAudit plugin. The plugin supports PostgreSQL 9.5 and later. For details about the pgAudit plugin, see https://github.com/pgaudit/pgaudit.

Configuring event receiving consists of the following steps:

1. Installing the pdAudit plugin.

2. Creating a LogAnalyzer collector for PostgreSQL events.

   To receive PostgreSQL events using rsyslog, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] PostgreSQL pgAudit syslog** normalizer.

3. Installing a collector in the LogAnalyzer network infrastructure.

4. Configuring the event source server.

5. Verifying receipt of PostgreSQL events in the LogAnalyzer collector

   You can verify that the PostgreSQL event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Installing the pgAudit plugin

To install the pgAudit plugin:

1. On the OS command line, run the following commands as a user with administrator rights:

   ```
   sudo apt update
   ```

   ```
   sudo apt -y install postgresql-<PostgreSQL version>-pgaudit
   ```

   You must select the plugin version to match the PostgresSQL version. For information about PostgreSQL versions and the matching plugin versions, see https://github.com/pgaudit/pgaudit#postgresql-version-compatibility.

   Example:
   ```
   sudo apt -y install postgresql-12-pgaudit
   ```

2. Find the postgres.conf configuration file. To do so, run the following command on the PostgreSQL command line:

   ```
   show data_directory
   ```

   The response will indicate the location of the configuration file.

3. Create a backup copy of the postgres.conf configuration file.

4. Open the postgres.conf file and copy or replace the values in it with the values listed below.

```
## pgAudit settings
shared_preload_libraries = 'pgaudit'
## database logging settings
log_destination = 'syslog'
## syslog facility
syslog_facility = 'LOCAL0'
## event ident
syslog_ident = 'Postgres'
## sequence numbers in syslog
syslog_sequence_numbers = on
## split messages in syslog
syslog_split_messages = off
## message encoding
lc_messages = 'en_US.UTF-8'
## min message level for logging
client_min_messages = log
## min error message level for logging
log_min_error_statement = info
## log checkpoints (buffers, restarts)
log_checkpoints = off
## log query duration
log_duration = off
## error description level
log_error_verbosity = default
## user connections logging
log_connections = on
## user disconnections logging
log_disconnections = on
## log prefix format
log_line_prefix = '%m|%a|%d|%p|%r|%i|%u| %e '
## log_statement
log_statement = 'none'
## hostname logging status. dns bane resolving affect
#performance!
log_hostname = off
## logging collector buffer status
#logging_collector = off
## pg audit settings
pgaudit.log_parameter = on
```

```
pgaudit.log='ROLE, DDL, MISC, FUNCTION'
```

5. Restart the PostgreSQL service using the command:

   ```
   sudo systemctl restart postgresql
   ```

6. To load the pgAudit plugin to PostgreSQL, run the following command on the PostgreSQL command line:

   ```
   CREATE EXTENSION pgaudit
   ```

The pgAudit plugin is installed.

Configuring a Syslog server to send events

The rsyslog service is used to transmit events from the server to KUMA.

To configure the sending of events from the server where PostgreSQL is installed to the collector:

1. To verify that the rsyslog service is installed on the event source server, run the following command as administrator:

   ```
   sudo systemctl status rsyslog.service
   ```

   If the rsyslog service is not installed on the server, install it by executing the following commands:

   ```
   yum install rsyslog
   sudo systemctl enable rsyslog.service
   sudo systemctl start rsyslog.service
   ```

2. In the /etc/rsyslog.d/ directory, create a pgsql-to-siem.conf file with the following content:

   ```
   If $programname contains 'Postgres' then @<IP address of the collector>:<port of the collector>
   For example:
   If $programname contains 'Postgres' then @192.168.1.5:1514
   ```

   If you want to send events via TCP, the contents of the file must be as follows:

   ```
   If $programname contains 'Postgres' then @@<IP address of the collector>:<port of the collector>
   ```

   Save changes to the pgsql-to-siem.conf configuration file.

3. Add the following lines to the /etc/rsyslog.conf configuration file:

   ```
   $IncludeConfig /etc/pgsql-to-siem.conf
   $RepeatedMsgReduction off
   ```

   Save changes to the /etc/rsyslog.conf configuration file.

4. Restart the rsyslog service by executing the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Configuring receipt of IVK Kolchuga-K events

You can configure the receipt of events from the IVK Kolchuga-K system to the LogAnalyzer SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring the sending of IVK Kolchuga-K events to LogAnalyzer.
2. Creating a LogAnalyzer collector for receiving events from the IVK Kolchuga-K system.

   To receive IVK Kolchuga-K events using Syslog, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Kolchuga-K syslog** normalizer.
3. Installing a LogAnalyzer collector for receiving IVK Kolchuga-K events.
4. Verifying the receipt of IVK Kolchuga-K events in LogAnalyzer.

   You can verify that the IVK Kolchuga-K event source is configured correctly in the Searching for related events section of the LogAnalyzer web interface.

Configuring the sending of IVK Kolchuga-K events to LogAnalyzer.

To configure the export of events of the IVK Kolchuga-K firewall via syslog to the LogAnalyzer collector:

1. Connect to the firewall over SSH with administrator rights.
2. Create a backup copy of the /etc/services and /etc/syslog.conf files.
3. In the /etc/syslog.conf configuration file, specify the FQDN or IP address of the LogAnalyzer collector. For example:

   . @kuma.example.com

   or

   . @192.168.0.100

   Save changes to the configuration file /etc/syslog.conf.
4. In the /etc/services configuration file, specify the port and protocol used by the KUMA collector. For example:

   syslog 10514/udp

   Save changes to the /etc/services configuration file.
5. Restart the syslog server of the firewall:

   ```
   service syslogd restart
   ```

Configuring receipt of CryptoPro NGate events

You can configure the receipt of CryptoPro NGate events in the LogAnalyzer SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring export of CryptoPro NGate events to LogAnalyzer.
2. Creating a LogAnalyzer collector for receiving CryptoPro NGate events.

   To receive CryptoPro NGate events using Syslog, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] NGate syslog** normalizer.
3. Installing a LogAnalyzer collector for receiving CryptoPro NGate events.
4. Verifying the receipt of CryptoPro NGate events in the LogAnalyzer collector.

   You can verify that the CryptoPro NGate event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Configuring the export of CryptoPro NGate events to LogAnalyzer.

To configure the export of events from CryptoPro NGate to LogAnalyzer:

1. Connect to the web interface of the NGate management system.
2. Connect remote syslog servers to the management system. To do so:

   a. Open the page with the list of syslog servers: **External Services** → **Syslog Server** → **Add Syslog Server**.

   b. Enter the settings of the syslog server and click ✔.

3. Assign syslog servers to the configuration for recording logs of the cluster. To do so:

   a. In the **Clusters** → **Summary** section, select the cluster that you want to configure**.**

   b. On the **Configurations** tab, click the **Configuration** control for the relevant cluster to go to the configuration settings page.

   c. **Syslog Servers**
   of the configuration you are configuring, click

   Assign
   .

   d. Select the check boxes for syslog servers that you want to assign and click

   the icon.

   You can assign an unlimited number of servers.

   To add new syslog servers, click .

   e. Publish the configuration to activate the new settings.

4. Assign syslog servers to the management system for recording Administrator activity logs. To do so:

   a. Select the **Management Center Settings** menu item and on the page that is displayed, under **Syslog servers**, click **Assign**.

   b. In the **Assign Syslog Servers to Management Center** window, select the check box for those syslog servers that you want to assign, then click .

   You can assign an unlimited number of servers.

As a result, events of CryptoPro NGate are sent to LogAnalyzer.

Configuring receipt of Ideco UTM events

You can configure the receipt of Ideco UTM application events in LogAnalyzer via the Syslog protocol.

Configuring event receiving consists of the following steps:

1. Configuring the export of Ideco UTM events to LogAnalyzer.

2. Creating a LogAnalyzer collector for receiving Ideco UTM events.

   To receive Ideco UTM events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Ideco UTM syslog** normalizer.

3. Installing a LogAnalyzer collector for receiving Ideco UTM events.

4. Verifying the receipt of Ideco UTM events in LogAnalyzer.

   You can verify that the Ideco UTM event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Configuring the export of Ideco UTM events to LogAnalyzer.

To configure the sending of events from Ideco UTM to LogAnalyzer:

1. Connect to the Ideco UTM web interface under a user account with administrator rights.
2. In the **System message forwarding** menu, move the **Syslog** toggle switch to the **enabled** position.
3. For the **IP address** setting, specify the IP address of the LogAnalyzer collector.
4. For the Port setting, enter the port that the LogAnalyzer collector is listening on.
5. Click **Save** to apply the changes.

The export of Ideco UTM events to LogAnalyzer is configured.

Configuring receipt of KWTS events

You can configure the receipt of events from the Kaspersky Web Traffic Security (KWTS) web traffic analysis and filtering system in LogAnalyzer.

Configuring event receiving consists of the following steps:

1. Configuring the export of KWTS events to LogAnalyzer
2. Creating a LogAnalyzer collector for receiving KWTS events.

   To receive KWTS events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] KWTS** normalizer.
3. Installing a LogAnalyzer collector for receiving KWTS events.
4. Verifying the receipt of KWTS events in the LogAnalyzer collector.

   You can verify that KWTS event export is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Configuring the export of KWTS events to LogAnalyzer.

To configure the export of KWTS events to LogAnalyzer:

1. Connect to the KWTS server over SSH as root.
2. Before making changes, create backup copies of the following files:
   - /opt/kaspersky/kwts/share/templates/core_settings/event_logger.json.template
   - /etc/rsyslog.conf

3. Make sure that the settings in the /opt/kaspersky/kwts/share/templates/core_settings/event_logger.json.template configuration file have the following values, and make changes if necessary:

```
"siemSettings":
{
"enabled": true,
"facility": "Local5",
"logLevel": "Info",
"formatting":
{
```

4. Save your changes.

5. To send events via UDP, make the following changes to the /etc/rsyslog.conf configuration file:

```
$WorkDirectory /var/lib/rsyslog
$ActionQueueFileName ForwardToSIEM
$ActionQueueMaxDiskSpace 1g
$ActionQueueSaveOnShutdown on
$ActionQueueType LinkedList
$ActionResumeRetryCount -1
local5.* @<< IP address of the KUMA collector >:< port of the collector >>
```

If you want to send events over TCP, the last line should be as follows:

```
local5.* @@<< IP address of the KUMA collector >:< port of the collector >>
```

6. Save your changes.

7. Restart the rsyslog service with the following command:

```
sudo systemctl restart rsyslog.service
```

8. Go to the KWTS web interface, to the **Settings** → **Syslog** tab and enable the **Log information about traffic profile** option.

9. Click **Save**.

Configuring receipt of KLMS events

You can configure the receipt of events from the Kaspersky Linux Mail Server (KLMS) mail traffic analysis and filtering system to the LogAnalyzer SIEM system.

Configuring event receiving consists of the following steps:

1. Depending on the version of KLMS you are using, select one of the following options:
   - Configuring the export of KLMS 8 events to LogAnalyzer
   - Configuring the export of KLMS 10 events to LogAnalyzer
2. Creating a LogAnalyzer collector for receiving KLMS events

   To receive KLMS events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] KLMS syslog CEF normalizer**.
3. Installing a LogAnalyzer collector for receiving KLMS events.
4. Verifying the receipt of KLMS events in the LogAnalyzer collector

   You can verify that the KLMS event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Configuring the export of KLMS events to LogAnalyzer

To configure the export of KLMS events to LogAnalyzer:

1. Connect to the KLMS server over SSH and go to the **Technical Support Mode** menu.
2. Use the klms-control utility to download the settings to the settings.xml file:

   ```
   sudo /opt/kaspersky/klms/bin/klms-control --get-settings EventLogger -n -f
   /tmp/settings.xml
   ```
3. Make sure that the settings in the /tmp/settings.xml file have the following values; make changes if necessary:

   ```
   <siemSettings>
   <enabled>1</enabled>
   <facility>Local1</facility>
   ...
   </siemSettings>
   ```
4. Apply settings with the following command:

   ```
   sudo /opt/kaspersky/klms/bin/klms-control --set-settings EventLogger -n -f
   /tmp/settings.xml
   ```

5. To send events via UDP, make the following changes to the /etc/rsyslog.conf configuration file:

   ```
   $WorkDirectory /var/lib/rsyslog
   $ActionQueueFileName ForwardToSIEM
   $ActionQueueMaxDiskSpace 1g
   $ActionQueueSaveOnShutdown on
   $ActionQueueType LinkedList
   $ActionResumeRetryCount -1
   local1.* @<< IP address of the KUMA collector >:< port of the collector >>
   ```

   If you want to send events over TCP, the last line should be as follows:

   ```
   local1.* @@<< IP address of the KUMA collector >:< port of the collector >>
   ```

6. Save your changes.

7. Restart the rsyslog service with the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Configuring receipt of KSMG events

You can configure the receipt of events from the Kaspersky Secure Mail Gateway (KSMG) 1.1 mail traffic analysis and filtering system in the LogAnalyzer SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring the export of KSMG events to LogAnalyzer

2. Creating a LogAnalyzer collector for receiving KSMG events

   To receive KSMG events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] KSMG** normalizer.

3. Installing a LogAnalyzer collector for receiving KSMG events.

4. Verifying the receipt of KSMG events in the LogAnalyzer collector

   You can verify that the KSMG event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Configuring the export of KSMG events to LogAnalyzer

To configure the export of KSMG events to LogAnalyzer:

1. Connect to the KSMG server via SSH using an account with administrator rights.

2. Use the ksmg-control utility to download the settings to the settings.xml file:

   ```
   sudo /opt/kaspersky/ksmg/bin/ksmg-control --get-settings EventLogger -n -f
   /tmp/settings.xml
   ```

3. Make sure that the settings in the /tmp/settings.xml file have the following values; make changes if necessary:

   ```
   <siemSettings>
   <enabled>1</enabled>
   <facility>Local1</facility>
   ```

4. Apply settings with the following command:

   ```
   sudo /opt/kaspersky/ksmg/bin/ksmg-control --set-settings EventLogger -n -f
   /tmp/settings.xml
   ```

5. To send events via UDP, make the following changes to the /etc/rsyslog.conf configuration file:

   ```
   $WorkDirectory /var/lib/rsyslog
   $ActionQueueFileName ForwardToSIEM
   $ActionQueueMaxDiskSpace 1g
   $ActionQueueSaveOnShutdown on
   $ActionQueueType LinkedList
   $ActionResumeRetryCount -1
   local1.* @<< IP address of the KUMA collector >:< port of the collector >>
   ```

   If you want to send events over TCP, the last line should be as follows:

   ```
   local1.* @@<< IP address of the KUMA collector >:< port of the collector >>
   ```

6. Save your changes.

7. Restart the rsyslog service with the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Configuring the receipt of KICS for Networks events

You can configure the receipt of events from Kaspersky Industrial CyberSecurity for Networks (KICS for Networks) 4.2 in LogAnalyzer.

Configuring event receiving consists of the following steps:

1. Creating a KICS for Networks connector for sending events to LogAnalyzer

2. Configuring the export of KICS for Networks events to LogAnalyzer

3. Creating and installing a LogAnalyzer collector to receive KICS for Networks events.

4. Verifying the receipt of KICS for Networks events in the LogAnalyzer collector.

   You can verify that KICS for Networks event export is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Creating a KICS for Networks connector for sending events to LogAnalyzer

To create a connector for sending events in the web interface of KICS for Networks:

1. Log in to the KICS for Networks web interface using an administrator account.
2. Go to the **Settings** → **Connectors** section.
3. Click the **Add connector** button.
4. Specify the following settings:
   a. In the **Connector type** drop-down list, select **SIEM**.
   b. In the **Connector name** field, specify a name for the connector.
   c. In the **Server address** field, enter the IP address of the KICS for Networks Server.
   d. In the **Connector deployment node** drop-down list, select the node on which you are installing the connector.
      You can specify any name.
   e. In the **User name** field, specify the user name for LogAnalyzer to use for connecting to the application through the connector. You must specify the name of one of the KICS for Networks users.
   f. In the **SIEM server address** field, enter the IP address of the LogAnalyzer collector server.
   g. In the **Port number** field, enter the port number of the LogAnalyzer collector.
   h. In the **Transport protocol** drop-down list, select **TCP** or **UDP**.
   i. Select the **Allow sending audit entries** check box.
   j. Select the **Allow sending application entries** check box.
5. Click the **Save** button.
   The connector is created. It is displayed in the table of KICS for Networks connectors with the **Running** status.

The KICS for Networks connector for sending events to LogAnalyzer is ready for use.

Configuring the export of KICS for Networks events to LogAnalyzer

To configure the sending of security events from KICS for Networks to LogAnalyzer:

1. Log in to the KICS for Networks web interface using an administrator account.
2. Go to the **Settings** → **Event types** section.
3. Select the check boxes for the types of events that you want to send to LogAnalyzer.

4. Click **Select connectors**.

5. This opens a window; in that window, select the connector that you created for sending events to LogAnalyzer.

6. Click **OK**.

   Events of selected types will be sent to LogAnalyzer. In the **Event types** table, such events are marked with a check box in the column with the connector name.

Creating a LogAnalyzer collector to receive KICS for Networks events

After managing the event export settings, you must create a collector for KICS for Networks events in the LogAnalyzer web interface.

For details on creating a LogAnalyzer collector, refer to Creating a collector.

When creating a collector in the LogAnalyzer web interface, you must satisfy the following requirements:

1. At the **Transport** step, select the transport protocol type matching the type you selected when you created the connector in KICS for Networks at step 4i (TCP or UDP) and the port number matching the port number you specified at step 4h.

2. At the **Event parsing** step, select the **[OOTB] KICS4Net v3.x** normalizer.

3. At the **Routing** step, make sure that the following destinations are added to the collector resource set:

   - **storage**—used to transmit data to the storage.

   If destinations have not been added to the collector, you must create them.

4. At the last step of the wizard, a command is displayed in the lower part of the window, which you can use to install the service on the server that you want to receive events. Copy this command and use it when installing the second part of the collector.

Configuring receipt of PT NAD events

You can configure the receipt of PT NAD events in the LogAnalyzer SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring the export of PT NAD events to LogAnalyzer.

2. Creating a LogAnalyzer collector for receiving PT NAD events.

   To receive PT NAD events using Syslog, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] PT NAD json** normalizer.

3. Installing a LogAnalyzer collector for receiving PT NAD events.

4. Verifying the receipt of PT NAD events in the LogAnalyzer collector.

   You can verify that the PT NAD event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Configuring the export of PT NAD events to LogAnalyzer.

Configuring the export of events from PT NAD 11 to KUMA over Syslog involves the following steps:

1. Configuring the ptdpi-worker@notifier module.
2. Configuring the sending of syslog messages with information about activities, attacks and indicators of compromise.

Configuring the ptdpi-worker@notifier module.

To enable the sending of information about detected information security threats, you must configure the ptdpi-worker@notifier module.

In a multi-server configuration, these instructions must be followed on the primary server.

To configure the ptdpi-worker@notifier module:

1. Open the /opt/ptsecurity/etc/ptdpi.settings.yaml file:

   ```
   sudo nano /opt/ptsecurity/etc/ptdpi.settings.yaml
   ```

2. In the **General settings** group of settings, uncomment the 'workers' setting and add 'notifier' to its list of values.

   For example:

   ```
   workers: ad alert dns es hosts notifier
   ```

3. To the end of the file, append a line of the form: notifier.yaml.nad_web_url: <URL of the PT NAD web interface>

   For example:

   ```
   notifier.yaml.nad_web_url: https://ptnad.example.com
   ```

   The ptdpi-worker@notifier module uses the specified URL to generate links to session and activity cards when sending messages.

4. Restart the sensor:

   ```
   sudo ptdpictl restart-all
   ```

The ptdpi-worker@notifier module is configured.

Configuring the sending of syslog messages with information about activities, attacks and indicators of compromise

The settings listed in the following instructions may not be present in the configuration file. If a setting is missing, you must add it to the file.

In a multi-server PT NAD configuration, edit the settings on the primary server.

To configure the sending of syslog messages with information about activities, attacks and indicators of compromise:

1. Open the /opt/ptsecurity/etc/ptdpi.settings.yaml file:

   ```
   sudo nano /opt/ptsecurity/etc/ptdpi.settings.yaml
   ```

2. By default, PT NAD sends activity information in Russian. To receive information in English, change the value of the notifier.yaml.syslog_notifier.locale setting to `en`.

   For example:

   ```
   notifier.yaml.syslog_notifier.locale: en
   ```

3. In the notifier.yaml.syslog_notifier.addresses setting, add a section with settings for sending events to KUMA.

   The <Connection name> setting can only contain Latin letters, numerals, and the underscore character.

   For the 'address' setting, specify the IP address of the KUMA collector.

   Other settings can be omitted, in which case the default values are used.

   ```
   notifier.yaml.syslog_notifier.addresses:
   <Connection name>:
   address: <For sending to a remote server, specify protocol: UDP (default) or TCP, address and port; for local connection, specify Unix domain socket>
   doc_types: [<Comma-separated message types ('alert' for information about attacks, 'detection' for activities, and 'reputation' for information about indicators of compromise). By default, all types of messages are sent>]
   facility: <Numeric value of the subject category>
   ident: <software tag>
   <Connection name>:
   ...
   ```

   The following is an example configuration of sending syslog messages with information about activities, attacks, and indicators of compromise to two remote servers via TCP and UDP without writing to the local log:

   ```
   notifier.yaml.syslog_notifier.addresses:
   remote1:
   address: tcp://198.51.100.1:1514
   remote2:
   address: udp://198.51.100.2:2514
   ```

4. Save your changes in the /opt/ptsecurity/etc/ptdpi.settings.yaml.

5. Restart the ptdpi-worker@notifier module:

   ```
   sudo ptdpictl restart-worker notifier
   ```

The sending of events to KUMA via Syslog is configured.

Configuring receipt of events using the MariaDB Audit Plugin

LogAnalyzer allows auditing events using the MariaDB Audit Plugin. The plugin supports MySQL 5.7 and MariaDB. The audit plugin does not support MySQL 8. Detailed information about the plugin is available on the official MariaDB website.

We recommend using MariaDB Audit Plugin version 1.2 or later.

Configuring event receiving consists of the following steps:

1. Configuring the MariaDB Audit Plugin to send MySQL events and configuring the Syslog server to send events.
2. Configuring the MariaDB Audit Plugin to send MariaDB events and configuring the Syslog server to send events.
3. Creating a LogAnalyzer Collector for MySQL 5.7 and MariaDB Events.

   To receive MySQL 5.7 and MariaDB events using the MariaDB Audit Plugin, in the LogAnalyzer Collector Installation Wizard, at the **Event parsing** step, in the **Normalizer** field, select **[OOTB] MariaDB Audit Plugin syslog**.
4. Installing a collector in the LogAnalyzer network infrastructure.
5. Verifying the receipt of MySQL and MariaDB events by the LogAnalyzer collector.

   To verify that the MySQL and MariaDB event source server is configured correctly, you can search for related events.

Configuring the MariaDB Audit Plugin to send MySQL events

The MariaDB Audit Plugin is supported for MySQL 5.7 versions up to 5.7.30 and is bundled with MariaDB.

To configure MySQL 5.7 event reporting using the MariaDB Audit Plugin:

1. Download the MariaDB distribution kit and extract it.

   You can download the MariaDB distribution kit from the official MariaDB website. The operating system of the MariaDB distribution must be the same as the operating system on which MySQL 5.7 is running.
2. Connect to MySQL 5.7 using an account with administrator rights by running the following command:

   ```
   mysql -u <username> -p
   ```
3. To get the directory where the MySQL 5.7 plugins are located, on the MySQL 5.7 command line, run the following command:

   ```
   SHOW GLOBAL VARIABLES LIKE 'plugin_dir'
   ```
4. In the directory obtained at step 3, copy the MariaDB Audit Plugin from `<directory to which the distribution kit was extracted>/mariadb-server-<version>/lib/plugins/server_audit.so`.
5. On the operating system command line, run the following command:

   ```
   chmod 755 <directory to which the distribution kit was extracted>server_audit.so
   ```

   For example:

   ```
   chmod 755 /usr/lib64/mysql/plugin/server_audit.so
   ```
6. On the MySQL 5.7 command line, run the following command:

   ```
   install plugin server_audit soname 'server_audit.so'
   ```
7. Create a backup copy of the /etc/mysql/mysql.conf.d/mysqld.cnf configuration file.

8. In the configuration file /etc/mysql/mysql.conf.d/mysqld.cnf, in the `[mysqld]` section, add the following lines:

   ```
   server_audit_logging=1
   server_audit_events=connect,table,query_ddl,query_dml,query_dcl
   server_audit_output_type=SYSLOG
   server_audit_syslog_facility=LOG_SYSLOG
   ```

   If you want to disable event export for certain audit event groups, remove some of the values from the `server_audit_events` setting. Descriptions of settings are available on the MariaDB Audit Plugin vendor's website.

9. Save changes to the configuration file.

10. Restart the MariaDB service by running one of the following commands:

    - `systemctl restart mysqld` for a system with systemd initialization.
    - `service mysqld restart` for a system with init initialization.

MariaDB Audit Plugin for MySQL 5.7 is configured. If necessary, you can run the following commands on the MySQL 5.7 command line:

- `show plugins` to review the list of current plug-ins.
- `SHOW GLOBAL VARIABLES LIKE 'server_audit%'` to review the current audit settings.

Configuring the MariaDB Audit Plugin to send MariaDB Events

The MariaDB Audit Plugin is included in the MariaDB distribution kit starting with versions 5.5.37 and 10.0.10.

To configure MariaDB event export using the MariaDB Audit Plugin:

1. Connect to MariaDB using an account with administrator rights by running the following command:

   ```
   mysql -u <username> -p
   ```

2. To check if the plugin is present in the directory where operating system plugins are located, run the following command on the MariaDB command line:

   ```
   SHOW GLOBAL VARIABLES LIKE 'plugin_dir'
   ```

3. On the operating system command line, run the following command:

   ```
   ll <directory obtained by the previous command> | grep server_audit.so
   ```

   If the command output is empty and the plugin is not present in the directory, you can either copy the MariaDB Audit Plugin to that directory or use a newer version of MariaDB.

4. On the MariaDB command line, run the following command:

   ```
   install plugin server_audit soname 'server_audit.so'
   ```

5. Create a backup copy of the /etc/mysql/my.cnf configuration file.

6. In the /etc/mysql/my.cnf configuration file, in the `[mysqld]` section, add the following lines:

   ```
   server_audit_logging=1
   server_audit_events=connect,table,query_ddl,query_dml,query_dcl
   server_audit_output_type=SYSLOG
   server_audit_syslog_facility=LOG_SYSLOG
   ```

   If you want to disable event export for certain audit event groups, remove some of the values from the `server_audit_events` setting. Descriptions of settings are available on the MariaDB Audit Plugin vendor's website.

7. Save changes to the configuration file.

8. Restart the MariaDB service by running one of the following commands:

   - `systemctl restart mariadb` for a system with systemd initialization.
   - `service mariadb restart` for a system with init initialization.

MariaDB Audit Plugin for MariaDB is configured. If necessary, you can run the following commands on the MariaDB command line:

- `show plugins` to review the list of current plug-ins.
- `SHOW GLOBAL VARIABLES LIKE 'server_audit%'` to review the current audit settings.

Configuring a Syslog server to send events

The rsyslog service is used to transmit events from the server to the collector.

To configure the sending of events from the server where MySQL or MariaDB is installed to the collector:

1. Before making any changes, create a backup copy of the /etc/rsyslog.conf configuration file.

2. To send events via UDP, add the following line to the /etc/rsyslog.conf configuration file:

   ```
   *.* @<IP address of the LogAnalyzer collector>:<port of the LogAnalyzer collector>
   ```

   For example:

   ```
   *.* @192.168.1.5:1514
   ```

   If you want to send events over TCP, the line should be as follows:

   ```
   *.* @@192.168.1.5:2514
   ```

   Save changes to the /etc/rsyslog.conf configuration file.

3. Restart the rsyslog service by executing the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Configuring receipt of Apache Cassandra events

LogAnalyzer allows receiving information about Apache Cassandra events.

Configuring event receiving consists of the following steps:

1. Configuring Apache Cassandra event logging in LogAnalyzer.
2. Creating a LogAnalyzer collector for Apache Cassandra events.

   To receive Apache Cassandra events, in the LogAnalyzer Collector Installation Wizard, at the **Transport** step, select a **file** type connector; at the **Event parsing** step, in the **Normalizer** field, select **[OOTB] Apache Cassandra file**.
3. Installing a collector in the LogAnalyzer network infrastructure.
4. Verifying the receipt of Apache Cassandra events in the LogAnalyzer collector.

   To verify that the Apache Cassandra event source server is configured correctly, you can search for related events.

Configuring Apache Cassandra event logging in LogAnalyzer.

To configure Apache Cassandra event logging in LogAnalyzer:

1. Make sure that the server where Apache Cassandra is installed has 5 GB of free disk space.
2. Connect to the Apache Cassandra server using an account with administrator rights.
3. Before making changes, create backup copies of the following configuration files:
   - /etc/cassandra/cassandra.yaml
   - /etc/cassandra/logback.xml
4. Make sure that the settings in the /etc/cassandra/cassandra.yaml configuration file have the following values; make changes if necessary:

   a. in the `audit_logging_options` section, set the `enabled` setting to `true`.

   b. in the `logger` section, set the `class_name` to `FileAuditLogger`.

5. Add the following lines to the /etc/cassandra/logback.xml configuration file:

```
<!-- Audit Logging (FileAuditLogger) rolling file appender to audit.log -->
<appender name="AUDIT" class="ch.qos.logback.core.rolling.RollingFileAppender">
<file>${cassandra.logdir}/audit/audit.log</file>
<rollingPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">
<!-- rollover daily -->
<fileNamePattern>${cassandra.logdir}/audit/audit.log.%d{yyyy-MM-
dd}.%i.zip</fileNamePattern>
<!-- each file should be at most 50MB, keep 30 days worth of history, but at most 5GB
-->
<maxFileSize>50MB</maxFileSize>
<maxHistory>30</maxHistory>
<totalSizeCap>5GB</totalSizeCap>
</rollingPolicy>
<encoder>
<pattern>%-5level [%thread] %date{ISO8601} %F:%L - %replace(%msg){'\n', '
'}%n</pattern>
</encoder>
</appender>
<!-- Audit Logging additivity to redirect audt logging events to audit/audit.log -->
<logger name="org.apache.cassandra.audit" additivity="false" level="INFO">
<appender-ref ref="AUDIT"/>
</logger>
```

6. Save changes to the configuration file.

7. Restart the Apache Cassandra service using the following commands:

   a. `sudo systemctl stop cassandra.service`

   b. `sudo systemctl start cassandra.service`

8. After restarting, check the status of Apache Cassandra using the following command:

   ```
   sudo systemctl status cassandra.service
   ```

   Make sure that the command output contains the following sequence of characters:

   ```
   Active: active (running)
   ```

Apache Cassandra event export is configured. Events are located in the /var/log/cassandra/audit/ directory, in the audit.log file (${cassandra.logdir}/audit/audit.log).

Configuring receipt of FreeIPA events

You can configure the receipt of FreeIPA events in LogAnalyzer via the Syslog protocol.

Configuring event receiving consists of the following steps:

1. Configuring the export of FreeIPA events to LogAnalyzer.
2. Creating a LogAnalyzer collector for receiving FreeIPA events.

   To receive FreeIPA device events, in the LogAnalyzer Collector Installation Wizard, at the **Event parsing** step, in the **Normalizer** field, select **[OOTB] FreeIPA**.
3. Installing the LogAnalyzer collector in the network infrastructure.
4. Verifying the receipt of FreeIPA events by LogAnalyzer.

   To verify that the FreeIPA event source server is configured correctly, you can search for related events.

Configuring the export of FreeIPA events to LogAnalyzer.

To configure the export of FreeIPA events to LogAnalyzer via the Syslog protocol in JSON format:

1. Connect to the FreeIPA server via SSH using an account with administrator rights.
2. In the /etc/rsyslog.d/ directory, create a file named freeipa-to-siem.conf.

3. Add the following lines to the /etc/rsyslog.d/freeipa-to-siem.conf configuration file:

```
$ModLoad imfile
input(type="imfile"
File="/var/log/httpd/error_log"
Tag="tag_FreeIPA_log_httpd"
facility="local3")
input(type="imfile"
File="/var/log/dirsrv/slapd-*/audit"
Tag="tag_FreeIPA_log_audit"
StartMsg.regex="^time:"
facility="local3")
input(type="imfile"
File="/var/log/dirsrv/slapd-*/errors"
Tag="tag_FreeIPA_log_errors"
facility="local3")
input(type="imfile"
File="/var/log/dirsrv/slapd-*/access"
Tag="tag_FreeIPA_log_access"
facility="local3")
input(type="imfile"
File="/var/log/krb5kdc.log"
Tag="tag_FreeIPA_log_krb5kdc"
facility="local3")
template(name="ls_json" type="list" option.json="on") {
constant(value="{")
constant(value="\"@timestamp\":\"") property(name="timegenerated"
dateFormat="rfc3339")
constant(value="\",\"@version\":\"1")
constant(value="\",\"message\":\"") property(name="msg")
constant(value="\",\"host\":\"") property(name="fromhost")
constant(value="\",\"host_ip\":\"") property(name="fromhost-ip")
constant(value="\",\"logsource\":\"") property(name="fromhost")
constant(value="\",\"severity_label\":\"") property(name="syslogseverity-text")
constant(value="\",\"severity\":\"") property(name="syslogseverity")
constant(value="\",\"facility_label\":\"") property(name="syslogfacility-text")
constant(value="\",\"facility\":\"") property(name="syslogfacility")
constant(value="\",\"program\":\"") property(name="programname")
constant(value="\",\"pid\":\"") property(name="procid")
constant(value="\",\"syslogtag\":\"") property(name="syslogtag")
constant(value="\"}\n")
}
```

```
if $syslogtag contains 'tag_FreeIPA_log' then {
action(type="omfwd"
target="< IP address of KUMA collector >"
port="< port of KUMA collector >"
protocol="< udp or tcp >"
template="ls_json")
stop
}
```

4. Add the following lines to the /etc/rsyslog.conf configuration file:

   ```
   $IncludeConfig /etc/rsyslog.d/freeipa-to-siem.conf
   $RepeatedMsgReduction off
   ```

5. If necessary, disable the local recording of FreeIPA events by doing the following:

   a. Open the rsyslog configuration file /etc/rsyslog.conf or the default rules file /etc/rsyslog.d/50-default.conf for editing.

   b. In that file, find the line that is responsible for recording system events, and add the `local3.none` parameter to it because in the /etc/rsyslog.d/freeipa-to-siem.conf configuration file, the `facility` is set to `local3`. The edited line should look like this:

      ```
      *.info;mail.none;authpriv.none;cron.none;local3.none /var/log/messages
      ```

      If `facility` is set to `local3` for another application on the server, change the `facility` of FreeIPA in the /etc/rsyslog.d/freeipa-to-siem.conf file to a different value and edit the chosen rsyslog file accordingly.

   c. Save your changes.

6. Save changes to the /etc/rsyslog.d/freeipa-to-siem.conf configuration file.

7. Restart the rsyslog service by executing the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Configuring receipt of VipNet TIAS events

You can configure the receipt of ViPNet TIAS events in LogAnalyzer via the Syslog protocol.

Configuring event receiving consists of the following steps:

1. Configuring the export of ViPNet TIAS events to LogAnalyzer.

2. Creating a LogAnalyzer collector for receiving ViPNet TIAS events.

   To receive ViPNet TIAS events using Syslog, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Syslog-CEF** normalizer.

3. Installing a LogAnalyzer collector for receiving ViPNet TIAS events.

4. Verifying the receipt of ViPNet TIAS events in LogAnalyzer.

   You can verify that ViPNet TIAS event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Configuring the export of ViPNet TIAS events to LogAnalyzer.

To configure the export of ViPNet TIAS events to LogAnalyzer via the syslog protocol:

1. Connect to the ViPNet TIAS web interface under a user account with administrator rights.

2. Go to the **Management** – **Integrations** section.

3. On the **Integration** page, go to the Syslog tab.

4. In the toolbar of the list of receiving servers, click **New server**.

5. This opens the new server card; in that card:

   1. In the **Server address** field, enter the IP address or domain name of the LogAnalyzer collector.

      For example, 10.1.2.3 or syslog.siem.ru

   2. In the **Port** field, specify the inbound port of the LogAnalyzer collector. The default port number is 514.

   3. In the **Protocol** list, select the transport layer protocol that the LogAnalyzer collector is listening on. UDP is selected by default.

   4. In the **Organization** list, use the check boxes to select the organizations of the ViPNet TIAS infrastructure.

      Messages are sent only for incidents detected based on events received from sensors of selected organizations of the infrastructure.

   5. In the **Status** list, use check boxes to select incident statuses.

      Messages are sent only when selected statuses are assigned to incidents.

   6. In the **Severity level** list, use check boxes to select the severity levels of the incidents.

      Messages are sent only about incidents with the selected severity levels. By default, only the high severity level is selected in the list.

   7. In the **UI language** list, select the language in which you want to receive information about incidents in messages. Russian is selected by default.

6. Click **Add**.

7. In the toolbar of the list, enable the **Do not send incident information in CEF format** toggle switch.

   As a result, when new incidents are detected or the statuses of previously detected incidents change, depending on the statuses selected during configuration, the corresponding information is sent to the specified addresses of receiving servers via the syslog protocol in CEF format.

8. Click **Save changes**.

Export of events to the LogAnalyzer collector is configured.

Configuring receipt of Nextcloud events

You can configure the receipt of Nextcloud 26.0.4 events in the LogAnalyzer SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring audit of Nextcloud events.
2. Configuring a Syslog server to send events.

   The rsyslog service is used to transmit events from the server to the collector.
3. Creating a LogAnalyzer collector for receiving Nextcloud events.

   To receive Nextcloud events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Nextcloud syslog** normalizer, and at the **Transport** step, select the **tcp** or **udp** connector type.
4. Installing LogAnalyzer collector for receiving Nextcloud events
5. Verifying the receipt of Nextcloud events in the LogAnalyzer collector

   You can verify that the Nextcloud event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Configuring audit of Nextcloud events

To configure the export of Nextcloud events to LogAnalyzer:

1. On the server where Nextcloud is installed, create a backup copy of the `/home/localuser/www/nextcloud/config/config.php` configuration file.
2. Edit the `/home/localuser/www/nextcloud/config/config.php` Nextcloud configuration file.
3. Edit the settings as follows:

   ```
   'log_type' => 'syslog',
   'syslog_tag' => 'Nextcloud',
   'logfile' => '',
   'loglevel' => 0,
   'log.condition' => [
   'apps' => ['admin_audit'],
   ],
   ```
4. Restart the Nextcloud service:

   ```
   sudo service restart nextcloud
   ```

Export of events to the LogAnalyzer collector is configured.

Configuring a Syslog server to send Nextcloud events

To configure the sending of events from the server where Nextcloud is installed to the collector:

1. In the /etc/rsyslog.d/ directory, create a Nextcloud-to-siem.conf file with the following content:

   ```
   If $programname contains 'Nextcloud' then @<IP address of the collector>:<port of the collector>
   ```

   Example:

   ```
   If $programname contains 'Nextcloud' then @192.168.1.5:1514
   ```

   If you want to send events via TCP, the contents of the file must be as follows:

   ```
   If $programname contains 'Nextcloud' then @<IP address of the collector>:<port of the collector>
   ```

2. Save changes to the Nextcloud-to-siem.conf configuration file.

3. Create a backup copy of the /etc/rsyslog.conf file.

4. Add the following lines to the /etc/rsyslog.conf configuration file:

   ```
   $IncludeConfig /etc/Nextcloud-to-siem.conf
   ```

   ```
   $RepeatedMsgReduction off
   ```

5. Save your changes.

6. Restart the rsyslog service by executing the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

The export of Nextcloud events to the collector is configured.

Configuring receipt of Snort events

You can configure the receipt of Snort 3 events in the LogAnalyzer SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring logging of Snort events.

2. Creating a LogAnalyzer collector for receiving Snort events.

   To receive Snort events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Snort 3 json file** normalizer, and at the **Transport** step, select the **file** connector type.

3. Installing a LogAnalyzer collector for receiving Snort events

4. Verifying the receipt of Snort events in the LogAnalyzer collector

   You can verify that the Snort event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Configuring logging of Snort events

Make sure that the server running Snort has at least 500 MB of free disk space for storing a single Snort event log.
When the log reaches 500 MB, Snort automatically creates a new file with a name that includes the current time in unixtime format.
We recommend monitoring disk space usage.

To configure Snort event logging:

1. Connect to the server where Snort is installed using an account with administrator rights.
2. Edit the Snort configuration file. To do so, run the following command on the command line:

   ```
   sudo vi /usr/local/etc/snort/snort.lua
   ```

3. In the configuration file, edit the alert_json block:

   ```
   alert_json =
   {
   file = true,
   limit = 500,
   fields = 'seconds action class b64_data dir dst_addr dst_ap dst_port eth_dst eth_len \
   eth_src eth_type gid icmp_code icmp_id icmp_seq icmp_type iface ip_id ip_len msg mpls
   \
   pkt_gen pkt_len pkt_num priority proto rev rule service sid src_addr src_ap src_port \
   target tcp_ack tcp_flags tcp_len tcp_seq tcp_win tos ttl udp_len vlan timestamp',
   }
   ```

4. To complete the configuration, run the following command:

   ```
   sudo /usr/local/bin/snort -c /usr/local/etc/snort/snort.lua -s 65535 -k none -l
   /var/log/snort -i <name of the interface that Snort is listening on> -m 0x1b
   ```

   As a result, Snort events are logged to /var/log/snort/alert_json.txt.

Configuring receipt of Suricata events

You can configure the receipt of Suricata 7.0.1 events in the LogAnalyzer SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring the export of Suricata events to LogAnalyzer
2. Creating a LogAnalyzer collector for receiving Suricata events.

   To receive Suricata events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Suricata json file** normalizer, and at the **Transport** step, select the **file** connector type.

3. Installing LogAnalyzer collector for receiving Suricata events

4. Verifying the receipt of Suricata events in the LogAnalyzer collector

   You can verify that the Suricata event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Configuring audit of Suricata events.

To configure Suricata event logging:

1. Connect via SSH to the server that has administrative user accounts.

2. Create a backup copy of the /etc/suricata/suricata.yaml file.

3. Set the following values in the eve-log section of the /etc/suricata/suricata.yaml configuration file:

   ```
   - eve-log:
   enabled: yes
   filetype: regular #regular|syslog|unix_dgram|unix_stream|redis
   filename: eve.json
   ```

4. Save your changes to the /etc/suricata/suricata.yaml configuration file.

As a result, Suricata events are logged to the /usr/local/var/log/suricata/eve.json file.

Suricata does not support limiting the size of the eve.json event file. If necessary, you can manage the log size by using rotation. For example, to configure hourly log rotation, add the following lines to the configuration file:

```
outputs:

- eve-log:

filename: eve-%Y-%m-%d-%H:%M.json

rotate-interval: hour
```

Configuring receipt of FreeRADIUS events

You can configure the receipt of FreeRADIUS 3.0.26 events in the LogAnalyzer SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring audit of FreeRADIUS events.

2. Configuring a Syslog server to send FreeRADIUS events.

3. Creating a LogAnalyzer collector for receiving FreeRADIUS events.

   To receive FreeRADIUS events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] FreeRADIUS syslog** normalizer, and at the **Transport** step, select the **tcp** or **udp** connector type.

4. Installing LogAnalyzer collector for receiving FreeRADIUS events.

5. Verifying the receipt of FreeRADIUS events in the LogAnalyzer collector.

   You can verify that the FreeRADIUS event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Configuring audit of FreeRADIUS events

To configure event audit in the FreeRADIUS system:

1. Connect to the server where the FreeRADIUS system is installed using an account with administrator rights.

2. Create a backup copy of the FreeRADIUS configuration file:

   ```
   sudo cp /etc/freeradius/3.0/radiusd.conf /etc/freeradius /3.0/radiusd.conf.bak
   ```

3. Open the FreeRADIUS configuration file for editing:

   ```
   sudo nano /etc/freeradius/3.0/radiusd.conf
   ```

4. In the 'log' section, edit the settings as follows:

   ```
   destination = syslog
   syslog_facility = daemon
   stripped_names = no
   auth = yes
   auth_badpass = yes
   auth_goodpass = yes
   ```

5. Save the configuration file.

FreeRADIUS event audit is configured.

Configuring a Syslog server to send FreeRADIUS events

The rsyslog service is used to transmit events from the FreeRADIUS server to the LogAnalyzer collector.

To configure the sending of events from the server where FreeRADIUS is installed to the collector:

1. In the /etc/rsyslog.d/ directory, create the FreeRADIUS-to-siem.conf file and add the following line to it:

   ```
   If $programname contains 'radiusd' then @<IP address of the collector>:<port of the collector>
   ```

   If you want to send events via TCP, the contents of the file must be as follows:

   ```
   If $programname contains 'radiusd' then @<IP address of the collector>:<port of the collector>
   ```

2. Create a backup copy of the /etc/rsyslog.conf file.

3. Add the following lines to the /etc/rsyslog.conf configuration file:

   ```
   $IncludeConfig /etc/FreeRADIUS-to-siem.conf
   $RepeatedMsgReduction off
   ```

4. Save your changes.

5. Restart the rsyslog service:

   ```
   sudo systemctl restart rsyslog.service
   ```

The export of events from the FreeRADIUS server to the LogAnalyzer collector is configured.

Configuring receipt of VMware vCenter events

You can configure the receipt of VMware vCenter events in the LogAnalyzer SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring the connection to VMware vCenter.

2. Creating a LogAnalyzer collector for receiving VMware vCenter events.

   To receive VMware vCenter events, in the Collector Installation Wizard, at the **Transport** step, select the vmware connector type. Specify the required settings:

   - The URL at which the VMware API is available, for example, https://vmware-server.com:6440.
   - VMware credentials — a secret that specifies the username and password for connecting to the VMware API.

   At the **Event parsing** step, select the **[OOTB] VMware vCenter API** normalizer.

3. Installing a LogAnalyzer collector for receiving VMware vCenter events.

4. Verifying the receipt of VMware vCenter events in the LogAnalyzer collector.

   You can verify that the VMware vCenter event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Configuring the connection to VMware vCenter

To configure a connection to VMware vCenter to receive events:

1. Connect to the VMware Vcenter web interface under a user account with administrator rights.

2. Go to the **Security&Users** section and select **Users**.

3. Create a user account.

4. Go to the **Roles** section and assign the **Read-only: See details of objects role, but not make changes** role to the created account.

   You will use the credentials of this user account in the secret of the collector.

   For details about creating user accounts, refer to the VMware vCenter documentation.

The connection to VMware vCenter for receiving events is configured.

Configuring receipt of zVirt events

You can configure the receipt of zVirt 3.1 events in the LogAnalyzer SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring the export of zVirt events to LogAnalyzer.

2. Creating a LogAnalyzer collector for receiving zVirt events.

   To receive zVirt events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] OrionSoft zVirt syslog** normalizer, and at the **Transport** step, select the **tcp** or **udp** connector type.

3. Installing LogAnalyzer collector for receiving zVirt events

4. Verifying the receipt of zVirt events in the LogAnalyzer collector

   You can verify that the zVirt event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Configuring export of zVirt events

ZVirt can send events to external systems in Hosted Engine installation mode.

To configure the export of zVirt events to LogAnalyzer:

1. In the zVirt web interface, under **Resources**, select **Virtual machines**.

2. Select the machine that is running the HostedEngine virtual machine and click **Edit**.

3. In the **Edit virtual machine** window, go to the **Logging** section.

4. Select the **Determine Syslog server address** check box.

5. In the text box, enter the collector information in the following format: `<IP address or FQDN of the LogAnalyzer collector>: <port of the LogAnalyzer collector>`.

6. If you want to use TCP instead of UDP for sending logs, select the **Use TCP connection** check box.

Event export is configured.

Configuring receipt of Zeek IDS events

You can configure the receipt of Zeek IDS 1.8 events in the LogAnalyzer SIEM system.

Configuring event receiving consists of the following steps:

1. Conversion of the Zeek IDS event log format.

   The LogAnalyzer normalizer supports Zeek IDS logs in the JSON format. To send events to the LogAnalyzer normalizer, log files must be converted to the JSON format.

2. Creating a LogAnalyzer collector for receiving Zeek IDS events.

   To receive Zeek IDS events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] ZEEK IDS json file** normalizer, and at the **Transport** step, select the **file** connector type.

3. Installing LogAnalyzer collector for receiving Zeek IDS events

4. Verifying the receipt of Zeek IDS events in the LogAnalyzer collector

   You can verify that the Zeek IDS event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Conversion of the Zeek IDS event log format

By default, Zeek IDS events are logged in files in the /opt/zeek/logs/current directory.

The "[OOTB] ZEEK IDS json file" normalizer supports Zeek IDS logs in the JSON format. To send events to the LogAnalyzer normalizer, log files must be converted to the JSON format.

> This procedure must be repeated every time before receiving Zeek IDS events.

To convert the Zeek IDS event log format:

1. Connect to the server where Zeek IDS is installed using an account with administrator rights.

2. Create the directory where JSON event logs must be stored:

   ```
   sudo mkdir /opt/zeek/logs/zeek-json
   ```

3. Change to this directory:

   ```
   sudo cd /opt/zeek/logs/zeek-json
   ```

4. Run the command that uses the jq utility to convert the original event log format to the target format:

   ```
   jq . -c <path to the log file to be converted to a different format> >> <new file name>.log
   ```

   Example:

   ```
   jq . -c /opt/zeek/logs/current/conn.log >> conn.log
   ```

As a result of running the command, a new file is created in the /opt/zeek/logs/zeek-json directory if this file did not exist before. If the file was already present in the current directory, new information is appended to the end of the file.

Configuring Windows event reception using Kaspersky Endpoint Security for Windows

In Kaspersky Endpoint Security for Windows, starting from version 12.6, events can be sent from Windows logs to a LogAnalyzer collector. In this way, LogAnalyzer can get events from Windows logs (Kaspersky Endpoint Security for Windows версии 12.6–12.10 supports a limited set of EventIDs of Microsoft products) from all hosts with versions of Kaspersky Endpoint Security for Windows 12.6 and later without installing LogAnalyzer agents on such hosts. To activate the functionality, you need:

- A valid LogAnalyzer license
- Kaspersky Security Center 14.2 or later
- Kaspersky Endpoint Security for Windows 12.6 or later

Configuring event receiving consists of the following steps:

1. **Importing the normalizer into LogAnalyzer.**

   In LogAnalyzer, you must configure updating via the Kaspersky update servers.

   Click Import resources and in the list of normalizers available for installation, select **[OOTB] Microsoft Products via KES WIN**.

2. **Creating a LogAnalyzer collector for receiving Windows events.**

   Depending on the version of Kaspersky Endpoint Security for Windows, you need to do the following:

   - If you are using Kaspersky Endpoint Security for Windows version 12.6–12.10, at the **Transport** step, select **TCP** or **UDP** and specify the port number that the connector will listen on. At the **Event parsing** step, select the **[OOTB] Microsoft Products via KES WIN** normalizer, and at the **Event filtering** step, select the **[OOTB] Microsoft Products via KES WIN - Event filter for collector** filter.

   - If you are using Kaspersky Endpoint Security for Windows version 12.11 or later, at the **Transport** step, select **TCP** or **UDP** and specify the port number that the connector will listen on. At the **Event parsing** step, select one of the following options:

     - For LogAnalyzer version 3.2–4.0, select the **[OOTB] Microsoft Products via KES WIN NG for LogAnalyzer 3.2** normalizer.

     - For LogAnalyzer version 4.0.1 or later, select the **[OOTB] Microsoft Products via KES WIN NG for LogAnalyzer 4.0.1** normalizer.

3. **Requesting a key from Technical Support**

   If your license does not cover the functionality of sending Windows logs to the LogAnalyzer collector, send the following message to Technical Support: "We have purchased a LogAnalyzer license and are using Kaspersky Endpoint Security for Windows version 12.6. We want to activate the functionality of sending Windows logs to the LogAnalyzer collector. Please provide a key file to activate the functionality." New LogAnalyzer users do not need to make a Technical Support request because new users get two license keys: one for LogAnalyzer and one for activating the Kaspersky Endpoint Security for Windows functionality.

   In response to your message, you will get a key file.

4. **Configuration on the side of Kaspersky Security Center and Kaspersky Endpoint Security for Windows.**

   A key file that activates the functionality of sending Windows events to LogAnalyzer collectors must be imported into Kaspersky Security Center and distributed to Kaspersky Endpoint Security endpoints in accordance with the instructions. You must also add LogAnalyzer server addresses to the Kaspersky Endpoint Security policy and specify network connection settings.

5. **Verifying the receipt of Windows events in the LogAnalyzer collector**

   You can verify that the Windows event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

   The list of events from Microsoft products that Kaspersky Endpoint Security for Windows version 12.6–12.10 sends to LogAnalyzer is given in the following table:

Event log	Event identifier
System	12 13 7040 7045 42 104 107 109 1074 6005 6006 7034 7036 8003
Security	1102 4614 4649 4696 4698 4704 4706 4713 4715 4717 4720 4723 4724 4725 4726 4727 4728 4729 4738 4739 4740 4741 4742 4756 4757 4765 4766 4767 4768 4769 4770 4771 4775 4776 4778 4780 4781 4794 4817 4876 4877 4882 4885 4886 4887 4890 4891 4898 4904 4905 4928 4950 4964 5136 5137 5138 5139 5141 5142 5143 5144 5155 5376 5377

Event log	Event identifier
	5632 5888 5890 6416 4622 4648 4672 4697 4702 4719 4732 4733 4798 4946 4947 4948 4949 5145 4616 4625 4663 4624 4799 5140 1008 1105 2722 4615 4618 4626 4627 4634 4647 4653 4654 4656 4657 4658 4659 4660 4661 4664 4666 4667 4670 4673 4674 4688 4689 4690 4691 4692 4693 4694 4695 4699 4700 4701 4703 4705 4707 4714 4716 4718 4730 4731 4734 4737 4743 4744 4745 4746 4747 4748 4749 4750 4751 4752 4753 4754

Event log	Event identifier
	4755 4758 4759 4760 4761 4762 4763 4764 4772 4773 4774 4777 4779 4782 4783 4784 4785 4786 4787 4788 4789 4790 4791 4792 4793 4797 4800 4801 4802 4803 4818 4819 4820 4821 4822 4823 4824 4825 4826 4865 4866 4867 4868 4869 4870 4871 4872 4873 4874 4875 4883 4884 4888 4892 4893 4896 4906 4907 4908 4911 4912 4913 4929 4930 4931 4932 4933 4935 4944 4945 4951 4953 4956 4957 4958 4981 4982 4983 4984 4985

Event log	Event identifier
	5024 5031 5033 5039 5049 5051 5056 5057 5058 5059 5060 5061 5063 5064 5065 5066 5067 5068 5069 5070 5071 5122 5123 5146 5147 5152 5153 5154 5156 5157 5158 5159 5168 5378 5379 5380 5381 5382 5447 5448 5451 5452 5459 5461 5472 5474 5477 5478 5483 5484 5633 6144 6145 6272 6273 6274 6276 6278 6279 6280 6281 6410 6419 6420 6421 6422 6423 6424 6889

Event log	Event identifier
PowerShell	4100 4103 4104 4105 4106 8193 8194 8197 24577 24595 24596 24597 24598 24599 53249 53250 53504
MS SQL Server	615 919 958 1945 2007 2812 3406 3407 3421 3454 5084 5579 5701 5703 6253 8128 9013 9666 15268 15457 17104 17110 17111 17125 17137 17152 17164 17176 17177 17199 17201 17550 17551 17561 17663 18264 18265 18456 18488 18496 19030 19031 19032 26022 26037 26048 26067 33090 49903 49904

Event log	Event identifier
Microsoft Defender	1006 1015 1116 1117 1000 1001 2000 5000 5001 5002 5004 5007 5010 5012
Terminal Server	1149 21 22 24 25 39 40
Microsoft Active Directory Federation Service (AD FS)	106 217 251 335 342 349 358 364 381 385 400 401 417 424 435 436
Sysmon	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 255
Microsoft Active Directory Domain Service (AD DS)	1213 1317 1644 2041 2889

Configuring Linux event reception using Kaspersky Endpoint Security for Linux

In Kaspersky Endpoint Security for Linux, starting from version 12.2, events can be sent from Linux logs to a LogAnalyzer collector. This allows LogAnalyzer to receive events from Linux logs from all hosts on which Kaspersky Endpoint Security for Linux version 12.2 is installed. To activate the functionality, you need:

- A valid LogAnalyzer license
- Kaspersky Security Center 14.2 or later
- Kaspersky Endpoint Security for Windows 12.2 or later

Configuring event receiving consists of the following steps:

1. Importing the normalizer into LogAnalyzer.

   In LogAnalyzer, you must configure updating via the Kaspersky update servers.

   Click Resource import and select **[OOTB] KESL syslog cef** in the list of available normalizers.

2. Creating a LogAnalyzer collector for receiving Linux events.

   To receive Linux events, at the **Transport** step, select **TCP** or **UDP** and specify the port number that the collector must listen on. At the **Event parsing** step, select the **[OOTB] KESL syslog cef** normalizer.

3. Requesting a key from Technical Support.

   If your license does not cover the functionality of sending Linux logs to the LogAnalyzer collector, send the following message to Technical Support: "We have purchased a LogAnalyzer license and are using Kaspersky Endpoint Security for Linux version 12.2. We want to activate the functionality of sending Linux logs to the LogAnalyzer collector. Please provide a key file to activate the functionality." New LogAnalyzer users do not need to make a Technical Support request because new users get two license keys: one for LogAnalyzer and one for activating the Kaspersky Endpoint Security for Linux functionality.

   In response to your message, you will get a key file.

4. Configuration on the side of Kaspersky Security Center and Kaspersky Endpoint Security for Linux.

   A key file that activates the functionality of sending Linux events to LogAnalyzer collectors must be imported into Kaspersky Security Center and distributed to Kaspersky Endpoint Security endpoints in accordance with the instructions. You must also add LogAnalyzer server addresses to the Kaspersky Security Center policy and specify network connection settings.

5. Verifying the receipt of Linux events in the LogAnalyzer collector

   You can verify that the Linux event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

   Kaspersky Endpoint Security for Linux sends the following events:

   - ProcessCreate
   - ProcessTerminate
   - FileChange
   - New user account created
   - User account deleted
   - Group created
   - Group deleted
   - New member added to a group
   - Member removed from the group
   - User password changed
   - Linux authentication performed
   - Linux session started
   - Linux session finished
   - Service started
   - Service stopped
   - Promiscuous mode modified
   - Audit configuration changed
   - Account expiration date changed

Configuring receipt of Codemaster Mirada events

You can configure the receipt of Codemaster Mirada events in the LogAnalyzer SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring audit of the Codemaster Mirada system
2. Creating a LogAnalyzer collector for receiving Codemaster Mirada events.

   To receive Codemaster Mirada events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Codemaster Mirada syslog** normalizer, and at the **Transport** step, select the **tcp** or **udp** connector type.
3. Installing a collector in the LogAnalyzer network infrastructure.
4. Verifying the receipt of Codemaster Mirada events in the LogAnalyzer collector

   You can verify that the Codemaster Mirada event source server is correctly configured in the **Events** section of the LogAnalyzer web interface.

Configuring audit of the Codemaster Mirada system

The Codemaster Mirada system can send events over the Syslog protocol.

To configure event audit in the Codemaster Mirada system:

1. Connect to the Codemaster Mirada web interface under a user account with administrator rights.
2. Go to the **Settings** → **Syslog** section.
3. Enable the event transmission service using the toggle switch.
4. Select the type and format of the protocol by clicking the dot-in-a-circle icon.
5. In the **Host** field, specify the IP address of the LogAnalyzer collector.
6. In the **Protocol** field, specify the **UDP** or **TCP** transport protocol.
7. In the **Port** field, specify the port that the LogAnalyzer collector is listening on.

   The default port is 514.
8. In the **Format** field, specify the **RFC 3164** standard.
9. Click **Save** in the lower part of the page to save the changes.

Configuring receipt of Postfix events

You can configure the receipt of Postfix events in LogAnalyzer. Integration is only possible when sending events via syslog using the TCP protocol. The resources described in this article are available for LogAnalyzer 3.0 and newer versions.

Configuring event receiving consists of the following steps:

1. Configuring Postfix to send events.
2. Configuring the LogAnalyzer collector for receiving Postfix events
3. Verifying the receipt of Postfix events in the LogAnalyzer collector

   You can verify that the Postfix event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

The Postfix system generates events in two formats:

- Multi-line events containing information about messages (with a unique ID). These events have the following form:

  <syslog PRI> time host process_name: ID: information from base event 1

  <syslog PRI> time host process_name: id: info from base event 2

- Single-line events containing information about errors (without an ID). These events have the following form:

  <syslog PRI> time host process_name: severity: basic information for parsing

A set of LogAnalyzer resources is used to process Postfix events; this resource set must be applied when creating a collector:

- Normalizer
- Aggregation rule
- Filters for destinations

While the collector is running, multi-line base events are be aggregated based on the event ID and normalized, and a single event is sent to the storage.

The aggregated event has the following form:

Service information from the aggregation rule: ID: information from base event 1, information from base event 2, information from base event n

After aggregation, the received event is sent to the same collector where the aggregated event is normalized.

Processing algorithm for Postfix events

The following algorithm was implemented to process Postfix events:

1. Initial normalization

   At this stage, initial normalization is performed for base events received via syslog that begin with the "<" character. The events are brought to a format suitable for subsequent aggregation: the first character is extracted from the event and put into the FlexString1 field, the identifier is put into the ExternalID field, and the host name is put into the DeviceHostName field. Basic normalization is performed in the main normalizer.

2. Checking for aggregation

   The event is examined to see if it is aggregated or not. As a result, non-aggregated events (the first character is not "{" and the ID is not empty) have an aggregation rule applied, and then aggregated events are sent to re-normalization.

3. Applying the aggregation rule

   At this stage, the aggregation rule is applied to the events, the base events are collated and take the following form:

   Service information from the aggregation rule: ID: information from base event 1, information from base event 2, information from base event n

4. After aggregation, the collated event is sent back to the same collector to subject the aggregated event to normalization.

   To close the event processing loop, you must specify the same collector as the destination. In the diagram, the destination is named "Loop" to draw attention to the event processing loop. You can give an arbitrary name to your destination.

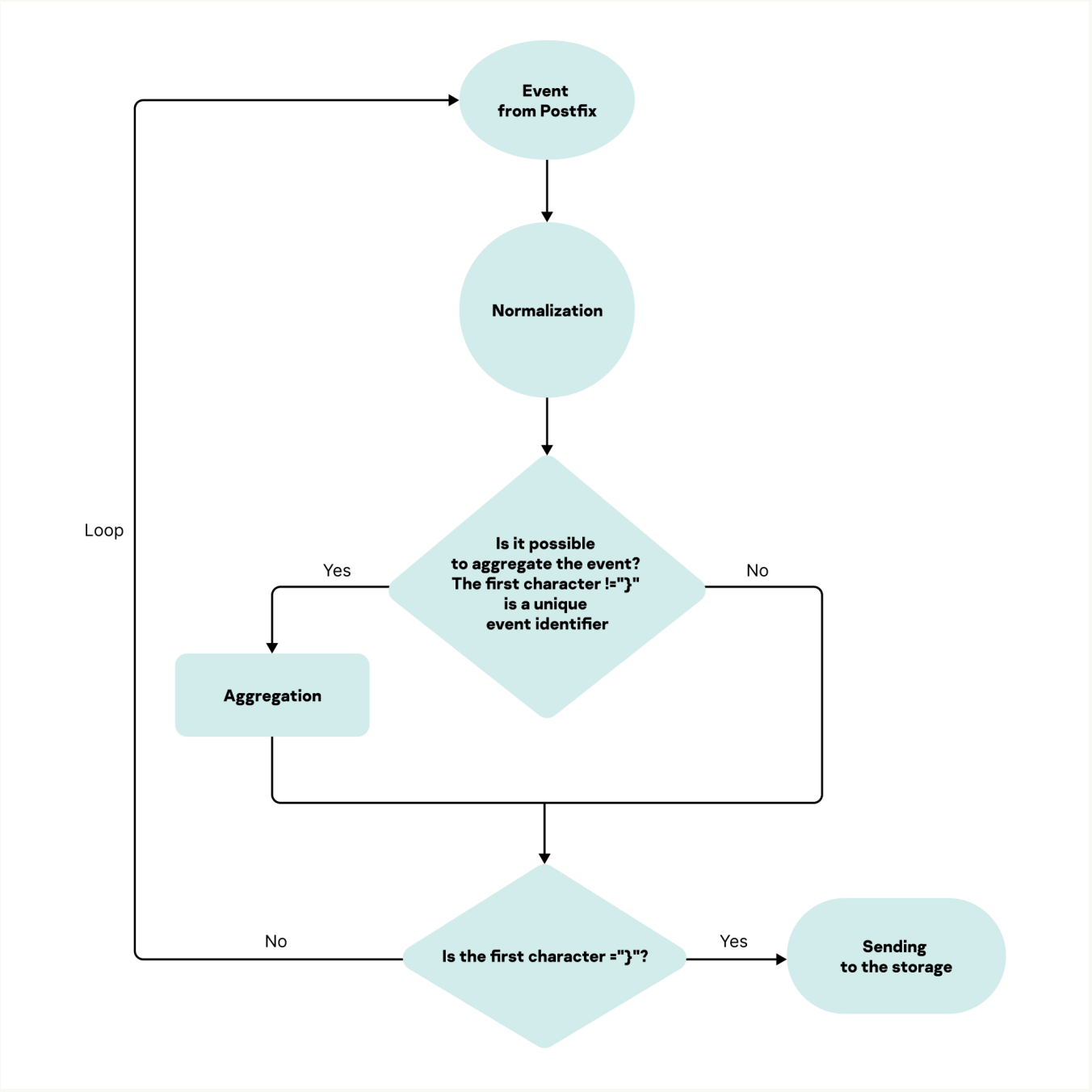
5. Normalization of the aggregated event

   Normalization of the aggregated event that begins with a "{" character is performed in the following extra normalizers: Aggregated events, Aggregated events. Message KV parser, Aggregated events. Message regex 1, Aggregated events. Message regex 2.

6. Sending to storage and the correlator

   Aggregated and normalized events are sent to storage and the correlator.

The following figure shows the flow chart of Postfix event processing.



Processing diagram for events from the Postfix system

Configuring Postfix to send events

By default, audit events of the Postfix system are output to /var/log/maillog or /var/log/mail.

To send events to LogAnalyzer:

1. Create a backup copy of the /etc/rsyslog.conf file.
2. Open the /etc/rsyslog.conf file for editing.
3. Add the following line to the end of the /etc/rsyslog.conf file:

   ```
   mail.* @@<IP address of the KUMA collector>:<port of the KUMA collector>
   ```

4. Save the /etc/rsyslog.conf file.

5. Restart the rsyslog service:

```
sudo systemctl restart rsyslog
```

Configuring a LogAnalyzer collector for receiving and processing Postfix events

To configure a LogAnalyzer collector for receiving Postfix events:

1. Import the [OOTB] Postfix package from the LogAnalyzer repository. The package is available for LogAnalyzer 3.0 and newer versions.

2. Create a new collector, and in the Collector Installation Wizard, configure the following:

    1. At the **Transport** step, in the **Type** field, select the **tcp** type, and in the **URL** field, specify the FQDN or IP address and port of the collector.

    2. At the **Event parsing** step, click **Add event parsing**, and in the displayed **Basic event parsing** window, in the **Normalizer** drop-down list, select the **[OOTB] Postfix syslog** normalizer.

    3. At the **Event aggregation** step, click **Add aggregation** rule, and in the displayed **Event aggregation** window, in the **Aggregation rule** drop-down list, select **[OOTB] Postfix. Aggregation rule**.

    4. At the **Routing** step, click **Add** and in the displayed **Create destination** window, create two destination points one by one—the same collector with the name `Loop`, and a storage.

        a. Create a destination named `Loop` with the following parameters.

        - On the **Basic settings** tab, in the Type drop-down list, select the **tcp** transport type; in the **URL** field, specify the FQDN or IP address and port of the collector that you specified before at step 2.1 of these instructions.

        - On the **Advanced settings** tab, in the **Filter** drop-down list, select the **Postfix. Filter for event aggregation** filter.

          This configuration is necessary to send the aggregated event to the same collector for subsequent normalization.

        b. Create a storage destination:

        - On the **Basic settings** tab, in the **Type** drop-down list, select **storage** and fill in the **URL** field.

        - On the **Advanced settings** tab, in the **Filter** drop-down list, select the **Postfix. Aggregated events to storage and correlator** filter.

          This configuration is necessary to send the aggregated normalized event to the storage.

3. Click the **Create** button.

   The collector service is created with the settings specified in the LogAnalyzer web interface. The command for installing the service on the server is displayed.

4. Copy the collector installation command and run it on the relevant server.

The collector is configured to receive and process Postfix events.

Configuring receipt of CommuniGate Pro events

You can configure the receipt of CommuniGate Pro 6.1 and 6.5.1 events in LogAnalyzer. Integration is only possible when sending events via syslog using the TCP protocol. The resources described in this article are available for LogAnalyzer 3.0 and newer versions. Processing of some SIP module events is supported (such events contain the "SIPDATA" character sequence).

Configuring event receiving consists of the following steps:

1. Configuring CommuniGate Pro to send events
2. Configuring the LogAnalyzer collector for receiving CommuniGate Pro events
3. Verifying the receipt of CommuniGate Pro events in the LogAnalyzer collector

   You can verify that the CommuniGate Pro event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

The CommuniGate Pro system generates an audit event as several separate records that look like this:

```
<event code> timestamp ID direction: information from base event 1

<event code> timestamp ID direction: information from base event 2

<event code> timestamp ID direction: base information n
```

A set of LogAnalyzer resources is used to process CommuniGate Pro events; this resource set must be applied when creating a collector:

- Normalizer
- Aggregation rule
- Filters for destinations

While the collector is running, multi-line base events are be aggregated based on the event ID and normalized, and a single event is sent to the storage.

The aggregated event has the following form:

```
Service information from the aggregation rule: ID: information from base event 1,
information from base event 2, information from base event n
```

After aggregation, the received event is sent to the same collector where the aggregated event is normalized.

Processing algorithm for CommuniGate Pro events

The following algorithm was implemented to process CommuniGate Pro events:

1. Initial normalization

   At this stage, the initial normalization of base events is performed. The first character in the base event is a numeral. The events are brought to a format suitable for subsequent aggregation: the first character is extracted from the event and put into the DeviceCustomString1 field, the identifier is put into the ExternalID field, and the host name is put into the DeviceHostName field. Basic normalization is performed in the main normalizer.

2. Checking for aggregation

   The event is examined to see if it is aggregated or not. As a result, non-aggregated events (the first character is a numeral) have an aggregation rule applied, and then aggregated events are sent to re-normalization. Aggregation is performed using the **[OOTB] CommuniGate Pro. Aggregation rule**.

3. Applying the aggregation rule

   At this stage, the aggregation rule is applied to the events, the base events are collated and take the following form:

   Service information from the aggregation rule: ID: information from base event 1, information from base event 2, information from base event n

   After aggregation, the collated event is sent back to the same collector to subject the aggregated event to normalization.

   To close the event processing loop, you must specify the same collector as the destination. In the diagram, the destination is named "Loop" to draw attention to the event processing loop. You can give an arbitrary name to your destination.

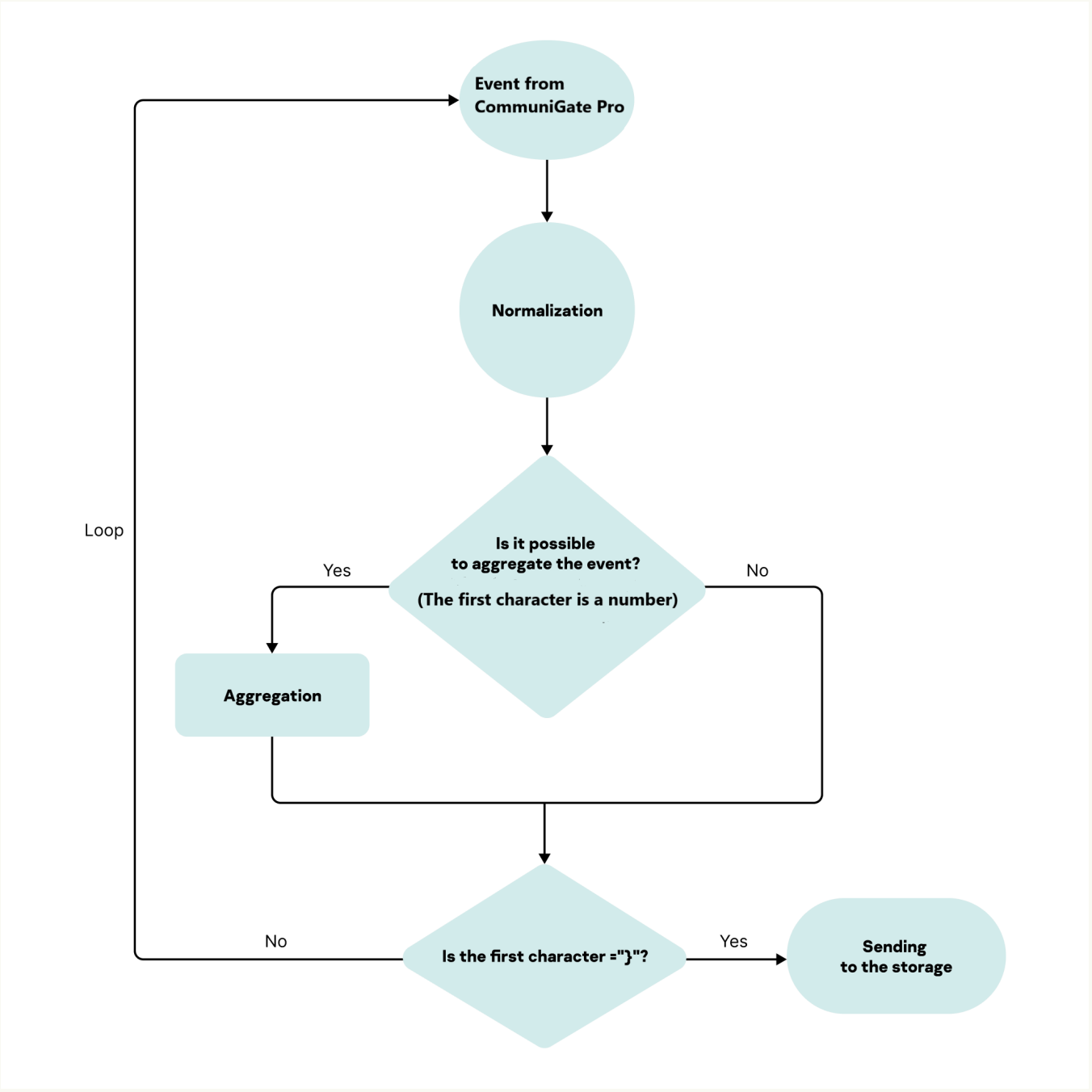
4. Normalization of the aggregated event

   Normalization of the aggregated event that begins with a "{" character is performed in the following extra normalizers: Aggregated events, Aggregated events - kv part.

5. Sending to storage and the correlator

   Aggregated and normalized events are sent to storage and the correlator.

The following figure shows the flow chart of CommuniGate Pro event processing.



Processing diagram for events from the CommuniGate Pro system

Configuring CommuniGate Pro to send events

By default, CommuniGate Pro audit events are sent to .log files in the /var/CommuniGate/SystemLogs/ directory.

To send events to LogAnalyzer, you need to install the LogAnalyzer agent on the CommuniGate Pro server and configure it to read .log in the /var/CommuniGate/SystemLogs/ directory and send them to the LogAnalyzer collector over TCP.

To create an agent that will read and send events to LogAnalyzer:

1. In the LogAnalyzer web console, go to **Resources and services** → **Agents** and click **Add**.
2. This opens the **Create agent** window; in that window, on the **Basic settings** tab, in the **Name** field, specify the agent name.
3. On the **Config #1** tab, fill in the following fields:
   a. In the **Connector** group of settings on the **Basic settings** tab, set the following values for the connector:
      1. In the **Name** field, enter a name, for example, `CommuniGate file`.
      2. In the **Type** drop-down list, select **file**.
      3. In the **File path** field, enter the following value:
         /var/CommuniGate/SystemLogs/.*.log
   b. In the **Destinations** group of settings on the **Basic settings** tab, set the following values for the destination:
      1. In the **Name** field, enter a name, for example, `CommuniGate TCP collector`.
      2. In the **Type** drop-down list, select **tcp**.
      3. In the **URL** field, enter the FQDN or IP address and port of the LogAnalyzer collector.
4. Click the **Create** button.
5. When the agent service is created in LogAnalyzer, install the agent on the network infrastructure devices from which you want to send data to the collector.

Configuring a LogAnalyzer collector for receiving and processing CommuniGate Pro events

To configure a LogAnalyzer collector for receiving CommuniGate Pro events:

1. Import the [OOTB] CommuniGate Pro package from the LogAnalyzer repository. The package is available for LogAnalyzer 3.0 and newer versions.

2. Create a new collector, and in the Collector Installation Wizard, configure the following:

    1. At the **Transport** step, in the **Type** field, select the **tcp** type, and in the **URL** field, specify the FQDN or IP address and port of the collector.

    2. At the **Event parsing** step, click **Add event parsing**, and in the displayed **Basic event parsing** window, in the **Normalizer** drop-down list, select the **[OOTB] CommuniGate Pro** normalizer.

    3. At the **Event aggregation** step, click **Add aggregation rule**, and in the displayed **Event aggregation** window, in the **Aggregation rule** drop-down list, select **[OOTB] CommuniGate Pro. Aggregation rule**.

    4. At the **Routing** step, click **Add** and in the displayed **Create destination** window, create two destination points one by one—the same collector with the name `Loop`, and a storage.

        a. Create a destination named `Loop` with the following parameters.

        - On the **Basic settings** tab, in the **Type** drop-down list, select the **tcp** transport type; in the **URL** field, specify the FQDN or IP address and port of the collector that you specified before at step 2.1 of these instructions.

        - On the **Advanced settings** tab, in the **Filter** drop-down list, select the **[OOTB] CommuniGate Pro. Filter for event aggregation** filter.

          This configuration is necessary to send the aggregated event to the same collector for subsequent normalization.

        a. Create a storage destination:

        - On the **Basic settings** tab, in the **Type** drop-down list, select **storage** and fill in the **URL** field.

        - On the **Advanced settings** tab, in the **Filter** drop-down list, select the **[OOTB] CommuniGate Pro. Aggregated events to storage and correlator** filter.

          This configuration is necessary to send the aggregated normalized event to the storage.

3. Click the **Create** button.

    The collector service is created with the settings specified in the LogAnalyzer web interface. The command for installing the service on the server is displayed.

4. Copy the collector installation command and run it on the relevant server.

The collector is configured to receive and process CommuniGate Pro events.

Configuring receipt of Yandex Cloud events

You can configure the receipt of Yandex Cloud events in LogAnalyzer. This normalizer supports processing configuration-level audit events stored in .json files.

Configuring event receiving consists of the following steps:

1. Configuring audit of Yandex Cloud events.
2. Configuring export of Yandex Cloud events.
3. Configuring a LogAnalyzer collector for receiving and processing Yandex Cloud events.

   To receive Yandex Cloud events in the LogAnalyzer Collector Installation Wizard:

   1. In the LogAnalyzer Collector Installation Wizard, at the **Transport** step, select the connector of the **file** type.
   2. In the **URL** field, enter `/var/log/yandex-cloud/<audit_trail_id>/*/*/*/*.json`, where <audit_trail_id> is the ID of the audit.
   3. At the **Event parsing** step, in the **Normalizer** field, select **[OOTB] Yandex Cloud**.
4. Installing a collector in the LogAnalyzer network infrastructure.
5. Verifying the receipt of Yandex Cloud events in the LogAnalyzer collector

   To verify that the Yandex Cloud event source server is configured correctly, you can search for related events.

Configuring audit of Yandex Cloud events

Configuring event export involves the following steps:

1. Preparing the environment for working with Yandex Cloud.
2. Creating a bucket for audit logs.
3. Creating an encryption key in the Key Management Service.
4. Enabling bucket encryption.
5. Creating service accounts.
6. Creating a static key.
7. Assigning roles to service accounts.
8. Creating an audit trail.

Preparing the environment for working with Yandex Cloud.

To manage the configuration, you need Yandex Cloud CLI; install and initialize it.

Note: by default, audit is performed in the Yandex Cloud folder specified in the CLI profile. You can specify a different folder using the `--folder-name` or `--folder-id` parameter.

To configure the audit, you need an active billing account because a fee is charged for using the Yandex Cloud infrastructure.

To configure Yandex Cloud audit, you need an active billing account:

1. Go to the management console, then log in to Yandex Cloud or register.
2. On the Yandex Cloud Billing page, make sure that you have a billing account connected and that it has the ACTIVE or TRIAL_ACTIVE status. If you do not have a billing account, create one.

If you have an active billing account, you can create or select a Yandex Cloud folder in which your infrastructure will work, on the cloud page.

Creating a bucket for audit logs

To create a bucket:

1. In the management console, go to the folder in which you want to create the bucket, for example, **example-folder**.
2. Select the **Object Storage** service.
3. Click **Create bucket**.
4. On the bucket creation page:
    1. Enter the bucket name in accordance with the naming rules, for example **kumabucket**.
    2. If necessary, limit the maximum size of the bucket. Size 0 means no limit and is equivalent to the enabled **No limit** option.
    3. Select the type of access: **Restricted**.
    4. Select the default storage class.
    5. Click **Create bucket**.

The bucket is created.

Creating an encryption key in the Key Management Service

To create an encryption key:

1. In the management console, go to the **example-folder** folder.
2. Select the **Key Management Service**.

3. Click the **Create key** button and specify the following settings:
   - Name (for example, **kuma-kms**).
   - Encryption algorithm, **AES-256**.
   - Keep default values for the rest of the settings.
4. Click **Create**.

   The encryption key is created.

Enabling bucket encryption

To enable bucket encryption:

1. In the management console, go to the bucket you created earlier.
2. In the left pane, select **Encryption**.
3. In the **KMS key** field, select the **kuma-kms** key.
4. Click **Save**.

   Bucket encryption is enabled.

Creating service accounts

To create service accounts (a separate account for the trail and a separate account for the bucket):

1. Create the **sa-kuma** service account:
    1. In the management console, go to the **example-folder** folder.
    2. In the upper part of the scree, go to the **Service accounts** tab.
    3. Click **Create service account** and enter the name of the service account, for example, **sa-kuma**, making sure the name complies with the naming rules:
        - length: 3 to 63 characters
        - may contain lower-case letters of the Latin alphabet, numerals, and hyphens
        - the first character must be a letter, the last character may not be a hyphen.
    4. Click **Create**.
2. Create the **sa-kuma-bucket** service account:
    1. In the management console, go to the **example-folder** folder.
    2. In the upper part of the scree, go to the **Service accounts** tab.
    3. Click **Create service account** and enter the name of the service account, for example, **sa-kuma-bucket**, making sure the name complies with the naming rules:
        - length: 3 to 63 characters
        - may contain lower-case letters of the Latin alphabet, numerals, and hyphens
        - the first character must be a letter, the last character may not be a hyphen.
    4. Click **Create**.

The service accounts are created.

Creating a static key

You will need the key ID and the private key when mounting the bucket. You can create a key using the management console or the CLI.

To create a key using the management console:

1. In the management console, go to the **example-folder** folder.
2. In the upper part of the screen, go to the **Service accounts** tab.
3. Select the **sa-kuma-bucket** service account and click the row with its name.
4. In the upper panel, click **Create new key**.

5. Select **Create static access key**.
6. Enter a description for the key and click **Create**.
7. Save the ID and the secret key.

The static access key is created. The key value will become unavailable when you close the dialog.

To create a key using the CLI:

1. Create an access key for the **sa-kuma-bucket** service account:

   ```
   yc iam access-key create --service-account-name sa-kuma-bucket
   ```

   Result:

   ```
   access_key:
   id: aje*******k2u
   service_account_id: aje*******usm
   created_at: "2022-09-22T14:37:51Z"
   key_id: 0n8*******0YQ
   secret: JyT*******zMP1
   ```

2. Save the key_id and the key from the 'secret' value. You will not be able to get the key value again.

The access key is created.

Assigning roles to service accounts

To assign the audit-trails.viewer, storage.uploader, and kms.keys.encrypterDecrypter roles to the sa-kuma service account:

1. In the CLI, assign the audit-trails.viewer role to the folder:

   ```
   yc resource-manager folder add-access-binding \
   --role audit-trails.viewer \
   --id <folder_id> \
   --service-account-id <service_account_id>
   ```

   Where:

   - `--role` is the assigned role.
   - `--id` is the ID of the 'example-folder' folder.
   - `--service-account-id` is the ID of the sa-kuma service account.

2. Assign the storage.uploader role to the folder with the bucket:

   ```
   yc resource-manager folder add-access-binding \
   --role storage.uploader \
   --id <folder_id> \
   --service-account-id <service_account_id>
   ```

   Where:

   - `--role` is the assigned role.
   - `--id` is the ID of the 'example-folder' folder.
   - `--service-account-id` is the ID of the sa-kuma service account.

3. Assign the kms.keys.encrypterDecrypter role to the kuma-kms encryption key:

   ```
   yc kms symmetric-key add-access-binding \
   --role kms.keys.encrypterDecrypter \
   --id <key_id> \
   --service-account-id <service_account_id>
   ```

   Where:

   - `--role` is the assigned role.
   - `--id` is the ID of the kuma-kms KMS key.
   - `--service-account-id` is the ID of the sa-kuma service account.

To assign the storage.viewer and kms.keys.encrypterDecrypter roles to the sa-kuma-bucket service account:

1. In the CLI, assign the storage.viewer role to the folder:

   ```
   yc resource-manager folder add-access-binding \
   --id <folder_id> \
   --role storage.viewer \
   --service-account-id <service_account_id>
   ```

   Where:

   - `--id` is the ID of the 'example-folder' folder.
   - `--role` is the assigned role.
   - `--service-account-id` is the ID of the sa-kuma-bucket service account.

2. Assign the kms.keys.encrypterDecrypter role to the kuma-kms encryption key:

   ```
   yc kms symmetric-key add-access-binding \
   --role kms.keys.encrypterDecrypter \
   --id <key_id> \
   --service-account-id <service_account_id>
   ```

   Where:

   - `--role` is the assigned role.
   - `--id` is the ID of the kuma-kms KMS key.
   - `--service-account-id` is the ID of the sa-kuma-bucket service account.

Creating an audit trail

To create an audit trail:

1. In the management console, go to the **example-folder** folder.
2. Select the **Audit Trails** service.
3. Click **Create trail** and specify a name for the trail you are creating, for example, **kuma-trail**.
4. In the **Destination** section, specify the parameters of the destination object:
   - **Destination**: **Object Storage**.
   - **Bucket**: The name of the bucket, for example **kumabucket**.
   - **Object prefix**: Optional parameter used in the full name of the audit log file.

     Use a prefix if you store audit logs and third-party data in the same bucket. Do not use the same prefix for logs and other objects in the bucket because this may cause logs and third-party objects to overwrite each other.
   - **Encryption key**: specify the kuma-kms encryption key that the bucket is encrypted with.

5. In the **Service account** section, select **sa-kuma**.
6. In the **Collecting management events** section, specify the settings for collecting management events audit logs:
   - **Collecting events**: select **Enabled**.
   - **Resource**: select **Folder**.
   - **Folder**: does not require filling, contains the name of the current folder.
7. In the **Collecting data events**, in the **Collecting events** field, select **Disabled**.
8. Click **Create**.

Configuring export of Yandex Cloud events

The bucket must be mounted on the server on which the LogAnalyzer collector will be installed.

To mount the bucket:

1. On the server, create a directory for the 'kuma' user:
   ```
   sudo mkdir /home/kuma
   ```
2. On the server, create a file with a static access key for the sa-kuma-bucket service account and grant appropriate access permissions to the 'kuma' user:
   ```
   sudo bash -c 'echo <access_key_ID>:<secret_access_key> > /home/kuma/.passwd-s3fs'
   sudo chmod 600 /home/kuma/.passwd-s3fs
   sudo chown -R kuma:kuma /home/kuma
   ```
3. Install the s3fs package:
   ```
   sudo apt install s3fs
   ```
4. Create a directory in which you want to mount the bucket and grant permissions to the kuma user:
   ```
   sudo mkdir /var/log/yandex-cloud/
   sudo chown kuma:kuma /var/log/yandex-cloud/
   ```

5. Mount the bucket:

   ```
   sudo s3fs kumabucket /var/log/yandex-cloud -o passwd_file=/home/kuma/.passwd-s3fs -o url=https://storage.yandexcloud.net -o use_path_request_style -o uid=$(id -u kuma) -o gid=$(id -g kuma)
   ```

   You can configure the bucket to be mounted at operating system startup by adding a line to /etc/fstab, for example:

   ```
   s3fs#kumabucket /var/log/yandex-cloud fuse _netdev,uid=<kuma_uid>,gid=<kuma_gid>,use_path_request_style,url=https://storage.yandexcloud.net,passwd_file=/home s3fs 0 0
   ```

   Where:

   <kuma_uid> is the ID of the 'kuma' user in the operating system.

   <kuma_gid> is the ID of the 'kuma' group in the operating system.

   To find out the kuma_uid and kuma_gid, run the following command in the console:

   ```
   id kuma
   ```

6. Verify that the bucket is mounted:

   ```
   sudo ls /var/log/yandex-cloud/
   ```

   If everything is configured correctly, the command returns <audit_trail_id>, where <audit_trail_id> is the audit trail ID.

Export of Yandex Cloud events is configured. Events will be located in directories in .json files:

/var/log/yandex-cloud/{audit_trail_id}/{year}/{month}/{day}/*.json

Configuring receipt of Microsoft 365 events

You can configure the receipt of events from the Microsoft 365 (Office 365) cloud solution in LogAnalyzer.

Configuring event receiving consists of the following steps:

1. Configuring access to Office 365 management APIs https://learn.microsoft.com/en-us/office/office-365-management-api/get-started-with-office-365-management-apis using standard Microsoft methods

   To receive events in LogAnalyzer, grant the necessary set of API permissions:

   Microsoft.Graph

   Directory.Read.All

   Office 365 management API

   ActivityFeed.Read

   ActivityFeed.Read.Dlp

2. Creating a LogAnalyzer collector:

   To receive Microsoft 365 events, create a collector with the following parameters:

   - At the **Transport** step, specify the **office365** connector type.
   - At the **Event parsing** step, specify the **[OOTB] Microsoft Office 365 json** normalizer.

3. Installing a collector in the LogAnalyzer network infrastructure.

4. Verifying the receipt of Windows Microsoft 365 in the LogAnalyzer collector

   To verify that the Microsoft 365 event source server is configured correctly, you can search for related events.

Configuring receipt of the Kontinent encryption system events

The LogAnalyzer distribution kit includes the kuma-kont utility, which consists of an executable file and a configuration file. Supported operating systems include Astra Linux, Ubuntu 22.04 LTS, and RED OS 7.3.4 and 8. This utility allows lets you connect to the MSSQL database, receive events of the Kontinent encryption system, and send data to the LogAnalyzer collector in JSON format. Data import is supported for Kontinent 3.9.2.

Configuring the retrieval of Kontinent events from an MSSQL database and sending them to the LogAnalyzer collector involves the following steps:

1. Preparing to receive Kontinent events.

   At this step, you need to create a LogAnalyzer collector that will receive events, and also create a user in the MSSQL database.

2. Configuring the kuma-kont utility.

   At this step, you need to create the configuration file of the utility and the service of the utility.

3. Starting the kuma-kont utility.

   After activation, the utility starts exporting events from the MSSQL database and sending them to the LogAnalyzer collector. It keeps a log in the kuma-kont.log file.

Preparing to receive Kontinent events

1. Creating a LogAnalyzer collectorю

   1. Creating a LogAnalyzer collector for Kontinent 3.9 events.

      To receive audit events of the Kontinent 3.9 system using the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] SecurityCode Continent 3.9 json** normalizer. On the **Transport** tab, you also need to specify the port and protocol that are specified in the kuma-kont-config.yaml configuration file in the KUMA Collector group of settings.

   2. Installing a collector in the LogAnalyzer network infrastructure.

2. Creating a user account in MS SQL.

   Create an account in the database with the minimum necessary rights to connect and read data from the following tables: ALERTLOG, SERVERACCESSLOG, SYSTEMLOG, PACKETLOG, FILTERS.

3. Ensuring network connectivity.

   You must ensure network connectivity between the server on which the utility is running and the database server of the Kontinent system, as well as between the server on which the utility is running and the LogAnalyzer collector server.

Configuring the kuma-kont utility

Configuring the kuma-kont utility involves the following steps:

1. Creating the kuma-kont-config.yaml configuration file.
2. Creating the kuma-kont service.

Creating the kuma-kont-config.yaml configuration file

To create the kuma-kont-config.yaml configuration file:

1. Go to the LogAnalyzer installer directory by executing the following command:

   ```
   cd kuma-ansible-installer/roles/kuma/files
   ```

2. Extract the kuma-kont.tar archive:

   ```
   tar -xvf kuma-kont.tar
   ```

3. Go to the directory of the utility:

   ```
   cd kuma-kont
   ```

4. Copy the kuma-kont-config-template.yaml template to create a configuration file named kuma-kont-config.yaml:

   ```
   cp kuma-kont-config-template.yaml kuma-kont-config.yaml
   ```

5. Change read permissions for the configuration file:

   ```
   sudo chown <user name>:<group> kuma-kont-config.yaml
   sudo chmod 600 kuma-kont-config.yaml
   ```

   Replace <user name> with the operating system user name that the service will use.

6. Edit the settings in the kuma-kont-config.yaml configuration file.

   Specify the values of the following settings:

   - General settings: state_file, log_file_path.
   - Kontinent Database: name, user, password, host, port.
   - LogAnalyzer Collector group parameters: address, port, protocol.

     An example of configuration is given in the kuma-kont-config-template.yaml file.

7. Save the changes to the file.

The configuration file is created.

Creating the service

To create the kuma-kont service:

1. Change the execution permissions for the executable file:

   ```
   sudo chown <user name>:<group> <path to executable file>
   sudo chmod 700 <path to executable file>
   ```

2. Create the /etc/systemd/system/kuma-kont.service file with the following contents:

   ```
   [Unit]
   Description=Connector Kontinent 3.9
   After=network.target
   [Service]
   User=<user name>
   ExecStart=<path to executable file> --config <path to configuration file>
   Restart=always
   RestartSec=10
   [Install]
   WantedBy=multi-user.target
   ```

   Replace <user name> with the user name of the operating system user that the service will use.

3. Specify the path to the executable file.

   Specify the path to the kuma-kont executable file in the ExecStart field.

   For example, if the kuma-kont file is located in the /usr/bin/ directory, leave the default value.

4. Specify the path to the configuration file.

   Specify the path to the kuma-kont-config.yaml configuration file in the ExecStart field. For example, if the kuma-kont-config.yaml file is located in the /path/to/ directory, specify that path.

   Example kuma-kont.service file

   ```
   [Unit]
   Description=Continent Service
   After=network.target
   [Service]
   User=user-example
   ExecStart=./usr/bin/kuma-kont --config /path/to/kuma-kont-config.yaml
   Restart=always
   RestartSec=10
   [Install]
   WantedBy=multi-user.target
   ```

5. Load the systemd configuration:

   ```
   sudo systemctl daemon-reload
   ```

6. Make the service run at startup:

   ```
   sudo systemctl enable kuma-kont
   ```

The kuma-kont utility will be started as a service at system startup.

Starting and managing the kuma-kont service

To start the utility service, run the following command:

```
sudo systemctl start kuma-kont
```

The collection of events from the MSSQL database begins and the events are sent to the LogAnalyzer collector.

The utility service can be controlled using the following commands:

1. Start the kuma-kont service:

   ```
   sudo systemctl start kuma-kont
   ```

2. Stop the kuma-kont service:

   ```
   sudo systemctl stop kuma-kont
   ```

3. Restart the kuma-kont service:

   ```
   sudo systemctl restart kuma-kont
   ```

You can also send events using the kuma-kont utility running as an application. To start sending events to the LogAnalyzer collector, you need to run the following command while in the directory with the executable file:

```
./kuma-kont --config <path to the kuma-kont-config.yaml file>
```

Behavior when connection is lost

Connection to the MSSQL database.

If the connection to the MSSQL database is lost, the utility tries to reconnect to it every n seconds (where n is equal to the value of the poll_interval setting in the configuration file) until the connection is restored.

Connection to the LogAnalyzer collector

If the TCP protocol is used to send events, when the connection to the LogAnalyzer collector is lost, the utility also attempts to restore it. The interval between connection attempts increases until it reaches one hour, and subsequent reconnection attempts are made once per hour.

Known limitations

The event from the `AlertLog` table with AL_CATEGORY=5 is not fully decoded.

Settings in the kuma-kont-config.yaml configuration file

The table lists the settings that you can specify in the kuma-kont-config.yaml file.

Settings in the kuma-kont-config.yaml configuration file

Setting	Description
`debug`	Optional setting in the **General settings** group. Used when debug level log output is required. If false, only start and stop messages are included, as well as errors that occurred during the sending process. Available values: • **true** • **false** Default setting: **false**.
`need_reconnect`	An optional setting in the **General settings** group. This setting determines whether the utility reconnects to the database to make each request. Available values: • **true** • **false** Default setting: **false**.
`state_file`	Required setting in the **General settings** group. Absolute path to the JSON file in which the state of the processed logs is saved. The file is created automatically at this path. Make sure the user has sufficient permissions to write to this directory. Example: /home/user/kuma-kont/state.json
`log_file_path`	Required setting in the **General settings** group. Absolute path to the log file for recording connector activity. The file is created automatically at this path. Make sure the user has sufficient permissions to write to this directory. Example: /home/user/kuma-kont/kuma-kont.log
`poll_interval`	An optional setting in the **General settings** group. Database polling interval. The default value is 10 seconds. For high EPS, we recommend a value less than 15 seconds.
`name`	Required setting in the **Kontinent Database** group. Name of the database in which Kontinent logs are received.
`user`	Required setting in the **Kontinent Database** group. Name of the database user.
`password`	Required setting in the **Kontinent Database** group. Password of the database user.
`host`	Required setting in the **Kontinent Database** group. URL of the database server.
`port`	Required setting in the **Kontinent Database** group. Port of the database server. Make sure it is accessible from the server where the utility is installed.
`limit`	Optional setting in the **Kontinent Database** group. Limit on the number of events that can be downloaded from the database as part of one poll. By default, 100,000 events per database poll. If EPS is high, we recommend a value equal or greater than this.
`alert_log`	Optional setting in the **Last Id** group. Specifies the event of the AlertLog type from which you want to start sending events to the LogAnalyzer collector. Only works on first run or when the state file does not exit. The default value is 0.
`packet_log`	Optional setting in the **Last Id** group. Specifies the event of the PacketLog type from which you want to start sending events to the LogAnalyzer collector. Only works on first run or when the state file does not exit. The default value is 0.

Setting	Description
`server_access_log`	Optional setting in the **Last Id** group. Specifies the event of the ServerAccessLog type from which you want to start sending events to the LogAnalyzer collector. Only works on first run or when the state file does not exit. The default value is 0.
`system_log`	Optional setting in the **Last Id** group. Specifies the event of the SystemLog type from which you want to start sending events to the LogAnalyzer collector. Only works on first run or when the state file does not exit. The default value is 0.
`address`	Required setting in the **KUMA Collector** group. Fully qualified domain name (FQDN) or IP address of the server where the LogAnalyzer collector is installed.
`port`	Required setting in the **KUMA Collector** group. Port of the LogAnalyzer collector.
`protocol`	Required setting in the **KUMA Collector** group. Data transfer protocol used by the LogAnalyzer collector. If the udp protocol is used, events are not resent if connection to the LogAnalyzer collector is lost. Available values: • **tcp** • **udp**

Configuring receipt of VK WorkSpace Mail events

LogAnalyzer allows monitoring VK WorkSpace Mail 1.23 events received in syslog format.

Configuring event receiving consists of the following steps:

1. Configuring the export of VK WorkSpace Mail events
2. Creating a LogAnalyzer collector for VK WorkSpace Mail events.

   To receive VK WorkSpace Mail audit events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] VK WorkSpace Mail syslog** normalizer. Also, on the **Transport** tab, you need to specify the port and protocol in accordance with steps 2.2, 2.3 of the *Configuring the export of VK WorkSpace Mail events* instructions.
3. Installing a collector in the LogAnalyzer network infrastructure.
4. Verifying the receipt of VK WorkSpace Mail events in the LogAnalyzer collector.

   You can verify that the VK WorkSpace Mail event source server is correctly configured in the Searching for related events section of the LogAnalyzer web interface.

Configuring the export of VK WorkSpace Mail events

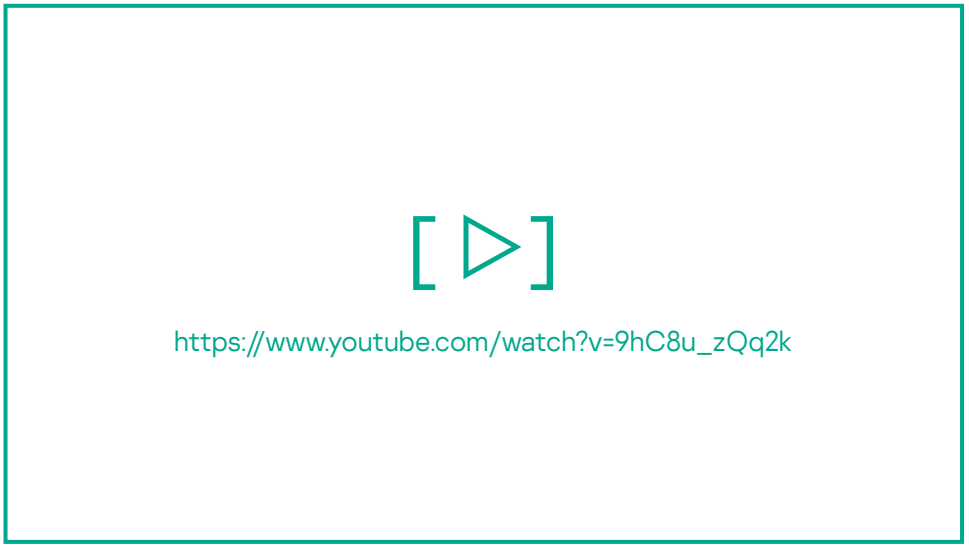
To configure the export of events to LogAnalyzer:

1. Go to the **Settings for duplicating user actions to external storage** menu.
2. Select the **Duplicate user actions to syslog** check box.
   a. In the **Syslog server address** field, enter the IP address or FQDN of the LogAnalyzer collector.
   b. In the **Syslog server port** field, specify the port that the LogAnalyzer collector is listening on.
   c. In **Syslog protocol, TCP or UDP** field, select the communication protocol.
   d. In the **Syslog data ID** field, specify bein.

The export of VK WorkSpace Mail events is configured.

Monitoring event sources

This section provides information about monitoring event sources.



Source status

In LogAnalyzer, you can monitor the state of the sources of data received by collectors. There can be multiple sources of events on one server, and data from multiple sources can be received by one collector.

You can configure automatic identification of event sources using one of the following sets of fields:

- Custom set of fields. You can specify from 1 to 9 fields in the order you want. TenantID does not need to be specified separately, it is determined automatically.

- **Apply default mapping** — DeviceProduct, DeviceHostName, DeviceAddress, DeviceProcessName. The field order cannot be changed.

  Sources are identified if the following fields in events are not empty: the DeviceProduct field, the DeviceAddress and/or DeviceHostname field, and the TenantID field (you do not need to specify the TenantID field, it is determined automatically). The DeviceProcessName field can be empty. If the DeviceProcessName field is not empty, and the other required fields are filled, a new source is identified.

  Identification of event sources depending on non-empty event fields

DeviceProduct	DeviceHostName	DeviceAddress	DeviceProcessName	TenantID (determined automatically)	
+	+			+	Source 1 identified
+		+		+	Source 2 identified
+	+	+		+	Source 3 identified
+	+		+	+	Source 4 identified
+		+	+	+	Source 5 identified
+	+	+	+	+	Source 6 identified
	+	+		+	Source not identified
	+		+	+	Source not identified
		+	+	+	Source not identified
+			+	+	Source not identified

Only one set of fields is applied for the entire installation. When upgrading to a new LogAnalyzer version, the default set of fields is applied. Only a user with the General Administrator role can configure the set of fields for identifying an event source. After you save changes to the set of fields, previously identified event sources are deleted from the LogAnalyzer web interface and from the database. If necessary, you can revert to using a set of fields to determine default event sources. For the edited settings to take effect and LogAnalyzer to begin identifying sources based on the new settings, you must restart the collectors.

To identify event sources:

1. In the LogAnalyzer web interface, go to the **Source status** section.

2. This opens the **Source status** window; in that window, click the wrench button.

3. This opens the **Settings of event source detection** window; in that window, in the **Grouping fields for determining the source** drop-down list, select the event fields by which you want to identify event sources.

   You can specify from 1 to 9 fields in the order you want. In a custom configuration, LogAnalyzer identifies sources in which the TenantID field is filled (you do not need to specify this field separately, it is determined automatically) and at least one field from the **Grouping fields for determining the source** is filled. For numeric fields, 0 is considered an empty value. If a single numeric field is selected for source identification, and the value of the numeric field is 0, the source is not detected.

   After you save the modified set of fields, an audit event is created and all previously identified sources are deleted from the LogAnalyzer web interface and from the database; assigned policies are disabled.

4. If you want to go back to the list of fields for identifying the default event source, click **Apply default mapping**. The default field order cannot be changed. If you manually specify the fields in the wrong order, an error is displayed and the save settings button becomes unavailable. The correct default sequence of fields is DeviceProduct, DeviceHostName, DeviceAddress, DeviceProcessName. Minimum configuration for identifying event sources using the default set of events: non-empty values in the DeviceProduct field, the DeviceAddress and/or DeviceHostName field, and the TenantID field (TenantID is determined automatically).

5. Click **Save**.

6. Restart the collectors to apply the changes and begin identifying event sources by the specified list of fields.

Source identification is configured.

To view events that are associated with an event source:

1. In the LogAnalyzer web interface, go to the **Source status** section.

2. This opens the **List of event sources** window; in that window, select your event source in the list, and in the **Name** column, expand the menu for the selected event source, click the **Events for <number> days** button.

   LogAnalyzer takes you to the **Events** section, where you can view a list of events for the selected source over the last 5 minutes. Values of fields configured in the event source identification settings are automatically specified in the query. If necessary, in the **Events** section, you can change the time period in the query and click **Run query** again to view the queried data for the specified time period.

Limitations

1. In a configuration with the default field set, LogAnalyzer registers the event source only if the raw event contains the DeviceProduct + DeviceAddress fields and the and/or DeviceHostName field.

   If the raw event does not contain the DeviceProduct field and the DeviceAddress and/or DeviceHostName fields, you can:

   - Configure enrichment in the normalizer: on the **Enrichment** tab of the normalizer, select the **Event** data type, specify the **Source field** setting, and for the **Target field**, select the DeviceProduct + DeviceAddress and/or DeviceHostName and click OK.

   - Use an enrichment rule: select the **Event** data source type, specify the **Source field** setting, and as the **Target field**, select DeviceProduct + DeviceAddress and/or DeviceHostName, then click **Create**. The created enrichment rule must be linked to the collector at the Event enrichment step.

   LogAnalyzer performs enrichment and registers the event source.

2. If LogAnalyzer receives events with identical values of the fields that identify the source, LogAnalyzer registers different sources if the following conditions are satisfied:

   - The values of the required fields are identical, but different tenants are determined for the events.

   - The values of the required fields are identical, but one of the events has an optional DeviceProcessName field specified.

   - The values of the required fields are identical, but the data in these fields have different character case.

If you want LogAnalyzer to log such events under the same source, you can further configure the fields in the normalizer.

Lists of sources are generated in collectors, merged in the LogAnalyzer Core, and displayed in the application web interface in the **Source status** section on the **List of event sources** tab. Data is updated every minute.

The rate and number of incoming events serve as an important indicator of the state of the observed system. You can configure monitoring policies such that changes are tracked automatically and notifications are automatically created when indicators reach specific boundary values. Monitoring policies are displayed in the LogAnalyzer web interface under **Source status** on the **Monitoring policies** tab.

When monitoring policies are triggered, monitoring events are created and include data about the source of events.

List of event sources

Event sources are displayed in the table in the **Source status** → **List of event sources** section. You can sort the table by clicking the column heading of the relevant parameter and selecting **Ascending** or **Descending**.

You can use the **Search** field to search for event sources. To search for a single character, enter that character and press **Enter**. You can also filter the table by the **Status** or **Monitoring policy** columns by clicking the heading of the relevant column and selecting the values that you want to display.

If necessary, you can configure the interval for updating data in the table. Available update periods: **1 minute**, **5 minutes**, **15 minutes**, **30 minutes**, **1 hour**. The default value is **No refresh**. You may need to configure the update period to track changes made to the list of sources.

Viewing information about event sources

In the **Source status** → **List of event sources** section, information about event sources is displayed in the following columns:

- **Status**—status of the event source:
  - Green—events are being received within the limits of the assigned monitoring policies.
  - Red—the frequency or number of incoming events go beyond the boundaries defined in at least one assigned monitoring policy.
  - Gray—monitoring policies have not been assigned to the source of events.

  If the status is red, an event of the Monitoring type generated. The monitoring event is generated in the tenant that owns the event source and is sent to the storage of the Main tenant (the storage must already be deployed in the Main tenant). If you have access to the tenant of the event source and do not have access to the Main tenant, you can still search for monitoring events in the storage of the Main tenant; the monitoring events of the tenants available to you will be displayed for you. You can also configure notifications to be sent to an arbitrary email address.

  The table can be filtered by status.

- **Name**—name of the event source. The name is generated automatically from the values of fields configured in the event source identification settings.

  You can rename an event source in the table of event sources by hovering over its name and clicking the pencil icon. The name can contain no more than 128 Unicode characters.

- **Host name or IP address**—name or IP address of the host from which the events originate if the DeviceHostName or DeviceAddress fields are specified in the event source identification settings.

- **Monitoring policy**—list of the monitoring policies assigned to the event source.

  If you want to filter the list of event sources by applied monitoring policies, click the name of this column and select one or more monitoring policies. If necessary, you can find policies in the list using the **Search** field. To find a policy by searching for a single character, enter that character and press **Enter**.

  You can view information about all monitoring policies assigned to an event source by clicking the row of the source. This opens a window that displays the settings of monitoring policies, as well as the status of the source according to each policy. If several monitoring policies are assigned to the source, the red status in the table of sources in this window lets you identify the policy that was triggered. You can also see which policies are enabled and which are disabled, and when the disabled policies will be enabled again.

- **Stream**—frequency at which events are received from the event source. If only monitoring policies of the **byCount** type or monitoring policies of different types are assigned to the source, this value is displayed as the number of events. If only monitoring policies of the **byEPS** type are assigned to the source, or no policies are assigned, the value is displayed as the number of events per second.

- **Tenant**—the tenant that owns the events received from the event source.

Managing event sources

You can select one or more event sources by selecting the check boxes in the first column of the table. You can select multiple event sources at once for performing group operations by selecting the check box in the heading of the first column and selecting **Select all** or **Select all in page**. The **Select all in page** option applies only to event sources displayed in the list: if only 500 out of 1500 sources are displayed in the list, then group actions to download, enable or disable policies, or delete event sources are applied only to the selected 500 sources. If you want to perform an action on all sources in the table, select **Select all**.

If you select sources of events, the following buttons become available:

- The **Enable policy** button enables the monitoring policy for event sources. You must select policies in the displayed window to apply them.

- You can use the **Disable policy** to disable the monitoring policy for event sources. When disabling a policy, you must specify if want to disable the policy temporarily or forever.

- The **Update policy** button applies the monitoring policies that are enabled for the event sources, or change the monitoring policies that are already assigned. When a policy is updated, a task is started in the task manager.

  This button becomes available after you change the monitoring policies assigned to event sources.

- You can click the **Remove** button to remove event sources from the table. The statistics on this source will also be removed. If a collector continues to receive data from the source, the event source will re-appear in the table but its old statistics will not be taken into account.

  If you want to delete all event sources, but some time has passed since the table was last refreshed, sources added during this time may not be displayed in the table, but they will be deleted regardless.

  If you delete more than 100,000 event sources to which a filter or search was applied, only the first 100,000 event sources will be deleted. You can select all filtered event sources again and delete them, and then repeat this until you have deleted all event sources that you intended to delete. You can delete over 100,000 event sources if no filters or searches are applied to them by selecting sources using the **Select all** button.

- You can click **CSV** to download the data of the selected event sources to a CSV file.

  The `Stream` field is downloaded only if a monitoring policy has been assigned to the event source; in that case, the unit of measurement taken from the policy is specified in the downloaded file. If no policy is assigned an empty `Stream` field is the expected behavior.

- You can click the **Build chart** button to plot a chart of incoming events for the last seven days for the selected event sources. You can select up to five event sources.
  .

Downloading event source information as a CSV file

You can download information about one or more event sources and the monitoring policies applied to them as a CSV file (comma-separated) in UTF-8 encoding. If multiple monitoring policies are applied to a source, in the file for that source, each monitoring policy and its parameters starts on a new line. For each monitoring policy applied to a source, the file contains the following parameters: **Status**, **Name**, **Monitoring policy**, **Lower limit**, **Upper limit**, **Stream**, **Tenant**. Identical fields for identifying sources are saved in the CSV file in the order specified by the user when configuring the identification of sources. When downloading data for multiple sources, the order of sources configured in the LogAnalyzer web interface is not preserved. If you want to sort the list of sources, you can do so in Excel or use grep with sorting.

To download event source information to a CSV file:

1. In the LogAnalyzer web console, in the **Source status** → **List of event sources** section, select one or more event sources in the table by selecting the check boxes in the first column next to the relevant sources.

   In the lower left part of the table, you can find the number of selected sources and the total number of sources in the table. You can select up to 150,000 event sources.

   You can select multiple event sources in the table by selecting the check box in the header of the first column of the table and selecting one of the following values:

   - **Select all** to select all event sources on all pages of the table. If you have used search to filter sources, this will select all sources that match the search query.

   - **Select all in page** to select all event sources on the currently displayed page. If you have used search to filter sources, this will select all sources on the currently displayed page that match the search query.

2. Click the **CSV** button in the upper part of the table.

   > Depending on the size of your browser window, the **CSV** button may be located in the ⋮ menu.

   An event source export task is created in the task manager.

3. Go to the **Task manager** section and find the created task.

   When the file is ready, the **Status** column of the task displays the **Completed** status.

4. Click the task type name and select **Download** from the drop-down list.

The CSV file with event source information is downloaded in accordance with your browser settings. The default file name is event-source-list_YYYYMMDD_HHMMSS.csv.

Viewing the dynamics of incoming events

You can examine the dynamics of events received from a source over the last seven days, taking into account the applied monitoring policies, in one of the following ways:

- View the graph for an individual event source.
- Plot a chart based on graphs for several (up to five) sources.

You can view the graph for a single event source in the LogAnalyzer web console in the **Source status** → **List of event sources** section by clicking the arrow icon › in the row of the relevant event source. The graph of incoming events is displayed under the row of the source.

The data in the graph is displayed as follows:

- The data is displayed for the days on which the events were received. The maximum period is seven days.

  In the upper left corner above the graph, the number of days is displayed, and in the upper right corner, the data display period is displayed. You can click the **Events for <number> days** button to go to the **Events** section and view the list of events for the selected source.

- The X-axis represents days, and the Y-axis represents the frequency of events (EPS).
- The lines represent the average, maximum, and minimum number of events for every 15-minute period during the last seven days.

  If you want to view the number of events at a specific time, hover over a point on the graph. A tooltip is displayed with the average, maximum, and minimum event count at a specific date and time.

You can also plot a chart of incoming events based on graphs for several event sources, for example, if you need to compare the activity of event sources of the same type that should behave in a similar way, but in fact behave in different ways.

To plot a chart based on graphs for multiple event sources:

1. In the LogAnalyzer web console, in the **Source status** → **List of event sources** section, select one or more event sources in the table by selecting the check boxes in the first column next to the relevant sources.

   You can plot a chart for up to 5 event sources at the same time.

2. Click the **Build chart** button in the upper part of the table.

   > Depending on the size of your browser window, the **Build chart** button may be found in the additional menu that you can open by clicking on the icon with the three dots ⋮.

   The displayed **Chart** pane contains a chart of incoming events for all selected sources as well as a table that displays the current number of events, the maximum number of events, and the average number of events for each source, calculated based on the data from the chart. You can compare how the data for the selected sources relates to each other over time.

   The data in the chart is displayed as follows:

   - The data is displayed for the days on which the events were received. The maximum period is seven days.

     In the upper right corner above the chart, the data display period is displayed.

   - The X-axis represents days, and the Y-axis represents the frequency of events (EPS).

   - The lines in the chart represent the average number of incoming events from the selected event sources for every 15-minute interval during the last seven days.

   You can hover over the chart to view the average number of events for each source at a specific time.

3. If necessary, clear the check boxes in the table below the chart next to the event sources that you want to hide in the chart.

4. If you want to display the diagram in more detail, click the two arrows ↙↗ icon to open the panel in full screen mode and zoom in on the diagram.

Monitoring policies

The rate and number of incoming events serve as an important indicator of the state of the system. For example, you can detect when there are too many events, too few, or none at all. Monitoring policies are designed to detect such situations. In a policy, you can specify a lower threshold, an optional upper threshold, and the way the events are counted: by frequency or by total number.

You can use the **Search** field to look for monitoring policies. The search will be carried out in the following columns: **Name**, **Tenant**, **Type**, **Schedule** (name of the day and time). To find a policy by searching for a single character, enter that character and press **`Enter`**.

The policy must be applied to the event source. You can apply one or more monitoring policies to a source. After applying the policy, you can monitor the status of the source on the **List of event sources** tab.

Policies for monitoring the sources of events are displayed in the table under **Source status** → **Monitoring policies**. You can sort the table by clicking the column header of the relevant setting. The maximum size of the policy list is not limited.

In the **Sources** column, you can click the **Show** button to view all event sources to which the policy is applied. When you click this button, you are taken to the **List of event sources** section, and the table of sources is filtered by the selected policy.

Algorithm of monitoring policies

Monitoring policies are applied to an event source in accordance with the following algorithm:

1. The event stream is counted at the collector.
2. The LogAnalyzer Core server gets information about the stream from the collectors every 15 seconds.
3. The obtained data is stored on the LogAnalyzer Core server in the Victoria Metrics time series database, and the data storage depth on the LogAnalyzer Core server is 15 days.
4. An inventory of event sources is taken once per minute.
5. The stream is counted separately for each event source in accordance with the following rules:
   - If a monitoring policy is applied to the event source, the displayed maximum number of events is calculated in accordance with the currently applied monitoring policies for the time interval specified in the policy.

     Depending on the policy type, the number of the event stream is counted as the number of events (for the **byCount** policy type or policies of different types) or as the number events per second (EPS, for the **byEPS** policy type). You can look up how the stream is counted for the applied policy in the **Stream** column on the List of event sources page.
   - If no monitoring policy is applied to the event source, the number for the event stream corresponds to the last value.
6. Once a minute, the application checks if any monitoring policies exist that must be applied to event sources or stopped according to the monitoring policy schedule.
7. Once a minute, the stream of events is checked for compliance with policy settings.

If the event stream from the source crosses the thresholds specified in the monitoring policy, information about this is recorded in the following way:

- A notification about a monitoring policy getting triggered is sent to the email addresses specified in the policy. For each policy, you can also configure a notification template.
- A stream monitoring informational event of type `5(Type=5)` is generated. The fields of the event are described in the table below.

Fields of the monitoring event

Event field name	Field value
ID	Unique ID of the event.
Timestamp	Event time.
Type	Type of the audit event. For the audit event, the value is `5` (monitoring).
Name	Name of the monitoring policy.
DeviceProduct	`KUMA`
DeviceCustomString1	The value from the `value` field in the notification. Displays the value of the metric for which the notification was sent.

The generated monitoring event is sent to all storages of the Main tenant.

Adding a monitoring policy

To add a new monitoring policy:

1. In the LogAnalyzer web interface, under **Source status** → **Monitoring policies**, click **Add policy** and configure the monitoring policy in the displayed window:

2. In the **Policy name** field, enter a name for the policy you are creating. The name must contain from 1 to 128 Unicode characters and must be unique within the selected tenant.

   We recommend choosing a name that reflects the configured schedule of the monitoring policy.

3. In the **Tenant** drop-down list, select the tenant that will own the policy. Your tenant selection determines the specific sources of events that can covered by the monitoring policy.

4. In the **Type** field, select one of the following monitoring policy types:

   - **by count**—by the number of events over a certain period of time.

   - **by EPS**—by the number of events per second (EPS) over a certain period of time. The average value over the entire period is calculated. You can additionally track spikes during specific periods.

5. In the **Count interval** field, specify the period during which the monitoring policy must take into account the data from the monitoring source. You can use the drop-down list on the right to select a value in minutes, hours, or days. The maximum value is 14 days.

6. If you selected the **by EPS** policy type, in the **Control interval, minutes** field, specify the control time interval (in minutes) within which the number of events must cross the threshold for the monitoring policy to trigger:

   - If, during this time period, all checks (performed once per minute) find that the stream is crossing the threshold, the monitoring policy is triggered.

   - If, during this time period, one of the checks (performed once per minute) finds that the stream is within the thresholds, the monitoring policy is not triggered, and the count of check results is reset.

   If you do not specify the frequency of measurement, the monitoring policy is triggered immediately after the stream is found to cross the threshold.

7. In the **Lower limit** and **Upper limit** fields, define the boundaries representing normal behavior. Deviations outside of these boundaries will trigger the monitoring policy, create an alert, and forward notifications.

   The **Lower limit** setting is required.

8. In the **Evaluation interval** field, specify the frequency with which the VMalert service will query VictoriaMetrics for policy data while the policy is being applied to the event source. You can use the drop-down list on the right to select a value in minutes, hours, or days. The default interval is 5 minutes.

   When specifying the evaluation interval, keep in mind the policy schedule. For example, if you configured the policy to be applied once every few hours, we do not recommend configuring a short interval and causing excessive load on VictoriaMetrics.

9. If necessary, in the **Send notifications** field, specify the email addresses to which notifications about the activation of the LogAnalyzer monitoring policy must be sent. To add an address, enter it in the field and press **Enter** or click **Add**. You can specify multiple email addresses.

   To forward notifications, you must configure a connection to the SMTP server.

10. In the **Notification template** drop-down list, select the template that you want to use for notifications. If necessary, click the **Create new** button to start creating a new notification template.

    By default, the basic notification template is selected. You can reset the template selection and switch to the base template by clicking the X icon.

11. In the **Schedule** section, configure how often you want to apply the monitoring policy to event sources. By default, the policy is applied every week, every day, from 00:00 to 23:59. To configure the monitoring policy schedule, do some of the following:

    - If you want to apply the monitoring policy weekly on specific days of the week:

      a. Enable the **Set up a schedule by days of the week** toggle switch.

      b. In the **Days of week** drop-down list, select the days of the week on which you want the policy to be applied to the source.

      If you want to clear the selection, click the X icon.

      c. In the **Time** field, specify the start and end time of the policy, with minute precision.

      The policy applicability interval is inclusive of its bounds; for example, if the end time is set to 23:59, the policy will be applied until 23:59:59.999. The default interval is 00:00 to 23:59. The start time must be earlier than the end time.

      d. If you want to add another period, click the **Add period** button and repeat steps 'b' and 'c'.

      You can add any number of periods.

    - If you want to apply the monitoring policy weekly on specific calendar dates:

      a. Enable the **Set up a schedule for specific days** toggle switch.

      b. Click the **Days of the month** field and use the calendar to select the dates on which you want to apply the policy to the source. You can select a period of several days or an individual day. The start date of the period must be earlier than the end date of the period.

      The dates are configured without a year value, so the policy will be applied annually on the specified days until you delete this period. If you want to clear the selection, click the X icon.

      c. In the **Time** field, specify the start and end time of the policy, with minute precision.

      The policy applicability interval is inclusive of its bounds; for example, if the end time is set to 23:59, the policy will be applied until 23:59:59.999. The default interval is 00:00 to 23:59. The start time must be earlier than the end time.

      d. If you want to add another period, click the **Add period** button and repeat steps 'b' and 'c'.

      You can add any number of periods.

    If you applied a schedule by day of the week and by day of the month at the same time, the day-of-the-month policy is applied first.

12. Click **Add**.

The monitoring policy will be added.

Editing monitoring policies

The **Source status** → **Monitoring policies** section displays the added monitoring policies and their settings that you specified when creating the policy. You can click a policy to display a sidebar with all of its settings. If necessary, you can edit the policy settings in this sidebar.

If a monitoring policy is applied to an event source, if you edit certain policy settings, you may need to update the policy to apply the changes. Every 30 minutes, LogAnalyzer checks if any monitoring policies require updating, and if that is the case, it automatically runs a task to update those monitoring policies. You can also run the update task manually by clicking the **Update policy** button at the top of the table. One task updates all policies that need updating.

The **Update policy** button becomes active only if some monitoring policies need updating. Information about whether the policy needs updating is displayed in the table of monitoring policies in the **Policy update status** as one of the following statuses:

- **Update required** if one of the following monitoring policy settings was edited, but the changes have not been applied to event sources:
  - **Policy name**
  - **Type**.
  - **Lower limit**
  - **Upper limit**
  - **Count interval**
  - **Control interval**
  - **Evaluation interval**
  - **Schedule**

    Schedule changes are applied within a minute, when LogAnalyzer checks for policies that need to be enabled or disabled.
- **Updated** in any of the following cases:
  - After editing the policy, the task to apply the policy was started, and the changes were applied to event sources.
  - You have edited one of the following policy settings, which does not require starting the update task:
    - **Send notifications**
    - **Notification template**

    In this case, the edited policy settings are applied to event sources after a minute. Changes of the **Notification template** setting are applied instantly.
  - The modified monitoring policy is not applied to event sources.

The date and time when the policy was last applied to event sources is displayed in the **Date the policy was last applied** column.

> While the policy update task is running, the **Update policy** button is unavailable for all users. If another user has edited the settings of the policy that necessitate an update, the **Update policy** button becomes active for you only after you refresh the page or edit the policy or an event source.

Applying monitoring policies

To apply monitoring policies to event sources:

1. In the LogAnalyzer web console, in the **Source status** → **List of event sources** section, select one or more event sources in the table by selecting the check boxes in the first column next to the relevant sources. You can select several event sources by clicking the check box in the heading of the first column and selecting one of the following options:

   - **Select all** to select all event sources on all pages of the table. If you have used search to filter sources, this will select all sources that match the search query.
   - **Select all in page** to select all event sources that are loaded on the currently displayed page. If you have used search to filter sources, this will select all sources on the currently displayed page that match the search query.

   In the lower left part of the table, you can find the number of selected sources and the total number of sources in the table.

   After you select the event sources to which you want to apply the monitoring policy, the **Enable policy** button becomes available on the toolbar.

2. Click **Apply policy**.

3. This opens the **Apply policy** window; in that window, select one or more monitoring policies that you want to apply to the selected event sources. The table lists only monitoring policies that you can assign to the selected sources: policies that belong to the same tenant or to the Shared tenant, if you have access to it. If no shared policies exist for the selected event sources and you do not have access to the Shared tenant, the policy table is empty.

   To select all available policies, you can select the check box in the heading of the first column. You can also use context search by policy name or sort the policies by clicking the heading of the column by which you want to sort the table and selecting **Ascending** or **Descending**.

   > Search or sorting is not available for the **Sources**, **Schedule**, **Policy update status**, **Date the policy was last applied** columns.

4. Click **Apply**.

5. In the table of sources, click **Update policy** to apply the changes to event sources.

The monitoring policies are applied to the selected event sources; the status of these sources changes to green. The names of the policies applied to the sources are displayed in the **Monitoring policy** column. A message is also displayed indicating the number of sources to which the policies have been applied. If the monitoring policy is triggered for an event source, the new status of that source is displayed after you manually refresh the page or it is refreshed automatically. We recommend configuring an automatic data refresh period to keep track of changes in the list of sources.

If you have selected more than 100,000 event sources and applied one or more policies to them, these policies are applied only to the first 100,000 sources to which these policies have not yet been applied. If you need to apply policies to the remaining sources, you can do one of the following:

- Select all sources again and apply the policies to them.
- Filter the table of sources by any parameter so that the table displays less than 100,000 sources, then apply the policies to them.

Repeat the action until the policies have been applied to all the sources that you need.

Disabling monitoring policies

To disable monitoring policies for event sources:

1. In the LogAnalyzer web console, in the **Source status → List of event sources** section, select one or more event sources in the table by selecting the check boxes in the first column next to the relevant sources.

   In the lower left part of the table, you can find the number of selected sources and the total number of sources in the table. After you select the event sources to which monitoring policies are applied in the list, the **Disable policy** button becomes available on the toolbar.

   You can select several event sources by clicking the check box in the heading of the first column and selecting one of the following options:

   - **Select all** to select all event sources on all pages of the table. If you have used search to filter sources, this will select all sources that match the search query.
   - **Select all in page** to select all event sources that are loaded on the currently displayed page. If you have used search to filter sources, this will select all sources on the currently displayed page that match the search query.

2. Click **Disable policy**.

3. This opens the **Disable policy** window; in that window, select one or more monitoring policies that you want to disable for the selected event sources. The table lists all monitoring policies applied to at least one of the selected event sources.

   To select all available policies, you can select the check box in the heading of the first column. You can also use context search or sort the policies by clicking the heading of the column by which you want to sort the table and selecting **Ascending** or **Descending**.

   Search or sorting is not available for the **Sources**, **Schedule**, **Policy update status**, **Date the policy was last applied** columns.

4. In the settings section above the policy table, do one of the following:

   - If you want to temporarily suspend the policies, select **For** and specify the time in minutes, hours, or days after which the selected policies will be reapplied to event sources. Maximum values:

     - For days: 30
     - For hours: 743
     - For minutes: 44579

   - If you want to permanently disable the selected policies for event sources, select **Until you enable the policy manually**.

   The default selection is **For**, and the value is set to **5 minutes**.

5. Click **Disable**.

6. In the table of sources, click **Update policy** to apply the changes to event sources.

   The monitoring policies are disabled for selected event sources or suspended for the specified time. The status of these sources in the table changes to gray. A message is displayed indicating the number of sources for which the policies have been disabled.

If you have selected more than 100,000 event sources and disabled one or more policies for them, these policies are disabled only for the first 100,000 sources to which these policies are applied. If you need to disable policies for the remaining sources, you can do one of the following:

- Select all sources again and disable the policies for them.
- Filter the table of sources by any parameter so that the table displays less than 100,000 sources, then disable the policies for them.

Repeat the action until the policies have been disabled for all the sources that you need.

Adding a new monitoring policy based on an existing policy

To create a new monitoring policy based on an existing policy:

1. In the LogAnalyzer web interface, in the **Source status** → **Monitoring policies** section, select the monitoring policy that you want to base the new policy on.

   If necessary, you can find monitoring policies in the list using the **Search** field. The search will be carried out in the following columns: **Name**, **Tenant**, **Type**, **Schedule** (name of the day and time). To find a policy by searching for a single character, enter that character and press **`Enter`**.

2. Click **Duplicate policy**.

3. This opens the **Add policy** window, in which you can edit policy settings.

   By default, `- copy` is appended to the name of the new policy. The rest of the settings are the same as in the policy that you are duplicating.

4. Click the **Add** button to create the new policy.

The monitoring policy is created based on an existing policy.

Deleting monitoring policies

To delete a monitoring policy:

1. In the LogAnalyzer web interface, in the **Source status** → **Monitoring policies** section, select one or more monitoring policies that you want to delete.

   If necessary, you can find monitoring policies in the list using the **Search** field. The search will be carried out in the following columns: **Name**, **Tenant**, **Type**, **Schedule** (name of the day and time). To find a policy by searching for a single character, enter that character and press **Enter**.

2. Click **Delete policy** and confirm the action.

The selected monitoring policies are deleted.

> You cannot remove predefined monitoring policies or policies that are assigned to data sources.

Integration with other solutions

In this section, you'll learn how to integrate LogAnalyzer with other solutions to enrich its functionality.

Integration with Kaspersky CyberTrace

Kaspersky CyberTrace (hereinafter CyberTrace) is a tool that integrates threat data streams with SIEM solutions. It provides users with instant access to analytics data, increasing their awareness of security decisions.

You can integrate CyberTrace with LogAnalyzer in one of the following ways:

- Integrate CyberTrace indicator search feature to enrich LogAnalyzer events with information from CyberTrace data streams.
- Integrate the entire CyberTrace web interface into LogAnalyzer to get full access to CyberTrace.

  > Integration with the CyberTrace web interface requires the CyberTrace TIP Enterprise license.

You can also configure retroscan and event enrichment using CyberTrace. This allows LogAnalyzer to get notifications for events in which threat indicators had not been initially detected, but were discovered later after an update of the feeds. This can improve the accuracy of threat detection based on previously logged events.

https://www.youtube.com/watch?v=KMOaROHuwkw

Integrating CyberTrace indicator search

To integrate CyberTrace indicator search:

1. Configure CyberTrace to receive and process LogAnalyzer requests.

   You can configure the integration with LogAnalyzer immediately after installing CyberTrace in the Quick Start Wizard or later in the CyberTrace web interface.

2. Create an event enrichment rule in LogAnalyzer.

   In the enrichment rule, you can specify which data from CyberTrace you want to enrich the event with. We recommend selecting cybertrace-http as the source kind.

3. Create a collector to receive events that you want to enrich with CyberTrace data.

4. Link the enrichment rule to the collector.

5. Save and create the service:

   - If you linked the rule to a new collector, click **Save and create**, copy the collector ID in the opened window and use the copied ID to install the collector on the server using the command line interface.

   - If you linked the rule to an existing collector, click **Save and restart services** to apply the settings.

   The configuration of the integration of CyberTrace indicator search is complete and LogAnalyzer events will be enriched with CyberTrace data.

Example of testing CyberTrace data enrichment.

By default, LogAnalyzer does not test the connection with CyberTrace.

If you want to test the integration with CyberTrace and make sure that event enrichment is working, you can follow the steps of the following example or adapt the example to your situation. The example shows an integration test, which performs enrichment and shows that the event contains the specified test URL.

To run the test:

1. Create a test enrichment rule with parameters listed in the table below.

Setting	Value
Name	Test CT enrichment
Tenant	Shared
Source kind	cybertrace-http
URL	<URL of the cybertrace server to which you want to send requests>:9999
Mapping	LogAnalyzer: RequestURL field CyberTrace indicator: url
Debug	Enabled

1. Create a test collector with the following parameters:

   At step **2 Transport**, specify the http connector.

   At step **3 Parsing**, specify the normalizer and select the json parsing method, set the mapping of the RequestUrl – RequestUrl fields.

   At step **6 Enrichment**, specify the 'Test CT enrichment' rule.

   At step **7 Routing**, specify the storage where events must be sent.

2. Click **Create and save service**.

   A complete command for installing the collector is displayed in the window.

3. Click **Copy** to copy the command to the clipboard and run the command on the command line. Wait for the command to complete, return to the LogAnalyzer web interface, and click **Save collector**.

   A test collector is created and the test enrichment rule is linked to the collector.

4. Use the command line interface to send a request to the collector, which will trigger an event, which will then be enriched with the test URL `http://fakess123bn.nu`. For example:

   ```
   curl --request POST \
     --url http://<ID of the host where the collector is installed>:<port of the collector>/input \
     --header 'Content-Type: application/json' \
     --data '{"RequestUrl":"http://fakess123bn.nu"}'
   ```

5. Go to the KUMA **Events** section and run the following query to filter event output and find the enriched event:

   ```
   SELECT * FROM `events` WHERE RequestUrl = 'http://fakess123bn.nu' ORDER BY Timestamp DESC LIMIT 250
   ```

   Result:

   Enrichment is successful, the event now has a **RequestURL** field with the `http://fakess123bn.nu` value, as well as a TI indicator and indicator category with CyberTrace data.

If the test did not result in enrichment, for example, if the TI indicator is missing, we recommend to do the following:

1. Check the settings of the collector and enrichment rules.
2. Download the collector logs using the following command and look for errors in the logs:

   ```
   tail -f /opt/kaspersky/kuma/collector/<collector ID>/log/collector
   ```

Configuring CyberTrace to receive and process requests

You can configure CyberTrace to receive and process requests from LogAnalyzer immediately after its installation in the Quick Start Wizard or later in the application web interface.

To configure CyberTrace to receive and process requests in the Quick Start Wizard:

1. Wait for the CyberTrace Quick Start Wizard to start after the application is installed.

   The wizard starts at step 1, **Welcome to Kaspersky CyberTrace**. You can go to the next step of the wizard by clicking **Next**.

2. At step 2, **Proxy settings**, if your organization uses a proxy server, enter its connection settings. If your organization does not use a proxy server, leave all fields blank.

3. At step 3, **Licensing settings**, select the method for adding a license key for CyberTrace: an activation code or a license key file. Depending on the selected method, specify the activation code or upload a license key file.

4. At step 4, **Service settings**, keep default settings.

5. At step 5, **Data management settings**:

   a. In the **SIEM system** drop-down list, select **KUMA**.

   b. Under **Listen on**, select the **IP and port** option.

   c. In the **IP address** field, enter `0.0.0.0`.

   d. In the **Port** field, enter the port to listen on for events. The default port is `9999`.

   e. Under **Send detection alerts**, in the **IP address** field, enter `127.0.0.1`, and in the **Port** field, enter `9998`.

   Leave the default values for everything else.

6. At step 6, **Certificate settings**, select **Commercial certificate** and add a certificate that allows you to download data feeds from update servers.

7. At step 7, **Feeds settings**, keep default settings.

CyberTrace is configured.

To configure CyberTrace to receive and process requests in the application web interface:

1. In the window of the CyberTrace web interface, switch **Data management** mode: in the left menu, select **System**, and then in the displayed menu, select **General**.
2. Select the **Settings** → **General** section.
3. Under **Listen on**:
   a. Select **IP and port**.
   b. In the **IP address** field, enter `0.0.0.0`.
   c. In the **Port** field, enter the port to listen on for events. The default port is `9999`.
4. Select the **Settings** → **Service alerts** section.
5. In the **Service alert format** field, enter `%Date% alert=%Alert%%RecordContext%`.
6. In the **Records context format** field, enter `|%ParamName%=%ParamValue%`.
7. Select the **Settings** → **Detection alerts** section.
8. In the **Alert format** field, enter `Category=%Category%|MatchedIndicator=%MatchedIndicator%%RecordContext%`.
9. On the **Context** tab, in the **Actionable fields** field, enter `%ParamName%:%ParamValue%`.
10. Switch to the **System management** mode: in the left menu, select **General**, then in the displayed menu, select **System**.
11. Select the **Settings** → **Service** section.
12. Under **Web interface**, in the **IP address or host name**, enter `127.0.0.1`.
13. In the upper toolbar, click **Restart service**.
14. Restart the CyberTrace server.

CyberTrace is configured.

Creating event Enrichment rules

To create event enrichment rules:

1. In the LogAnalyzer web interface, open **Resources** → **Enrichment rules**.
   The list of available enrichment rules will be displayed.
2. Click the **Add** button to create the new rule.

3. This opens a window; in that window, specify the enrichment rule settings:

   a. In the **Name** field, enter a unique name for the rule. The name must contain 1 to 128 Unicode characters.

   b. In the **Tenant** drop-down list, select the tenant that will own this resource.

   c. In the **Source kind** drop-down list, select **cybertrace-http**.

   d. Specify the **URL** of the CyberTrace server to which you want to connect. For example, *example.domain.com:443*.

   e. In the **Secret** field, select or create a secret for the connection.

   f. In the **Key fields** drop-down list, select the event fields used for enrichment of events with data from CyberTrace.

   g. If necessary, use the **Timeout** field to specify the maximum number of connections to the CyberTrace server that can be simultaneously established by LogAnalyzer. By default, this value is equal to the number of vCPUs of the LogAnalyzer Core server.

   h. In the **RPS** field, enter the number of requests to the CyberTrace server per second that LogAnalyzer can make. The default value is `1000`.

   i. In the **Timeout** field, specify the maximum number of seconds LogAnalyzer should wait for a response from the CyberTrace server. If a response is received before the timeout, it is added to the `TI` field of the event and the event processing continues.

   j. In the **Maximum number of events in enrichment queue** field, specify the maximum number of events stored in the enrichment queue for re-sending. The default setting is 1000000.

   k. Use the **Major version of Cybertrace** toggle switch to specify the CyberTrace version. Possible values: <5 and ≥ 5. If you select version ≥ 5, requests are executed in the General tenant in CyberTrace and use the new API version corresponding to CyberTrace 5.0 and higher; CyberTrace can also display statistics and retrospective analysis information for enrichment requests from LogAnalyzer that are made over HTTP. If you select <5, requests use the Kaspersky CyberTrace 4.x API. The default setting is <5.

   l. Use the **Debug** toggle switch to indicate whether or not to enable logging of service operations. Logging is disabled by default.

   m. In the **Tags** drop-down list, select the tags for the event enrichment rule.

   n. If necessary, in the **Description** field, add up to 4,000 Unicode characters describing the resource.

   o. Under **Filter parameters**, you can specify conditions to identify events that will be processed using the enrichment rule. You can select an existing filter from the **Filter** drop-down list or **Create** a new filter.

      Creating a filter in resources

To create a filter:

1. In the **Filter** drop-down list, select **Create new**.

2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.

3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

4. Under **Conditions**, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, fields of additional parameters for identifying the value to be passed to the filter may be displayed. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

   c. In the **Operator** drop-down list, select an operator.

Filter operators

- **=** means the left operand equals the right operand.
- **<** means the left operand is less than the right operand.
- **<=** means the left operand is less than or equal to the right operand.
- **>** means the left operand is greater than the right operand.
- **>=** means the left operand is greater than or equal to the right operand.
- **inSubnet** means the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains** means the left operand contains values of the right operand.
- **startsWith** means the left operand starts with one of the values of the right operand.
- **endsWith** means the left operand ends with one of the values of the right operand.
- **match** means the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit** checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
- **hasVulnerability** checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- The **inActiveList** operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary** checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory** means the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup** means the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- The **TIDetect** operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage.

- **inContextTable** checks for the presence of the entry in the specified context table.
- **intersect** checks for the presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. This check box does not apply to the **InSubnet**, **InActiveList**, **InCategory**, or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, from the **If** drop-down list, select **If not**.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

4. Click **Create**.

A new enrichment rule will be created.

CyberTrace indicator search integration is now configured. You can now add the created enrichment rule to a collector. You must restart LogAnalyzer collectors to apply the new settings.

If any of the CyberTrace fields in the events details area contains "[{" or "}]" values, it means that information from CyberTrace data feed was processed incorrectly and it's possible that some of the data is not displayed. You can get all information from a data feed by copying value of the **TI indicator** event field from LogAnalyzer and searching for it in CyberTrace in the **Indicators** section. All information about the found indicator is displayed on the **Indicator details** page.

Integrating CyberTrace interface

You can integrate the CyberTrace web interface into the LogAnalyzer web interface. When this integration is enabled, the LogAnalyzer web interface includes a **CyberTrace** section that provides access to the CyberTrace web interface. You can configure the integration in the **Integrations** → **Kaspersky CyberTrace** section of the LogAnalyzer web interface.

To integrate the CyberTrace web interface in LogAnalyzer:

1. In the LogAnalyzer web interface, open the **Resources** → **Resources configuration** → **Secrets**.

   The list of available secrets is displayed.

2. Click the **Add** button to create a new secret. This resource is used to store credentials of the CyberTrace server.

   This opens the incident creation window.

3. Enter information about the secret:

   a. In the **Name** field, choose a name for the added secret. The name must contain 1 to 128 Unicode characters.

   b. In the **Tenant** drop-down list, select the tenant that will own this resource.

   c. In the **Type** drop-down list, select **credentials**.

   d. In the **User** and **Password** fields, enter credentials for your CyberTrace server.

   e. If necessary, select the tags for the secret from the **Tags** drop-down list.

   f. If necessary, in the **Description** field, add up to 4,000 Unicode characters describing the resource.

4. Click **Create**.

   The CyberTrace server credentials are now saved and can be used in other LogAnalyzer resources.

5. In the LogAnalyzer web interface, open **Settings** → **Integrations** → **Kaspersky CyberTrace**.

   The window with CyberTrace integration parameters opens.

6. Manage the settings as follows:

   - Enable the **State** toggle switch if you want to enable the integration of the CyberTrace web interface into the LogAnalyzer web interface.
   - **Host** (required)—enter the address of the CyberTrace server.
   - **Port** (required)—enter the port of the CyberTrace server; the default port for managing the web interface is 443.
   - In the **Secret** drop-down list, select a previously created secret of the **credentials** type.
   - In the **Allow hosts** drop-down list, select the hosts that support the CORS technology.
   - In the **Major version of CyberTrace** field, select the CyberTrace version. Possible values: **<5**, **≥ 5.0**. If version **≥ 5.0** is selected, requests are executed in the General tenant in CyberTrace and use the new version of the API provided by Kaspersky CyberTrace 5.0 and later. If you select **<5**, requests use the Kaspersky CyberTrace 4.x API. The default value is **<5**.

7. You can configure access to the CyberTrace web interface in the following ways:

   - Use hostname or IP when logging into the LogAnalyzer web interface.

     To do this, in the Allow hosts section, click Add host and in the field that is displayed, enter the IP or hostname of the device

     on which the LogAnalyzer web interface is deployed.

   - Use the FQDN when logging into the LogAnalyzer web interface.

     If you are using the Mozilla Firefox browser to manage the application web interface, the CyberTrace section may fail to display data. In this case, configure the data display (see the instructions below).

8. Click **Save**.

CyberTrace is now integrated with LogAnalyzer, and the **CyberTrace** section is displayed in the LogAnalyzer web interface.

*To configure the data display in the **CyberTrace** section when using the FQDN to log in to LogAnalyzer in Mozilla Firefox:*

1. Clear your browser cache.
2. In the browser's address bar, enter the FQDN of the LogAnalyzer web interface with port number 7222 as follows: https://kuma.example.com:7222.

   A window will open to warn you of a potential security threat.
3. Click the **Details** button.
4. In the lower part of the window, click the **Accept risk and continue** button.

   An exclusion will be created for the URL of the LogAnalyzer web interface.
5. In the browser's address bar, enter the URL of the LogAnalyzer web interface with port number 7220.
6. Go to the **CyberTrace** section.

Data will be displayed in this section.

Updating CyberTrace deny list (Internal TI)

When the CyberTrace web interface is integrated into the LogAnalyzer web interface, you can update the CyberTrace denylist or **Internal TI** with information from LogAnalyzer events.

To update CyberTrace Internal TI:

1. Open the event details area from the events table and click the link on a domain, web address, IP address, or file hash.

   The context menu opens.
2. Select **Add to Internal TI of CyberTrace**.

   A confirmation window opens.
3. If you want to confirm your actions and update the **Internal TI** with data from LogAnalyzer events, click **Yes**.

The selected object is now added to the CyberTrace denylist.

Configuring event enrichment and retroscan

This functionality is only available in the commercial version of LogAnalyzer.

Integration of LogAnalyzer with Kaspersky CyberTrace (hereinafter referred to as CyberTrace) version 5.0 and later allows enriching events with information about indicators of compromise (IoC) and retrospectively scanning events (performing a retroscan). Retroscan helps detect indicators that had not been detected at the time of initial enrichment, but appeared later in updated CyberTrace feeds. Retrospective scanning can be used if you want to rescan saved events for new threats.

To set up data enrichment and retroscan:

1. Perform the initial configuration in LogAnalyzer by following these steps:

   a. Configure CyberTrace integration.

   b. Create two collectors to receive events that you want to enrich with CyberTrace data.

      - To get notifications from CyberTrace about the triggering of indicators, create a collector in which a connector of the **tcp** type is used as the transport, and the normalizer is **[OOTB] CyberTrace**.

      - To configure event enrichment, create a collector with the following parameters:

        - Select any connector (for example, **http**) as the transport.

        - You can choose any normalization method (for example, **json**) with the mapping of the following fields: `SourceHostName`, `DestinationHostName`, `Message`, `DeviceHostName`, `DeviceAddress`, `FileHash`, `OldFileHash`, `RequestUrl`.

        - As the enrichment, create an event enrichment rule in which the data source type is **cybertrace-http**, the CyberTrace version is **≥ 5.0**, and the following normalizer fields are selected in the **Key fields** drop-down list: `SourceHostName`, `DestinationHostName`, `Message`, `DeviceHostName`, `DeviceAddress`, `FileHash`, `OldFileHash`, `RequestUrl`.

2. To configure retroscan in the CyberTrace web interface, follow these steps:

   a. In the **Settings** → **General** section, enable **API lookup** to save detected indicators and collect statistics a public API request is used.

   b. On the **Settings** → **Retroscan** page, enable retroscan, then select the **Regular expressions** and enable the sources and regular expressions that you want to use for retroscan.

   c. In the **Settings** → **General** section, under **Detection alerts**, specify the IP address and port that CyberTrace will use to send alerts about the detection of indicators of compromise.

   d. In the **Settings** → **General** section, under **Service alerts**, configure CyberTrace service notifications (for example, notifications about the retroscan task finishing or license key replacement) by specifying the IP address of the collector and the port of the LogAnalyzer TCP connector.

   For more information on configuring retroscan, see the *Kaspersky CyberTrace Help*.

Event processing stages during retroscan

Retroscan allows detecting indicators of compromise that had not been known when an event was received, but became available later after an update of Kaspersky CyberTrace feeds.

The retroscan algorithm comprises the following steps:

1. **Sending an event to CyberTrace**

   An event containing indicators is received at the CyberTrace collector. When received, these indicators may not be present in active feeds.

   Example event:

   ```
   {
   "SourceHostName": "45.54.64.52",
   "DestinationHostName": "geardox.site",
   "Message": "5BA7335AD847E5ED6211FB27094B9855",
   "DeviceHostName": "test.domain.ru",
   "DeviceAddress": "5.8.16.77",
   "FileHash": "E1B85D995DC09DE25603D0B7A67D998015F6C045",
   "OldFileHash": "B276263D373A6B56EEAB09673429E1490AEFC5E14738CDF9E1661C36EBD80656",
   "RequestUrl": "https://kaspersky.com/"
   }
   ```

2. **Saving the event in the CyberTrace database**

   If integration over TCP is used, CyberTrace uses regular expressions to extract indicators from the event. If integrated via the REST API (using API lookup), indicators are passed directly in the body of the request. If at the moment when an indicator is processed, the feeds do not contain information that allows to classify it, CyberTrace saves the indicator to the retroscan database for subsequent analysis after feeds are updated.

3. **Running the retroscan task**

   The retroscan task can be scheduled or run manually. To run a retroscan manually, go to the **System** → **Retroscan** page and click **Start now**.

4. **Detecting indicators after a feed update**

   If, during the update process, the feeds deliver indicators that match previously saved events, CyberTrace detects such matches when running the retroscan task.

5. **Generating alerts based on retroscan results**

   After the retroscan is completed, CyberTrace generates alerts about the detection of indicators of compromise and sends these alerts to the LogAnalyzer collector.

   Example alert:

   ```
   Category=KUMA_Demo_1_Hash_SHA1|MatchedIndicator=E1B85D995DC09DE25603D0B7A67D998015F6C045|
   05:34|geo=bd, eg, in|last_seen=02.02.2021
   11:36|popularity=2|threat=HEUR:Trojan.MSIL.Taskun.gen
   ```

6. **Event processing in KUMA**

   The event is processed by the **[OOTB] CyberTrace** normalizer. The user can match the received notification with the original event for subsequent analysis.

7. **Getting a service alert about the completion of the task**

If CyberTrace is configured to send service alerts, CyberTrace sends an alert about the completion of the retroscan task.

Example alert:

```
Jun 05 16:08:33
alert=KL_ALERT_RetroScanCompleted|iocs_detected=168|iocs_rescanned=1511177|retroscan_repo
C2DF-4AED-B430-2E64AEF781FC
```

Integration with Kaspersky Threat Intelligence Portal

This functionality is only available in the commercial version of LogAnalyzer.

The Kaspersky Threat Intelligence Portal combines all of Kaspersky's knowledge about cyberthreats and how they're related into a single web service. When integrated with LogAnalyzer, the portal helps LogAnalyzer users to make faster and better-informed decisions, providing them with data about URLs, domains, IP addresses, including WHOIS and DNS data.

In the Kaspersky Threat Intelligence Portal Help, you can find information about data provision for the following objects:

- Hashes
- IP addresses
- Domains
- Web addresses

Access to the Kaspersky Threat Intelligence Portal is provided based on a fee. License certificates are created by Kaspersky staff. To obtain a certificate for Kaspersky Threat Intelligence Portal, contact your Kaspersky Technical Account Manager.

Initializing the Kaspersky Threat Intelligence Portal integration

This functionality is only available in the commercial version of LogAnalyzer.

To integrate Kaspersky Threat Intelligence Portal into LogAnalyzer:

1. In the LogAnalyzer web interface, open the **Resources** section.
2. Under **Resources configuration**, select **Secrets**.

   The list of available secrets will be displayed.
3. Click the **Add** button to create a new secret. This resource is used to store credentials of your Kaspersky Threat Intelligence Portal account.

   The secret window is displayed.

4. Enter information about the secret:

    a. In the **Name** field, choose a name for the added secret.

    b. In the **Tenant** drop-down list, select the tenant that will own the created resource.

    c. In the **Type** drop-down list, select **ktl**.

    d. In the **User** and **Password** fields, enter credentials for your Kaspersky Threat Intelligence Portal account.

5. Upload your Kaspersky Threat Intelligence Portal certificate key:

    a. Click the **Upload PFX** button and select the PFX file with your certificate.

    The name of the selected file appears to the right of the **Upload PFX** button.

    b. Enter the password to the PFX file in the **PFX password** field.

6. If necessary, select one or more tags for the secret from the **Tags** drop-down list.

7. If you want, enter a **Description** of the secret.

8. Click **Create**.

    The Kaspersky Threat Intelligence Portal account credentials are now saved and can be used in other LogAnalyzer resources.

9. Go to the **Settings** → **Integrations** → **Kaspersky Threat Lookup integration**.

10. Use the **Status** toggle switch to enable the integration. The integration is disabled by default.

11. In the **Secret** drop-down list, select the secret you created before.

    You can create a new secret by clicking the button with the plus sign. The created secret will be saved in the **Resources** → **Secrets** section.

12. If necessary, select a proxy server in the **Proxy** drop-down list.

13. If you want to accept the License Agreement for OSINT data, click the **License Agreement** link. This opens the **License Agreement** window; in that window, select **I accept the License Agreement** and click **Save**. LogAnalyzer will get OSINT data from the Kaspersky Threat Intelligence Portal after the integration is set up.

14. Click **Save**.

15. After you save the settings, log in to the web interface and accept the **Terms of Use**. Otherwise, an error will be returned in the API.

The integration process of Kaspersky Threat Intelligence Portal with LogAnalyzer is completed.

Once Kaspersky Threat Intelligence Portal and LogAnalyzer are integrated, you can request additional information from the event details area about hosts, domains, URLs, IP addresses, and file hashes (MD5, SHA1, SHA256).

Requesting information from Kaspersky Threat Intelligence Portal

This functionality is only available in the commercial version of LogAnalyzer.

To request information from Kaspersky Threat Intelligence Portal:

1. Open the event details area from the events table and click the link on a domain, web address, IP address, or file hash.

   The **Threat Lookup enrichment** area opens in the right part of the screen.

2. Select check boxes next to the data types you want to request.

   If neither check box is selected, all information types are requested.

3. In the **Maximum number of records in each data group** field enter the number of entries per selected information type you want to receive. The default value is `10`.

4. Click **Request**.

   A *ktl* task has been created. When it is completed, events are enriched with data from Kaspersky Threat Intelligence Portal which can be viewed from the events table.

Viewing information from Kaspersky Threat Intelligence Portal

This functionality is only available in the commercial version of LogAnalyzer.

To view information from Kaspersky Threat Intelligence Portal:

Open the event details area from the events table, and click the link on a domain, web address, IP address, or file hash for which you have requested information from Kaspersky Threat Intelligence Portal.

The event details area opens in the right part of the screen with data from Kaspersky Threat Intelligence Portal; the time when it was received is indicated at the bottom of the screen.

Information received from Kaspersky Threat Intelligence Portal is cached. Clicking a domain, web address, IP address, or file hash in the details pane of an event for which LogAnalyzer has information available displays the information from Kaspersky Threat Intelligence Portal with the time when it was received, instead of the **Threat Lookup enrichment window**. You can update the data.

Updating information from Kaspersky Threat Intelligence Portal

This functionality is only available in the commercial version of LogAnalyzer.

To update information, received from Kaspersky Threat Intelligence Portal:

1. Open the event details area from the events table, and click the link on a domain, web address, IP address, or file hash for which you have requested information from Kaspersky Threat Intelligence Portal.

2. Click **Update** in the event details area containing the data received from the Kaspersky Threat Intelligence Portal.

   The **Threat Lookup enrichment** area opens in the right part of the screen.

3. Select the check boxes next to the types of information you want to request.

   If neither check box is selected, all information types are requested.

4. In the **Maximum number of records in each data group** field enter the number of entries per selected information type you want to receive. The default value is `10`.

5. Click **Update**.

   The *KTL* task is created and the new data received from Kaspersky Threat Intelligence Portal is requested.

6. Close the **Threat Lookup enrichment** window and the details area with KTL information.

7. Open the event details area from the events table and click the link corresponding to the domain, URL, IP address, or file hash for which you updated Kaspersky Threat Intelligence Portal information, then select **Show info from Threat Lookup**.

The event details area opens on the right with data from Kaspersky Threat Intelligence Portal, indicating the time when it was received on the bottom of the screen.

Authentication using domain accounts

To enable users to perform authentication in the LogAnalyzer web interface using their own domain account credentials, perform the following configuration steps.

1. Enable domain authentication

   Domain authorization is disabled by default, and a domain connection is not configured.

2. Configure a connection to the domain controller.

   The following connections are available:

   - Active Directory (AD)
   - Active Directory Federation Services (ADFS)
   - FreeIPA

   The AD and ADFS connection settings can be configured at the same time.

   You can connect to one domain only.

3. **Add groups of user roles.**

   You can specify a domain group for each LogAnalyzer role. After performing authentication using their domain accounts, the users from this group get access to the LogAnalyzer web interface in accordance with the specified role.

The application checks whether the user's group matches the specified filter in the following order of precedence of roles in the LogAnalyzer web interface: Junior analyst → Tier 1 analyst → Tier 2 analyst → Tenant administrator → General administrator. Upon the first match, the application assigns a role to the user and does not check any further. If a user matches two groups in the same tenant, the role with the most privileges is used. If multiple groups are matched for different tenants, the user will be assigned the specified role in each tenant.

When you click on **Tenant**:

- Under the tenant, the entire list of roles and groups that belong to this tenant is expanded.
- Under the row that contains the name of the tenant, the total number of roles and the total number of groups belonging to this tenant, rows with the name of the tenant, the name of the role and the name of the group are displayed.
- Duplicate names of roles and groups are merged when copying and adding a filter (that is, if a **junior analyst role, group 1** belongs to a tenant, when trying to add or copy the same entry (**junior analyst role**, **group 1**) from another tenant, only one **junior analyst role**, **group 1** entry will be present in the table, without duplicates.
- Roles are sorted in descending order of seniority.

You can copy configured user roles and their corresponding groups to a different tenant. To do this, select the check box next to these and click the **Copy to clipboard** button. This opens a window with a list of available tenants. When copying user roles, you can:

- Select one or more roles within a tenant to copy settings.
- Copy an entire tenant with its configured roles and groups.
- Copy settings to multiple tenants at the same time, but only from one tenant. The Shared tenant and the tenant from which you are copying cannot be selected in the list of available tenants.

After their first authentication, all domain users gain access to the default space, which is specified in the Access to spaces section.

If necessary, you can delete multiple tenants or user roles with their groups at the same time. To do so, select check boxes next to the entities that you want to delete and click the **Delete** button. You can delete selected roles within a tenant or from different tenants.

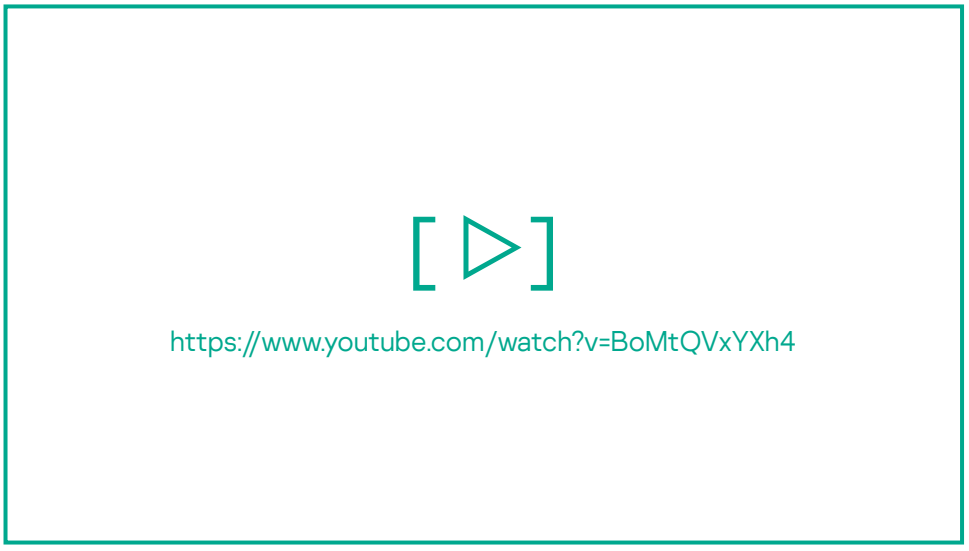
Special considerations for logging in after configuring domain authentication

For successful authentication, the following conditions must be met:

- **FreeIPA**: when logging into the system, the user must capitalize the domain name in the login. Example: user@FREEIPA.COM.
- **AD/ADFS**: when logging into the system, the user must specify UserPrincipalName in the login. Example: user@domain.ru.

If you complete all the configuration steps but the users are not able to use their domain accounts for authentication in the LogAnalyzer web interface, it is recommended to check the configuration for the following issues:

- An email address is not indicated in the properties of the user account in Active Directory. If this is the case, an error message is displayed during the user's first authentication attempt and a LogAnalyzer account is not created.
- There is an existing local LogAnalyzer account with the email address indicated in the domain account properties. If this is the case, the error message is displayed when the user attempts to perform authentication with the domain account.
- Domain authorization is disabled in the LogAnalyzer settings.
- An error occurred when entering the group of roles.
- The domain user name contains a space.



https://www.youtube.com/watch?v=BoMtQVxYXh4

Enabling and disabling domain authentication

Domain authentication is disabled by default. To enable it, enable the **State** toggle switch. If you want to temporarily suspend domain authentication after configuring a connection, you can disable it in the LogAnalyzer web interface without deleting the previously specified settings. If necessary, you can enable authentication again at any time.

To enable or disable domain authorization of users in the LogAnalyzer web interface:

1. In the application web interface, select **Settings** → **Access** → **Domain authorization**.
2. In the **Authorization type** drop-down list, select one of the options:
   - **AD/ADFS**
   - **FreeIPA**

3. Do one of the following:

   - If you want to enable domain authentication, enable the **State** toggle switch.
   - If you want to disable domain authentication, disable the **State** toggle switch.

4. Click the **Save** button.

The selected settings are saved and applied.

Configuring connection between KUMA and FreeIPA

You can connect only to one FreeIPA domain. To do so, you must configure a connection to the domain controller.

To configure a connection to a FreeIPA domain controller:

1. In the application web interface, select **Settings** → **Access** → **Domain authorization**.
2. In the **Authorization type** drop-down list, select **FreeIPA**.
3. Under **FreeIPA**, enable the **State** toggle switch.
4. In the **Base DN** field, enter the DistinguishedName of the root record to search for access groups in the FreeIPA catalog service. Record format: dc=example,dc=com.
5. In the **URL** field, indicate the address of the domain controller in the format `<hostname or IP address of server>:<port>`.

   In case of server availability issues, you can specify up to three servers with domain controllers by separating them with commas. All of the specified servers must reside in the same domain.

6. If you want to use TLS encryption for the connection with the domain controller, select one of the following options from the **TLS mode** drop-down list:

   - **startTLS**.

     When the startTLS method is used, first it establishes an unencrypted connection over port 389, then it sends an encryption request. If the STARTTLS command ends with an error, the connection is terminated.

     Make sure that port 389 is open. Otherwise, a connection with the domain controller will be impossible.

   - **LDAPS**.

     When using LDAPS, an encrypted connection is immediately established over port 636.

   - **insecure**.

   When using an encrypted connection, it is impossible to specify an IP address as a URL.

7. If TLS encryption is enabled, the **Secret** field becomes required and you must specify a secret of the 'certificate' type in that field. If you have previously uploaded a secret, select it from the **Secret drop-down list.** If necessary, click the + button to create a new secret of the 'certificate' type and select the secret from the drop-down list.

8. In the **Timeout in seconds** field, indicate the amount of time to wait for a response from the domain controller server. The default value is 0.

   If multiple addresses are specified in the **URL** field, LogAnalyzer waits for a response from the first server for the specified number of seconds. If no response is received during that time, the application attempts to connect to the next server. If none of the indicated servers responds during the specified amount of time, the connection will be terminated with an error.

9. In the **Custom integration secret** drop-down list, select a secret with the 'credentials' type.

   If you want to upload a new secret of the 'credentials' type, click the + button on the right of the **Custom integration secret** drop-down list. This opens the **Secret** window; in that window, in the **Name** field, enter the name of the secret that will be displayed in the list after it is saved. In the **User** field, specify the DistinguishedName in the following format: uid=admin,cn=users,cn=accounts,dc=ipa,dc=test. Enter the **Password** and click **Save**.

   The secret is uploaded and becomes available for selection in the **Custom integration secret** drop-down list.

10. If you want to configure domain authentication for a user with the LogAnalyzer general administrator role, use the **General administrators group** field to specify the DistinguishedName of the FreeIPA group containing the user. Additional roles for the General administrator are automatically activated in LogAnalyzer, therefore, you do not need to add them separately.

    > In the case when multiple groups are specified for a user in the same tenant, the role with the highest-level permissions is used, with additional roles, if additional roles are assigned.

    Filter input example: `CN=KUMA team,OU=Groups,OU=Clients,DC=test,DC=domain`.

11. Click the **Save** button.

A connection with the FreeIPA domain controller is now configured.

You can also check the connection for the previously entered domain controller connection settings.

To check the connection to the domain controller:

1. In the application web interface, select **Settings** → **Access** → **Domain authorization**.
2. In the **Authorization type** drop-down list, select **FreeIPA**.
3. In the **FreeIPA** settings block, select the relevant secret in the **User credentials** field.

   If necessary, you can create a new secret by clicking the + button or change the settings of an existing secret by clicking the ✎ button. If integration with FreeIPA is enabled, the secret selection is always reset when the page is loaded.

4. Click **Check**.

   After clicking the **Check** button, the system tests the connection with the domain and returns a notification with the test results. The system does not check if the users can log in or if the user group is configured correctly.

For domain authentication, add the groups for the LogAnalyzer user roles.

> You can specify the groups only for the roles that require the configuration of domain authentication. You can leave the rest of the fields empty.

To add groups of user roles:

1. In the application web interface, select **Settings** → **Access** → **Domain authorization**.
2. Under **Administration groups**, click **Add group**.
3. In the **Tenant** drop-down list, select the tenant of the users for whom you want to configure domain authentication.
4. In the **Roles** drop-down list, specify the roles for the user. You can select multiple roles. The following values are available:
   - **Tenant administrator**
   - **Tier 2 analyst**
   - **Tier 1 analyst**
   - **Junior analyst**
   - **Access to shared dashboards**. The role is only available when the Shared tenant is selected from the **Tenant** drop-down list. For the Shared tenant, this is the only role that can be added via the domain authentication settings.

   After you select the roles, a group filter field is displayed for each role. In the fields for each role, specify the full path to the domain group. The users of this domain group must have the capability to perform authentication with their domain accounts. Example of group specification: `CN=KUMA team`, `OU=Groups`, `OU=Clients`, `DC=test`, `DC=domain`.

   You can define a separate set of role filters for each tenant.

   If no filter is specified for a role, this means that conditions for creating an account through domain authentication are not specified for that role. Authentication with that role is impossible.

   After the first authentication under a domain account, domain user cards are created for users in the **Settings** → **Access** → **Users** section. For a domain user, the ability to change the main role (General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst) is blocked in the user card, while additional roles can be added or removed (Access to CII, Interaction with NCIRCC, Read shared resources), including management of additional role assignment to tenants. Roles assigned in the Domain authorization section and roles assigned in the user card supplement each other. For the General administrator, additional roles in LogAnalyzer are automatically activated, therefore you do not need to add them separately. If the General administrator role was assigned to a domain user, and the General administrator role was subsequently revoked, additional roles must be reassigned in the user card in the **Settings** → **Access** → **Users** section.

   > You can specify only one domain group for each role. If you want to specify multiple groups, you must repeat steps 2 to 4 for each group while specifying the same tenant.

5. If necessary, repeat steps 2–4 for each tenant for which you want to configure domain authentication with the following roles: Junior analyst, Tier 1 analyst, Tier 2 analyst, or Tenant administrator.
6. Click **Add**.

The groups of user roles will be added. The defined settings will be applied the next time the user logs in to the LogAnalyzer web interface.

After the first authentication of the user, information about this user is displayed in the **Settings** → **Access** → **Users** section. The **Login** and **Password** fields received from the domain cannot be edited. The user role will also be unavailable for editing. To edit a role, you will have to change the user role groups. Changes to a role are applied after the next authentication of the user. The user continues working under the current role until the current session expires.

If the user name or email address is changed in the domain account properties, these changes must be manually made in the LogAnalyzer account.

Configuring a connection between LogAnalyzer and Active Directory

You can connect only to one Active Directory domain. To do so, you must configure a connection to the domain controller.

To configure a connection to an Active Directory domain controller:

1. In the application web interface, select **Settings** → **Access** → **Domain authorization**.
2. In the **Authorization type** drop-down list, select **AD/ADFS**.
3. Under **Active Directory (AD)**, enable the **State** toggle switch.
4. In the **Base DN** field, enter the DistinguishedName of the root record to search for access groups in the Active Directory catalog service.
5. In the **URL** field, indicate the address of the domain controller in the format `<hostname or IP address of server>:<port>`.

   In case of server availability issues, you can specify multiple servers with domain controllers by separating them with commas. All of the specified servers must reside in the same domain.

6. If you want to use TLS encryption for the connection with the domain controller, select one of the following options from the **TLS mode** drop-down list:

   - **startTLS**.

     When the startTLS method is used, first it establishes an unencrypted connection over port 389, then it sends an encryption request. If the STARTTLS command ends with an error, the connection is terminated.

     Make sure that port 389 is open. Otherwise, a connection with the domain controller will be impossible.

   - **LDAPS**.

     When using LDAPS, an encrypted connection is immediately established over port 636.

   - **insecure**.

   When using an encrypted connection, it is impossible to specify an IP address as a URL.

7. If you enabled TLS encryption at the previous step, add a TLS certificate:

   - If you previously uploaded a certificate, select it from the **Secret** drop-down list.

     If no certificate was previously added, the drop-down list shows **No data**.

   - If you want to upload a new certificate, click the + button on the right of the **Secret** list. In the opened window, in the **Name** field, enter the name that will be displayed in the list of certificates after the certificate is added. Add the file containing the Active Directory certificate (X.509 certificate public keys in Base64 are supported) by clicking the **Upload certificate file** button. Click the **Save** button.

     The certificate will be uploaded and displayed in the **Secret** list.

8. In the **Timeout in seconds** field, indicate the amount of time to wait for a response from the domain controller server.

   If multiple addresses are specified in the **URL** field, LogAnalyzer waits for a response from the first server for the specified number of seconds. If no response is received during that time, the application attempts to connect to the next server. If none of the indicated servers responds during the specified amount of time, the connection will be terminated with an error.

9. To configure domain authentication for a user with the LogAnalyzer general administrator role, specify the DistinguishedName of the Active Directory group the user belongs to in the **General administrators group** field. Additional roles for the General administrator are automatically activated in LogAnalyzer, therefore you do not need to add them separately.

   > In the case when multiple groups are specified for a user in the same tenant, the role with the highest-level permissions is used, with additional roles, if additional roles are assigned.

   Filter input example: `CN=KUMA team,OU=Groups,OU=Clients,DC=test,DC=domain`.

10. Click the **Save** button.

A connection with the Active Directory domain controller is now configured.

You can also check the connection for the previously entered domain controller connection settings.

To check the connection to the domain controller:

1. In the application web interface, select **Settings** → **Access** → **Domain authorization**.

2. In the **Authorization type** drop-down list, select **AD/ADFS**.

3. Under **Test connection**, select the relevant secret in the **User credentials** field.

   If necessary, you can create a new secret by clicking the + button or change the settings of an existing secret by clicking the button.

   The following formats for specifying a user are available in the **User** field: UserPrincipalName and domain\user.

4. Click **Check**.

   After clicking the **Check** button, the system tests the connection with the domain and returns a notification with the test results. The system does not check if the users can log in or if the user group is configured correctly.

For domain authentication, add the groups for the LogAnalyzer user roles.

You can specify the groups only for the roles that require the configuration of domain authentication. You can leave the rest of the fields empty.

To add groups of user roles:

1. In the application web interface, select **Settings** → **Access** → **Domain authorization**.
2. Under **Administration groups**, click **Add group**.
3. In the **Tenant** drop-down list, select the tenant of the users for whom you want to configure domain authentication.
4. In the **Roles** drop-down list, specify the roles for the user. You can select multiple roles. The following values are available:
   - **Tenant administrator**
   - **Tier 2 analyst**
   - **Tier 1 analyst**
   - **Junior analyst**
   - **Access to shared dashboards**. The role is only available when the Shared tenant is selected from the **Tenant** drop-down list. For the Shared tenant, this is the only role that can be added via the domain authentication settings.

   After you select the roles, a group filter field is displayed for each role. In the fields for each role, specify the full path to the domain group. The users of this domain group must have the capability to perform authentication with their domain accounts. Example of group specification: `CN=KUMA team`, `OU=Groups`, `OU=Clients`, `DC=test`, `DC=domain`.

   You can define a separate set of role filters for each tenant.

   If no filter is specified for a role, this means that conditions for creating an account through domain authentication are not specified for that role. Authentication with that role is impossible.

   After the first authentication under a domain account, domain user cards are created for users in the **Settings** → **Access** → **Users** section. For a domain user, the ability to change the main role (General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst) is blocked in the user card, while additional roles can be added or removed (Access to CII, Interaction with NCIRCC, Read shared resources), including management of additional role assignment to tenants. Roles assigned in the Domain authorization section and roles assigned in the user card supplement each other. For the General administrator, additional roles in LogAnalyzer are automatically activated, therefore you do not need to add them separately. If the General administrator role was assigned to a domain user, and the General administrator role was subsequently revoked, additional roles must be reassigned in the user card in the **Settings** → **Access** → **Users** section.

   You can specify only one domain group for each role. If you want to specify multiple groups, you must repeat steps 2 to 4 for each group while specifying the same tenant.

5. If necessary, repeat steps 2–4 for each tenant for which you want to configure domain authentication with the following roles: Junior analyst, Tier 1 analyst, Tier 2 analyst, or Tenant administrator.

6. Click **Add**.

   The groups of user roles will be added. The defined settings will be applied the next time the user logs in to the LogAnalyzer web interface.

After the first authentication of the user, information about this user is displayed in the **Settings** → **Access** → **Users** section. The **Login** and **Password** fields received from the domain cannot be edited. The user role will also be unavailable for editing. To edit a role, you will have to change the user role groups. Changes to a role are applied after the next authentication of the user. The user continues working under the current role until the current session expires.

If the user name or email address is changed in the domain account properties, these changes must be manually made in the LogAnalyzer account.

Configuring a connection between LogAnalyzer and Active Directory Federation Services

To configure domain authentication in LogAnalyzer and ensure that users can log in to LogAnalyzer using their accounts without specifying a user name and password, first create a connection group and configure the rules in AD FS or make sure that the necessary connection groups and rules already exist.

After configuration, the **Sign in via ADFS** button appears on the LogAnalyzer login page.

The **Sign in via ADFS** button is hidden on the KUMA login page in the following conditions:

- The **FreeIPA** option is selected in the **Authorization type** drop-down list.
- The **AD/ADFS** option is selected in the **Authorization type** drop-down list and the settings for AD FS are not specified or the **State** toggle switch is enabled for AD FS settings.

You can connect only to one ADFS domain. To do so, you must configure a connection to the domain controller.

To configure a connection to an AD FS domain controller:

1. In the application web interface, select **Settings** → **Access** → **Domain authorization**.
2. In the **Authorization type** drop-down list, select **AD/ADFS**.
3. Under **Active Directory Federation Services (ADFS)**, enable the **State** toggle switch.
4. In the **Client ID** field, enter the KUMA ID from the **Client ID** field in AD FS.
5. In the **Relying party identifier** field, enter the KUMA ID from the **Relying party identifiers** field in the AD FS.
6. Enter the **Connect Metadata URI** from the **Connect Metadata URI** field. The URI consists of the host of the AD FS (https://adfs.example.com) and the endpoint configuration parameter (/adfs/.well-known/openid-configuration).

   For example, https://adfs.example.com/adfs/.well-known/openid-configuration.

7. Enter the **Redirect URL** from the **Redirect URL** field in the AD FS. The value of the **Redirect URL** field in the AD FS is defined when the Application group is configured. In the AD FS, you need to indicate the KUMA FQDN and the </sso-callback> substring. In KUMA, the URL must be indicated without the substring, for example: https://kuma-example:7220/

   In a distributed installation with LogAnalyzer Core in a Raft cluster, you must configure the balancer and then specify the FQDN of the balancer. For example, in the AD FS settings, you need to specify https://kuma-load-balancer.com:7220</sso-callback>, and in LogAnalyzer in the **Redirect URL** field, you need to specify https://kuma-load-balancer.com:7220

8. If you want to configure domain authentication for a user with the LogAnalyzer general administrator role, use the **General administrators group** field to specify the DistinguishedName of the Active Directory Federation Services group containing the user. Additional roles for the General administrator are automatically activated in LogAnalyzer, therefore, you do not need to add them separately.

   > In the case when multiple groups are specified for a user in the same tenant, the role with the highest-level rights is used, with additional rights, if additional roles are assigned.

   Filter input example: `CN=KUMA team,OU=Groups,OU=Clients,DC=test,DC=domain`.

9. Click the **Save** button.

A connection with the Active Directory Federation Services domain controller is now configured.

> If, when trying to log in to LogAnalyzer via ADFS, the user gets an `Access denied` pop-up message, click the **Reset certificate** button. A new certificate will be generated automatically.

For domain authentication, add the groups for the LogAnalyzer user roles.

> You can specify the groups only for the roles that require the configuration of domain authentication. You can leave the rest of the fields empty.

To add groups of user roles:

1. In the application web interface, select **Settings** → **Access** → **Domain authorization**.
2. Under **Administration groups**, click **Add group**.
3. In the **Tenant** drop-down list, select the tenant of the users for whom you want to configure domain authentication.

4. In the **Roles** drop-down list, specify the roles for the user. You can select multiple roles. The following values are available:

   - **Tenant administrator**
   - **Tier 2 analyst**
   - **Tier 1 analyst**
   - **Junior analyst**
   - **Access to shared dashboards**. The role is only available when the Shared tenant is selected from the **Tenant** drop-down list. For the Shared tenant, this is the only role that can be added via the domain authentication settings.

   After you select the roles, a group filter field is displayed for each role. In the fields for each role, specify the full path to the domain group. The users of this domain group must have the capability to perform authentication with their domain accounts. Example of group specification: `CN=KUMA team`, `OU=Groups`, `OU=Clients`, `DC=test`, `DC=domain`.

   You can define a separate set of role filters for each tenant.

   If no filter is specified for a role, this means that conditions for creating an account through domain authentication are not specified for that role. Authentication with that role is impossible.

   After the first authentication under a domain account, domain user cards are created for users in the **Settings** → **Access** → **Users** section. For a domain user, the ability to change the main role (General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst) is blocked in the user card, while additional roles can be added or removed (Access to CII, Interaction with NCIRCC, Read shared resources), including management of additional role assignment to tenants. Roles assigned in the Domain authorization section and roles assigned in the user card supplement each other. For the General administrator, additional roles in LogAnalyzer are automatically activated, therefore you do not need to add them separately. If the General administrator role was assigned to a domain user, and the General administrator role was subsequently revoked, additional roles must be reassigned in the user card in the **Settings** → **Access** → **Users** section.

   > You can specify only one domain group for each role. If you want to specify multiple groups, you must repeat steps 2 to 4 for each group while specifying the same tenant.

5. If necessary, repeat steps 2–4 for each tenant for which you want to configure domain authentication with the following roles: Junior analyst, Tier 1 analyst, Tier 2 analyst, or Tenant administrator.

6. Click **Add**.

   The groups of user roles will be added. The defined settings will be applied the next time the user logs in to the LogAnalyzer web interface.

After the first authentication of the user, information about this user is displayed in the **Settings** → **Access** → **Users** section. The **Login** and **Password** fields received from the domain cannot be edited. The user role will also be unavailable for editing. To edit a role, you will have to change the user role groups. Changes to a role are applied after the next authentication of the user. The user continues working under the current role until the current session expires.

If the user name or email address is changed in the domain account properties, these changes must be manually made in the LogAnalyzer account.

Configuring connection in Active Directory Federation Services

This section provides instructions on how to create a new connection group and configure rules for the created connection group in Active Directory Federation Services (ADFS).

The ADFS role must already be configured on the server.

Creating a new connection group

1. In **Server Manager**, in the **Tools** menu, select **ADFS Management**.

   In ADFS, select the **Application groups** section and in the **Actions** section click **Add Application Group**.

2. In the **Add Application Group Wizard** window that opens, in the **Welcome** section **Name** field, specify the name of the new connection group. Example: new-application-group.

   In the **Template** field, in the **Client-Server applications** group, select **Native application accessing a web API**.

   Click **Next** to proceed to the next step of creating and configuring a connection group.

3. In the **Native application** section that opens, the **Name** and
   Client Identifier

   fields are filled in automatically.

   Specify the value of the **Client Identifier** field in the **Client Identifier** field in KUMA, when configuring domain authentication.

   In the

   Redirect URI field, enter the URI for redirection from ADFS with the /sso-callback substring, and click **Add**. Example: https://adfs.example.com:7220/sso-callback

   Click **Next** to proceed to the next configuration step.

4. In the **Configure Web API** section that opens, in the Identifiers field, add the trusted party ID and click **Add**. It can be any arbitrary value. Example: test-demo

   Specify the value of the **Identifier** field in KUMA, in the **Relying party identifiers** field, when configuring domain authentication.

   Click **Next** to proceed to the next configuration step.

5. In the **Apply Access Control Policy** section that opens, select the **Permit everyone** policy value.

   Click **Next** to proceed to the next configuration step.

6. In the **Configure Application Permissions** section that opens, the **Client application** field is filled in automatically.

   In the **Permitted scopes** field, select the check box for the **allatclaims** and **openid** options.

   Click **Next** to proceed to the next configuration step.

7. In the **Summary** section that opens, check the settings.

   If the settings are correct and you are ready to add a group, click **Next**.

A new group is added. You can proceed to configure the rules for the created group.

Adding rules for a connection group

1. In **Server Manager**, in the **Tools** menu, select **ADFS Management**.

   In ADFS, select the **Application groups** section and select the required connection group from the list. Example: new-application-group.

2. In the **Application groups** window, in the **Actions** section, click **Properties**.

   In the **new-application-group Properties** window that opens, in the **Applications** section, double-click **new-application-group - Web API**.

   In the **new-application-group - Web API Properties** window that opens, open the

   Issuance Transform Rules

   tab and click **Add rule**.

   In the **Add Transform Claim Rule Wizard** window that opens, in the **Choose Rule Type** section, select **Send LDAP Attributes as Claims** from the drop-down list.

   Click **Next** to proceed to the next configuration step.

3. In the **Configure Claim Rule** section, specify the rule name in the **Claim rule name** field. Example: rule-name-01.

   In the **Attribute store** drop-down list, select **Active directory**.

   In the **Mapping of LDAP attributes to outgoing claim types** field, map the following fields:

LDAP Attribute	Outgoing Claim Type
`User-Principal-Name`	`UserPrincipalName`
`Display-Name`	`displayName`
`E-Mail-Addresses`	`Mail`
`Is-Member-Of-DL`	`MemberOf`

   Click **Finish** to complete the configuration.

4. Go to the **new-application-group - Web API Properties** window, open the
   Issuance Transform Rules

   tab and click **Add rule**. In the **Add Transform Claim Rule Wizard** window that opens, in the **Choose Rule Type** section, select **Send claims using a custom rule** from the drop-down list.

   Click **Finish** to continue the configuration.

5. In the **Configure Claim Rule** section, specify the rule name in the **Claims rule name** field. Example: rule-name-02.

   In the **Custom rule** field, specify the following settings:

   ```
   c:[Type ==
   "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer ==
   "AD AUTHORITY"]
   => issue(store = "Active Directory", types = ("ObjectGUID"), query = ";ObjectGUID;
   {0}", param = c.Value);
   ```

   Click **Finish** to complete the configuration.

6. The system proceeds to the **new-application-group - Web API Properties** window and the **Issuance Transform Rules** tab.

   To apply the rules, on the **Issuance Transform Rules** tab that opens, click **Apply** or **OK**.

The configuration of groups and rules in ADFS is completed. You can proceed to configure domain authentication in KUMA.

Troubleshooting the Access denied error

When you try to log in to KUMA using ADFS, the `Access denied` or `Insufficient rights` pop-up message may appear. The KUMA Core log shows the `Data source certificate has been changed` error.

This error indicates that the ADFS certificate has been changed. To fix the error and resume domain authentication, in domain controller connection settings, click the **Reset certificate** button. A new certificate will be generated automatically.

Integration with NeuroDAT SIEM IM

NeuroDAT SIEM IM is an information security monitoring system.

You can configure the export of LogAnalyzer events to NeuroDAT SIEM IM. Based on incoming events, NeuroDAT SIEM IM automatically generates information security incidents.

To configure integration with NeuroDAT SIEM IM:

1. Connect to the NeuroDAT SIEM IM server over SSH using an account with administrative privileges.
2. Create a backup copy of the /opt/apache-tomcat-<server version>/conf/neurodat/soz_settings.properties configuration file.
3. In the /opt/apache-tomcat-<server version>/conf/neurodat/soz_settings.properties configuration file, edit the following settings as follows:
   - `kuma.on=true`

     This setting is an attribute of NeuroDAT SIEM IM interaction with LogAnalyzer.
   - `job_kuma=com.cbi.soz.server.utils.scheduler.KumaIncidentsJob`
   - `jobDelay_kuma=5000`
   - `jobPeriod_kuma=60000`
4. Save changes of the configuration file.
5. Run the following command to restart the tomcat service:

   ```
   sudo systemctl restart tomcat
   ```

6. Obtain a token for the user in LogAnalyzer. To do so:

   a. Open the LogAnalyzer web interface, click the name of your user account in the bottom-left corner of the window and click the **Profile** button in the opened menu.

      The **User** window with your user account parameters opens.

   b. Click the **Generate token** button.

      The **New token** window opens.

   c. If necessary, set the token expiration date:

      - Select the **No expiration date** check box.
      - In the **Expiration date** field, use the calendar to specify the date and time when the created token will expire.

   d. Click the **Generate token** button.

      The **Token** field with an automatically generated token is displayed in the user details area. Copy it.

      When the window is closed, the token is no longer displayed. If you did not copy the token before closing the window, you will have to generate a new token.

   e. Click **Save**.

7. Log in to NeuroDAT SIEM IM using the 'admin' account or another account that has the Administrator role for the organization you are configuring or the Administrator role for all organizations.

8. In the **Administration** → **Organization structure** menu item, select or create an organization that you want to receive incidents from LogAnalyzer.

9. On the organization form, do the following:

   a. Select the **Configure integration with KUMA** check box.

   b. In the **LogAnalyzer IP address and port** field, specify the LogAnalyzer API address, for example, `https://192.168.58.27:7223/api/v1/`.

   c. In the **KUMA API key** field, specify the user token obtained at step 6.

   d. Save the organization information.

Integration with LogAnalyzer is configured.

NeuroDAT SIEM IM tests access to LogAnalyzer and, if successful, displays a message about being ready to receive data from LogAnalyzer.

Importing asset information from RedCheck

This integration is an example and may require additional configuration depending on the versions used and the specifics of the infrastructure.
Compatibility is confirmed only for LogAnalyzer 2.0 or later and RedCheck 2.6.8 or later.
The terms and conditions of premium technical support do not apply to this integration; support requests are processed without a guaranteed response time.

RedCheck is a system for monitoring and managing the information security of an organization.

You can import asset information from RedCheck network device scan reports into LogAnalyzer.

Import is available from simple "Vulnerabilities" and "Inventory" reports in CSV format, grouped by hosts.

Imported assets are displayed in the LogAnalyzer web interface in the **Assets** section. If necessary, you can edit the settings of assets.

Data is imported through the API using the redcheck-tool.py utility. The utility requires Python 3.6 or later and the following libraries:

- csv
- re
- json
- requests
- argparse
- sys

To import asset information from a RedCheck report:

1. Generate a network asset scan report in RedCheck in CSV format and copy the report file to the server where the script is located.

   For more details about scan tasks and output file formats, refer to the RedCheck documentation.

2. Create a file with the token for accessing the LogAnalyzer REST API.

   The account for which the token is created must satisfy the following requirements:

   - Tenant administrator or Tier 2 analyst role.
   - Access to the tenant into which the assets will be imported.
   - Rights to use API requests: GET /assets, GET /tenants, POST/assets/import.

3. Download the script:

   https://box.kaspersky.com/d/2dfd1d677c7547a7ac1e/

4. Copy the redcheck-tool.py tool to the server hosting the LogAnalyzer Core and make the tool file executable:

   ```
   chmod +x <path to the redcheck-tool.py file>
   ```

5. Run the redcheck-tool.py utility:

   ```
   python3 redcheck-tool.py --kuma-rest <address and port of the KUMA REST API server> --token <API token> --tenant <name of the tenant in which the assets must be placed> --vuln-report <full path to the "Vulnerabilities" report file> --inventory-report <full path to the "Inventory" report file>
   ```

   Example:

   ```
   python3 --kuma-rest example.kuma.com:7223 --token 949fc03d97bad5d04b6e231c68be54fb --tenant Main --vuln-report /home/user/vuln.csv --inventory-report /home/user/inventory.csv
   ```

   You can use additional flags and commands for import operations. For example, the `-v` command displays an extended report on the received assets. A detailed description of the available flags and commands is provided in the "Flags and commands of redcheck-tool.py" table. You can also use the `--help` command to view information on the available flags and commands.

The asset information is imported from the RedCheck report to LogAnalyzer. The console displays information on the number of new and updated assets.

Example:

```
inventory has been imported for 2 host(s)
software has been imported for 5 host(s)
vulnerabilities has been imported for 4 host(s)
```

Example of extended import information:

```
[inventory import] Host: localhost Code: 200 Response: {'insertedIDs': {'0': '52ca11c6-a0e6-4dfd-8ef9-bf58189340f8'}, 'updatedCount': 0, 'errors': []}
[inventory import] Host: 10.0.0.2 Code: 200 Response: {'insertedIDs': {'0': '1583e552-5137-4164-92e0-01e60fb6edb0'}, 'updatedCount': 0, 'errors': []}
[software import][error] Host: localhost Skipped asset with FQDN localhost or IP 127.0.0.1
[software import] Host: 10.0.0.2 Code: 200 Response: {'insertedIDs': {}, 'updatedCount': 1, 'errors': []}
[vulnerabilities import] Host: 10.0.0.2 Code: 200 Response: {'insertedIDs': {}, 'updatedCount': 1, 'errors': []}
[vulnerabilities import] Host: 10.0.0.1 Code: 200 Response: {'insertedIDs': {'0': '0628f683-c20c-4107-abf3-d837b3dbbf01'}, 'updatedCount': 0, 'errors': []}
[vulnerabilities import] Host: localhost Code: 200 Response: {'insertedIDs': {}, 'updatedCount': 1, 'errors': []}
[vulnerabilities import] Host: 10.0.0.3 Code: 200 Response: {'insertedIDs': {'0': 'ed01e0a8-dcb0-4609-ab2b-91e50092555d'}, 'updatedCount': 0, 'errors': []}
inventory has been imported for 2 host(s)
software has been imported for 1 host(s)
vulnerabilities has been imported for 4 host(s)
```

The utility works as follows when importing assets:

- LogAnalyzer overwrites the data of assets imported through the API, and deletes information about their resolved vulnerabilities.
- LogAnalyzer skips assets with invalid data.

Flags and commands of redcheck-tool.py

Flags and commands	Mandatory	Description
`--kuma-rest <address and port of the LogAnalyzer server>`	Yes	Port 7223 is used for API requests by default. You can change the port if necessary.
`--token <token>`	Yes	The value of the option must contain only the token. The Tenant administrator or Tier 2 analyst role must be assigned to the user account for which the API token is being generated.
`--tenant <tenant name>`	Yes	Name of the LogAnalyzer tenant in which the assets from the RedCheck report will be imported.
`--vuln-report <full path to the "Vulnerabilities" report>`	Yes	"Vulnerabilities" report file in CSV format.
`--inventory-report <full path to the "Inventory" report file>`	No	"Inventory" report file in CSV format.
`-v`	No	Display extended information about the import of assets.

Possible errors

Error message	Description
Tenant %w not found	The tenant name was not found.
Tenant search error: Unexpected status Code: %d	An unexpected HTTP response code was received while searching for the tenant.
Asset search error: Unexpected status Code: %d	An unexpected HTTP response code was received while searching for an asset.
[%w import][error] Host: %w Skipped asset with FQDNlocalhost or IP 127.0.0.1	When importing inventory/vulnerabilities information, host with fqdn=localhost or ip=127.0.0.1 was skipped.

Configuring receipt of Sendmail events

You can configure the receipt of Sendmail mail agent events in the LogAnalyzer SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring Sendmail logging.
2. Configuring the event source server.
3. Creating a LogAnalyzer collector.

   To receive Sendmail events, use the following values in the Collector Installation Wizard:

   - At the **Event parsing** step, select the **[OOTB] Sendmail syslog** normalizer.
   - At the **Transport** step, select the **tcp** or **udp** connector type.

4. Installing the LogAnalyzer collector.

5. Verifying the receipt of Sendmail events in the LogAnalyzer collector

   You can verify that the Sendmail event source server is correctly configured in the Searching for related events section of the KUMA web interface.

Configuring Sendmail logging

By default, events of the Sendmail system are logged to syslog.

To make sure that logging is configured correctly:

1. Connect via SSH to the server on which the Sendmail system is installed.

2. Run the following command:

   ```
   cat /etc/rsyslog.d/50-default.conf
   ```

   The command should return the following string:

   ```
   mail.* -/var/log/mail.log
   ```

If logging is configured correctly, you can proceed to configuring the export of Sendmail events.

Configuring export of Sendmail events

Events are sent from the Sendmail mail agent server to the LogAnalyzer collector using the rsyslog service.

To configure transmission of Sendmail events to the collector:

1. Connect to the server where Sendmail is installed using an account with administrator rights.

2. In the /etc/rsyslog.d/ directory, create the Sendmail-to-siem.conf file and add the following line to it:

   ```
   If $programname contains 'sendmail' then @<<IP address of the collector>:<port of the collector>>
   ```

   Example:

   ```
   If $programname contains 'sendmail' then @192.168.1.5:1514
   ```

   If you want to send events via TCP, the contents of the file must be as follows:

   ```
   If $programname contains 'sendmail' then @@<<IP address of the collector>:<port of the collector>>
   ```

3. Create a backup copy of the /etc/rsyslog.conf file.

4. Add the following lines to the /etc/rsyslog.conf configuration file:

   ```
   $IncludeConfig /etc/Sendmail-to-siem.conf
   $RepeatedMsgReduction off
   ```

5. Save your changes.

6. Restart the rsyslog service by executing the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Managing LogAnalyzer

This section describes the general settings of LogAnalyzer.

Logging in to the application web interface

Only users with an *Active* user account status can log in to the LogAnalyzer web interface.

We recommend configuring password requirements and rules for logging in to the application web interface in the security policy.

To log in to the application web interface:

1. Enter the following address in your browser:

   `https://<IP address or FQDN of LogAnalyzer Core server>:7220`

   The web interface authorization page will open and prompt you to enter your login and password.

2. Enter the login of your account in the **Login** field.

3. Enter the password for the specified account in the **Password** field.

4. Click the **Login** button.

   If you enter incorrect credentials, LogAnalyzer makes you wait for 3 seconds before you can try to log in again. This delay helps prevent an attacker from brute-forcing a password to log in to the LogAnalyzer web interface. If you enter incorrect credentials several times in a row, your user account is blocked. The maximum number of unsuccessful login attempts and the lockout duration are governed by the security policy.

The main window of the application web interface is displayed.

In multitenancy mode, a user logging in to the application web interface for the first can see data only for those tenants that were selected for the user account when it was created.

To log out of the application web interface:

open the LogAnalyzer web interface, click your user account name in the bottom-left corner of the window, and click the **Logout** button in the opened menu.

Service monitoring

You can use the following functionality to monitor the status of all services except cold storage and agent:

- Viewing Victoria Metrics alerts

  Users with the General administrator role can configure thresholds for LogAnalyzer services, and if a specified threshold is exceeded, the following changes take place:

  - LogAnalyzer logs an audit event: VictoriaMetrics alert registered for service.
  - LogAnalyzer sends a notification email message to the General administrator.
  - Services are displayed in the **Active services** section with a yellow status. If you hover over the status icon, the error message will be displayed.

    Possible service statuses

    - Green means the service is running and accessible from the Core server.
    - Red means the service is not running or is not accessible from the Core server.
    - Yellow is the status that applies to all services except the agent. The yellow status means that the service is running, but there are errors in the service log, or there are alerts for the service from Victoria Metrics. You can view the error message by hovering the mouse cursor over the status of the service in the **Active services** section.
    - Purple is the status that is applied to running services whose configuration file in the database has changed, but that have no other errors. If a service has an incorrect configuration file and has errors, for example, from Victoria Metrics, status of the service is yellow.
    - Gray means that if a deleted tenant had a running service that is still running, that service is displayed with a gray status on the Active services page. Services with the gray status are kept when you delete the tenant to let you copy the ID and remove services on your servers. Only the General administrator can delete services with the gray status. When a tenant is deleted, the services of that tenant are assigned to the Main tenant.

- View VictoriaMetrics metrics if the user has a role with metrics access rights.

The following examples show how to monitor service status.

1. If the collector service has a yellow status in the **Active services** section and you see the **Enrichment errors increasing** message, you can:

   - Go to **Metrics** → <service type> → <service name> → **Enrichment** → **Errors** section of LogAnalyzer for the service with the yellow status, find out which enrichment is causing errors, and view the chart to find out when the problem started and how it evolved.

   - The likely cause of the enrichment errors may be DNS server unavailability or CyberTrace enrichment errors, therefore you can check your DNS or CyberTrace connection settings.

2. If the collector service has a yellow status in the **Active services** section and you see the **Output Event Loss increasing** message, you can:

   - Go to the **Metrics** → <service type> → <service name> → **IO** → **Output Event Loss** section of LogAnalyzer for the service with the yellow status and view the chart to find out when the problem started and how it evolved.

   - The likely cause of the enrichment errors may be a buffer overflow or unavailability of destination, therefore you can check the availability and the connection of the destination or find out why the buffer capacity is exceeded.

Configuring service monitoring

To configure the services:

1. In the LogAnalyzer web console, go to the **Settings** → **Services monitoring** section.

2. Specify the values of monitoring parameters for the services.

   Service monitoring does not apply to cold storage.

   If you specify an invalid value that does not fit the range or format, the value is reset to the previously configured value.

3. Click **Save**.

   After saving the parameters, LogAnalyzer registers an audit event: Monitoring thresholds changed for the service.

LogAnalyzer monitors the status of services in accordance with the specified parameters.

In the **Active services** section, you can filter services by statuses or enter a word from the error text, for example, "QPS" or "buffer", in the search field and press ENTER. This results in a list of services with errors. Special characters ", },{, are not allowed in the search string and will produce irrelevant results.

Disabling service monitoring

To disable service monitoring:

1. In the LogAnalyzer web console, go to the **Settings** → **Services monitoring** section.
2. If you want to disable service monitoring only for collectors, in the **Service monitoring. Thresholds setting** window, under **Collectors**, select the **Disable connector errors** check box.

   This disables only the analysis of the **Connector errors** metric for collectors.
3. If you want to disable monitoring for all services, in the **Service monitoring. Thresholds setting** select the **Disable** check box.

LogAnalyzer service monitoring is disabled, and services do not get the yellow status.

Viewing LogAnalyzer metrics

To monitor the performance of its components and the event stream, LogAnalyzer collects and stores a large number of parameters. The VictoriaMetrics time series database is used to collect, store and analyze the parameters. The collected metrics are visualized using Grafana. Dashboards that visualize key performance parameters of various LogAnalyzer components can be found in the **Metrics** section of LogAnalyzer.
The LogAnalyzer Core service configures VictoriaMetrics and Grafana automatically, no user action is required.

The collected metrics are visualized using the Grafana solution. The RPM package of the 'kuma-core' service generates the Grafana configuration and creates a separate dashboard for visualizing the metrics of each service. Graphs in the **Metrics** section appear with a delay of approximately 1.5 minutes.

For full information about the metrics, you can refer to the **Metrics** section of the LogAnalyzer web interface. Selecting this section opens the Grafana portal that is deployed as part of Core installation and is updated automatically. If the **Metrics** section shows core: <port number>, this means that LogAnalyzer is deployed in a high availability configuration and the metrics were received from the host on which the LogAnalyzer Core was installed. In other configurations, the name of the host from which LogAnalyzer receives metrics is displayed.

To determine on which host the Core is running, run the following command in the terminal of one of the controllers:

```
k0s kubectl get pod -n kuma -o wide
```

To determine on which host the Core is running, run the following command in the terminal of one of the controllers:

```
k0s kubectl get pod -n kuma -o wide
```

Collector metrics

Metric name	Description
IO—metrics related to the service input and output.	
Processing EPS	The number of events processed per second.
Output EPS	The number of events per second sent to the destination.

Metric name	Description
Output Latency	The time in milliseconds that passed while sending an event packet and receiving a response from the destination. The median value is displayed.
Output Errors	The number of errors occurring per second while event packets were sent to the destination. Network errors and errors writing to the disk buffer of the destination are displayed separately.
Output Event Loss	The number of events lost per second. Events can be lost due to network errors or errors writing the disk buffer of the destination. Events are also lost if the destination responds with an error code, for example, in case of an invalid request.
Output Disk Buffer Slze	The size of the disk buffer of the collector associated with the destination, in bytes. If a zero value is displayed, no event packets have been placed in the collector's disk buffer and the service is operating correctly.
Write Network BPS	The number of bytes received into the network per second.
Connector errors	The number of errors in the connector logs.

Normalization—metrics related to the normalizers.	
Raw & Normalized event size	The size of the raw event and size of the normalized event. The median value is displayed.
Errors	The number of normalization errors per second.

Filtration—metrics related to filters.	
EPS	The number of events per second matching the filter conditions and sent for processing. The collector only processes events that match the filtering criteria if the user has added the filter to the configuration of the collector service.

Aggregation—metrics related to the aggregation rules.	
EPS	The number of events received and generated by the aggregation rule per second. This metric helps determine the effectiveness of aggregation rules.
Buckets	The number of buckets in the aggregation rule.

Enrichment—metrics related to enrichment rules.	
Cache RPS	The number of requests per second to the local cache.
Source RPS	The number of requests per second to an enrichment source, such as a dictionary.
Source Latency	Time in milliseconds passed while sending a request to the enrichment source and receiving a response from it. The median value is displayed.
Queue	The size of the enrichment request queue. This metric helps to find bottleneck enrichment rules.
Errors	The number of errors per second while sending requests to the enrichment source.

Correlator metrics

Metric name	Description
IO—metrics related to the service input and output.	
Processing EPS	The number of events processed per second.
Output EPS	The number of events per second sent to the destination.
Output Latency	The time in milliseconds that passed while sending an event packet and receiving a response from the destination. The median value is displayed.
Output Errors	The number of errors occurring per second while event packets were sent to the destination. Network errors and errors writing to the disk buffer of the destination are displayed separately.

Metric name	Description
Output Event Loss	The number of events lost per second. Events can be lost due to network errors or errors writing the disk buffer of the destination. Events are also lost if the destination responds with an error code, for example, in case of an invalid request.
Output Disk Buffer SIze	The size of the disk buffer of the collector associated with the destination, in bytes. If a zero value is displayed, no event packets have been placed in the collector's disk buffer and the service is operating correctly.

Correlation—metrics related to correlation rules.	
EPS	The number of correlation events per second generated by the correlation rule.
Buckets	The number of buckers in a correlation rule of the standard type.
Rate Limiter Hits	The number of times the correlation rule exceeded the rate limit per second.
Active Lists OPS	The number of operations requests per second sent to the active list, and the operations themselves.
Active Lists Records	The number of records in the active list.
Active Lists On-Disk Size	The size of the active list on the disk, in bytes.

Enrichment—metrics related to enrichment rules.	
Cache RPS	The number of requests per second to the local cache.
Source RPS	The number of requests per second to an enrichment source, such as a dictionary.
Source Latency	Time in milliseconds passed while sending a request to the enrichment source and receiving a response from it. The median value is displayed.
Queue	The size of the enrichment request queue. This metric helps to find bottleneck enrichment rules.
Errors	The number of errors per second while sending requests to the enrichment source.

Response—metrics associated with response rules.	
RPS	The number of times a response rule was activated per second.

Storage metrics

Metric name	Description
Storage / IO — metrics related to the status of ClickHouse cluster nodes.	
Output EPS	Number of events per second that were inserted into the ClickHouse node or into the disk buffer of the node.
Output Latency	Latency of inserting a batch either into the ClickHouse node or into the disk buffer of the node.
Output Errors	Number of errors per second when inserting a batch into the ClickHouse node or into the disk buffer of the node.
Output Disk Buffer SIze	Size of the disk buffer of the ClickHouse node.
Output Event Loss	Number of events that were permanently lost on each ClickHouse node.
Batch size	The size of the batch of events to be inserted into the ClickHouse node.
Insert interval 5min Q1	Duration of the first quartile of the event insertion interval for the last 5 minutes.

ClickHouse / General—metrics related to the general settings of the ClickHouse cluster.	
Active Queries	The number of active queries sent to the ClickHouse cluster. This metric is displayed for each ClickHouse instance.
QPS	The number of queries per second sent to the ClickHouse cluster.

Failed QPS	The number of failed queries per second sent to the ClickHouse cluster.
Allocated memory	The amount of RAM, in gigabytes, allocated to the ClickHouse process.

ClickHouse / Insert—metrics related to inserting events into a ClickHouse instance.	
Insert EPS	The number of events per second inserted into the ClickHouse instance.
Insert QPS	The number of ClickHouse instance insert queries per second sent to the ClickHouse cluster. If the **Insert QPS** metric demonstrates a growing queue of queries and the metric is greater than 1, we recommend also looking at the **Batch size** metric value to adjust the storage buffering settings in the settings of the storage service configuration. Example: The **Insert QPS** metric is greater than 1 and equals 8. The **Batch size** metric is 1.2 GB (in bytes). In this case, you can find the buffer size by multiplying **Insert QPS** by **Batch size**: 8 * 1.2 = 9.6 GB. Round the resulting value of 9.6 and specify it in bytes (for example, 10000000000 bytes) as the **Buffer size** setting on the **Advanced settings** tab of the storage service configuration. Also specify a **Buffer flush interval** of 2 seconds. Increasing the buffer size and buffer flush interval will help relieve the query queue. Normally, the **Insert QPS** metric should be less than 1.
Failed Insert QPS	The number of failed ClickHouse instance insert queries per second sent to the ClickHouse cluster.
Delayed Insert QPS	The number of delayed ClickHouse instance insert queries per second sent to the ClickHouse cluster. Queries were delayed by the ClickHouse node due to exceeding the soft limit on active merges.
Rejected Insert QPS	The number of rejected ClickHouse instance insert queries per second sent to the ClickHouse cluster. Queries were rejected by the ClickHouse node due to exceeding the hard limit on active merges.
Active Merges	The number of active merges.
Distribution Queue	The number of temporary files with events that could not be inserted into the ClickHouse instance because it was unavailable. These events cannot be found using search.

ClickHouse / Select—metrics related to event selections in the ClickHouse instance.	
Select QPS	The number of ClickHouse instance event select queries per second sent to the ClickHouse cluster.
Failed Select QPS	The number of failed ClickHouse instance event select queries per second sent to the ClickHouse cluster.

ClickHouse / Replication—metrics related to replicas of ClickHouse nodes.	
Active Zookeeper Connections	The number of active connections to the Zookeeper cluster nodes. In normal operation, this number should be equal to the number of nodes in the Zookeeper cluster.
Read-only Replicas	The number of read-only replicas of ClickHouse nodes. In normal operation, no such replicas of ClickHouse nodes must exist.
Active Replication Fetches	The number of active processes of downloading data from the ClickHouse node during data replication.
Active Replication Sends	The number of active processes of sending data to the ClickHouse node during data replication.
Active Replication Consistency Checks	The number of active data consistency checks on replicas of ClickHouse nodes during data replication.

ClickHouse / Networking—metrics related to the network of the ClickHouse cluster.	
Active HTTP Connections	The number of active connections to the HTTP server of the ClickHouse cluster.
Active TCP Connections	The number of active connections to the TCP server of the ClickHouse cluster.
Active Interserver Connections	The number of active service connections between ClickHouse nodes.

LogAnalyzer Core metrics

Metric name	Description
Raft—metrics related to reading and updating the state of the LogAnalyzer Core.	
Lookup RPS	The number of lookup procedure requests per second sent to the LogAnalyzer Core, and the procedures themselves.
Lookup Latency	Time in milliseconds spent running the lookup procedures, and the procedures themselves. The time is displayed for the 99th percentile of lookup procedures. One percent of lookup procedures may take longer to run.
Propose RPS	Number of Raft (SQLite) propose procedure requests per second sent to the LogAnalyzer, and the procedures themselves.
Propose Latency	Time in milliseconds spent running the Raft (SQLite) propose procedures, and the procedures themselves. The time is displayed for the 99th percentile of propose procedures. One percent of propose procedures may take longer to run.
Tasks—metrics related to monitoring the running of tasks on the LogAnalyzer Core	
Active tasks	Number of tasks run per unit time.
Task Execution latency	Duration of running tasks in seconds.
Errors	Number of errors when running tasks.

API—metrics related to API requests.	
RPS	The number of API requests made to the LogAnalyzer Core per second.
Latency	The time in milliseconds spent processing a single API request to the LogAnalyzer Core. The median value is displayed.
Errors	The number of errors per second while sending API requests to the LogAnalyzer Core.

Notification Feed—metrics related to user activity.	
Subscriptions	The number of clients connected to the LogAnalyzer Core via SSE to receive server messages in real time. This number is normally equal to the number of clients that are using the LogAnalyzer web interface.
Errors	The number of errors per second while sending notifications to users.

Schedulers—metrics related to LogAnalyzer Core tasks.	
Active	The number of repeating active system tasks. The tasks created by the user are ignored.
Latency	The time in milliseconds spent running the task. The median value is displayed.
Errors	The number of errors that occurred per second while performing tasks.

LogAnalyzer agent metrics

Metric name	Description
IO—metrics related to the service input and output.	
Processing EPS	The number of events processed per second.
Output EPS	The number of events per second sent to the destination.
Output Latency	The time in milliseconds that passed while sending an event packet and receiving a response from the destination. The median value is displayed.
Output Errors	The number of errors occurring per second while event packets were sent to the destination. Network errors and errors writing to the disk buffer of the destination are displayed separately.
Output Event Loss	The number of events lost per second. Events can be lost due to network errors or errors writing the disk buffer of the destination. Events are also lost if the destination responds with an error code, for example, in case of an invalid request.

Metric name	Description
Output Disk Buffer SIze	The size of the disk buffer of the collector associated with the destination, in bytes. If a zero value is displayed, no event packets have been placed in the collector's disk buffer and the service is operating correctly.
Write Network BPS	The number of bytes received into the network per second.

Event routers metrics

Metric name	Description
IO—metrics related to the service input and output.	
Processing EPS	The number of events processed per second.
Output EPS	The number of events per second sent to the destination.
Output Latency	The time in milliseconds that passed while sending an event packet and receiving a response from the destination. The median value is displayed.
Output Errors	The number of errors occurring per second while event packets were sent to the destination. Network errors and errors writing to the disk buffer of the destination are displayed separately.
Output Event Loss	The number of events lost per second. Events can be lost due to network errors or errors writing the disk buffer of the destination. Events are also lost if the destination responds with an error code, for example, in case of an invalid request.
Output Disk Buffer SIze	The size of the disk buffer of the collector associated with the destination, in bytes. If a zero value is displayed, no event packets have been placed in the collector's disk buffer and the service is operating correctly.
Write Network BPS	The number of bytes received into the network per second.
Connector Errors	The number of errors in the connector log.

General metrics common for all services

Metric name	Description
Process—General process metrics.	
Memory	RAM usage (RSS) in megabytes.
DISK BPS	The number of bytes read from or written to the disk per second.
Network BPS	The number of bytes received/transmitted over the network per second.
Network Packet Loss	The number of network packets lost per second.
GC Latency	The time, in milliseconds, spent executing a GO garbage collection cycle The median value is displayed.
Goroutines	The number of active goroutines. This number is different from the operating system's thread count.

OS—metrics related to the operating system.	
Load	Average load.
CPU	CPU load as a percentage.
Memory	RAM usage (RSS) as a percentage.
Disk	Disk space usage as a percentage.

Metrics storage period

LogAnalyzer operation data is retained for three months by default. This storage period can be changed.

To change the retention period for LogAnalyzer metrics:

1. Log in to the OS of the server on which the LogAnalyzer Core is installed.

2. In the file /etc/systemd/system/multi-user.target.wants/kuma-victoria-metrics.service, in the ExecStart parameter, edit the `--retentionPeriod=<metrics storage period, in months>` flag by inserting the necessary period. For example, `--retentionPeriod=4` means that the metrics will be stored for 4 months.

3. Restart LogAnalyzer by running the following commands in sequence:

   a. `systemctl daemon-reload`

   b. `systemctl restart kuma-victoria-metrics`

The storage period for metrics has been changed.

Managing LogAnalyzer tasks

When working in the application web interface, you can use tasks to perform various operations. For example, you can import assets or export LogAnalyzer event information to a TSV file.

You can run multiple tasks at the same time. If you run tasks of different types, they are executed in parallel within the same tenant. Tasks of the same type can be run simultaneously only if the tenants are different. If you try to run multiple tasks of the same type in one tenant, only the first such task is run.

Viewing the tasks table

The tasks table contains a list of created tasks and is located in the **Task manager** section of the application web interface window.

The ability to view other users' tasks depends on the user role. A user with the General administrator role can view the tasks of all users. Users with roles that allow reading and editing shared resources do not have access to tasks. Users with other roles, including users with the Tenant administrator role, can view all tasks in tenants available to them.

By default, the **View just mine** check box is selected in the **Task manager** section. To view all tasks, clear this check box.

The tasks table contains the following information:

- **State**—the state of the task. One of the following statuses can be assigned to a task:
  - **Running**—the task is active.
  - **Completed**—the task is complete.
  - **Cancel**—the task was canceled by the user.
  - **Error**—the task was not completed because of an error. The error message is displayed if you hover the mouse over the exclamation mark icon.
- **Task**—the task type. The application provides the following types of tasks:
  - **Events export** exports LogAnalyzer events.
  - **Threat Lookup** requests data from the Kaspersky Threat Intelligence Portal.
  - **KSC assets import** KSC assets import.
  - **KSC assets attributes import** KSC assets attributes import.
  - **KICS/KATA assets import** imports asset information from KICS/KATA.
  - **Accounts import** imports user information from Active Directory.
  - **Run program via KEDR** starts LogAnalyzer using Kaspersky Endpoint Detection and Response.
  - **Repository update** updates the LogAnalyzer repository to receive the resource packages from the source specified in settings of the source.
  - **Update monitoring policies** updates the monitoring policy settings.
  - **Clearing resource change history** deletes information about resource changes.
  - **Export event sources** exports event sources in CSV format.
  - **Query to KIRA** Query to KIRA.
- **Created by**—the user who created the task. If the task was created automatically, the column will show **Scheduled task**.
- **Created**—the date and time when the task was created.
- **Last updated**—date and time of the last update of the task.
- **Tenant**—the name of the tenant in which the task was started.

The date format depends on the localization language selected in the application settings. Possible date format options:

- English localization: YYYY-MM-DD.
- Russian localization: DD.MM.YYYY.

Configuring the display of the tasks table

You can customize the display of columns and the order in which they appear in the tasks table.

To customize the display in the task table:

1. In the LogAnalyzer web interface, select the **Task manager** section.

   The tasks table is displayed.

2. In the table header, click the gear ( ) icon.

3. In the pane that is displayed:

   - If you want to enable display of a column in the table, select the check box next to the name of the parameter that you want to display in the table.

   - If you do not want the parameter to be displayed in the table, clear the check box.

   At least one check box must be selected.

4. If you want to reset the settings, click the **Default** link.

5. If you want to change the order in which the columns are displayed in the table, move the mouse cursor over the name of the column, hold down the left mouse button and drag the column to the necessary position.

The display of columns in the tasks table will be configured.

Viewing task run results

To view the results of a task:

1. In the LogAnalyzer web interface, select the **Task manager** section.

   The tasks table is displayed.

2. Do one of the following:

   - For tasks of the **Events export** and **Export event sources** type, click the link with the task type in the **Task** column and select **Download**.

     Task results are downloaded as a TSV file.

   - Click the name of the user that created the task or **Scheduled task** in the **Created by** column.

     Task results are displayed in the sidebar.

   For **Repository update** tasks, results are not displayed.

The presentation of task results in LogAnalyzer depend on the type of the task. For example, for **Threat Lookup**, information about the type of the detected object is displayed along with information about the file, detections, and access paths is displayed. For **KSC assets import** and **KICS/KATA assets import** tasks, information about the tenant and the number of downloaded objects is displayed. For the **Query to KIRA** task, information about task status, the date when the query was sent and the response received, and a link to the event are displayed.

In this section, the **View just mine** check box in the **Created by** column is selected by default. To view all tasks, clear this check box.

Restarting a task

To restart a task:

1. In the LogAnalyzer web interface, select the **Task manager** section.

   The tasks table is displayed.

2. Select the check box next to the task you want to restart.

3. Click the **Restart** button.

The task will be restarted.

Connecting to an SMTP server

LogAnalyzer can be configured to send email notifications using an SMTP server. Users will receive notifications if the **Receive email notifications** check box is selected in their profile properties.

Only one SMTP server can be added to process LogAnalyzer notifications. You can manage the SMTP server connection in the LogAnalyzer web interface under **Settings** → **Other** → **General** → **SMTP server settingsSMTP server settings**.

To configure SMTP server connection:

1. Open the LogAnalyzer web interface and select **Settings** → **Other** → **Common**.
2. Under **SMTP server settings**, change the relevant settings:
   - Disable the **State** toggle switch if you want to disable connection to the SMTP server.
   - **Host** (required)—SMTP host in one of the following formats: hostname, IPv4, IPv6.
   - **Port** (required)—SMTP port. The value must be an integer from 1 to 65,535.
   - **From** (required)—email address of the message sender. For example, `kuma@company.com`.
   - **Alias for KUMA Core server**—name of the LogAnalyzer Core server that is used in your network. Must be different from the FQDN.
   - If necessary, use the **Secret** drop-down list to select a secret of the **credentials** type that contains the account credentials for connecting to the SMTP server.

     Add secret

     To create a secret:

     1. In the **Name** field, enter the name of the secret.
     2. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector.
     3. If necessary, enter a description of the secret in the **Description** field.
     4. Click the **Create** button.

     The secret is added and displayed in the **Secret** drop-down list.
   - Select the necessary frequency of notifications in the **Monitoring notifications interval** drop-down list.

     Notifications from the source about a monitoring policy triggering are repeated after the selected period until the status of the source becomes green again.

     If you select **Notify once**, the notification about the monitoring policy triggering is sent only once during the whole time when the source has the red status. However, a notification about the monitoring policy triggering is sent every time the source status changes from red to green.
   - Turn on the **Disable monitoring notifications** toggle button if you do not want to receive notifications about the state of event sources. The toggle switch is turned off by default. If, at the time of enabling notifications in LogAnalyzer, there are event sources with a red status and you have not yet received notifications about this, you will receive one notification for the whole period, even if the status of the sources changed more than once while monitoring notifications were turned off.
3. Click **Save**.

The SMTP server connection is now configured, and users can receive email messages from LogAnalyzer.

LogAnalyzer notifications

Notifications in the LogAnalyzer web interface

By default, the **Notify about alerts in web interface** check box is selected for all users, and alert notifications are displayed in the LogAnalyzer web interface.

If necessary, you can turn off the display of such notifications by clearing the check box in the user profile.

Standard notifications

LogAnalyzer can be configured to send email notifications using an SMTP server. To do so, configure a connection to an SMTP server and select the **Receive email notifications** check box. Only a user with the General administrator role can receive email notifications.

If the **Receive email notifications** check box is selected for a user with the General administrator role, after enabling the setting, an email notification is sent to the user every 6 hours in accordance with the following rule:

- If at least one server has a non-empty Warning field at the time scheduled for sending the message, the message is sent.
- One message is sent for all services with the yellow status. If no services have the yellow status, no message is sent.

The 6-hour interval is respected unless the LogAnalyzer Core is restarted. After each restart of the Core, the 6-hour interval is reset.

LogAnalyzer automatically notifies users about the following events:

- A report was created (the users listed in the report template receive a notification).
- A task was performed (the users who created the task receive a notification).
- New resource packages are available. They can be obtained by updating the LogAnalyzer repository (the users whose email address is specified in the task settings are notified).
- The daily average EPS has exceeded the limit set by the license.
- The hourly average EPS has exceeded the limit set by the SMB license.

LogAnalyzer logs

LogAnalyzer provides the following types of logs:

- Installer logs
- Component logs

You can also generate a report with diagnostic information about your device using the collect.sh utility. For detailed information on using the utility, please refer to the Knowledge Base.

Installer logs

LogAnalyzer automatically creates files containing logs of installation, reconfiguration, or removal.

The logs are stored in the ./log/ subdirectory in the installer directory. The name of the log file reflects the date and time when the corresponding script was started.

Names are generated in the following formats:

- Installation log: install-YYYYMMDD-HHMMSS.log. For example: install-20231031-102409.log
- Removal logs: uninstall-YYYYMMDD-HHMMSS.log. For example: uninstall-20231031-134011.log
- Reconfiguration logs: expand-YYYYMMDD-HHMMSS.log. For example: expand-20231031-105805.log

LogAnalyzer creates a new log file each time the installation, reconfiguration, or removal script is started. Log rotation or automatic deletion is not performed.

The log incorporates the lines of the inventory file used when the corresponding command was invoked, and the ansible log. For each task, the following information is listed in this order: task start time (Tuesday, October 31, 2023 10:29:14 +0300), run time of the previous task (0:00:02.611), and the total time passed since the installation, reconfiguration, or removal was initiated (0:04:56.906).

Example:

TASK [Add columns to the replicated table] ************************************

Tuesday, October 31, 2023 10:29:14 +0300 (0:00:02.611) 0:04:56.906 *******

Component logs

By default, only errors are logged for all LogAnalyzer components. To receive detailed data in logs, configure **Debug** mode in the component settings.

The Core logs are stored in the /opt/kaspersky/kuma/core/<LogAnalyzer Core service ID>/log/corecore directory and are archived when they reach the size of 5 GB or the age of 7 days, whichever occurs first. These conditions are checked once daily. Archives are kept in the log folder for 7 days, after 7 days the archive is deleted. A maximum of four archived logs are stored on the server at the same time. Whenever a new log archive is created, if the total number of archives becomes greater than four, the oldest log archive is deleted. If the logs fill up quickly, you must have enough disk space to create a copy of the log file and archive it as part of log rotation.

The component logs are appended until the file reaches 5 GB. When the log reaches 5 GB, it is archived and new events are written to a new log. Archives are kept in the log folder for 7 days, after 7 days the archive is deleted. A maximum of four archived logs are stored on the server at the same time. Whenever a new log archive is created, if the total number of archives becomes greater than four, the oldest log archive is deleted.

Debug mode is available for the following components:

Core	You can enable it in the LogAnalyzer web interface, in the **Settings** → **Other** → **General** → **Core settings** → **Debug** section. Storage location: /opt/kaspersky/kuma/core/<`LogAnalyzer Core service ID`>/log/core You can download the Core logs from the LogAnalyzer web interface, in the **Resources** → **Active services** section by selecting the Core service and clicking **Log**. If LogAnalyzer is installed in a high availability configuration, refer to the *Viewing Core logs in Kubernetes* section below.

Services:
 	
• Storage
• Collectors
• Agents
	To enable it, use the **Debug** toggle switch in the settings of the service.
	
	Storage location: the service installation directory. For example, /opt/kaspersky/kuma/<`service type, collector/storage`>/<`service ID`>/log/<`service type, collector/storage`>. You can download the service logs from the LogAnalyzer web interface, in the **Resources** → **Active services** section by selecting the desired service and clicking **Log**.
	
	Logs residing on Linux machines can be viewed by running the journalctl and tail command. For example:
	
	• Storage. To return the latest logs from the storage installed on the server, run the following command:
	`journalctl -f -u kuma-storage-< storage ID >`
	• Collectors. To return the latest logs from a specific collector installed on the server, run the following command:
	`journalctl -f -u kuma-collector-< collector ID >`
	• Agents. To return the latest logs from an agent installed on the server, run the following command:
	`tail -f /opt/kaspersky/agent/< Agent ID >/log/agent`
	The activity of Agents on Windows machines is always logged if they are assigned the logon as a service permission. Data is specified in more detail when the **Debug** check box is selected. Agent logs on Windows machines can be viewed in the file located at the path %PROGRAMDATA%\Kaspersky Lab\KUMA\<Agent ID>\agent.log. Logs of Agents on Linux machines are stored in the agent installation directory.

Resources: • Connectors • Destinations • Enrichment rules	To enable it, use the **Debug** toggle switch in the settings of the service to which the resource is linked. The logs are stored on the machine hosting the installed service that uses the relevant resource. Detailed data for resources can be viewed in the log of the service linked to a resource.

Viewing Core logs in Kubernetes

When Core log files reach 100 MB, they are archived and a new log is written. No more than five files are stored at a time. If there are more than five files when a new log appears, the oldest file is deleted.

On worker nodes, you can view the logs of containers and pods residing on these nodes in the file system of the node.
For example:
/var/log/pods/kuma_core-deployment-<UID>/core/*.log
/var/log/pods/kuma_core-deployment-<UID>/mongodb/*.log

To view the logs of all containers in the Core pod:

```
k0s kubectl logs -l app=core --all-containers -n kuma
```

To view the log of a specific container:

```
k0s kubectl logs -l app = core -c <container_name> -n kuma
```

To enable real-time log viewing, add the -f switch:

```
k0s kubectl logs -f -l app=core --all-containers -n kuma
```

To view the logs of the previous pod that was overwritten by a new one (for example, when recovering from a critical error or after redeployment), add the --previous switch:

```
k0s kubectl logs -l app=core -c core -n kuma --previous
```

To access the logs from other hosts that are not included in the cluster, you need the k0s-kubeconfig.yml file containing the access credentials created during LogAnalyzer installation, and the locally installed kubectl cluster management utility.
The cluster controller or traffic balancer specified in the server parameter of the k0s-kubeconfig.yml file must be accessible over the network.

The file path must be exported to a variable:
`export KUBECONFIG=/< file path >/k0s-kubeconfig.yml`

You can use kubeclt to view the logs. For example:

```
kubectl logs -l app=core -c mongodb -n kuma
```

Working with geographic data

A list of mappings of IP addresses or ranges of IP addresses to geographic data can be uploaded to LogAnalyzer for use in event enrichment.

Geodata format

Geodata can be uploaded to LogAnalyzer as a CSV file in UTF-8 encoding. A comma is used as the delimiter. The first line of the file contains the field headers: `Network,Country,Region,City,Latitude,Longitude`.

CSV file description

Field header name in CSV	Field description	Example
`Network`	IP address in one of the following formats: • Single IP address • Range of IP addresses • IP address in CIDR format. Mixing of IPv4 and IPv6 addresses is allowed. Required field.	• `192.168.2.24` • `192.168.2.25-192.168.2.35` • `131.10.55.70/8` • `2001:DB8::0/120`
`Country`	Country designation used by your organization. For example, this could be its name or code. Required field.	• `Russia` • `RU`
`Region`	Regional designation used by your organization. For example, this could be its name or code.	• `Sverdlovsk Oblast` • `RU-SVE`
`City`	City designation used by your organization. For example, this could be its name or code.	• `Yekaterinburg` • `65701000001`

`Latitude`	Latitude of the described location in decimal format. This field can be empty, in which case the value of 0 is used when importing data into LogAnalyzer.	`56.835556`
`Longitude`	Longitude of the described location in decimal format. This field can be empty, in which case the value of 0 is used when importing data into LogAnalyzer.	`60.612778`

Converting geographic data from MaxMind to IP2Location

LogAnalyzer can use geodata obtained from MaxMind, IP2Location, or data in a format compatible with these. Before using archives with CSV geodata files, you must convert these to the format that LogAnalyzer supports. The names of CSV files being converted inside archives must match the following templates:

- (GeoIP2|GeoLite2|RU-GeoIP)-(City|Country)-Blocks-IPv(4|6).csv
- (GeoIP2|GeoLite2|RU-GeoIP)-(City|Country)-Locations-%s.csv

Conversion can be done using the script below. Make sure that the CSV files with geodata do not contain duplicate records. For example, if a file has few columns, different records may contain data from the same network with the same geodata. Such files cannot be converted. To successfully perform the conversion, make sure that there are no duplicate rows and that every row has at least one unique field.

Download script

Python 2.7 or later is required to run the script.

Script start command:

```
python converter.py --type <type of geographic data being processed: "maxmind" or
"ip2location"> --out <directory where a CSV file containing geographic data in
LogAnalyzer format will be placed> --input <path to the ZIP archive containing
geographic data from MaxMind or IP2location>
```

When the script is run with the `--help` flag, help is displayed for the available script parameters: `python converter.py --help`

Command for converting a file containing a Russian database of IP address ranges from a MaxMind ZIP archive:

```
python converter.py --type maxmind --lang ru --input MaxMind.zip --out
geoip_maxmind_ru.csv
```

If the `--lang` parameter is not specified, the script receives information from the GeoLite2-City-Locations-en.csv file from the ZIP archive by default.

Absence of the `--lang` parameter for MaxMind is equivalent to the following command:

```
python converter.py --type maxmind --input MaxMind.zip --out geoip_maxmind.csv
```

Command for converting a file from an IP2Location ZIP archive:

```
python converter.py --type ip2location --input IP2LOCATION-LITE-DB11.CSV.ZIP --out
geoip_ip2location.csv
```

Command for converting a file from several IP2Location ZIP archives:

```
python converter.py --type ip2location --input IP2LOCATION-LITE-DB11.CSV.ZIP
IP2LOCATION-LITE-DB11.IPV6.CSV.ZIP --out geoip_ip2location_ipv4_ipv6.csv
```

The `--lang` parameter is not used for IP2Location.

Required sets of fields

The MaxMind source files GeoLite2-City-Blocks-IPv4.csv and GeoLite2-City-Blocks-IPv6.csv must contain the following set of fields:

network,geoname_id,registered_country_geoname_id,represented_country_geoname_id,
is_anonymous_proxy,is_satellite_provider,postal_code,latitude,longitude,accuracy_radius

Example set of source data:

```
network,geoname_id,registered_country_geoname_id,represented_country_geoname_id,
is_anonymous_proxy,is_satellite_provider,postal_code,latitude,longitude,accuracy_radius

1.0.0.0/24,2077456,2077456,,0,0,,-33.4940,143.2104,1000

1.0.1.0/24,1814991,1814991,,0,0,,34.7732,113.7220,1000
```

The remaining CSV files with the locale code must contain the following set of fields:

geoname_id,locale_code,continent_code,continent_name,country_iso_code,country_name,
subdivision_1_iso_code,subdivision_1_name,subdivision_2_iso_code,subdivision_2_name,
city_name,metro_code,time_zone,is_in_european_union

Example set of source data:

```
geoname_id,locale_code,continent_code,continent_name,country_iso_code,country_name,
subdivision_1_iso_code,subdivision_1_name,subdivision_2_iso_code,subdivision_2_name,
city_name,metro_code,time_zone,is_in_european_union

1392,de,AS,Asien,IR,Iran,02,Mazandaran,,,,,Asia/Tehran,0

7240,de,AS,Asien,IR,Iran,28,Nord-Chorasan,,,,,Asia/Tehran,0
```

The source IP2Location files must contain data on the network ranges, Country, Region, City, Latitude, and Longitude

Example set of source data:

```
"0","16777215","-","-","-","-","0.000000","0.000000","-","-"

"16777216","16777471","US","United States of America","California","Los
Angeles","34.052230","-118.243680","90001","-07:00"
```

```
"16777472","16778239","CN","China","Fujian","Fuzhou","26.061390","119.306110","350004","+6
```

If the source files contain a different set of fields than the one indicated in this section, or if some fields are missing, the missing fields in the target CSV file will be empty after conversion.

Importing and exporting geographic data

If necessary, you can manually import and export geographic data into LogAnalyzer. Geographic data is imported and exported in a CSV file. If the geographic data import is successful, the previously added data is overwritten and an audit event is generated in LogAnalyzer.

To import geographic data into LogAnalyzer:

1. Prepare a CSV file containing geographic data.

   Geographic data received from MaxMind and IP2Location must be converted to a format supported by LogAnalyzer.

2. In the LogAnalyzer web interface, open the **Settings** → **Other** → **General** section.

3. Under **Geographic data**, click the **Import from file** button and select a CSV file containing geographic data.

   Wait for the geographic data import to finish. The data import is interrupted if the page is refreshed.

The geographic data is uploaded to LogAnalyzer.

To export geographic data from LogAnalyzer:

1. In the LogAnalyzer web interface, open the **Settings** → **Other** → **General** section.

2. Under **Geographic data**, click the **Export** button.

Geographic data will be downloaded as a CSV file named geoip.csv (in UTF-8 encoding) based on the settings of your browser.

The data is exported in the same format as it was uploaded, with the exception of IP address ranges. If a range of addresses was specified in the `1.0.0.0/24` format in the file imported into LogAnalyzer, the range has the `1.0.0.0-1.0.0.255` format in the exported file.

Default mapping of geographic data

If you select the `SourceAddress`, `DestinationAddress` and `DeviceAddress` event fields as the IP address source when configuring a geographic data enrichment rule, the **Apply default mapping** button becomes available. You can use this button to add preconfigured mapping pairs of geographic data attributes and event fields as described below.

Default mappings for the SourceAddress event field

Geodata attribute	Event field
Country	`SourceCountry`

Region	SourceRegion
City	SourceCity
Latitude	SourceLatitude
Longitude	SourceLongitude

Default mappings for the DestinationAddress event field

Geodata attribute	Event field
Country	DestinationCountry
Region	DestinationRegion
City	DestinationCity
Latitude	DestinationLatitude
Longitude	DestinationLongitude

Default mappings for the DeviceAddress event field

Geodata attribute	Event field
Country	DeviceCountry
Region	DeviceRegion
City	DeviceCity
Latitude	DeviceLatitude
Longitude	DeviceLongitude

User guide

This section provides information about managing the LogAnalyzer SIEM system.

LogAnalyzer resources

Resources are LogAnalyzer components that contain parameters for implementing various functions: for example, establishing a connection with a given web address or converting data according to certain rules. These modular components are assembled into resource sets for services, which in turn are used to create LogAnalyzer services.

Resources are contained in the **Resources** section, **Resources** block of LogAnalyzer web interface. The following resource types are available:

- **Normalizers**—resources of this type contain rules for converting incoming events into the format used by LogAnalyzer. After processing in the normalizer, the "raw" event becomes normalized and can be processed by other LogAnalyzer resources and services.
- **Connectors**—resources of this type contain settings for establishing network connections.
- **Aggregation rules**—resources of this type contain rules for combining several basic events of the same type into one aggregation event.
- **Enrichment rules**—resources of this type contain rules for supplementing events with information from third-party sources.
- **Destinations**—resources of this type contain settings for forwarding events to a destination for further processing or storage.
- **Filters**—resources of this type contain criteria for selecting individual events from the event stream to be sent to processing.
- **Dictionaries**—resources of this type are used to store keys and their values, which may be required by other LogAnalyzer resources and services.
- **Proxies**—resources of this type contain settings for using proxy servers.
- **Secrets**—resources of this type are used to securely store confidential information (such as credentials) that LogAnalyzer needs to interact with external services.

When you click on a resource type, a window opens displaying a table with the available resources of this type. The table contains **the following columns**:

You can configure the set of columns for the table and their order, as well as configure the display of information in some columns:

- You can show or hide columns in the menu that you can open by clicking the gear (⚙) icon in the toolbar in the upper part of the table.
- You can reorder the columns by dragging column headings to the left or right.
- You can configure the display of information in a column by clicking the down arrow (⌄) icon in the heading of the column.

- **Name**—the name of the resource. Can be used to search for resources and sort them.
- **Updated**—the date and time of the last update of a resource. Can be used to sort resources.
- **Created by**—the name of the user who created a resource.
- **Description**—the description of a resource.
- **Type**—the type of the resource. Displayed for all resource types except **Aggregation rules**, **Enrichment rules**, **Filters**, and **Proxy servers**.
- **Resource path**—address in the resource tree. Displayed in the tree of folders, starting from the tenant in which the resource was created.
- **Tags**—tags assigned to the resource. A resource can have more than one tag.

  Tags are part of the resource and are imported with the resource.

- **Package name**—the name of the package in which the resource was imported from the repository.

The maximum table size is not limited. If you want to select all resources, scroll to the end of the table and select the **Select all** check box, which selects all available resources in the table.

The table of resources in the lower part displays the number of resources from tenants that are available to you in the table:

- **Total** is the total amount or the amount with the filter or search applied.
- **Selected** is the number of selected resources.

When filters are applied, the resource selection and the **Selected** value are reset. If the amount of resources changes due to the actions (for example, deletion) undertake by another user, the displayed number of resources changes after you refresh the page, perform an action with the resource, or apply a filter.

Resources can be organized into folders. The folder structure is displayed in the left part of the window: root folders correspond to tenants and contain a list of all resources of the tenant. All other folders nested within the root folder display the resources of an individual folder. When a folder is selected, the resources it contains are displayed as a table in the right pane of the window.

Resources can be created, edited, copied, moved from one folder to another, and deleted. Resources can also be exported and imported.

KUMA comes with a set of predefined resources, which can be identified by the "[OOTB]<resource_name>" name. OOTB resources are protected from editing.

If you want to adapt a predefined OOTB resource to your organization's infrastructure:

1. In the **Resources**–<resource type> section, select the OOTB resource that you want to edit.
2. In the upper part of the LogAnalyzer web interface, click **Duplicate**, then click **Save**.
3. A new resource named "[OOTB]<resource_name> - copy" is displayed in the web interface.
4. Edit the copy of the predefined resource as necessary and save your changes.

The adapted resource is available for use.

Operations with resources

To manage LogAnalyzer resources, you can create, move, copy, edit, delete, import, and export them. These operations are available for all resources, regardless of the resource type.

The table of resources in the lower part displays the number of resources from tenants that are available to you in the table:

- **Total** is the total amount or the amount with the filter or search applied.
- **Selected** is the number of selected resources.

When filters are applied, the resource selection and the **Selected** value are reset. If the amount of resources changes due to the actions (for example, deletion) undertake by another user, the displayed number of resources changes after you refresh the page, perform an action with the resource, or apply a filter.

LogAnalyzer resources reside in folders. You can add, rename, move, or delete resource folders.

Creating, renaming, moving, and deleting resource folders

Resources can be organized into folders. The folder structure is displayed in the left part of the window: root folders correspond to tenants and contain a list of all resources of the tenant. All other folders nested within the root folder display the resources of an individual folder. When a folder is selected, the resources it contains are displayed as a table in the right pane of the window.

You can create, rename, move and delete folders.

To create a folder:

1. Select the folder in the tree where the new folder is required.
2. Click the **Add folder** button.

The folder will be created.

To rename a folder:

1. Locate required folder in the folder structure.
2. Hover over the name of the folder.
   The ... icon will appear near the name of the folder.
3. Open the ... drop-down list and select **Rename**.
   The folder name will become active for editing.
4. Enter the new folder name and press **ENTER**.

> The folder name cannot be empty.

The folder will be renamed.

To move a folder,

Drag and drop the folder to a required place in folder structure by clicking its name.

> Folders cannot be dragged from one tenant to another.

To delete a folder:

1. Select the relevant folder in the folder structure.
2. Right-click to bring up the context menu and select **Delete**.
   The conformation window appears.
3. Click **OK**.

The folder will be deleted.

> The program does not delete folders that contain files or subfolders.

Creating, duplicating, moving, editing, and deleting resources

You can create, move, copy, edit, and delete resources.

To create the resource:

1. In the **Resources** → **<resource type>** section, select or create a folder where you want to add the new resource.

   Root folders correspond to tenants. For a resource to be available to a specific tenant, it must be created in the folder of that tenant.

2. Click the **Add <resource type>** button.

   The window for configuring the selected resource type opens. The available configuration parameters depend on the resource type.

3. Enter a unique resource name in the **Name** field.

4. Specify the required parameters (marked with a red asterisk).

5. If necessary, specify the optional parameters (not required).

6. Click **Save**.

The resource will be created and available for use in services and other resources.

To move the resource to a new folder:

1. In the **Resources** → **<resource type>** section, find the required resource in the folder structure.

2. Select the check box near the resource you want to move. You can select multiple resources.

   The ⣿ icon appears near the selected resources. The number of selected resources is displayed in the lower part of the table.

3. Use the ⣿ icon to drag and drop resources to the required folder.

The resources will be moved to the new folders.

> You can only move resources to folders of the tenant in which the resources were created. Resources cannot be moved to another tenant's folders.

To copy the resource:

1. In the **Resources** → **<resource type>** section, find the required resource in the folder structure.

2. Select the check box next to the resource that you want to copy and click **Duplicate**.

   The number of selected resources is displayed in the lower part of the table.

   A window opens with the settings of the resource that you have selected for copying. The available configuration parameters depend on the resource type.

   The `<selected resource name> - copy` value is displayed in the **Name** field.

3. Make the necessary changes to the parameters.

4. Enter a unique name in the **Name** field.

5. Click **Save**.

The copy of the resource will be created.

To edit the resource:

1. In the **Resources** → **<resource type>** section, find the required resource in the folder structure.
2. Select the resource.

   A window with the settings of the selected resource opens. The available configuration parameters depend on the resource type.
3. Make the necessary changes to the parameters.
4. Do one of the following:

   - Click **Save** to save your changes.
   - Click **Save with comment**, and in the displayed window, add a comment that describes your changes. The changes are saved and the comment is added to the created version of the resource.

The resource is updated and a new version is created for it. If this resource is used in a service, restart the service to apply the new version of the resource.

If the current resource is not editable, you can go to the card of another resource by clicking the **View** button. This button becomes available in batch resources when you click another resource linked to your current resource.

If, when saving changes to a resource, it turns out that the current version of the resource has been modified by another user, you are prompted to select one of the following actions:

- Save your changes as a new version of the resource on top of the changes made by the other user.
- Save your changes as a new resource.

  In this case, a duplicate of the original resource is created with the changed settings. The `- copy` string is added to the name of the new resource, and the name and version of the resource that was duplicated is specified in the version comments of the new resource.
- Discard your changes.

  Discarded changes cannot be restored.

To delete the resource:

1. In the **Resources** → **<resource type>** section, find the required resource in the folder structure.
2. Select the check box next to the resource that you want to delete and click **Delete**.

   The number of selected resources is displayed in the lower part of the table. A confirmation window opens.
3. Click **OK**.

The resource and all its saved versions are deleted.

Bulk deletion of resources

In the LogAnalyzer web interface, you can select multiple resources and delete them.

You must have the right to delete resources.

To delete resources:

1. In the **Resources** → **<resource type>** section, find the required resource in the folder structure.
2. Select the check boxes next to the resources that you want to delete.

   In the lower part of the table, the total number of resources and the number of resources selected will be displayed.
3. Click **Delete**.

   This opens a window that tells you whether it is safe to delete resources, depending on whether the resources selected for deletion are linked to other resources.

   For all resources that cannot be deleted, the application displays a table of links in a modal window.
4. Click **Delete**.

   Only resources without links are deleted.

Deleting folders with resources

You can select the delete operation for any folder at any level, except the tenant.

To delete a folder with resources:

1. In the **Resources** section, select a folder.
2. Click the ⋮ button and select the **Delete** option.

   This opens a window prompting you to confirm deletion. The window displays a field in which you can enter the generated value. Also, if dependent resources exist in the folder, a list of dependencies is displayed.
3. Enter the generated value.
4. Confirm the deletion.

You can delete a folder if:

- The folder does not contain any subfolders or resources.
- The folder does not contain any subfolders, but does contain unlinked resources.
- None of the resources in the folder are dependencies of anything (services, resources, integrations).

Updating resources

Kaspersky regularly releases packages with resources that can be imported from the repository. You can specify an email address in the settings of the **Repository update** task. After the first execution of the task, KUMA starts sending notifications about the packages available for update to the specified address. You can update the repository, analyze the contents of each update, and decide if to import and deploy the new resources in the operating infrastructure. KUMA supports updates from Kaspersky servers and from custom sources, including updates without direct internet access using the update mirror mechanism. If you have other Kaspersky products in the infrastructure, you can connect KUMA to existing update mirrors. The update subsystem expands KUMA capabilities to respond to the changes in the threat landscape and the infrastructure. The capability to use it without direct Internet access ensures the privacy of the data processed by the system.

To update resources, perform the following steps:

1. Update the repository to deliver the resource packages to the repository. The repository update is available in two modes:
   - Automatic update
   - Manual update
2. Import the resource packages from the updated repository into the tenant.

For the service to start using the resources, make sure that the updated resources are mapped after performing the import. If necessary, link the resources to collectors or agents, and update the settings.

To enable automatic update:

1. In the **Settings** → **Repository update** section, configure the **Data refresh interval in hours**. The default value is 24 hours.
2. Specify the **Update source**. The following options are available:
   - Kaspersky update servers.
     You can view the list of update servers in the Knowledge Base.
   - Custom source:
     - The URL to the shared folder on the HTTP server.
     - The full path to the local folder on the host where the KUMA Core is installed.
       If a local folder is used, the kuma system user must have read access to this folder and its contents.
3. If necessary, in the **Proxy server** list, select an existing proxy server to be used when running the **Repository update** task.
   You can also create a new proxy server by clicking the plus (+) icon.

4. Specify the **Emails for notification** by clicking the **Add** button. The notifications that new packages or new versions of the packages imported into the tenant are available in the repository are sent to the specified email addresses.

   If you specify the email address of a KUMA user, the **Receive email notifications** check box must be selected in the user profile. For emails that do not belong to any KUMA user, the messages are received without additional settings. The settings for connecting to the SMTP server must be specified in all cases.

5. Click **Save**. The update task starts shortly. Then the task restarts according to the schedule.

To manually start the repository update:

1. To disable automatic updates, in the **Settings** → **Repository update** section, select the **Disable automatic update** check box. This check box is cleared by default. You can also start a manual repository update without disabling automatic update. Starting an update manually does not affect the automatic update schedule.

2. Specify the **Update source**. The following options are available:

   - Kaspersky update servers.
   - Custom source:
     - The URL to the shared folder on the HTTP server.
     - The full path to the local folder on the host where the KUMA Core is installed.

       If a local folder is used, the kuma user must have access to this folder and its contents.

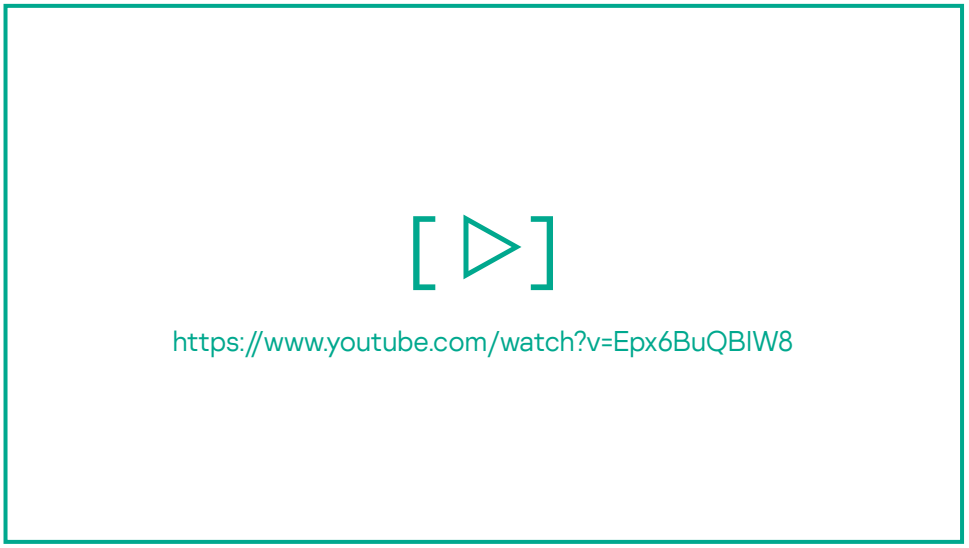
3. If necessary, in the **Proxy server** list, select an existing proxy server to be used when running the **Repository update** task.

   You can also create a new proxy server by clicking the plus (+) icon.

4. Specify the **Emails for notification** by clicking the **Add** button. The notifications that new packages or new versions of the packages imported into the tenant are available in the repository are sent to the specified email addresses.

   If you specify the email address of a KUMA user, the **Receive email notifications** check box must be selected in the user profile. For emails that do not belong to any KUMA user, the messages are received without additional settings. The settings for connecting to the SMTP server must be specified in all cases.

5. Click **Run update**. Thus, you simultaneously save the settings and manually start the **Repository update** task.



https://www.youtube.com/watch?v=Epx6BuQBIW8

Configuring a custom source using Kaspersky Update Utility

You can update resources without Internet access by using a custom update source via the Kaspersky Update Utility.

Configuration consists of the following steps:

1. Configuring a custom source using Kaspersky Update Utility:

   a. Installing and configuring Kaspersky Update Utility on one of the computers in the corporate LAN.

   b. Configuring copying of updates to a shared folder in Kaspersky Update Utility settings.

2. Configuring update of the KUMA repository from a custom source.

Configuring a custom source using Kaspersky Update Utility:

You can download the Kaspersky Update Utility distribution kit from the Kaspersky Technical Support website.

1. In Kaspersky Update Utility, enable the download of updates for LogAnalyzer 4.0:

   - Select **Application list**, then click the **Select** button. This opens a window; in that window, in the **Perimeter control** section, select the check box next to LogAnalyzer 4.0 to enable updates.

   - If you work with Kaspersky Update Utility using the command line, add the following line to the [ComponentSettings] section of the updater.ini configuration file or specify the `true` value for an existing line:

     ```
     KasperskyUnifiedMonitoringAndAnalysisPlatform_4_0=true
     ```

2. In the **Downloads** section, specify the update source. By default, Kaspersky update servers are used as the update source.

3. In the **Downloads** section, in the **Update folders** group of settings, specify the shared folder for Kaspersky Update Utility to download updates to. The following options are available:

   - Specify the local folder on the host where Kaspersky Update Utility is installed. Deploy the HTTP server for distributing updates and publish the local folder on it. In KUMA, in the **Settings** → **Repository update** → **Custom source** section, specify the URL of the local folder published on the HTTP server.

   - Specify the local folder on the host where Kaspersky Update Utility is installed. Make this local folder available over the network. Mount the network-accessible local folder on the host where KUMA is installed. In KUMA, in the **Settings** → **Repository update** → **Custom source** section, specify the full path to the local folder.

For detailed information about working with Kaspersky Update Utility, refer to the Kaspersky Knowledge Base.

Exporting resources

If shared resources are hidden for a user, the user cannot export shared resources or resources that use shared resources.

To export resources:

1. In the **Resources** section, click **Export resources**.

   The **Export resources** window opens with the tree of all available resources.

2. In the **Password** field enter the password that must be used to protect exported data.

3. In the **Tenant** drop-down list, select the tenant whose resources you want to export.

4. Check boxes near the resources you want to export.

   If selected resources are linked to other resources, linked resources will be exported, too. The number of selected resources is displayed in the lower part of the table.

5. Click the **Export** button.

Current versions of resources in a password-protected file are saved on your computer in accordance with your browser settings. Previous versions of the resources are saved in the file. The Secret resources are exported blank.

To export a previous version of a resource:

1. In the LogAnalyzer web interface, in the **Resources** section, select the type of resources that you need.

   This opens a window opens with a table of available resources of this type.

   If you want to view all resources, in the **Resources** section, go to the **List** tab.

2. Select the check box for the resource whose change history you want to view, and click the **View versions history** button in the upper part of the table.

   This opens the window with the version history of the resource.

3. Click the row of the version that you want to export and click the **Export** button in the lower part of the displayed window.

   You can only export a previous version of a resource. The **Export** button is not displayed when the current version of the resource is selected.

The resource version is saved in a JSON file on your computer in accordance with your browser settings.

Importing resources

In LogAnalyzer 3.4, we recommended using resources from the "[OOTB] KUMA 3.4 resources" package and resources published in the repository after the release of this package.

LogAnalyzer supports importing resources from an earlier version to a later version, for example from LogAnalyzer 3.4 to LogAnalyzer 4.0. Importing from a later version to an earlier version is possible but not recommended. Such a resource is not guaranteed to work.

To import resources:

1. In the **Resources** section, click **Resource import**.

   The **Resource import**window opens.

2. In the **Tenant** drop-down list, select the tenant to assign the imported resources to.

3. In the **Import source** drop-down list, select one of the following options:

   - **File**

     If you select this option, enter the password and click the **Import** button.

   - **Repository**

     If you select this option, a list of packages available for import is displayed. We recommend you to ensure that the repository update date is relatively recent and configure automatic updates if necessary.

     You can select one or more packages to import and click the **Import** button. The dependent resources of the Shared tenant are imported into the Shared tenant, the rest of the resources are imported into the selected tenant. You do not need special rights for the Shared tenant; you must only have the right to import in the selected tenant.

     Imported resources marked as **This resource is a part of the package. You can delete it, but it is impossible to edit.** can only be deleted. To rename, edit or move an imported resource, make a copy of the resource using the **Duplicate** button and perform the desired actions with the copy of the resource. When importing future versions of the package, the duplicate is not updated because it is a separate object.

     You can edit Imported resources in the `Integration` folder. Such resources are marked as **This resource is a part of the package** To a package resource in the `Integration` folder, you can add a dictionary of the `Table type`; adding other resources is not allowed. When importing subsequent versions of the package, the edited resource will not be replaced with the corresponding resource from the package, which allows you to keep the changes you made.

4. Resolve the conflicts between the resources imported from the file and the existing resources if they occur. Read more about resource conflicts below.

   a. If the name, type, and guid of an imported resource fully match the name, type, and guid of an existing resource, the **Conflicts** window opens with the table displaying the type and the name of the conflicting resources. Resolve displayed conflicts:

      - To replace the existing resource with a new one, click **Replace**.

        To replace all conflicting resources, click **Replace all**.

      - To leave the existing resource, click **Skip**.

        For dependent resources, that is, resources that are associated with other resources, the **Skip** option is not available; you can only **Replace** dependent resources.

        To keep all existing resources, click **Skip all**.

   b. Click the **Resolve** button.

   The resources are imported to LogAnalyzer. The Secret resources are imported blank.

Importing resources that use the extended event schema

If you import a normalizer that uses one or more fields of the extended event schema, LogAnalyzer automatically creates an extended schema field that is used in the normalizer.

If you import other types of resources that use fields of the extended event schema in their logic, the resources are imported successfully. To make sure the imported resources work as intended, you need to create the corresponding extended schema fields in the **Settings** → **Extended extended event schema fields** section or import a normalizer that uses the required fields.

If a normalizer that uses an extended event schema field is imported into LogAnalyzer and the same field already exists in LogAnalyzer, the previously created field is used.

If a normalizer is imported into LogAnalyzer that uses an extended event schema field that does not meet the LogAnalyzer requirements, the import is completed, but the extended event schema field is created with the **Disabled** status and you cannot use this field in other normalizers and resources. An extended event schema field runs afoul of requirements if, for example, its name contains special characters or spaces. If you want to use such a field that does not meet the requirements, you need to fix its problems (for example, by renaming it) and then enable the field.

About conflict resolving

When resources are imported into LogAnalyzer from a file, they are compared with existing resources; the following parameters are compared:

- Name and kind. If an imported resource's name and kind parameters match those of the existing one, the imported resource's name is automatically changed.
- ID. If identifiers of two resources match, a conflict appears that must be resolved by the user. This could happen when you import resources to the same LogAnalyzer server from which they were exported.

When resolving a conflict you can choose either to *replace existing resource* with the imported one or to *keep exiting resource*, skipping the imported one.

In this case, if a conflict occurs, the imported resource is added as a new version of the existing resource. A *resources imported* comment is added to this version.

Some resources are linked: for example, in some types of connectors, the connector secret must be specified. The secrets are also imported if they are linked to a connector. Such linked resources are exported and imported together.

Special considerations of import:

1. Resources are imported to the selected tenant.
2. If a linked resource was in the Shared tenant, it ends up in the Shared tenant when imported.
3. In the **Conflicts** window, the **Parent object** column always displays the top-most parent resource among those that were selected during import.
4. If a conflict occurs during import and you choose to replace existing resource with a new one, it would mean that all the other resources linked to the one being replaced are automatically replaced with the imported resources.

Known errors:

1. The linked resource ends up in the tenant specified during the import, and not in the Shared tenant, as indicated in the **Conflicts** window, under the following conditions:

   a. The linked resource is initially in the Shared tenant.

   b. In the **Conflicts** window, you select **Skip** for all parent objects of the linked resource from the Shared tenant.

   c. You leave the linked resource from the Shared tenant for replacement.

2. After importing, the categories do not have a tenant specified in the filter under the following conditions:

   a. The filter contains linked asset categories from different tenants.

   b. Asset category names are the same.

   c. You are importing this filter with linked asset categories to a new server.

3. In Tenant 1, the name of the asset category is duplicated under the following conditions:

   a. in Tenant 1, you have a filter with linked asset categories from Tenant 1 and the Shared tenant.

   b. The names of the linked asset categories are the same.

   c. You are importing such a filter from Tenant 1 to the Shared tenant.

4. You cannot import conflicting resources into the same tenant.

   The *Import conflicting resources in the same tenant is not allowed* error means that the imported package contains conflicting resources from different tenants and cannot be imported into the Shared tenant.

   Solution: Select a tenant other than Shared to import the package. In this case, during the import, resources originally located in the Shared tenant are imported into the Shared tenant, and resources from the other tenant are imported into the tenant selected during import.

5. Only the general administrator can import categories into the Shared tenant.

   The *Only general admin is allowed to import categories into shared tenant* error means that the imported package contains resources with linked shared asset categories. The categories or resources with linked shared asset categories are displayed in the LogAnalyzer Core log. Path to the Core log:

   /opt/kaspersky/kuma/core/log/core

   Solution. Choose one of the following options:

   - Do not import resources to which shared categories are linked: clear the check boxes next to the relevant resources.
   - Perform the import under a General administrator account.

6. Only the general administrator can import resources into the Shared tenant.

   The *Only general admin is allowed to import resources into shared tenant* error means that the imported package contains resources with linked shared resources. The resources with linked shared resources are displayed in the LogAnalyzer Core log. Path to the Core log:

   /opt/kaspersky/kuma/core/log/core

   Solution. Choose one of the following options:

   - Do not import resources that have linked resources from the Shared tenant, and the shared resources themselves: clear the check boxes next to the relevant resources.
   - Perform the import under a General administrator account.

Resource search

You can search resources by name or tags. You can also use full-text search to search for resources by all of their fields. Searching by filter finds all resources that use the filter.

The search is carried out only on the latest version of the resources.

To find resources:

1. In the LogAnalyzer web interface, in the **Resources** section, select the type of resources that you need.

   This opens a window opens with a table of available resources of this type.

   If you want to view all resources, in the **Resources** section, go to the **List** tab.

2. If necessary, to toggle the search mode, click the table icon in the upper part of the resource table.

   You can search by name, tags, or full-text search in all fields of resources. By default, search by name and tags is active.

   You can tell which search mode is currently enabled by the default text that is displayed in the search field.

3. In the **Search** field, start typing the search text.

   The search is initiated as you type characters in the field and is case-insensitive. The table displays only those resources that satisfy the search conditions, and the number of such resources is displayed in the lower part of the table.

   For full-text search, the results are sorted in the descending order of the number of words from the search string found in the fields of a resource. LogAnalyzer searches the JSON of a resource; if another resource is specified in it, it searches the specified resource too. If the resource refers to other resources, LogAnalyzer also traverses these resources and searches in their content.

4. If you want to reset the search result, clear the **Search** field or click the × icon.

Tag management

To help manage resources, the LogAnalyzer web interface lets you add tags to resources. You can use tags to search for a component, as well as manage tags, link or unlink tags.

You cannot add tags to resources that are created from the interface of other resources. Tags can be added only from the resource's own card. You also cannot add tags to a resource that is not editable.

Tag management

The list of tags is displayed in the **Settings** → **Tags** section and is displayed as a table with the following columns: **Name**, **Tenant**, **Used in resources**.

In the **Tags** table, you can:

- Sort tags by **Name**, **Used in resources** fields.
- Filter by values of the **Tenant** field.
- Find a tag by the **Name** field.
- Go to the list of resources that have the selected tag.

Adding a tag

To add a tag:

1. Go to the **Resources** section and select a resource.
2. In the panel above the table, click **Add**.
3. In the **Tags** field of the selected resource, add a new tag, or select a tag from the list.
4. Click **Create**.

The new tag is added.

You can also add tags from existing ones.

When adding a tag, keep in mind the following special considerations:

- You can add multiple tags.
- A tag can contain characters of various alphabets (for example, Cyrillic, Latin, or Greek characters), numerals, underscores, and spaces.
- A tag may not contain any special characters other than the underscore and the space.
- You can enter the tag in uppercase or lowercase, but after saving, the tag is always displayed in lowercase.
- The tag inherits the tenant of the resource in which it is used.
- A tag is part of a resource and exists as long as the resource exists in which the tag was created or is used.
- Tags are unique within a tenant.
- Tags are imported or exported together with the resource as part of the resource.

Searching by tags

In the **Resources** section, you can search for resources:

- By tags
- By resource name

The search is performed across all resource types and services.

The search results display a list of resources and services.

To find resources by tags:

1. Go to the **Resources** section and select a resource.
2. In the table of the resource, select the **Tags** column.
3. In the **Search** field that is displayed, enter or select a tag name.

   Searching by tag in the resource itself and in the list of resources returns full as well as partial matches of the entered text with the name of the tag.

Displays a list of resources if the specified tag is used in those resources.

In the list of resources, you can:

- Sort the list by name and type of resource or service.
- Filter resources or services by resource or service type, or by tag.
- Link or unlink tags.

Displaying tags while adding dependent resources

While adding a dependent resource, the tags of this resource are displayed in a drop-down list under **Filter settings**.

Resource tags that to not fit a single line are hidden. A special indicator lets you see that hidden tags exist.

The drop-down list can be filtered by resource name and by resource tags.

Linking and unlinking tags

To link tags to a resource or unlink tags from a resource:

1. Go to the **Resources** section.
2. Select the **List** tab.
3. In the **Name** column, select the check boxes next to the relevant resources.
4. In the panel above the list, select the **Tags** tab.
5. Click **Link** or **Unlink**, select the tags that you need or create a new tag.

   The selected tags are linked to or unlinked from the resources.

   You can also link or unlink tags in the card of the resource.

Resource usage tracing

For stable operation of LogAnalyzer, it is important to understand how some resources affect the performance of other resources, what connections exist between resources and other LogAnalyzer objects. You can visualize these interdependencies on an interactive graph in the LogAnalyzer web interface.

Displaying the links of a resource on a graph

To display the relations of the selected resource:

1. In the LogAnalyzer web console, in the **Resources** section, select a resource type.

   A list of resources of the selected type is displayed.
2. Select the resource that you need.
3. In the toolbar or in the menu (⋮), click the **Show dependencies** button.
4. Click the **Show dependencies** button.

   This opens the a window with the dependency graph of the selected resource. If you do not have rights to view the resource, it is marked in the graph with the (inaccessible resource) icon. If necessary, you can close the graph window to go back to the list of resources.

Resource dependency graph

The graph displays all relations that are formed based on the universal unique identifier (UUID) of resources used in the configuration of the resource selected for display, as well as relations of resources that have the UUID of the selected resource in their configuration. Downward links, that is, resources referenced (used) by the selected resource, are displayed down to the last level, while for upward links, that is, resources that reference the selected resource, only one level is displayed.

On the graph, you can view the dependencies of the following resources:

- Aggregation rules
- Enrichment rules
- Normalizers
- Connectors
- Destinations
- Filters
- Dashboards

  A peculiarity of link display: dashboards do not have an "upward" link because dashboards are not part of other resources.

- Report templates

  A peculiarity of link display: report templates do not have an "upward" link because report templates are not part of other resources.

- Notification templates
- dictionaries (available only in the commercial version of LogAnalyzer)
- Proxy servers
- Secrets
- Collectors

  Note that if a collector was initially selected for displaying links, "upward" links are not displayed.

- Storages
- Agents (autoagents)

  Note that if an agent is selected for displaying links, the collector is displayed with the **linked** relation type only if the collector is running as a service and in the collector is correctly (fqdn+port) specified as the destination of the agent.

- Event routers

  Note that if an event router was initially selected for displaying links, "upward" links are not displayed.

- Integrations

  The name of the integration corresponds to the name of the tab in the **Integrations** section.

-

Resource group

  The number before the parentheses indicates the number of resources from the group displayed in the graph; the number in parentheses indicates the total number of resources in the group.

- Inaccessible resource (if you do not have the rights to view it).

Clicking a resource node lets you view the following information about the resource:

- **Name**

  Contains a link to the resource; clicking the link opens the resource in a separate tab; this does not close the graph window.

- **Type**

- **Path**

  Resource path without a link.

- **Tags**

- **Tenant**

- **Package name**

You can open the context menu of the resource and perform the following actions:

- Show relations of resource

  The dependencies of the selected resource are displayed.

- Hide resource on graph

  The selected resource is hidden. Resources at the lower level that the selected resource references are marked with "*" as having hidden links. Resources that refer to a hidden resource are marked with the icon as having hidden links. In this case, the graph becomes broken.

- Hide "downward" relations of resource on graph

  The selected resource remains. Only those lower-level resources that do not have any links remaining on the first higher level on the graph are hidden. Resources referenced by resources of the first (hidden) level are marked with "*" as having hidden links.

- Hide all resources of this type on graph

  All resources of the selected type are hidden. This operation is applied to each resource of the selected type.

- Update resource relations

  You can update the resource state if the resource was edited by another user while you were managing the graph. Only changes of visible links are displayed.

- Group

  If there is no group node on the screen: the group node appears on the screen, resources of the same type as the selected resource and resources that refer to the same resource are hidden. The edges are redrawn from the group. The **Group** button is available only when more than 10 links to resource of the same type exist.

  If there is a group node on the screen: the resource is hidden and added to the group, the edges are redrawn from the group.

Several types of relations are displayed on the graph:

- Solid line without a caption.

  Represents a direct link by UUID, including the use of secrets and proxies in integrations.

- Dotted line captioned **linked**.

  Represents a link by URL, for example, of a destination with a collector, or of a destination with a storage.

Resources created inline are shown on the graph as a dotted line with the **linked** type.

We do not recommend building large dependency graphs. We recommend limiting the number of nodes to 100 nodes.

When you open the graph, the resource selected for display is highlighted with a blinking circle for some time to set it apart graphically from other resources and draw attention to it.

You can look at the map of the graph to get an idea of where you are on the graph. You can use the selector and move it to display the necessary part of the graph.

By clicking the **Arrange** button, you can improve the display of resources on the graph.

If you select **Show links**, the focus on the graph does not change, and the resources are displayed so that you do not have to return to where you started.

When you select a group node in the graph, a sidebar is displayed, in which you can hide or show the resources that are part of the group. To do so, select the check box next to the relevant resource and click the **Show on graph** or **Hide on graph** button.

The graph retains its state if you displayed something on the graph, then switched to editing a resource, and then reopened the graph tab.

The previously displayed resources on the graph remain in their places when new resources are added to the graph.

When you close the graph, all changes are discarded.

After the resource links are drawn on the graph, you can search for a node:

- By name

- By tag

- By path
- By package

Nodes, including groups that match the selection criterion, are highlighted with a yellow circle.

You can filter the graph by resource type:

- Hide or show resources of a certain type.
- Hide resources of multiple types. Display all types of resources.

With the filter window closed, you can tell the selected filters by the indicator, a red dot in the toolbar.

Your actions when managing the graph (the last 50 actions) are saved in memory; you can undo changes by pressing **Ctrl/Command+Z**.

You can save the displayed graph can be saved to an SVG file. The visible part of the graph is saved in the file.

Resource versioning

LogAnalyzer stores the change history of resources in the form of versions. A resource version is created automatically when you create a new resource or save changes made to the settings of an existing resource.

> The change history is not available for the **Dictionaries** resource. To save the history of dictionaries, you can export data.

Resource versions are retained for the duration specified in the **Settings** section. When the age of a resource version reaches the specified value, the version is automatically deleted.

You can view the change history of LogAnalyzer resources, compare versions, and restore a previous version of a resource, for example, if it fails and you need to recover it.

To view the change history of a resource:

1. In the LogAnalyzer web interface, in the **Resources** section, select the type of resources that you need.

   This opens a window opens with a table of available resources of this type.

   If you want to view all resources, in the **Resources** section, go to the **List** tab.

2. Select the check box for the resource whose change history you want to view, and click the **View versions history** button in the upper part of the table.

This opens a window with a table of saved versions of the selected resource. New resources have only one version, the current version.

For each version, the table displays the following information:

- **Version** is the serial number of the resource version. When you save changes to the resource and create a new version, the serial number is increased by 1.

  The version with the highest number and the most recent publication date reflects the current state of the resource. Version 1 reflects the state of the resource at the moment when it was created.

- **Published** is the date and time when the resource version was created.

- **Author** is the login of the user that saved the changes to the resource.

  If the changes were made by the system or by the migration script, the displayed value is **system**.

- **Comment** is a text comment added by the author when saving changes, or a system comment describing the changes made.

- **Retention period** is the number of days and the date after which the resource version will be deleted.

  If necessary, you can configure the retention period for resource versions.

- **Actions** is the button that restores the resource version.

You can sort the table of resource versions by the **Version**, **Published**, and **Author** columns by clicking the header and selecting **Ascending** or **Descending**. You can also display only changes made by a specific author or authors in the table by clicking the heading of the **Author** column and selecting the authors as needed.

If you want to view the status of a resource in a specific version, click that version in the table. This opens a window with the resource of the selected version, in which you can:

- View the settings specified in that version of the resource.
- Restore this version of the resource by clicking the **Restore** button.
- Export this version of the resource to a JSON file by clicking the **Export** button.

Comparing resource versions

You can compare any two versions of a resource, for example, if you need to track changes.

To compare versions of a resource:

1. In the LogAnalyzer web interface, in the **Resources** section, select the type of resources that you need.

   This opens a window opens with a table of available resources of this type.

   If you want to view all resources, in the **Resources** section, go to the **List** tab.

2. Select the check box next to a resource and click the **View versions history** button in the upper part of the table.

   This opens the window with the version history of the resource.

3. Select the check boxes next to the two versions of the resource that you want to compare and click the **Compare** button in the upper part of the table.

This opens the resource version comparison window. Resource fields are displayed as a list or in JSON format. Differences between the two versions are highlighted. You can select other versions to compare using the drop-down lists above the resource fields.

Restoring a resource version

You can restore a previous version of a resource, for example, if you need to recover the resource in case of mistakes made when making changes.

> Versions of automatically generated agents cannot be restored separately because they are created when the parent collector is modified. If you want to restore a version of an automatically generated agent, you need to restore the corresponding version of the parent collector.

To restore a previous version of a resource:

1. In the LogAnalyzer web interface, in the **Resources** section, select the type of resources that you need.

   This opens a window opens with a table of available resources of this type.

   If you want to view all resources, in the **Resources** section, go to the **List** tab.

2. Select the check box next to a resource and click the **View versions history** button in the upper part of the table.

   This opens the window with the version history of the resource.

3. In the row of the relevant version, in the **Action** column, click the **Restore** button.

   You can also restore a version by clicking the row of this version and clicking the **Restore** button in the lower part of the window.

   > You can restore only previous versions of a resource; for the current version, the **Restore** button is not available.

   > If the structure of the resource has changed after a LogAnalyzer update, restoring its saved versions may not be possible.

4. Confirm the action and, if necessary, add a comment. If you do not add a comment, the *Restored from v. <number of the restored version>* comment is automatically added to the version.

The resource version is restored as a new version and become the current version.

If the resource for which you restored the version is added to the active service, this also changes the state of the service. You must restart the service to apply the resource change.

Configuring the retention period for resource versions

You can change the retention period of resource versions in the LogAnalyzer web interface in the **Settings** → **Other** → **Common** section by changing the **Resource history retention period, days**.

The default setting is 30 days. If you want to keep all versions of resources without time limits, specify 0 (store indefinitely).

> Only a user with the General administrator role can view and manage the retention period of resource versions.

The retention period of resource versions is checked daily, and versions of resources that have been stored in LogAnalyzer for longer than the specified period are automatically deleted. In the **Task manager** section, the *Clearing resource change history* task is created to check the storage duration of resource versions and delete old versions. This task also runs after a restart of the Core component.

You can check the time remaining until a resource version is deleted in the table of versions, in the **Retention period** column.

Destinations

Destinations define network settings for sending normalized events. Collectors use destinations to describe where to send processed events. Typically, a storage acts as the destination.

You can specify destination point settings on the **Basic settings** and **Advanced settings** tabs. The available settings depend on the selected type of destination.

Destinations can have the following types:

- internal (available only in the commercial version of LogAnalyzer) is used to receive data from LogAnalyzer services via the 'internal' protocol.
- storage for sending data to storage.
- eventRouter (available only in the commercial version of LogAnalyzer) is used to send events to the event router.

The LogAnalyzer distribution kit includes the predefined **[OOTB] Storage** destination that sends events to storage.

Destination, internal type

> This functionality is only available in the commercial version of LogAnalyzer.

Destinations of the **internal** type are used for receiving data from LogAnalyzer services using the 'internal' protocol. You can transfer the following data using the internal protocol:

- Internal data, such as event routes.
- File attributes. If while creating the collector at the Transport step of the Installation Wizard, you specified a connector of the file, 1c-xml, or 1c-log type, at the Event parsing step, in the **Mapping** table, you can pass the name of the file being processed by the collector or the path to the file in the LogAnalyzer event field. To do this, in the **Source** column, specify one of the following values:
  - `$kuma_fileSourceName` to pass the name of the file being processed by the collector in the LogAnalyzer event field.
  - `$kuma_fileSourcePath` to pass the path to the file being processed by the collector in the LogAnalyzer event field.

  When you use a **file**, **1c-xml**, or **1c-log** connector, the new variables in the normalizer will only work with **destinations of the internal type**.
- Events to the event router. The event router can only receive events over the 'internal' protocol, therefore you can only use **internal** destinations when sending events to the event router.

Settings for a destination of the **internal** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
State	This toggle switch enables sending events to the destination. This toggle switch is turned on by default.
Type	Destination type: **internal**. Required setting.
URL	URL that you want to connect to. The following URL formats are supported: • `< host name >:< port number >` • `< IPv4 address >:< port number >` • `< IPv6 address >:< port number >` You can specify IPv6 addresses in the following format: `[< IPv6 address >%< interface >:< port number >]`, for example, `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`. You can add multiple URLs or remove an URL. LogAnalyzer does not allow saving a resource or service if the URL field contains a tab or space character. To add an URL, click the **+ Add** button. To remove an URL, click the delete ✕ icon next to it. Required setting.
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Buffer size	Buffer size in bytes for accumulating events in the RAM of the server before sending them for further processing or storage. The value must be a positive integer. Default buffer size: 16777216 (16 MB), displayed as 0 in the interface.
Buffer flush interval	Interval (in seconds) for sending events to the destination. The default value is 1 second.
Disk buffer size limit	Size of the disk buffer in bytes. Default value: 10 GB, displayed as 0 in the interface.
Handlers	Number of handlers that the service can run simultaneously to process response rules in parallel. The value must be a positive integer up to 999.
Output format	Format in which events are sent to the destination: • **JSON**. • **CEF**. If this value is selected, the transmitted events contain the CEF header and only non-empty fields.
Proxy server	The proxy server for the destination. You can select an existing proxy server or create a new proxy server. To create a new proxy server, select **Create new**. If you want to edit the settings of an existing proxy server, click the pencil icon next to it.
URL selection policy	Method of determining URLs to which events must be sent first if you added multiple URLs in the **URL** field on the **Basic settings** tab: • **Any** means events are sent to a randomly selected available URL as long as the URL accepts events. If the URL becomes unavailable, events are sent to another randomly selected available URL. This value is selected by default. • **Prefer first** means events are sent to the first added URL. If the URL becomes unavailable, events are sent to the next added available URL. If the first added URL becomes available again, events are sent to the first added URL again. • **Round robin** means events are evenly balanced among the available URLs. This method does not guarantee that events are evenly balanced among the URLs because the buffer may overflow or events may be sent to the destination. You can specify the buffer size in bytes in the **Buffer size limit** field; you can also specify the interval in seconds for sending events to the destination in the **Buffer flush interval** field.
Health check timeout	Interval, in seconds, for checking the health of the destination.
Disk buffer disabled	This toggle switch that enables the disk buffer. This toggle switch is turned on by default. The disk buffer is used if the collector cannot send normalized events to the destination. You can specify the size of the disk buffer in the **Disk buffer size limit** field. If the disk buffer runs out of free space, new normalized events will overwrite old normalized events, starting with the oldest.
Timeout	The time, in seconds, for which the destination waits for a response from another service or component.

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Filter	Filter for determining which events must be processed by the resource. You can select an existing filter or create a new filter. To create a new filter, select **Create new**. If you want to edit the settings of an existing filter, click the pencil icon next to it. **How to create a filter?**

Setting	Description
	To create a filter: 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

Setting	Description
	4. Under **Conditions**, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, fields of additional parameters for identifying the value to be passed to the filter may be displayed. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **Operator** drop-down list, select an operator.

Setting	Description
	Filter operators • **=** means the left operand equals the right operand. • **<** means the left operand is less than the right operand. • **<=** means the left operand is less than or equal to the right operand. • **>** means the left operand is greater than the right operand. • **>=** means the left operand is greater than or equal to the right operand. • **inSubnet** means the left operand (IP address) is in the subnet of the right operand (subnet). • **contains** means the left operand contains values of the right operand. • **startsWith** means the left operand starts with one of the values of the right operand. • **endsWith** means the left operand ends with one of the values of the right operand. • **match** means the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit** checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability** checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • The **inActiveList** operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary** checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory** means the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup** means the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

Setting	Description
	• The **TIDetect** operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage. • **inContextTable** checks for the presence of the entry in the specified context table. • **intersect** checks for the presence in the left operand of the list items specified in the right operand. d. If you want the operator to be case-insensitive, select the **do not match case** check box. This check box does not apply to the **InSubnet**, **InActiveList**, **InCategory**, or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, from the **If** drop-down list, select **If not**. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Destination, storage type

Destinations of the **storage** type are used to transfer data to storage. Settings for a destination of the **storage** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
State	This toggle switch enables sending events to the destination. This toggle switch is turned on by default.
Type	Destination type: **storage**. Required setting.

Setting	Description
URL	URL that you want to connect to. The following URL formats are supported: • `<host name>:<port number>` • `<IPv4 address>:<port number>` • `<IPv6 address>:<port number>` You can specify IPv6 addresses in the following format: `[<IPv6 address>%<interface>:<port number>`, for example, `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`. You can add multiple URLs or remove an URL. LogAnalyzer does not allow saving a resource or service if the URL field contains a tab or space character. To add an URL, click the **+ Add** button. To remove an URL, click the delete ✕ icon next to it. Required setting.
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Buffer size	Buffer size in bytes for accumulating events in the RAM of the server before sending them for further processing or storage. The value must be a positive integer. Default buffer size: 67,108,864 (64 MB), displayed as 0 in the interface.
Buffer flush interval	Interval (in seconds) for sending events to the destination. The default value is 1 second.
Disk buffer size limit	Size of the disk buffer in bytes. Default value: 10 GB, displayed as 0 in the interface.
Handlers	Number of handlers that the service can run simultaneously to process response rules in parallel. The value must be a positive integer up to 999.
Proxy server	The proxy server for the destination. You can select an existing proxy server or create a new proxy server. To create a new proxy server, select **Create new**. If you want to edit the settings of an existing proxy server, click the pencil icon next to it.
URL selection policy	Method of determining URLs to which events must be sent first if you added multiple URLs in the **URL** field on the **Basic settings** tab: • **Any** means events are sent to a randomly selected available URL as long as the URL accepts events. If the URL becomes unavailable, events are sent to another randomly selected available URL. This value is selected by default. • **Prefer first** means events are sent to the first added URL. If the URL becomes unavailable, events are sent to the next added available URL. If the first added URL becomes available again, events are sent to the first added URL again. • **Round robin** means events are evenly balanced among the available URLs. This method does not guarantee that events are evenly balanced among the URLs because the buffer may overflow or events may be sent to the destination. You can specify the buffer size in bytes in the **Buffer size limit** field; you can also specify the interval in seconds for sending events to the destination in the **Buffer flush interval** field.
Health check timeout	Interval, in seconds, for checking the health of the destination.
Disk buffer disabled	This toggle switch that enables the disk buffer. This toggle switch is turned on by default. The disk buffer is used if the collector cannot send normalized events to the destination. You can specify the size of the disk buffer in the **Disk buffer size limit** field. If the disk buffer runs out of free space, new normalized events will overwrite old normalized events, starting with the oldest.
Timeout	The time, in seconds, for which the destination waits for a response from another service or component.

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Filter	Filter for determining which events must be processed by the resource. You can select an existing filter or create a new filter. To create a new filter, select **Create new**. If you want to edit the settings of an existing filter, click the pencil icon next to it. **How to create a filter?**

Setting	Description
	To create a filter: 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

Setting	Description
	4. Under **Conditions**, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, fields of additional parameters for identifying the value to be passed to the filter may be displayed. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **Operator** drop-down list, select an operator.

Setting	Description

Filter operators

- **=** means the left operand equals the right operand.
- **<** means the left operand is less than the right operand.
- **<=** means the left operand is less than or equal to the right operand.
- **>** means the left operand is greater than the right operand.
- **>=** means the left operand is greater than or equal to the right operand.
- **inSubnet** means the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains** means the left operand contains values of the right operand.
- **startsWith** means the left operand starts with one of the values of the right operand.
- **endsWith** means the left operand ends with one of the values of the right operand.
- **match** means the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit** checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
- **hasVulnerability** checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- The **inActiveList** operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary** checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory** means the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup** means the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

Setting	Description
	• The **TIDetect** operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage. • **inContextTable** checks for the presence of the entry in the specified context table. • **intersect** checks for the presence in the left operand of the list items specified in the right operand. d. If you want the operator to be case-insensitive, select the **do not match case** check box. This check box does not apply to the **InSubnet**, **InActiveList**, **InCategory**, or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, from the **If** drop-down list, select **If not**. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Destination, eventRouter type

This functionality is only available in the commercial version of LogAnalyzer.

Destinations of the **eventRouter** type are used for sending events to the event router. Settings for a destination of the **eventRouter** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
State	This toggle switch enables sending events to the destination. This toggle switch is turned on by default.
Type	Destination type: **eventRouter**. Required setting.

Setting	Description
URL	URL that you want to connect to. The following URL formats are supported: • `< host name >:< port number >` • `< IPv4 address >:< port number >` • `< IPv6 address >:< port number >` You can specify IPv6 addresses in the following format: `[< IPv6 address >%< interface >:< port number >`, for example, `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`. You can add multiple URLs or remove an URL. LogAnalyzer does not allow saving a resource or service if the URL field contains a tab or space character. To add an URL, click the **+ Add** button. To remove an URL, click the delete ✕ icon next to it. Required setting.
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Buffer size	Buffer size in bytes for accumulating events in the RAM of the server before sending them for further processing or storage. The value must be a positive integer. Default buffer size: 67,108,864 (64 MB), displayed as 0 in the interface.
Buffer flush interval	Interval (in seconds) for sending events to the destination. The default value is 1 second.
Disk buffer size limit	Size of the disk buffer in bytes. Default value: 10 GB, displayed as 0 in the interface.
Handlers	Number of handlers that the service can run simultaneously to process response rules in parallel. The value must be a positive integer up to 999.
Output format	Format in which events are sent to the destination: • **JSON**. • **CEF**. If this value is selected, the transmitted events contain the CEF header and only non-empty fields.
Proxy server	The proxy server for the destination. You can select an existing proxy server or create a new proxy server. To create a new proxy server, select **Create new**. If you want to edit the settings of an existing proxy server, click the pencil icon next to it.
URL selection policy	Method of determining URLs to which events must be sent first if you added multiple URLs in the **URL** field on the **Basic settings** tab: • **Any** means events are sent to a randomly selected available URL as long as the URL accepts events. If the URL becomes unavailable, events are sent to another randomly selected available URL. This value is selected by default. • **Prefer first** means events are sent to the first added URL. If the URL becomes unavailable, events are sent to the next added available URL. If the first added URL becomes available again, events are sent to the first added URL again. • **Round robin** means events are evenly balanced among the available URLs. This method does not guarantee that events are evenly balanced among the URLs because the buffer may overflow or events may be sent to the destination. You can specify the buffer size in bytes in the **Buffer size limit** field; you can also specify the interval in seconds for sending events to the destination in the **Buffer flush interval** field.
Health check timeout	Interval, in seconds, for checking the health of the destination.
Disk buffer disabled	This toggle switch that enables the disk buffer. This toggle switch is turned on by default. The disk buffer is used if the collector cannot send normalized events to the destination. You can specify the size of the disk buffer in the **Disk buffer size limit** field. If the disk buffer runs out of free space, new normalized events will overwrite old normalized events, starting with the oldest.
Timeout	The time, in seconds, for which the destination waits for a response from another service or component.

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Filter	Filter for determining which events must be processed by the resource. You can select an existing filter or create a new filter. To create a new filter, select **Create new**. If you want to edit the settings of an existing filter, click the pencil icon next to it. **How to create a filter?**

Setting	Description
	To create a filter: 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

Setting	Description
	4. Under **Conditions**, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, fields of additional parameters for identifying the value to be passed to the filter may be displayed. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **Operator** drop-down list, select an operator.

Setting	Description

Filter operators

- **=** means the left operand equals the right operand.
- **<** means the left operand is less than the right operand.
- **<=** means the left operand is less than or equal to the right operand.
- **>** means the left operand is greater than the right operand.
- **>=** means the left operand is greater than or equal to the right operand.
- **inSubnet** means the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains** means the left operand contains values of the right operand.
- **startsWith** means the left operand starts with one of the values of the right operand.
- **endsWith** means the left operand ends with one of the values of the right operand.
- **match** means the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit** checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
- **hasVulnerability** checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- The **inActiveList** operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary** checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory** means the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup** means the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

Setting	Description
	• The **TIDetect** operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage. • **inContextTable** checks for the presence of the entry in the specified context table. • **intersect** checks for the presence in the left operand of the list items specified in the right operand. d. If you want the operator to be case-insensitive, select the **do not match case** check box. This check box does not apply to the **InSubnet**, **InActiveList**, **InCategory**, or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, from the **If** drop-down list, select **If not**. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Normalizers

Normalizers are used for converting raw events that come from various sources in different formats to the LogAnalyzer event data model. Normalized events become available for processing by other LogAnalyzer resources and services.

A normalizer consists of the *main* event parsing rule and optional *additional event parsing rules*. By creating a main parsing rule and a set of additional parsing rules, you can implement complex event processing logic. Data is passed along the tree of parsing rules depending on the conditions specified in the

Extra normalization conditions setting. The sequence in which parsing rules are created is significant: the event is processed sequentially and the processing sequence is indicated by arrows.

The following event normalization options are now available:

- 1 collector — 1 normalizer.

  We recommend using this method if you have many events of the same type or many IP addresses from which events of the same type may originate. You can configure one collector with only one normalizer, which is optimal in terms of performance.

- 1 collector — multiple normalizers linked to the IP address.

  This method is available for collectors with a connector of UDP, TCP, or HTTP type. If a UDP, TCP, or HTTP connector is specified in the collector at the 'Transport' step, then at the 'Event parsing' step, you can specify multiple IP addresses on the 'Parsing settings' tab and choose the normalizer that you want to use for events coming from the specified addresses. The following types of normalizers are available: json, cef, regexp, syslog, csv, kv, xml. For normalizers of the Syslog and regexp types, you can specify extra normalization conditions depending on the value of the DeviceProcessName field.

A normalizer is created in several steps:

1. **Preparing to create a normalizer**

   A normalizer can be created in the LogAnalyzer web interface:

   - In the **Resources** → **Normalizers** section.
   - When creating a collector, at the **Event parsing** step.

   Then parsing rules must be created in the normalizer.

2. **Creating the main parsing rule for an event**

   The main parsing rule is created using the **Add event parsing** button. This opens the **Event parsing** window, where you can specify the settings of the main parsing rule:

   - Specify event parsing settings.
   - Specify event enrichment settings.

   The main parsing rule for an event is displayed in the normalizer as a dark circle. You can view or modify the settings of the main parsing rule by clicking this circle. When you hover the mouse over the circle, a plus sign is displayed. Click it to add the parsing rules.

   The name of the main parsing rule is used in LogAnalyzer as the normalizer name.

3. **Creating additional event parsing rules**

   Clicking the plus icon that is displayed when you hover over the circle or the block corresponding to the normalizer opens the **Additional event parsing** window where you can specify the settings of the additional parsing rule:

   - Specify the conditions for sending data to the new normalizer.
   - Specify event parsing settings.
   - Specify event enrichment settings.

   The additional event parsing rule is displayed in the normalizer as a dark block. The block displays the triggering conditions for the additional parsing rule, the name of the additional parsing rule, and the event field. When this event field is available, the data is passed to the normalizer. Click the block of the additional parsing rule to view or modify its settings.

If you hover over the additional normalizer, a plus button appears. You can use this button to create a new additional event parsing rule. To delete a normalizer, use the button with the trash icon.

Extra normalizers are applied from top to bottom until the first condition matches.

Conditions for processing an event by extra normalizers

If the event matches the conditions to be processed by the extra normalizer, the following may happen:

- If the event is processed by an extra normalizer:

  1. After processing, the event is sent to be processed to child extra normalizers, if any are configured.

  2. The processed event is returned to the main normalizer and passed down the chain.

- If the event is not processed, an error occurs, and **Keep raw event** is set to **Only errors** for the collector:

1. The event is returned to the main normalizer.

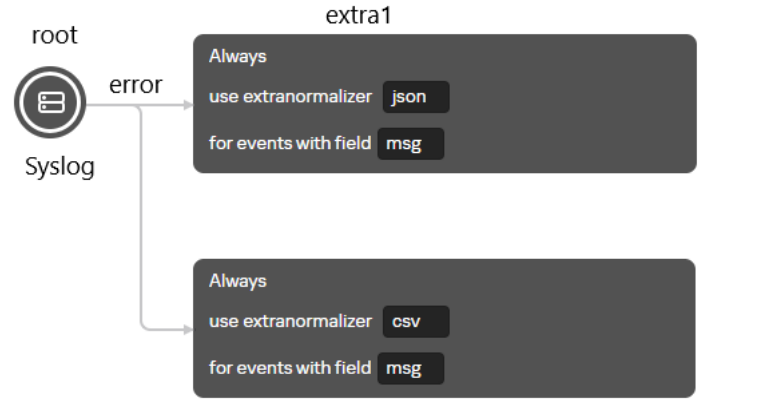
2. LogAnalyzer logs an error in the collector log.

3. The raw event is written to the **Raw** field.

Example 1

Prerequisite:

A collector has been created with a normalizer that has the **Keep original event** setting set to **Only errors**.



Example of error occurrence

Result

The event goes to the `extra 1 JSON` normalizer only if the event matches the selection condition.

The `extra1 JSON` extra normalizer fails to process the event and an error occurs (see the figure above). Processing ends, an error is logged in the collector log, and the raw event is written to the **Raw** field.

Example 2

Prerequisite:

A collector has been created with a normalizer that has the **Keep original event** setting set to **Only errors**.

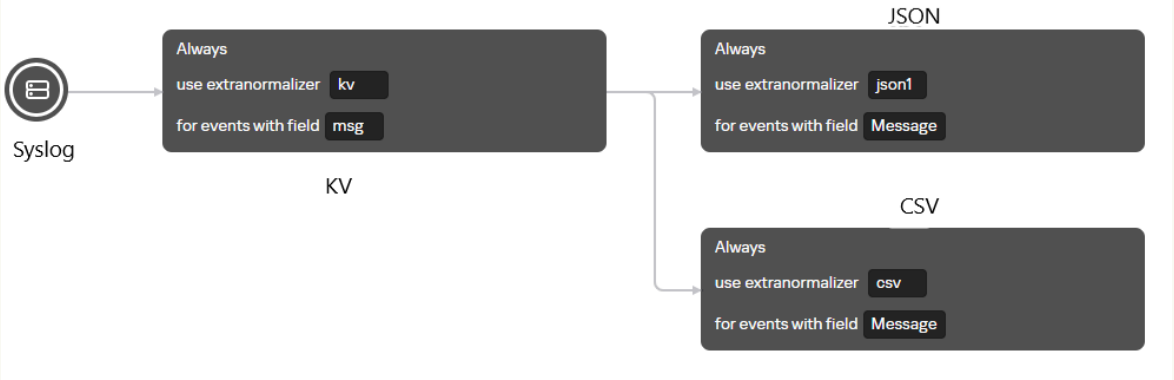


Diagram of event normalization

Sent event:

```
<1>May 31 11:55:21 srv-06 for_secondlvl_grp1[786]: Message=|test1|test2|;
```

Diagram of error occurrence

Result

An error occurred at the second level of `json1` extra normalization, processing is terminated (see the figure above).

An error is logged in the collector log, and the raw event is written to the **Raw** field.

Example 3

Prerequisite:

A collector has been created with a normalizer that has the **Keep original event** setting set to **Only errors**.

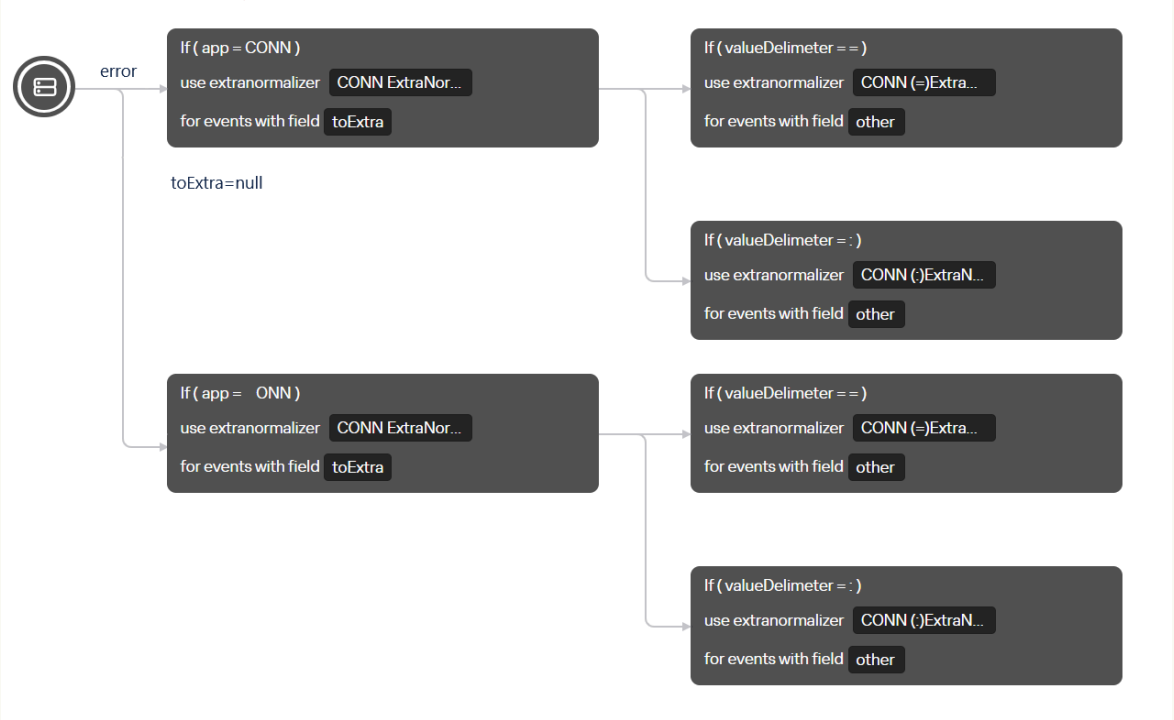


Diagram of error occurrence

Result

The level 1 extra normalizer fails to process the event and an error occurs (see the figure above). The branch of this extra normalizer is terminated and level 2 child extra normalizers are not applied.

Event processing fails, the event is returned to the main normalizer, no error is logged in the collector log, and the raw event is not written to the **Raw** field.

- If the event is not processed, the field passed to the extra normalizer is empty, and **Keep raw event** is set to **Only errors** for the collector:

  1. The event is returned to the main normalizer.
  2. No error is logged in the collector log.
  3. The raw event is not written to the **Raw** field.

  Example 4

Prerequisite:

A collector has been created with a normalizer that has the **Keep original event** setting set to **Only errors**.

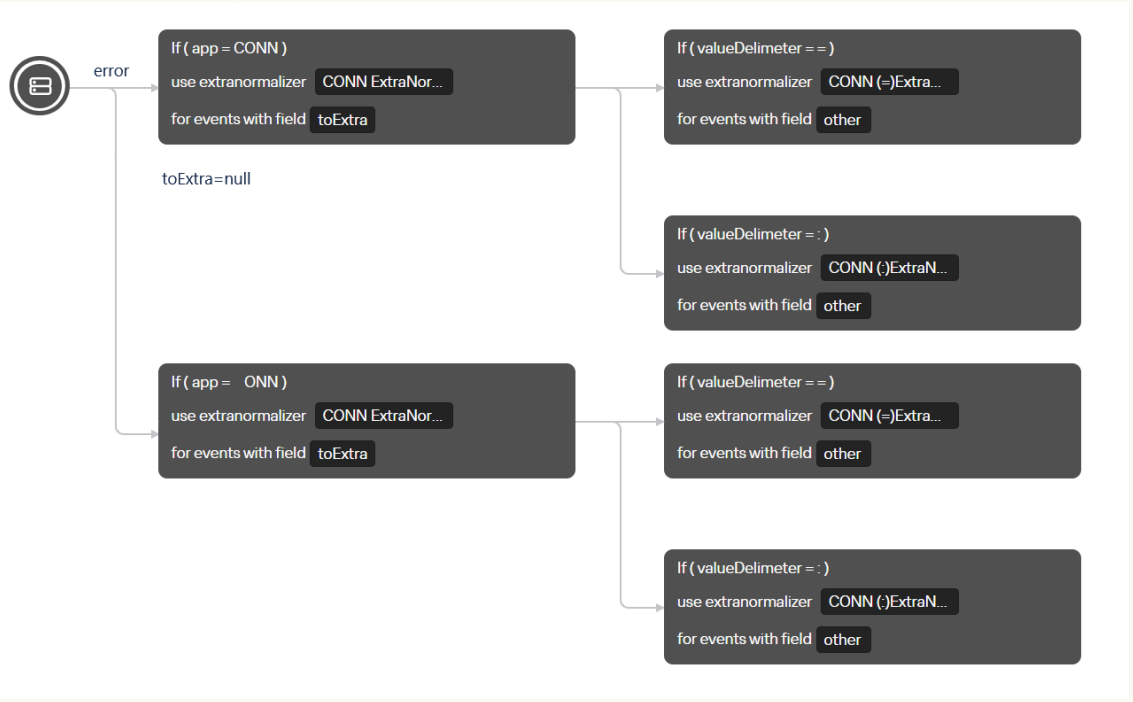


Diagram of error occurrence

Result

The field passed to extra normalizer 1 is empty (see the figure above).

Event processing fails, the event is returned to the main normalizer, no error is logged in the collector log, and the raw event is not written to the **Raw** field.

- If the event does not match the conditions of any extra normalizer, and **Keep raw event** is set to **Only errors** for the collector:

  1. The raw event is written to the **Raw** field.

  2. No error is logged in the collector log.

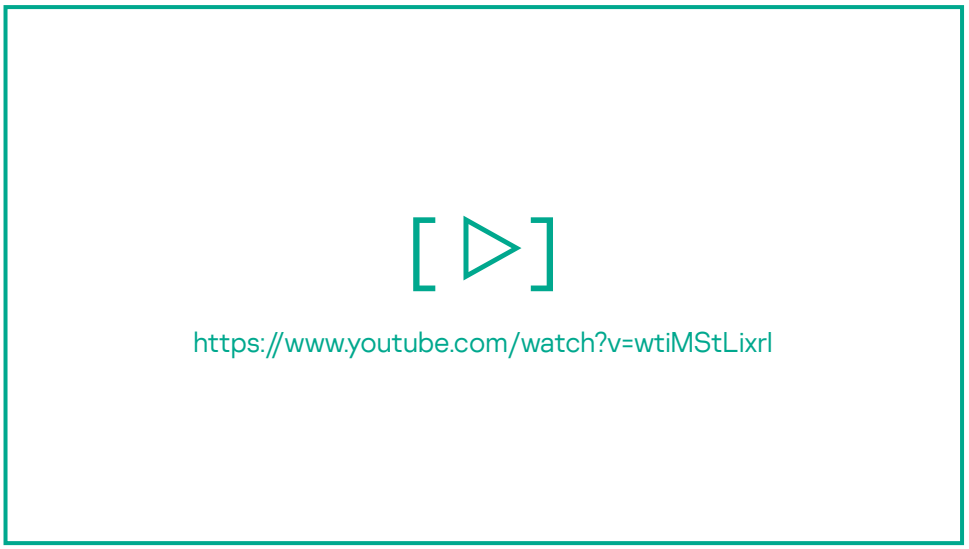
4 **Completing the creation of the normalizer**

To finish the creation of the normalizer, click **Save**.

In the upper right corner, in the search field, you can search for additional parsing rules by name.

For normalizer resources, you can enable the display of control characters in all input fields except the **Description** field.

If, when changing the settings of a collector resource set, you change or delete conversions in a normalizer connected to it, the edits will not be saved, and the normalizer itself may be corrupted. If you need to modify conversions in a normalizer that is already part of a service, the changes must be made directly to the normalizer under **Resources** → **Normalizers** in the web interface.



https://www.youtube.com/watch?v=wtiMStLixrI

Event parsing settings

You can configure the rules for converting incoming events to the LogAnalyzer format when creating event parsing rules in the normalizer settings window, on the **Normalization scheme** tab. Available event parsing settings are listed in the table below.

When normalizing events, you can use extended event schema fields in addition to standard LogAnalyzer event schema fields.

Available event parsing settings

Setting	Description

Parsing method	The type of incoming events. Depending on the selected parsing method, you can use the predefined event field matching rules or define your own rules. When you select some parsing methods, additional settings may become available they you must specify. Available parsing methods: • **json** This parsing method is used to process JSON data where each object, including its nested objects, occupies a single line in a file. When processing files with hierarchically structured data, you can reference the fields of nested objects using the dot notation. For example, the `username` parameter from the string `"user": {"username": "system: node: example-01"}` can be accessed by using the `user.username` query. Files are processed line by line. Multi-line objects with nested structures may be normalized incorrectly. In complex normalization schemes where additional normalizers are used, all nested objects are processed at the first normalization level, except for cases when the extra normalization conditions are not specified and, therefore, the event being processed is passed to the extra normalizer in its entirety. You can use `\n` and `\r\n` as newline characters. Strings must be UTF-8 encoded. If you want to send the raw event for advanced normalization, at each nesting level in the **Advanced event parsing** window, in the **Use raw event** drop-down list, select **Yes**. • **cef** This parsing method is used to process CEF data. If you select this parsing method, you can use the predefined rules for converting events to the LogAnalyzer format by clicking **Apply default mapping**. • **regexp**

This parsing method is used to create custom rules for processing data in a format using regular expressions.

You must add a regular expression (RE2 syntax) with named capturing groups to the field under **Normalization**. The name of the capturing group and its value are considered the field and value of the raw event that can be converted to an event field in LogAnalyzer format.

To add event handling rules:

1. If necessary, copy an example of the data you want to process to the **Event examples** field. We recommend completing this step.

2. In the field under **Normalization**, add a regular expression with named capture groups following the RE2 syntax, for example `"(?P<name>regexp)"`. The regular expression added to the field under **Normalization** must exactly match the event. When designing the regular expression, we recommend using special characters that match the starting and ending positions of the text: `^`, `$`.

   You can add multiple regular expressions or remove regular expressions. To add a regular expression, click **Add regular expression**. To remove a regular expression, click the delete icon ✕ next to it.

3. Click the **Copy field names to the mapping table** button.

   Capture group names are displayed in the **KUMA field** column of the **Mapping** table. You can select the corresponding LogAnalyzer field in the column opposite each capturing group. If you followed the CEF format when naming the capturing groups, you can use automatic CEF mapping by selecting the **Use CEF syntax for normalization** check box.

Event handling rules are added.

- syslog

  This parsing method is used to process data in syslog format.

  If you select this parsing method, you can use the predefined rules for converting events to the LogAnalyzer format by clicking **Apply default mapping**.

  To parse events in rfc5424 format with a `structured-data` section, in the **Keep extra fields** drop-down list, select **Yes**. This makes the values from the `structured-data` section available in the `Extra` fields.

- csv

  This parsing method is used to create custom rules for processing CSV data.

  When choosing this parsing method, you must specify the separator of values in the string in the **Delimiter** field. Any single-byte ASCII character can be used as a delimiter for values in a string.

- kv

This parsing method is used to process data in key-value pair format. Available parsing method settings are listed in the table below.

Available parsing method settings

Setting	Description
Pair delimiter	The character used to separate key-value pairs. You can specify any single-character (1 byte) value. The specified value must not match the value specified in the **Value delimiter** field.
Value delimiter	The character used to separate a key from its value. You can specify any single-character (1 byte) value. The specified value must not match the value specified in the **Pair delimiter** field.

- xml

This parsing method is used to process XML data in which each object, including nested objects, occupies a single line in a file. Files are processed line by line.

If you want to send the raw event for advanced normalization, at each nesting level in the **Advanced event parsing** window, in the **Use raw event** drop-down list, select **Yes**.

If you select this parsing method, under **XML attributes**, you can specify the key XML attributes to be extracted from tags. If an XML structure has multiple XML attributes with different values in the same tag, you can identify the necessary value by specifying the key of the value in the **Source** column of the **Mapping** table.

To add key XML attributes:

1. Click **+ Add field**.
2. This opens a window; in that window, specify the path to the XML attribute.

You can add multiple XML attributes or remove XML attributes. To remove an individual XML attribute, click the delete icon ✕ next to it. To remove all XML attributes, click **Reset**.

If XML key attributes are not specified, then in the course of field mapping the unique path to the XML value will be represented by a sequence of tags.

Tag numbering

Starting with LogAnalyzer 2.1.3, you can use automatic tag numbering in XML events. This lets you parse an event with the identical tags or unnamed tags, such as `<Data>`.

As an example, we will number the tags of the `EventData` attribute of the Microsoft Windows PowerShell event ID 800.

```
<Event xmlns="http://schemas
.microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Microsoft-Windows-
    ActiveDirectory_DomainService" Guid="{0e8478c5-3605-4e8c-8497-
    1e730c959516}" EventSourceName="NTDS" />
    <EventID Qualifiers="0000">0000</EventID>
    <Version>@</Version>
    <Level>4</Level>
    <Task>15</Task>
    <Opcode>0</Opcode>
    <Keywords >0x8080000000000000</Keywords>
    <TimeCreated SystemTime="2000-01-01T00:00:00.659495900Z" />
    <EventRecordID>55647</EventRecordID>
    <Correlation />
    <Execution ProcessID="1" ThreadID="1" />
    <Channel>service</Channel>
    <Computer>computer</Computer>
    <Security UserID="0000" />
  </System>
  <EventData>
    <Data>583</Data>
    <Data>36</Data>
    <Data>192.168.0.1:5084</Data>
    <Data>level</Data>
    <Data>name, lDAPDisplayName</Data>
    <Data />
    <Data>5545</Data>
    <Data>3</Data>
    <Data>0</Data>
    <Data>0</Data>
    <Data>0</Data>
    <Data>15</Data>
    <Data>none</Data>
  </EventData>
</Event>
```

To parse events with identical tags or unnamed tags, you need to configure tag numbering and data mapping for numbered tags with KUMA event fields.

LogAnalyzer 3.0.x supports using XML attributes and tag numbering at the same time in the same extra normalizer. If an XML attribute contains unnamed tags or identical tags, we recommend using tag numbering. If the XML attribute contains only named tags, we recommend using XML attributes.

To use XML attributes and tag numbering in extra normalizers, you must sequentially enable the **Use raw event** setting in each extra normalizer along the path that the event follows to the target extra normalizer, and in the target extra normalizer itself.

For an example of how tag numbering works, you can refer to the **MicrosoftProducts** normalizer. The **Use raw event** setting is enabled sequentially in both **AD FS** and **424** extra normalizers.

To set up the parsing of events with unnamed or identical tags:

1. Open an existing normalizer or create a new normalizer.
2. In the **Basic event parsing** window of the normalizer, in the **Parsing method** drop-down list, select **xml**.
3. In the **Tag numbering** field, click **+ Add field**.
4. In the displayed field, enter the full path to the tag to whose elements you want to assign a number, for example, `Event.EventData.Data`. The first tag gets number 0. If the tag is empty, for example, `<Data />`, it is also assigned a number.
5. To configure data mapping, under **Mapping**, click **+ Add row** and do the following:
   a. In the displayed row, in the **Source** field, enter the full path to the tag and the index of the tag. For example, for the Microsoft Windows PowerShell event ID 800 from the example above, the full paths to tags and tag indices are as follows:
      - `Event.EventData.Data.0`
      - `Event.EventData.Data.1`
      - `Event.EventData.Data.2` and so on.
   b. In the **KUMA field** drop-down list, select the field in the KUMA event that will receive the value from the numbered tag after parsing.
6. Save changes in one of the following ways:
   - If you created a new normalizer, click **Save**.
   - If you edited an existing normalizer, in the collector to which the normalizer is linked, click **Update configuration**.

Parsing is configured.

- ipfix

  This parsing method is used to process IPFIX data.

  If you select this parsing method, you can use the predefined rules for converting events to the LogAnalyzer format by clicking **Apply default mapping**. If the **ipfix** parsing method is selected for the main parsing, extra normalization is not available.

- sql

  The normalizer uses this parsing method to process data obtained by making a selection from the database.

- normalizer

	This parsing method allows you to reuse a previously created normalizer as the primary or extra normalizer for processing events. If you select this parsing method, in the **Normalizer** drop-down list, you need to select one of the previously created normalizers of the json, cef, regexp, syslog, csv, kv, xml, or sql type. Normalizers with the **normalizer** parsing method cannot be selected. Required setting.

Setting	Description
Name	Name of the parsing rule. Maximum length of the name: 128 Unicode characters. The name of the main parsing rule is used as the name of the normalizer. Required setting.
Tenant	The name of the tenant that owns the resource. This setting is not available for extra parsing rules.
Keep raw event	Keeping raw events in the newly created normalized event. Available values: • **Don't save**—do not save the raw event. This is the default setting. • **Only errors**—save the raw event in the `Raw` field of the normalized event if errors occurred when parsing it. This value is useful for debugging because an event having a non-empty `Raw` field indicates a problem. If fields containing the names `*Address` or `*Date*` do not comply with normalization rules, these fields are ignored. No normalization error occurs in this case, and the values of the fields are not displayed in the `Raw` field of the normalized event even if the **Keep raw event** → **Only errors**. • **Always**—always save the raw event in the `Raw` field of the normalized event. Required setting. This setting is not available for extra parsing rules.

Keep extra fields	Keep fields and values for which no mapping rules are configured. This data is saved as an array in the `Extra` event field. Normalized events can be searched and filtered based on the data stored in the `Extra` field.
	Filtering based on data from the Extra event field
	Conditions for filters based on data from the `Extra` event field:
	• Condition—**If**.
	• Left operand—**event field**.
	• In this event field, you can specify one of the following values:
	  • `Extra` field.
	  • Value from the `Extra` field in the following format:
	  `Extra.<field name>`
	  For example, `Extra.app`.
	  You must specify the value manually.
	  • Value from the array written to the `Extra` field in the following format:
	  `Extra.<field name>.<array element>`
	  For example, `Extra.array.0`.
	  The values in the array are numbered starting from 0. You must specify the value manually. To work with a value in the `Extra` field at a depth of 3 and lower, you must use backticks ``, for example, `` `Extra.lev1.lev2.lev3` ``.
	• Operator – =.
	• Right operand—**constant**.
	• Value—the value by which you need to filter events.
	By default, no extra fields are saved.
	Required setting. This setting is not available for the **normalizer** parsing method.

Description	Description of the resource. Maximum length of the description: 4000 Unicode characters. This setting is not available for extra parsing rules.

Event examples	Example of data that you want to process. This setting is not available for the following parsing methods: **ipfix**, **sql**, and **normalizer**. If the event was parsed successfully, and the type of the data obtained from the raw event matches the type of the LogAnalyzer field, the **Event examples** field is filled with data obtained from the raw event. For example, the `"192.168.0.1"` value in quotation marks does not appear in the `SourceAddress` field. However, the `192.168.0.1` value is displayed in the **Event examples** field.
Mapping	Settings for configuring the mapping of source event fields to fields of the event in the LogAnalyzer format: • **Source** lists the names of the raw event fields that you want to convert into LogAnalyzer event fields. Next to field names in the **Source** column, clicking opens the **Conversion** window, in which you can click **Add conversion** to create rules for modifying the source data before writing them to the LogAnalyzer event fields. You can reorder and delete created rules. To change the position of a rule, click next to it. To delete a rule, click next to it. Available conversions

Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.

- **lower**—is used to make all characters of the value lowercase

- **upper**—is used to make all characters of the value uppercase

- **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.

- **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.

- **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:

  - **Replace chars** specifies the sequence of characters to be replaced.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, the new value is `soft-Windows-Sys`.

- **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:

  - **Expression** is the RE2 regular expression whose results you want to replace.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.
- Converting the IP address to the IPv4 format:
  - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.
  - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append**, **prepend**.

- **LogAnalyzer field** lists fields of LogAnalyzer events. You can search for fields by entering their names.
- **Label** is a unique custom label for event fields that begin with `DeviceCustom*` and `Flex*`.

You can add new table rows or delete table rows. To add a new table row, click **Add row**. To delete a single row in the table, click ✕ next to it. To delete all table rows, click **Clear all**.

If you have loaded data into the **Event examples** field, the table will have an **Examples** column containing examples of values carried over from the raw event field to the LogAnalyzer event field.

If the size of the LogAnalyzer event field is less than the length of the value placed in it, the value is truncated to the size of the event field.

These settings are not available for the **normalizer** parsing method.

Extended event schema

You can use the extended event schema fields in normalizers for normalizing events and in other LogAnalyzer resources, for example, as widget fields or to filter and search for events. You can view the list of all extended event schema fields that exist in LogAnalyzer in the **Settings** → **Extended event schema fields** section. The list of extended event schema fields is the same for all tenants.

Only users with the General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Read shared resources, and Manage shared resources roles can view the table of extended event schema fields.

The **Extended event schema fields** table contains the following information:

- **Type**—Data type of the extended event schema field.

  Possible data types:

Type	Availability in the normalizer	Data type
S	All types	String.
N	All types	Number.
F	All types	Floating point number.
SA	KV, JSON	Array of strings. The order of the array elements is the same as the order of the elements of the raw event.
NA	KV, JSON	Array of integers. The order of the array elements is the same as the order of the elements of the raw event.
FA	KV, JSON	Array of floats. The order of the array elements is the same as the order of the elements of the raw event.

- **Field name**—Name of the extended event schema field, without a type.

  You can click the name to edit the settings of the extended event schema field.

- **Status**—Whether the extended event schema field can be used in resources.

  You can **Enable** or **Disable** the toggle switch to allow or forbid using this extended event schema field in new resources. However, a disabled field is still used in resource configurations that are already operational, until you manually remove the field from the configuration; the field also remains available in the list of table columns in the **Events** section for managing old events.

  Only a user with the General administrator role can disable an extended event schema field.

- **Updated at**—Date and time of the last modification of the extended event schema field.

- **Created by**—Name of the user that created the extended event schema field.

- **Dependencies**—Number of LogAnalyzer resources, dashboard layouts, reports, presets, and field sets for searching event sources that use the extended event schema field.

  You can click the number to open a pane with a table of all resources and other LogAnalyzer entities that are using this field. For each dependency, the table displays the name, tenant (only for resources), and type. Dependencies in the table are sorted by name. Clicking the name of a dependency takes you to its page (except for dashboard layouts, presets, and saved user queries).

  You can view the dependencies of an extended event schema field only for resources and entities to whose tenants you have access. If you do not have access to the tenant, its resources are not displayed in the table, but still count towards the number of dependencies.

- **Description**—Text description of the field.

By default, the table of extended event schema fields is sorted by update date in descending order. If necessary, you can sort the table by clicking a column heading and selecting **Ascending** or **Descending**; you can also use context search by field name.

By default, the following service extended event schema fields are automatically added to LogAnalyzer:

- `KL_EventRoute`, type `S` for storing information about the route of the event.

  You can use this field in normalizers and enrichment rules. You cannot use this field to detect event sources.

You cannot edit, delete, export, or disable service fields. All extended event schema fields with the `KL_` prefix are service fields and can be managed only from Kaspersky servers. We do not recommend using the `KL_` prefix when adding new extended event schema fields.

Adding extended event schema fields

Users with the General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Manage shared resources roles can add new extended event schema fields.

To add an extended event schema field:

1. In the LogAnalyzer web console, in the **Settings** → **Extended event schema fields** section, click the **Add** button in the upper part of the table.

   This opens the **Create extended field** pane.

2. Enable or disable the **Status** toggle switch to enable or disable this extended event schema field for resources.

   The toggle switch is turned on by default. A disabled field remains available in the list of table columns in the **Events** section for managing old events.

3. In the **Type** field, select the data type of the extended event schema field.

   Possible data types

Type	Availability in the normalizer	Data type
S	All types	String.
N	All types	Number.
F	All types	Floating point number.
SA	KV, JSON	Array of strings. The order of the array elements is the same as the order of the elements of the raw event.
NA	KV, JSON	Array of integers. The order of the array elements is the same as the order of the elements of the raw event.
FA	KV, JSON	Array of floats. The order of the array elements is the same as the order of the elements of the raw event.

4. In the **Name** field, specify the name of the extended event schema field.

   Consider the following when naming extended event schema fields:

   - The name must be unique within the LogAnalyzer instance.
   - Names are case-sensitive. For example, `Field_name` and `field_name` are different names.
   - You can use Latin, Cyrillic characters and numerals. Spaces or " ~ ` @ # $ % ^ & * ( ) + - [ ] { } | \ | / . " < > ; ! , : ? = characters are not allowed.
   - If you want to use the extended event schema fields to search for event sources, you can only use Latin characters and numerals.
   - The maximum length is 128 characters.

5. If necessary, in the **Description** field, enter a description for the extended event schema field.

   We recommend describing the purpose of the extended event schema field. Only Unicode characters are allowed in the description. The maximum length is 256 characters.

6. Click the **Save** button.

A new extended event schema field is added and displayed at the top of the table. An audit event is generated for the creation of the extended event schema field. If you have enabled the field, you can use it in normalizers and when configuring resources.

Editing extended event schema fields

Users with the General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Manage shared resources roles can edit existing extended event schema fields.

To edit an extended event schema field:

1. In the LogAnalyzer web console, in the **Settings** → **Extended event schema fields** section, click the name of the field that you want to edit.

   This opens the **Update extended field** pane. This pane displays the settings of the selected field, as well as the **Dependencies** table with a list of resources, dashboard layouts, reports, presets, and sets of fields for finding event sources that use this field. Only resources to whose tenants you have access are displayed. If the field is used by resources to whose tenant you do not have access, such resources are not displayed in the table. Resources in the table are sorted by name.

   Clicking the name of a resource or entity takes you to its page (except for dashboard resources, presets, and saved user queries).

2. Make the changes you need in the available settings.

   You can edit the **Type** and **Field name** settings only if the extended event schema field does not have dependencies. You can edit the **Status** and **Description** settings for any extended event scheme field. However, a field with the **Disabled** status is still used in resource configurations that are already operational, until you manually remove the field from the configuration; the field also remains available in the list of table columns in the **Events** section for managing old events.

   Disabling an extended event schema field using the **Status** field requires the General administrator role.

3. Click the **Save** button.

The extended event schema field is updated. An audit event is generated about the modification of the field.

Importing and exporting extended event schema fields

You can add multiple new extended event schema fields at once by importing them from a JSON file. You can also export all extended event schema fields with information about them to a file, for example, to propagate the list of fields to other LogAnalyzer instances to maintain resources.

Users with the General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, and Manage shared resources roles can import an export extended event schema fields. Users with the Read shared resources role can only export extended event schema fields.

To import extended event schema fields into LogAnalyzer from a file:

1. In the LogAnalyzer web console, in the **Settings** → **Extended event schema fields** section, click the **Import** button.

2. This opens a window; in that window, select a JSON file with a list of extended event schema field objects.

   Example JSON file:

```
[
{"kind": "SA",
"name": "<fieldName1>",
"description": "<description1>",
"disabled": false},
{"kind": "N",
"name": "<fieldName2>",
"description": "<description2>",
"disabled": false},
....
{"kind": "FA",
"name": "<fieldNameX>",
"description": "<descriptionX>",
"disabled": false}
]
```

When importing fields from a file, their names are checked for possible conflicts with fields of the same type. If a field with the same name and type already exists in LogAnalyzer, such fields are not imported from the file.

Extended event schema fields are imported from the file to LogAnalyzer. An audit event about the import of fields is generated, and a separate audit event is generated for each added field.

To export extended event schema fields to a file:

1. In the LogAnalyzer web console, go to the **Settings** → **Extended event schema fields** section.

2. If you want to export specific extended event schema fields:

   a. Select the check boxes in the first column of the table for the required fields.

      You cannot select service fields.

   b. Click the **Export selected** button in the upper part of the table.

3. If you want to export all extended event schema fields, click the **Export all** button in the upper part of the table.

A JSON file with a list of extended event schema field objects and information about them is downloaded.

Deleting extended event schema fields

Only a user with the General administrator role can delete extended event schema fields.

You can delete only those extended event schema fields that are not service fields, that have the **Disabled** status, and that are not used in LogAnalyzer resources and other entities (do not have dependencies). We recommend deleting extended event schema fields after enough time has passed to make sure that all events in which the field was used have been deleted from LogAnalyzer. When you delete a field, it is no longer displayed in event tips.

To delete extended event schema fields:

1. In the LogAnalyzer web console, go to the **Settings** → **Extended event schema fields** section.
2. Select the check boxes in the first column of the table next to one or more fields that you want to delete.

   To select all fields, you can select the check box in the heading of the first column.
3. Click the **Delete** button in the upper part of the table.

   The **Delete** button is active only if all selected fields are disabled and have no dependencies. If at least one field is enabled or has a dependency, the button is inactive.

   If you want to delete a field that is used in at least one LogAnalyzer resource (has a dependency), but you do not have access to its tenant, the **Delete** button is active when this field is selected, but an error is displayed when you try to delete it.

The selected fields are deleted. An audit event is generated about the deletion of the fields.

Using extended event schema fields in normalizers

When using extended event schema fields, the general limit for the maximum size of an event that can be processed by the collector is the same, 4 MB. Information about the types of extended event schema fields is shown in the table below (step 6 of the instructions).

Using many unique fields of the extended event schema can reduce the performance of the system, increase the amount of disk space required for storing events, and make the information difficult to understand.

We recommend consciously choosing a minimal set of additional fields of the extended event schema that you want to use in normalizers.

To use the fields of the extended event schema:

1. Open an existing normalizer or create a new normalizer.
2. Specify the basic settings of the normalizer.
3. Click **Add row**.
4. For the **Source** setting, enter the name of the source field in the raw event.
5. For the **KUMA field** setting, start typing the name of the extended event schema field and select the field from the drop-down list.

   The extended event schema fields in the drop-down list have names in the `<type>.<field name>` format.
6. Click the **Save** button to save the event normalizer.

The normalizer is saved with the selected extended event schema field.

If the data in the fields of the raw event does not match the type of the KUMA field, the value is not saved during the normalization of events if type conversion cannot be performed. For example, the string `test` cannot be written to the `DeviceCustomNumber1` KUMA field of the Number type.

If you want to minimize the load on the storage server when searching events, preparing reports, and performing other operations on events in storage, use KUMA event schema fields as your first preference, extended event schema fields as your second preference, and the `Extra` fields as your last resort.

Enrichment in the normalizer

When creating event parsing rules in the normalizer settings window, on the **Enrichment** tab, you can configure the rules for adding extra data to the fields of the normalized event using enrichment rules. Enrichment rules are stored in the settings of the normalizer where they were created.

You can create enrichment rules by clicking the **Add enrichment** button. To delete an enrichment rule, click ✕ next to it. Extended event schema fields can be used for event enrichment. Available enrichment rule settings are listed in the table below.

Available enrichment rule settings

Setting	Description

Source kind

Enrichment type. Depending on the selected enrichment type, advanced settings that will also need to be completed will be displayed. Available types of enrichment:

- **constant**

  This type of enrichment is used when a constant needs to be added to an event field. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Constant	The value to be added to the event field. Maximum length of the value: 255 Unicode characters. If you leave this field blank, the existing event field value is removed.
Target field	The LogAnalyzer event field that you want to populate with the data.

  If you are using the event enrichment functions for extended schema fields of `String`, `Number`, or `Float` type with a constant, the constant is added to the field.

  If you are using the event enrichment functions for extended schema fields of `Array of strings`, `Array of numbers`, or `Array of floats` type with a constant, the constant is added to the elements of the array.

- **event**

This type of enrichment is used when you need to write a value from another event field to the current event field. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Target field	The LogAnalyzer event field that you want to populate with the data.
Source field	The event field whose value is written to the target field.

Clicking opens the **Conversion** window, in which you can click **Add conversion** to create rules for modifying the source data before writing them to the LogAnalyzer event fields. You can reorder and delete created rules. To change the position of a rule, click next to it. To delete a rule, click next to it.

Available conversions

Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.

- **lower**—is used to make all characters of the value lowercase

- **upper**—is used to make all characters of the value uppercase

- **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.

- **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.

- **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:

  - **Replace chars** specifies the sequence of characters to be replaced.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, the new value is `soft-Windows-Sys`.

- **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:

  - **Expression** is the RE2 regular expression whose results you want to replace.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.
- Converting the IP address to the IPv4 format:
  - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.
  - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append**, **prepend**.

When using enrichment of events that have **eventevent** selected as the **Source kind** and the extended event schema fields are used as arguments, the following special considerations apply:

- If the source extended event schema field has the `Array of strings` type, and the target extended event schema field has the `String` type, the values are written to the target extended event schema field in TSV format.

  Example: the `SA.StringArray` extended event schema field contains values: `"string1"`, `"string2"`, `"string3"`. An event enrichment operation is performed. The result of the event enrichment operation is written to the `DeviceCustomString1` extended event schema field. The `DeviceCustomString1` extended event schema field contains values: `["string1", "string2", "string3"]`.

- If the source and target extended event schema fields have the `Array of strings` type, values of the source extended event schema field are added to the values of the target extended event schema field, and the "`,`" character is used as the delimiter.

  Example: The `SA.StringArrayOne` field of the extended event scheme contains the `["string1", "string2", "string3"]` values, and the `SA.StringArrayTwo` field of the extended event scheme contains the `["string4", "string5", "string6"]` values. An event enrichment operation is performed. The result of the event enrichment operation is written to the `SA.StringArrayTwo` field of the extended event scheme. The `SA.StringArrayTwo` extended event schema field contains values: `["string4", "string5", "string6", "string1", "string2", "string3"]`.

- template

  This type of enrichment is used when you need to write the result of processing Go templates into the event field. We recommend matching the value and the size of the field. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Template	The Go template. Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field from which the value must be passed to the script, for example, `{{.DestinationAddress}} attacked from {{.SourceAddress}}`.
Target field	The LogAnalyzer event field that you want to populate with the data.

  If you are using enrichment of events that have **template** selected as the **Source kind**, and in which the target field has the `String` type, and the source field is an extended event schema field containing an array of strings, you can use one of the following examples for the template:

  - `{{.SA.StringArrayOne}}`
  - `{{- range $index, $element := . SA.StringArrayOne -}}`
    `{{- if $index}}, {{end}}"{{$element}}"{{- end -}}`

  To convert the data in an array field in a template into the TSV format, use the `toString` function, for example:

  `template {{toString .SA.StringArray}}`

Required setting.

Setting	Description
Setting	**Description**
Setting	**Description**
Setting	**Description**
Target field	The LogAnalyzer event field that you want to populate with the data. Required setting.

Conditions for forwarding data to an extra normalizer

When creating additional event parsing rules, you can specify the conditions. When these conditions are met, the events are sent to the created parsing rule for processing. Conditions can be specified in the **Additional event parsing** window, on the **Extra normalization conditions** tab. This tab is not available for the basic parsing rules.

Available settings:

- **Use raw event** — If you want to send a raw event for extra normalization, select **Yes** in the **Use raw event** drop-down list. The default value is **No**. We recommend passing a raw event to normalizers of json and xml types. If you want to send a raw event for extra normalization to the second, third, etc nesting levels, at each nesting level, select **Yes** in the **Use raw event** drop-down list.

  > The **Use raw event** option is not applied if the sql parsing method is used in the main event parsing rule.

- **Field to pass into normalizer**—indicates the event field if you want only events with fields configured in normalizer settings to be sent for additional parsing.

  If this field is blank, the full event is sent to the extra normalizer for processing.

  If, when configuring the additional normalizer, you set the **Use raw event** setting to **No**, and left the **Source field** blank, any conversions configured for the field that is passed to the normalizer will not work. For conversions to be performed correctly, the source field must be filled, or the **Use raw event** setting must be set to **Yes**.

- Set of filters—used to define complex conditions that must be met by the events received by the normalizer.

  You can use the **Add condition** button to add a string containing fields for identifying the condition (see below).

  You can use the **Add group** button to add a group of filters. Group operators can be switched between **AND**, **OR**, and **NOT**. You can add other condition groups and individual conditions to filter groups.

  You can swap conditions and condition groups by dragging them by the ⣿ icon; you can also delete them using the ✕ icon.

Filter condition settings:

- **Left operand** and **Right operand**—used to specify the values to be processed by the operator.

  In the left operand, you must specify the source field of events coming into the normalizer. For example, if the eventType - DeviceEventClass mapping is configured in the **Basic event parsing** window, then in the **Additional event parsing** window on the **Extra normalization conditions** tab, you must specify eventType in the left operand field of the filter. Data is processed only as text strings.

- Operators:

  - **=** – full match of the left and right operands.
  - **startsWith** – the left operand starts with the characters specified in the right operand.
  - **endsWith** – the left operand ends with the characters specified in the right operand.
  - **match** – the left operand matches the regular expression (RE2) specified in the right operand.
  - **in** – the left operand matches one of the values specified in the right operand.

The incoming data can be converted by clicking the 🔧 button. This opens the **Conversion** window, in which you can click the **Add conversion** button to create the rules for converting the source data before any actions are performed on them. In the **Conversion** window, you can swap the added rules by dragging them by the ⣿ icon; you can also delete them using the ✕ icon.

Available conversions

Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.
- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.
- **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.
- **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:
  - **Replace chars** specifies the sequence of characters to be replaced.
  - **With chars** is the character sequence to be used instead of the character sequence being replaced.
- **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, the new value is `soft-Windows-Sys`.
- **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.
- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.
- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:
  - **Expression** is the RE2 regular expression whose results you want to replace.
  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.
- Converting the IP address to the IPv4 format:
  - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.
  - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append**, **prepend**.

Supported event sources

LogAnalyzer supports the normalization of events coming from systems listed in the table below. Normalizers for these systems are included in the distribution kit.

Supported event sources

System name	Normalizer name	Type	Normalizer description
1C EventJournal	[OOTB] 1C EventJournal Normalizer	xml	Designed for processing the event log of the 1C system. The event source is the 1C log.
1C TechJournal	[OOTB] 1C TechJournal Normalizer	regexp	Designed for processing the technology event log. The event source is the 1C technology log.

System name	Normalizer name	Type	Normalizer description
Absolute Data and Device Security (DDS)	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
AhnLab Malware Defense System (MDS)	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
AhnLab UTM	[OOTB] Ahnlab UTM	regexp	Designed for processing events from the Ahnlab system. The event sources are system logs, operation logs, connections, the IPS module.
Alcatel AOS-W	[OOTB] Alcatel AOS-W Syslog	regexp	Designed for processing some of the events received from Alcatel AOS-W 6.4 via syslog.
Alcatel Network Switch	[OOTB] Alcatel Network Switch Syslog	syslog	Designed for processing events received from Alcatel network switches via Syslog.
Apache Cassandra	[OOTB] Apache Cassandra file	regexp	Designed for processing events from the logs of the Apache Cassandra database version 4.0.
Apache Kafka	[OOTB] Apache Kafka Syslog	regexp	Designed for processing Apache Kafka 3.8.1 system audit events received via Syslog.
Apache Tomcat	[OOTB] Apache Tomcat json	syslog	Designed for processing events from the Apache Tomcat system (version 9.0 and later, 10.1 and later, 11.0 and later) via Syslog in JSON and key-value formats.
APC Smart-UPS	[OOTB] APC Smart UPS	regexp	Designed for processing APC Smart-UPS system audit events.
Aruba ClearPass	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Atlassian Confluence	[OOTB] Atlassian Jira Conflunce file	regexp	Designed for processing events of Atlassian Jira, Atlassian Confluence systems (Jira 9.12, Confluence 8.5) stored in files.
Atlassian Jira	[OOTB] Atlassian Jira Conflunce file	regexp	Designed for processing events of Atlassian Jira, Atlassian Confluence systems (Jira 9.12, Confluence 8.5) stored in files.
Avanpost FAM	[OOTB] Avanpost FAM Syslog	regexp	Designed for processing events of the Avanpost Federated Access Manager (FAM) 1.9 received via syslog.
Avanpost IDM	[OOTB] Avanpost IDM Syslog	regexp	Designed for processing events of the Avanpost IDM system received via syslog.
Avanpost PKI	[OOTB] Avanpost PKI Syslog CEF	syslog	Designed for processing events received from Avanpost PKI 6.0 in CEF format via syslog.
Avaya Aura Communication Manager	[OOTB] Avaya Aura Communication Manager Syslog	regexp	Designed for processing some of the events received from Avaya Aura Communication Manager 7.1 via syslog.
Avigilon Access Control Manager (ACM)	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Ayehu eyeShare	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Arbor Pravail	[OOTB] Arbor Pravail Syslog	syslog	Designed for processing events of the Arbor Pravail system received via syslog.
Aruba Aruba AOS-S	[OOTB] Aruba Aruba AOS-S Syslog	regexp	Designed for processing events received from Aruba network devices with Aruba AOS-S 16.10 firmware via syslog. This normalizer supports the following types of events: accounting events, ACL events, ARP protect events, authentication events, console events, loop protect events.
Barracuda Cloud Email Security Gateway	[OOTB] Barracuda Cloud Email Security Gateway Syslog	regexp	Designed for processing events from Barracuda Cloud Email Security Gateway via syslog.
Barracuda Networks NG Firewall	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.

System name	Normalizer name	Type	Normalizer description
Barracuda Web Security Gateway	[OOTB] Barracuda Web Security Gateway Syslog	syslog	Designed for processing some of the events received from Barracuda Web Security Gateway 15.0 via syslog.
BeyondTrust Privilege Management Console	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
BeyondTrust's BeyondInsight	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Bifit Mitigator	[OOTB] Bifit Mitigator Syslog	syslog	Designed for processing events from the DDOS Mitigator protection system received via syslog.
Bloombase StoreSafe	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
BMC CorreLog	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Bricata ProAccel	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Brinqa Risk Analytics	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Broadcom Symantec Advanced Threat Protection (ATP)	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Broadcom Symantec Endpoint Protection	[OOTB] Broadcom Symantec Endpoint Protection	regexp	Designed for processing events from the Symantec Endpoint Protection system.
Broadcom Symantec Endpoint Protection Mobile	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Broadcom Symantec Threat Hunting Center	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Brocade Fabric OS	[OOTB] Brocade Fabric OS Syslog	syslog	Designed for processing events of Brocade Fabric 9.1 received via syslog.
Canonical LXD	[OOTB] Canonical LXD Syslog	syslog	Designed for processing events received via syslog from the Canonical LXD system version 5.18.
Check Point	[OOTB] Checkpoint Syslog	syslog	Designed for processing events received from the Check Point R81 firewall via the syslog protocol.
Cisco Access Control Server (ACS)	[OOTB] Cisco ACS Syslog	regexp	Designed for processing events of the Cisco Access Control Server (ACS) system received via syslog.
Cisco ASA	[OOTB] Cisco ASA and IOS Syslog	syslog	Designed for processing events from network devices: Cisco ASA, Cisco IOS, Cisco Firepower Threat Defense (version 7.2), FWSM, NGIPS received via syslog.
Cisco Email Security Appliance (WSA)	[OOTB] Cisco WSA AccessFile	regexp	Designed for processing the event log of the Cisco Email Security Appliance (WSA) 14.2 proxy server, the access.log file.
Cisco ESA syslog	[OOTB] Cisco ESA Syslog	syslog	Designed for processing certain types of events received from Cisco ESA 16.0 via syslog.
Cisco Firepower Threat Defense	[OOTB] Cisco ASA and IOS Syslog	syslog	Designed for processing events from network devices: Cisco ASA, Cisco IOS, Cisco Firepower Threat Defense (version 7.2), FWSM, NGIPS received via syslog.
Cisco Firewall Services Module (FWSM)	[OOTB] Cisco ASA and IOS Syslog	syslog	Designed for processing events from network devices: Cisco ASA, Cisco IOS, Cisco Firepower Threat Defense (version 7.2), FWSM, NGIPS received via syslog.

System name	Normalizer name	Type	Normalizer description
Cisco Identity Services Engine (ISE)	[OOTB] Cisco ISE Syslog	regexp	Designed for processing events of the Cisco Identity Services Engine (ISE) system received via syslog.
Cisco IOS	[OOTB] Cisco ASA and IOS Syslog	syslog	Designed for processing events from network devices: Cisco ASA, Cisco IOS, Cisco Firepower Threat Defense (version 7.2), FWSM, NGIPS received via syslog.
Cisco Next-Generation Intrusion Prevention System (NGIPS)	[OOTB] Cisco ASA and IOS Syslog	syslog	Designed for processing events from network devices: Cisco ASA, Cisco IOS, Cisco Firepower Threat Defense (version 7.2), FWSM, NGIPS received via syslog.
Cisco Prime	[OOTB] Cisco Prime Syslog	syslog	Designed for processing events of the Cisco Prime system version 3.10 received via syslog.
Cisco Secure Email Gateway (SEG)	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Cisco Secure Firewall Management Center	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Cisco Wireless LAN Controllers (WLCs)	[OOTB] Cisco WLC Syslog	regexp	Designed for processing events received from Cisco WLC network devices (2500 Series Wireless Controllers, 5500 Series Wireless Controllers, 8500 Series Wireless Controllers, Flex 7500 Series Wireless Controllers) via syslog.
Cisco Web Security Appliance (WSA)	[OOTB] Cisco WSA file, [OOTB] Cisco WSA Syslog	regexp	[OOTB] Cisco WSA file is a normalizer designed for processing the event log of the Cisco WSA proxy server (versions 14.2, 15.0). This normalizer supports processing events generated using the following template: %t %e %a %w/%h %s %2r %A %H/%d %c %D %Xr %?BLOCK_SUSPECT_USER_AGENT,MONITOR_SUSPECT_USER_AGENT?%<User-Agent:%!%-%. %) %q %k %u %m [OOTB] Cisco WSA syslog is a normalizer designed for processing events received from the Cisco WSA system (version 15.0) via syslog.
Citrix NetScaler	[OOTB] Citrix NetScaler Syslog	regexp	Designed for processing events received from the Citrix NetScaler 13.7 load balancer, Citrix ADC NS13.0.
Claroty Continuous Threat Detection	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
CloudPassage Halo	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Codemaster Mirada	[OOTB] Codemaster Mirada Syslog	syslog	Designed for processing events of the Codemaster Mirada system received via syslog.
CollabNet Subversion Edge	[OOTB] CollabNet Subversion Edge Syslog	syslog	Designed for processing events received from the Subversion Edge (version 6.0.2) system via syslog.
CommuniGate Pro	[OOTB] CommuniGate Pro	regexp	Contains a resource set for processing some of the events of the SIP module of the CommuniGate Pro system versions 6.1 and 6.5.1, transmitted by the LogAnalyzer agent via TCP. For details on configuring the collector, see Creating a collector.
Corvil Network Analytics	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Cribl Stream	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
CrowdStrike Falcon	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
CyberArk Privileged Threat Analytics (PTA)	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
CyberPeak Spektr	[OOTB] CyberPeak Spektr Syslog	syslog	Designed for processing events of the CyberPeak Spektr system version 3 received via syslog.

System name	Normalizer name	Type	Normalizer description
Cyberprotect Cyber Backup	[OOTB] Cyberprotect Cyber Backup SQL [OOTB] Cyberprotect Cyber Backup Syslog	sql, regexp	[OOTB] Cyberprotect Cyber Backup SQL is a normalizer designed to process events received by the connector from the database of the Cyber Backup system (version 16.5). [OOTB] Cyberprotect Cyber Backup syslog is a normalizer designed to process events received from the Cyber Backup system (version 17.2) via syslog in CEF format. This package is available for LogAnalyzer version 3.2 or later.
Deep Instinct	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Delinea Secret Server	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Dell Network Switch	[OOTB] Dell Network Switch Syslog	regexp	Designed for processing events received from Dell network switches via syslog.
Digital Guardian Endpoint Threat Detection	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
BIND DNS server	[OOTB] BIND syslog [OOTB] BIND file	syslog regexp	[OOTB] BIND syslog is designed for processing events of the BIND DNS server received via syslog. [OOTB] BIND file is designed for processing event logs of the BIND DNS server.
Docsvision	[OOTB] Docsvision Syslog	syslog	Designed for processing audit events received from the Docsvision system via syslog.
Dovecot	[OOTB] Dovecot Syslog	syslog	Designed for processing events of the Dovecot mail server received via syslog. The event source is POP3/IMAP logs.
Dragos Platform	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Dr.Web Enterprise Security Suite	[OOTB] Syslog-CEF	syslog	Designed for processing Dr.Web Enterprise Security Suite 13.0.1 events in the CEF format.
EclecticIQ Threat Intelligence Platform	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Edge Technologies AppBoard and enPortal	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Eltex ESR	[OOTB] Eltex ESR Syslog	syslog	Designed to process part of the events received from Eltex ESR network devices via syslog.
Eltex MES	[OOTB] Eltex MES Syslog	regexp	Designed for processing events received from Eltex MES network devices via syslog (supported device models: MES14xx, MES24xx, MES3708P).
ESET PROTECT	[OOTB] ESET Protect Syslog CEF	syslog	Designed for processing events received in the CEF format received from the ESET Protect system (version 11.0) via syslog. This package is available for LogAnalyzer version 3.2 or later.
Extreme Networks Summit Wireless Controller	[OOTB] Extreme Networks Summit Wireless Controller	regexp	Designed for processing of audit events of the Extreme Networks Summit Wireless Controller (model: WM3700, firmware version: 5.5.5.0-018R).
Factor-TS Dionis NX	[OOTB] Factor-TS Dionis NX Syslog	regexp	Designed for processing audit events received from the Dionis-NX system (version 2.0.3) via syslog.
F5 Advanced Web Application Firewall	[OOTB] F5 Advanced Web Application Firewall Syslog	regexp	Designed for processing audit events received from the F5 Advanced Web Application Firewall system via syslog.
F5 BigIP Advanced Firewall Manager (AFM)	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
FFRI FFR yarai	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.

System name	Normalizer name	Type	Normalizer description
FireEye CM Series	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
FireEye Malware Protection System	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Forcepoint NGFW	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Forcepoint SMC	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Fortinet FortiAnalyzer	[OOTB] Syslog-CEF	syslog	Designed for processing events received from Fortinet FortiAnalyzer 7.0, 7.2 via syslog in CEF format.
Fortinet FortiGate	[OOTB] FortiGate syslog KV, [OOTB] FortiGate syslog CEF	regexp	The [OOTB] FortiGate syslog KV normalizer is designed for processing events received from FortiGate 7.0 firewalls via syslog. The event source is FortiGate logs in key-value format. The [OOTB] FortiGate syslog CEF normalizer is designed for processing events received from Fortinet FortiGate (FortiOS 6.4) via syslog in CEF format.
Fortinet FortiMail	[OOTB] Fortimail	regexp	Designed for processing events of the FortiMail email protection system. The event source is FortiMail mail system logs.
Fortinet FortiSOAR	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Fortinet FortiWeb	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
FreeBSD	[OOTB] FreeBSD file	regexp	Designed for processing events of the FreeBSD operating system (version 13.1-RELEASE) stored in a file. The normalizer can process files produced by the praudit utility. Example: praudit -xl /var/audit/AUDITFILE >> file_name.log
FreeIPA	[OOTB] FreeIPA	json	Designed for processing events from the FreeIPA system. The event source is FreeIPA directory service logs.
FreeRADIUS	[OOTB] FreeRADIUS Syslog	syslog	Designed for processing events of the FreeRADIUS system received via syslog. This normalizer supports events from FreeRADIUS version 3.0.
GajShield Firewall	[OOTB] GajShield Firewall Syslog	regexp	Designed for processing part of the events received from the GajShield Firewall version GAJ_OS_Bulwark_Firmware_v4.35 via syslog.
Garda Monitor	[OOTB] Garda Monitor Syslog	syslog	Designed for processing events of the Garda Monitor system version 3.4 received via syslog.
Gardatech Garda DB	[OOTB] Gardatech GardaDB Syslog	syslog	Designed for processing events of the Gardatech Perimeter system version 5.3, 5.4 received via syslog.
Gardatech Perimeter	[OOTB] Gardatech Perimeter Syslog	syslog	Designed for processing events of the Gardatech Perimeter system version 5.3 received via syslog.
Gigamon GigaVUE	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
HAProxy	[OOTB] HAProxy Syslog	syslog	Designed for processing events received from HAProxy 2.4 via syslog. This normalizer supports events in the Default, TCP, TCP CLF (version 3.2 and later), HTTP, HTTP CLF, HTTPS (2.6 and later), and Error formats.
HashiCorp Vault	[OOTB] HashiCorp Vault json	json	Designed for processing events received from the HashiCorp Vault system version 1.16 in JSON format. The normalizer package is available in LogAnalyzer 3.0 and later.
Huawei Eudemon	[OOTB] Huawei Eudemon	regexp	Designed for processing events from Huawei Eudemon firewalls. The event source is logs of Huawei Eudemon firewalls.
Huawei iManager 2000	[OOTB] Huawei iManager 2000 file	regexp	Designed for processing of some of the events of the Huawei iManager 2000 system, which are stored in the \client\logs\rpc, \client\logs\deploy\ossDeployment files.
Huawei USG	[OOTB] Huawei USG Basic	syslog	Designed for processing events received from Huawei USG security gateways via syslog.

System name	Normalizer name	Type	Normalizer description
Huawei VRP	[OOTB] Huawei VRP Syslog	regexp	Designed for processing Huawei VRP events received via syslog. The normalizer makes a partial selection of event data. The normalizer is available in LogAnalyzer 3.0 and later.
IBM InfoSphere Guardium	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Ideco UTM	[OOTB] Ideco UTM Syslog	syslog	Designed for processing events received from Ideco UTM via syslog. This normalizer supports events of Ideco UTM 14.7, 14.10, 17.5.
Illumio Policy Compute Engine (PCE)	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Imperva Incapsula	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Imperva SecureSphere	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Indeed Access Manager	[OOTB] Indeed Access Manager Syslog	syslog	Designed for processing events received from the Indeed Access Manager system via syslog.
Indeed Privileged Access Manager	[OOTB] Indeed PAM Syslog	syslog	Designed for processing events of Indeed Privileged Access Manager version 2.6.
Indeed SSO	[OOTB] Indeed SSO xml	xml	Designed for processing events of the Indeed SSO (Single Sign-On) system. This normalizer supports LogAnalyzer 2.1.3 and later.
InfoWatch Person Monitor	[OOTB] InfoWatch Person Monitor SQL	sql	Designed for processing system audit events from the MS SQL database of InfoWatch Person Monitor 10.2.
InfoWatch Traffic Monitor	[OOTB] InfoWatch Traffic Monitor SQL	sql	Designed for processing events received by the connector from the database of the InfoWatch Traffic Monitor system.
Intralinks VIA	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
IPFIX	[OOTB] IPFIX	ipfix	Designed for processing events in the IP Flow Information Export (IPFIX) format.
Juniper JUNOS	[OOTB] Juniper-JUNOS	regexp	Normalizer for Juniper-JUNOS (version 24.2) events received via syslog.
Kaspersky Anti Targeted Attack (KATA)	[OOTB] KATA	cef	Designed for processing events from the Kaspersky Anti Targeted Attack activity log.
Kaspersky CyberTrace	[OOTB] CyberTrace	regexp	Designed for processing Kaspersky CyberTrace events.
Kaspersky DFI	[OOTB] Kaspersky DFI json	json	Designed for processing audit events of the Kaspersky DFI system in JSON format. This normalizer supports the processing of events generated by the "dfi" LogAnalyzer connector. Available in the resource set for LogAnalyzer 4.0 and later.
Kaspersky Endpoint Detection and Response (KEDR)	[OOTB] KEDR telemetry	json	Designed for processing Kaspersky EDR telemetry tagged by KATA. The event source is kafka, EnrichedEventTopic.
Kaspersky Endpoint Security for Linux	[OOTB] KESL Syslog cef	syslog	Designed for processing events from Kaspersky Endpoint Security for Linux (KESL) 12.2 in CEF format via syslog.
KICS/KATA	[OOTB] KICS4Net v2.x	cef	Designed for processing KICS/KATA version 2.x events.
KICS/KATA	[OOTB] KICS4Net v3.x	syslog	Designed for processing KICS/KATA version 3.x events.
Kaspersky Industrial CyberSecurity for Networks 4.2	[OOTB] Kaspersky Industrial CyberSecurity for Networks 4.2 Syslog	syslog	Designed for processing events received from Kaspersky Industrial CyberSecurity for Networks 4.2 via syslog.
Kaspersky IoT Secure Gateway (KISG)	[OOTB] Kaspersky KISG Syslog	syslog	Designed for processing events received from Kaspersky IoT Secure Gateway (KISG) 3.0 via syslog.

System name	Normalizer name	Type	Normalizer description
Kaspersky NGFW	[OOTB] Kaspersky NGFW CEF	cef	Designed for processing events received from the Kaspersky NGFW system version 1.0 in CEF format.
Kaspersky NDR	[OOTB] Kaspersky NDR Syslog	syslog	Designed for processing events received from the Kaspersky NDR 7.0 system via syslog. This package is available for LogAnalyzer version 3.2 or later.
Kaspersky Security Center	[OOTB] KSC	cef	Designed for processing Kaspersky Security Center events received in CEF format.
Kaspersky Security Center	[OOTB] KSC from SQL	sql	Designed for processing events received by the connector from the database of the Kaspersky Security Center system.
Kaspersky Security for Linux Mail Server (KLMS)	[OOTB] KLMS Syslog CEF	syslog	Designed for processing events from Kaspersky Security for Linux Mail Server in CEF format via syslog.
Kaspersky Security for MS Exchange SQL	[OOTB] Kaspersky Security for MS Exchange SQL	sql	Designed for processing Kaspersky Security for Exchange (KSE) 9.0 events stored in the database.
Kaspersky Secure Mail Gateway (KSMG)	[OOTB] KSMG syslog CEF [OOTB] KSMG 2.1+ Syslog CEF	syslog	[OOTB] KSMG syslog CEF is a normalizer for processing KSMG 2.0 events received in CEF format via syslog. [OOTB] KSMG 2.1+ syslog CEF is a normalizer for processing KSMG 2.1.1 events received in CEF format via syslog.
Kaspersky Unified Monitoring and Analysis Platform (LogAnalyzer)	[OOTB] LogAnalyzer forwarding	json	Designed for processing events forwarded from Kaspersky Unified Monitoring and Analysis Platform (LogAnalyzer).
Kaspersky Web Traffic Security (KWTS)	[OOTB] KWTS Syslog CEF, [OOTB] KWTS (KV)	syslog	The [OOTB] KWTS syslog CEF normalizer is designed for processing events received from Kaspersky Web Traffic Security in CEF format via syslog. The [OOTB] KWTS (KV) normalizer is designed for processing Kaspersky Web Traffic Security events in key-value format.
Keycloak	[OOTB] Keycloak Syslog	syslog	Designed for processing events of the Keycloak 26.3 system received via syslog.
Kemptechnologies LoadMaster	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Kerio Control	[OOTB] Kerio Control	syslog	Designed for processing events of Kerio Control firewalls.
LogAnalyzer	[OOTB] LogAnalyzer forwarding	json	Designed for processing events forwarded from LogAnalyzer.
LastLine Enterprise	[OOTB] LastLine Enterprise Syslog cef	syslog	Designed for processing events received from LastLine Enterprise 7.3, 8.1, 9.1 via syslog in CEF format.
Libvirt	[OOTB] Libvirt Syslog	syslog	Designed for processing events of Libvirt version 8.0.0 received via syslog.
Lieberman Software ERPM	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Linux	[OOTB] Linux audit and iptables Syslog v1	syslog	Designed for processing events of the Linux operating system. This normalizer does not support processing events in the ENRICHED format.
Linux auditd	[OOTB] Linux auditd Syslog for LogAnalyzer 3.2	syslog	Designed for processing audit events (auditd package) of the Linux operating system received via syslog. This normalizer supports events that have been processed by a LogAnalyzer collector version 3.2 or later.
Linux auditd	[OOTB] Linux auditd file for LogAnalyzer 3.2	regexp	Designed for processing audit events (auditd package) of the Linux operating system saved to a file. This normalizer supports events that have been processed by a LogAnalyzer collector version 3.2 or later.
MariaDB	[OOTB] MariaDB Audit Plugin Syslog	syslog	Designed for processing events coming from the MariaDB audit plugin over syslog.
McAfee Endpoint DLP	[OOTB] McAfee Endpoint DLP Syslog	syslog	Designed for processing events received from McAfee Endpoint DLP Windows 11.10.200 via syslog. This package is available for LogAnalyzer version 3.2 or later.

System name	Normalizer name	Type	Normalizer description
Microsoft Active Directory Certificate Services (AD CS)	[OOTB] Microsoft Products for LogAnalyzer 3	xml	Designed for processing Active Directory Certificate Services (AD CS) log events in XML format. The [OOTB] Microsoft Products for LogAnalyzer 3 normalizer supports this event source in LogAnalyzer 3.0.1 and later versions.
Microsoft Active Directory Federation Service (AD FS)	[OOTB] Microsoft Products for LogAnalyzer 3	xml	Designed for processing Microsoft AD FS events. The [OOTB] Microsoft Products for LogAnalyzer 3 normalizer supports this event source in LogAnalyzer 3.0.1 and later versions.
Microsoft Active Directory Domain Service (AD DS)	[OOTB] Microsoft Products for LogAnalyzer 3	xml	Designed for processing Microsoft AD DS events. The [OOTB] Microsoft Products for LogAnalyzer 3 normalizer supports this event source in LogAnalyzer 3.0.1 and later versions.
Microsoft Defender	[OOTB] Microsoft Products, [OOTB] Microsoft Products for LogAnalyzer 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing Microsoft Defender events.
Microsoft DHCP	[OOTB] MS DHCP file	regexp	Designed for processing Microsoft DHCP server events. The event source is Windows DHCP server logs.
Microsoft DNS	[OOTB] DNS Windows [OOTB] Microsoft DNS ETW logs json	regexp	The [OOTB] Windows DNS normalizer is designed to process Microsoft DNS server events. The event source is Windows DNS server logs. The normalizer does not support processing debug log events with the **Details** option enabled. The [OOTB] Microsoft DNS ETW logs json normalizer is designed to process some Microsoft DNS Server audit events supplied by the ETW provider. This package is available for LogAnalyzer version 3.2 or later.
Microsoft Exchange	[OOTB] Exchange CSV	csv	Designed for processing the event log of the Microsoft Exchange system. The event source is Exchange server MTA logs.
Microsoft Hyper-V	[OOTB] Microsoft Products, [OOTB] Microsoft Products for LogAnalyzer 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing Microsoft Windows events. The event source is Microsoft Hyper-V logs: Microsoft-Windows-Hyper-V-VMMS-Admin, Microsoft-Windows-Hyper-V-Compute-Operational, Microsoft-Windows-Hyper-V-Hypervisor-Operational, Microsoft-Windows-Hyper-V-StorageVSP-Admin, Microsoft-Windows-Hyper-V-Hypervisor-Admin, Microsoft-Windows-Hyper-V-VMMS-Operational, Microsoft-Windows-Hyper-V-Compute-Admin.
Microsoft IIS	[OOTB] IIS Log File Format	regexp	Designed for processing events in the format described at https://learn.microsoft.com/en-us/windows/win32/http/iis-logging. The event source is Microsoft IIS logs.
Microsoft Network Policy Server (NPS)	[OOTB] Microsoft Products, [OOTB] Microsoft Products for LogAnalyzer 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing Microsoft Windows events. The event source is Network Policy Server events.
Microsoft Office 365	[OOTB] Microsoft Office 365 json	json	Designed for processing Microsoft Office 365 audit events. This normalizer supports processing events received from Microsoft Teams, Azure Active Directory, SharePoint systems. This package is available for LogAnalyzer version 3.4 or later.
Microsoft SCCM	[OOTB] Microsoft SCCM file	regexp	Designed for processing events of the Microsoft SCCM system version 2309. This normalizer supports processing of some of the events stored in the AdminService.log file.
Microsoft SharePoint Server	[OOTB] Microsoft SharePoint Server diagnostic log file	regexp	Designed for processing some of Microsoft SharePoint Server 2016 events stored in diagnostic logs.
Microsoft Sysmon	[OOTB] Microsoft Products, [OOTB] Microsoft Products for LogAnalyzer 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing Microsoft Sysmon module events.

System name	Normalizer name	Type	Normalizer description
Microsoft Windows 7, 8.1, 10, 11	[OOTB] Microsoft Products, [OOTB] Microsoft Products for LogAnalyzer 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing part of events from the Security, System, Application logs of the Microsoft Windows operating system. The "[OOTB] Microsoft Products via KES WIN" normalizer supports a limited number of audit event types sent to LogAnalyzer by Kaspersky Endpoint Security 12.6 for Windows via syslog.
Microsoft → Windows → DNS-Server → Audit	[OOTB] Microsoft Products for LogAnalyzer 3	xml	Designed for processing Microsoft → Windows → DNS-Server → Audit log events in XML format.
Microsoft → Windows → Group Policy → Operational	[OOTB] Microsoft Products for KUMA 3	xml	Designed to process Microsoft → Windows → Group Policy → Operational log events in XML format.
Microsoft → Windows → PowerShell	[OOTB] Microsoft Products, [OOTB] Microsoft Products for LogAnalyzer 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing Microsoft Windows PowerShell log events.
Microsoft → Windows → PowerShell-Operational	[OOTB] Microsoft Products for LogAnalyzer 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing Microsoft Windows PowerShell-Operational log events. The "[OOTB] Microsoft Products via KES WIN" normalizer supports a limited number of audit event types sent to LogAnalyzer by Kaspersky Endpoint Security 12.6 for Windows via syslog.
Microsoft SQL Server	[Deprecated] [OOTB] Microsoft SQL Server xml	xml	Designed for processing events of MS SQL Server versions 2008, 2012, 2014, 2016. This normalizer supports LogAnalyzer 2.1.3 and later.
Microsoft SQL Server, Microsoft SQL Server Express	[OOTB] Microsoft Products for LogAnalyzer 3	xml	Designed to process events of MS SQL Server 2008 or newer.
Microsoft Windows Remote Desktop Services	[OOTB] Microsoft Products, [OOTB] Microsoft Products for LogAnalyzer 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing Microsoft Windows events. Event source → Applications and Services Logs → Microsoft → Windows → TerminalServices-LocalSessionManager → Operational log The "[OOTB] Microsoft Products via KES WIN" normalizer supports a limited number of audit event types sent to LogAnalyzer by Kaspersky Endpoint Security 12.6 for Windows via syslog.
Microsoft Windows Service Control Manager	[OOTB] Microsoft Products for LogAnalyzer 3 [OOTB] Microsoft Products via KES WIN	xml	Designed for processing events from the Service Control Manager logs (System log) of the Microsoft Windows operating system.
Microsoft Windows Server 2008 R2, 2012 R2, 2016, 2019, 2022	[OOTB] Microsoft Products, [OOTB] Microsoft Products for LogAnalyzer 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing part of events from the Security, System logs of the Microsoft Windows Server operating system. The "[OOTB] Microsoft Products via KES WIN" normalizer supports a limited number of audit event types sent to LogAnalyzer by Kaspersky Endpoint Security 12.6 for Windows via syslog.
Microsoft Windows XP/2003	[OOTB] SNMP. Windows {XP/2003}	json	Designed for processing events received from workstations and servers running Microsoft Windows XP, Microsoft Windows 2003 operating systems using the SNMP protocol.
Microsoft WSUS	[OOTB] Microsoft WSUS file	regexp	Designed for processing events of the Gardatech Perimeter system version 5.3, 5.4 received via syslog.
MikroTik	[OOTB] MikroTik Syslog	regexp	Designed for events received from MikroTik devices via syslog.

System name	Normalizer name	Type	Normalizer description
Minerva Labs Minerva EDR	[OOTB] Minerva EDR	regexp	Designed for processing events from the Minerva EDR system.
MongoDB	[OOTB] MongoDb Syslog	syslog	Designed for processing some events received from the MongoDB 7.0 database via syslog.
Multifactor Radius Server for Windows	[OOTB] Multifactor Radius Server for Windows Syslog	syslog	Designed for processing events received from the Multifactor Radius Server 1.0.2 for Microsoft Windows via syslog.
MySQL 5.7	[OOTB] MariaDB Audit Plugin Syslog	syslog	Designed for processing events coming from the MariaDB audit plugin over syslog.
NetApp ONTAP (AFF, FAM)	[OOTB] NetApp syslog, [OOTB] NetApp file	regexp	[OOTB] NetApp syslog — designed for processing events of the NetApp system (version — ONTAP 9.12) received via syslog. [OOTB] NetApp file is a normalizer designed for processing events of the NetApp system (ONTAP 9.12 version) stored in a file.
NetApp SnapCenter	[OOTB] NetApp SnapCenter file	regexp	Designed to process part of the events of the NetApp SnapCenter system (SnapCenter Server 5.0). This normalizer supports processing some of the events from the C:\Program Files\NetApp\SnapCenter WebApp\App_Data\log\napManagerWeb.*.log file. Types of supported events in xml format from the SnapManagerWeb.*.log file: SmDiscoverPluginRequest, SmDiscoverPluginResponse, SmGetDomainsResponse, SmGetHostPluginStatusRequest, SmGetHostPluginStatusResponse, SmGetHostRequest, SmGetHostResponse, SmRequest. This normalizer supports processing some of the events from the C:\Program Files\NetApp\SnapCenter WebApp\App_Data\log\audit.log file.
NetIQ Identity Manager	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
NetScout Systems nGenius Performance Manager	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Netskope Cloud Access Security Broker	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Netwrix Auditor	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Netwrix Endpoint Protector	[OOTB] Netwrix Endpoint Protector Syslog	regexp	Designed for processing events received from the Netwrix Endpoint Protector 5.9 system via syslog. This package is available for LogAnalyzer version 3.2 or later.
Nextcloud	[OOTB] Nextcloud Syslog	syslog	Designed for events of Nextcloud version 26.0.4 received via syslog. The normalizer does not save information from the Trace field.
Nexthink Engine	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Nginx	[OOTB] Nginx regexp	regexp	Designed for processing the event log of the Nginx web server in the Combined format. This normalizer supports the processing of events received via syslog and stored in a file. This package is available for LogAnalyzer version 3.2 or later.
NIKSUN NetDetector	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
One Identity Privileged Session Management	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
OpenLDAP	[OOTB] OpenLDAP	regexp	Designed for line-by-line processing of some events of the OpenLDAP 2.5 system in an auditlog.ldif file.
Open VPN	[OOTB] OpenVPN file	regexp	Designed for processing the event log of the OpenVPN system.
Oracle	[OOTB] Oracle Audit Trail	sql	Designed for processing database audit events received by the connector directly from an Oracle database.
OrionSoft Termit	[OOTB] OrionSoft Termit Syslog	syslog	Designed for processing events received from the OrionSoft Termit 2.2 system via syslog.

System name	Normalizer name	Type	Normalizer description
Orion soft zVirt	[OOTB] Orion Soft zVirt Syslog	regexp	Designed for processing events of the Orion soft zVirt 3.1 virtualization system.
PagerDuty	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Palo Alto Cortex Data Lake	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Palo Alto Networks NGFW	[OOTB] Palo Alto NGFW Syslog csv	syslog	Designed for processing events from Palo Alto Networks firewalls received via syslog in CSV format.
Palo Alto Networks PANOS	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Parsec ParsecNet	[OOTB] Parsec ParsecNet	sql	Designed for processing events received by the connector from the database of the Parsec ParsecNet 3 system.
Passwork	[OOTB] Passwork Syslog	syslog	Designed for processing events received from the Passwork version 050219 system via Syslog.
Penta Security WAPPLES	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Positive Technologies ISIM	[OOTB] PTsecurity ISIM	regexp	Designed for processing events from the PT Industrial Security Incident Manager system.
Positive Technologies Network Attack Discovery (NAD)	[OOTB] PT NAD json	json	Designed for processing events coming from PT NAD in json format. This normalizer supports events from PT NAD version 11.1, 11.0.
Positive Technologies Sandbox	[OOTB] PTsecurity Sandbox	regexp	Designed for processing events of the PT Sandbox system.
Positive Technologies Web Application Firewall	[OOTB] PTsecurity WAF	syslog	Designed for processing events from the PTsecurity (Web Application Firewall) system.
Postfix	[OOTB] Postfix Syslog	regexp	Contains a resource set for processing Postfix 3.6 events. It supports processing syslog events received over TCP. The package is available for LogAnalyzer 3.0 and newer versions.
PostgreSQL pgAudit	[OOTB] PostgreSQL pgAudit Syslog	syslog	Designed for processing events of the pgAudit audit plug-n for PostgreSQL database received via syslog.
PowerDNS	[OOTB] PowerDNS Syslog	syslog	Designed for processing events of PowerDNS Authoritative Server 4.5 received via syslog.
Proftpd	[OOTB] Proftpd Syslog	regexp	Designed for processing events received from Proftpd 1.3.8c via Syslog.
Proofpoint Insider Threat Management	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Proxmox	[OOTB] Proxmox file	regexp	Designed for processing events of the Proxmox system version 7.2-3 stored in a file. The normalizer supports processing of events in access and pveam logs.
PT NAD	[OOTB] PT NAD json	json	Designed for processing events coming from PT NAD in json format. This normalizer supports events from PT NAD version 11.1, 11.0.
QEMU - hypervisor logs	[OOTB] QEMU - Hypervisor file	regexp	Designed for processing events of the QEMU hypervisor stored in a file. QEMU 6.2.0 and Libvirt 8.0.0 are supported.
QEMU - virtual machine logs	[OOTB] QEMU - Virtual Machine file	regexp	Designed for processing events from logs of virtual machines of the QEMU hypervisor version 6.2.0, stored in a file.
Radware DefensePro AntiDDoS	[OOTB] Radware DefensePro AntiDDoS	syslog	Designed for processing events from the DDOS Mitigator protection system received via syslog.
Reak Soft Blitz Identity Provider	[OOTB] Reak Soft Blitz Identity Provider file	regexp	Designed for processing events of the Reak Soft Blitz Identity Provider system version 5.16, stored in a file.

System name	Normalizer name	Type	Normalizer description
RedCheck Desktop	[OOTB] RedCheck Desktop file	regexp	Designed for processing logs of the RedCheck Desktop 2.6 system stored in a file.
RedCheck WEB	[OOTB] RedCheck WEB file	regexp	Designed for processing logs of the RedCheck Web 2.6 system stored in files.
RED SOFT RED ADM	[OOTB] RED SOFT RED ADM Syslog	regexp	Designed for processing events received from the RED ADM system (RED ADM: Industrial edition 1.1) via syslog. The normalizer supports processing: • Management subsystem events • Controller events
ReversingLabs N1000 Appliance	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Rubicon Communications pfSense	[OOTB] pfSense Syslog	syslog	Designed for processing events from the pfSense firewall received via syslog.
Rubicon Communications pfSense	[OOTB] pfSense w/o hostname	syslog	Designed for processing events from the pfSense firewall. The syslog header of these events does not contain a hostname.
SailPoint IdentityIQ	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
SecurityCode Continent 3.9	[OOTB] SecurityCode Continent 3.9 json	json	Designed for processing SecurityCode Continent 3.9.2 events received from the kuma-kont utility in json format. This package is available for LogAnalyzer version 3.4 or later.
SecurityCode Continent 4	[OOTB] SecurityCode Continent 4 Syslog	regexp	Designed for processing events of the SecurityCode Continent system version 4 received via syslog.
Sendmail	[OOTB] Sendmail Syslog	syslog	Designed for processing events of Sendmail version 8.15.2 received via syslog.
SentinelOne	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Skype for Business	[OOTB] Microsoft Products for LogAnalyzer 3	xml	Designed for processing some of the events from the log of the Skype for Business system, the Lync Server log.
Snort	[OOTB] Snort 3 json file	json	Designed for processing events of Snort version 3 in JSON format.
Sophos Central	[OOTB] Sophos Central Syslog	syslog	Designed for processing some events received from Sophos Central 1.2 via syslog in CEF format from the Sophos-Central-SIEM-Integration script.
Sonicwall TZ	[OOTB] Sonicwall TZ Firewall	syslog	Designed for processing events received via syslog from the SonicWall TZ firewall.
Solar webProxy	[OOTB] Solar WebProxy Syslog	regexp	Designed for processing events received from Solar webProxy 4.2 in siem-log format via syslog.
SolarWinds DameWare MRC	[OOTB] SolarWinds DameWare MRC xml	xml	Designed for processing some of the DameWare Mini Remote Control (MRC) 7.5 events stored in the Application log of Windows. The normalizer processes events generated by the dwmrcs provider.
SolarWinds SFTP & SCP Server file	[OOTB] SolarWinds SFTP & SCP Server file	regexp	Designed for processing the SolarWinds SFTP & SCP Server 2.2 event log stored in the SftpServer.log file. This package is available for LogAnalyzer version 3.2 or later.
Sophos Firewall	[OOTB] Sophos Firewall Syslog	regexp	Designed for processing events received from Sophos Firewall 20 via syslog.
Sophos XG	[OOTB] Sophos XG	regexp	Designed for processing events from the Sophos XG firewall.
Squid	[OOTB] Squid access Syslog	syslog	Designed for processing events of the Squid proxy server received via the syslog protocol.
Squid	[OOTB] Squid access.log file	regexp	Designed for processing Squid log events from the Squid proxy server. The event source is access.log logs

System name	Normalizer name	Type	Normalizer description
Staffcop Enterprise	[OOTB] Staffcop Enterprise Syslog CEF	regexp	Designed for processing events received from Staffcop Enterprise 5.4, 5.5 in CEF format via syslog.
S-Terra VPN Gate	[OOTB] S-Terra	syslog	Designed for processing events from S-Terra VPN Gate devices.
Suricata	[OOTB] Suricata json file	json	Designed for processing Suricata 7.0.1 events stored in a JSON file. This normalizer supports processing the following event types: flow, anomaly, alert, dns, http, ssl, tls, ftp, ftp_data, ftp, smb, rdp, pgsql, modbus, quic, dhcp, bittorrent_dht, rfb.
ThreatConnect Threat Intelligence Platform	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
ThreatQuotient	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Tionix Cloud Platform	[OOTB] Tionix Cloud Platform Syslog	syslog	Designed for processing events of the Tionix Cloud Platform system version 2.9 received via syslog. The normalizer makes a partial selection of event data. The normalizer is available in LogAnalyzer 3.0 and later.
Tionix VDI	[OOTB] Tionix VDI file	regexp	Designed for processing some of the Tionix VDI system (version 2.8) events stored in the tionix_Intmov.log file.
TrapX DeceptionGrid	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Trend Micro Control Manager	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Trend Micro Deep Security	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Trend Micro NGFW	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Trustwave Application Security DbProtect	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Unbound	[OOTB] Unbound Syslog	syslog	Designed for processing events from the Unbound DNS server received via syslog.
UserGate	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format received from the UserGate system via syslog.
Varonis DatAdvantage	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Veeam Software – Veeam Backup & Replication	[OOTB] Veeam Software Veeam Backup & Replication Syslog	regexp	Designed for processing events from the Veeam Software – Veeam Backup & Replication 12.1 system via ыуslog.
Veriato 360	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
ViPNet TIAS	[OOTB] Vipnet TIAS Syslog	syslog	Designed for processing events of ViPNet TIAS 3.8 received via syslog.
VK WorkSpace Mail	[OOTB] VK WorkSpace Mail Syslog	syslog	Designed for processing events received from the VK WorkSpace Mail 1.23 system via syslog in key-value format.
VMware ESXi	[OOTB] VMware ESXi Syslog	regexp	Designed for processing VMware ESXi events (support for a limited number of events from ESXi versions 5.5, 6.0, 6.5, 7.0) received via syslog.
VMWare Horizon	[OOTB] VMware Horizon - Syslog	syslog	Designed for processing events received from the VMware Horizon 2106 system via syslog.
VMwareCarbon Black EDR	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Vmware Vcenter	[OOTB] VMware Vcenter API	xml	Designed for processing VMware Vcenter 7 events received via API.
Vormetric Data Security Manager	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.

System name	Normalizer name	Type	Normalizer description
Votiro Disarmer for Windows	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Vsftpd	[OOTB] Vsftpd Syslog	regexp	Designed for processing events received from Vsftpd 3.0.5 via Syslog.
Wallix AdminBastion	[OOTB] Wallix AdminBastion Syslog	regexp	Designed for processing events received from the Wallix AdminBastion system via syslog.
WatchGuard - Firebox	[OOTB] WatchGuard Firebox	syslog	Designed for processing WatchGuard Firebox events received via syslog.
Webmonitorx	[OOTB] Webmonitorx json	regexp	Designed for processing events coming from the Webmonitorx system in JSON format.
Webroot BrightCloud	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Windchill FRACAS	[OOTB] PTC Winchill Fracas	regexp	Designed for processing events of the Windchill FRACAS failure registration system.
Yandex Cloud	[OOTB] Yandex Cloud	regexp	Designed for processing part of Yandex Cloud audit events. This normalizer supports processing audit log events of the configuration level: IAM (Yandex Identity and Access Management), Compute (Yandex Compute Cloud), Network (Yandex Virtual Private Cloud), Storage (Yandex Object Storage), Resourcemanager (Yandex Resource Manager).
Zabbix	[OOTB] Zabbix SQL	sql	Designed for processing events of Zabbix 6.4.
Zecurion DLP	[OOTB] Zecurion DLP Syslog	regexp	Designed for processing events of the Zecurion DLP system version 12.0 received via syslog.
ZEEK IDS	[OOTB] ZEEK IDS json file	json	Designed for processing logs of the ZEEK IDS system in JSON format. The normalizer supports events from ZEEK IDS version 1.8.
Zettaset BDEncrypt	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
Zscaler Nanolog Streaming Service (NSS)	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format.
IT-Bastion – Synonyx	[OOTB] Bastion Synonyx CEF	regexp	Designed for processing events in the CEF format received from the IT-Bastion Synonyx 1.7 system. This normalizer package is available for LogAnalyzer version 3.2 or later.
IT-Bastion – SKDPU	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format received from the IT-Bastion SKDPU system via syslog.
A-Real Internet Control Server (ICS)	[OOTB] A-real IKS Syslog	regexp	Designed for processing events of the A-Real Internet Control Server (ICS) system received via syslog. This normalizer supports events from A-Real ICS version 7.0 and later.
Angie web server	[OOTB] Angie Syslog	syslog	Designed for processing events received from Angie and Angie Pro (version 1.10.1) via syslog. This normalizer package is available for LogAnalyzer version 3.2 or later.
Apache web server	[OOTB] Apache HTTP Server file	regexp	Designed for processing Apache HTTP Server 2.4 events stored in a file. The normalizer supports processing of events from the Application log in the Common or Combined Log formats, as well as the Error log. Expected format of the Error log events: `[%t] [%-m:%l] [pid %P:tid %T] [server\ %v] [client\ %a] %E: %M;\ referer\ %-{Referer}i`
Apache web server	[OOTB] Apache HTTP Server Syslog	syslog	Designed for processing events of the Apache HTTP Server received via syslog. The normalizer supports processing of Apache HTTP Server 2.4 events from the Access log in the Common or Combined Log format, as well as the Error log. Expected format of the Error log events: `[%t] [%-m:%l] [pid %P:tid %T] [server\ %v] [client\ %a] %E: %M;\ referer\ %-{Referer}i`
Lighttpd web server	[OOTB] Lighttpd Syslog	syslog	Designed for processing Access events of the Lighttpd system received via syslog. The normalizer supports processing of Lighttpd version 1.4 events. Expected format of Access log events: `$remote_addr $http_request_host_name $remote_user [$time_local] "$request" $status $body_bytes_sent "$http_referer" "$http_user_agent"`

System name	Normalizer name	Type	Normalizer description
IVK Kolchuga-K	[OOTB] Kolchuga-K Syslog	syslog	Designed for processing events from the IVK Kolchuga-K system, version LKNV.466217.002, via syslog.
infotecs ViPNet IDS	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format received from the infotecs ViPNet IDS system via syslog.
infotecs ViPNet Coordinator	[OOTB] VipNet Coordinator Syslog	syslog	Designed for processing events from the ViPNet Coordinator system received via syslog.
Kod Bezopasnosti — Continent	[OOTB][regexp] Continent IPS/IDS & TLS	regexp	Designed for processing events of Continent IPS/IDS device log.
Kod Bezopasnosti — Continent	[OOTB] Continent SQL	sql	Designed for getting events of the Continent system from the database.
Kod Bezopasnosti SecretNet 7	[OOTB] SecretNet SQL	sql	Designed for processing events received by the connector from the database of the SecretNet system.
Confident – Dallas Lock Unified Control Center	[OOTB] Confident Dallas Lock Syslog CEF	regexp	Designed for processing events received from Dallas Lock Unified Control Center (version 4.0) in CEF format.
Confident - Dallas Lock	[OOTB] Confident Dallas Lock Syslog CEF	regexp	Designed for processing events received from Dallas Lock 8 (version 11.4.2) in CEF format.
CryptoPro NGate	[OOTB] Ngate Syslog	syslog	Designed for processing events received from the CryptoPro NGate system via syslog.
H3C (Huawei-3Com) routers	[OOTB] H3C Routers Syslog	regexp	Designed for processing events received from H3C (Huawei-3Com) SR6600 network devices (Comware 7 firmware) via syslog. This normalizer supports the "standard" event format (RFC 3164-compliant format).
NT Monitoring and Analytics	[OOTB] Syslog-CEF	syslog	Designed for processing events in the CEF format received from the NT Monitoring and Analytics system via syslog.
BlueCoat proxy server	[OOTB] BlueCoat Proxy v0.2	regexp	Designed to process BlueCoat proxy server events. The event source is the BlueCoat proxy server event log.
SKDPU NT Access Gateway	[OOTB] Bastion SKDPU-GW Syslog	syslog	Designed for processing events of the SKDPU NT Access gateway 7.0 system received via syslog.
Solar Dozor	[OOTB] Solar Dozor Syslog	syslog	Designed for processing events received from the Solar Dozor system version 7.9 via syslog. This normalizer supports custom format events and does not support CEF format events.
Yandex 360	[OOTB] Yandex 360 json	json	Designed for processing events received from the Yandex 360 system in JSON format. The transport for receiving events will be available in LogAnalyzer 4.2.
Yandex Browser corporate	[OOTB] Yandex Browser	json	Designed for processing events received from the corporate version of Yandex Browser 23, 24.4, 25.2.
-	[OOTB] Syslog header	syslog	Designed for processing events received via syslog. The normalizer parses the header of the Syslog event, the message field of the event is not parsed. If necessary, you can parse the message field using other normalizers.

Aggregation rules

Aggregation rules let you combine repetitive events of the same type and replace them with one common event. Aggregation rules support fields of the standard LogAnalyzer event schema as well as fields of the extended event schema. In this way, you can reduce the number of similar events sent to the storage, reduce the workload on services, conserve data storage space and licensing quota (EPS). An aggregation event is created when a time or number of events threshold is reached, whichever occurs first.

For aggregation rules, you can configure a filter and apply it only to events that match the specified conditions.

You can configure aggregation rules under **Resources** → **Aggregation rules**, and then select the created aggregation rule from the drop-down list in the collector settings. You can also configure aggregation rules directly in collector settings. Available aggregation rule settings are listed in the table below.

Available aggregation rule settings

Setting	Description
Name	Unique name of the resource. Maximum length of the name: 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Threshold	Threshold on the number of events. After accumulating the specified number of events with identical fields, the collector creates an aggregation event and begins accumulating events for the next aggregated event. The default value is `100`.
Triggered rule lifetime	Threshold on time in seconds. When the specified time expires, the accumulation of base events stops, the collector creates an aggregated event and starts obtaining events for the next aggregated event. The default value is `60`. Required setting.
Description	Description of the resource. Maximum length of the description: 4000 Unicode characters.
Identical fields	Fields of normalized events whose values must match. For example, for network events, `SourceAddress`, `DestinationAddress`, and `DestinationPort` normalized event fields can be used. In the aggregation event, these normalized event fields are populated with the values of the base events. Required setting.
Unique fields	Normalized event fields whose values must be unique for those events to be considered in an aggregated event. If the value of a field specified under **Unique Fields** is duplicated in multiple events, only one of such events is included in the aggregated event, and the duplicates are ignored. If you specify more than one field under **Unique Fields**, the combinations of values of all of these fields must be unique. Unique fields are not saved in the aggregated event.

Setting	Description
Sum fields	Fields whose values are summed up during aggregation and written to the same-name fields of the aggregated event. The following special considerations are relevant to field behavior: • The values of fields of the `Number` and `Float` types are summed up. • The values of fields of the `String` type are concatenated with commas added as separators. • The values of fields with the types `Array of strings`, `Array of numbers` and `Array of floats` are appended to the end of the array.
Filter	Conditions for determining which events must be processed by the resource. In the drop-down list, you can select an existing filter **Create new** to create a new filter. In aggregation rules, do not use filters with the **TI** operand or the **TIDetect**, **inActiveDirectoryGroup**, or **hasVulnerability** operators. The `Active Directory` fields for which you can use the **inActiveDirectoryGroup** operator will appear during the enrichment stage (after aggregation rules are executed). Creating a filter in resources

Setting	Description
	To create a filter: 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

Setting	Description
	4. Under **Conditions**, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, fields of additional parameters for identifying the value to be passed to the filter may be displayed. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **Operator** drop-down list, select an operator.

Setting	Description

Filter operators

- **=** means the left operand equals the right operand.
- **<** means the left operand is less than the right operand.
- **<=** means the left operand is less than or equal to the right operand.
- **>** means the left operand is greater than the right operand.
- **>=** means the left operand is greater than or equal to the right operand.
- **inSubnet** means the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains** means the left operand contains values of the right operand.
- **startsWith** means the left operand starts with one of the values of the right operand.
- **endsWith** means the left operand ends with one of the values of the right operand.
- **match** means the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit** checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
- **hasVulnerability** checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- The **inActiveList** operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary** checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory** means the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup** means the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

Setting	Description
	• The **TIDetect** operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage. • **inContextTable** checks for the presence of the entry in the specified context table. • **intersect** checks for the presence in the left operand of the list items specified in the right operand. d. If you want the operator to be case-insensitive, select the **do not match case** check box. This check box does not apply to the **InSubnet**, **InActiveList**, **InCategory**, or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, from the **If** drop-down list, select **If not**. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Enrichment rules

Event enrichment involves adding information to events that can be used to identify and investigate an incident.

Enrichment rules let you add supplementary information to event fields by transforming data that is already present in the fields, or by querying data from external systems. For example, suppose that a user name is recorded in the event. You can use an enrichment rule to add information about the department, position, and manager of this user to the event fields.

Enrichment rules can be used in the following LogAnalyzer services and features:

- Collector. In the collector, you can create an enrichment rule, and it becomes a resource that you can reuse in other services. You can also link an enrichment rule created as a standalone resource.
- Normalizer. In the normalizer, you can only create an enrichment rule linked to that normalizer. Such a rule will not be available as a standalone resource for reuse in other services.

Available enrichment rule settings are listed in the table below.

Basic settings tab

Setting	Description

Source kind

Required setting.

Drop-down list for selecting the type of incoming events. Depending on the selected type, the following additional settings will be displayed:

- constant

  This type of enrichment is used when a constant needs to be added to an event field. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Constant	The value to be added to the event field. Maximum length of the value: 255 Unicode characters. If you leave this field blank, the existing event field value is removed.
Target field	The LogAnalyzer event field that you want to populate with the data.

  If you are using the event enrichment functions for extended schema fields of `String`, `Number`, or `Float` type with a constant, the constant is added to the field.

  If you are using the event enrichment functions for extended schema fields of `Array of strings`, `Array of numbers`, or `Array of floats` type with a constant, the constant is added to the elements of the array.

- dictionary (available only in the commercial version of LogAnalyzer)

  This type of enrichment is used if you need to add a value from the dictionary of the **Dictionary** type. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Dictionary name	The dictionary from which the values are to be taken.
Key fields	Event fields whose values are to be used for selecting a dictionary entry. To add an event field, click **Add field**. You can add multiple event fields.

  If you are using event enrichment with the **dictionary** type selected as the **Source kind** setting, and an array field is specified in the **Key enrichment fields** setting, when an array is passed as the dictionary key, the array is serialized into a string in accordance with the rules of serializing a single value in the TSV format.

  Example: The **Key fields** setting of the **enrichment** uses the `SA.StringArrayOne` extended schema field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`. The following values are passed to the dictionary as the key: `['a','b','c']`.

  If the **Key enrichment fields** setting uses an extended schema array field and a regular event schema field, the field values are separated by the `|` character when the dictionary is queried.

  Example: The **Key enrichment fields** setting uses the `SA.StringArrayOne` extended schema field and the `Code` string field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`, and the `Code` string field contains the `myCode` sequence of characters. The following values are passed to the dictionary as the key: `['a','b','c']|myCode`.

- table (available only in the commercial version of LogAnalyzer)

This type of enrichment is used if you need to add a value from the dictionary of the **Table** type. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Dictionary name	The dictionary from which the values are to be taken.
Key fields	Event fields whose values are to be used for selecting a dictionary entry. To add an event field, click **Add field**. You can add multiple event fields.
Mapping	Event fields for data transfer: • **Dictionary field** specifies dictionary fields from which data is to be transmitted. The available fields depend on the selected dictionary resource. • **KUMA field** specifies event fields to which data is to be transmitted. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written there.

The first field in the table (**Dictionary field**) is taken as the key with which the fields selected from the event as key fields are matched (**KUMA field**). As the key in the **Dictionary field**, you must select an indicator of compromise by which the enrichment is to be performed, for example, IP address, URL, or hash. In the rule, you must select the event field that corresponds to the selected indicator of compromise in the dictionary field.

If you want to select multiple key fields, you can specify them using `|` as a separator (when specifying in the web interface or importing as a CSV file), for example, `<IP address>|<user name>`.

You can add new table rows or delete table rows. To add a new table row, click **Add new element**. To delete a row in the table, click the ✕ button.

- event

This type of enrichment is used when you need to write a value from another event field to the current event field. Settings of this type of enrichment:

- In the **Target field** drop-down list, select the LogAnalyzer event field to which you want to write the data.
- In the **Source field** drop-down list, select the event field whose value will be written to the target field.
- Under **Conversion**, you can create rules for modifying the original data before it is written to the LogAnalyzer event fields. The conversion type can be selected from the drop-down list. You can use the **Add conversion** and **Delete** buttons to add or delete a conversion, respectively. The order of conversions is important.

  Available conversions

Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.

- **lower**—is used to make all characters of the value lowercase

- **upper**—is used to make all characters of the value uppercase

- **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.

- **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.

- **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:

  - **Replace chars** specifies the sequence of characters to be replaced.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if for a `Microsoft-Windows-Sysmon` value, you perform a **trim** conversion with a value of `Micromon`, the new value is `soft-Windows-Sys`.

- **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:

  - **Expression** is the RE2 regular expression whose results you want to replace.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.

- Converting the IP address to the IPv4 format:
  - **ipDecimalToDotted** converts an IP address in decimal format to an IP address in IPv4 format, in which octets are separated by dots.
  - **ipHexToDotted** converts a hexadecimal IP address to a decimal IP address in IPv4 format, in which octets are separated by dots.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append**, **prepend**.

- template

This type of enrichment is used when you need to write the result of processing Go templates into the event field. We recommend matching the value and the size of the field. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Template	The Go template. Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field from which the value must be passed to the script, for example, `{{.DestinationAddress}} attacked from {{.SourceAddress}}`.
Target field	The LogAnalyzer event field that you want to populate with the data.

If you are using enrichment of events that have **template** selected as the **Source kind**, and in which the target field has the `String` type, and the source field is an extended event schema field containing an array of strings, you can use one of the following examples for the template:

- `{{.SA.StringArrayOne}}`
- ```
  {{- range $index, $element := . SA.StringArrayOne -}}
  {{- if $index}}, {{end}}"{{$element}}"{{- end -}}
  ```

To convert the data in an array field in a template into the TSV format, use the `toString` function, for example:

```
template {{toString.SA.StringArray}}
```

- dns (available only in the commercial version of LogAnalyzer)
```

This type of enrichment is used to send requests to a private network DNS server to convert IP addresses into domain names or vice versa. IP addresses are converted to DNS names only for private addresses: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 100.64.0.0/10.

Available settings:

- **URL**—in this field, you can specify the URL of a DNS server to which you want to send requests. You can use the **Add URL** button to specify multiple URLs.
- **RPS**—maximum number of requests sent to the server per second. The default value is `1,000`.
- **Workers** is the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.
- **Max tasks**—maximum number of simultaneously fulfilled requests. By default, this value is equal to the number of vCPUs of the LogAnalyzer Core server.
- **Cache TTL**—the lifetime (in seconds) of the values stored in the cache. The default value is `60` (seconds).
- **Cache disabled**—you can use this toggle switch to enable or disable caching. Caching is enabled by default.
- **Cache update disabled**—if cache update is enabled, then after the cache TTL expires, the collector sends repeat queries to the DNS server to update the information stored in the cache. This setting is available only when cache is enabled. By default, cache update is enabled.
- The **Recursive queries** setting is available starting with LogAnalyzer 3.4.1. You can use this toggle switch to make a LogAnalyzer collector send recursive queries to authoritative DNS servers for the purposes of enrichment. The default value is **Disabled**.

- **cybertrace** (available only in the commercial version of LogAnalyzer)

  > For systems with a high load on the collector, we recommend using the cybertrace-http enrichment type. This type returns a structured response in JSON format and correctly distinguishes between strings and arrays of custom attributes, preventing parsing errors.

  > When enriched through cybertrace, the collector gets an unstructured response. If the value of a custom attribute in CyberTrace contains square brackets, for example, **[str1**, **str2]**, LogAnalyzer treats it as an array, not a string.

  This type of enrichment is used to add information from CyberTrace data streams to event fields.

  Available settings:

  - **URL** (required)—in this field, you can specify the URL of a CyberTrace server to which you want to send requests. The default CyberTrace port is 9999.
  - **Number of connections**—maximum number of connections to the CyberTrace server that can be simultaneously established by LogAnalyzer. By default, this value is equal to the number of vCPUs of the LogAnalyzer Core server.
  - **RPS**—maximum number of requests sent to the server per second. The default value is `1,000`.
  - **Timeout**—amount of time to wait for a response from the CyberTrace server, in seconds. The default value is `30`.
  - **Maximum number of events in the enrichment queue**—maximum number of events stored in the enrichment queue for re-sending. The default value is 1,000,000.
  - **Mapping** (required)—this section of settings contains the mapping table for mapping LogAnalyzer event fields to CyberTrace indicator types. The **LogAnalyzer field** column shows the names of LogAnalyzer event fields, and the **CyberTrace indicator** column shows the types of CyberTrace indicators.

    Available types of CyberTrace indicators:

    - **ip**
    - **url**
    - **hash**

    In the mapping table, you must provide at least one string. You can use the **Add row** button to add a string, and can use the ✕ button to remove a string.

- **cybertrace-http** (available only in the commercial version of LogAnalyzer)

This is a new streaming event enrichment type in CyberTrace that allows you to send a large number of events with a single request to the CyberTrace API. Recommended for systems with a lot of events. The performance of cybertrace-http is superior to that of the cybertrace type.

Limitations:

- The cybertrace-http enrichment type cannot be used for retroscan in KUMA.
- If the cybertrace-http enrichment type is being used, detections are not saved in CyberTrace history in the Detections window.

Available settings:

- **Name** (required) is a unique name for the enrichment rule. Up to 128 Unicode characters.
- **Tenant** (required) is the name of the tenant in which you plan to work with CyberTrace.
- **URL** (required)—in this field, you can specify the URL of a CyberTrace server to which you want to send requests and the port that CyberTrace API is using. The default port is 443.
- **Secret** (required) is a drop-down list in which you can select the secret which stores the credentials for the connection.
- **Timeout**—amount of time to wait for a response from the CyberTrace server, in seconds. The default value is `30`.
- **Key fields** (required) is the list of event fields used for enriching events with data from CyberTrace.
- **Maximum number of events in the enrichment queue**—maximum number of events stored in the enrichment queue for re-sending. The default value is 1,000,000. After reaching 1 million events received from the CyberTrace server, events stop being enriched until the number of received events is reduced to less than 500,000.
- **Major version of CyberTrace** selects the CyberTrace version. Possible values: **<5**, **≥ 5.0**. If version **≥ 5.0** is selected, requests are executed in the General tenant in CyberTrace and use the new version of the API provided by Kaspersky CyberTrace 5.0 and later. If you select **<5**, requests use the Kaspersky CyberTrace 4.x API. The default value is **<5**.
- The **Debug** toggle switch enables resource logging. The toggle switch is turned off by default.
- **Tags** for resource search.

- timezone

This type of enrichment is used in collectors to assign a specific timezone to an event. Timezone information may be useful when searching for events that occurred at unusual times, such as nighttime.

When this type of enrichment is selected, the required timezone must be selected from the **Timezone** drop-down list.

> Make sure that the required time zone is set on the server hosting the enrichment-utilizing service. For example, you can do this by using the `timedatectl list-timezones` command, which shows all time zones that are set on the server. For more details on setting time zones, please refer to your operating system documentation.

When an event is enriched, the time offset of the selected timezone relative to Coordinated Universal Time (UTC) is written to the DeviceTimeZone event field in the `+-hh:mm` format. For example, if you select the **Asia/Yekaterinburg** timezone, the value `+05:00` will be written to the DeviceTimeZone field. If the enriched event already has a value in the DeviceTimeZone field, it will be overwritten.

By default, if the timezone is not specified in the event being processed and enrichment rules by timezone are not configured, the event is assigned the timezone of the server hosting the service (collector) that processes the event. If the server time is changed, the service must be restarted.

Permissible time formats when enriching the DeviceTimeZone field

> When processing incoming raw events in the collector, the following time formats can be automatically converted to the +-hh:mm format:
>
> | Time format in a processed event | Example |
> |---|---|
> | +-hh:mm | -07:00 |
> | +-hhmm | -0700 |
> | +-hh | -07 |
>
> If the date format in the `DeviceTimeZone` field differs from the formats listed above, the collector server timezone is written to the field when an event is enriched with timezone information. You can create custom normalization rules for non-standard time formats.

- geographic data

This type of enrichment is used to add IP address geographic data to event fields. Learn more about linking IP addresses to geographic data.

When this type is selected, under **Mapping geographic data to event fields**, you must specify from which event field the IP address will be read, select the required attributes of geographic data, and define the event fields in which geographic data will be written:

1. In the **Event field with IP address** drop-down list, select the event field from which the IP address is read. Geographic data uploaded to LogAnalyzer is matched against this IP address.

   You can use the **Add event field with IP address** button to specify multiple event fields with IP addresses that require geographic data enrichment. You can delete event fields added in this way by clicking the **Delete event field with IP address** button.

   > When the `SourceAddress`, `DestinationAddress`, and `DeviceAddress` event fields are selected, the **Apply default mapping** button becomes available. You can use this button to add preconfigured mapping pairs of geographic data attributes and event fields.

2. For each event field you need to read the IP address from, select the type of geographic data and the event field to which the geographic data should be written.

   You can use the **Add geodata attribute** button to add field pairs for **Geodata attribute** – **Event field to write to**. You can also configure different types of geographic data for one IP address to be written to different event fields. To delete a field pair, click ×.

   - In the **Geodata attribute** field, select which geographic data corresponding to the read IP address should be written to the event. Available geographic data attributes: **Country**, **Region**, **City**, **Longitude**, **Latitude**.
   - In the **Event field to write to**, select the event field which the selected geographic data attribute must be written to.

   You can write identical geographic data attributes to different event fields. If you configure multiple geographic data attributes to be written to the same event field, the event will be enriched with the last mapping in the sequence.

- check DLL Hijacking

DLL Hijacking is an attack technique that involves delivering vulnerable legitimate software along with a malicious dynamic link library (DLL) to the target system. To detect such attacks, LogAnalyzer uses the AI module. The AI module analyzes the launch and runtime parameters of applications and identifies suspicious launches of legitimate software with malicious libraries.

To get a status, LogAnalyzer sends a request to KSN with information from event fields. In response to the request, KSN sends the analysis result.

You can configure DLL Hijacking attack detection in the collector.

To use an enrichment of the **check DLL Hijacking** type, your license must include the AI module. The General administrator must also accept the additional KSN license agreement in the **Kaspersky Security Network** section (**Settings** → **Integrations** → **AI services** → **Kaspersky Security Network**).

Available settings:

- **Workers** is the number of workers that the service can run simultaneously to process events in parallel. Specify a positive integer number from 0 to 999. If 0 is specified, the number of logical CPU cores is used.
- **Maximum cache size** is the maximum cache size for KSN responses, in megabytes. The default value is `128 MB`.

  Each KSN response has its own lifetime, during which it is retained in the cache. If LogAnalyzer sends a second request for a DLL file for which KSN has already issued a response, and the lifetime of that response has not yet expired, the request is not actually sent to KSN, but the response is taken from the cache.
- If the **Maximum number of events in enrichment queue** is exceeded, new events are not processed. The default value is `1,000,000`. `0` means that event processing stops if the queue contains any number of events.

- **Perform a repeat request**. KSN may return the **Unclassified** status. This status means that KSN has accepted the request and is currently processing it, but there is no status yet. Processing may take a long time, so the request must be re-sent. If this setting is enabled, specify additional settings:

  - If the **Create duplicates** setting is enabled, LogAnalyzer creates a duplicate event and uses it for repeated requests to KSN. In this case, the main event is routed in accordance with the workflow. A duplicate event is written to the database when one of the following conditions is satisfied:

    - A status other than **Unclassified** is received from KSN.
    - The disk buffer configured using the **Buffer size** setting is full.
    - The **Request count** is exceeded.

    If this setting is disabled, LogAnalyzer continues to send requests using the main event. The event is written to the database when one of the following conditions is satisfied:

    - A status other than **Unclassified** is received from KSN.
    - The disk buffer configured using the **Buffer size** setting is full.
    - The **Request count** is exceeded.

  - The **Request count** is the maximum number of requests to KSN. If, after the specified number of requests, a status other than **Unclassified** is not received, the event is written to the database with the **Unclassified** status. The default setting is `10`.

  - **Poll interval** is the interval between requests, in minutes. The default value is `1 minute`. The minimum value is `1 minute`.

  - **Buffer size** is the size of the disk buffer for events awaiting a repeat request, in megabytes. The default value is `10240 MB`. The minimum value is `10 MB`.

- **Proxy** specifies proxy server settings, if one is needed to connect to the KSN server. You can select one of the available proxy servers or create a new proxy server. By default, a proxy server is not used.

- The **Mapping** table contains a list of parameters that must be sent in a KSN request to get a status, and a list of the corresponding LogAnalyzer event fields. List of parameters in the request:

  - **dll hash (md5)**. Recommended field for mapping: OldFileHash.

  - **dll path**. Full path including the file name; recommended field for mapping: OldFilePath. The path may contain personal data, so LogAnalyzer sends this parameter to KSN in an anonymized form. For example:

    Path before anonymization:
    `C:\Windows\Device\HarddiskVolume1\Users\Alex\malware.dll`

    The same path in anonymized form: `device\harddiskvolume1\users\username`

  - **process hash (md5)**. Recommended field for mapping: FileHash.

  - **process path**. Full path including the file name. The path may contain personal data, so LogAnalyzer sends this parameter to KSN in an anonymized form. For example:

    Path before anonymization:
    `C:\Windows\Device\HarddiskVolume1\Users\Alex\malware.dll`

    The same path in anonymized form: `device\harddiskvolume1\users\username`

  - **dll name**. Recommended field for mapping: OldFileName.

  - **process name**. Recommended field for mapping: FileName.

  - **process signature**. Optional; recommended field for mapping: FileID.

  - **dll size**. Optional; recommended field for mapping: OldFileSize.

  For each required parameter, specify a corresponding LogAnalyzer event field in the table.

Setting	Description
Setting	Description
Setting	Description
Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
Debug	You can use this toggle switch to enable the logging of service operations. Logging is disabled by default.

Description	Resource description: up to 4,000 Unicode characters.
Filter	Group of settings in which you can specify the conditions for identifying events that must be processed by this resource. You can select an existing filter from the drop-down list or **create** a new filter. Creating a filter in resources

To create a filter:

1. In the **Filter** drop-down list, select **Create new**.

2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.

3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

4. Under **Conditions**, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, fields of additional parameters for identifying the value to be passed to the filter may be displayed. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

   c. In the **Operator** drop-down list, select an operator.

Filter operators

- **=** means the left operand equals the right operand.
- **<** means the left operand is less than the right operand.
- **<=** means the left operand is less than or equal to the right operand.
- **>** means the left operand is greater than the right operand.
- **>=** means the left operand is greater than or equal to the right operand.
- **inSubnet** means the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains** means the left operand contains values of the right operand.
- **startsWith** means the left operand starts with one of the values of the right operand.
- **endsWith** means the left operand ends with one of the values of the right operand.
- **match** means the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit** checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
- **hasVulnerability** checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- The **inActiveList** operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary** checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory** means the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup** means the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- The **TIDetect** operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage.
- **inContextTable** checks for the presence of the entry in the specified context table.
- **intersect** checks for the presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. This check box does not apply to the **InSubnet**, **InActiveList**, **InCategory**, or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, from the **If** drop-down list, select **If not**.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Predefined enrichment rules

The LogAnalyzer distribution kit includes enrichment rules listed in the table below.

Predefined enrichment rules

Enrichment rule name	Description
[OOTB] KATA alert	Used to enrich events received from KATA in the form of a hyperlink to an alert. The hyperlink is put in the DeviceExternalId field.

Filters

Filters let you select events based on specified conditions. The collector service uses filters to select events that you want to send to LogAnalyzer. Events that satisfy the filter conditions are sent to LogAnalyzer for further processing.

You can use filters in the following LogAnalyzer services and features:

- Collector.
- Storage.
- Enrichment rules.

- Aggregation rules.
- Destinations.

You can use standalone filters or built-in filters that are stored in the service or resource in which they were created. For resources in input fields except the **Description** field, you can enable the display of control characters. Available filter settings are listed in the table below.

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting. Inline filters are created in other resources or services and do not have names.
Tenant	The name of the tenant that owns the resource. Required setting.
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

You can create filter conditions and filter groups, or add existing filters to a filter.

To create filtering criteria, you can use *builder mode* or *source code mode*. In builder mode, you can create or edit filter criteria by selecting filter conditions and operators from drop-down lists. In source code mode, you can use text commands to create and edit search queries. The builder mode is used by default.

You can freely switch between modes when creating filtering criteria. To switch to source code mode, select the **Code** tab. When switching between modes, the created condition filters are preserved. If the filter code is not displayed on the **Code** tab after linking the created filter to the resource, go to the **Builder** tab and then go back to the **Code** tab to display the filter code.

Creating filtering criteria in builder mode

To create filtering criteria in builder mode, you need to select one of the following operators from the drop-down list:

- **AND**: The filter selects events that match all of the specified conditions.
- **OR**: The filter selects events that match one of the specified conditions.
- **NOT**: The filter selects events that match none of the specified conditions.

You can add filtering criteria in one of the following ways:

- To add a condition, click the **+ Add condition** button.
- To add a group of conditions, click the **+ Add group** button. When adding groups of conditions, you can also select the **AND**, **OR**, and **NOT** operators. In turn, you can add conditions and condition groups to a condition group.

You can add multiple filtering criteria, reorder the filtering criteria, or remove filtering criteria. To reorder filtering criteria, use the reorder ⣿ icons. To remove a filtering criterion, click the delete ✕ icon next to it.

Available condition settings are listed in the table below.

Setting	Description
<Condition type>	Condition type. The default is **If**. You can click the default value and select **If not** from the displayed drop-down list. Required setting.
<Left operand> and **<Right operand>**	Values to be processed by the operator. The available types of values of the right operand depend on the selected operator.

Setting	Description
	Operands of filters • In the **Event fields** section, you can specify the event field that you want to use as a filter operand. • In the **Active lists** section, you can specify an active list or field of an active list that you want to use as an operand of the filter. When selecting an active list, you must specify one or more event fields that are used to create an active list entry and act as the key of the active list entry. To finish specifying event fields, press **Ctrl**/**Command**+**F1**. If you have not specified the **inActiveList** operator, you need to specify the name of the active list field that you want to use as a filter operand. • In the **Context tables** section, you can specify the value of the context table that you want to use as the filter operand. When selecting a context table, you must specify an event field: • **context table name** (required) is the context table that you want to use. • **key fields** (required) are event fields or local variables that are used to create a context table record and serve as the key for the context table record. • **field** is the name of the context table field from which you want to get the value of the operand. • **index** is the index of the list field of the table from which you want to get the value of the operand. • **Dictionary** is a value from the dictionary resource that you want to assign to the operand. Advanced settings: • **dictionary** (required) is the dictionary that you want to use. • **key fields** (required) are event fields that are used to generate the key of the dictionary value. • **Constant** is a user-defined value that you want to assign to the operand. Advanced settings: • **value** (required) is the constant that you want to assign to the operand. • **Table** specifies user-defined values that you want to assign to the operand. Advanced settings: • **dictionary** (required) is the type of the dictionary. You need to select the **Table** dictionary type. • **key fields** (required) are event fields that are used to generate the key of the dictionary value.

Setting	Description

	• **List** specifies user-defined values that you want to assign to the operand. Advanced settings:
	  • **value** (required) are the constants that you want to assign to the operand. When you type the value in the field and press **ENTER**, the value is added to the list and you can enter a new value.
	• **TI** specifies the settings for reading CyberTrace threat intelligence (TI) data from the events. Advanced settings:
	  • **stream** (required) is the CyberTrace threat category.
	  • **key fields** (required) is the event field with CyberTrace threat indicators.
	  • **field** (required) is the field of the CyberTrace feed with threat indicators.
	
	Required settings.

<Operator>	Condition operator. When selecting a condition operator in the drop-down list, you can select the **do not match case** check box if you want the operator to ignore the case of values. This check box is ignored if the **inSubnet**, **inActiveList**, **inCategory**, **InActiveDirectoryGroup, hasBit**, and **inDictionary** operators are selected. By default, this check box is cleared.

Setting	Description
	Filter operators • **=** means the left operand equals the right operand. • **<** means the left operand is less than the right operand. • **<=** means the left operand is less than or equal to the right operand. • **>** means the left operand is greater than the right operand. • **>=** means the left operand is greater than or equal to the right operand. • **inSubnet** means the left operand (IP address) is in the subnet of the right operand (subnet). • **contains** means the left operand contains values of the right operand. • **startsWith** means the left operand starts with one of the values of the right operand. • **endsWith** means the left operand ends with one of the values of the right operand. • **match** means the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit** checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability** checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • The **inActiveList** operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary** checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory** means the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup** means the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

Setting	Description
	• The **TIDetect** operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage. • **inContextTable** checks for the presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand. You can change or delete the specified operator. To change the operator, click it and specify a new operator. To delete the operator, click it, then press **Backspace**.

The available operand kinds depends on whether the operand is left (L) or right (R).

Available operand kinds for left (L) and right (R) operands

Operator	Event field type	Dictionary type	Table type	TI type	Constant type	List type
=	L, R	L, R	L, R	L, R	R	R
>	L, R	L, R	L, R	L	R	—
>=	L, R	L, R	L, R	L	R	—
<	L, R	L, R	L, R	L	R	—
<=	L, R	L, R	L, R	L	R	—
inSubnet	L, R	L, R	L, R	L, R	R	R
contains	L, R	L, R	L, R	L, R	R	R
startsWith	L, R	L, R	L, R	L, R	R	R
endsWith	L, R	L, R	L, R	L, R	R	R
match	L	L	L	L	R	R
hasVulnerability	L	L	L	—	—	—
hasBit	L	L	L	—	R	R
inActiveList	—	—	—	—	—	—
inDictionary	—	—	—	—	—	—
inCategory	L	L	L	—	R	R
inContextTable	—	—	—	—	—	—
inActiveDirectoryGroup	L	L	L	—	R	R
TIDetect	—	—	—	—	—	—

You can use hotkeys when managing filters. Hotkeys are described in the table below.

Hotkeys and their functions

Key	Function
`e`	Invokes a filter by the event field
`d`	Invokes a filter by the dictionary field
`t`	Invokes a filter by the table field
`f`	Invokes a filter
`t+i`	Invokes a filter using TI
`Ctrl+Enter`	Finish editing a condition

The usage of extended event schema fields of the `String`, `Number`, or `Float` types is the same as the usage of fields of the LogAnalyzer event schema.

When using filters with extended event schema fields of the `Array of strings`, `Array of numbers`, and `Array of floats` types, you can use the following operations:

- The `contains` operation returns `True` if the specified substring is present in the array, otherwise it returns `False`.
- The `match` operation matches the string against a regular expression.
- The `intersec` operation.

Creating filtering criteria in source code mode

The source code mode allows you to quickly edit conditions, select and copy blocks of code. In the right part of the builder, you can find the navigator, which lets you to navigate the filter code. Line wrapping is performed automatically at **AND**, **OR**, **NOT** logical operators, or at commas that separate the items in the list of values.

Names of resources used in the filter are automatically specified. Fields containing the names of linked resources cannot be edited. The names of shared resource categories are not displayed in the filter if you do not have the "Read shared resources" role. To view the list of resources for the selected operand inside the expression, press **`Ctrl+Space`**. This displays a list of resources.

The filters listed in the table below are included in the LogAnalyzer distribution kit.

Predefined filters

Filter name	Description
[OOTB][AD] A member was added to a security-enabled global group (4728)	Selects events of adding a user to an Active Directory security-enabled global group.
[OOTB][AD] A member was added to a security-enabled universal group (4756)	Selects events of adding a user to an Active Directory security-enabled universal group.
[OOTB][AD] A member was removed from a security-enabled global group (4729)	Selects events of removing a user from an Active Directory security-enabled global group.
[OOTB][AD] A member was removed from a security-enabled universal group (4757)	Selects events of removing a user from an Active Directory security-enabled universal group.
[OOTB][AD] Account Created	Selects Windows user account creation events.
[OOTB][AD] Account Deleted	Selects Windows user account deletion events.
[OOTB][AD] An account failed to log on (4625)	Selects Windows logon failure events.
[OOTB][AD] Successful Kerberos authentication (4624, 4768, 4769, 4770)	Selects successful Windows logon events and events with IDs 4769, 4770 that are logged on domain controllers.
[OOTB][AD][Technical] 4768. TGT Requested	Selects Microsoft Windows events with ID 4768.
[OOTB][Net] Possible port scan	Selects events that may indicate a port scan.
[OOTB][SSH] Accepted Password	Selects events of successful SSH connections with a password.
[OOTB][SSH] Failed Password	Selects attempts to connect over SSH with a password.

Proxies

Proxy servers are used to store proxy server configuration settings, for example, in destinations. The http type is supported. Available proxy server settings are listed in the table below.

Available proxy server settings

Setting	Description
Name	Unique name of the proxy server. Maximum length of the name: 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Secret separately	Viewing information about the connection. If this check box is selected, the following settings are displayed in the window: • **URL** is the connection URL. • **Secret** is the secret of the 'credentials' type. This lets you view connection information without having to re-create a large number of connections if the password of the user account that you used for the connections changes. This check box is cleared by default.
Use URL from the secret	The secret resource that stores URLs of proxy servers. You can create or edit a secret. To create a secret, click the plus (+) icon. To edit a secret, click the pencil ( ) icon. We do not recommend using this option. Instead, manually specify the URL and the secret separately by selecting the **Secret separately** check box.
Do not use for domains	One or more domains to which direct access is required.
Description	Description of the proxy server. Maximum length of the description: 4000 Unicode characters.

Dictionaries

This functionality is only available in the commercial version of LogAnalyzer.

Description of parameters

Dictionaries are resources storing data that can be used by other LogAnalyzer resources and services. Dictionaries can be used in the following LogAnalyzer services and features:

- Collector.
- Normalizers.

Available dictionary settings are listed in the table below.

Available dictionary settings

Setting	Description
Name	Unique name for this resource type. Maximum length of the name: 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Description	Description of the resource. Maximum length of the description: 4000 Unicode characters.

Setting	Description
Type	Dictionary type. The selected dictionary type determines the format of the data that the dictionary can contain: • You can add key-value pairs to the **Dictionary** type. We do not recommend adding more than 50,000 entries to dictionaries of this type using the LogAnalyzer web interface. When adding lines with the same keys to the dictionary, each new line will overwrite the existing line with the same key. This means that only one line will be added to the dictionary. • Data in the form of complex tables can be added to the **Table** type. You can interact with dictionaries of this type using the REST API. When adding dictionaries using the API, there is no limit on the number of entries that can be added. Required setting.
Values	Table with dictionary data. • For the **Dictionary** type, this block displays a list of **Key**—**Value** pairs. You can add and remove rows from the table. To add a row to the table, click +. To remove a row from the table, hover over the table row until the × button appears and click the button. In the **Key** field, you must specify a unique key. Maximum length of the key: 128 Unicode characters. The first character cannot be $. In the **Value** field, you must specify a value. Maximum length of the value: 255 Unicode characters. The first character cannot be $. You may add one or more **Key** – **Value** pairs. • For the **Table** type, this block displays a table containing data. You can add and remove rows and columns from the table. To add a row or column to the table, click +. To remove a row or column from the table, hover over the row or the heading of the column until the × button appears and click the button. You can edit the headings of table columns. If the dictionary contains more than 5,000 entries, they are not displayed in the LogAnalyzer web interface. To view the contents of such dictionaries, the contents must be exported in CSV format. If you edit the CSV file and import it back into LogAnalyzer, the dictionary is updated.

Importing and exporting dictionaries

You can import or export dictionary data in CSV format (in UTF-8 encoding) by using the **Import CSV** or **Export CSV** buttons.

The format of the CSV file depends on the dictionary type:

- **Dictionary** type:

  ```
  {KEY},{VALUE}\n
  ```

- **Table** type:

  ```
  {Column header 1}, {Column header N}, {Column header N+1}\n
  {Key1}, {ValueN}, {ValueN+1}\n
  {Key2}, {ValueN}, {ValueN+1}\n
  ```

  The keys must be unique for both the CSV file and the dictionary. In tables, the keys are specified in the first column. Keys must contain 1 to 128 Unicode characters.

  Values must contain 0 to 256 Unicode characters.

During an import, the contents of the dictionary are overwritten by the imported file. When imported into the dictionary, the resource name is also changed to reflect the name of the imported file.

If the key or value contains comma or quotation mark characters (, and "), they are enclosed in quotation marks (") when exported. Also, quotation mark character (") is shielded with additional quotation mark (").

If incorrect lines are detected in the imported file (for example, invalid separators), these lines will be ignored during import into the dictionary, and the import process will be interrupted during import into the table.

Interacting with dictionaries via API

You can use the REST API to read the contents of **Table**-type dictionaries. You can also modify them even if these resources are being used by active services. This lets you, for instance, configure enrichment of events with data from dynamically changing tables exported from third-party applications.

Predefined dictionaries

The dictionaries listed in the table below are included in the LogAnalyzer distribution kit.

Predefined dictionaries

Dictionary name	Type	Description
[OOTB] Ahnlab. Severity	dictionary	Contains a table of correspondence between a priority ID and its name.
[OOTB] Ahnlab. SeverityOperational	dictionary	Contains values of the SeverityOperational parameter and a corresponding description.
[OOTB] Ahnlab. VendorAction	dictionary	Contains a table of correspondence between the ID of the operation being performed and its name.
[OOTB] Cisco ISE Message Codes	dictionary	Contains Cisco ISE event codes and their corresponding names.
[OOTB] DNS. Opcodes	dictionary	Contains a table of correspondence between decimal opcodes of DNS operations and their IANA-registered descriptions.
[OOTB] IANAProtocolNumbers	dictionary	Contains the port numbers of transport protocols (TCP, UDP) and their corresponding service names, registered by IANA.
[OOTB] Juniper - JUNOS	dictionary	Contains JUNOS event IDs and their corresponding descriptions.
[OOTB] KEDR. AccountType	dictionary	Contains the ID of the user account type and its corresponding type name.
[OOTB] KEDR. FileAttributes	dictionary	Contains IDs of file attributes stored by the file system and their corresponding descriptions.
[OOTB] KEDR. FileOperationType	dictionary	Contains IDs of file operations from the KATA API and their corresponding operation names.
[OOTB] KEDR. FileType	dictionary	Contains modified file IDs from the KATA API and their corresponding file type descriptions.
[OOTB] KEDR. IntegrityLevel	dictionary	Contains the SIDs of the Microsoft Windows INTEGRITY LEVEL parameter and their corresponding descriptions.
[OOTB] KEDR. RegistryOperationType	dictionary	Contains IDs of registry operations from the KATA API and their corresponding values.
[OOTB] Linux. Sycall types	dictionary	Contains Linux call IDs and their corresponding names.
[OOTB] MariaDB Error Codes	dictionary	The dictionary contains MariaDB error codes and is used by the [OOTB] MariaDB Audit Plugin syslog normalizer to enrich events.
[OOTB] Microsoft SQL Server codes	dictionary	Contains MS SQL Server error IDs and their corresponding descriptions.
[OOTB] MS DHCP Event IDs Description	dictionary	Contains Microsoft Windows DHCP server event IDs and their corresponding descriptions.
[OOTB] S-Terra. Dictionary MSG ID to Name	dictionary	Contains IDs of S-Terra device events and their corresponding event names.
[OOTB] S-Terra. MSG_ID to Severity	dictionary	Contains IDs of S-Terra device events and their corresponding Severity values.
[OOTB] Syslog Priority To Facility and Severity	table	The table contains the **Priority** values and the corresponding **Facility and Severity** field values.
[OOTB] VipNet Coordinator Syslog Direction	dictionary	Contains direction IDs (sequences of special characters) used in ViPNet Coordinator to designate a direction, and their corresponding values.
[OOTB] Wallix EventClassId - DeviceAction	dictionary	Contains Wallix AdminBastion event IDs and their corresponding descriptions.
[OOTB] Windows.Codes (4738)	dictionary	Contains operation codes present in the MS Windows audit event with ID 4738 and their corresponding names.

[OOTB] Windows.Codes (4719)	dictionary	Contains operation codes present in the MS Windows audit event with ID 4719 and their corresponding names.
[OOTB] Windows.Codes (4663)	dictionary	Contains operation codes present in the MS Windows audit event with ID 4663 and their corresponding names.
[OOTB] Windows.Codes (4662)	dictionary	Contains operation codes present in the MS Windows audit event with ID 4662 and their corresponding names.
[OOTB] Windows. EventIDs and Event Names mapping	dictionary	Contains Windows event IDs and their corresponding event names.
[OOTB] Windows. FailureCodes (4625)	dictionary	Contains IDs from the **Failure Information\Status** and **Failure Information\Sub Status** fields of Microsoft Windows event 4625 and their corresponding descriptions.
[OOTB] Windows. ImpersonationLevels (4624)	dictionary	Contains IDs from the **Impersonation level** field of Microsoft Windows event 4624 and their corresponding descriptions.
[OOTB] Windows. KRB ResultCodes	dictionary	Contains Kerberos v5 error codes and their corresponding descriptions.
[OOTB] Windows. LogonTypes (Windows all events)	dictionary	Contains IDs of user logon types and their corresponding names.
[OOTB] Windows_Terminal Server. EventIDs and Event Names mapping	dictionary	Contains Microsoft Terminal Server event IDs and their corresponding names.
[OOTB] Windows. Validate Cred. Error Codes	dictionary	Contains IDs of user logon types and their corresponding names.
[OOTB] ViPNet Coordinator Syslog Direction	dictionary	Contains direction IDs (sequences of special characters) used in ViPNet Coordinator to designate a direction, and their corresponding values.
[OOTB] Syslog Priority To Facility and Severity	table	Contains the Priority values and the corresponding Facility and Severity field values.

Connectors

Connectors are used for establishing connections between LogAnalyzer services and for receiving events actively and passively.

You can specify connector settings on the **Basic settings** and **Advanced settings** tabs. The available settings depend on the selected type of connector.

Connectors can have the following types:

- internal for receiving data from LogAnalyzer services using the 'internal' protocol.
- tcp for passively receiving events over TCP when working with Windows and Linux agents.
- udp for passively receiving events over UDP when working with Windows and Linux agents.
- nats-jetstream for interacting with a NATS message broker when working with Windows and Linux agents.
- kafka for communicating with the Apache Kafka data bus when working with Windows and Linux agents.
- http for receiving events over HTTP when working with Windows and Linux agents.

- sql for querying databases. LogAnalyzer supports multiple **types of databases**. When creating a connector of the **sql** type, you must specify general connector settings and individual database connection settings.

  The application supports the following types of SQL databases:

  - SQLite.
  - MariaDB 10.5 or later.
  - MSSQL.
  - MySQL 5.7 or later.
  - PostgreSQL.
  - Cockroach.
  - Oracle.
  - Firebird.

- file for getting data from text files when working with Windows and Linux agents. One line of a text file is considered to be one event. `\n` is used as the newline character.
- 1c-log for getting data from 1C technology logs when working with Linux agents. `\n` is used as the newline character. The connector accepts only the first line from a multi-line event record.
- 1c-xml for getting data from 1C registration logs when working with Linux agents. When the connector handles multi-line events, it converts them into single-line events.
- diode for unidirectional data transmission in industrial ICS networks using data diodes.
- ftp for getting data over File Transfer Protocol (FTP) when working with Windows and Linux agents.
- nfs for getting data over Network File System (NFS) when working with Windows and Linux agents.
- wmi for getting data using Windows Management Instrumentation when working with Windows agents.
- wec for getting data using Windows Event Forwarding (WEF) and Windows Event Collector (WEC), or local operating system logs of a Windows host when working with Windows agents.
- etw for getting extended logs of DNS servers.
- snmp for getting data over Simple Network Management Protocol (SNMP) when working with Windows and Linux agents. To process events received over SNMP, you must use the json normalizer. Supported SNMP protocol versions:
  - snmpV1
  - snmpV2
  - snmpV3

- snmp-trap for passively receiving events using SNMP traps when working with Windows and Linux agents. The connector receives snmp-trap events and prepares them for normalization by mapping SNMP object IDs to temporary keys. Then the message is passed to the JSON normalizer, where the temporary keys are mapped to the LogAnalyzer fields and an event is generated. To process events received over SNMP, you must use the json normalizer. Supported SNMP protocol versions:
  - snmpV1
  - snmpV2
- kata/edr for getting KEDR data via the API.
- vmware for getting VMware vCenter data via the API.
- elastic for getting Elasticsearch data. Elasticsearch version 7.0.0 is supported.
- office365 for receiving Microsoft 365 (Office 365) data via the API.
- dfi for receiving Kaspersky Digital Footprint Intelligence data via the API.

Some connector types (such as tcp, sql, wmi, wec, and etw) support TLS encryption. LogAnalyzer supports TLS 1.2 and 1.3. When TLS mode is enabled for these connectors, the connection is established according to the following algorithm:

- If LogAnalyzer is being used as a client:
  1. LogAnalyzer sends a connection request to the server with a ClientHello message specifying the maximum supported TLS version (1.3), as well as a list of supported ciphersuites.
  2. The server responds with the preferred TLS version and a ciphersuite.
  3. Depending on the TLS version in the server response:
     - If the server responds to the request with TLS 1.3 or 1.2, LogAnalyzer establishes a connection with the server.
     - If the server responds to the request with TLS 1.1, LogAnalyzer terminates the connection with the server.
- If LogAnalyzer is being used as a server:
  1. The client sends a connection request to LogAnalyzer with the maximum supported TLS version, as well as a list of supported ciphersuites.
  2. Depending on the TLS version in the client request:
     - If the ClientHello message of the client request specifies TLS 1.1, LogAnalyzer terminates the connection.
     - If the client request specifies TLS 1.2 or 1.3, LogAnalyzer responds to the request with the preferred TLS version and a ciphersuite.

Viewing connector settings

To view connector settings:

1. In the LogAnalyzer web interface, select **Resources** → **Connectors**.
2. In the folder structure, select the folder containing the relevant connector.
3. Select the connector whose settings you want to view.

The settings of connectors are displayed on two tabs: **Basic settings** and **Advanced settings**. For a detailed description of each connector settings, please refer to the *Connector settings* section.

Adding a connector

You can enable the display of non-printing characters for all entry fields except the **Description** field.

To add a connector:

1. In the LogAnalyzer web interface, select **Resources** → **Connectors**.
2. In the folder structure, select the folder in which you want the connector to be located.

   Root folders correspond to tenants. To make a connector available to a specific tenant, the resource must be created in the folder of that tenant.

   If the required folder is absent from the folder tree, you need to create it.

   By default, added connectors are created in the **Shared** folder.
3. Click the **Add connector** button.
4. Define the settings for the selected connector type.

   The settings that you must specify for each type of connector are provided in the *Connector settings* section.
5. Click the **Save** button.

Connector settings

This section describes the settings of all connector types supported by LogAnalyzer.

Connector, internal type

Connectors of the **internal** type are used for receiving data from LogAnalyzer services using the 'internal' protocol. For example, you must use such a connector to receive the following data:

- Internal data, such as event routes.
- File attributes. If while creating the collector at the Transport step of the Installation Wizard, you specified a connector of the file, 1c-xml, or 1c-log type, at the Event parsing step, in the **Mapping** table, you can pass the name of the file being processed by the collector or the path to the file in the LogAnalyzer event field. To do this, in the **Source** column, specify one of the following values:
  - `$kuma_fileSourceName` to pass the name of the file being processed by the collector in the LogAnalyzer event field.
  - `$kuma_fileSourcePath` to pass the path to the file being processed by the collector in the LogAnalyzer event field.

  When you use a **file**, **1c-xml**, or **1c-log** connector, the new variables in the normalizer will only work with destinations of the internal type.
- Events to the event router. The event router can only receive events over the 'internal' protocol, therefore you can only use **internal** destinations when sending events to the event router.

Settings for a connector of the **internal** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **internal**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	The URL and port that the connector is listening on. You can enter a value in one of the following formats: • `<host name>:<port number>` • `<IPv4 address>:<port number>` • `:<port number>` You can specify IPv6 addresses in the following format: `[<IPv6 address>%<interface>:<port number>`, for example, `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`. You can add multiple values or delete values. To add a value, click the **+ Add** button. To delete a value, click the delete ✕ icon next to it. Required setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.

Connector, tcp type

Connectors of the **tcp** type are used for passively receiving events over TCP when working with Windows and Linux agents. Settings for a connector of the **tcp** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **tcp**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	URL that you want to connect to. You can enter a URL in one of the following formats: • `< host name >:< port number >` • `< IPv4 address >:< port number >` • `< IPv6 address >:< port number >` • `:< port number >` Required setting.

Setting	Description
Auditd	This toggle switch enables the auditd mechanism to group auditd event lines received from the connector into an auditd event. If you enable this toggle switch, you cannot select a value in the **Delimiter** drop-down list because **\n** is automatically selected for the mechanism. If you enable this toggle switch in the connector settings of the agent, you need to select **\n** in the **Delimiter** drop-down list in the cor settings of the collector to which the agent sends events. The maximum size of a grouped auditd event is approximately 4,174,304 characters. LogAnalyzer classifies Auditd events according to an algorithm. For example, suppose the following records were received for proc `type=LOGIN msg=audit(1712820601.957:21458): pid=4987 uid=0 subj=0:63:0:0 old-auid=4294967295 auid=0 tty= old-ses=4294967295 ses=2348 res=1` `type=SYSCALL msg=audit(1712820601.957:21458): arch=c000003e syscall=1 success=yes exit=1 a0=7 a1=7ffc9a0 a3=0 items=0 ppid=429 pid=4987 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) comm="cron" exe="/usr/sbin/cron" subj=0:63:0:0 key=(null)` `type=PROCTITLE msg=audit(1712820601.957:21458): proctitle=2F7573722F7362696E2F43524F4E002D66` The algorithm gives one single-line event of the LOGIN type (because the LOGIN type has code 1006 and it is less than 1300, which is AUDIT_FIRST_EVENT), and one multi-line event with SYSCALL and PROCTITLE. When grouping a multi-line auditd event containing records with different prefixes, the resulting event uses the prefix of the record th first. This is necessary for the normalizer to parse the event correctly. For example, suppose the following records were received for p `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit_log: type=PROCTITLE msg=audit(1742441361.524:787467): proctitle=686170726F7879002D57002D6462002D66002F7573722F6C6F63616C2F6574632F686170726F78792F686170726F78` `<182>2025-03-20T06:29:22.643145+03:00 test-server-01 tag_audit_log: type=SOCKADDR msg=audit(1742441361.524:787467): saddr=020026FC0A5E41E80000000000000000` `<182>2025-03-20T06:29:22.643143+03:00 test-server-01 tag_audit_log: type=SYSCALL msg=audit(1742441361.52 arch=c000003e syscall=42 success=no exit=-115 a0=1b a1=55d3f02e8c80 a2=10 a3=55d3eec440d0 items=0 ppid=1 pid=1356 auid=4294967295 uid=99 gid=99 euid=99 suid=99 fsuid=99 egid=99 sgid=99 fsgid=99 tty=(none) ses= comm="haproxy" exe="/usr/local/sbin/haproxy" subj=docker-default key="network_events"<` The `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit_log: type=PROCTITLE msg=audit(...)` arrived first, therefore the resulting event uses the `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit` prefix.
Delimiter	The character that marks the boundary between events: • **\n** • **\t** • **\0** If you do not select a value in this drop-down list, an empty value is selected by default. In this case, the connector does not split one several and passes it to the normalizer unchanged.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Buffer size	Buffer size in bytes for accumulating events in the RAM of the server before sending them for further processing or storage. The value must be a positive integer. Default buffer size: 67,108,864 (64 MB), displayed as 0 in the interface.
Character encoding	Character encoding. The default is **UTF-8**.

Setting	Description
Event Buffer TTL	Buffer lifetime for auditd event lines, in milliseconds. Auditd event lines enter the LogAnalyzer collector and accumulate in the buffer. This allows multiple auditd event lines to be grouped into a single auditd event. The buffer lifetime countdown begins when the first auditd event line is received or when the previous buffer lifetime expires. Possible values: from 50 to 30,000. The default value is `2000`. This field is available if you have enabled the **Auditd** toggle switch on the **Basic settings** tab. The auditd event lines accumulated in the buffer are kept in the RAM of the server. We recommend caution when increasing the buffer size because memory usage by the LogAnalyzer collector may become excessive. You can verify how much server RAM the LogAnalyzer collector is using in LogAnalyzer metrics. If you want a buffer lifetime to exceed 30,000 milliseconds, we recommend using a different auditd event transport. For example, you can use an agent or pre-accumulate auditd events in a file, and then process this file with the LogAnalyzer collector.
Transport header	Regular expression for auditd events, which is used to identify auditd event lines. You can use the default value or edit it. The regular expression must contain the `record_type_name`, `record_type_value`, and `event_sequence_number` groups. If a multi-line auditd event contains a prefix, the prefix is retained for the first line of the auditd event and discarded for the following lines. You can revert to the default regular expression for auditd events by clicking **Reset to default value**.

Setting	Description
TLS mode	TLS encryption mode. When using TLS encryption, you cannot specify an IP address in the URL field on the **Basic settings** tab. Available values: • **Disabled** means TLS encryption is not used. This value is selected by default. • **Enabled** means TLS encryption is used, but certificates are not verified. • **With verification** means TLS encryption is used with verification of the certificate signed with the LogAnalyzer root certificate. The root certificate and key of LogAnalyzer are created automatically during application installation and are stored on the LogAnalyzer Core server in the /opt/kaspersky/kuma/core/certificates/ directory. Obsolete encryption mode, deprecated. Will be removed in future versions of LogAnalyzer. • **Custom PFX** means TLS encryption with a PFX secret. You must generate a PFX certificate with a private key in PKCS#12 container format, then upload the PFX certificate to the LogAnalyzer web interface as a PFX secret. If you select this value, in the PFX secret drop-down list, specify a PFX secret with a certificate signed by a certification authority. You can select an existing PFX secret or create a new PFX secret. To create a new PFX secret, select **Create new**. If you want to edit the settings of an existing secret, click the pencil icon next to it. How to create a PFX secret? *To create a PFX secret:* 1. In the **Name** field, enter the name of the PFX secret. 2. Click **Upload PFX** and select the PKCS#12 container file to which you exported the PFX file with the private key. 3. In the **Password** field, enter the password of the PFX file that was set in the PFX Export Wizard. 4. Click the **Create** button. The PFX secret is created and displayed in the **PFX secret** drop-down list. • **Custom PFX with client authentication** means TLS encryption with a PFX secret. In this mode, the client is authenticated using a certificate. You must generate a PFX certificate with a private key in PKCS#12 container format, then upload the PFX certificate to the LogAnalyzer web interface as a PFX secret. If you select this value, in the PFX secret drop-down list, specify a PFX secret with a certificate signed by a certification authority. You can select an existing PFX secret or create a new PFX secret. To create a new PFX secret, select **Create new**. If you want to edit the settings of an existing secret, click the pencil icon next to it. How to create a PFX secret? *To create a PFX secret:* 1. In the **Name** field, enter the name of the PFX secret. 2. Click **Upload PFX** and select the PKCS#12 container file to which you exported the PFX file with the private key. 3. In the **Password** field, enter the password of the PFX file that was set in the PFX Export Wizard. 4. Click the **Create** button. The PFX secret is created and displayed in the **PFX secret** drop-down list.
Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**.

Connector, udp type

Connectors of the **udp** type are used for passively receiving events over UDP when working with Windows and Linux agents. Settings for a connector of the **udp** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **udp**. Required setting.
URL	URL that you want to connect to. You can enter a URL in one of the following formats: • `< host name >:< port number >` • `< IPv4 address >:< port number >` • `< IPv6 address >:< port number >` • `:< port number >` Required setting.
Auditd	This toggle switch enables the auditd mechanism to group auditd event lines received from the connector into an auditd event. If you enable this toggle switch, you cannot select a value in the **Delimiter** drop-down list because **\n** is automatically selected for the mechanism. If you enable this toggle switch in the connector settings of the agent, you need to select **\n** in the **Delimiter** drop-down list in the cor settings of the collector to which the agent sends events. The maximum size of a grouped auditd event is approximately 4,174,304 characters. LogAnalyzer classifies Auditd events according to an algorithm. For example, suppose the following records were received for proc `type=LOGIN msg=audit(1712820601.957:21458): pid=4987 uid=0 subj=0:63:0:0 old-auid=4294967295 auid=0 tty= old-ses=4294967295 ses=2348 res=1` `type=SYSCALL msg=audit(1712820601.957:21458): arch=c000003e syscall=1 success=yes exit=1 a0=7 a1=7ffc9a0 a3=0 items=0 ppid=429 pid=4987 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) comm="cron" exe="/usr/sbin/cron" subj=0:63:0:0 key=(null)` `type=PROCTITLE msg=audit(1712820601.957:21458): proctitle=2F7573722F7362696E2F43524F4E002D66` The algorithm gives one single-line event of the LOGIN type (because the LOGIN type has code 1006 and it is less than 1300, which is AUDIT_FIRST_EVENT), and one multi-line event with SYSCALL and PROCTITLE. When grouping a multi-line auditd event containing records with different prefixes, the resulting event uses the prefix of the record th first. This is necessary for the normalizer to parse the event correctly. For example, suppose the following records were received for p `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit_log: type=PROCTITLE msg=audit(1742441361.524:787467): proctitle=686170726F7879002D57002D6462002D66002F7573722F6C6F63616C2F6574632F686170726F78792F686170726F78` `<182>2025-03-20T06:29:22.643145+03:00 test-server-01 tag_audit_log: type=SOCKADDR msg=audit(1742441361.524:787467): saddr=020026FC0A5E41E80000000000000000` `<182>2025-03-20T06:29:22.643143+03:00 test-server-01 tag_audit_log: type=SYSCALL msg=audit(1742441361.52 arch=c000003e syscall=42 success=no exit=-115 a0=1b a1=55d3f02e8c80 a2=10 a3=55d3eec440d0 items=0 ppid=1 pid=1356 auid=4294967295 uid=99 gid=99 euid=99 suid=99 fsuid=99 egid=99 sgid=99 fsgid=99 tty=(none) ses= comm="haproxy" exe="/usr/local/sbin/haproxy" subj=docker-default key="network_events"<` The `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit_log: type=PROCTITLE msg=audit(...)` arrived first, therefore the resulting event uses the `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit` prefix.

Setting	Description
Delimiter	The character that marks the boundary between events: • \n • \t • \0 If you do not select a value in this drop-down list, an empty value is selected by default. This means that the connector does not parse into many, but instead passes it to the normalizer as it is.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Buffer size	Buffer size in bytes for accumulating events in the RAM of the server before sending them for further processing or storage. The value must be a positive integer. Default buffer size: 67,108,864 (64 MB), displayed as 0 in the interface.
Number of handlers	Number of handlers that the service can run simultaneously to process response rules in parallel. The value must be a positive integer up to 999.
Character encoding	Character encoding. The default is **UTF-8**.
Event Buffer TTL	Buffer lifetime for auditd event lines, in milliseconds. Auditd event lines enter the LogAnalyzer collector and accumulate in the buffer. This allows multiple auditd event lines to be grouped into a single auditd event. The buffer lifetime countdown begins when the first auditd event line is received or when the previous buffer lifetime expires. Possible values: from 50 to 30,000. The default value is `2000`. This field is available if you have enabled the **Auditd** toggle switch on the **Basic settings** tab. The auditd event lines accumulated in the buffer are kept in the RAM of the server. We recommend caution when increasing the buffer size because memory usage by the LogAnalyzer collector may become excessive. You can verify how much server RAM the LogAnalyzer collector is using in LogAnalyzer metrics. If you want a buffer lifetime to exceed 30,000 milliseconds, we recommend using a different auditd event transport. For example, you can use an agent or pre-accumulate auditd events in a file, and then process this file with the LogAnalyzer collector.
Transport header	Regular expression for auditd events, which is used to identify auditd event lines. You can use the default value or edit it. The regular expression must contain the `record_type_name`, `record_type_value`, and `event_sequence_number` groups. If a multi-line auditd event contains a prefix, the prefix is retained for the first line of the auditd event and discarded for the following lines. You can revert to the default regular expression for auditd events by clicking **Reset to default value**.
Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**.

Connector, nats-jetstream type

Connectors of the **nats-jetstream** type are used for interacting with a NATS message broker when working with Windows and Linux agents. Settings for a connector of the **nats-jetstream** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **nats-jetstream**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	URL that you want to connect to. The following URL formats are supported: • `<host name>:<port number>` • `<IPv4 address>:<port number>` • `<IPv6 address>:<port number>` You can specify IPv6 addresses in the following format: `[<IPv6 address>%<interface>:<port number>`, for example, `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`. You can add multiple URLs or remove an URL. LogAnalyzer does not allow saving a resource or service if the URL field contains a tab or space character. To add an URL, click the **+ Add** button. To remove an URL, click the delete ✕ icon next to it. Required setting.
Authorization	Type of authorization when connecting to the URL specified in the **URL** field: • **Disabled**. This value is selected by default. • **Plain**. Authorization of this type is performed using the SASL/PLAIN mechanism, which transmits the user name and password in plain text. Typically, TLS is used to provide transmission security (encryption) in SASL/PLAIN. If this option is selected, in the **Secret** drop-down list, specify the secret containing user account credentials for authorization when connecting to the destination. You can select an existing secret or create a new secret. To create a new secret, select **Create new**. If you want to edit the settings of an existing secret, click the pencil icon next to it. How to create a secret? *To create a secret:* 1. In the **Name** field, enter the name of the secret. 2. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector. 3. If necessary, enter a description of the secret in the **Description** field. 4. Click the **Create** button. The secret is added and displayed in the **Secret** drop-down list.
Subject	The topic of NATS messages. Characters are entered in Unicode encoding. Required setting.
GroupID.	The value of the `GroupID` parameter for NATS messages. Maximum length of the value: 255 Unicode characters. The default value is `default`.

Setting	Description
Delimiter	The character that marks the boundary between events: • **\n** • **\t** • **\0** If you do not select a value in this drop-down list, an empty value is selected by default. In this case, the connector does not split one event into several and passes it to the normalizer unchanged.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Buffer size	Buffer size in bytes for accumulating events in the RAM of the server before sending them for further processing or storage. The value must be a positive integer. Default buffer size: 67,108,864 (64 MB), displayed as 0 in the interface.
Number of handlers	Number of handlers that the service can run simultaneously to process response rules in parallel. The value must be a positive integer up to 999.
Character encoding	Character encoding. The default is **UTF-8**.

Setting	Description

TLS mode	TLS encryption mode. When using TLS encryption, you cannot specify an IP address in the **URL** field on the **Basic settings** tab. Available values:
	• **Disabled** means TLS encryption is not used. This value is selected by default.
	• **Enabled** means TLS encryption is used, but certificates are not verified.
	• **With verification** means TLS encryption is used with verification of the certificate signed with the LogAnalyzer root certificate. The root certificate and key of LogAnalyzer are created automatically during application installation and are stored on the LogAnalyzer Core server in the /opt/kaspersky/kuma/core/certificates/ directory.
	Obsolete encryption mode, deprecated. Will be removed in future versions of LogAnalyzer.
	• **Custom CA** means TLS encryption is used with verification that the certificate was signed by a Certificate Authority. If you select this value, in the **Custom CA** drop-down list, specify a secret with a certificate signed by a certification authority. You can select an existing secret or create a new secret. To create a new secret, select **Create new**.
	If you want to edit the settings of an existing secret, click the pencil icon next to it.
	How to create a certificate signed by a Certificate Authority?
	You can create a CA-signed certificate on the LogAnalyzer Core server (the following command examples use OpenSSL).
	To create a certificate signed by a Certificate Authority:
	1. Generate a key to be used by the Certificate Authority, for example:
	`openssl genrsa -out ca.key 2048`
	2. Create a certificate for the generated key, for example:
	`openssl req -new -x509 -days 365 -key ca.key -subj "/CN=<common host name of Certificate Authority>" -out ca.crt`
	3. Create a private key and a request to have it signed by the Certificate Authority, for example:
	`openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN=<common host name of LogAnalyzer server>" -out server.csr`
	4. Create the certificate signed by the Certificate Authority. You need to include the domain names or IP addresses of the server for which you are creating the certificate in the `subjectAltName` variable, for example:
	`openssl x509 -req -extfile <(printf "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1") -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt`
	5. Upload the created `server.crt` certificate in the LogAnalyzer web interface to a secret of the **certificate** type, then in the **Custom CA** drop-down list, select the secret of the **certificate** type.
	To use LogAnalyzer certificates on third-party devices, you must change the certificate file extension from CERT to CRT. Otherwise, you can get the `x509: certificate signed by unknown authority` error.
	• **Custom PFX** means TLS encryption with a PFX secret. You must generate a PFX certificate with a private key in PKCS#12 container format, then upload the PFX certificate to the LogAnalyzer web interface as a PFX secret. If you select this value, in the **PFX secret** drop-down list, specify a PFX secret with a certificate signed by a certification authority. You can select an existing PFX secret or create a new PFX secret. To create a new PFX secret, select **Create new**.
	If you want to edit the settings of an existing secret, click the pencil icon next to it.
	How to create a PFX secret?

Setting	Description
	To create a PFX secret: 1. In the **Name** field, enter the name of the PFX secret. 2. Click **Upload PFX** and select the PKCS#12 container file to which you exported the PFX file with the private key. 3. In the **Password** field, enter the password of the PFX file that was set in the PFX Export Wizard. 4. Click the **Create** button. The PFX secret is created and displayed in the **PFX secret** drop-down list. The **Custom PFX** mode is not displayed when **Authorization** is set to **Plain**.
Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**.

Connector, kafka type

Connectors of the **kafka** type are used for communicating with the Apache Kafka data bus when working with Windows and Linux agents. Settings for a connector of the **kafka** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **kafka**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	URL that you want to connect to. The following URL formats are supported: • `< host name >:< port number >` • `< IPv4 address >:< port number >` • `< IPv6 address >:< port number >` You can specify IPv6 addresses in the following format: `[< IPv6 address >%< interface >:< port number >`, for example, `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`. You can add multiple URLs or remove an URL. LogAnalyzer does not allow saving a resource or service if the URL field contains a tab or space character. To add an URL, click the **+ Add** button. To remove an URL, click the delete ✕ icon next to it. Required setting.

Setting	Description
Authorization	Type of authorization when connecting to the URL specified in the **URL** field: • **Disabled**. This value is selected by default. • **Plain**. Authorization of this type is performed using the SASL/PLAIN mechanism, which transmits the user name and password in plain text. Typically, TLS is used to provide transmission security (encryption) in SASL/PLAIN. If this option is selected, in the **Secret** drop-down list, specify the secret containing user account credentials for authorization when connecting to the destination. You can select an existing secret or create a new secret. To create a new secret, select **Create new**. If you want to edit the settings of an existing secret, click the pencil icon next to it. How to create a secret? *To create a secret:* 1. In the **Name** field, enter the name of the secret. 2. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector. 3. If necessary, enter a description of the secret in the **Description** field. 4. Click the **Create** button. The secret is added and displayed in the **Secret** drop-down list. • **PFX** means TLS encryption with a PFX secret. You must generate a certificate with a private key in PKCS#12 container format in an external Certificate Authority, export the PFX file from the key store, and upload the PFX file to the LogAnalyzer web interface as a PFX secret. If you select this value, in the **PFX secret** drop-down list, specify a PFX secret with a certificate signed by a certification authority. You can select an existing PFX secret or create a new PFX secret. To create a new PFX secret, select **Create new**. If you want to edit the settings of an existing secret, click the pencil icon next to it. How to create a PFX secret? *To create a PFX secret:* 1. In the **Name** field, enter the name of the PFX secret. 2. Click **Upload PFX** and select the PKCS#12 container file to which you exported the PFX file with the private key. 3. In the **Password** field, enter the password of the PFX file that was set in the PFX Export Wizard. 4. Click the **Create** button. The PFX secret is created and displayed in the **PFX secret** drop-down list. If you are using a chain of several certificates to authenticate the server, we recommend building the PFX file using OpenSSL to ensure the correct order of certificates in the chain.
Topic	Subject of Kafka messages. The maximum length of the subject is 255 characters. Space and tab characters are not allowed. You can use the following characters: a–z, A–Z, 0–9, ".", "_", "-". Required setting.
GroupID.	The value of the `GroupID` parameter for Kafka messages. Maximum length of the value: 255 characters. You can use the following characters: a–z, A–Z, 0–9, ".", "_", and "-".

Setting	Description
Delimiter	The character that marks the boundary between events: • **\n** • **\t** • **\0** If you do not select a value in this drop-down list, an empty value is selected by default. In this case, the connector does not split one event into several and passes it to the normalizer unchanged.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Number of handlers	Number of handlers that the service can run simultaneously to process response rules in parallel. The value must be a positive integer up to 999. By default, the number of handlers is 256.
Character encoding	Character encoding. The default is **UTF-8**.

Setting	Description

TLS mode	TLS encryption mode. When using TLS encryption, you cannot specify an IP address in the **URL** field on the **Basic settings** tab. Available values:
	• **Disabled** means TLS encryption is not used. This value is selected by default.
	• **Enabled** means TLS encryption is used, but certificates are not verified.
	• **With verification** means TLS encryption is used with verification of the certificate signed with the LogAnalyzer root certificate. The root certificate and key of LogAnalyzer are created automatically during application installation and are stored on the LogAnalyzer Core server in the /opt/kaspersky/kuma/core/certificates/ directory.
	Obsolete encryption mode, deprecated. Will be removed in future versions of LogAnalyzer.
	• **Custom CA** means TLS encryption is used with verification that the certificate was signed by a Certificate Authority. If you select this value, in the **Custom CA** drop-down list, specify a secret with a certificate signed by a certification authority. You can select an existing secret or create a new secret. To create a new secret, select **Create new**.
	If you want to edit the settings of an existing secret, click the pencil icon next to it.
	How to create a certificate signed by a Certificate Authority?
	You can create a CA-signed certificate on the LogAnalyzer Core server (the following command examples use OpenSSL).
	To create a certificate signed by a Certificate Authority:
	1. Generate a key to be used by the Certificate Authority, for example:
	`openssl genrsa -out ca.key 2048`
	2. Create a certificate for the generated key, for example:
	`openssl req -new -x509 -days 365 -key ca.key -subj "/CN=< common host name of Certificate Authority >" -out ca.crt`
	3. Create a private key and a request to have it signed by the Certificate Authority, for example:
	`openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN= < common host name of LogAnalyzer server >" -out server.csr`
	4. Create the certificate signed by the Certificate Authority. You need to include the domain names or IP addresses of the server for which you are creating the certificate in the `subjectAltName` variable, for example:
	`openssl x509 -req -extfile <(printf "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1") -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt`
	5. Upload the created `server.crt` certificate in the LogAnalyzer web interface to a secret of the **certificate** type, then in the **Custom CA** drop-down list, select the secret of the **certificate** type.
	To use LogAnalyzer certificates on third-party devices, you must change the certificate file extension from CERT to CRT. Otherwise, you can get the `x509: certificate signed by unknown authority` error.

Size of message to fetch	Size of one message in the request, in bytes. The default value of 16 MB is applied if no value is specified or 0 is specified. It cannot exceed 2147483647 bytes.
Maximum fetch wait time	Timeout for one message in seconds. The default value of 5 seconds is applied if no value is specified or 0 is specified. Maximum possible value: 2147483647.
Connection timeout	Kafka broker connection timeout in seconds. Maximum possible value: 2147483647. The default value is 30 seconds.

Setting	Description
Read timeout	Read operation timeout in seconds. Maximum possible value: 2147483647. The default value is 30 seconds.
Write timeout	Write operation timeout in seconds. Maximum possible value: 2147483647. The default value is 30 seconds.
Group heartbeat interval	Group status update interval in seconds Cannot exceed session time. The recommended value is 1/3 of the session time. Maximum possible value: 2147483647. The default value is 3 seconds.
Session timeout	Session time in seconds. Maximum possible value: 2147483647. The default value is 10 seconds.
Max processing time	Maximum time to process one message by a single thread, in milliseconds. Maximum possible value: 2147483647. The default value is 100 milliseconds.
Auto commit enabled	Enabled by default.
Auto commit interval	Autocommit interval in seconds The default value is 1 second. Maximum possible value: 18446744073709551615. Any positive number can be specified.
PFX secret	Secret of the pfx type. This field is available if **Authorization** is set to **PFX**. You can select an existing secret or create a new secret. To create a new secret, select **Create new**. Required field.
Secret	Secret of the 'credentials' type. This field is available if **Authorization** is set to **Plain**. You can select an existing secret or create a new secret. To create a new secret, select **Create new**. Required field.

Connector, http type

Connectors of the **http** type are used for receiving events over HTTP when working with Windows and Linux agents. Settings for a connector of the **http** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **http**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	URL that you want to connect to. You can enter a URL in one of the following formats: • `<host name>:<port number>` • `<IPv4 address>:<port number>` • `<IPv6 address>:<port number>` • `:<port number>` For data from the event source to be sent to the connector, the specified URL must end with `/input`. Required setting.

Setting	Description
Delimiter	The character that marks the boundary between events: • **\n** • **\t** • **\0** If you do not select a value in this drop-down list, an empty value is selected by default. In this case, the connector does not split one event into several and passes it to the normalizer unchanged.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.
TLS mode	TLS encryption mode. When using TLS encryption, you cannot specify an IP address in the URL field on the **Basic settings** tab. Available values: • **Disabled** means TLS encryption is not used. This value is selected by default. • **Enabled** means TLS encryption is used, but certificates are not verified. • **With verification** means TLS encryption is used with verification of the certificate signed with the LogAnalyzer root certificate. The root certificate and key of LogAnalyzer are created automatically during application installation and are stored on the LogAnalyzer Core server in the /opt/kaspersky/kuma/core/certificates/ directory. Obsolete encryption mode, deprecated. Will be removed in future versions of LogAnalyzer. • **Custom PFX** means TLS encryption with a PFX secret. You must generate a PFX certificate with a private key in PKCS#12 container format, then upload the PFX certificate to the LogAnalyzer web interface as a PFX secret. If you select this value, in the PFX secret drop-down list, specify a PFX secret with a certificate signed by a certification authority. You can select an existing PFX secret or create a new PFX secret. To create a new PFX secret, select Create new. If you want to edit the settings of an existing secret, click the pencil icon next to it. How to create a PFX secret? *To create a PFX secret:* 1. In the **Name** field, enter the name of the PFX secret. 2. Click **Upload PFX** and select the PKCS#12 container file to which you exported the PFX file with the private key. 3. In the **Password** field, enter the password of the PFX file that was set in the PFX Export Wizard. 4. Click the **Create** button. The PFX secret is created and displayed in the **PFX secret** drop-down list. • **Custom PFX with client authentication** means TLS encryption with a PFX secret. In this mode, the client is authenticated using a certificate. You must generate a PFX certificate with a private key in PKCS#12 container format, then upload the PFX certificate to the LogAnalyzer web interface as a PFX secret. If you select this value, in the PFX secret drop-down list, specify a PFX secret with a certificate signed by a certification authority. You can select an existing PFX secret or create a new PFX secret. To create a new PFX secret, select Create new. If you want to edit the settings of an existing secret, click the pencil icon next to it. How to create a PFX secret? *To create a PFX secret:* 1. In the **Name** field, enter the name of the PFX secret. 2. Click **Upload PFX** and select the PKCS#12 container file to which you exported the PFX file with the private key. 3. In the **Password** field, enter the password of the PFX file that was set in the PFX Export Wizard. 4. Click the **Create** button. The PFX secret is created and displayed in the **PFX secret** drop-down list.

Connector, sql type

Connectors of the **sql** type are used for querying databases. LogAnalyzer supports multiple **types of databases**. When creating a connector of the **sql** type, you must specify general connector settings and individual database connection settings. Settings for a connector of the **sql** type are described in the following tables.

The application supports the following types of SQL databases:

- SQLite.
- MariaDB 10.5 or later.
- MSSQL.
- MySQL 5.7 or later.
- PostgreSQL.
- Cockroach.
- Oracle.
- Firebird.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **sql**. Required setting.
Tags	Tags for resource search. Optional setting.
Default query	SQL query that is executed when connecting to the database. Required setting.
Reconnect to DB each time the request is sent	This toggle enables reconnection of the connector to the database every time a query is sent. This toggle switch is turned off by default.
Poll interval, sec	Interval for executing SQL queries in seconds. The default value is 10 seconds.

Setting	Description
Connection	Database connection settings: • **Database kind** is the type of the database to connect to. When you select a database type, the prefix corresponding to the communication protocol is displayed in the **URL** field. For example, for a **ClickHouse** database, the **URL** field contains the `clickhouse://` prefix. • The **Secret separately** check box allows viewing the connection information. • **URL** is the connection URL. This lets you view connection information without having to re-create a large number of connections if the password of the user account that you used for the connections changes. When creating connections, if connection information is specified in the URL, strings with credentials containing special characters may not be handled correctly. If an error occurs when creating a connection, but you are sure that the specified settings are correct, enter the special characters in percent encoding. Codes of special characters Setting / Description ! # $ % & ' ( ) * + %21 %23 %24 %25 %26 %27 %28 %29 %2A %2B , / : ; = ? @ [ ] \ %2C %2F %3A %3B %3D %3F %40 %5B %5D %5C The following special characters are not supported in passwords used to access SQL databases: space, [,], :, /, #, %, \. If you select the **Secret separately** check box, the credentials are specified in the secret and are encoded automatically. In this case, you do not need to encode special characters. In the **Use URL from the secret** drop-down list, you can select an existing URL or create a new URL. To create a new URL, select **Create new**. If you want to edit the settings of an existing URL, click the pencil icon next to it. • **Secret** is an urls secret that stores a list of URLs for connecting to the database. This field is displayed if the **Secret separately** check box is selected. • **Identity column** is the name of the column that contains the ID for each row of the table. Required setting. • **Identity seed** is the value in the identity column for determining the row from which you want to start reading data from the SQL table. • **Query** is the additional SQL query that is executed instead of the default SQL query. • **Poll interval, sec** is the SQL query execution interval in seconds. The specified interval is used instead of the default interval for the connector. The default value is 10 seconds. You can add multiple connections or delete a connection. To add a connection, click the **+ Add connection** button. To remove a connection, click the delete icon next to it.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**. LogAnalyzer converts SQL responses to UTF-8 encoding. You can configure the SQL server to send responses in UTF-8 encoding or change the encoding of incoming messages on the LogAnalyzer side.

Within a single connector, you can **create a connection** for multiple supported databases. If a collector with a connector of the **sql** type cannot be started, check if the /opt/kaspersky/kuma/collector/<`collector ID`>/sql/state-<`file ID`> state file is empty. If the state file is empty, delete it and restart the collector.

To create a connection for multiple SQL databases:

1. Click the **Add connection** button.
2. Specify the **URL**, **Identity column**, **Identity seed**, **Query**, and **Poll interval, sec** values.
3. Repeat steps 1–2 for each required connection.

Supported SQL types and their specific usage features

The following SQL types are supported:

- MSSQL.

  For example:

  - `sqlserver://{user}:{password}@{server:port}/{instance_name}?database={database}`

  We recommend using this URL variant.

  - `sqlserver://{user}:{password}@{server}?database={database}`

  The characters `@p1` are used as a placeholder in the SQL query.

  > If you want to connect using domain account credentials, specify the account name in `< domain >%5C< user >` format. For example: `sqlserver://domain%5Cuser:password@ksc.example.com:1433/SQLEXPRESS?database=KAV`.

- MySQL/MariaDB

  For example:

  `mysql://{user}:{password}@tcp({server}:{port})/{database}`

  The characters `?` are used as placeholders in the SQL query.

- PostgreSQL.

  For example: `postgres://{user}:{password}@{server}/{database}?sslmode=disable`

  The characters `$1` are used as a placeholder in the SQL query.

- CockroachDB

  For example:

  `postgres://{user}:{password}@{server}:{port}/{database}?sslmode=disable`

  The characters `$1` are used as a placeholder in the SQL query.

- SQLite3

  For example:

  `sqlite3://file:{file_path}`

  A question mark (`?`) is used as a placeholder in the SQL query.

  When querying SQLite3, if the initial value of the ID is in datetime format, you must add a date conversion with the `sqlite datetime` function to the SQL query. For example:

  ```
  select * from connections where datetime(login_time) > datetime(?, 'utc') order by login_time
  ```

  In this example, `connections` is the SQLite table, and the value of the variable `?` is taken from the **Identity seed** field, and it must be specified in the `{< date >}T{< time >}Z` format, for example, `2021-01-01T00:10:00Z`).

- Oracle DB

  Example URL of a secret with the 'oracle' driver:

  `oracle://{user}:{password}@{server}:{port}/{service_name}`

  `oracle://{user}:{password}@{server}:{port}/?SID={SID_VALUE}`

  If the query execution time exceeds 30 seconds, the oracle driver aborts the SQL request, and the following error appears in the collector log: **user requested cancel of current operation**. To increase the execution time of an SQL query, specify the value of the **timeout** parameter in seconds in the connection string, for example:

  `oracle://{user}:{password}@{server}:{port}/{service_name}?timeout=300`

  The `:val` variable is used as a placeholder in the SQL query.

  When querying Oracle DB, if the identity seed is in the datetime format, you must consider the type of the field in the database and, if necessary, add conversions of the time string in the SQL query to make sure the SQL connector works correctly. For example, if the `Connections` table in the database has a `login_time` field, the following conversions are possible:

  - If the **login_time** field has the TIMESTAMP type, then depending on the configuration of the database, the **login_time** field may contain a value in the `YYYY-MM-DD HH24:MI:SS` format, for example, `2021-01-01 00:00:00`. In this case, you need to specify `2021-01-01T00:00:00Z` in the **Identity seed** field, and in the SQL query, perform the conversion using the `to_timestamp` function, for example:

    ```
    select * from connections where login_time > to_timestamp(:val, 'YYYY-MM-DD"T"HH24:MI:SS"Z"')
    ```

  - If the **login_time** field has the TIMESTAMP WITH TIME ZONE type, then depending on the configuration of the database, the **login_time** field may contain a value in the `YYYY-MM-DD"T"HH24:MI:SSTZH:TZM` format (for example, `2021-01-01T00:00:00+03:00`). In this case, you need to specify `2021-01-01T00:00:00+03:00` in the Identity seed field, and in the SQL query, perform the conversion using the `to_timestamp_tz` function, for example:

    ```
    select * from connections_tz where login_time > to_timestamp_tz(:val, 'YYYY-MM-DD"T"HH24:MI:SSTZH:TZM')
    ```

    For details about the `to_timestamp` and `to_timestamp_tz` functions, please refer to the official Oracle documentation.

To interact with Oracle DB, you must install the libaio1 Astra Linux package.

- Firebird® SQL

  For example:

  `firebirdsql://{user}:{password}@{server}:{port}/{database}`

  A question mark (`?`) is used as a placeholder in the SQL query.

  If a problem occurs when connecting Firebird on Windows, use the full path to the database file, for example:

  `firebirdsql://{user}:{password}@{server}:{port}/C:\Users\user\firebird\db.FDB`

- ClickHouse

  For example:

  `clickhouse://{user}:{password}@{server}:{port}/{database}`

  A question mark (`?`) is used as a placeholder in the SQL query.

  LogAnalyzer supports the following data types:

  - Data that can be cast to string (such as strings, numeric values, and BLOBs) is displayed as strings.
  - Arrays and maps are displayed in JSON format or using the built-in `go fmt.Sprintf("%v",v)` function to display them in the best possible way.

  Two methods of connecting to ClickHouse are possible:

  - Without credentials, by entering a URL: `clickhouse://host:port/database`
  - With credentials, by entering a URL: `clickhouse://user:password@host:port/database`

  When using TLS encryption, by default, the connector works only on port 9440. If TLS encryption is not used, by default, the connector works with ClickHouse only on port 9000.

  > The connector does not work over HTTP.

  If TLS encryption mode is configured on the ClickHouse server, and in connector settings, in the **TLS mode** drop-down list, you have selected **Disabled** or vice versa, the database connection cannot be established.

  > The **TLS mode** is used only if the ClickHouse driver is specified.

  If you want to connect to the LogAnalyzer ClickHouse, in the **SQL** connector settings, specify the **PublicPKI** secret type, which contains the base64-encoded PEM private key and the public key.

  In the parameters of the **SQL** connector for the ClickHouse connection type, you need to select **Disabled** in the **TLS mode** drop-down list. This value must not be specified if a certificate is used for authentication. If in the **TLS mode** drop-down list, you select **Custom CA**, you need to specify the ID of a secret of the 'certificate' type in the **Identity column** field. You also need to select one of the following values in the **Authorization type** drop-down list:

  - **Disabled**. If you select this value, you need to leave the **Identity column** field blank.
  - **Plain**. Select this value if the **Secret separately** check box is selected and the ID of a secret of the 'credentials' type is specified in the **Identity column** field.
  - **PublicPKI**. Select this value if the **Secret separately** check box is selected and the ID of a secret of the PublicPKI type is specified in the **Identity column** field.

If the initial value of the ID contains an indication of time (`datetime`), in the query, you must use a variable for time conversion (`parseDateTimeBestEffort`). For example, if the time is specified as `2021-01-01 00:10:00`, the following query may be used:

```
select connections, username, host, login_time  from connections where login_time > parseDateTimeBestEffort(?) order by login_time
```

A sequential request for database information is supported in SQL queries. For example, if in the **Query** field, you enter `select * from <name of data table> where id > <placeholder>`, the value of the **Identity seed** field is used as the placeholder value the first time you query the table. In addition, the service that utilizes the SQL connector saves the ID of the last read entry, and the ID of this entry will be used as the placeholder value in the next query to the database.

We recommend adding the `order by` command to the query string, followed by the sorting field. For example, `select * from table_name where id > ? order by id`.

Examples of SQL requests

SQLite, Firebird, MySQL, MariaDB, ClickHouse: `select * from table_name where id > ? order by id`

MsSQL: `select * from table_name where id > @p1 order by id`

PostgreSQL, Cockroach: `select * from table_name where id > $1 order by id`

Oracle: `select * from table_name where id > :val order by id`

Connector, file type

Connectors of the **file** type are used for getting data from text files when working with Windows and Linux agents. One line of a text file is considered to be one event. `\n` is used as the newline character.

If while creating the collector at the Transport step of the Installation Wizard, you specified a connector of the **file** type, at the Event parsing in the **Mapping** table, you can pass the name of the file being processed by the collector or the path to the file in the LogAnalyzer event field. To do this, in the **Source** column, specify one of the following values:

- `$kuma_fileSourceName` to pass the name of the file being processed by the collector in the LogAnalyzer event field.
- `$kuma_fileSourcePath` to pass the path to the file being processed by the collector in the LogAnalyzer event field.

When you use a **file** connector, the new variables in the normalizer will only work with destinations of the internal type.

To read Windows files, you need to create a connector of the **file** type and manually install the agent on Windows. The Windows agent must not read its files in the folder where the agent is installed. The connector will work even with a FAT file system; if the disk is defragmented, the connector re-reads all files from scratch because all inodes of files are reset.

> We do not recommend running the agent under an administrator account; read permissions for folders/files must be configured for the user account of the agent. We do not recommend installing the agent on important systems; it is preferable to send the logs and read them on dedicated hosts with the agent.

For each file that the connector of the **file** type interacts with, a state file (states.ini) is created with the `offset`, `dev`, `inode`, and `filename` parameters. The state file allows the connector, to resume reading from the position where the connector last stopped instead of starting over when rereading the file. Some special considerations are involved in rereading files:

- If the `inode` parameter in the state file changes, the connector rereads the corresponding file from the beginning. When the file is deleting and recreated, the `inode` setting in the associated state file may remain unchanged. In this case, when rereading the file, the connector resumes reading in accordance with the `offset` parameter.
- If the file has been truncated or its size has become smaller, the connector start reading from the beginning.
- If the file has been renamed, when rereading the file, the connector resumes reading from the position where the connector last stopped.
- If the directory with the file has been remounted, when rereading the file, the connector resumes reading from the position where the connector last stopped. You can specify the path to the files with which the connector interacts when configuring the connector in the **File path** field.

Settings for a connector of the **file** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **file**. Required setting.
Tags	Tags for resource search. Optional setting.

Setting	Description

Path to the file.	Connectors of the **file** type are used for getting data from text files when working with Windows and Linux agents. One line of a text f considered to be one event. `\n` is used as the newline character.
	
	The full path to the file that the connector interacts with. Both Linux and Windows path formats are supported, for example:
	• Windows: `C:\var\log\test123.log`
	• Linux: `/var/log/test123.log`
	
	Paths that match the following regular expressions are not allowed:
	• `(?i)^[a-zA-Z]:\\Program Files`
	• `(?i)^[a-zA-Z]:\\Program Files\(x86\)`
	• `(?i)^[a-zA-Z]:\\Windows`
	• `(?i)^[a-zA-Z]:\\ProgramData\\Kaspersky Lab\\KUMA`
	
	In these regular expressions, the double backslashes (\\) are parsed by the system as part of the regular expression syntax and must n manually specified in the path itself.
	File and folder mask templates
	
	Masks:
	• `'*'`—matches any sequence of characters.
	• `'[' [ '^' ] { <range of characters> } ']'`—class of characters (may not be left bla
	• `'?'`—matches any single character.
	
	Ranges of characters:
	• `[0-9]` for numerals
	• `[a-zA-Z]` for Latin alphabet characters
	
	Setting / Descri
	
	Example:
	Masks can be used in both Linux and Windows paths, with backslashes replacing forward slashes as appropriate:
	• Linux:
	  • `/var/log/*som?[1-9].log`
	  • `/mnt/dns_logs/*/dns.log`
	  • `/mnt/proxy/access*.log`
	• Windows:
	  • `C:\var\log\test123.log`
	  • `D:\data\*\report?.txt`
	
	Limitations when using prefixes in file paths

Setting	Description
	Prefixes that cannot be used when specifying paths to files: • /* • /bin • /boot • /dev • /etc • /home • /lib • /lib64 • /proc • /root • /run • /sys • /tmp • /usr/* • /usr/bin/ • /usr/local/* • /usr/local/sbin/ • /usr/local/bin/ • /usr/sbin/ • /usr/lib/ • /usr/lib64/ • /var/* • /var/lib/ • /var/run/ • /opt/kaspersky/kuma/

Setting	Description

	These prefixes are used by the system as regular expressions for checking the path. The * charact prefixes matches any sequence of characters and must not be manually specified in the path itsel not part of a valid path.
	
	Files are available at the following paths:
	
	• `/opt/kaspersky/kuma/clickhouse/logs/`
	
	• `/opt/kaspersky/kuma/metrics/log/`
	
	Limiting the number of files for watching by mask
	
	The number of files simultaneously watched by mask can be limited by the `max_user_watches` se of the Core. To view the value of this setting, run the command:
	
	`cat /proc/sys/fs/inotify/max_user_watches`
	
	If the number of files for watching exceeds the value of the `max_user_watches` setting, the colle cannot read any more events from the files and the following error is written to the collector log:
	
	`Failed to add files for watching {"error": "no space left on device"}`
	
	To make sure that the collector continues to work correctly, you can configure the appropriate rot of files so that the number of files does not exceed the value of the `max_user_watches` setting, increase the `max_user_watches` value.
	
	To increase the value of this setting, run the command:
	
	`sysctl fs.inotify.max_user_watches=<number of files>`
	
	`sysctl -p`
	
	You can also add the value of the `max_user_watches` setting to sysctl.conf so make sure it is kep indefinitely.
	
	After you increase the value of the `max_user_watches` setting, the collector resumes correct operation.
	
	Required setting.

Update timeout, sec	The time in seconds for which the file must not be updated for LogAnalyzer to apply the action specified in the **Timeout action** drop- the file. Default value: `0`, meaning that if the file is not updated, LogAnalyzer does not apply any action to it. The entered value must not be less than the value that you entered on the **Advanced settings** in the **Poll interval, ms** field.
Timeout action	The action that LogAnalyzer applies to the file after the time specified in the **Update timeout, sec**: • **Do nothing**. The default value. • **Add suffix** adds the .kuma_processed extension to the file name and does not process the file even when it is updated. • **Delete** deletes the file.

Setting	Description
Auditd	This toggle switch enables the auditd mechanism to group auditd event lines received from the connector into an auditd event. If you enable this toggle switch, you cannot select a value in the **Delimiter** drop-down list because **\n** is automatically selected for the mechanism. If you enable this toggle switch in the connector settings of the agent, you need to select **\n** in the **Delimiter** drop-down list in the cor settings of the collector to which the agent sends events. The maximum size of a grouped auditd event is approximately 4,174,304 characters. LogAnalyzer classifies Auditd events according to an algorithm. For example, suppose the following records were received for proc `type=LOGIN msg=audit(1712820601.957:21458): pid=4987 uid=0 subj=0:63:0:0 old-auid=4294967295 auid=0 tty= old-ses=4294967295 ses=2348 res=1` `type=SYSCALL msg=audit(1712820601.957:21458): arch=c000003e syscall=1 success=yes exit=1 a0=7 a1=7ffc9a0 a3=0 items=0 ppid=429 pid=4987 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) comm="cron" exe="/usr/sbin/cron" subj=0:63:0:0 key=(null)` `type=PROCTITLE msg=audit(1712820601.957:21458): proctitle=2F7573722F7362696E2F43524F4E002D66` The algorithm gives one single-line event of the LOGIN type (because the LOGIN type has code 1006 and it is less than 1300, which is AUDIT_FIRST_EVENT), and one multi-line event with SYSCALL and PROCTITLE. When grouping a multi-line auditd event containing records with different prefixes, the resulting event uses the prefix of the record th first. This is necessary for the normalizer to parse the event correctly. For example, suppose the following records were received for p `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit_log: type=PROCTITLE msg=audit(1742441361.524:787467): proctitle=686170726F7879002D57002D6462002D66002F7573722F6C6F63616C2F6574632F686170726F78792F686170726F78` `<182>2025-03-20T06:29:22.643145+03:00 test-server-01 tag_audit_log: type=SOCKADDR msg=audit(1742441361.524:787467): saddr=020026FC0A5E41E80000000000000000` `<182>2025-03-20T06:29:22.643143+03:00 test-server-01 tag_audit_log: type=SYSCALL msg=audit(1742441361.52 arch=c000003e syscall=42 success=no exit=-115 a0=1b a1=55d3f02e8c80 a2=10 a3=55d3eec440d0 items=0 ppid=1 pid=1356 auid=4294967295 uid=99 gid=99 euid=99 suid=99 fsuid=99 egid=99 sgid=99 fsgid=99 tty=(none) ses= comm="haproxy" exe="/usr/local/sbin/haproxy" subj=docker-default key="network_events"<` The `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit_log: type=PROCTITLE msg=audit(...)` arrived first, therefore the resulting event uses the `<182>2025-03-20T06:29:22.643147+03:00 test-server-01 tag_audit` prefix.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Buffer size	Buffer size in bytes for accumulating events in the RAM of the server before sending them for further processing or storage. The value must be a positive integer. Default buffer size: 67,108,864 (64 MB), displayed as 0 in the interface.
Number of handlers	Number of handlers that the service can run simultaneously to process response rules in parallel. The value must be a positive integer up to 999.
File/folder polling mode	Specifies how the connector rereads files in the directory: • **Track changes** means the connector rereads files in the directory at an interval in milliseconds specified in the **Poll interval, ms** field if the files are not being updated. The default value. For example, if the files are constantly being updated, and the value of **Poll interval, ms** is `5000`, the connector rereads the files continuously instead of every 5000 milliseconds. If the files are not being updated, the connector rereads them every 5000 milliseconds. • **Track periodic** means the connector rereads files in the directory at an interval in milliseconds specified in the **Poll interval, ms** field, regardless of whether the files are being updated or not.

Setting	Description
Poll interval, ms	The interval in milliseconds at which the connector rereads files in the directory. Default value: `0` means the connector rereads files in the directory every 700 milliseconds. In the **File/folder polling mode** drop-down list, select the mode the connector must use to reread files in the directory. The entered value must not be less than the value that you entered on the **Basic settings** in the **Update timeout, sec** field. We recommend entering a value less than the value that you entered in the **Event Buffer TTL** field because this may adversely affect the performance of Auditd.
Character encoding	Character encoding. The default is **UTF-8**.
Event Buffer TTL	Buffer lifetime for auditd event lines, in milliseconds. Auditd event lines enter the LogAnalyzer collector and accumulate in the buffer. This allows multiple auditd event lines to be grouped into a single auditd event. The buffer lifetime countdown begins when the first auditd event line is received or when the previous buffer lifetime expires. Possible values: 700 to 30,000. The default value is `2000`. This field is available if you have enabled the **Auditd** toggle switch on the **Basic settings** tab. The auditd event lines accumulated in the buffer are kept in the RAM of the server. We recommend caution when increasing the buffer size because memory usage by the LogAnalyzer collector may become excessive. You can verify how much server RAM the LogAnalyzer collector is using in LogAnalyzer metrics. If you want a buffer lifetime to exceed 30,000 milliseconds, we recommend using a different auditd event transport. For example, you can use an agent or pre-accumulate auditd events in a file, and then process this file with the LogAnalyzer collector.
Transport header	Regular expression for auditd events, which is used to identify auditd event lines. You can use the default value or edit it. The regular expression must contain the `record_type_name`, `record_type_value`, and `event_sequence_number` groups. If a multi-line auditd event contains a prefix, the prefix is retained for the first line of the auditd event and discarded for the following lines. You can revert to the default regular expression for auditd events by clicking **Reset to default value**.

Connector, 1c-log type

Connectors of the **1c-log** type are used for getting data from 1C technology logs when working with Linux agents. `\n` is used as the newline character. The connector accepts only the first line from a multi-line event record.

Settings for a connector of the **1c-log** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **1c-log**. Required setting.
Tags	Tags for resource search. Optional setting.

Setting	Description
Directory path	The full path to the directory with the files that you want to interact with, for example, `/var/log/1c/logs/`. Limitations when using prefixes in file paths

Setting	Description
	Prefixes that cannot be used when specifying paths to files: • /* • /bin • /boot • /dev • /etc • /home • /lib • /lib64 • /proc • /root • /run • /sys • /tmp • /usr/* • /usr/bin/ • /usr/local/* • /usr/local/sbin/ • /usr/local/bin/ • /usr/sbin/ • /usr/lib/ • /usr/lib64/ • /var/* • /var/lib/ • /var/run/ • /opt/kaspersky/kuma/

Setting	Description
	These prefixes are used by the system as regular expressions for checking the path. The * character in prefixes matches any sequence of characters and must not be manually specified in the path itself if it is not part of a valid path. Files are available at the following paths: • `/opt/kaspersky/kuma/clickhouse/logs/` • `/opt/kaspersky/kuma/metrics/log/` Required setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

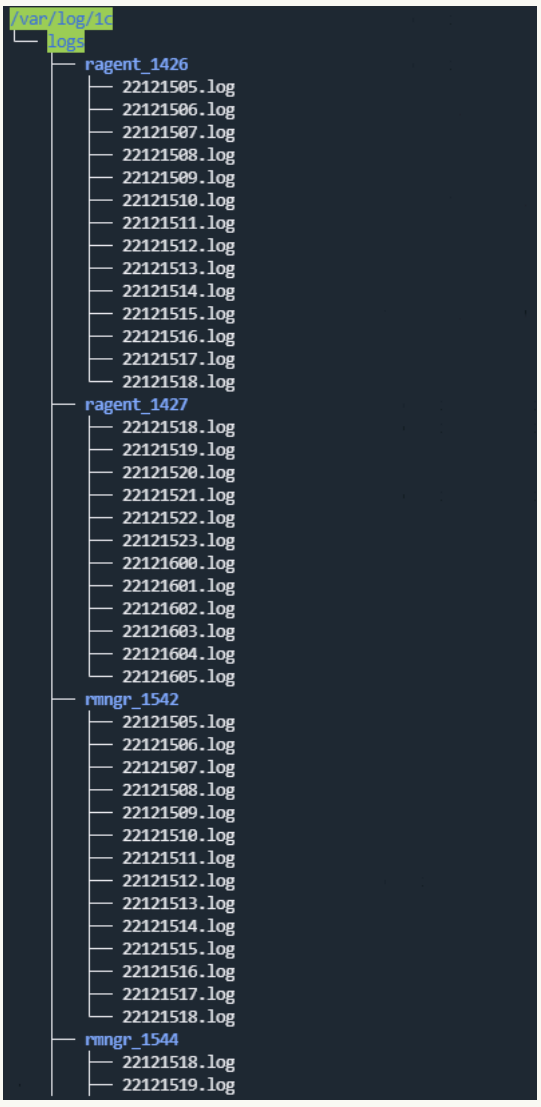
Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Buffer size	Buffer size in bytes for accumulating events in the RAM of the server before sending them for further processing or storage. The value must be a positive integer. Default buffer size: 67,108,864 (64 MB), displayed as 0 in the interface.
File/folder polling mode	Specifies how the connector rereads files in the directory: • **Track changes** means the connector rereads files in the directory at an interval in milliseconds specified in the **Poll interval, ms** field if the files are not being updated. The default value. For example, if the files are constantly being updated, and the value of **Poll interval, ms** is `5000`, the connector rereads the files continuously instead of every 5000 milliseconds. If the files are not being updated, the connector rereads them every 5000 milliseconds. • **Track periodic** means the connector rereads files in the directory at an interval in milliseconds specified in the **Poll interval, ms** field, regardless of whether the files are being updated or not.
Poll interval, ms	The interval in milliseconds at which the connector rereads files in the directory. Default value: `0` means the connector rereads files in the directory every 700 milliseconds. In the **File/folder polling mode** drop-down list, select the mode the connector must use to reread files in the directory.
Character encoding	Buffer size in bytes for accumulating events in the RAM of the server before sending them for further processing or storage. The value must be a positive integer. Default buffer size: 67,108,864 (64 MB), displayed as 0 in the interface.

Connector operation diagram:

1. All 1C technology log files are searched. Log file requirements:

   - Files with the LOG extension are created in the log directory (`/var/log/1c/logs/` by default) within a subdirectory for each process (click the link below to see an image).

     Example of a supported 1C technology log structure

     Example of a supported 1C technology log structure

   - Events are logged to a file for an hour; after that, the next log file is created.
   - The file names have the following format: <YY><MM><DD><HH>.log. For example, 22111418.log is a file created in 2022, in the 11th month, on the 14th at 18:00.
   - Each event starts with the event time in the following format: <mm>:<ss>.<microseconds>-<duration in microseconds>.

2. The processed files are discarded. Information about processed files is stored in the file /<collector working directory>/1c_log_connector/state.json.
3. Processing of the new events starts, and the event time is converted to the RFC3339 format.
4. The next file in the queue is processed.

Connector limitations:

- Installation of a collector with a 1c-log connector is not supported in a Windows operating system. To set up transfer of 1C log files for processing by the KUMA collector:

  1. On the Windows server, grant read access over the network to the folder with the 1C log files.
  2. On the Linux server, mount the shared folder with the 1C log files on the Linux server (see the list of supported operating systems).
  3. On the Linux server, install the collector that you want to process 1C log files from the mounted shared folder.

- Only the first line from a multi-line event record is processed.
- The normalizer processes only the following types of events:
  - ADMIN
  - ATTN
  - CALL
  - CLSTR
  - CONN
  - DBMSSQL
  - DBMSSQLCONN
  - DBV8DBENG
  - EXCP
  - EXCPCNTX
  - HASP
  - LEAKS
  - LIC
  - MEM
  - PROC
  - SCALL
  - SCOM
  - SDBL
  - SESN

- SINTEG
- SRVC
- TLOCK
- TTIMEOUT
- VRSREQUEST
- VRSRESPONSE

Connector, 1c-xml type

Connectors of the **1c-xml** type are used for getting data from 1C registration logs when working with Linux agents. When the connector handles multi-line events, it converts them into single-line events.

Settings for a connector of the **1c-xml** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **1c-xml**. Required setting.
Tags	Tags for resource search. Optional setting.

Setting	Description
Directory path	The full path to the directory with the files that you want to interact with, for example, `/var/log/1c/logs/`. Limitations when using prefixes in file paths

Setting	Description
	Prefixes that cannot be used when specifying paths to files: • /* • /bin • /boot • /dev • /etc • /home • /lib • /lib64 • /proc • /root • /run • /sys • /tmp • /usr/* • /usr/bin/ • /usr/local/* • /usr/local/sbin/ • /usr/local/bin/ • /usr/sbin/ • /usr/lib/ • /usr/lib64/ • /var/* • /var/lib/ • /var/run/ • /opt/kaspersky/kuma/

Setting	Description
	These prefixes are used by the system as regular expressions for checking the path. The * character in prefixes matches any sequence of characters and must not be manually specified in the path itself if it is not part of a valid path. Files are available at the following paths: • `/opt/kaspersky/kuma/clickhouse/logs/` • `/opt/kaspersky/kuma/metrics/log/` Required setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Buffer size	Buffer size in bytes for accumulating events in the RAM of the server before sending them for further processing or storage. The value must be a positive integer. Default buffer size: 67,108,864 (64 MB), displayed as 0 in the interface.
File/folder polling mode	Specifies how the connector rereads files in the directory: • **Track changes** means the connector rereads files in the directory at an interval in milliseconds specified in the **Poll interval, ms** field if the files are not being updated. The default value. For example, if the files are constantly being updated, and the value of **Poll interval, ms** is `5000`, the connector rereads the files continuously instead of every 5000 milliseconds. If the files are not being updated, the connector rereads them every 5000 milliseconds. • **Track periodic** means the connector rereads files in the directory at an interval in milliseconds specified in the **Poll interval, ms** field, regardless of whether the files are being updated or not.
Poll interval, ms	The interval in milliseconds at which the connector rereads files in the directory. Default value: `0` means the connector rereads files in the directory every 700 milliseconds. In the **File/folder polling mode** drop-down list, select the mode the connector must use to reread files in the directory.
Character encoding	Character encoding. The default is **UTF-8**.

Connector operation diagram:

1. The files containing 1C logs with the XML extension are searched within the specified directory. Logs are placed in the directory either manually or using an application written in the 1C language, for example, using the `ВыгрузитьЖурналРегистрации()` function. The connector only supports logs received this way. For more information on how to obtain 1C logs, see the official 1C documentation.

2. Files are sorted by the last modification time in ascending order. All the files modified before the last read are discarded.

   Information about processed files is stored in the file /<collector working directory>/1c_xml_connector/state.ini and has the following format: `offset=< number >\ndev=< number >\ninode=< number >`.

3. Events are defined in each unread file.

4. Events from the file are processed one by one. Multi-line events are converted to single-line events.

Connector limitations:

- Installation of a collector with a 1c-xml connector is not supported in a Windows operating system. To set up transfer of 1C log files for processing by the KUMA collector:

  1. On the Windows server, grant read access over the network to the folder with the 1C log files.
  2. On the Linux server, mount the shared folder with the 1C log files on the Linux server (see the list of supported operating systems).
  3. On the Linux server, install the collector that you want to process 1C log files from the mounted shared folder.

- Files with an incorrect event format are not read. For example, if event tags in the file are in Russian, the collector does not read such events.

  Example of a correct XML file with an event.

```
<?xml version="1.0" encoding="UTF-8"?>
<v8e:EventLog xmlns: v8e="http://v8.1c.ru/eventLog"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <v8e:Event>
    <v8e:Level>Information</v8e:Level>
    <v8e:Date>2022-12-07T01:55:44+03:00</v8e:Date>
    <v8eApplicationName>generator.go</v8e:ApplicationName>
<v8e:ApplicationPresentation>generator.go</v8e:ApplicationPresentation>
    <v8e:Event>Test event type: Count test</v8e:Event>
    <v8e:EventPresentation></v8e:Event Presentation>
    <v8e:User>abcd_1234</v8e:User>
    <v8e:UserName>TestUser</v8e:UserName>
    <v8e:Computer>Test OC</v8e:Computer>
    <v8e:Metadata></v8e:Metadata>
    <v8e:MetadataPresentation></v8e:MetadataPresentation>
    <v8e:Comment></v8e:Comment>
    <v8e:Data>
      <v8e:Name></v8e:Name>
      <v8e:CurrentOSUser></v8e:CurrentOSUser>
    </v8e:Data>
    <v8e:DataPresentation></v8e:DataPresentation>
    <v8e:TransactionStatus>NotApplicable</v8e:TransactionStatus>
    <v8e:TransactionID></v8e:TransactionID>
    <v8e:Connection>0</v8e:Connection>
    <v8e:Session></v8e:Session>
    <v8e:ServerName>kuma-test</v8e:ServerName>
    <v8e:Port>80</v8e:Port>
    <v8e:SyncPort>0</v8e:SyncPort>
  </v8e:Event>
</v8e:EventLog>
```

Example of a processed event.

```
<v8e:Event><v8e:Level>Information</v8e:Level><v8e:Date>2022-12-
07T01:55:44+03:00</v8e:Date><v8e:ApplicationName>generator.go</v8e:ApplicationName><v8e:ApplicationPresentation>generator.go
</v8e:ApplicationPresentation><v8e:Event>Test event type: Count
test</v8e:Event><v8e:EventPresentation></v8e:EventPresentation><v8e:User>abcd_1234</v8e:User><v8e:UserName>TestUser</v8e:Use
rName><v8e:Computer>Test
OC</v8e:Computer><v8e:Metadata></v8e:Metadata><v8e:MetadataPresentation></v8e:MetadataPresentation><v8e:Comment></v8e:Commen
t><v8e:Data><v8e:Name></v8e:Name><v8e:CurrentOSUser></v8e:CurrentOSUser></v8e:Data><v8e:DataPresentation></v8e:DataPresentat
ion><v8e:TransactionStatus>NotApplicable</v8e:TransactionStatus><v8e:TransactionID></v8e:TransactionID><v8e:Connection>0</v8
e:Connection><v8e:Session></v8e:Session><v8e:ServerName>kuma-
test</v8e:ServerName><v8e:Port>80</v8e:Port><v8e:SyncPort>0</v8e:SyncPort></v8e:Event>
```

Example of a processed event

- If a file read by the connector is enriched with the new events and if this file is not the last file read in the directory, all events from the file are processed again.

Connector, diode type

Connectors of the **diode** type are used for unidirectional data transmission in industrial ICS networks using data diodes. Settings for a connector of the **diode** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **diode**. Required setting.
Tags	Tags for resource search. Optional setting.

Setting	Description
Data diode destination directory	Full path to the directory on the LogAnalyzer collector server, into which the data diode moves files with events from the isolated network segment. After the connector has read these files, the files are deleted from the directory. Maximum length of the path: 255 Unicode characters. Limitations when using prefixes in paths

Setting	Description
	Prefixes that cannot be used when specifying paths to files: • /* • /bin • /boot • /dev • /etc • /home • /lib • /lib64 • /proc • /root • /run • /sys • /tmp • /usr/* • /usr/bin/ • /usr/local/* • /usr/local/sbin/ • /usr/local/bin/ • /usr/sbin/ • /usr/lib/ • /usr/lib64/ • /var/* • /var/lib/ • /var/run/ • /opt/kaspersky/kuma/

Setting	Description
	These prefixes are used by the system as regular expressions for checking the path. The * character in prefixes matches any sequence of characters and must not be manually specified in the path itself if it is not part of a valid path. Files are available at the following paths: • `/opt/kaspersky/kuma/clickhouse/logs/` • `/opt/kaspersky/kuma/metrics/log/` Required setting.
Delimiter	The character that marks the boundary between events: • **\n** • **\t** • **\0** If you do not select a value in this drop-down list, an empty value is selected by default. In this case, the connector does not split one event into several and passes it to the normalizer unchanged. You must select the same value in the **Delimiter** drop-down list in the settings of the connector and the destination being used to transmit events from the isolated network segment using a data diode.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Number of handlers	Number of handlers that the service can run simultaneously to process response rules in parallel. The value must be a positive integer up to 999.
Poll interval, sec	Interval at which the files are read from the directory containing events from the data diode. The default value is 2 seconds.
Character encoding	Character encoding. The default is **UTF-8**.
Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**. You must select the same value in the **Snappy** drop-down list in the settings of the connector and the destination being used to transmit events from the isolated network segment using a data diode.

Connector, ftp type

Connectors of the **ftp** type are used for getting data over File Transfer Protocol (FTP) when working with Windows and Linux agents. Settings for a connector of the **ftp** type are described in the following tables.

Basic settings tab

<table>
<tr><th>Setting</th><th>Description</th></tr>
<tr><td>Name</td><td>Unique name of the resource. The maximum length of the name is 128 Unicode characters.
Required setting.</td></tr>
<tr><td>Tenant</td><td>The name of the tenant that owns the resource.
Required setting.</td></tr>
<tr><td>Type</td><td>Connector type: ftp.
Required setting.</td></tr>
<tr><td>Tags</td><td>Tags for resource search.
Optional setting.</td></tr>
<tr><td>URL</td><td>URL of file or file mask that begins with the <code>'ftp://'</code> schema. You can use <code>* ? [...]</code> for the file mask.
File mask templates

Masks:
• <code>'*'</code> —matches any sequence of characters.
• <code>'[' ['^'] { <range of characters> } ']'</code> —class of characters (may not be left blank).
• <code>'?'</code> —matches any single character.

Ranges of characters:
• <code>[0-9]</code> for numerals
• <code>[a-zA-Z]</code> for Latin alphabet characters

Example:
Masks can be used in both Linux and Windows paths, with backslashes replacing forward slashes as appropriate:
• Linux:
  • <code>/var/log/*som?[1-9].log</code>
  • <code>/mnt/dns_logs/*/dns.log</code>
  • <code>/mnt/proxy/access*.log</code>
• Windows:
  • <code>C:\var\log\test123.log</code>
  • <code>D:\data*\report?.txt</code>

If the URL does not contain the port number of the FTP server, port <code>21</code> is automatically specified.
Required setting.</td></tr>
<tr><td>Secret</td><td></td></tr>
<tr><td>Description</td><td>Description of the resource. The maximum length of the description is 4000 Unicode characters.</td></tr>
</table>

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.

Connector, nfs type

Connectors of the **nfs** type are used for getting data over Network File System (NFS) when working with Windows and Linux agents. Settings for a connector of the **nfs** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **nfs**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	Path to the remote directory in the `nfs://< host name >/< path >` format. Required setting.
File name mask	A mask used to filter files containing events. The following wildcards are acceptable "`*`", "`?`", "`[...]`".
Poll interval, sec	Poll interval in seconds. The time interval after which files are re-read from the remote system. The default value is `0`.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.

Connector, wmi type

Connectors of the **wmi** type are used for getting data using Windows Management Instrumentation when working with Windows agents. Settings for a connector of the **wmi** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **wmi**. Required setting.
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.
URL	URL of the collector that you created to receive data using Windows Management Instrumentation, for example, `kuma-collector.example.com:7221`. When a collector is created, an agent is automatically created that will get data on the remote device and forward it to the collector service. If you know which server the collector service will be installed on, the URL is known in advance. You can enter the URL of the collector in the **URL** field after completing the installation wizard. To do so, you first need to copy the URL of the collector in the **Resources** → **Active services** section. Required setting.

Setting	Description
Default credentials	In the **Remote hosts** table, you need to specify credentials for connecting to hosts that can be used by default. This field is required if the secret for at least one host is not specified in the **Remote hosts** table.
Remote hosts	Settings of remote Windows devices to connect to. • **Server** is the IP address or name of the device from which you want to receive data, for example, `machine-1`. Required setting. • **Domain** is the name of the domain in which the remote device resides, for example, `example.com`. Required setting. • **Log type** are the names of the Windows logs that you want to get. By default, this drop-down list includes only preconfigured logs, but you can add custom log to the list. To do so, enter the names of the custom logs in the **Windows logs** field, then press **ENTER**. LogAnalyzer service and resource configurations may require additional changes in order to process custom logs correctly. Logs that are available by default: ◦ Application ◦ ForwardedEvents ◦ Security ◦ System ◦ HardwareEvents If a WMI connection uses at least one log with an incorrect name, the agent that uses the connector does not receive events from all the logs within this connection, even if the names of other logs are specified correctly. The WMI agent connections for which all log names are specified correctly will work properly. Required setting. • In the **Secret** field, specify account credentials for accessing a remote Windows device with permissions to read the logs. You can select an existing secret or create a new secret. To create a new secret, select **Create new**. When creating a secret, you do not need to specify the domain for accessing the host in the **User** field because the **Domain** value from the **Remote hosts** table is used. If you want to edit the settings of an existing secret, click the pencil icon. If you leave the **default** value in this field, the credentials specified in the **Default credentials** field are used when connecting to remote Windows devices. How to create a secret? *To create a secret:* 1. In the **Name** field, enter the name of the secret. 2. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector. 3. If necessary, enter a description of the secret in the **Description** field. 4. Click the **Create** button. The secret is added and displayed in the **Secret** drop-down list. You can add multiple remote Windows devices or remove a remote Windows device. To add a remote Windows device, click **+ Add**. To remove a remote Windows device, select the check box next to it and click **Delete**. You can also click the **Import** button to import a file with a list of remote Windows devices in TSV format, which lets you avoid filling in fields manually when you have many such devices. LogAnalyzer automatically fills in the data from the file and performs a check. An error is displayed for incorrectly filled fields. You can edit these fields manually and save the settings.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.
TLS mode	TLS encryption mode. When using TLS encryption, you cannot specify an IP address in the **URL** field on the **Basic settings** tab. Available values: • **Disabled** means TLS encryption is not used. This value is selected by default. • **Enabled** means TLS encryption is used, but certificates are not verified. • **With verification** means TLS encryption is used with verification of the certificate signed with the LogAnalyzer root certificate. The root certificate and key of LogAnalyzer are created automatically during application installation and are stored on the LogAnalyzer Core server in the /opt/kaspersky/kuma/core/certificates/ directory. Obsolete encryption mode, deprecated. Will be removed in future versions of LogAnalyzer.
Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**.

If you edit a connector of this type, the **TLS mode** and **Compression** settings are visible and available on the connector resource as well as the collector. If you are using a connector of this type on a collector, the values of **TLS mode** and **Compression** settings are sent to the destination of automatically created agents.

Receiving events from a remote device

Conditions for receiving events from a remote Windows device hosting a LogAnalyzer agent:

- To start the LogAnalyzer agent on the remote device, you must use an account with the “Log on as a service” permissions.
- To receive events from the LogAnalyzer agent, you must use an account with Event Log Readers permissions. For domain servers, one such user account can be created so that a group policy can be used to distribute its rights to read logs to all servers and workstations in the domain.
- TCP ports 135, 445, and 49152–65535 must be opened on the remote Windows devices.
- You must run the following services on the remote machines:
  - Remote Procedure Call (RPC)
  - RPC Endpoint Mapper

Connector, wec type

Connectors of the **wec** type are used for getting data using Windows Event Forwarding (WEF) and Windows Event Collector (WEC), or local operating system logs of a Windows host when working with Windows agents. Settings for a connector of the **wec** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **wec**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	URL of the collector that you created to receive data using Windows Event Collector, for example, `kuma-collector.example.com:7221`. When a collector is created, an agent is automatically created that will get data on the remote device and forward it to the collector service. If you know which server the collector service will be installed on, the URL is known in advance. You can enter the URL of the collector in the **URL** field after completing the installation wizard. To do so, you first need to copy the URL of the collector in the **Resources** → **Active services** section. Required setting.
Windows logs	The names of the Windows logs that you want to get. By default, this drop-down list includes only preconfigured logs, but you can add custom log to the list. To do so, enter the names of the custom logs in the **Windows logs** field, then press **ENTER**. LogAnalyzer service and resource configurations may require additional changes in order to process custom logs correctly. Preconfigured logs: • Application • ForwardedEvents • Security • System • HardwareEvents If the name of at least one log is specified incorrectly, the agent using the connector does not receive events from any log, even if the names of other logs are correct. Required setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.

Setting	Description
TLS mode	TLS encryption mode. When using TLS encryption, you cannot specify an IP address in the **URL** field on the **Basic settings** tab. Available values: • **Disabled** means TLS encryption is not used. This value is selected by default. • **Enabled** means TLS encryption is used, but certificates are not verified. • **With verification** means TLS encryption is used with verification of the certificate signed with the LogAnalyzer root certificate. The root certificate and key of LogAnalyzer are created automatically during application installation and are stored on the LogAnalyzer Core server in the /opt/kaspersky/kuma/core/certificates/ directory. Obsolete encryption mode, deprecated. Will be removed in future versions of LogAnalyzer.
Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**.

If you edit a connector of this type, the **TLS mode** and **Compression** settings are visible and available on the connector resource as well as the collector. If you are using a connector of this type on a collector, the values of **TLS mode** and **Compression** settings are sent to the destination of automatically created agents.

To start the KUMA agent on the remote device, you must use a service account with the "Log on as a service" permissions. To receive events from the operating system log, the service user account must also have Event Log Readers permissions.

You can create one user account with "Log on as a service" and "Event Log Readers" permissions, and then use a group policy to extend the rights of this account to read the logs to all servers and workstations in the domain.

We recommend that you disable interactive logon for the service account.

Connector, etw type

Connectors of the **etw** type are used for getting extended logs of DNS servers. Settings for a connector of the **etw** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **etw**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	URL of the DNS server. Required setting.

Setting	Description
Session name	Session name that corresponds to the ETW provider: Microsoft-Windows-DNSServer {EB79061A-A566-4698-9119-3ED2807060E7}. If in a connector of the **etw** type, the session name is specified incorrectly, an incorrect provider is specified in the session, or an incorrect method is specified for sending events (to send events correctly, on the Windows Server side, you must specify "Real time" or "File and Real time" mode), events will not arrive from the agent, an error will be recorded in the agent log on Windows, and the status of the agent will be green. At the same time, no attempt will be made to get events every 60 seconds. If you modify session settings on the Windows side, you must restart the etw agent and/or the session for the changes to take effect. For details about specifying session settings on the Windows side to receive DNS server events, see the Configuring receipt of DNS server events using the ETW agent section. Required setting.
Extract event information	This toggle switch enables the extraction of the minimum set of event information that can be obtained without having to download third-party metadata from the disk. This method helps conserve CPU resources on the computer with the agent. By default, this toggle switch is enabled and all event data is extracted.
Extract event properties	This toggle switch enables the extraction of event properties. If this toggle switch is disabled, event properties are not extracted, which helps save CPU resources on the machine with the agent. By default, this toggle switch is enabled and event properties are extracted. You can enable the **Extract event properties** switch only if the **Extract event information** toggle switch is enabled.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.
TLS mode	TLS encryption mode. When using TLS encryption, you cannot specify an IP address in the **URL** field on the **Basic settings** tab. Available values: • **Disabled** means TLS encryption is not used. This value is selected by default. • **Enabled** means TLS encryption is used, but certificates are not verified. • **With verification** means TLS encryption is used with verification of the certificate signed with the LogAnalyzer root certificate. The root certificate and key of LogAnalyzer are created automatically during application installation and are stored on the LogAnalyzer Core server in the /opt/kaspersky/kuma/core/certificates/ directory. Obsolete encryption mode, deprecated. Will be removed in future versions of LogAnalyzer.
Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**.

If you edit a connector of this type, the **TLS mode** and **Compression** settings are visible and available on the connector resource as well as the collector. If you are using a connector of this type on a collector, the values of **TLS mode** and **Compression** settings are sent to the destination of automatically created agents.

Connector, snmp type

Connectors of the **snmp** type are used for getting data over Simple Network Management Protocol (SNMP) when working with Windows and Linux agents. To process events received over SNMP, you must use the json normalizer. Supported SNMP protocol versions:

- snmpV1
- snmpV2
- snmpV3

Only one **snmp** connector created in the agent settings can be used in an agent. If you need to use multiple **snmp** connectors, you must create one or all **snmp** connectors as a separate resource and select it in the connection settings.

Available settings for a connector of the **snmp** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **snmp**. Required setting.
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Setting	Description

SNMP resource	Settings for connecting to an SNMP resource:
	• **SNMP version** is the version of the SNMP protocol being used.
	Required setting.
	• **Host** is the name or IP address of the host. Possible formats:
	  • `<host name>`
	  • `<IPv4 address>`
	  • `<IPv6 address>`
	  Required setting.
	• **Port** is the port number to be used when connecting to the host. Typical values are `161` or `162`.
	Required setting.
	• **Secret** is the secret that stores the credentials for connecting over the Simple Network Management Protocol. The secret type must match the SNMP version.
	You can select an existing secret or create a new secret. To create a new secret, select **Create new**.
	If you want to edit the settings of an existing secret, click the pencil icon next to it.
	How to create a secret?
	To create a secret:
	1. In the **Name** field, enter the name of the secret.
	2. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector.
	3. If necessary, enter a description of the secret in the **Description** field.
	4. Click the **Create** button.
	The secret is added and displayed in the **Secret** drop-down list.
	Required setting.
	You can add multiple connections to SNMP resources or delete an SNMP resource connection. To create a connection to an SNMP resource, click the **+ SNMP resource** button. To delete a connection to an SNMP resource, click the delete icon next to the SNMP resource.

Settings	Rules for naming the received data, according to which OIDs (object identifiers) are converted to the keys with which the normalizer can interact. Available settings: • **Parameter name** is the name for the data type, for example, `Host name` or `Host uptime`. Required setting. • **OID** is a unique identifier that determines where to look for the required data at the event source, for example, `1.3.6.1.2.1.1.5`. LogAnalyzer requests data using the SNMP GET command. Required setting. • **Key** is a unique identifier returned in response to a request to the device with the value of the requested parameter, for example, `sysName`. You can reference the key when normalizing data. Required setting. • If the **MAC address** check box is selected, LogAnalyzer correctly decodes data where the OID contains information about the MAC address in OctetString format. After decoding, the MAC address is converted to a `String` value of the XX:XX:XX:XX:XX:XX format. You can do the following with rules: • Add multiple rules. To add a rule, click the **+ Add** button. • Delete rules. To delete a rule, select the check box next to it and click **Delete**. • Clear rule settings. To do so, click the **Clear all values** button.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.

Connector, snmp-trap type

Connectors of the **snmp-trap** type are used for passively receiving events using SNMP traps when working with Windows and Linux agents. The connector receives snmp-trap events and prepares them for normalization by mapping SNMP object IDs to temporary keys. Then the message is passed to the JSON normalizer, where the temporary keys are mapped to the LogAnalyzer fields and an event is generated. To process events received over SNMP, you must use the json normalizer. Supported SNMP protocol versions:

- snmpV1
- snmpV2

Settings for a connector of the **snmp-trap** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **snmp-trap**. Required setting.
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Setting	Description

SNMP resource	Connection settings for receiving snmp-trap events:
	• **SNMP version** is the version of the SNMP protocol being used:
	  • **snmpV1**
	  • **snmpV2**
	  For example, Windows uses the snmpV2 version of the SNMP protocol by default.
	Required setting.
	• **URL** is the URL for receiving SNMP trap events. You can enter a URL in one of the following formats:
	  • `< host name >:< port number >`
	  • `< IPv4 address >:< port number >`
	  • `< IPv6 address >:< port number >`
	  • `:< port number >`
	  Required setting.
	You can add multiple connections or delete a connection. To add a connection, click the **+ SNMP resource** button. To remove a SNMP resource, click the delete ✕ icon next to it.

Settings	Rules for naming the received data, according to which OIDs (object identifiers) are converted to the keys with which the normalizer can interact. Available settings: • **Parameter name** is the name for the data type, for example, `Host name` or `Host uptime`. Required setting. • **OID** is a unique identifier that determines where to look for the required data at the event source, for example, `1.3.6.1.2.1.1.5`. LogAnalyzer requests data using the SNMP GET command. Required setting. • **Key** is a unique identifier returned in response to a request to the device with the value of the requested parameter, for example, `sysName`. You can reference the key when normalizing data. Required setting. • If the **MAC address** check box is selected, LogAnalyzer correctly decodes data where the OID contains information about the MAC address in OctetString format. After decoding, the MAC address is converted to a `String` value of the XX:XX:XX:XX:XX:XX format. You can do the following with rules: • Add multiple rules. To add a rule, click the **+ Add** button. • Delete rules. To delete a rule, select the check box next to it and click **Delete**. • Clear rule settings. To do so, click the **Clear all values** button. • Populate the table with mappings for OID values received in WinEventLog logs. To do this, click the **Apply OIDs for WinEventLog** button. If more data needs to be determined and normalized in the incoming events, add to the table rows containing OID objects and their keys. Data is processed according to the allow list principle: objects that are not specified in the table are not sent to the normalizer for further processing.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**. When receiving snmp-trap events from Windows with Russian localization, if you encounter invalid characters in the event, we recommend changing the character encoding in the **snmp-trap** connector to Windows 1251.

Configuring the source of SNMP trap messages for Windows

Configuring a Windows device to send SNMP trap messages to the LogAnalyzer collector involves the following steps:

1. Configuring and starting the SNMP and SNMP trap services
2. Configuring the Event to Trap Translator service

Events from the source of SNMP trap messages must be received by the LogAnalyzer collector, which uses a connector of the snmp-trap type and a json normalizer.

Configuring and starting the SNMP and SNMP trap services

To configure and start the SNMP and SNMP trap services in Windows 10:

1. Open **Settings** → **Apps** → **Apps and features** → **Optional features** → **Add feature** → **Simple Network Management Protocol (SNMP)** and click **Install**.
2. Wait for the installation to complete and restart your computer.
3. Make sure that the SNMP service is running. If any of the following services are not running, enable them:
   - **Services** → **SNMP Service**.
   - **Services** → **SNMP Trap**.
4. Right-click **Services** → **SNMP Service**, and in the context menu select **Properties**. Specify the following settings:
   - On the **Log On** tab, select the **Local System account** check box.
   - On the **Agent** tab, fill in the **Contact** (for example, specify `User-win10`) and Location (for example, specify `detroit`) fields.
   - On the **Traps** tab:
     - In the **Community Name** field, enter **community public** and click **Add to list**.
     - In the **Trap destination** field, click **Add**, specify the IP address or host of the LogAnalyzer server on which the collector that waits for SNMP events is deployed, and click **Add**.
   - On the **Security** tab:
     - Select the **Send authentication trap** check box.
     - In the **Accepted community names** table, click **Add**, enter **Community Name public** and specify **READ WRITE** as the **Community rights**.
     - Select the **Accept SNMP packets from any hosts** check box.
5. Click **Apply** and confirm your selection.
6. Right click **Services** → **SNMP Service** and select **Restart**.

To configure and start the SNMP and SNMP trap services in Windows XP:

1. Open **Start** → **Control Panel** → **Add or Remove Programs** → **Add / Remove Windows Components** → **Management and Monitoring Tools** → **Details**.
2. Select **Simple Network Management Protocol** and **WMI SNMP Provider**, and then click **OK** → **Next**.
3. Wait for the installation to complete and restart your computer.

4. Make sure that the SNMP service is running. If any of the following services are not running, enable them by setting the **Startup type** to **Automatic**:

   - **Services** → **SNMP Service**.
   - **Services** → **SNMP Trap**.

5. Right-click **Services** → **SNMP Service**, and in the context menu select **Properties**. Specify the following settings:

   - On the **Log On** tab, select the **Local System account** check box.
   - On the **Agent** tab, fill in the **Contact** (for example, specify `User-win10`) and Location (for example, specify `detroit`) fields.
   - On the **Traps** tab:
     - In the **Community Name** field, enter **community public** and click **Add to list**.
     - In the **Trap destination** field, click **Add**, specify the IP address or host of the LogAnalyzer server on which the collector that waits for SNMP events is deployed, and click **Add**.
   - On the **Security** tab:
     - Select the **Send authentication trap** check box.
     - In the **Accepted community names** table, click **Add**, enter **Community Name public** and specify **READ WRITE** as the **Community rights**.
     - Select the **Accept SNMP packets from any hosts** check box.

6. Click **Apply** and confirm your selection.

7. Right click **Services** → **SNMP Service** and select **Restart**.

Changing the port for the SNMP trap service

You can change the SNMP trap service port if necessary.

To change the port of the SNMP trap service:

1. Open the C:\Windows\System32\drivers\etc folder.

2. Open the **services** file in Notepad as an administrator.

3. In the **service name** section of the file, specify the snmp-trap connector port added to the LogAnalyzer collector for the **SNMP trap** service.

4. Save the file.

5. Open the Control Panel and select **Administrative Tools** → **Services**.

6. Right-click **SNMP Service** and select **Restart**.

Configuring the Event to Trap Translator service

To configure the Event to Trap Translator service that translates Windows events to SNMP trap messages:

1. In the command line, type `evntwin` and press **Enter**.
2. Under **Configuration type**, select **Custom**, and click the **Edit** button.
3. In the **Event sources** group of settings, use the **Add** button to find and add the events that you want to send to LogAnalyzer collector with the SNMP trap connector installed.
4. Click the **Settings** button, in the opened window, select the **Don't apply throttle** check box, and click **OK**.
5. Click **Apply** and confirm your selection.

Connector, kata/edr type

Connectors of the **kata/edr** type are used for getting KEDR data via the API. Settings for a connector of the **kata/edr** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **kata/edr**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	URL that you want to connect to. The following URL formats are supported: • `< host name >:< port number >` • `< IPv4 address >:< port number >` • `< IPv6 address >:< port number >` You can specify IPv6 addresses in the following format: `[< IPv6 address >%< interface >:< port number >`, for example, `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`. You can add multiple URLs or remove an URL. LogAnalyzer does not allow saving a resource or service if the URL field contains a tab or space character. To add an URL, click the **+ Add** button. To remove an URL, click the delete ✕ icon next to it. Required setting.

Setting	Description
Secret	Secret that stores the credentials for connecting to the KATA/EDR server. You can select an existing secret or create a new secret. To create a new secret, select **Create new**. If you want to edit the settings of an existing secret, click the pencil icon next to it. How to create a secret? *To create a secret:* 1. In the **Name** field, enter the name of the secret. 2. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector. 3. If necessary, enter a description of the secret in the **Description** field. 4. Click the **Create** button. The secret is added and displayed in the **Secret** drop-down list. Required setting.
External ID	Identifier for external systems. KUMA automatically generates an ID and populates this field with it.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. We only recommend configuring a conversion if you find invalid characters in the fields of the normalized event. By default, no value is selected.
Number of events	Maximum number of events in one request. By default, the value set on the KATA/EDR server is used.
Events fetch timeout	The time in seconds to wait for receipt of events from the KATA/EDR server. Default value: `0`, which means that the value set on the KATA/EDR server is used.
Client timeout	Time in seconds to wait for a response from the KATA/EDR server. Default value: `0`, corresponding to 1800 seconds.
KEDRQL filter	Filter of requests to the KATA/EDR server. For more details on the query language, please refer to the KEDR Help.

Connector, vmware type

Connectors of the **vmware** type are used for getting VMware vCenter data via the API. Settings for a connector of the **vmware** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.

Setting	Description
Type	Connector type: **vmware**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	URL of the VMware API. You need to include the hostname and port number in the URL. You can only specify one URL. Required setting.
VMware credentials	Secret that stores the user name and password for connecting to the VMware API. You can select an existing secret or create a new secret. To create a new secret, select **Create new**. If you want to edit the settings of an existing secret, click the pencil icon next to it. How to create a secret? *To create a secret:* 1. In the **Name** field, enter the name of the secret. 2. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector. 3. If necessary, enter a description of the secret in the **Description** field. 4. Click the **Create** button. The secret is added and displayed in the **Secret** drop-down list. Required setting.
Sleep time, sec	Time to wait after a request that did not return events before making a new request. The default value is 5 seconds. If you specify `0` , the default value is used.
Fetch size	Number of events requested from the VMware API in one request. The default value is `100`. The maximum value is `1000`.
Start timestamp	Starting date and time from which you want to read events from the VMware API. By default, events are read from the VMware API from the time when the collector was started. If started after the collector is stopped, the events are read from the last saved date.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.

TLS mode	TLS encryption mode. When using TLS encryption, you cannot specify an IP address in the **URL** field on the **Basic settings** tab. Available values:
	• **Disabled** means TLS encryption is not used. This value is selected by default.
	• **Enabled** means TLS encryption is used, but certificates are not verified.
	• **Custom CA** means TLS encryption is used with verification that the certificate was signed by a Certificate Authority. If you select this value, in the **Custom CA** drop-down list, specify a secret with a certificate signed by a certification authority. You can select an existing secret or create a new secret. To create a new secret, select **Create new**.
	If you want to edit the settings of an existing secret, click the pencil icon next to it.
	How to create a certificate signed by a Certificate Authority?
	
	You can create a CA-signed certificate on the LogAnalyzer Core server (the following command examples use OpenSSL).
	
	To create a certificate signed by a Certificate Authority:
	
	1. Generate a key to be used by the Certificate Authority, for example:
	`openssl genrsa -out ca.key 2048`
	
	2. Create a certificate for the generated key, for example:
	`openssl req -new -x509 -days 365 -key ca.key -subj "/CN=<common host name of Certificate Authority>" -out ca.crt`
	
	3. Create a private key and a request to have it signed by the Certificate Authority, for example:
	`openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN=<common host name of LogAnalyzer server>" -out server.csr`
	
	4. Create the certificate signed by the Certificate Authority. You need to include the domain names or IP addresses of the server for which you are creating the certificate in the `subjectAltName` variable, for example:
	`openssl x509 -req -extfile <(printf "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1") -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt`
	
	5. Upload the created `server.crt` certificate in the LogAnalyzer web interface to a secret of the **certificate** type, then in the **Custom CA** drop-down list, select the secret of the **certificate** type.
	
	To use LogAnalyzer certificates on third-party devices, you must change the certificate file extension from CERT to CRT. Otherwise, you can get the `x509: certificate signed by unknown authority` error.

Connector, elastic type

Connectors of the **elastic** type are used for getting Elasticsearch data. Elasticsearch version 7.0.0 is supported. Settings for a connector of the **elastic** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **elastic**. Required setting.
Tags	Tags for resource search. Optional setting.

Setting	Description
Connection	Elasticsearch server connection settings: • **URL** is the URL of the Elasticsearch server. You can add multiple URLs or remove an URL. To add an URL, click the **+ Add** button. To remove an URL, click the delete ✕ icon next to it. Required setting. • **Index name** is the name of the index in Elasticsearch. Required setting. • **Query** is the Elasticsearch query. We recommend specifying the `size` parameter in the query to prevent performance problems with LogAnalyzer and Elasticsearch, as well as the `sort` parameter for the sorting order. The following values are possible for the `sort` parameter in the query: `asc`, `desc`, or a custom sorting order by specific fields in accordance with the Elasticsearch syntax. To sort by a specific field, we recommend also specifying the `"missing" : "_first"` parameter next to the `"order"` parameter to prevent errors in cases when this field is absent in any document. For example, `"sort": { "DestinationDnsDomain.keyword": {"order": "desc", "missing" : "_first" } }`. For more details on sorting, please refer to the Elasticsearch documentation. Query example: `"query" : { "match_all" : {} }, "size" : 25, "sort": {"_doc" : "asc"}` Required setting. • **Elastic credentials** is the secret that stores the credentials for connecting to the Elasticsearch server. You can select an existing secret or create a new secret. To create a new secret, select **Create new**. If you want to edit the settings of an existing secret, click the pencil icon next to it. How to create a secret? *To create a secret:* 1. In the **Name** field, enter the name of the secret. 2. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector. 3. If necessary, enter a description of the secret in the **Description** field. 4. Click the **Create** button. The secret is added and displayed in the **Secret** drop-down list. • **Elastic fingerprint** is the secret that stores secrets of the 'fingerprint' type for connecting to the Elasticsearch server and secrets of the 'certificate' type for using a CA certificate. You can select an existing secret or create a new secret. To create a new secret, select **Create new**. If you want to edit the settings of an existing secret, click the pencil icon next to it. • **Poll interval, sec** is the interval between queries to the Elasticsearch server in seconds if the previous query did not return any events. If Elasticsearch contained events at the time of the request, the connector will receive events until all available events have been received from Elasticsearch. You can add multiple connections to an Elasticsearch server resources or delete an Elasticsearch server connection. To add an Elasticsearch server connection, click the **+ Add connection** button. To delete an Elasticsearch server connection, click the delete ✕ icon next to the Elasticsearch server.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.

Connector, office365 type

Connectors of the **office365** type are used for receiving Microsoft 365 (Office 365) data via the API.

Available settings for a connector of the **office365** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **office365**. Required setting.
Tags	Tags for resource search. Optional setting.
Office365 content types	Content types that you want to receive in LogAnalyzer. The following content types are available, providing information about actions and events in Microsoft 365, grouped by information source: • **Audit.General** • **Audit.AzureActiveDirectory** • **Audit.Exchange** • **Audit.Sharepoint** • **DLP.All** You can find detailed information about the properties of the available content types and related events in the schema on the Microsoft website. Required setting. You can select one or more content types.
Office365 tenant ID	Unique ID that you get after registering an account with Microsoft 365. If you do not have one, contact your administrator or Microsoft. Required setting.
Office365 client ID	Unique ID that you get after registering an account with Microsoft 365. If you do not have one, contact your administrator or Microsoft. Required setting.
Authorization	Authorization method for connecting to Microsoft 365. The following authorization methods are available: • **PFX**. Using a PFX secret. • **Token**. Using a 'token' secret. For more information, see the section on secrets.
Office365 credentials	The field becomes available after selecting the authorization method. You can select one of the available authorization secrets or create a new secret of the selected type. Required setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.
Authentication endpoint	The URL that is used for connection and authorization. By default, a connection is made to https://login.microsoftonline.com.
Resource endpoint	URL from which the events are to be received. The default address is https://manage.office.com.

Setting	Description
Look back interval, hours	The period for which all new events are requested, in hours. To avoid losing some events, it is important to set overlapping event reception intervals, because some types of Microsoft 365 content may be sent with a delay. In this case, previously received events are not duplicated. By default, all new events for the last 12 hours are requested.
Request timeout, sec	Time to wait for a response to a request to get new events, in seconds. The default response timeout is 30 seconds.
Retry interval, sec	The time in seconds after which a failed request to get new events must be repeated. By default, a request to get new events is repeated 10 seconds after getting an error or no response within the specified timeout.
Clean interval, sec	How often obsolete data is deleted, in seconds. The minimum value is 300 seconds. By default, obsolete data is deleted every 1800 seconds.
Poll interval, min	How often requests for new events are sent, in minutes. By default, requests are sent every 10 minutes.
Proxy server	Proxy settings, if necessary to connect to Microsoft 365. You can select one of the available proxy servers or create a new proxy server.

Connector, dfi type

Connectors of the **dfi** are used for receiving Kaspersky Digital Footprint Intelligence data via the API. Settings for a connector of the **dfi** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **dfi**. Required setting.
Tags	Tags for resource search. Optional setting.

URL	URL of the Kaspersky Threat Intelligence Portal service API. In the URL, you need to include the hostname and port number, or just the hostname. You can only specify one URL. Kaspersky TIP address: https://tip.kaspersky.com/api.
	The following URL formats are supported:
	• `host name : port` or `host name`
	Examples:
	`http://tip_hostname:2134/api`
	`http://tip_hostname/api`
	• `IPv4 address : port` or `IPv4 address`
	Examples:
	`http://127.0.0.1:2134/api`
	`http://127.0.0.1/api`
	• `[ IPv6 address ]: port` or `[ IPv6 address ]`.
	The IPv6 address must be specified in square brackets. Examples:
	`http://[1050:0:0:0:5:600:300c:326b]:2134/api`
	`http://[1050:0:0:0:5:600:300c:326b]/api`.
	LogAnalyzer does not allow saving a resource or service if the URL field contains a tab or space character.
	Required setting.

Setting	Description
Secret	Secret that stores the credentials for connecting to the Kaspersky Threat Intelligence Portal server. You can select an existing secret or create a new secret. To create a new secret, select **Create new**. If you want to edit the settings of an existing secret, click the pencil icon next to it. How to create a secret? *To create a secret:* 1. In the **Name** field, enter the name of the secret. 2. In the **Token** field, specify the API token for authentication on the Kaspersky Threat Intelligence Portal. You can find detailed information on getting an API token in the Kaspersky Threat Intelligence Portal Help. 3. If necessary, in the **Tags** field, add tags for the secret. 4. If necessary, enter a description of the secret in the **Description** field. 5. Click the **Create** button. The secret is added and displayed in the **Secret** drop-down list. Required setting.
Poll interval	The interval in seconds between requests to the Kaspersky Digital Footprint Intelligence service. Enter a number from 60 to 3600 (that is, from 1 minute to 60 minutes). The default value is `120`. Required setting.
Start timestamp	The starting date and time for the data that the Kaspersky Digital Footprint Intelligence service generates in response to the request. By default, a request to Kaspersky Digital Footprint Intelligence is generated with data for the last 30 days up to the moment when the collector was started. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	The switch enables resource logging. The toggle switch is turned off by default.
Proxy server	Proxy server settings, if one is necessary to connect to the Kaspersky Threat Intelligence Portal server. You can select one of the available proxy servers or create a new proxy server. By default, a proxy server is not used.
Character encoding	Character encoding. The default is **UTF-8**.

Predefined connectors

The connectors listed in the table below are included in the LogAnalyzer distribution kit.

Predefined connectors

Connector name	Comment
[OOTB] Continent SQL	Obtains events from the database of the Continent hardware and software encryption system. To use it, you must configure the settings of the corresponding secret type.

[OOTB] InfoWatch Traffic Monitor SQL	Obtains events from the database of the InfoWatch Traffic Monitor system. To use it, you must configure the settings of the corresponding secret type.
[OOTB] KSC MSSQL	Obtains events from the MS SQL database of the Kaspersky Security Center system. To use it, you must configure the settings of the corresponding secret type.
[OOTB] KSC MySQL	Obtains events from the MySQL database of the Kaspersky Security Center system. To use it, you must configure the settings of the corresponding secret type.
[OOTB] KSC PostgreSQL	Obtains events from the PostgreSQL database of the Kaspersky Security Center 15.0 system. To use it, you must configure the settings of the corresponding secret type.
[OOTB] Oracle Audit Trail SQL	Obtains audit events from the Oracle database. To use it, you must configure the settings of the corresponding secret type.
[OOTB] SecretNet SQL	Obtains events from the SecretNet SQL database. To use it, you must configure the settings of the corresponding secret type.

Secrets

Secrets are used to securely store sensitive information such as user names and passwords that must be used by LogAnalyzer to interact with external services. If a secret stores account data such as user login and password, when the collector connects to the event source, the account specified in the secret may be blocked in accordance with the password policy configured in the event source system.

Secrets can be used in the following LogAnalyzer services and features:

- Collector (when using TLS encryption).
- Connector (when using TLS encryption).
- Destinations (when using TLS encryption or authorization).
- Proxy servers.

Available settings:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—the type of secret.

  When you select the type in the drop-down list, the parameters for configuring this secret type also appear. These parameters are described below.
- **Description**—up to 4,000 Unicode characters.

Depending on the secret type, different fields are available. You can select one of the following secret types:

- **credentials**—this type of secret is used to store account credentials required to connect to external services, such as SMTP servers. If you select this type of secret, you must fill in the **User** and **Password** fields. If the **Secret** resource uses the 'credentials' type to connect the collector to an event source, for example, a database management system, the account specified in the secret may be blocked in accordance with the password policy configured in the event source system.
- **token**—this secret type is used to store tokens for API requests. Tokens are used when connecting to IRP systems, for example. If you select this type of secret, you must fill in the **Token** field.

- The **ktl** secret type (available only in the commercial version of LogAnalyzer) is used to store Kaspersky Threat Intelligence Portal user account credentials. If you select this type of secret, you must fill in the following fields:

  - **User** and **Password** (required fields)—user name and password of your Kaspersky Threat Intelligence Portal account.

  - **PFX file** (required)—lets you upload a Kaspersky Threat Intelligence Portal certificate key.

  - **PFX password** (required)—the password for accessing the Kaspersky Threat Intelligence Portal certificate key.

- **urls**—this secret type is used to store URLs for connecting to SQL databases and proxy servers. In the **Description** field, you must provide a description of the connection for which you are using the secret of **urls** type.

  You can specify URLs in the following formats: hostname:port, IPv4:port, IPv6:port, :port.

- **pfx**—this type of secret is used for importing a PFX file containing certificates. If you select this type of secret, you must fill in the following fields:

  - **PFX file** (required)—this is used to upload a PFX file. The file must contain a certificate and key. PFX files may include CA-signed certificates for server certificate verification.

    > The certificate included in the PFX file cannot be encrypted with the AES-128-CBC algorithm. LogAnalyzer does not support such certificates.

  - **PFX password** (required)—this is used to enter the password for accessing the certificate key.

- **kata/edr**—this type of secret is used to store the certificate file and private key required when connecting to the Kaspersky Endpoint Detection and Response server. If you select this type of secret, you must upload the following files:

  - Certificate file—LogAnalyzer server certificate.

    The file must be in PEM format. You can upload only one certificate file.

  - Private key for encrypting the connection—LogAnalyzer server RSA key.

    The key must be without a password and with the PRIVATE KEY header. You can upload only one key file.

    > You can generate certificate and key files by clicking the ⭳ button.

- **snmpV1**—this type of secret is used to store the value of **Community** (for example, `public` or `private`) that is required for interaction over the Simple Network Management Protocol.

- **snmpV3**—this type of secret is used for storing data required for interaction over the Simple Network Management Protocol. If you select this type of secret, you must fill in the following fields:
  - **User**—user name indicated without a domain.
  - **Security Level**—security level of the user.
    - **NoAuthNoPriv**—messages are forwarded without authentication and without ensuring confidentiality.
    - **AuthNoPriv**—messages are forwarded with authentication but without ensuring confidentiality.
    - **AuthPriv**—messages are forwarded with authentication and ensured confidentiality.

    Additional settings depending on the selected level may be displayed.
  - **Password**—SNMP user authentication password. This field becomes available when the **AuthNoPriv** or **AuthPriv** security level is selected.
  - **Authentication Protocol**—the following protocols are available: MD5, SHA, SHA224, SHA256, SHA384, SHA512. This field becomes available when the **AuthNoPriv** or **AuthPriv** security level is selected.
  - **Privacy Protocol**—protocol used for encrypting messages. Available protocols: DES, AES. This field becomes available when the **AuthPriv** security level is selected.
  - **Privacy password**—encryption password that was set when the SNMP user was created. This field becomes available when the **AuthPriv** security level is selected.
- **certificate**—this secret type is used for storing certificate files. Files are uploaded to a resource by clicking the **Upload certificate file** button. X.509 certificate public keys in Base64 are supported.
- **fingerprint**—this type of secret is used to store the **Elastic fingerprint** value that can be used when connecting to the Elasticsearch server.
- **PublicPKI**—this type of secret is used to connect a LogAnalyzer collector to ClickHouse. If you select this option, you must specify the secret containing the base64-encoded PEM private key and the public key.

Predefined secrets

The secrets listed in the table below are included in the LogAnalyzer distribution kit.

Predefined secrets

Secret name	Description
[OOTB] Continent SQL connection	Stores confidential data and settings for connecting to the APKSh Kontinent database. To use it, you must specify the login name and password of the database.
[OOTB] KSC MSSQL connection	Stores confidential data and settings for connecting to the MS SQL database of Kaspersky Security Center (KSC). To use it, you must specify the login name and password of the database.
[OOTB] KSC MySQL Connection	Stores confidential data and settings for connecting to the MySQL database of Kaspersky Security Center (KSC). To use it, you must specify the login name and password of the database.
[OOTB] Oracle Audit Trail SQL Connection	Stores confidential data and settings for connecting to the Oracle database. To use it, you must specify the login name and password of the database.
[OOTB] SecretNet SQL connection	Stores confidential data and settings for connecting to the MS SQL database of the SecretNet system. To use it, you must specify the login name and password of the database.

Analytics

LogAnalyzer provides extensive analytics on the data available to the application from the following sources:

- Events in storage
- Assets
- Data from collectors on the number of processed events
- Metrics

You can configure and receive analytics in the **Dashboard**, **Reports**, and **Source status** sections of the LogAnalyzer web interface. Analytics are built by using only the data from tenants that the user can access.

The date format depends on the localization language selected in the application settings. Possible date format options:

- English localization: YYYY-MM-DD.
- Russian localization: DD.MM.YYYY.

Working with events

In the **Events** section of the LogAnalyzer web interface, you can view events received by the application. The events table displays the data received after the SQL query is executed.

The event date format depends on the localization language selected in the application settings. Possible date format options:

- English localization: YYYY-MM-DD.
- Russian localization: DD.MM.YYYY.

Filtering and searching events

The **Events** section of the LogAnalyzer web interface does not show any data by default. To view events, you need to define an SQL query in the search field and click the **Run query** button. The SQL query can be entered manually or it can be generated using a query builder.

Data aggregation and grouping is supported in SQL queries.

You can search for events across multiple storages. For example, you can find events to determine where a user account is being blocked or which IP addresses were used to log in to which URLs. Example query for finding a blocked user account:

```
SELECT * FROM `events` WHERE DestinationUserName = 'username' AND DeviceEventClassID =
'4625' LIMIT 250
```

To search events across multiple repositories, in the **Events** section, in the drop-down list at the top right of the section, select the check boxes next to the repositories you want.

The list displays the following storages:

- Storages of the Main tenant.
- Available storages of tenants that satisfy one of the following conditions:
  - The tenant that owns the storage is enabled in the tenant filter and the user has permissions to read events in this tenant.
  - The user has access to the tenant of one of the partitions of the storage and has permissions to read events in this tenant.

    For example, if you have access to the collector tenant, but do not have access to the storage tenant, by default, the inaccessible tenant's storage is not displayed in the list of available storages. If a destination in the storage of a tenant that is not available to you is added to the collector of an available tenant, after an event arrives in the partition of the tenant of the collector, the storage of the unavailable tenant appears in the list of storages in the **Events** section.

The drop-down list of storages in the upper-right part of the **Events** section displays the name of the first of the selected storages and the number of selected storages, if there are several. You can hover over the drop-down list to display all of the selected storages. If a space in a storage is deleted, the storage drop-down list shows "deleted<ID of deleted space>", and the events remain searchable for the duration of the TTL.

The tenants selected in the tenant filter affect which storages are displayed in the drop-down list of storages. If you disable tenants whose storages are available to you in the tenant filter, these storages are no longer displayed in the drop-down list of storages. If these storages had been selected in the drop-down list of storages, their check boxes are cleared and events from these storages are not displayed. If only one storage is selected in the drop-down list of storages that is not from the Main tenant, and if in tenant selection you disabled the tenant that owns the selected storage, this storage is not displayed in the list of storages and LogAnalyzer automatically changes the selection to one of the storages of the Main tenant.

A simple query to all selected storages is allowed, as in the example above. If at least one of the selected storages is not available for the query, LogAnalyzer returns an error.

Limitations for searching events across multiple storages:

- When querying multiple storages, export to TSV, retroscan, or REST API requests are not available.
- A SELECT can contain only * and/or names of event fields. Aliases, functions, expressions are not allowed.
- An ORDER BY clause must also contain only event fields (no functions, constants, expressions, and so on). If a field is not present in the list of fields for the SELECT, such a field is automatically added when sending to a specific cluster. You cannot set an ORDER BY ClusterID.
- GROUP BY is not available.

Complex queries with grouping and aggregation are allowed for a single selected storage.

You can add filter conditions to an already generated SQL query in the window for viewing statistics, the events table, and the event details area:

- Changing a query from the Statistics window

  *To change the filtering settings in the **Statistics** window:*

  1. Open **Statistics** details area by using one of the following methods:

     - In the ••• drop-down list in the top right corner of the events table select **Statistics**.
     - In the events table click any value and in the opened context menu select **Statistics**.

     The **Statistics** details area appears in the right part of the web interface window.

  2. Open the drop-down list of the relevant parameter and hover your mouse cursor over the necessary value.

  3. Use the plus and minus signs to change the filter settings by doing one of the following:

     - If you want the events selection to include only events with the selected value, click the + icon.
     - If you want the events selection to exclude all events with the selected value, click the – icon.

     As a result, the filter settings and the events table will be updated, and the new search query will be displayed in the upper part of the screen.

- Changing a query from the events table

  To change the filtering settings in the events table:

  1. In the **Events** section of the LogAnalyzer web interface, click any event parameter value in the events table.

  2. In the opened menu, select one of the following options:

     - If you want the table to show only events with the selected value, select **Filter by this value**.
     - If you want to exclude all events with the selected value from the table, select **Exclude from filter**.

     As a result, the filter settings and the events table are updated, and the new search query is displayed in the upper part of the screen.

- Changing a query from the Event details area

To change the filter settings in the event details area:

1. In the LogAnalyzer web interface, in the **Events** section, click the relevant event.

   The **Event details** area appears in the right part of the window.

2. Change the filtering settings by clicking the plus or minus icons:

   - If you want to apply the filter instantly and update the events table, click + or — next to the parameters that you need.
   - If you want to continue to change filtering parameters, hold down the **Ctrl** key and click + or —. This adds the condition to the body of the query, but does not automatically run the query.

     When adding a condition to the query, an expression for the selected parameter is added to the `WHERE` clause. When adding conditions for multiple parameters, the expressions are joined with the `AND` operator. A field to be filtered by is added to the `SELECT` clause only if the `*` character is not used in the query.

     Example:
     Original query:
     ```
     SELECT *
     FROM `events`
     ORDER BY Timestamp DESC
     LIMIT 250
     ```
     All events are selected without filters, the last 250 events are displayed in descending order by the `Timestamp` field.
     When adding new conditions by **Ctrl**-clicking the + icon, the query looks like this:
     ```
     SELECT *
     FROM `events`
     WHERE DeviceProduct = 'Windows' AND StartTime = 1755767943436
     ORDER BY Timestamp DESC
     LIMIT 250
     ```
     Conditions are added to the `WHERE` clause with the `AND` operator. Fields are not added to the `SELECT` clause because the `*` character is used in this clause. The query is not executed automatically.

3. After adding all the necessary filtering conditions, run the query in one of the following ways:

   - Add the last condition without holding down the **Ctrl** key. The query is run automatically.
   - Close the event details area and run the query manually by clicking **Run query** button or pressing **Ctrl+Enter**.

The filter settings and the events table are updated, and the modified search query is displayed in the upper part of the screen.

After modifying a query, all query parameters, including the added filter conditions, are transferred to the query builder and the search field.

When you switch to the query builder, the parameters of a query entered manually in the search field are not transferred to the builder, so you will need to create your query again. Also, the query created in the builder does not overwrite the query that was entered into the search string until you click the **Apply query** button in the builder window.

In the SQL query input field, you can enable the display of control characters.

You can also filter events by time period. Search results can be automatically updated.

The filter configuration can be saved. Existing filter configurations can be deleted.

Filter functions are available for users regardless of their roles.

When accessing certain event fields with IDs, LogAnalyzer returns the corresponding names.

For more details on SQL, refer to the ClickHouse documentation. For SQL operators and functions supported in LogAnalyzer, see also the LogAnalyzer operator usage and supported functions.

Selecting Storage

Events that are displayed in the **Events** section of the LogAnalyzer web interface are retrieved from storage (from the ClickHouse cluster). Depending on the demands of your company, you may have more than one Storage. However, you can only receive events from one Storage at a time, so you must specify which one you want to use.

To select the Storage you want to receive events from,

In the **Events** section of the LogAnalyzer web interface, open the drop-down list of storages in the upper-right part of the section and select one or more spaces of one or more storages.

The list displays the following spaces:

- Spaces added when the storage service was created.
- The LogAnalyzer Default space
- The LogAnalyzer Audit space, if the Main tenant is selected and if there is at least one audit event in this space.

LogAnalyzer displays only those spaces to which the user has access permissions.

Events from the selected storage are displayed in the table of events. Names of the selected storages are displayed in the drop-down list.

If you have access rights to the space that stores events of a tenant to which you do not have access permissions, such a space is displayed in the list, but is not searchable.

Generating an SQL query using a builder

In LogAnalyzer, you can use a query builder to generate an SQL query for filtering events.

To generate an SQL query using a builder:

1. In the **Events** section of the LogAnalyzer web interface, click the button.

   The filter constructor window opens.

2. Generate a search query by providing data in the following parameter blocks:

   SELECT—event fields that should be returned. The * value is selected by default, which means that all available event fields must be returned. To make viewing the search results easier, select the necessary fields in the drop-down list. In this case, the data only for the selected fields is displayed in the table. Note that Select * increases the duration of the request execution, but eliminates the need to manually indicate the fields in the request.

   When selecting an event field, you can use the field on the right of the drop-down list to specify an alias for the column of displayed data, and you can use the right-most drop-down list to select the operation to perform on the data: **count**, **max**, **min**, **avg**, **sum**.

   If you use aggregation functions in an SQL query, you can customize the display of columns in the event table. After executing the query without *, the columns will match the query.

   If the query uses grouping or aggregation functions, instead of one number that was received as a response to the `SELECT max(BytesOut) FROM `events` LIMIT 250` query, the radar shows a distribution of events from which this number is obtained.

   - **FROM**—data source. Select the **events** value.

   - **WHERE**—conditions for filtering events.

     Conditions and groups of conditions can be added by using the **Add condition** and **Add group** buttons. The **AND** operator value is selected by default in the group of conditions, but you can change the operator by clicking it. Available values: **AND**, **OR**, **NOT**. The structure of conditions and condition groups can be changed by using the icon to drag and drop expressions.

     Adding filter conditions:

     a. In the drop-down list on the left, select the event field that you want to use for filtering.

     b. Select the necessary operator from the middle drop-down list. The available operators depend on the type of value of the selected event field.

     c. Enter the value of the condition. Depending on the selected type of field, you may have to manually enter the value, select it from the drop-down list, or select it on the calendar.

     You can delete filter conditions and filter condition groups by clicking .

   - **GROUP BY**—event fields or aliases to be used for grouping the returned data.

     If you use data grouping in an SQL query, custom fields specified in the query are displayed in the list of events. Table columns in a group cannot be customized. In group events, you can customize the display of columns.

   - **ORDER BY**—columns used as the basis for sorting the returned data. In the drop-down list on the right, you can select the necessary order: **DESC**—descending, **ASC**—ascending.

   - **LIMIT**—number of strings displayed in the table.

     The default value is 250.

     If you are filtering events by user-defined period and the number of strings in the search results exceeds the defined value, you can click the **Show next** button to display additional strings in the table. This button is not displayed when filtering events by the standard period.

3. Click **Apply query**.

   The current SQL query will be overwritten. The generated SQL query is displayed in the search field.

   If you want to reset the builder settings, click the **Default query** button.

   If you want to close the builder without overwriting the existing query, click the × button in the upper-right part of the query creation window.

4. Click the **Run query** button to display the data in the table.

The table will display the search results based on the generated SQL query.

When switching to another section of the web interface, the query generated in the builder is not preserved. If you return to the **Events** section from another section, the builder will display the default query.

For more details on SQL, refer to the ClickHouse documentation. See also LogAnalyzer operator usage and supported functions.

Manually creating an SQL query

You can use the search field manually to create SQL queries of any complexity to filter events.

To manually generate an SQL query:

1. Go to the **Events** section of the LogAnalyzer web interface.

   An input form opens.

2. Enter your SQL query into the input field. You must use single quotes in your queries.

3. Click **Run query** to run the query.

A table of events that satisfy the criteria of your query will be displayed. If necessary, you can filter events by period.

Supported functions and operators

Function	Description
`SELECT`	Event fields that you want to be returned. The following functions and operators are supported: • Aggregation functions: `count`, `avg`, `max`, `min`, `sum`. • Arithmetic operators: `+`, `-`, `*`, `/`, `<`, `>`, `=`, `!=`, `>=`, `<=`. You can combine functions and operators in an SQL query. If you use aggregation functions in an SQL query, you can customize the display of columns in the event table. After executing the query without *, the columns will match the query. If the query uses grouping or aggregation functions, instead of one number that was received as a response to the `SELECT max(BytesOut) FROM `events` LIMIT 250` query, the radar shows a distribution of events from which this number is obtained.
`FROM`	Data source. When creating an SQL query, you need to specify the events value as the data source.

Function	Description
`WHERE`	Conditions for filtering events: • `AND`, `OR`, `NOT`, `=`, `!=`, `>`, `>=`, `<`, `<=` • `IN` • `BETWEEN` • `LIKE` • `ILIKE` • `inSubnet` • `match` (the re2 syntax of regular expressions is used in SQL queries) You must additionally escape special characters with the backslash `\`.
`GROUP BY`	Event fields or aliases to be used for grouping the returned data. If you use data grouping in an SQL query, custom fields specified in the query are displayed in the list of events. Table columns in a group cannot be customized. In group events, you can customize the display of columns.
`ORDER BY`	Columns by which you want to sort the returned data. Possible values: • `DESC` to sort in descending order • `ASC` to sort in ascending order
`OFFSET`	The number of rows to skip before displaying the results of the SQL query.
`LIMIT`	The number of rows that can be displayed in the table. The default value is `250`. If you are filtering events by user-defined period and the number of rows in the search results exceeds the defined value, you can click the **Show next** button to display additional rows in the table. This button is not displayed when filtering events by the standard period.

Examples of SQL queries

- `SELECT * FROM `events` WHERE Type IN ('Base', 'Audit') ORDER BY Timestamp DESC LIMIT 250`
  In the events table, all events with the **Base** and **Audit** type are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.
- `SELECT * FROM `events` WHERE BytesIn BETWEEN 1000 AND 2000 ORDER BY Timestamp ASC LIMIT 250`
  All events of the events table for which the **BytesIn** field contains a value of received traffic in the range from 1,000 to 2,000 bytes are sorted by the **Timestamp** column in ascending order. The number of strings that can be displayed in the table is 250.
- `SELECT * FROM `events` WHERE Message LIKE '%ssh:%' ORDER BY Timestamp DESC LIMIT 250`
  In the events table, all events whose **Message** field contains data corresponding to the defined `%ssh:%` template in lowercase are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.
- `SELECT * FROM `events` WHERE inSubnet(DeviceAddress, '00.0.0.0/00') ORDER BY Timestamp DESC LIMIT 250`
  In the events table, all events for the hosts that are in the 00.0.0.0/00 subnet are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.
- `SELECT * FROM `events` WHERE match(Message, 'ssh.*') ORDER BY Timestamp DESC LIMIT 250`
  In the events table, all events whose **Message** field contains text corresponding to the `ssh.*` template are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.
- `SELECT max(BytesOut) / 1024 FROM `events``
  Maximum amount of outbound traffic (KB) for the selected time period.
- `SELECT count(ID) AS "Count", SourcePort AS "Port" FROM `events` GROUP BY SourcePort ORDER BY Port ASC LIMIT 250`
  Number of events and port number. Events are grouped by port number and sorted by the **Port** column in ascending order. The number of strings that can be displayed in the table is 250.
  The **ID** column in the events table is named Count, and the **SourcePort** column is named Port.

If you want to use a special character in a query, you need to escape this character by placing a backslash (\) character in front of it.

Example:

```
SELECT * FROM `events` WHERE match(Message, 'ssh:\'connection.*') ORDER BY Timestamp DESC LIMIT 250
```

In the events table, all events whose **Message** field contains text corresponding to the `ssh: 'connection'` template are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.

When creating a normalizer for events, you can choose whether to retain the field values of the raw event. The data is stored in the **Extra** event field. This field is searched for events by using the LIKE operator.

Example:
```
SELECT * FROM `events` WHERE DeviceAddress = '00.00.00.000' AND Extra LIKE '%"app":"example"%' ORDER BY Timestamp DESC LIMIT 250
```
In the events table, all events for hosts with the IP address 00.00.00.000 where the example process is running are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.

If you created an SQL query manually in the search field and then switched to the builder, the SQL query parameters are not transferred to the builder. In this case, you will need to re-create the SQL query in the builder. The SQL query created in the builder does not overwrite the SQL query that was entered into the search string until you click the **Apply query** button in the builder window. If you created an SQL query in the query builder and then switched to the search field, the query parameters are transferred automatically.

Aliases must not contain spaces.

For more details on SQL, refer to the ClickHouse documentation. See also the supported ClickHouse functions.

Saving query history

In the LogAnalyzer web interface, the **Events** section keeps a personal history of SQL search queries. You can refer to the history of queries and quickly find a query you have used in the past.

The queries are saved in the browser. Changing your browser resets the query history. The history stores the last 32 of your queries. Queries are displayed with formatting. The date and time when the query was executed is displayed.

Only queries with correct syntax are saved in the query history.

To find a search query in history:

1. In the **Events** section, place the cursor in the query bar.
2. Click the (**SQL history**) button in the toolbar of the query window or press **Ctrl+]** or **Ctrl+[**.

   This opens the query history window. The last query you ran is displayed at the top of the list of queries.
3. Select a search query. To navigate the list of saved queries, press **Ctrl+]** or **Ctrl+[**.

   The selected query is pasted to the search bar.
4. If necessary, run the search query. To do so, press **Ctrll+Enter**.

   When you run a query, the query history window is closed.

You can also close the query history window by pressing the **Esc** key; this moves the cursor to the query editing window.

You can copy the query to the clipboard by pressing **Ctrl+C**. This copies the part of the query that is highlighted with the mouse, or the entire query string, limited by the box, if there was no selection.

You can save the query by pressing **Ctrl+S**, or by clicking the button (**Save query**) in the row. This closes the history window and opens the sidebar for saving the query with populated fields.

You can paste a query into the query window without running it by clicking the (**Insert query**) button, or by pressing **Enter**.

Managing saved search queries

In LogAnalyzer, in the **Events** section, you can organize saved SQL queries in a tree of folders for structured storage and quick search of SQL queries. You can edit previously saved queries, rename them, arrange queries in groups (folders) of accessible tenants, and search for previously saved queries in the search bar.

Saved queries follow the tenant access model and are visible to all users that have access to the corresponding tenants.

Saved queries are placed in the **Saved queries pane.** You can open or close the **Saved queries** pane using the (bookmark) button on the toolbar of the query window.

The **Saved queries** navigation pane on the left contains the following:

- The query search window where you can search by folder name and query name.
- Tree of folders and queries, the **Favorites** folder.
- The query filter that displays the queries you have created (on the **My** tab) and all queries (on the **All** tab)

Displaying a query in the query window

To display a query in the query window:

1. In the **Events** section, click the (bookmark) button.

   This opens the **Saved queries** navigation pane.

2. In the tree of folders in the **Saved queries** pane, click to select the relevant query.

The text of the selected query is displayed in the query window.

Saving a query to a folder or a tenant

In the **Events** section, after entering a query in the query window and clicking the **Save current query** button (floppy disk icon), you can save the SQL query to one of the folders or tenants that you can select.

To save a query to a folder:

1. In the **New search** window, in the **Filter name** field, enter a name for the filter.

   Consider the following when naming the filter:

   - The name can contain Unicode characters.
   - The minimum length of a name is 1 character, and the maximum length is 128 characters.
   - Names must be unique within a tenant.
   - Tab characters, new line characters, and paragraph characters are replaced with space characters.

   When saving the name:

   - Leading and trailing spaces are removed.
   - Multiple spaces between characters are replaced with a single space character.

2. In the **Query** field, enter the text of the SQL query.
3. Select a tenant, a folder from the list of folders created in the tenant that is available to you, or create a new folder by clicking the **Add folder** button.

   You can also add a folder in the **Saved queries** window: click the + button next to the tenant to open the **New folder** window and add the folder in that window.

4. Click the **Save** button.

The query is saved in the selected folder or tenant.

Viewing the text of a query

To view the text of a query:

1. In the **Events** section, click the (bookmark) button.

   This opens the **Saved queries** navigation pane.

2. In the tree of folders in the **Saved queries** pane, click the button next to the relevant query.

This opens a window with the text of the query.

Setting the default query

You can set your default query that the **Events** section displays when opened.

To set the default query:

1. In the **Events** section, click the (bookmark) button.
   This opens the **Saved queries** navigation pane.
2. In the tree of folders on the **Saved queries** pane, select the relevant query.
3. Click the button and in the displayed menu, select **Use by default**.

The selected query becomes the default query.

Adding folders and queries to favorites

To add a folder or individual query to favorites:

1. In the **Events** section, click the (bookmark) button.
   This opens the **Saved queries** navigation pane.
2. In the tree of folders in the **Saved queries** pane, select the relevant folder or query.
3. Click the button and in the displayed menu, select **To favorites**.

The selected folder or resource is added to the **Favorites** folder.

Editing a query

To edit a query:

1. In the **Events** section, click the (bookmark) button.
   This opens the **Saved queries** navigation pane.
2. In the tree of folders on the **Saved queries** pane, select the relevant query.
3. Click the button and in the displayed menu, select **Edit**.
   This opens the window for editing the saved query.
4. In the window, edit the query name or text, or the folder in which you want to save the query.
5. Click the **Save** button.

The query is updated and saved in the selected folder.

Deleting a query

To delete a query from a folder (if you have sufficient rights):

1. In the **Events** section, click the bookmark () icon.
   This opens the **Saved queries** navigation pane.
2. In the tree of folders on the **Saved queries** pane, select the relevant query.

3. Click the three-dot icon (⋮) and in the menu that appears, select **Delete**.

4. Confirm deletion in the displayed window.

   The query is deleted.

Filtering events by period

In LogAnalyzer, you can specify the time period to display events from.

To filter events by period:

1. In the **Events** section of the LogAnalyzer web interface in the upper part of the window, open the time period drop-down list in the top right part of the section.

2. Select a period in one of the following ways:

   - If you want to specify an exact date, in the calendar on the left, select the start and end date of the period and click **Apply**.

     You can select a date up to and including the current date. The date and time format depends on your browser settings. If the **Range from** or **Range to** field has a value and you have not edited the time value manually, when you select a date in the calendar, the **Range from** field is automatically populated with 00:00:00.000, and the **Date to** field with 23:59:59.999. If you have manually deleted the value in the **Range from** or **Range to** field, when you select a date in the calendar, the field is automatically populated with the current time. After you select a value in one of the fields, the focus switches to the other field. If your **Range to** is earlier than your **Range from**, this earlier value is automatically inserted into the **Range from** field.

   - If you want to specify a relative period, select one of the available periods in the **Relative period** list on the right.

     The period is calculated relative to the current time.

   - If you want to specify a custom period, edit the value of the **Range from** and **Range to** fields.

     You can enter an exact date and time in the **DD.MM.YYYY HH:mm:ss.SSS** format for the Russian localization and **YYYY-MM-DD HH:mm:ss.SSS** for the English localization or a period relative to the current time as a formula. You can also combine these methods if necessary. If you do not specify milliseconds when entering the exact date, 000 is substituted automatically. If you have edited the time in the **Range from** or **Range to** fields, picking a date in the calendar does not change the time component.

     In the relative period formulas, you can use the **now** parameter for the current date and time and the interval parameterization language: **+** (only in the **Range to** field), **-**, **/** (rounding to the nearest), as well as time units: **y** (year), **M** (month), **w** (week), **d** (day), **h** (hour), **m** (minute), **s** (second). For example, you can specify the period **now-5d** to get data for the last five days, or **now/w** to get data from the beginning of the first day of the current week (00:00:00:000 UTC) to the current time (**now**).

     > The **Range from** field is required, and its value cannot exceed the value of the **Range to** field, and also cannot be earlier than 1970-01-01 (if specifying an exact date or a relative period). The **Range to** cannot be earlier than the **Range from**. If you do not specify a value in the **Range to** from field, **now** is specified automatically.

The default setting is **5 minutes** (**now-5m**). The bounds of the period are inclusive: for example, for the **Today** time range, events are displayed from the beginning (00:00:00:000 UTC) of the current day to the current time (**now**) inclusive, and for the **Yesterday** period, events are displayed from the beginning (00:00:00:000 UTC) of the previous day to 00:00:00:000 UTC of the current day.

LogAnalyzer stores time values in UTC, but in the user interface time is converted to the time zone of your browser. This is relevant to the relative periods: **Today**, **Yesterday**, **This week**, and **This month**. For example, if the time zone in your browser is UTC+3, and you select the **Today** period, events are displayed for the period from 03:00:00.000 until now, not from 00:00:00.000 until now.

If, when selecting a relative period, such as **Today**, **Yesterday**, **This week**, or **This month**, you want to take your time zone into account when displaying events, you need to manually add a time offset in the **Range from** and **Range to** fields (if a value other than **now** is specified) by adding or subtracting the correct number of hours. For example, if your browser's time zone is UTC+3 and you want to display data for **Yesterday**, you need to change **Range from** to **now-1d/d-3h** and **Date to** to **now/d-3h**. If you want to display data for the **Today** period, you only need to change the value in the **Range from** field to **now/d-3h**.

If you need results up to 23:59:59:999 of yesterday, you can use an SQL query with a filter by Timestamp or specify an exact date and time.

3. If you want to capture the last relative period for which a query was executed as an exact date and time, click the button.

   The relative period is then converted to an exact date and time. Hovering over this button displays the date and time that will be captured. Relative periods (**Today**, **Yesterday**, **This week**, **This month**) are converted to an exact UTC date and time, other periods are converted to a date and time that takes into account the time zone of the browser.

   If you have executed an SQL query with event grouping, when switching between groups in the query results, the relative time range is automatically converted to an exact date and time.

4. Click **Run query** to run the query.

   Only events registered during the specified time period are displayed. The period will be displayed in the upper part of the window.

You can also configure the display of events by using the events histogram that is displayed when you click the button in the upper part of the **Events** section. Events are displayed if you click the relevant data column or select the relevant time period and click the **Show events** button.

Grouping events

After getting a list of events, you often need to split the events into groups to localize an information security event. LogAnalyzer can group events in a list by one or more fields.

To group events, you no longer need to manually edit the text of the query; instead, you can click a column heading in the **Events** section and select **Add "Group by" in query** in the context menu. You can select a sequence of multiple fields to group by, and the fields will be automatically added to the query string. Having selected your fields, click **Run query**. As a result, events are grouped by the specified fields. Found groups are displayed in the **Groups** section. They can be displayed as a table and as cards. You can toggle between the display modes.

You can exclude a group from the query:

- In **Cards** mode, click the **-** button.
- In **Tables** mode, right-click the group and in the context menu, select **Exclude group from filter**.

As a result, the query is automatically modified and the group is excluded from the query.

If you want to go back to the original query, click **Return to initial select query**.

You can navigate through the groups and view the contents of each group.

You can do a global search in all groups or a local search in events within a selected group.

You can use more complex grouping by adding one or more fields.

You can remove a group from the grouping and in this way, go back one step.

If the grouping query returns many events, only the first 1000 events are displayed. If the query contains SELECT Count(ID), you can click the link with the total number of events in the query result to view all events. If the request does not contain Count(ID), the number of events in the group is not indicated, but you still can click the link and view the total number of events in the group.

Statistics, retrospective check by group, and export to TSV are available.

If you want the grouping result to be independent of time (because events arrive continuously), you can set a fixed relative interval and apply it as an absolute interval so that the events of interest do not drop out of the selection. To fix a relative interval, in the **Events** section, in the time interval drop-down list, select **Apply current range**. You can now manage groups within this query.

If you want to arrange the selected events by months, days, minutes, and seconds, you can group events by the Timestamp field. To group events, select a grouping option in the context menu of the Timestamp field in the event table.

If you want to normalize the value of the Timestamp field and display the time values from different sources in the same UTC time scale, select **Convert to UTC timezone** in the context menu of the Timestamp field in the events table.

Displaying names instead of IDs

When accessing certain event fields with IDs, LogAnalyzer returns the corresponding names rather than IDs. This helps make the information more readable. For example, if you access the `TenantID` event field (which stores the tenant ID), you get the value of the `TenantName` event field (which stores the tenant name).

When exporting events, values of both fields are written to the file, the ID as well as the name.

The table below lists the fields that are substituted when accessed:

Requested field	Returned field
`TenantID`	`TenantName`
`ServiceID`	`ServiceName`
`DeviceAssetID`	`DeviceAssetName`
`SourceAssetID`	`SourceAssetName`
`DestinationAssetID`	`DestinationAssetName`
`SourceAccountID`	`SourceAccountName`
`DestinationAccountID`	`DestinationAccountName`

Substitution does not occur if an alias is assigned to the field in the SQL query. Examples:

- `SELECT TenantID FROM `events` LIMIT 250` — in the search result, the name of the tenant is displayed in the TenantID field.
- `SELECT TenantID AS Tenant_name FROM `events` LIMIT 250` — in the search result, the tenant ID will be displayed in the Tenant_name field.

Presets

You can use presets to simplify work with queries if you regularly view data for a specific set of event fields. In the line with the SQL query, you can type `Select *` and select a saved preset; in that case, the output is limited only to the fields specified in the preset. This method slows down performance but eliminates the need to write a query manually every time.
Presets are saved on the KUMA Core server and are available to all KUMA users of the specified tenant.

To create a preset:

1. In the **Events** section, click the icon.
2. In the window that opens, on the **Event field columns** tab, select the required fields.

   To simplify your search, you can start typing the field name in the **Search** area.
3. To save the selected fields, click **Save current preset**.

   The **New preset** window opens.
4. In that window, specify the **Name** of the preset, and in the drop-down list, select the **Tenant**.
5. Click **Save**.

   The preset is created and saved.

To apply a preset:

1. In the query entry field, enter **Select ***.
2. In the **Events** section of the KUMA web interface, click the icon.
3. In the opened window, use the **Presets** tab to select the relevant preset and click the button.

   The fields from the selected preset are added to the SQL query field, and the columns are added to the table. No changes are made in Builder.
4. Click **Run query** to run the query.

   After the query execution completes, the columns are filled in.

Saving and selecting events filter configuration

In LogAnalyzer, you can save a filter configuration and use it in the future. Other users can also use the saved filters if they have the appropriate access rights. When saving a filter, you are saving the configured settings of all the active filters at the same time, including the time-based filter, query builder, and the events table settings. Search queries are saved on the LogAnalyzer Core server and are available to all LogAnalyzer users of the selected tenant.

To save the current settings of the filter, query, and period:

1. In the **Events** section of the LogAnalyzer web interface, click the icon next to the filter expression.

   This opens the **New filter** window.
2. In the window that opens, enter the name of the filter configuration in the **Name** field. The name can contain up to 128 Unicode characters.

3. Under **Select folder**, select the tenant to which you want the created filter to belong and the folder inside the tenant, if necessary.

   To simplify your search, you can start typing the name in the **Search** area.

4. If necessary, click **Add folder** to create a new folder inside the selected tenant.

5. Click **Save**.

The filter configuration is now saved.

To select a previously saved filter configuration:

In the **Events** section of the LogAnalyzer web interface, click the ☆ icon next to the filter expression and select the relevant filter.

The selected configuration is active, which means that the search field is displaying the search query, and the upper part of the window is showing the configured settings for the period and frequency of updating the search results. Click the **Run query** button to submit the search query.

To add the filter configuration to favorites, you can hover over the name of the filter configuration, click the ⋮ icon next to the name of the configuration and click the **To favorites** button.

Deleting event filter configurations

To delete a previously saved filter configuration:

1. In the **Events** section of the LogAnalyzer web interface, click the ☆ icon next to the filter search query.

   This opens the **Saved queries** window.

2. Hover over the configuration that you want to delete and click the ⋮ icon next to its name.

3. This opens a menu; in the menu, click the 🗑 icon.

The filter configuration is now deleted for all LogAnalyzer users.

Supported ClickHouse functions

The following ClickHouse functions are supported in LogAnalyzer:

- Arithmetic functions.
- Arrays.
- Comparison functions.
- Logical functions.
- Type conversion functions.
- Date and time functions.
- String functions.

- String search functions.
- Conditional functions: only the regular 'if' operator; the ternary operator is not supported.
- Mathematical functions.
- Rounding functions.
- Functions for splitting and merging strings and arrays.
- Bit functions.
- Functions for working with UUIDs.
- Functions for working with URLs.
- Functions for working with IP addresses.
- Functions for working with Nullable arguments.
- Functions for working with geographic coordinates.

Functions from other sections are not supported.

For more details on SQL, refer to the ClickHouse documentation.

Viewing event detail areas

To view information about an event:

1. In the application web interface window, select the **Events** section.
2. Search for events by using the query builder or by entering a query in the search field.
   The event table is displayed.
3. Select the event whose information you want to view.

   The event details window opens.

The **Event details** area appears in the right part of the web interface window and contains a list of the event's parameters with values. In this area you can:

- Include the selected field in the search or exclude it from the search by clicking + or − next to the setting value.
- Click a file hash in the **FileHash** field ещ open a list in which you can select one of the following actions:
  - Show info from **Threat Lookup**.
    This is available when integrated with **Kaspersky Threat Intelligence Portal**.
  - Add to **Internal TI of CyberTrace**.
    This is available when integrated with **Kaspersky CyberTrace**.

- Open a window containing information about the asset if it is mentioned in the event fields and registered in the program.
- Click the link containing the collector name in the **Service** field to view the settings of the service that registered the event.
- You can view all non-empty event fields by clicking **Show all event fields**. The following conditions must be satisfied for the rule to trigger:
  - The query is executed for a set of fields and is not a SELECT *.
  - Not all non-empty fields are displayed. If all non-empty fields are displayed in the event card, the **View all event fields** button is not displayed.
  - The event card is opened in LogAnalyzer in the **Events** section.

  This does not change the original query or rebuild the table of events.

  Example

  Prerequisite:

  An event was sent with multiple non-empty fields, such as Message, SourceAddress, DeviceCustomString1.

  A query was executed for a set of fields.

  ```
  SELECT Type, Message

  FROM `events`

  ORDER BY Timestamp

  DESC LIMIT 250
  ```

  Clicking an event opens the **Event Information** window with the following information:

  - The Type and Message fields from the query.
  - The default TenantID and Timestamp fields not specified in the query are displayed.

  Non-empty SourceAddress and DeviceCustomString1 fields of the event that are missing from the query but are present in the sent event are not displayed.

  If you want to view all non-empty fields, click **View all event fields**.

  This displays all non-empty fields of the event: Type, Message, TenantID, Timestamp, SourceAddress, DeviceCustomString1.

- Change the display of the event by clicking the **Format** button. By default, the event is displayed as a one-line string in XML or JSON format. When you click this button, the event is displayed in multi-line format with indentation. Clicking the button again reverts the display to the one-line format. Formatting is applied to all events in the table in the **Events** section.

In the **Event details** area, the name of the described object is shown instead of its ID in the values of the following settings. At the same time, if you change the filtering of events by this setting (for example, by clicking — to exclude events with a certain setting-value combination from search results), the object's ID, and not its name, is added to the SQL query:

- TenantID
- ServiceID
- DeviceAssetID
- SourceAssetID
- DestinationAssetID
- SourceAccountID
- DestinationAccountID

Exporting events

In LogAnalyzer, you can export information about events to a TSV file. The selection of events that will be exported to a TSV file depends on filter settings. The information is exported from the columns that are currently displayed in the events table. The columns in the exported file are populated with the available data even if they did not display in the events table in the LogAnalyzer web interface due to the special features of the SQL query.

To export information about events:

1. In the **Events** section of the LogAnalyzer web interface, click the **TSV** button in the upper part of the table of events.

   The new export TSV file task is created in the **Task manager** section.

2. Find the task you created in the **Task manager** section.

   When the file is ready to be downloaded, the **Status** column of the task displays the **Completed** status and the icon.

3. Click the task type name and select **Upload** from the drop-down list.

   The TSV file will be downloaded using your browser's settings. By default, the file name is event-export-<date>_<time>.tsv.

The file is saved based on your web browser's settings.

Configuring the table of events

Responses to user SQL queries are presented as a table in the **Events** section. The fields selected in the custom query appear at the end of the table, after the default columns. This table can be updated.

The following columns are displayed in the events table by default:

- **TenantID**.
- **Timestamp**.

- **Name**.
- **DeviceProduct**.
- **DeviceVendor**.
- **DestinationAddress**.
- **DestinationUserName**.

In LogAnalyzer, you can customize the displayed set of event fields and their display order. The selected configuration can be saved.

When using SQL queries with data grouping and aggregation for filtering events, statistics are not available and the order of displayed columns depends on the specific SQL query.

In the events table, in the event details area and in the widgets, the names of assets, accounts, and services are displayed instead of the IDs as the values of the SourceAssetID, DestinationAssetID, DeviceAssetID, SourceAccountID, DestinationAccountID, and ServiceID fields. When exporting events to a file, the IDs are saved, but columns with names are added to the file. The IDs are also displayed when you point the mouse over the names of assets, accounts, or services.

Searching for fields with IDs is only possible using IDs.

To configure the fields displayed in the events table:

1. Click the ⚙ icon in the top right corner of the events table.

   A window for selecting the event fields that should be displayed in the events table will be displayed.

2. Select the check boxes opposite the fields that you want to view in the table. You can search for relevant fields by using the **Search** field.

   You can configure the table to display any event field from the LogAnalyzer event data model and the extended event schema. The **Timestamp** and **Name** parameters are always displayed in the table. Click the **Default** button to display only default event parameters in the events table.

   When you select a check box, the events table is updated and a new column is added. When a check box is cleared, the column disappears.

   You can also remove columns from the events table by clicking the column title and selecting **Hide column** from the drop-down list.

3. If necessary, change the display order of the columns by dragging the column headers in the event tables.

4. If you want to sort the events by a specific column, click its title and in the drop-down list select one of the available options: **Ascending** or **Descending**.

The selected event fields will be displayed as columns in the table of the **Events** section in the order you specified.

Refreshing events table

You can update the displayed event selection with the most recent entries by refreshing the web browser page. You can also refresh the events table automatically and set the frequency of updates. Automatic refresh is disabled by default.

To enable automatic refresh,

Select a refresh rate from the drop-down list in the upper-right part of the **Events** section:

- **5 seconds**
- **15 seconds**
- **30 seconds**
- **1 minute**
- **5 minutes**
- **15 minutes**

The events table now refreshes automatically.

To disable automatic refresh:

In the drop-down list of refresh rates in the upper-right part of the **Events** section, select **No refresh**.

Getting events table statistics

You can get statistics for the current events selection displayed in the events table. The selected events depend on the filter settings.

To obtain statistics:

Select **Statistics** from the ••• drop-down list in the upper-right corner of the events table, or click on any value in the events table and select **Statistics** from the opened context menu.

The **Statistics** details area appears with the list of parameters from the current event selection. The numbers near each parameter indicate the number of events with that parameter in the selection. If a parameter is expanded, you can also see its five most frequently occurring values. You can find relevant parameters by using the **Search fields** field.

In a high availability configuration, for all event fields that contain the FQDN of the Core, the **Statistics** section displays **core** instead of the FQDN.

The **Statistics** window allows you to modify the events filter.

When using SQL queries with data grouping and aggregation for filtering events, statistics are not available.

Generating an SQL query using LogAnalyzer SQL functions

LogAnalyzer's SQL functions allow using the attributes of assets and accounts in search queries to filter events, generate reports and widgets (graph type: **Table**). You can enrich events with data from dictionaries, tables, assets, and accounts using the following sets of functions:

- The enrich function set. Allows enriching query results with fields of the asset or account, values from a dictionary of the **Dictionary** type or a dictionary of the **Table** type. The `enrich` function set includes the following functions:
  - `enrich_assets`
  - `enrich_accounts`
  - `enrich_table`
  - `enrich_dictionary`
- The lookup function set. Allows adding conditions for assets and accounts to the search query. The `lookup` function set includes the following functions:
  - `lookup_assets`
  - `lookup_accounts`
  - `lookup_assets_category`

The maximum number of events per query is 10,000.

Using the 'enrich' function set in SQL queries

The `enrich` function set (see the table below) is used in the `SELECT` clause of an SQL query to enrich the results.

When using this set of functions, you cannot group events by enriched fields.

Widgets and reports enriched by the functions must be owned by no more than one tenant.

The values of asset fields that are specified in SQL queries are case-sensitive.

Description of the 'enrich' set of functions

Function	Event field	Asset field	Description
enrich_assets	DeviceAssetID DestinationAssetID SourceAssetID	name os osBuild ipAddress macAddress fqdn source software cii ramSize kscLastAvBasesUpdate kscLastInfoUpdate kscLastUpdate kscLastSystemStart kscStatusMask kscRtProtectionState kscEncryptionState kscAntiSpamStatus kscEmailAvStatus kscDlpStatus kscStatusID kscEdrStatus lastVisible aistatus aiscore	Enriches the query result with asset fields based on the ID specified passed to the function. For example: `enrich_assets(DeviceAssetID, name)`
enrich_accounts	SourceAccountID DestinationAccountID	name guid domain cn dn employee_id email mail_nickname mobile sid sam_name telephone_number principal_name sn type title division department manager location company street_address physical_delivery_office_name uac	Enriches the query result with account fields based on the ID specified passed to the function. For example: `enrich_account(SourceAccountID, name)` This function is available only for tenants that have LDAP enrichment enabled.

Function	Event field	Asset field	Description

`enrich_table`	`Any`	–	Enriches the query result with values from a dictionary of the **Table** type based on event fields in the response from the ClickHouse cluster.
			For example:
			`enrich_table('Main/some-table', [DeviceProduct], 'value')`
			Only one column of the table can be the key, but the record key can be a compound key.
			For example:
			`enrich_table('Main/Dictionary', [BytesOut|BytesIn], 'value2')`
			The `enrich_table` function returns only one column of the table. If you need to call more columns, call the function again specifying the relevant fields.

`enrich_dictionary`	`Any`	–	Enriches the query result with values from a dictionary of the **Dictionary** type based on event fields in the response from the ClickHouse cluster. For example: • `enrich_dictionary('Main/some-dict', [SourceAddress]` • `enrich_dictionary('Main/some-dict', [SourceAddress, DeviceAddress])` You can view the path to the dictionary in the **Resources** → **Dictionaries** section.

Example SQL queries with an enrich function

You can specify an `enrich` function when manually creating SQL queries. Example queries with an `enrich` function

- All events of the 'events' table with the `Name` and `Message` fields, as well as the `macAddress` and `osBuild` asset fields to enrich the result with asset data. The number of rows that can be displayed in the table is 250.
```
SELECT Name,
Message,
enrich_assets(DeviceAssetID, macAddress),
enrich_assets(DeviceAssetID, osBuild)
FROM `events`
WHERE DeviceAssetID != ''
LIMIT 250
```
- All events of the 'events' table with event fields `SourceUserName` and `SourceAddress`, as well as values of the `mobile` field to enrich the result with account data . The number of rows that can be displayed in the table is 10.
```
SELECT SourceUserName,
SourceAddress,
enrich_accounts(SourceAccountID, mobile) as mobile
FROM `events`
WHERE SourceAccountID != ''
LIMIT 10
```
- All events in the 'events' table with `DeviceProduct` event fields with values from the **Main/[OOTB] Linux Auditd record types** table. Sorted by **Timestamp** in descending order. The number of rows that can be displayed in the table is 1000.
```
SELECT DeviceProduct,
enrich_table('Main/[OOTB] Linux. Auditd record types', [DeviceProduct], 'Description') as HTTP
FROM `events` WHERE Type!=4
LIMIT 1000
```
- All events in the events table with `DeviceAction` event fields with values from the **Main/[OOTB] KEDR** dictionary. The number of rows that can be displayed in the table is 250.
```
SELECT DeviceAction,
enrich_dictionary('Main/[OOTB] KEDR. AccountType', [BytesOut]) as accountType
FROM `events` WHERE Type!=4
LIMIT 255
```

Using the 'lookup' function set in SQL queries

The `lookup` function set (see the table below) is used in the `WHERE` clause of an SQL query to add conditions for assets and accounts to the search query.

Function type	Event field	Asset field	Operator	Description
lookup_assets	DeviceAssetID SourceAssetID DestinationAssetID	name os osBuild source cii ramSize kscRtProtectionState kscEncryptionState kscAntiSpamStatus kscEmailAvStatus kscDlpStatus kscStatusID kscEdrStatus aistatus aiscore	=	Allows you to add conditions t result for assets whose field v value specified in the query. For values that are number, yo string with the required numbe For example: lookup_assets(DeviceAss '=', '19045.000000') lookup_assets(DeviceAss '=', '3241234234')
		kscLastAvBasesUpdate kscLastInfoUpdate kscLastUpdate kscLastSystemStart lastVisible ramSize aiscore	< > <= >=	Allows you to add conditions f field values that are: • Less than the value speci • Greater than the value sp query • Less than or equal to the the query • Greater than or equal to t specified in the query For values that are number, yo string with the required numbe For example: lookup_assets(DeviceAss '>', '50') Date values can be specified i formats: • Number of seconds: lookup_assets(Device lastVisible, '>', '1 • Number of milliseconds: lookup_assets(Device lastVisible, '<', '1725971837834') • Number of days: lookup_assets(Device lastVisible, '<', 'n
		name os osBuild fqdn software	like	Allows you to add conditions t result for assets whose field v 'like' expression specified in th For more information about th please refer to the SQL Langu Expressions reference. lookup_assets(Destinati name, 'like','test.compa
		ipAddress macAddress fqdn software kscStatusMask	in	Allows you to add to the result whose values match the value query. For example: lookup_assets(DeviceAss 'in', ['test_fqdn1.compa 'test_fqdn2.company.ru'

Function type	Event field	Asset field	Operator	Description
		`ipAddress`	`insubnet`	Allows you to add conditions f addresses in the specified sub address is specified in CIDR fo For example: `lookup_assets(DeviceAss ipAddress, 'insubnet',`
		`ipAddress`	`inrange`	Allows you to add conditions f addresses in the specified ran The first IP address specified i start IP address, and the seco address. For example: `lookup_assets(DeviceAss ipAddress, 'inrange', [ '192.168.1.184'])` The query filters assets with IP range from 192.168.1.1 to 192.168

Function type	Event field	Asset field	Operator	Description
lookup_accounts	SourceAccountID DestinationAccountID	name guid domain cn dn employee_id email mail_nickname mobile sid sam_name telephone_number principal_name sn title division department manager location company street_address physical_delivery_office_name uac	=	Allows you to add conditions t result for accounts whose fiel the value specified in the quer For values that are number, yo string with the required numbe For example: lookup_accounts(SourceA employee_id, '=', '1545

Function type	Event field	Asset field	Operator	Description
		name guid domain cn dn employee_id email mail_nickname mobile sid sam_name telephone_number principal_name sn type title division department manager location company street_address physical_delivery_office_name	like	Allows you to add conditions t result for accounts whose fiel the 'like' expression specified i For more information about th please refer to the SQL Langu Expressions reference. lookup_accounts(Destina name, 'like', '_ccount1
lookup_assets_category	DeviceAssetID SourceAssetID DestinationAssetID	–	–	Finds all events that have assc the specified folder. For example: lookup_assets_category(['Main/Business Impact'] where Main is the name of th Business Impact is the nar

Example SQL queries with a lookup function

You can specify a `lookup` function when manually creating SQL queries. Example queries with a `lookup` function

- All events of the 'events' table with the `TenantID` and `DeviceProduct` event fields, as well as the `fqdn` asset field, whose value contains the win10 substring. The number of rows that can be displayed in the table is 250.

```
SELECT TenantID, DeviceProduct
FROM `events`
WHERE DeviceAssetID != ''
AND lookup_assets(DeviceAssetID, fqdn, 'match', 'win10')
LIMIT 250
```

- All events in the 'events' table, with the `TenantID` and `DeviceProduct` event fields, and with values of `the EmployeeID` field, whose value is `15452.` The number of rows that can be displayed in the table is 10.

```
SELECT TenantID, DeviceProduct
FROM `events`
WHERE SourceAccountID != ''
AND lookup_accounts(SourceAccountID, employee_id, '=', '15452')
LIMIT 10
```

- All events of the 'events' table with the `TenantID` and `DeviceProduct` event fields, with the folder in which the asset is located. The number of rows that can be displayed in the table is 100.

```
SELECT TenantID, DeviceProduct
FROM `events`
WHERE DeviceAssetID != ''
AND lookup_assets_category(DeviceAssetID, ['Main/Business Impact'])
LIMIT 100
```

Event route tracing

To find out which connection the events were received from, you can use the **Track event route** setting. You may need to trace the route of an event when multiple agents send events to the same collector, for example, in a large infrastructure where agents are installed on dedicated WEC servers and multiple agents send events to the same collector. The event route information can be useful for troubleshooting event routing.

For the **Track event route** toggle switch to become available, you must specify at least one **internal** destination in the agent. Also, to trace event routes, a connector of the **internal** type must be specified in the collector that is receiving events from the agent. After configuring and saving the agent, information about the agent's route is added to the **S.KL_EventRoute** field of the extended event schema.

The **S.KL_EventRoute** field is displayed only for new events that are received by the collector after enabling the **Track event route** setting. Other services through which the event passes, including the collector and router, parse the **S.KL_EventRoute** field, and, if the field is not empty, add their own information to the field when they process the event.

If the event passes through multiple hosts on its way from the sender to the agent, the address of the last host that established a connection with the agent is saved in the **S.KL_EventRoute** field. If there is a proxy server between the sender and the agent, the address of this proxy server is written to the field, and no separate proxy server configuration is required on the agent. The tcp/udp/http connectors pass the address of the host from which the event was received via the **internal** protocol to **S.KL_EventRoute**, and the WEC, WMI, and ETW agents specify the host name of the Windows server on which they are installed.

You can enable event route tracing in one of the following ways:

- Create a new agent, specify the **internal** destination in it and enable the **Track event route** option.

- In an existing agent, add a tab with a new connection and specify the **internal** destination on this tab. After you specify the **internal** destination, the **Track event route** option becomes available. Enable the **Track event route** toggle switch. After you save the agent settings, restart the agent to apply the changes. In this case, route tracing only affects the tab on which the **internal** destination is selected, not the entire agent.

- Unlink the automatically created agent from the collector to make the agent available for editing its settings, or duplicate the automatically created agent and proceed to configure the settings in the duplicate agent. After you specify an **internal** destination in an editable agent, the **Track event route** option becomes available. Enable the **Track event route** toggle switch. After you save the agent settings, restart the agent to apply the changes. In this case, route tracing only affects the tabs on which the **internal** destination is selected, not the entire agent.

The **Event tracking log** section becomes available in the event card. This section displays information from the **S.KL_EventRoute** field in a processed form. Service IDs are converted to service names and displayed as clickable links. Clicking a service name opens a new browser tab with the card of the service. If you rename the service, the name of the service also changes when the event is displayed in the card, both for new events and for events that have already been received and processed. If you delete a service in the **Active services** section, the **Event tracking log** section displays **Deleted** instead of the hyperlink. The rest of the route information is not deleted and is still displayed, including the connector type, FQDN, the value of the **SourceAddress** field, and other route parameters from the **S.KL_EventRoute** field. To view raw route information, you can add the **S.KL_EventRoute** column to the event table.

Using route information

You can use the 'Extract from JSON' function or other ClickHouse functions to get the parts of the route that you need for debugging. For more information about functions, please refer to the ClickHouse documentation: https://clickhouse.com/docs/en/sql-reference/functions/json-functions#jsonextractstringjson-indices-or-keys

The following table gives examples of queries.

Query example	Description
`SELECT *, simpleJSONExtractRaw(S.KL_EventRoute, 'version') as version FROM `events` Where Type !=4 ORDER BY Timestamp DESC LIMIT 25`	This query displays events for which the version of the collector is displayed in the **version** column.
`SELECT *, simpleJSONExtractRaw(S.KL_EventRoute, 'connectorKind') as connectorKind FROM `events` Where Type !=4 ORDER BY Timestamp DESC LIMIT 250`	This query displays events for which the type of the collector is displayed in the **connectorKind** column.
`SELECT *, simpleJSONExtractRaw(S.KL_EventRoute, 'connectorKind') as connectorKind FROM `events` Where Type !=4 and simpleJSONExtractRaw(S.KL_EventRoute, 'connectorKind') != '"wec"' ORDER BY Timestamp DESC LIMIT 250`	This query displays events received from connectors of a type other than 'wec'.

If want to select by a different value from the **S.KL_EventRoute** field, you can look up the name of your parameter in the events table in the **Raw** column and use it in your query.

Categorization of events

Categorization allows you to use special tags (categories) to describe events of the same type coming from different sources. This helps to find events related to, for example, user authentication or command execution, and helps in displaying data in a dashboard or in reports. You can use the LogAnalyzer resource set to categorize certain types of events for specific types of event sources.

The following types of LogAnalyzer resources are used to categorize events:

- A dictionary with categories of events
- Enrichment rules for enriching events with supplemental information.
- A normalizer for creating an extended event schema field used in enrichment rules.

As part of categorization, the following attributes can be added to an event:

- Object
- Action
- Result
- Threat
- Source type

Additional attributes can be assigned to the event, which are saved in the SA.KL_EventCategory extended event schema field.

Descriptions of the possible values of the attributes used for categorizing events are provided in tables: *Object attribute*, *Action attribute*, *Result attribute*, *Threat attribute*, *Source type attribute*.

To customize the categorization of events on a collector:

1. Import the [OOTB] Event Categorization package from the LogAnalyzer repository.
2. In the collector, at the **Event enrichment** step, apply an event enrichment rule.

   You can apply the enrichment rule when creating a new collector or for an existing collector. The rule that you need to apply depends on the event source. You can select a rule from the *Categorization rules by event source* table.

   > Event categorization works only when using Kaspersky normalizers.

3. In the LogAnalyzer web interface, select the **Resources** → **Active services** section.
4. Select the check box next to the modified collector and click the **Update configuration** button.

The enrichment rule is applied, and the categorization of events is performed.

Categorization rules by event source

System name	Normalizer name	Enrichment rule name
Microsoft, OS event logs	[OOTB] Microsoft Products for KUMA 3	[OOTB] Event categorization. Microsoft Products
Microsoft Sysmon	[OOTB] Microsoft Products for KUMA 3	[OOTB] Event categorization. Microsoft Products
Auditd	[OOTB] Linux auditd syslog for KUMA 3.2	[OOTB] Event categorization. Auditd
Kaspersky Security Center	[OOTB] KSC from SQL	[OOTB] Event categorization. Kaspersky Security Center
Kaspersky Security for Linux Mail Server	[OOTB] KLMS syslog CEF	[OOTB] Event categorization. Kaspersky Security for Linux Mail Server
Kaspersky IoT Secure Gateway	[OOTB] Kaspersky KISG Syslog	[OOTB] Event categorization. Kaspersky IoT Secure Gateway
Kaspersky Container Security	[OOTB] Syslog-CEF	[OOTB] Event categorization. Kaspersky Container Security
Kaspersky Industrial CyberSecurity for Networks	[OOTB] Kaspersky Industrial CyberSecurity for Networks 4.2 Syslog	[OOTB] Event categorization. Kaspersky Industrial CyberSecurity for Networks
Kaspersky Web Traffic Security	[OOTB] KWTS syslog CEF	[OOTB] Event categorization. Kaspersky Web Traffic Security
Kaspersky Mail Security Gateway	[OOTB] KSMG syslog CEF	[OOTB] Event categorization. Kaspersky Mail Security Gateway

Object attribute

Value of the Object attribute	Description
account	Accounts
configuration	Settings
connection	Network connection
container	Containers
data	Data or tables in the database
dns	DNS queries
email	Email
file	Files
group	Groups
host	Host

http	WEB requests
image	DLLs and drivers
malware	Malicious object
permission	Rights and privileges
process	Process
registry	Registry keys
service	Services and daemons
task	Tasks
VM	Virtual machine
device	Token, USB device, attached device
certificate	Certificate

Action attribute

Value of the Action attribute	Description
access	Request or grant access to an object
add	Create or add
authentication	Authenticate
block	Block, prevent
delete	Delete
detect	Detect or quarantine
end	End of the process
info	Informational event about the object
load	Load object
modify	Modify settings or state of the object
read	Read
receive	Receive
request	Query
send	Send
start	Start of the process
write	Write

Result attribute

Value of the Result attribute	Description
error	Error
failure	Failure
success	Success

Threat attribute

Value of the Threat attribute	Description
malware	Malicious object in the file system
vulnerability	Vulnerability exploits
attack	Attack
ddos	Indicators of a DDoS attack
phishing	Indicators of phishing
c2	Command and Control

discovery	Reconnaissance
policy violation	Violation of security policies
tools	Suspicious tools used
escalation	Privilege escalation

Source type attribute

Value of the Source type attribute	Description
application	Application-level events: logins, configuration changes, errors, and so on.
AV	Events received from anti-virus protection systems.
database	Events from databases related to data manipulation.
IDS	Events from traffic analyzers and network protection tools.
network	Events containing information about network connections coming from firewalls; also events from Jflow, and so on.
OS	Events from the operating system, such as Windows, Linux, Cisco IOS, HP-UX, and so on. Standard operating system tools or EDR can be the event provider.
vpn	VPN session events.
web	Events coming from WEB applications and Proxy. Application-level HTTP connection events.
container	Events providing information about the operation of containers.

Granular access to events

In LogAnalyzer, users with different rights can have granular access to events. Access to events is controlled at the level of storage spaces.

You can assign spaces to users in the **Access to spaces** section or directly in a user's card. After upgrading to version 4.0, the 'All spaces' space set is assigned to all existing users, that is, access to all spaces is unrestricted. An event contains a tenant ID and a space ID, therefore the user needs rights to the corresponding tenant and space to have access to the event.
Keep in mind the following special considerations involved in displaying storages:

- If a storage is not listed in the **Active services** section, the storage and its spaces are not displayed in the list of spaces of the set.
- If the storage service was stopped using the `systemctl stop kuma-<storage ID>` command, the storage and its spaces are not displayed in the list of spaces of the set.
- If the storage was started and then deleted using the `uninstall` command, the storage and its spaces remain in the list of spaces of the set.

In the list of events, you can add the **SpaceID** field to the table, which will display the name of the space. The space of audit events is displayed as KUMA Audit. KUMA Default is the space inside each storage, where all events go if the storage does not have configured spaces or if the event does not match the conditions of the existing spaces.

When you export the list of events to a TSV file, the space ID and name are displayed for spaces.

To differentiate access:

1. Assign appropriate roles to users.
2. Configure the space sets.

   You can create, edit, or delete space sets. These actions result in audit events.
3. Configure the access rights of the space set: you can grant or revoke access rights of selected users.

Creating a space set

To create a space set:

1. In the LogAnalyzer web console, go to the **Settings** → **Access** → **Access to spaces** section.
2. In the opened window, click the **Add** button.
3. In the **Create space sets** window, configure the following:
   a. In the **Name** field, specify the space set name. You can use up to 128 Unicode characters.
   b. In the **Spaces** drop-down list, select the spaces that you want to include in the set by selecting the check box next to the space name. To remove a space from the set, clear the check box.
   c. If you want the default space set to be available to all users that do not have any of the sets assigned, select the **Make default** check box.

      This makes the created set the default set. In the list of space sets, it is marked as the default set: in the **Default** column, it has **Yes**.

      There can be only one default set. If you select **Make default** check box in the settings of a different set and save the settings, the default set is reassigned.
4. Click **Create**.

The space set is created and displayed in the list of spaces. After the space set is created, an audit event is generated in LogAnalyzer.

Then you need to grant users access rights to the created space set.

Editing a space set

To edit a space set:

1. In the LogAnalyzer web console, go to the **Settings** → **Access** → **Access to spaces** section.
2. In the opened window, click a set of spaces.
3. In the **Edit space sets** window, configure the following:
   a. In the **Name** field, specify the space set name. You can use up to 128 Unicode characters.
   b. In the **Spaces** drop-down list, select the spaces that you want to include in the set by selecting the check box next to the space name. To remove a space from the set, clear the check box.
   c. If you want the default space set to be available to all users that do not have any of the sets assigned, select the **Make default** check box.

      This makes the set the default set when you save the changes. In the list of space sets, it is marked as the default set: in the **Default** column, it has **Yes**.

      There can be only one default set. If you select **Make default** check box in the settings of a different set and save the settings, the default set is reassigned.
4. Click **Save**.

The space set is saved and displayed in the list of spaces.

The edited set of spaces is accessible to all users who had access to the original space set. You can control access to the edited space set and grant or revoke access permissions using the **Access control** button.

Deleting a space set

To delete a space set:

1. In the LogAnalyzer web console, go to the **Settings** → **Access** → **Access to spaces** section.
2. In the **Access to spaces** window, select the check box next to the relevant space set.

   If you want to delete all space sets, select the **Select all** check box in the upper left part of the list and click **Delete**.

   > You cannot delete the default space set and the 'All spaces' space set.

3. Click **Delete**.

   The space set is deleted and removed from the list of spaces. After the space set is deleted, an audit event is generated in LogAnalyzer.

If a user had access only to the deleted space set and did not have access to any other sets, such a user automatically gets access rights to the default space set.

Grant access to a space set

To grant access to a space set

1. In the LogAnalyzer web console, go to the **Settings** → **Access** → **Access to spaces** section.
2. Select a set of spaces in the list by selecting the check box next to it, then click the **Access control** button on the toolbar.
3. This opens the **Users** window; in that window, click **Add users**.
4. In the list in the **Add users** window, select users to which you want to grant access to the space set and click **Add**.

   If you want to grant access to the selected space set to all users, select the check box in the upper left corner and select one of the following options: **Select all in page** or **Select all**, then click **Add**.

   This opens a confirmation window.
5. If you want to grant access to the listed users, click **Confirm** in the confirmation window.

The users now have access to the space set.

Revoke access to a space set

To revoke access to a space set

1. In the LogAnalyzer web console, go to the **Settings** → **Access to spaces** section.
2. Select a set of spaces in the list by selecting the check box next to it, then click the **Access control** button on the toolbar.
3. This opens the **Users** window; in that window, select users by selecting their check boxes, then click **Delete**.

   If you want to revoke access to the selected space set from all users, select the check box in the upper left corner and select one of the following options: **Select all in page** or **Select all**, then click **Delete**.

   This opens a confirmation window.
4. If you want to revoke access from the listed users, click **Confirm** in the confirmation window.

The users no longer have access to the space set.

If a user does not have access to any other space set, such a user automatically gets access to the default set.

Use cases

Migrating to KUMA 4.0 with differentiated access to events

Goal: Update the product.

Action: The administrator updates LogAnalyzer to version 4.0.

Result: The accessibility of events does not change for users; all users have access to events in those tenants in which users are allowed to view the list of events. When adding new users, tenants, or spaces, all users see the events of tenants depending on whether they are allowed to view the list of events, with no further restrictions.

Restricting access to spaces for all users

Goal: Some spaces in the installation must be inaccessible to some users. Current and new users must not have access to such spaces.

Action: The administrator creates a new set of spaces that includes the spaces to which all users must have access. Each space is specified as a tuple: clusterID, tenantID, spaceID. In this way, you can assemble a set of spaces from a finite number of spaces. After creating the set, the administrator makes this set the default set.

Result: For all users that do not have this space set explicitly specified, including new users, access to events is additionally restricted in accordance with the default space set. That is, such users have access to events in tenants in which the user has the right to view the list of events, and events in spaces that are included in the default set.

Allowing some users to view all events

Goal: LogAnalyzer has a default set in which only a finite number of spaces is explicitly specified. Also some users in LogAnalyzer must have unrestricted access to all events. You want to allow these users to view events from spaces that are not specified in the default set.

Action: The administrator goes to the **Access to spaces** section and selects the 'All spaces' set, which gives access to all spaces without restrictions, and grants access to this set to the selected users.

Result: Users that have been explicitly granted access to the 'All spaces' set now have unrestricted access to events in spaces. Their access to events depends only on the right to view the list of events in accordance with their assigned role. When you add new spaces or tenants, these users will have access to the events located in these new spaces and tenants, provided they have the right to view the list of events for the corresponding tenant. Users that have not been explicitly granted access to the 'All spaces' set will have restricted access to events in accordance with the default set of spaces.

Permitting some users to view events from a finite set of spaces

Goal: LogAnalyzer has a default set in which only a finite number of spaces is explicitly specified. Some users need to see a different space set, but not all available spaces.

Action: The administrator creates a space set that includes the relevant spaces and grants access to this set to the selected users.

Result: User access to events is restricted in accordance with the available space set. Users that have not been explicitly granted access to the set will have restricted access to events in accordance with the default set of spaces.

Supplementing an explicitly specified space set for a user

Goal: Users u1, u2, u3, u4 have been explicitly granted access to space set 'set1', which includes spaces to which Windows and Linux events go. Users u1 and u2, in addition to Windows and Linux events, need to view Cisco and VMware events that go to the respective spaces.

Action: The administrator creates a set of spaces, 'set2', which includes the spaces of Cisco and VMware events. The administrator assigns set2 to u1 and u2 in addition to set1.

Result: Users u1 and u2 now have access to events from the spaces of all of the sets available to them, that is, events from the set1 and set2: Windows, Linux, Cisco, VMware. Access of users u3 and u4 to events remains unchanged.

Editing a space set

Goal: Some users have access to a space set (access is granted explicitly or the set is the default set). For all of these users, you need to add access to a certain space that is not in the set, or revoke access to a space that is in the set.

Action: The administrator edits the existing set of spaces by adding or removing the relevant spaces.

Result: All users that have access to the set can now view events in accordance with the updated space set.

Deleting a space set

Goal: The space set is no longer relevant, and it needs to be deleted from LogAnalyzer.

Action: The administrator deletes an existing space set. The 'All spaces' set cannot be deleted. The default space set cannot be deleted.

Result: The deleted space set is gone from LogAnalyzer. All users that have been explicitly granted access to this set of spaces now have access to events in accordance with the space sets that remain available to them. If a user does not have any space sets left to which access has been explicitly granted, such a user's access to events is controlled by the default set.

Dashboards

In the **Dashboards** section, you can monitor the security status of your organization's network.

The dashboard is a set of widgets that display network security data analytics. You can view data only for those tenants to which you have access.

You can create dashboards manually or use predefined dashboards. If necessary, you can manage the settings of the widgets in the predefined dashboards. The predefined Alerts Overview dashboard is used as the default dashboard.

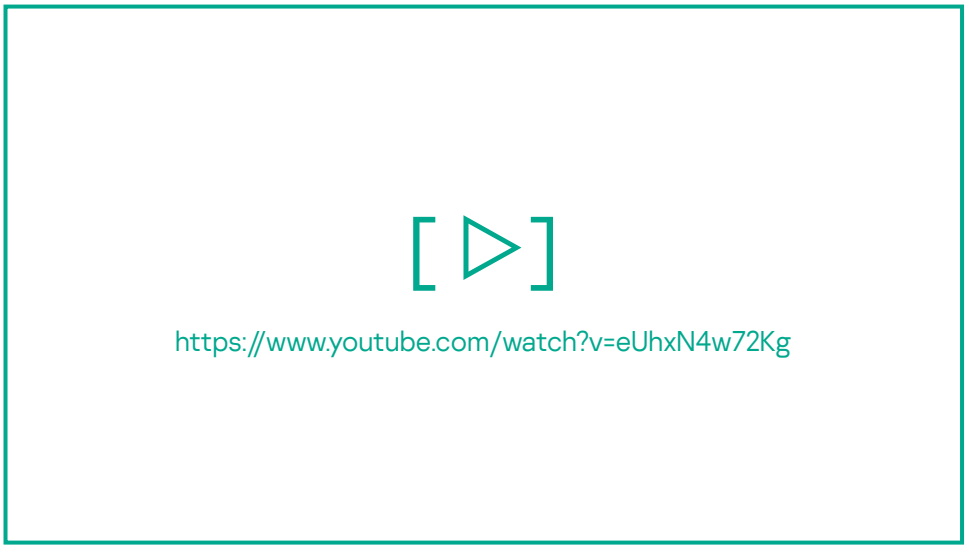
Only users with the Main administrator, Tenant administrator, Tier 2 analyst, and Tier 1 analyst roles can create, manage, or delete dashboards. Users accounts with all roles can view dashboards and set default dashboards. If a dashboard is set as default, that dashboard is displayed for the account every time the user navigates to the **Dashboards** section. The selected default dashboard is saved for the current user account.

The information in the **Dashboards** section is updated in accordance with the schedule configured in dashboards settings. If necessary, you can force the update of the data.

For convenient presentation of information on dashboards, you can enable TV mode. This mode lets you view the dashboards in full-screen mode in FullHD resolution. In TV mode, you can also configure a slide show display for the selected dashboards.

If necessary, you can view a list of all available dashboards for tenants to which they belong in the left part of the window. You can click the **Hide dashboards** button to collapse this list, in which case the widgets of existing dashboards are displayed. You can bring back the list of all dashboards available for tenants by clicking the **Show dashboards** button.

In the list of all dashboards available for tenants, you can also click the add icon + to create folders for grouping dashboards in tenants. You can manage the settings of these folders or delete them by clicking the menu icon ⋮ and selecting an action.



https://www.youtube.com/watch?v=eUhxN4w72Kg

Creating a dashboard

To create a dashboard:

1. Open the LogAnalyzer web interface and select the **Dashboards** section.
2. Do one of the following:
   - Click the menu ⋮ icon in the upper-right corner of the window and select **Create dashboard**.
   - Click the **Create dashboard** button under the list of all dashboards.

   This opens the dashboard settings window.
3. In that window, specify a unique name for the dashboard. The name must contain 1 to 128 Unicode characters.
4. In the **Dashboard tenant** drop-down list, select the tenant that will own the new dashboard, or a certain folder in the tenant.

   If you had any set of charts from a folder in the tenant open before creating the dashboard, LogAnalyzer automatically specifies this tenant when creating the dashboard. You can change the tenant as necessary.
5. In the **Collect data from tenants** drop-down list, select one or more tenants whose data you want to be represented in the dashboard.

6. If necessary, in the **Time period** drop-down list, select the time period from which you want to get analytics in one of the following ways:

   - If you want to specify an exact date, in the calendar on the left, select the start and end date of the period and click **Apply**.

     You can select a date up to and including the current date. The date and time format depends on your browser settings. If the **Range from** or **Range to** field has a value and you have not edited the time value manually, when you select a date in the calendar, the **Range from** field is automatically populated with 00:00:00.000, and the **Date to** field with 23:59:59.999. If you have manually deleted the value in the **Range from** or **Range to** field, when you select a date in the calendar, the field is automatically populated with the current time. After you select a value in one of the fields, the focus switches to the other field. If your **Range to** is earlier than your **Range from**, this earlier value is automatically inserted into the **Range from** field.

   - If you want to specify a relative period, select one of the available periods in the **Relative period** list on the right.

     The period is calculated relative to the current time.

   - If you want to specify a custom period, edit the value of the **Range from** and **Range to** fields.

     You can enter an exact date and time in the **DD.MM.YYYY HH:mm:ss.SSS** format for the Russian localization and **YYYY-MM-DD HH:mm:ss.SSS** for the English localization or a period relative to the current time as a formula. You can also combine these methods if necessary. If you do not specify milliseconds when entering the exact date, 000 is substituted automatically. If you have edited the time in the **Range from** or **Range to** fields, picking a date in the calendar does not change the time component.

     In the relative period formulas, you can use the **now** parameter for the current date and time and the interval parameterization language: **+** (only in the **Range to** field), **-**, **/** (rounding to the nearest), as well as time units: **y** (year), **M** (month), **w** (week), **d** (day), **h** (hour), **m** (minute), **s** (second). For example, you can specify the period **now-5d** to get data for the last five days, or **now/w** to get data from the beginning of the first day of the current week (00:00:00:000 UTC) to the current time (**now**).

     > The **Range from** field is required, and its value cannot exceed the value of the **Range to** field, and also cannot be earlier than 1970-01-01 (if specifying an exact date or a relative period). The **Range to** cannot be earlier than the **Range from**. If you do not specify a value in the **Range to** from field, **now** is specified automatically.

By default, the **1 day** (**now-1d**) relative period is selected. The bounds of the period are inclusive: for example, for the **Today** time range, events are displayed from the beginning (00:00:00:000 UTC) of the current day to the current time (**now**) inclusive, and for the **Yesterday** period, events are displayed from the beginning (00:00:00:000 UTC) of the previous day to 00:00:00:000 UTC of the current day.

LogAnalyzer stores time values in UTC, but in the user interface time is converted to the time zone of your browser. This is relevant to the relative periods: **Today**, **Yesterday**, **This week**, and **This month**. For example, if the time zone in your browser is UTC+3, and you select **Today** as the data display period, data will be displayed for the period from 03:00:00.000 until now, not from 00:00:00.000 until now.

If you want to take your time zone into account when selecting a relative data display period, such as **Today**, **Yesterday**, **This week**, or **This month**, you need to manually add a time offset in the **Range from** and **Range to** (if a value other than **now** is specified) by adding or subtracting the correct number of hours. For example, if your browser's time zone is UTC+3 and you want to display data for **Yesterday**, you need to change **Range from** to **now-1d/d-3h** and **Date to** to **now/d-3h**. If you want to display data for the **Today** period, you only need to change the value in the **Range from** field to **now/d-3h**.

> If you need results up to 23:59:59:999 UTC of yesterday, you can use an SQL query with a filter by Timestamp or specify an exact date and time.

7. If necessary, in the **Refresh interval** drop-down list, select the data refresh period in the dashboard widgets:

   - **No refresh** means data in dashboard widgets are never refreshed.
   - **1 minute**
   - **5 minutes**
   - **15 minutes**
   - **1 hour** (default)
   - **3 hours**
   - **6 hours**
   - **12 hours**
   - **24 hours**

8. If necessary, select tags for the dashboard from the **Tags** drop-down list.

9. If necessary, add a description of the dashboard.

10. Click **Apply**.

11. Click the **Add widget** button and select a widget. This opens a window; in that window, configure the selected widget.

    You can add multiple widgets. You can drag widgets around the window by the icon ⠿ in the upper-left corner of the widget. If you need to resize the widget, you can do so using the icon ⤡ that is displayed when you hover over the lower-right corner of the widget.

    The following limitations apply to widgets with the **Pie chart**, **Bar chart**, **Line chart**, **Counter** , and **Date Histogram** chart types:

    - In `SELECT` queries, you can use extended event schema fields of `String`, `Number`, and `Float` types.
    - In `WHERE` queries, you can use all types of extended event schema fields (`String`, `Number`, `Float`, `Array of strings`, `Array of numbers`, and `Array of floats`).

    For widgets with the **Table** chart type, in `SELECT` queries, you can use all types of extended event schema fields (`String`, `Number`, `Float`, `Array of strings`, `Array of numbers`, and `Array of floats`).

    By clicking the ⋮ icon in the upper-right corner of the widget, you can do the following:

    - Manage the settings of widgets
    - Create copies of widgets
    - Delete widgets
    - Download widget data in CSV format

12. Click **Create**.

The new dashboard is created and displayed in the **Dashboards** section of the LogAnalyzer web interface.

You can also create a dashboard on the **List** tab in the **Resources** section. To do this, click the **Add** button and select the **Dashboard** resource type.

Selecting a dashboard

To select a dashboard:

1. Expand the list of folders with dashboards in the upper-left corner of the **Dashboards** window.

   Folders with dashboards are organized in accordance with the tenants they belong to.

2. If you want to view the dashboards that you have created, click the **Created by me** button.

   LogAnalyzer display only the dashboards that you have created and only those folders in which these dashboards are located. The folders in which your dashboards are located may have been created by other users.

3. Select a folder and a dashboard in that folder.

The selected dashboard is displayed in the **Dashboards** window of the LogAnalyzer web interface.

Selecting a dashboard as the default dashboard

To select the dashboard to be displayed by default:

1. In the LogAnalyzer web interface, select the **Dashboards** section.
2. Expand the list of folders with dashboards in the upper-left corner of the **Dashboards** window.
3. Click the relevant dashboard.
4. Do one of the following:

   - Click the menu icon ⋮ next to the dashboard in the list of folders with dashboards, and in the menu, select **Set as default**.
   - Click the menu icon ⋮ in the upper-right corner of the **Dashboards** window with the dashboard open, and in the menu, select **Set as default**.

The selected dashboard is marked with the **Default** attribute and becomes the default dashboard.

Managing dashboard settings

To manage dashboard settings:

1. In the LogAnalyzer web interface, select the **Dashboards** section.
2. Expand the list of folders with dashboards in the upper-left corner of the **Dashboards** window.
3. Click the relevant dashboard.
4. Click the menu icon ⋮ next to the dashboard in the list of folders with dashboards, and in the menu, select **Edit**.

5. Make your changes in the window that is displayed. The settings that are available for editing are the same as the settings available when creating a dashboard.

6. Click the **Save** button.

The dashboard settings are modified.

If the dashboard was deleted or assigned to a different tenant while you were managing its settings, an error is displayed when you click the **Save** button. The dashboard is not saved. Refresh the LogAnalyzer web interface page to see the list of available dashboards in the drop-down list.

Deleting a dashboard

To delete a dashboard:

1. In the LogAnalyzer web interface, select the **Dashboards** section.

2. If necessary, click the **Show dashboards** button in the upper-left corner of the **Dashboards** window and open the list of folders with dashboards.

3. Click the relevant dashboard.

4. Do one of the following:

   - Click the menu icon ⋮ next to the dashboard in the list of folders with dashboards, and in the menu, select **Delete**.

   - Click the menu icon ⋮ in the upper-right corner of the **Dashboards** window with the dashboard open, and in the menu, select **Delete**.

5. Confirm the deletion.

The dashboard is deleted.

Enabling and disabling the TV mode

It is recommended to create a separate user with the minimum required set of right to display analytics in TV mode.

To enable TV mode:

1. In the LogAnalyzer web interface, select the **Dashboards** section.

2. Click the **TV mode** button in the upper-right corner.

3. In the displayed window, move the **TV mode** toggle switch to the **Enabled** position.

4. To configure the dashboard slideshow:

    a. Move the **Slideshow** toggle switch to the **Enabled** position.

    b. In the **Timeout** field, specify how long each dashboard is displayed, in seconds.

    c. Under **Queue**, select the dashboards to display and click **Add to queue**. If no dashboard is selected, the slideshow iterates through all dashboards available to the user.

    d. If necessary, you can delete a dashboard. To do this, select the check box of the dashboard in the queue and click **Delete**.

    e. If necessary, you can change the order of the dashboards by dragging them with the right mouse button.

5. Click the **Save** button.

TV mode will be enabled.

To disnable TV mode:

Click the **Stop** button in the upper-right corner of the window where the dashboards are displayed.

TV mode is disabled. The left part of the screen shows a pane containing sections of the LogAnalyzer web interface.

If changes are made to the dashboards selected for the slideshow, such changes are automatically reflected in running slideshow sessions.

Predefined dashboards

LogAnalyzer comes with a set of predefined dashboards: The default refresh period for predefined dashboards is **Never**. You can modify these dashboards as necessary.

Predefined dashboards

Dashboard name	Description of widgets in the dashboards
Alerts Overview	• Active alerts—number of alerts that have not been closed. • Unassigned alerts—number of alerts that have the **New** status. • Latest alerts—table with information about the last 10 unclosed alerts belonging to the tenants selected in the dashboard. • Alerts distribution—number of alerts created during the period configured for the widget. • Alerts by severity—number of unclosed alerts grouped by their severity. • Alerts by assignee—number of alerts with the **Assigned** status. The grouping is by account name. • Alerts by status—number of alerts that have the **New**, **Opened**, **Assigned**, or **Escalated** status. The grouping is by status. • Affected users in alerts—number of users associated with alerts that have the **New**, **Assigned**, or **Escalated** status. The grouping is by account name. • Affected assets—table with information about the level of importance of assets and the number of unclosed alerts they are associated with. • Affected assets categories—categories of assets associated with unclosed alerts. • Top event source by alerts number—number of alerts with the **New**, **Assigned**, or **Escalated** status, grouped by alert source (DeviceProduct event field). The widget displays up to 10 event sources. • Alerts by rule—number of alerts with the **New**, **Assigned**, or **Escalated** status, grouped by correlation rules.
Incidents Overview	• Active incidents—number of incidents that have not been closed. • Unassigned incidents—number of incidents that have the **Opened** status. • Latest incidents—table with information about the last 10 unclosed incidents belonging to the tenants selected in the dashboard. • Incidents distribution—number of incidents created during the period configured for the widget. • Incidents by severity—number of unclosed incidents grouped by their severity. • Incidents by assignee—number of incidents with the **Assigned** status. The grouping is by user account name. • Incidents by status—number of incidents grouped by their status. • Affected assets in incidents—number of assets associated with unclosed incidents. • Affected users in incidents—users associated with unclosed incidents. • Affected asset categories in incidents—categories of assets associated with unclosed incidents. • Active incidents by tenant—number of incidents of all statuses, grouped by tenant.
Network Overview	• Netflow top internal IPs—total volume of netflow traffic received by the asset, in bytes. The data is grouped by internal IP addresses of assets. • The widget displays up to 10 IP addresses. • Netflow top external IPs—total volume of netflow traffic received by the asset, in bytes. The data is grouped by external IP addresses of assets. • Netflow top hosts for remote control—number of events associated with access attempts to one of the following ports: 3389, 22, 135. The data is grouped by asset name. • Netflow total bytes by internal ports—number of bytes sent to internal ports of assets. The data is grouped by port number. • Top Log Sources by Events count—top 10 sources from which the greatest number of events was received.
[OOTB] KATA & EDR	• KATA. Top-10 detections by type — visualizes the 10 most common types of events detected by the KATA system. • KATA. Top-10 detections by file type — visualizes the 10 most common file types detected by the KATA system. • KATA. Top-10 user names in detections — visualizes the 10 most common user names detected by the KATA system. • KATA. Top-10 IDS detections — visualizes the 10 most common threats detected by the IDS module of the KATA system. • KATA. Top-10 URL detections — visualizes the 10 most common suspicious URLs detected by the KATA system. • KATA. Top-10 AV detections — visualizes the 10 most common threats detected by the KATA anti-virus module. • EDR. Top-10 MITRE technique detections — visualizes the 10 most common MITRE ATT&CK matrix techniques detected by the EDR system. • EDR. Top-10 MITRE tactic detections — visualizes the 10 most common MITRE ATT&CK matrix tactics detected by the EDR system.

Dashboard name	Description of widgets in the dashboards
[OOTB] KSC	• KSC. Top-10 users with the most KAV alerts — visualizes the 10 most common user names present in events related to the detection of malicious software, information about which is contained in the KSC system. • KSC. Top-10 most common threats — visualizes the 10 most common types of malware, information about which is contained in the KSC system. • KSC. Number of devices that received AV database updates — visualizes the number of devices on which anti-virus database updates have been installed, information about which is contained in the KSC system. • KSC. Number of devices on which the virus was found — visualizes the number of devices on which malware was detected, information about which is contained in the KSC system. • KSC. Malware detections by hour — visualizes the distribution of the number of malware per hour, information about which is contained in the KSC system.
[OOTB] KSMG	• KSMG. Top-10 senders of blocked emails — visualizes the 10 most common senders of email messages blocked by the KSMG system. • KSMG. Top-10 events by action — visualizes the 10 most common actions performed by the KSMG system. • KSMG. Top-10 events by outcome — visualizes the 10 most common results of actions performed by the KSMG system. • KSMG. Blocked emails by hour — visualizes the distribution of the number of email messages blocked by the KSMG system, by hour.
[OOTB] KWTS	• KWTS. Top-10 IP addresses with the most blocked web traffic — visualizes the 10 most common IP addresses from which traffic blocked by the KWTS system originated. • KWTS. Top-10 IP addresses with the most allowed web traffic — visualizes the 10 most common IP addresses from which traffic allowed by the KWTS system originated. • KWTS. Top 10 requests by client application — visualizes the 10 most common applications used to gain access to network resources, as detected by the KWTS system. • KWTS. Top-10 blocked URLs — visualizes the 10 most common URLs from which traffic was allowed by the KWTS system. • KWTS. System action types — visualizes the 10 most common actions performed by the KWTS system. • KWTS. Top-10 users with the most allowed web traffic — visualizes the 10 most common user names of users whose traffic was allowed by the KWTS system.
[OOTB] KSMG files and hashes*	• KSMG. Top-5 blocked hashes — visualizes the 5 most common file hashes in email messages blocked by the KSMG system. • KSMG. Top-5 net-transferred hashes — visualizes the 5 most common "clean" file hashes in email messages tracked by the KSMG system. • KSMG. Top-5 clean file names — visualizes the 5 most common "clean" file names in email messages tracked by the KSMG system. • KSMG. Top-5 blocked files — visualizes the 5 most common file names in email messages blocked by the KSMG system.
[OOTB] KSMG rules and URLs*	• KSMG. Top-5 rules — visualizes the 5 most common triggered rules of the KSMG system. • KSMG. Top-5 URLs — visualizes the 5 most common domains from links in email messages tracked by the KSMG system.
[OOTB] KSMG results*	• KSMG. All results in the last 24 hours — visualizes the hour-by-hour distribution of actions performed by the KSMG system during the last 24-hour period. • KSMG. Top-5 results — visualizes the 5 most common actions performed by the KSMG system.
[OOTB] KSMG e-mail subjects and accounts*	• KSMG. Top-5 e-mail subjects — visualizes the 5 most common subjects of email messages tracked by the KSMG system. • KSMG. Top-5 source accounts — visualizes the 5 most common sender accounts of email messages tracked by the KSMG system. • KSMG. Top-5 destination accounts — visualizes the 5 most common recipient accounts of email messages tracked by the KSMG system.

*Dashboards are available starting from LogAnalyzer 3.4.1. Widgets will correctly display information when using the "[OOTB] KSMG 2.1+ syslog CEF" normalizer.

Universal dashboards

A dashboard in LogAnalyzer is considered universal if it is created in the Shared tenant. Dashboards created in other tenants are non-universal.

Universal dashboard

The universal dashboard has the following properties:

- The universal dashboard can be viewed by all user roles and in all tenants.
- Data for tenants is displayed in accordance with the general tenant filter.
- Data in widgets on events from the database is obtained in accordance with the space settings configured for the user.

To manage the universal dashboard, the user needs the following roles:

- For creating, configuring, and deleting, the Manage shared resources role is needed. This role rights grant access to all shared resources.
- The Read shared resources role allows viewing the list of universal dashboards. With this role, the user gets read access to all shared resources in the **Resources** section.

  If you have a role that grants access to universal dashboards, you can view the list of universal dashboards in the **Dashboards** and **Resources** sections, but you do not have access to other resources of the Shared tenant.

To view up-to-date data in universal dashboards, you need to subscribe to changes of such dashboards. You can subscribe to all dashboards in the Shared tenant, including the dashboards created by you.

To subscribe to a dashboard in the Shared tenant:

1. In the **Dashboards** section, in the list of dashboards, select and open a dashboard in the Shared tenant.
2. Click the menu ⋮ icon in the upper-right corner of the open dashboard.
3. In the menu, select **Subscribe on changes**.

   If necessary, you can unsubscribe by selecting **Unsubscribe from changes** in the list of actions.

If a user's role involves managing CII resources, the display of information on the dashboard takes into account the CII attribute.

Non-universal dashboard

A non-universal dashboard is created in any LogAnalyzer tenant except Shared. Such dashboards have the following properties:

- The dashboards can be viewed by users that have access to the tenant in which the dashboard was created.
- Only users that have access to the tenant in which the dashboard was created have access to the data of the widgets in the dashboard.
- If the dashboard has the CII attribute, the data on the dashboard is available to users whose rights allow access to CII.

Exporting dashboards

To export a dashboard:

1. In the **Resources** section, go to the **List** tab.

   This opens a window with a list of all resources available to you in LogAnalyzer.

   If necessary, you can configure the display of resources on existing dashboards by selecting **Dashboard** in the **Resource kind** column.

2. Use the check box to select the dashboard to be exported.

3. Click the **Export** button above the table of resources.

4. In the sidebar that is displayed:

   a. Enter the password for the file with exported resources. The password must satisfy the following requirements:

      - At least 16 characters.
      - At least one lowercase character.
      - At least one uppercase character.
      - At lease one numeral.
      - At least one special character.

   b. In the **Tenant** drop-down list, select the tenant from which you want to export the selected resources.

   c. Select check boxes to specify the resources that you want to export.

      If you select the **All resources** check box, all resources in folders of the selected tenant are exported.

5. Click the **Export** button.

   The exported file is saved to the downloads folder of your browser.

Consider the following when exporting dashboards:

- Dashboards are exported without tenants, storage, and space information because these parameters are configured individually for each LogAnalyzer deployment.
- Dashboard widgets are also exported without tenant, storage, and space information.
- The parameters of the enrich and lookup SQL functions in widget queries are exported unchanged. If necessary, you can modify these in accordance with your requirements after exporting resources.
- Links to active lists and tables are not exported. If necessary, you can customize these widgets after exporting resources.

Importing dashboards

To import a dashboard:

1. In the **Resources** section, go to the **List** tab.

   This opens a window with a list of all resources available to you in LogAnalyzer.

   If necessary, you can configure the display of resources on existing dashboards by selecting **Dashboard** in the **Resource kind** column.

2. Use the check box to select the dashboard to be imported.
3. Click the **Import** button above the table of resources.
4. In the sidebar, in the **Tenant** drop-down list, select the tenant to which you want to import the dashboard.
5. Select the import source: **File** or **Repository**.
6. Depending on the selected import source, follow these steps:

   a. If you selected **File** as the import source, follow these steps:

      - Specify the password for the imported file.
      - Click the **Select file** button to select the file that you want to import.
      - Select the check boxes of the resources that you want to import.

   b. If you selected **Repository** as the import source, select one or more repository packages in the displayed table.

7. Click the **Import** button.
8. If necessary, resolve conflicts between the resources imported from the file and the existing resources.

Consider the following when importing dashboards:

- If the dashboard contains an **Events** widget, we recommend importing the dashboard into the **Shared** tenant. This makes the dashboard universal.
- Importing a dashboard into the Shared tenant requires the **Manage shared resources** role. To view such a dashboard, you need to subscribe to the dashboard.
- If the dashboard is imported into the **Shared** tenant, all storages are automatically specified in the dashboard because access to these is governed by user settings.
- If a dashboard is imported into any tenant other than **Shared**, the tenant of the imported dashboard is specified as the resource tenant, the data source tenant, and the storage tenant. After importing resources, you need to specify spaces for widgets with events and if necessary, expand the list of tenants.
- The parameters of the enrich and lookup SQL functions in widget queries are imported unchanged. If necessary, you can modify these in accordance with your requirements after importing resources.
- Links to active lists and tables are not imported. If necessary, you can customize these widgets after importing resources.
- If the dashboard contains the **Active lists** or the **Context tables** widget, we recommend duplicating such dashboards and editing the settings of the duplicates to adapt them to your infrastructure: specifying a correlator, selecting active lists and context tables.

Reports

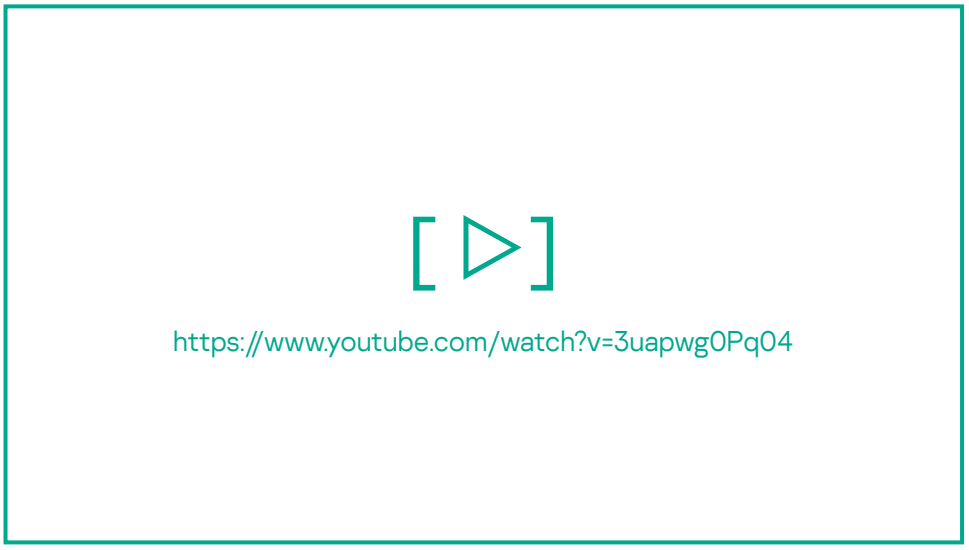
You can schedule the generation of process and object status reports in LogAnalyzer.

Reports are generated using *report templates*. You can find report templates in the **Reports** section on the **Templates** tab.

You can find the generated reports in the **Reports** section on the **Generated reports** tab.

To save the generated reports in HTML and PDF formats, install the required packages on the device with the LogAnalyzer Core.

When deploying LogAnalyzer in a high availability version, the time zone of the Application Core server and the time in the user's browser may differ. This difference is manifested by the discrepancy between the time in reports generated by schedule and the data that the user can export from widgets. To avoid this discrepancy, it is recommended to configure the report generation schedule to take into account the difference between the users' time zone and UTC.



https://www.youtube.com/watch?v=3uapwg0Pq04

Report template

Report templates are used to specify the conditions and data that a report is to be based on, as well as the reporting period. Users with the General administrator, Tenant administrator, Tier 2 analyst, and Tier 1 analyst roles can create, edit, or delete report templates. Reports that were generated using report templates are displayed in the **Generated reports** tab.

Report templates are available in the **Templates** tab of the **Reports** section, where the table of existing templates is displayed. The table contains the following information:

- **Name** is the name of the report template.

  You can sort report templates by clicking the sort icon ⌄ in the column header and selecting **Ascending** or **Descending**.

  You can also search for report templates by name using the **Search** field above the table.

  > Regular expressions are used when searching for report templates.

- **Tenant**—the name of the tenant that owns the report template.

- **Updated** is the date and time of the last update of a resource template.

  You can sort the table by this column by clicking the sort icon ⌄ in the column header and selecting **Ascending** or **Descending**.

- **Created by**—the name of the user who created the report template.

- **Description** is optional information about the report template.

- **Tags** to help find the report template. You can search for report templates by clicking the sort icon ⌄ in the column header, by specifying a value to search for in the **Search** field, or by selecting tags using check boxes.

- **Schedule**—the rate at which reports must be generated using the template. If the report schedule was not configured, the `disabled` value is displayed.
- **Last report**—the date and time when the last report was generated based on the report template.
- **Package name** is the name of the package of tools for processing events.
- **Send by email** indicates that email notification is configured for generated reports.

If necessary, you can customize the columns by clicking the gear icon.

After selecting a report template in the list, you can click the menu icon to open a drop-down list of available actions:

- **Show dependencies** opens the dependency graph window for the selected report template. If necessary, in the graph window, you can download dependency data in SVG format and configure filters to be applied to dependencies displayed on the graph.
- **View versions history** displays the version history of a report template. If necessary, in the sidebar, you can compare the versions and restore a previous version.
- **Run report** generates a report. The generated reports are displayed on the **Generated reports** tab.
- **View all generated reports** displays a list of all reports generated for the selected report template. If necessary, you can save the selected report in one of the supported formats or delete the report.
- **Edit schedule** lets you configure the reporting schedule and select users that you want to receive email notifications about generated reports.

You can also view the list of report templates available to you in the **Resources** → **List** section. To do this, select **Report template** in the **Resource kind** column.

Creating report template

To create report template:

1. Open the LogAnalyzer web interface and go to one of the following sections:
   - **Reports** → **Templates**
   - **Resources** → **List** If you have selected this section, for the convenience of viewing objects, you can select **Report template** in the **Resource kind** column of the table.
2. Depending on the selected section, do one of the following:
   - In the **Reports** → **Templates** section, click the **Create new** button above the table of report templates.
   - In the **Resources** → **List** section, click the **Create new** button and select **Report template** in the list of resources.
3. This opens a window; in that window, enter a unique name for the report template. The name must contain 1 to 128 Unicode characters.
4. In the **Template tenant** drop-down list, select the tenant that will own the layout being created.

5. In the **Collect data from tenants** drop-down list, select one or more tenants. Reports are generated based on data of the selected tenants.

6. If necessary, in the **Time period** drop-down list, select the time period from which you want to get analytics in one of the following ways:

   - If you want to specify an exact date, in the calendar on the left, select the start and end date of the period and click **Apply**.

     You can select a date up to and including the current date. The date and time format depends on your browser settings. If the **Range from** or **Range to** field has a value and you have not edited the time value manually, when you select a date in the calendar, the **Range from** field is automatically populated with 00:00:00.000, and the **Date to** field with 23:59:59.999. If you have manually deleted the value in the **Range from** or **Range to** field, when you select a date in the calendar, the field is automatically populated with the current time. After you select a value in one of the fields, the focus switches to the other field. If your **Range to** is earlier than your **Range from**, this earlier value is automatically inserted into the **Range from** field.

   - If you want to specify a relative period, select one of the available periods in the **Relative period** list on the right.

     The period is calculated relative to the current time.

   - If you want to specify a custom period, edit the value of the **Range from** and **Range to** fields.

     You can enter an exact date and time in the **DD.MM.YYYY HH:mm:ss.SSS** format for the Russian localization and **YYYY-MM-DD HH:mm:ss.SSS** for the English localization or a period relative to the current time as a formula. You can also combine these methods if necessary. If you do not specify milliseconds when entering the exact date, 000 is substituted automatically. If you have edited the time in the **Range from** or **Range to** fields, picking a date in the calendar does not change the time component.

     In the relative period formulas, you can use the **now** parameter for the current date and time and the interval parameterization language: **+** (only in the **Range to** field), **-**, **/** (rounding to the nearest), as well as time units: **y** (year), **M** (month), **w** (week), **d** (day), **h** (hour), **m** (minute), **s** (second). For example, you can specify the period **now-5d** to get data for the last five days, or **now/w** to get data from the beginning of the first day of the current week (00:00:00:000 UTC) to the current time (**now**).

     > The **Range from** field is required, and its value cannot exceed the value of the **Range to** field, and also cannot be earlier than 1970-01-01 (if specifying an exact date or a relative period). The **Range to** cannot be earlier than the **Range from**. If you do not specify a value in the **Range to** from field, **now** is specified automatically.

   By default, the **1 day** (**now-1d**) relative period is selected. The bounds of the period are inclusive: for example, for the **Today** time range, events are displayed from the beginning (00:00:00:000) of the current day to the current time (**now**) inclusive, and for the **Yesterday** period, events are displayed from the beginning (00:00:00:000) of the previous day to 00:00:00:000 of the current day.

   LogAnalyzer stores time values in UTC, but in the user interface time is converted to the time zone of your browser. This is relevant to the relative periods: **Today**, **Yesterday**, **This week**, and **This month**. For example, if the time zone in your browser is UTC+3, and you select **Today** as the data display period, data will be displayed for the period from 03:00:00.000 until now, not from 00:00:00.000 until now.

   If you want to take your time zone into account when selecting a relative data display period, such as **Today**, **Yesterday**, **This week**, or **This month**, you need to manually add a time offset in the **Range from** and **Range to** (if a value other than **now** is specified) by adding or subtracting the correct number of hours. For example, if your browser's time zone is UTC+3 and you want to display data for **Yesterday**, you need to change **Range from** to **now-1d/d-3h** and **Date to** to **now/d-3h**. If you want to display data for the **Today** period, you only need to change the value in the **Range from** field to **now/d-3h**.

   > If you need results up to 23:59:59:999 UTC of yesterday, you can use an SQL query with a filter by Timestamp or specify an exact date and time.

7. If necessary, in the **Retention period** field, specify how long you want to store reports that are generated according to this template.

   The default value is 3 months.

8. If necessary, select tags for the report template from the **Tags** drop-down list.

9. If necessary, add a description of the report template.

10. Click **Save**.

11. This opens a window; in that window, you can change the logo of the report template by clicking the **Upload a file** link. In the file selection window, select the image file for the logo. The format of the image must be JPEG, PNG, or GIF, and its size may not exceed 3 MB.

    The added logo is displayed in the report instead of LogAnalyzer logo.

12. Click the **Add widget** button and select a widget. This opens a window; in that window, configure the selected widget.

    You can add multiple widgets. You can drag widgets around the window by the handle icon ⠿ in the upper-left corner of the widget. If you need to resize the widget, you can do so using the resize icon ⤡ that is displayed when you hover over the lower-right corner of the widget.

    The following limitations apply to widgets with the **Pie chart**, **Bar chart**, **Line chart**, **Counter** , and **Date Histogram** chart types:

    - In `SELECT` queries, you can use extended event schema fields of `String`, `Number`, and `Float` types.
    - In `WHERE` queries, you can use all types of extended event schema fields (`String`, `Number`, `Float`, `Array of strings`, `Array of numbers`, and `Array of floats`).

    For widgets with the **Table** chart type, in `SELECT` queries, you can use all types of extended event schema fields (`String`, `Number`, `Float`, `Array of strings`, `Array of numbers`, and `Array of floats`).

    By clicking the menu ⋮ icon in the upper-right corner of the widget, you can do the following:

    - Manage the settings of widgets
    - Create copies of widgets
    - Delete widgets
    - Download widget data in CSV format

13. Click **Create**.

    The new report template is displayed on the **Reports** → **Templates** and **Resources** → **List** tabs of the LogAnalyzer web interface. You can run this report manually. If you want to have the reports generated automatically, you need to configure a schedule.

Configuring report schedule

To configure the report schedule:

1. Open the LogAnalyzer web interface and select **Reports** → **Templates**.

2. In the table with the list of report templates, select a report template and click the menu icon ⋮.

3. In the drop-down list, select **Edit schedule**.

   This opens the **Edit schedule** window.

4. If you want the report to be generated regularly:

   a. Enable the **Send report by schedule** toggle switch.

   b. Under **Recur every**, specify how often the reports must be generated.

      You can specify the frequency of generating reports by days, weeks, months, or years. Depending on the selected period, you should specify the time, day of the week, day of the month or the date of the report generation.

   c. In the **Time** field, enter the time when the report must be generated.

   d. In the **Timezone** drop-down list, select the time zone for the report generation time.

5. Under **Notification settings**, specify the following:

   a. In the **Notification template** drop-down list, select a notification form that you want to be sent to the user on a schedule.

   b. In the **Send to** group of settings, click **Add**.

   c. This opens a window; in that window, add a group of users that you want to get the report and select a format. To do so:

      - In the **Emails** drop-down list, select email addresses of report recipients.
      - In the **Format** drop-down list, select the format of the report. The following formats are available: HTML, PDF, CSV, Split CSV, Excel.

      You can add multiple groups of report recipients by clicking the **Add group** button.

      Reports are sent to the specified addresses every time you generate a report manually or KUMA generates a report automatically on schedule.

      > If you want the generated reports to be sent by email, you need to configure an SMTP connection.

      If recipients of an emailed report are KUMA users, they can download or view the report by clicking links in the email message. If the recipients are not KUMA users, they can follow the links but cannot log in to KUMA, so only attachments are available to them.

      We recommend viewing HTML reports by clicking links in the web interface, because at some screen resolutions, the HTML report from the attachment may not be displayed correctly.

      If you send an email message without attachments, the recipients can only get reports by following links, and only if they can authorize in KUMA, without restrictions on roles or tenants.

   d. Click **Add**.

6. Click **Save**.

Report schedule is configured.

Editing report template

To edit report template:

1. Open the LogAnalyzer web interface and go to one of the following sections:

   - **Reports** → **Templates**
   - **Resources** → **List** If you have selected this section, for the convenience of viewing objects, you can select **Report template** in the **Resource kind** column of the table.

2. Depending on the selected section, do one of the following:

   - In the **Reports** → **Templates** section, click the name of the report template in the table of report templates.
   - In the **Resources** → **List** section, click the edit settings icon next to the name of the report template in the list of resources.

   This opens the **Edit template** window.

   > You can also open this window on the **Reports** → **Generated reports** tab by selecting an existing report and clicking the **Edit template** button.

3. Make the necessary changes:

   - Change the name of the report template.
   - Modify the list of tenants that send data for the report.
   - Click the **Settings** button to change the time period for getting analytical data, the retention period, tags, and description of the report template.
   - Click the **Add widget** button to add a widget to the report template.
   - Click the menu icon (⋮) to view the list of reports generated by this template, or to delete the report template.
   - Change the report logo by clicking the **Upload a file** link and uploading a new logo. If the template already contains a logo, you must first delete it.
   - Change widgets positions by dragging them.
   - Resize widgets using the resize (⤡) icon that appears when you hover over a widget.
   - Click the menu icon (⋮) on the widget to manage widget settings, copy widgets, delete widgets, or download widget data in CSV format.

4. Click **Save**.

The report template is modified and displayed on the **Reports** → **Templates** and **Resources** → **List** tabs of the LogAnalyzer web interface.

Copying report template

To create a copy of a report template:

1. Open the LogAnalyzer web interface and go to one of the following sections:
   - **Reports** → **Templates**
   - **Resources** → **List** If you have selected this section, for the convenience of viewing objects, you can select **Report template** in the **Resource kind** column of the table.
2. In the table, select a report template and click the **Duplicate** button above the table.
3. This opens a window; in that window, enter the name of the report template. When you copy a report template, the name of the report mentions that it is a copy.
4. In the **Template tenant** drop-down list, select the tenants that will own the report template being created.
5. In the **Collect data from tenants** drop-down list, select one or more tenants whose data you want to be represented in the report template.
6. Change the following settings as needed:
   - Time interval for generating the report.
   - Report retention period.
   - Tags of the report template.
   - Description of the report template.
   - Generation of report data with CII information.
7. Click **Save**.

The copy of the report template is created and is displayed on the **Reports** → **Templates** and **Resources** → **List** tabs of the LogAnalyzer web interface.

Deleting report template

To delete report template:

1. Open the LogAnalyzer web interface and go to one of the following sections:

   - **Reports** → **Templates**

   - **Resources** → **List** If you have selected this section, for the convenience of viewing objects, you can select **Report template** in the **Resource kind** column of the table.

2. In the table of report templates, select the check box of a report template and click the **Delete** button above the table.

   A confirmation window opens.

3. Do one of the following:

   - If you want to delete only the report template, click the **Delete** button.

   - If you want to delete a report template and all the reports that were generated using that template, click the **Delete with reports** button.

The report template is deleted.

Exporting or importing a report template

To export a report template:

1. Open the LogAnalyzer web interface and go to one of the following sections:

   - **Reports** → **Templates**

   - **Resources** → **List** If you have selected this section, for the convenience of viewing objects, you can select **Report template** in the **Resource kind** column of the table.

   This opens a window with a list of all report templates available to you in LogAnalyzer.

2. Use the check box to select the report template to be exported.

3. Click the **Export** button above the table of report templates.

4. In the sidebar that is displayed:

   a. Enter the password for the file with exported resources. The password must satisfy the following requirements:

      - At least 16 characters.
      - At least one lowercase character.
      - At least one uppercase character.
      - At lease one numeral.
      - At least one special character.

   b. In the **Tenant** drop-down list, select the tenant from which you want to export the selected resources.

   c. Select check boxes to specify the resources that you want to export.

      If you select the **All resources** check box, all resources in folders of the selected tenant are exported.

5. Click the **Export** button.

Consider the following when exporting report templates:

- Report templates are exported without tenants, storage, and space information because these parameters are configured individually for each LogAnalyzer deployment.
- Report template widgets are also exported without tenant, storage, and space information.

To import a report template:

1. Open the LogAnalyzer web interface and go to one of the following sections:

   - **Reports** → **Templates**
   - **Resources** → **List** If you have selected this section, for the convenience of viewing objects, you can select **Report template** in the **Resource kind** column of the table.

   This opens a window with a list of all report templates available to you in LogAnalyzer.

2. Use the check box to select the report template to be imported.
3. Click the **Import** button above the table of resources.
4. In the sidebar, in the **Tenant** drop-down list, select the tenant to which you want to import the report template.
5. Select the import source: **File** or **Repository**.

6. Depending on the selected import source, follow these steps:

   a. If you selected **File** as the import source, follow these steps:

      - Specify the password for the imported file.
      - Click the **Select file** button to select the file that you want to import.
      - Select the check boxes of the resources that you want to import.

   b. If you selected **Repository** as the import source, select one or more repository packages in the displayed table.

7. Click the **Import** button.

8. If necessary, resolve conflicts between the resources imported from the file and the existing resources.

Consider the following when importing report templates:

- If a report template is imported into any tenant, the tenant of the imported report template is specified as the resource tenant and the data source tenant.
- After importing resources, you need to specify spaces for widgets with events and if necessary, expand the list of tenants.

Universal report templates

A report template in LogAnalyzer is considered universal if it is created in the Shared tenant. Report templates created in other tenants are non-universal.

To manage a universal report template, the user needs the following roles:

- For creation, management, or deletion, the Manage shared resources role is needed. This role rights grant access to all shared resources.

  The CII attribute can be set by any user with the Manage shared resources role if the user's role involves managing CII resources.

- The Read shared resources role allows viewing the list of universal report templates. With such rights, the user gains read access to all shared resources.

Widgets in a universal report template are displayed based on the following:

- Widget data for tenants is displayed according to the user's access to tenants, the global tenant filter, and the configuration of the report template or widget.
- You can view the widget data when managing report template settings by clicking the **Update preview** button.
- Event widget data is received from the database in accordance with the space set settings of the user. If, when generating a report, it turns out that the user does not have access to the spaces of one or more tenants, LogAnalyzer generates an empty report and displays a message indicating that other results cannot be provided because of the lack of access to the relevant spaces.

- Widge data in the report can include or exclude CII. For access to reports in which data includes CII, the user must have the Interaction with CII role.
- Limitations for active list and context table widgets apply to universal reports.

Generated reports

All reports are generated using report templates. Generated reports are available on the **Generated reports** tab in the **Reports** section and are displayed in the table with the following information:

- **Name**—the name of the report template.

  You can sort report templates by clicking the sort icon ⌄ in the column header and selecting **Ascending** or **Descending**.

  You can also search for report templates by name using the **Search** field above the table.
- **Time period**—the time period for which the report analytics were extracted.
- **Last report**—date and time when the last report was generated.

  You can sort the table by this column by clicking the sort icon ⌄ in the column header and selecting **Ascending** or **Descending**.
- **Tenant**—name of the tenant that owns the report.
- **User**—name of the user who generated the report manually. If the report was generated by schedule, the value is blank. If the report was generated in LogAnalyzer lower than 2.1, the value is blank.

If necessary, you can customize the columns by clicking the gear ⚙ icon.

After selecting a report in the list of generated reports, you can:

- Create a report.
- Open and view a report.
- Save a generated report in a certain format. Available formats: HTML, PDF, CSV, split CSV, Excel. By default, 250 rows are displayed in all formats. The maximum number of table rows in PDF and HTML formats is 500. If you want to output more than 500 rows in a report, set your value for the LIMIT parameter in the SQL query and save the report in CSV format.
- Manage the settings of a report template.
- Delete a report.

Viewing reports

To open report:

1. Open the LogAnalyzer web interface and select **Reports** → **Generated reports**.
2. In the table of reports, click the name of the generated report.

   This opens a window with widgets that display the analytics of the reports. Clicking the widget title opens the corresponding section of the LogAnalyzer web interface with an active filter and/or search query, which displays data from the widget. Default values are applied to widgets.

   To view the full data, download the report in the CSV format with the specified settings from the request.

   If you have enabled the **Show total** toggle switch in the widget settings, the widget displays the **Total** value, but when you attach or upload a report in CSV or extended CSV format, the **Total** value is not displayed. To display the **Total** value in the report, you can use Excel formulas or copy and paste the total value from the widget, or use PDF or HTML attachments.
3. If necessary, save the report in the selected format by clicking the **Save as** button.

Generating reports

You can generate report manually or configure a schedule to have it generated automatically.

To generate report manually:

1. Open the LogAnalyzer web interface and go to the **Reports** section.
2. Do one of the following:

   a. On the **Generated reports** tab:

   - Select the check box of a report in the table of reports.
   - Click the **Generate report** button above the table.

   b. On the **Templates** tab:

   - Select a report in the table of report templates by selecting its check box.
   - Click the menu icon ⋮ and select **Generate report** from the drop-down list.

Within 15 seconds, the report is generated and displayed on the **Reports** → **Generated reports** tab.

You can also configure a reporting schedule for automatic generation of reports.

Saving reports

To save the report in the desired format:

1. Open the LogAnalyzer web interface and select **Reports** → **Generated reports**.
2. Select the check box of a generated report in the table of generated reports.
3. In the **Save as** drop-down list above the table, select a format: HTML, PDF, CSV, Split CSV, Excel.
   The report is saved to the download folder configured in your browser.

You can also save the report in the desired format when you view it.

Updating reports

To update a generated report:

1. Open the LogAnalyzer web interface and go to the **Reports** → **Generated reports** section.
2. In the table of generated reports, select a report.
3. Click the **Generate report** button above the table.
   The report is updated within 15 seconds and becomes available in the **Reports** → **Generated reports** section.

Deleting reports

To delete report:

1. Open the LogAnalyzer web interface and select **Reports** → **Generated reports**.
2. In the table of generated reports, select a report.
3. Click the **Delete** button above the table.
   A confirmation window opens.
4. Click **Delete**.

The selected report is deleted.

Widgets

Widgets let you monitor the operation of the application. LogAnalyzer offers widgets of the **Events type, that is,** widgets for creating event-based analytics.

> In the events table, in the event details area and in the widgets, the names of assets, accounts, and services are displayed instead of the IDs as the values of the `SourceAssetID`, `DestinationAssetID`, `DeviceAssetID`, `SourceAccountID`, `DestinationAccountID`, `ServiceID` fields. When exporting events to a file, the IDs are saved, but columns with names are added to the file. The IDs are also displayed when you point the mouse over the names of assets, accounts, or services.
>
> Searching for fields with IDs is only possible using IDs.

Basics of managing widgets

The principle of data display in the widget depends on the type of the graph. The following graph types are available in LogAnalyzer:

- Pie chart ( ).
- Counter ( ).
- Table ( ).
- Bar chart ( ).
- Date Histogram ( ).
- Line chart
- Stacked Bar chart
- Speedometer chart.

Basics of general widget management

The name of the widget is displayed in the upper left corner of the widgets. You can click the link with the widget name to go to the corresponding section of the LogAnalyzer web interface.

A list of tenants for which data is displayed is located under the widget name.

In the upper right corner of the widget, the period for which data is displayed on the widget is indicated (for example, 30 days 30д). Keep in mind that the data displayed in the dashboard may lag behind real time because of caching. You can view the date and time of the last update by hovering over the period icon.

If the **Show data for previous period** setting is enabled for the widget, and the widget is displaying data for a relative period, the tooltip also displays the previous period. The previous period is calculated relative to the current period as start and end values of the current period minus the duration of the current period. For example, if data is updated daily and displayed for a month, but only the first 10 days of the month have passed, the previous period is taken to be the last 10 days of the previous month.

You can change the data display period for the widget by clicking the period icon and selecting an exact date or a relative period in the window that is displayed. If you want the widget to display data for the period selected for the layout, click the **Reset** button. Changing the displayed period on the layout also changes the period displayed in the widget.

The time in the widget is displayed in the local time zone set in the browser.

The **CSV** button is located to the left of the period icon. You can download the data displayed on the widget in CSV format (UTF-8 encoding). The downloaded file name has the format <widget name>_<download date (YYYYMMDD)>_<download time (HHMMSS)>.CSV.

The widget displays data for the period selected in widget or layout settings only for the tenants that are selected in widget or layout settings.

If you have enabled the **Show total** toggle switch in the widget settings, the widget displays the **Total** value, but when you download the widget in CSV format, the **Total** value is not displayed. To display the **Total** value in the report, you can use Excel formulas or manually copy the total value from the widget.

Basics of managing Pie chart graphs

A pie chart with a legend is displayed below the list of tenants. The legend lists the names of the categories corresponding to the sectors of the diagram. By left-clicking a category in the legend, you can go to the section of the LogAnalyzer web interface with the corresponding data. The data in that section is sorted in accordance with the filters and/or search query specified in the widget.

When going from the legend of the widget to the events details section, pay attention to the text of the legend item that you click to go to the events section. Any \ or ' characters used in the text are escaped in the text of the SQL query. However, the / character is not escaped.

Under the period icon, you can see the number of events grouped by the selected criteria for the data display period.

Example:
In the **Events** widget, for which the SQL query `SELECT count(ID) AS `metric`, Name AS `value` FROM `events` GROUP BY Name ORDER BY `metric` DESC LIMIT 10` is specified, 10 events are displayed below the period icon, grouped by name and sorted in descending order.

If you want to view events with specific names in the legend, you can clear the check boxes to the left of the names of events that you do not want to see in the legend.

Basics of managing Counter graphs

Graphs of this type display the sum total of selected data.

Basics of managing Table graphs

Graphs of this type display data in a table format.

Example:
In the **Events** widget, for which the SQL query `SELECT TenantID , Timestamp , Name , DeviceProduct , DeviceVendor FROM `events` LIMIT 10` is specified, displays an event table with **TenantID**, **Timestamp**, **Name**, **DeviceProduct**, and **DeviceVendor** columns. The table contains 10 rows.

You can customize the display of data in the table using two-color (red and green) or three-color (red, yellow, and green) gradient fill. After selecting the type of gradient, you can adjust the direction of the fill by clicking the toggle direction icon (⇆) and set thresholds. Two values must be specified for a two-color gradient, and three for a three-color gradient. These values are displayed in the legend and determine the color of the marker added to the cell of the selected column. The column is specified in the **Field for gradient indicator** drop-down list.

Basics of managing Bar chart graphs

A bar chart with a legend is displayed below the list of tenants. The legend lists the names of the categories corresponding to the bars of the diagram. By left-clicking a category in the legend, you can go to the section of the LogAnalyzer web interface with the corresponding data. The data in that section is sorted in accordance with the filters and/or search query specified in the widget.

When going from the legend of the widget to the events details section, pay attention to the text of the legend item that you click to go to the events section. Any \ or ' characters used in the text are escaped in the text of the SQL query. However, the / character is not escaped.

You can also customize the display of bars using two-color (red and green) or three-color (red, yellow, and green) gradient fill. After selecting the type of gradient, you can adjust the direction of the fill by clicking the toggle direction icon (⇆) and set thresholds. Two values must be specified for a two-color gradient, and three for a three-color gradient. These values are displayed in the legend and determine the coloring of the bars.

Basics of managing Date Histogram graphs

A Date Histogram with a legend is displayed below the list of tenants. The legend lists the names of the categories corresponding to the bars of the diagram. By left-clicking a category in the legend, you can go to the section of the LogAnalyzer web interface with the corresponding data. The data in that section is sorted in accordance with the filters and/or search query specified in the widget.

When going from the legend of the widget to the events details section, pay attention to the text of the legend item that you click to go to the events section. Any \ or ' characters used in the text are escaped in the text of the SQL query. However, the / character is not escaped.

If, when creating a widget with a **Date Histogram** type, the **Bar chart** view is selected, you can customize the display of the bars using two-color (red and green) or three-color (red, yellow, and green) gradient fill. After selecting the type of gradient, you can adjust the direction of the fill by clicking the toggle direction icon (⇆) and set thresholds. Two values must be specified for a two-color gradient, and three for a three-color gradient. These values are displayed in the legend and determine the coloring of the bars.

Example:

```
SELECT count(ID) AS `metric`, Timestamp AS `value`
FROM `events`
GROUP BY Timestamp
ORDER BY `metric` DESC
LIMIT 250
```

The X axis of the chart shows the date when the event was created, and the Y axis shows the number of events. The bars of the chart are arranged from left to right in descending order of the number of events.

If an additional field with the `category` alias is used in the SQL query, the bar chart is displayed as a stacked bar chart. In this case, gradient fill is not applied.

Example:

```
SELECT count(ID) AS `metric`, Timestamp AS `category`, Type AS `value`
FROM `events`
GROUP BY Timestamp, Type
ORDER BY metric DESC
LIMIT 250
```

The X axis indicates the time intervals, and the Y axis indicates the number of events. Each segment of the bar corresponds to a value of the field marked as `Type`.

If, when creating a widget with a **Date Histogram** type, the **Line chart** view is selected, there are two options for displaying the chart: single line (if only the `metric` and `value` fields are specified in the query) and multiple lines (if an additional field with the `category` alias is used). This lets you compare values for different groups, depending on which field is used for grouping in the query. For example, users or hosts can be used. No gradient fill is applied to such charts.

For convenience, we recommend using supported ClickHouse functions to round the values in the time field (`Timestamp`), for example, to 1 hour or 1 day.

For a **Date Histogram** widget with the **Line chart** view, you can also configure the display of the average value. To do this, on the **Appearance** tab, for the **Regression** setting, select **LOESS**.

Basics of managing Line chart graphs

A line chart with a legend is displayed below the list of tenants. The legend displays the names of the categories corresponding to the lines on the chart. By left-clicking a category in the legend, you can go to the section of the LogAnalyzer web interface with the corresponding data. The data in that section is sorted in accordance with the filters and/or search query specified in the widget.

When going from the legend of the widget to the events details section, pay attention to the text of the legend item that you click to go to the events section. Any \ or ' characters used in the text are escaped in the text of the SQL query. However, the / character is not escaped.

Example:
In the **Events** widget, for which the SQL query `SELECT count(ID) AS `metric`, SourcePort AS `value` FROM `events` GROUP BY SourcePort ORDER BY `value` ASC LIMIT 250` is specified, the x-axis corresponds to the approximate port number, and the y-axis corresponds to the number of events. The data is grouped by port number in ascending order.

For a **Line chart** widget, you can also configure the display of the average value. To do this, on the **Appearance** tab, for the **Regression** setting, select **Linear**. If the chart displays multiple categories, the average is displayed for each category.

Basics of managing graphs of the Stacked bar chart type

A stacked bar chart with a legend is displayed below the list of tenants. The legend displays the names of the categories that make up parts of each bar. To the left of each category is a check box that lets you hide or show the category. The number of bars in the chart corresponds to the number of values in the selected grouping. The bars have captions. The color of the corresponding category in the bar is assigned automatically. When you hover over the zones of the bars, a tooltip is displayed with the value and a description of the value. You can left-click a category in the legend to go to the section of the LogAnalyzer web interface with the relevant data.

When going from the legend of the widget to the events details section, pay attention to the text of the legend item that you click to go to the events section. Any \ or ' characters used in the text are escaped in the text of the SQL query. However, the / character is not escaped.

The meaning of bar height depends on the **Format** setting:

- If the **Absolute values** format is configured, the height of the bars corresponds to the sum of the values of the measured figure.

- If the **Relative values, %** format is configured, all bars have the same height of 100%, and the relative heights of colored zones on the bars correspond to the ratios of the values.

If, when creating a custom widget based on the stacked bar chart, you selected the **Show data for previous period** option, and the standard `value`, `category`, `metric` aliases are used in the query, the chart displays previous-period data as separate bars. However, if instead of the standard `metric`, the query uses a custom metric calculation with non-standard aliases, the **Show data for previous period** is not taken into account when displaying the chart (see example queries below).

Examples:

When creating a custom widget of the **Stacked Bar chart** type based on an SQL query of an **Events** widget, the following rules apply:

- The field in the query that is marked as the `value` alias (required field) is used in the chart as the field by which the legend is created and the bars are divided into categories.
- The field in the query that is marked as the `category` alias (required field) is used in the chart as the field that defines the arrangement of bars along the X axis.
- The field in the query marked as the `metric` alias is used in the chart as the field that defines how bar areas are counted. In this case, the distribution is performed automatically.

However, you can manage the count by using standard aggregation functions (`sum`, `avg`, `min`, `max`, `count`) and your own arbitrarily named aliases as metrics instead of the standard `metric` alias (in this case, the display of data for the previous period is not supported).

Example 1:

For the **Events** widget, the following SQL query is specified with standard aliases, and the **Show data for previous period** option was selected when creating the widget:

```
SELECT count(ID) AS `metric`,
       Type AS `value`,
       TenantID AS `category`
FROM `events`
GROUP BY value, category
ORDER BY metric DESC
```

The X-axis stands for tenants (the field marked as the `category`), the Y-axis stands for the number of events of a certain type (the field marked as the `value`). Inside each bar, the quantity corresponding to a certain type is represented by a certain color. In the legend, the names of the event types by which the bars are divided are displayed as categories. Each category has an automatically assigned corresponding color on the bar. If you want to view only certain event types in the chart, you can clear or select the check boxes to the left of the corresponding event types in the legend.

Next to each bar, an additional bar is displayed with historical data, if such data was received in the query response.

Example 2:

For the **Events** widget, the following SQL query is marked with custom metrics marked as the `Base` and `Audit` aliases instead of the standard `metric` alias:

```
SELECT SUM(IF (Type = 1,1,0)) AS `Base`,
       SUM(IF (Type = 4,1,0)) AS `Audit`,
       TenantID as `category`
FROM `events`
GROUP BY category
```

The X-axis stands for tenants (the field marked as `category`), the Y-axis stands for the number of events of each type (custom metrics marked as the `Base` and `Audit` aliases). Inside each bar, the quantity corresponding to a certain metric is represented by a certain color. In the legend, the `Base` and `Audit` metrics by which the bars are divided into categories are displayed as category names. Each category has an automatically assigned corresponding color on the bar. If you want to view only event corresponding to a certain metric in the chart, you can clear or select the check boxes to the left of the relevant metric in the legend.

The additional bar with historical data is not displayed for a query with custom metrics, even if the **Show data for previous period** option was selected when creating the widget.

Example 3:

For the **Events** widget, the following SQL query is specified with standard aliases:

```
SELECT count(ID) AS `metric`,
       TenantID as `value`,
       Type as `category`
FROM events
GROUP BY value, category
ORDER BY metric DESC
```

In contrast to the similar query in example 1, in this case, the X-axis stands for the types of events (the field marked as the `category`), and the tenants (the field marked as the `value`) are represented by ranges of values in the bars (along the X axis) and the corresponding captions in the legend.

Example 4:

For the **Events** widget, the following SQL query is specified with standard aliases:

```
SELECT count(ID) AS `metric`,
       CAST(fromUnixTimestamp64Milli(Timestamp) AS DATE) AS `category`,
       Type as `value`
FROM `events`
GROUP BY category, value
ORDER BY category DESC
```

The chart displays the days of the month on the X axis (the field marked as the `category`). The Y axis displays the number of events of a particular type (the field marked as the `value`).

To create a similar chart with bars arranged by date and/or time, use a query with grouping and sorting by the following fields of the `timestamp` type:

- `Timestamp`
- `DeviceCustomDate1`
- `DeviceCustomDate2`
- `EndTime`
- `FileCreateTime`
- `FileModificationTime`
- `FlexDate1`
- `OldFileCreateTime`
- `OldFileModificationTime`
- `DeviceReceiptTime`
- `StartTime`

We recommend using the **Date Histogram** to work with data that is arranged by date and/or time.

Basics of managing Speedometer charts

This type of chart displays the percentage of the number of objects of the selected category vs the total number of objects for a certain period. The **Speedometer chart** can be used for the following objects:

- Assets in a category that satisfy certain criteria vs the total number of assets in the same category or uncategorized assets
- Events that satisfy certain criteria vs the total number of events for a certain period

The value on the chart is displayed as a percentage with a color matching the configured ranges of values. By default, the following ranges are specified: 0–30, 30–70, 70–100.

Example:
In the **Events** widget, for the **Speedometer chart** chart type, the user specifies a tenant, a period for analysis, and the event data storage. The SQL query is displayed as follows:

```
SELECT count(ID) AS `metric`, SourcePort AS `value` FROM `events` GROUP BY SourcePort ORDER BY value ASC LIMIT 250
```

Special considerations for displaying data in widgets

Limitations for the displayed data

For improved readability, LogAnalyzer has limitations on the data displayed in widgets depending on its type:

- Pie chart displays a maximum of 20 slices.
- Bar chart displays a maximum of 40 bars.

- Table displays a maximum of 500 entries.
- Date histogram displays a maximum of 365 days.

Data that exceeds the specified limitations is displayed in the widget in the **Other** category.

You can download the full data used for building analytics in the widget in CSV format.

Summing up the data

The format of displaying the total sum of data on date histogram, bar chart and pie chart depends on the locale:

- English locale: decades (every three digits) are separated by commas, the decimal part is separated by a period.
- Russian locale: decades (every three digits) are separated by spaces, the decimal part is separated by a comma.

Creating a widget

You can create a widget on a dashboard while creating or editing the dashboard.

To create a widget:

1. Create a dashboard or go to edit mode of the selected dashboard.
2. Click **Add widget**.
3. Select a widget type from the drop-down list.
   This opens the widget settings window.
4. Edit the widget settings.
5. If you want to see how the data will be displayed in the widget, click **Preview**.
6. Click **Save** in the widget settings window.
   The widget appears on the dashboard.
7. Click the **Save** button in the dashboard settings window.

Editing a widget

To edit a widget:

1. In the LogAnalyzer web interface, select the **Dashboards** section.
2. Expand the list of dashboards in the upper-left corner of the window.
3. Select a dashboard and click the **Edit** button.
   This opens the dashboard settings window.

4. On the widget that you want to edit, click the menu (⋮) icon.
5. Select **Edit**.
   This opens the widget settings window.
6. Edit the widget settings.
7. Click **Save** in the widget settings window.
8. Click the **Save** button in the dashboard settings window.

The changes are applied to the widget.

Deleting a widget

To delete a widget:

1. In the LogAnalyzer web interface, select the **Dashboards** section.
2. Expand the list of dashboards in the upper-left corner of the window.
3. Select a dashboard and click the **Edit** button.
   This opens the dashboard settings window.
4. On the widget that you want to delete, click the menu icon ⋮.
5. Select **Delete**.
6. This opens a confirmation window; in that window, click **OK**.
7. Click the **Save** button.

The widget is deleted.

Events widget

You can use the **Events** widget to get analytics based on SQL queries.

When creating this widget, you must specify the settings described in the tables below.

General tab

The following table describes the settings on the **General** tab.

Description of parameters

Setting	Description
Graph	Graph type. The following graph types are available: • **Pie chart**. • **Bar chart**. • **Counter**. • **Line chart**. • **Table**. • **Date Histogram**. • **Stacked Bar chart** • **Speedometer chart**
Format	Data display format: **Absolute values** or **Relative values, %**. The setting is available for a **Stacked Bar chart**. If you select the **Absolute values** format, the heights of the bars correspond to the sum of the values of the measured indicator. If you select the **Relative values, %** format, all bars have the same height of 100%, and the relative heights of colored zones on the bars correspond to the ratios of indicator values. By default, **Absolute values** is selected.
Timeline view	Time data display format: **Bar chart** or **Line chart**. This setting is available for charts of the **Date Histogram** type. When you select the **Bar chart** format, the data is displayed as vertical bars by default. If the SQL query contains an additional field with the `category` alias, the chart becomes a stacked bar chart, that is, each bar is divided into stripes in accordance with the values of this field. When you select the **Line chart** format, the data is displayed as one or more (depending on the query) polylines drawn through points on the time axis. The **Bar chart** format is selected by default.
Tenant	The tenant for which data is displayed in the widget. You can select multiple tenants. By default, data is displayed for tenants selected in layout settings.
Time period	Period for which data is displayed in the widget. The default is **As layout**, meaning that data is displayed for the period selected for the layout. You can also specify a period for the widget in one of the following ways: • Select the exact start and end date and time of the period in the calendar and click **Apply**. • Select a period relative to the present time in the **Relative period** list. • Specify a value manually: select an exact date and time or a relative period, or a combination of both. For details, see the *Configuring a period* subsection below.
Show data for previous period	Enable the display of data for two periods at the same time: for the current period and for the previous period. When using a **Stacked Bar chart**, the **Show data for previous period** setting is taken into account if the query contains standard aliases: `value`, `category`, `metric`. However, if instead of the standard `metric`, the query uses a custom metric calculation with non-standard aliases, the **Show data for previous period** is not taken into account.
Defined variables	Local or global variables that you can use to filter widget data. Click **Add**. If you are adding a local variable, enter an arbitrary name for the variable, then enter the value of the variable. If you are adding a global variable, in the list, select the name of the variable that you specified in the dashboard properties.

Setting	Description
Storage	Storage that is searched for events. The list displays the available spaces. You can select only one storage, but you can select one or more spaces. The values in the **Storage** field are independent of the selected tenants in the **Tenant** field. The field displays storages and spaces, like in the **Events** section. When a new space is created in the storage, this new space is not selected by default in the widget settings. You must manually select the new space. If the user has access to all spaces listed in the widget, the widget can display the event information for tenants specified in the space set in the user's permissions. If the user does not have access rights to one or more spaces of the storage, the widget cannot display information; the user cannot edit the widget, but can duplicate the widget using the **Duplicate** button. Duplication does not depend on access rights to spaces. If a template is duplicated in widgets that have spaces specified that are not accessible to the user, the value in the **Storage** field is reset. Such widgets display an error: Access denied (Operation returns no results because of allowed and selected event spaces). To save the template, you need to specify spaces in widgets. In widgets that have spaces that are accessible to the user, the value of the **Storage** field is not reset and is saved when the template is duplicated. When the user downloads the template, the data of inaccessible files are hidden in the downloaded file. If the user's email address is included in the list of recipients of the scheduled report, the user gets the full version of the report, regardless of which spaces are accessible.
SQL query field	Query for filtering and searching for events manually. You can create a query in Builder by clicking **Use sql builder**. You can use extended event schema fields to create a chart of the **Table** type. Example

Setting	Description
	Prerequisite: Events are coming from a FWSaaS system. 10.08.2023 12:12:12 src=10.10.10.10 ReqUrls=[https://google.com, https:/ya.ru] 10.08.2023 12:12:23 src=10.10.10.20 ReqUrls=[https://google.com, https:/yndex.ru] 10.08.2023 12:12:33 src=10.10.10.30 ReqUrls=[https://google.com, https:/ya.ru, https://mail.ru] The LogAnalyzer General administrator has added an extended event schema field in LogAnalyzer in the **Settings** → **Other** → **Advanced event schema fields** section. At the LogAnalyzer normalizer, field mapping is configured for the **ReqUrls** and **SA.urls** fields. In LogAnalyzer, normalized events look as follows: DeviceReceiptTime=10.08.2023 12:12:12 SourceAddress=10.10.10.10 SA.urls=[https://google.com, https:/ya.ru] DeviceReceiptTime=10.08.2023 12:12:23 SourceAddress=10.10.10.20 SA.urls=[https://google.com, https:/yandex.ru] DeviceReceiptTime=10.08.2023 12:12:30 SourceAddress=10.10.10.30 SA.urls=[https://google.com, https:/ya.ru, https://mail.ru] You want to build a "Top URLs for 30 minutes" dashboard. Here is an example of a request in the **Events** section and in widgets: SELECT res.1 AS url, res.2 AS count FROM (SELECT arrayJoin(groupArray((url, count))) AS res FROM (SELECT url, count() AS count FROM (SELECT SA.urls AS urls FROM `events`) ARRAY JOIN urls AS url GROUP BY url) GROUP BY url) Result: url count google.com 3 ya.ru 2 mail.ru 1 yandex.ru 1 LogAnalyzer supports the following functions that allow using arrays in queries: • The `groupArray` function packs all values of the x argument into an array and is used in aggregate functions when grouping data. • The `arrayJoin` function expands the array into separate strings as a result of the query. You can use variables in the SQL query. You can specify variables only when creating the query manually. You cannot use variables in the SQL builder.

Setting	Description

	Example with variables
	
	Example SQL query with a local variable:
	
	`SELECT count(ID) AS 'metric', Name AS 'value', TenantID AS 'category' FROM 'events' WHERE filter(Name = $Name) GROUP BY Name, TenantID ORDER BY metric DESC LIMIT 250`
	
	This query uses the local variable `$Name`. Value of the Name event field are substituted into this variable. You can apply the variable by clicking the funnel icon ( ) on the widget. This opens a window; in that window, you can specify a value for the variable. The widget filters the selection by the selected value of the Name field.
	
	Note that if in the SQL query you specify a variable for a field that takes a numeric value, such a value must be converted to a string. You can do this using the `toString` function. Otherwise, the SQL query returns an error. Example:
	
	`SELECT count(ID) AS `metric`, SourcePort AS `value` FROM `events` WHERE (filter(toString(SourcePort) like $SourcePort)) GROUP BY SourcePort ORDER BY value ASC LIMIT 250`
	
	In this query, the SourcePort event field has a numeric value, therefore it is converted to a string: `toString(SourcePort)`.
	
	For detailed information on creating an SQL query in the query constructor, see below.
	
	The following limitations apply:
	
	• The `metric` and `value` aliases in SQL queries cannot be edited for any type of event analytics widget, except tables.
	• Aliases in widgets with charts of the **Table** type can contain Latin and Cyrillic characters, as well as spaces. When using spaces or Cyrillic, the alias must be enclosed in quotation marks: `"An alias with a space"`, `` `Another alias` ``.
	• `ARRAY JOIN` SQL queries are not supported.
	• When displaying data for the previous period, sorting by the `count(ID)` parameter may not work correctly. We recommend sorting by the `metric` parameter. For example, `SELECT count(ID) AS "metric", Name AS "value" FROM `events` GROUP BY Name ORDER BY metric ASC LIMIT 250`.
	• In widgets with charts of the **Counter** type, you must specify the method of data processing for the values of the `SELECT` function: `count`, `max`, `min`, `avg`, `sum`.
	• In widgets with charts of the **Table** type, the gradient is drawn based on the values of a number or time field (for example, `Timestamp`) specified in the SQL query. This field can be selected from the **Field for gradient indicator** drop-down list (if the query contains multiple number or time fields, any one of these can be selected). After this, the cells of the selected column are highlighted with gradient fill markers. If a field is not present in the query, the **Appearance** tab does not display the **Gradient** and **Field for gradient indicator** fields.
	• In **Stacked Bar chart** widgets, `Timestamp` values are not displayed on the x-axis when using a standard query. If you want to display time along the X axis, we recommend using a **Line chart**. If you need to use a **Stacked Bar chart**, the `Timestamp` value must be converted to string: `toString(Timestamp) AS `category``.

Configuring a period

To configure the data display period, do one of the following:

- If necessary, change the date and time in the **Time period** setting in one of the following ways:

  - If you want to specify an exact date, in the calendar on the left, select the start and end date of the period and click **Apply**.

    You can select a date up to and including the current date. The date and time format depends on your browser settings. If the **Range from** or **Range to** field has a value and you have not edited the time value manually, when you select a date in the calendar, the **Range from** field is automatically populated with 00:00:00.000, and the **Date to** field with 23:59:59.999. If you have manually deleted the value in the **Range from** or **Range to** field, when you select a date in the calendar, the field is automatically populated with the current time. After you select a value in one of the fields, the focus switches to the other field. If your **Range to** is earlier than your **Range from**, this earlier value is automatically inserted into the **Range from** field.

  - If you want to specify a relative period, select one of the available periods in the **Relative period** list on the right.

    The period is calculated relative to the current time.

  - If you want to specify a custom period, edit the value of the **Range from** and **Range to** fields.

    You can enter an exact date and time in the **DD.MM.YYYY HH:mm:ss.SSS** format for the Russian localization and **YYYY-MM-DD HH:mm:ss.SSS** for the English localization or a period relative to the current time as a formula. You can also combine these methods if necessary. If you do not specify milliseconds when entering the exact date, 000 is substituted automatically. If you have edited the time in the **Range from** or **Range to** fields, picking a date in the calendar does not change the time component.

    In the relative period formulas, you can use the **now** parameter for the current date and time and the interval parameterization language: **+** (only in the **Range to** field), **-**, **/** (rounding to the nearest), as well as time units: **y** (year), **M** (month), **w** (week), **d** (day), **h** (hour), **m** (minute), **s** (second). For example, you can specify the period **now-5d** to get data for the last five days, or **now/w** to get data from the beginning of the first day of the current week (00:00:00:000 UTC) to the current time (**now**).

    > The **Range from** field is required, and its value cannot exceed the value of the **Range to** field, and also cannot be earlier than 1970-01-01 (if specifying an exact date or a relative period). The **Range to** cannot be earlier than the **Range from**. If you do not specify a value in the **Range to** from field, **now** is specified automatically.

LogAnalyzer stores time values in UTC, but in the user interface time is converted to the time zone of your browser. This is relevant to the relative periods: **Today**, **Yesterday**, **This week**, and **This month**. For example, if the time zone in your browser is UTC+3, and you select **Today** as the data display period, data will be displayed for the period from 03:00:00.000 until now, not from 00:00:00.000 until now.

If you want to take your time zone into account when selecting a relative data display period, such as **Today**, **Yesterday**, **This week**, or **This month**, you need to manually add a time offset in the **Range from** and **Range to** (if a value other than **now** is specified) by adding or subtracting the correct number of hours. For example, if your browser's time zone is UTC+3 and you want to display data for **Yesterday**, you need to change **Range from** to **now-1d/d-3h** and **Date to** to **now/d-3h**. If you want to display data for the **Today** period, you only need to change the value in the **Range from** field to **now/d-3h**.

> If you need results up to 23:59:59:999 UTC of yesterday, you can use an SQL query with a filter by Timestamp or specify an exact date and time.

The bounds of the period are inclusive: for example, for the **Today** time range, events are displayed from the beginning (00:00:00:000 UTC) of the current day to the current time (**now**) inclusive, and for the **Yesterday** period, events are displayed from the beginning (00:00:00:000 UTC) of the previous day to 00:00:00:000 UTC of the current day. You can view the date and time of the last data update and the exact period for which the data is displayed by hovering over the period icon in the widget.

If the **Show data for previous period** setting is enabled for the widget, and the widget is displaying data for a relative period, the tooltip also displays the previous period. The previous period is calculated relative to the current period as start and end values of the current period minus the duration of the current period. For example, if data is updated daily and displayed for a month, but only the first 10 days of the month have passed, the previous period is taken to be the last 10 days of the previous month.

- If you want the widget to display data for the period selected for the layout, click the **Reset** button. Changing the displayed period on the layout also changes the period displayed in the widget.

How to create a query in Builder

To create a query in Builder:

1. Specify the values of the following parameters:

   a. **SELECT**—event fields that should be returned. The number of available fields depends on the selected graph type.

   - In the drop-down list on the left, select the event fields for which you want to display data in the widget.
   - The middle field displays what the selected field is used for in the widget: **metric** or **value**.

     If you selected the **Table** graph type, in the middle fields, you must specify column names using ANSII-ASCII characters.

   - In the drop-down list on the right, you can select an operation to be performed on the data:
     - **count**—event count. This operation is available only for the **ID** event field. Used by default for line charts, pie charts, bar charts, and counters. This is the only option for date histogram.
     - **max** is the maximum value of the **event** field from the event selection.
     - **min** is the minimum value of the **event** field from the event selection.
     - **avg** is the average value of the **event** field from the event selection.
     - **sum** is the sum of event field values from the event selection.

   b. **SOURCE** is the type of the data source. Only the **events** value is available for selection.

   c. **WHERE**—conditions for filtering events.

   - In the drop-down list on the left, select the event field that you want to use for filtering.
   - Select the necessary operator from the middle drop-down list. The available operators depend on the type of value of the selected event field.
   - In the drop-down list on the right, enter the value of the condition. Depending on the selected type of field, you may have to manually enter the value, select it from the drop-down list, or select it on the calendar.

   You can add search conditions by clicking **Add condition** or remove search conditions by clicking ×.

   You can also add groups of conditions by clicking **Add group**. By default, groups of conditions are added with the **AND** operator, but you can change the it if necessary. Available values: **AND**, **OR**, **NOT**. Group conditions are deleted using the **Delete group** button.

   d. **GROUP BY**—event fields or aliases to be used for grouping the returned data. This parameter is not available for **Counter** graph type.

   e. **ORDER BY**—columns used as the basis for sorting the returned data. This parameter is not available for the **Date Histogram** and **Counter** graph types.

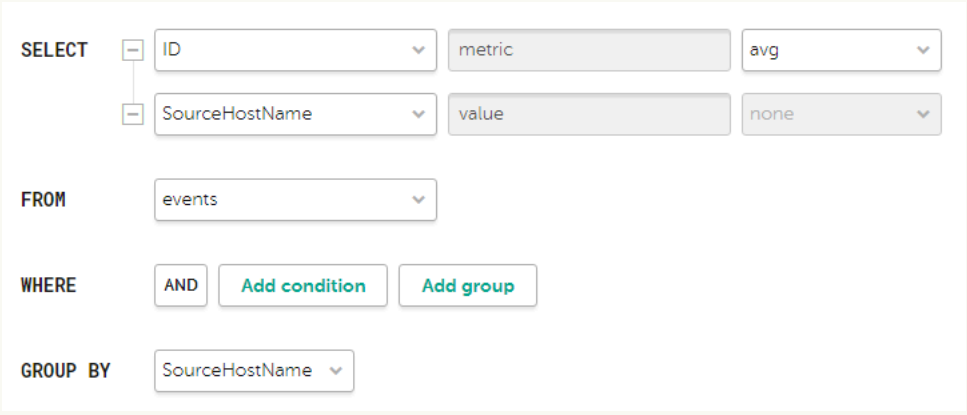
   - In the drop-down list to the left, select the value that will be used for sorting.

- Select the sort order from the drop-down list on the right: **ASC** for ascending, **DESC** for descending.
- For **Table** type graphs, you can add sorting conditions by clicking **Add column**.

f. **LIMIT** is the maximum number of data points for the widget. This parameter is not available for the **Date Histogram** and **Counter** graph types.

2. Click **Apply**.

[Example of search conditions in the query builder]



Parameters of search conditions for displaying the average number of bytes per host in the widget

Axes tab

The following table describes the settings on the **Axis** tab.

The tab is displayed if on the **General tab in the Graph** field you have selected one of the following values: **Bar chart**, **Line chart**, **Date Histogram**.

Description of parameters

Setting	Description
Point size	Point size on the graph. This field is displayed for the **Line chart** graph type.
Line-width	Width of the line on the graph. This field is displayed for the **Line chart** graph type.
Minimum and maximum values for the Y-axis	Scale of the Y axis. Negative values can be displayed on chart axes. This is due to the scaling of charts on the widget and can be fixed by setting zero as the minimum chart values instead of **Auto**.
Minimum and maximum values for the X-axis	Scale of the X axis. Negative values can be displayed on chart axes. This is due to the scaling of charts on the widget and can be fixed by setting zero as the minimum chart values instead of **Auto**.
Decimals	Number of decimals to which the displayed value must be rounded off.

Appearance tab

The following table describes the settings on the **Appearance** tab.

Description of parameters

Setting	Description
Name	Name of the widget.
Description	Description of the widget.
Show legend	Displays a legend for the analytics. The toggle switch is turned on by default.
Show nulls in legend	Displays parameters with a null value in the legend for analytics. The toggle switch is turned on by default. Null values are displayed only when the **Show data for previous period** setting is enabled and there is no data for the current period, but there was some data for the previous period.
Color	The color used for displaying the information: • **default** for your browser's default font color • **green** • **red** • **blue** • **yellow** This setting is available for graphs such as **Bar chart**, **Counter**, **Line chart**, **Date Histogram**.
Gradient	Gradient fill of the chart: • Two colors (red and green) • Three colors (red, yellow, and green) This setting is available for charts such as **Bar chart**, **Date Histogram**(with the **Bar chart** type), **Table**. You can adjust the direction of the fill by clicking the toggle direction icon (⇆); you can also set thresholds. If not set manually, gradient thresholds are calculated automatically from the data. If there is no data between the specified thresholds, the gradient is not applied. Instead, two colors are used: one for values below the lower threshold, and the other for values above the upper threshold.
Field for gradient indicator	The column in which the gradient fill indicators are displayed. This setting is available for charts of the **Table** type.
Horizontal	Makes the histogram horizontal instead of vertical. When this option is enabled, when a widget displays a large amount of data, horizontal scrolling is not available and all available information is fit into the fixed size of the widget. If there is a lot of data to display, it is recommended to increase the widget size.
Show total	Shows sums total of the values.
Period segments length	Duration of the segments into which you want to divide the period. Available for graphs of the **Date Histogram** type.
Scale	Scale for displaying data. Available for a **Stacked Bar chart**. The following values are possible: • **Linear** • **Square** • **Logarithm** The default is **Linear**.
Regression	An option for indicating the average value on the chart. The following values are possible: • **Not set** • **Linear** • **LOESS** (curve with locally estimated scatterplot smoothing). Available for the **Date Histogram** and **Line chart** types.

Actions tab

Settings on this tab let you configure left-click actions on various elements of widgets, for example, on a chart segment or the text of a legend item. You can select the following actions:

- Go to events.
- Filter by variable.

  The settings are described in the scenario for filtering data in multiple widgets.
- Drill down to another dashboard.

  The settings are described in the scenario for drilling down to another dashboard.

Contacting Technical Support

If you are unable to resolve your problem by consulting the documentation of the application, please contact Kaspersky Technical Support.

Kaspersky provides technical support for this application throughout its lifecycle (please refer to the product support lifecycle page).

When processing your request, Kaspersky Technical Support may need a report generated by the collect.sh utility. For detailed information on using the utility, please refer to the Knowledge Base.

REST API

You can access LogAnalyzer from third-party solutions using the API. The LogAnalyzer REST API operates over HTTP and consists of a set of request/response methods. REST API v3 is supported.

REST API requests must be sent to the following address:

https://<LogAnalyzer Core FQDN>:<REST API port>/api/<API version>/<request>

Example:
https://kuma.example.com:7223/api/v3

By default the 7223 port is used for API requests. You can change the port.

To change port used for REST API requests:

1. Log in to the OS of the server on which the LogAnalyzer Core is installed.
2. In the file /etc/systemd/system/multi-user.target.wants/kuma-core.service change the following string, adding required port:

   ```
   ExecStart=/opt/kaspersky/kuma/kuma core --external :7220 --internal :7210 --rest
   <required port number for REST API requests>
   ```

3. Restart LogAnalyzer by running the following commands in sequence:

   a. `systemctl daemon-reload`

   b. `systemctl restart kuma-core`

   New port is used for REST API.

Make sure that the port is available and is not closed by the firewall.

Authentication header: Authorization: Bearer <token>

Default data format: JSON

Date and time format: RFC 3339

Intensity of requests: unlimited

Creating a token

To generate a token for a user:

1. In the LogAnalyzer web interface, open **Settings** → **Users**.

   In the right part of the **Settings** section the **Users** table will be displayed.

2. Select the relevant user and click the **Generate token** button in the details area that opens on the right.

   The **New token** window opens.

3. If necessary, set the token expiration date:

   - Select the **No expiration date** check box.
   - In the **Expiration date** field, use the calendar to specify the date and time when the created token will expire.

4. Click the **Generate token** button.

   When you click this button, the user details area displays a field containing the automatically created token. When the window is closed, the token is no longer displayed. If you did not copy the token before closing the window, you will have to generate a new token.

5. Click **Save**.

The token is generated and can be used for API requests. These same steps can be taken to generate a token in your account profile.

Configuring permissions to access the API

In LogAnalyzer, you can configure which operations each user can perform. Permissions can be configured only for users created in LogAnalyzer.

To configure available operations for a user:

1. In the LogAnalyzer web interface, open **Settings** → **Users**.

   In the right part of the **Settings** section the **Users** table will be displayed.

2. Select the relevant user and click the **API access rights** button in the details area that opens on the right.

   This opens a window containing a list of available operations. By default, all API requests are available to a user.

3. Select or clear the check box next to the relevant operation.

4. Click **Save**.

Available operations for the user are configured.

The available operations can be configured in the same way in your account profile.

Authorizing API requests

Each REST API request must include token-based authorization. The user whose token is used to make the API request must have the permissions to perform this type of request.

Each request must be accompanied by the following header:

```
Authorization: Bearer <token>
```

Possible errors:

HTTP code	Description	message field value	details field value
400	Invalid header	invalid authorization header	Example: <example>
403	The token does not exist or the owner user is disabled	access denied	

Standard error

Errors returned by LogAnalyzer have the following format:

```
type Error struct {
    Message    string      `json:"message"`
    Details    interface{} `json:"details"`
}
```

REST API v3 operations

LogAnalyzer has a limited set of REST API methods compared to LogAnalyzer.

Open REST API Help

The table below lists the REST API methods available in the basic and commercial versions of LogAnalyzer. REST API methods described in the REST API reference but not listed in the table below are not available in LogAnalyzer.

REST API methods available in the basic and commercial versions of LogAnalyzer

REST API method	Description	Available in the basic version of LogAnalyzer	Available in the commercial version of LogAnalyzer
Dictionaries			
GET /dictionaries	Exporting dictionary	—	✓
POST /dictionaries/add_row	Add or modify a row with the specified key in the dictionary identified by the specified dictionary ID	—	✓

REST API method	Description	Available in the basic version of LogAnalyzer	Available in the commercial version of LogAnalyzer
POST /dictionaries/delete_row	Delete a row with the specified key in the dictionary identified by the specified dictionary ID	—	✓
POST /dictionaries/update	Dictionary updating in services	—	✓
Events			
POST /events	Searching events	✓	✓
GET /events/clusters	Viewing information about the cluster	✓	✓
Folders			
GET /folders	Search for folders	✓	✓
Reports			
GET /reports	Search for reports	✓	✓
Resources			
GET /resources	Resource search	—	✓
POST /resources/export	Exporting resources	—	✓
POST /resources/import	Importing resources	—	✓
POST /resources/toc	Viewing the contents of a resource file	—	✓
POST /resources/upload	Loading resource file	—	✓
POST /resources/:kind/create	Create a resource of the specified type	—	✓
POST /resources/:kind/validate	Validate resource contents	—	✓
GET /resources/:kind/:id	Get resource with specified type and ID	—	✓
PUT /resources/:kind/:id	Modify resource with specified type and ID	—	✓
GET /resources/:kind/:id/history	Get all versions of resource with specified type and ID	—	✓
GET /resources/:kind/:id/history/:changeNumber	Get previous version of resource with specified type, ID, and version number	—	✓
POST /resources/:kind/:id/history/:changeNumber	Restore previous version of resource with specified type, ID, and version number	—	✓
Services			
GET /services	Search for services	✓	✓
POST /services/create	Create a service based on the resource specified in the request body	✓	✓
POST /services/:id/reload	Restart service	✓	✓
POST /services/:id/restart	Restarting the service	✓	✓
System			
GET /system/backup	Create a backup copy of the LogAnalyzer Core	✓	✓
POST /system/restore	Restore LogAnalyzer Core from a backup copy	✓	✓
Tasks			
POST /tasks/create	Create task	✓	✓
Tenants			
GET /tenants	Tenant search	✓	✓
POST /tenants/create	Create tenant	✓	✓

REST API method	Description	Available in the basic version of LogAnalyzer	Available in the commercial version of LogAnalyzer
Users			
GET /users	Search for users	✓	✓
GET /users/id/:id	Search for a user by ID	✓	✓
GET /users/whoami	View token bearer information	✓	✓

Appendices

This section provides information that complements the main document text with reference information.

Commands for components manual starting and installing

This section contains the parameters of LogAnalyzer executable file /opt/kaspersky/kuma/kuma that can be used to manually start or install LogAnalyzer services. This may be useful for when you need to see output in the server operating system console.

Commands parameters

Commands	Description
`tools`	Start LogAnalyzer administration tools.
`collector`	Install, start, or remove a collector service.
`core`	Install, start, or uninstall a Core service.
`agent`	Install, start, or remove an agent service.
`help`	Get information about available commands and parameters.
`license`	Get information about license.
`storage`	Start or install a Storage.
`version`	Get information about version of the application.

Flags:

`-h`, `--h` are used to get help about any kuma command. For example, `kuma <component> --help`.

Examples:

- `kuma version` is used to get version of the LogAnalyzer installer.
- `kuma core -h` is used to get help about `core` command of LogAnalyzer installer.
- `kuma collector --core <address of the server where the collector should obtain its settings> --id <ID of the installed service> --api.port <port>` is used to start collector service installation.

Checking the integrity of LogAnalyzer files

You can check the integrity of LogAnalyzer components in the following ways:

- Manually, by running the script below
- On a schedule or automatically at application startup, with results recorded in the system log

Manual integrity check

The integrity of LogAnalyzer components is checked using a set of scripts based on the integrity_checker tool and located in the/opt/kaspersky/kuma/integrity/bin directory. An integrity check uses manifest xml files in the/opt/kaspersky/kuma/integrity/manifest/* directory, protected by a Kaspersky cryptographic signature.

Running the integrity check tool requires a user account with permissions at least matching those of the KUMA account.

The integrity check tool processes each LogAnalyzer component individually, and it must be run on servers that has the appropriate components installed. An integrity check also checks the xml file that was used.

To check the integrity of component files:

1. Run the following command to navigate to the directory that contains the set of scripts:

   ```
   cd /opt/kaspersky/kuma/integrity/bin
   ```

2. Then run one of the following commands depending on the LogAnalyzer component that you want to check:

   - `./check_all.sh` for LogAnalyzer Core components and storages.
   - `./check_core.sh` for the LogAnalyzer Core component.
   - `./check_collector.sh` for the LogAnalyzer collector component.
   - `./check_storage.sh` for the storage component.
   - `./check_event_router.sh` for the event router component.
   - `./check_kuma_exe.sh <full path to kuma.exe omitting file name>` for LogAnalyzer Agent for Windows. The standard location of the agent executable file on the Windows device is: C:\Program Files\Kaspersky Lab\KUMA\
   - `./check_linux_agent.sh <full path to the executable file of the LogAnalyzer agent without the file name>` for the LogAnalyzer agent for Linux.

The integrity of the component files is checked.

The result of checking each component is displayed in the following format:

- The Summary section describes the number of scanned objects along with the scan status: integrity not confirmed / object skipped / integrity confirmed:
  - Manifests – the number of manifest files processed.
  - Files – the number of LogAnalyzer files processed.
  - Directories – not used when LogAnalyzer integrity checking is performed.
  - Registries – not used when LogAnalyzer integrity checking is performed.
  - Registry values – not used when LogAnalyzer integrity checking is performed.
- Component integrity check result:
  - SUCCEEDED – integrity confirmed.
  - FAILED – integrity violated.

On a schedule or automatically at application startup

LogAnalyzer is a distributed, multi-component solution, and the location of its components on hosts is not known before the installation stage, therefore the configuration of the automatic integrity check of the components cannot be provided with the distribution kit and must be configured at the deployment stage.

We recommend checking the integrity of LogAnalyzer components when starting the application and on a schedule. We recommend scheduling an integrity check once a day. You can use scripting for this purpose. To get the necessary scripts, please contact Technical Support:

- manual_integrity_check.sh

  The script checks the integrity of all components or selected components. You can configure the scheduled integrity check with third-party applications and utilities, such as the cron utility. You can also run this script to manually check the integrity of components.

- systemd_integrity_check.sh

  Use this script to self-test the integrity of application components at startup. To add automatic integrity checking, run this script on each host where LogAnalyzer components are installed. The script should be run once. The integrity of the component is checked every time the LogAnalyzer service is started or restarted.

  If, after deployment, you add new LogAnalyzer services (for example, a new collector), run the systemd_integrity_check.sh script on the hosts with the new services. Otherwise, the integrity of the new components will not be checked at startup.

  If the check cannot verify the integrity of the component, the component does not run. LogAnalyzer will keep trying to start the service after checking its integrity. The results of the check are recorded in the system audit log. In this situation, either stop the service or eliminate the cause of the integrity violation.

Prerequisites

Before running the scripts:

- Move the script files to the /opt/kaspersky/kuma/integrity/bin/ directory.
- Grant the necessary permissions to run scripts to the 'kuma' user. To do so, run the following commands:

```
chown kuma:kuma ./systemd_integrity_check.sh
chmod +x ./systemd_integrity_check.sh
chown kuma:kuma ./manual_integrity_check.sh
chmod +x ./manual_integrity_check.sh
```

Running scripts

manual_integrity_check.sh

To check the integrity of the components, run the script on the host where the LogAnalyzer components are installed:

```
/opt/kaspersky/kuma/integrity/bin# ./manual_integrity_check.sh --core --collector --eventRouter --correlator --storage
```

This script checks the integrity of components which you specify in command line options. If you do not specify any components, the script checks all components.

systemd_integrity_check.sh

To add an automatic integrity check of application components at startup or on restart of the LogAnalyzer service, run the following command:

```
/opt/kaspersky/kuma/integrity/bin# ./systemd_integrity_check.sh
```

Both scripts log the results of component integrity checks in the system audit log. To view the log, use the dmesg command:

```
sudo dmesg
```

Normalized event data model

This section presents the LogAnalyzer normalized event data model. The maximum size of an event that can be processed by the KUMA collector is 4 MB.

Events that are not compliant to this data model must be converted to this format (or normalized) using Collectors.

Normalized event data model

Field name	Data type	Field size	Description
The name of a field reflects its purpose. The fields can be modified.			
`ApplicationProtocol`	String	31 characters	Name of the application layer protocol. For example, HTTPS, SSH, Telnet.
`BytesIn`	Number	From -9223372036854775808 to 9223372036854775807	Number of bytes received.

BytesOut	Number	From -9223372036854775808 to 9223372036854775807	Number of bytes sent.
DestinationAddress	String	45 characters	IPv4 or IPv6 address of the asset that the action will be performed on. For example, 0.0.0.0 or xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
DestinationCity	String	1,023 characters	City corresponding to the IP address from the DestinationAddress field.
DestinationCountry	String	1023 characters	Country corresponding to the IP address from the DestinationAddress field.
DestinationDnsDomain	String	255 characters	The DNS portion of the fully qualified domain name of the destination.
DestinationHostName	String	1023 characters	Host name of the destination. FQDN of the destination, if available.
DestinationLatitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Longitude corresponding to the IP address from the DestinationAddress field.
DestinationLongitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Latitude corresponding to the IP address from the DestinationAddress field.
DestinationMacAddress	String	17 characters	MAC address of the destination. For example, aa:bb:cc:dd:ee:00
DestinationNtDomain	String	255 characters	Windows Domain Name of the destination.
DestinationPort	Number	From -9223372036854775808 to 9223372036854775807	Port number of the destination.
DestinationProcessID	Number	From -9223372036854775808 to 9223372036854775807	System process ID registered on the destination.
DestinationProcessName	String	1023 characters	Name of the system process registered on the destination. For example, sshd, telnet.
DestinationRegion	String	1023 characters	Region corresponding to the IP address from the DestinationAddress field.
DestinationServiceName	String	1023 characters	Name of the service on the destination side. For example, sshd.
DestinationTranslatedAddress	String	45 characters	Translated IPv4 or IPv6 address of the destination. For example, 0.0.0.0 or xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
DestinationTranslatedPort	Number	From -9223372036854775808 to 9223372036854775807	Port number at the destination after translation.
DestinationUserID	String	1023 characters	User ID of the destination.
DestinationUserName	String	1023 characters	User name of the destination.
DestinationUserPrivileges	String	1023 characters	Names of roles that identify user privileges at the destination. For example, "User", "Guest", or "Administrator".
DeviceAction	String	63 characters	Action that was taken by the event source. For example, blocked, detected.
DeviceAddress	String	45 characters	IPv4 or IPv6 address of the device from which the event was received. For example, 0.0.0.0 or xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
DeviceCity	String	1023 characters	City corresponding to the IP address from the DeviceAddress field.
DeviceCountry	String	1023 characters	Country corresponding to the IP address from the DeviceAddress field.
DeviceDnsDomain	String	255 characters	DNS part of the fully qualified domain name of the device from which the event was received.
DeviceEventClassID	String	1023 characters	Event type ID assigned by the event source.

DeviceExternalID	String	255 characters	ID of the device or application assigned by the event source. If you want the DeviceExternalID field to be displayed as a link in the event properties, configure a template in the enrichment rule.
DeviceFacility	String	1023 characters	Value of the facility parameter set by the event source.
DeviceHostName	String	100 characters	Name of the device from which the event was received. FQDN of the device, if available.
DeviceInboundinterface	String	128 characters	Name of the incoming connection interface.
DeviceLatitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Longitude corresponding to the IP address from the DeviceAddress field.
DeviceLongitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Latitude corresponding to the IP address from the DeviceAddress field
DeviceMacAddress	String	17 characters	MAC address of the asset from which the event was received. For example, aa:bb:cc:dd:ee:00
DeviceNtDomain	String	255 characters	Windows Domain Name of the device.
DeviceOutboundinterface	String	128 characters	Name of the outgoing connection interface.
DevicePayloadID	String	128 characters	The payload's unique ID that is associated with the raw event.
DeviceProcessID	Number	From -9223372036854775808 to 9223372036854775807	ID of the system process on the device that generated the event.
DeviceProcessName	String	1023 characters	Name of the process.
DeviceProduct	String	63 characters	Name of the application that generated the event. The DeviceVendor, DeviceProduct, and DeviceVersion all uniquely identify the log source.
DeviceReceiptTime	Number	From -9223372036854775808 to 9223372036854775807	Time when the device received the event.
DeviceRegion	String	1023 characters	Region corresponding to the IP address from the DeviceAddress field.
DeviceTimeZone	String	255 characters	Time zone of the device on which the event was generated.
DeviceTranslatedAddress	String	45 characters	Re-translated IPv4 or IPv6 address of the device from which the event was received. For example, 0.0.0.0 or xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
DeviceVendor	String	63 characters	Vendor name of the event source. The DeviceVendor, DeviceProduct, and DeviceVersion all uniquely identify the log source.
DeviceVersion	String	31 characters	Application version of the event source. The DeviceVendor, DeviceProduct, and DeviceVersion all uniquely identify the log source.
EndTime	Number	From -9223372036854775808 to 9223372036854775807	Date and time (timestamp) when the event ended.
EventOutcome	String	63 characters	Result of the operation. For example, success, failure.
ExternalID	String	40 characters	Field in which the ID can be saved.
FileCreateTime	Number	From -9223372036854775808 to 9223372036854775807	File creation time.
FileHash	String	255 characters	Hash of the file.
FileID	String	1023 characters	ID of the file.
FileModificationTime	Number	From -9223372036854775808 to 9223372036854775807	Time when the file was last modified.

FileName	String	1023 characters	Filename without specifying the file path.
FilePath	String	1023 characters	File path, including the file name.
FilePermission	String	1023 characters	List of file permissions.
FileSize	Number	From -9223372036854775808 to 9223372036854775807	File size.
FileType	String	1023 characters	File type.
Message	String	1023 characters	Brief description of the event.
Name	String	512 characters	Name of the event.
OldFileCreateTime	Number	From -9223372036854775808 to 9223372036854775807	Time when the OLD file was created from the event. The time is specified in UTC0. In the KUMA web interface, the value is displayed based in the timezone of the user's browser.
OldFileHash	String	255 characters	Hash of the OLD file.
OldFileID	String	1023 characters	ID of the OLD file.
OldFileModificationTime	Number	From -9223372036854775808 to 9223372036854775807	Time when the OLD file was last modified.
OldFileName	String	1023 characters	Name of the OLD file (without the file path).
OldFilePath	String	1023 characters	Path to the OLD file, including the file name.
OldFilePermission	String	1023 characters	List of permissions of the OLD file.
OldFileSize	Number	From -9223372036854775808 to 9223372036854775807	Size of the OLD file.
OldFileType	String	1023 characters	Type of the OLD file.
Reason	String	1023 characters	Information about the reason for the event.
RequestClientApplication	String	1023 characters	Value of the "user-agent" parameter of the http request.
RequestContext	String	2,048 characters	Description of the http request context.
RequestCookies	String	1023 characters	Cookies associated with the http request.
RequestMethod	String	1023 characters	Method used when making the http request.
RequestUrl	String	1023 characters	Requested URL.
Severity	String	1023 characters	Priority. This can be the Severity field or the Level field of the raw event.
SourceAddress	String	45 characters	IPv4 or IPv6 address of the source. Example format: 0.0.0.0 or xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
SourceCity	String	1023 characters	City corresponding to the IP address from the SourceAddress field.
SourceCountry	String	1023 characters	Country corresponding to the IP address from the SourceAddress field.
SourceDnsDomain	String	255 characters	The DNS portion of the fully qualified domain name of the source.
SourceHostName	String	1023 characters	Windows Domain Name of the event source device.
SourceLatitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Longitude corresponding to the IP address from the SourceAddress field.
SourceLongitude	Floating point number	+/- 1.7E-308 to 1.7E+308	Latitude corresponding to the IP address from the SourceAddress field.
SourceMacAddress	String	17 characters	MAC address of the source. Format example: aa:bb:cc:dd:ee:00
SourceNtDomain	String	255 characters	Windows Domain Name of the source.

SourcePort	Number	From -9223372036854775808 to 9223372036854775807	Source port number.
SourceProcessID	Number	From -9223372036854775808 to 9223372036854775807	System process ID.
SourceProcessName	String	1023 characters	Name of the system process at the source. For example, sshd, telnet.
SourceRegion	String	1023 characters	Region corresponding to the IP address from the SourceAddress field.
SourceServiceName	String	1023 characters	Name of the service on the source side. For example, sshd.
SourceTranslatedAddress	String	45 characters	Translated IPv4 or IPv6 address of the source. Example format: 0.0.0.0 or xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
SourceTranslatedPort	Number	From -9223372036854775808 to 9223372036854775807	Port number of the source after translation.
SourceUserID	String	1023 characters	User ID of the source.
SourceUserName	String	1023 characters	User name of the source.
SourceUserPrivileges	String	1023 characters	Names of roles that identify user privileges of the source. For example, "User", "Guest", or "Administrator".
StartTime	Number	From -9223372036854775808 to 9223372036854775807	Date and time (timestamp) when the activity associated with the event began.
Tactic	String	128 characters	Name of the tactic from the MITRE ATT&CK matrix.
Technique	String	128 characters	Name of the technique from the MITRE ATT&CK matrix.
TransportProtocol	String	31 characters	Name of the OSI Transport layer protocol (for example, TCP, UDP).
Type	Number	From -9223372036854775808 to 9223372036854775807	Event type: 1 - basic, 2 - aggregated, 4 - audit, 5 - monitoring.
Fields the purpose of which can be defined by the user. The fields can be modified.			
DeviceCustomDate1	Number, timestamp	From -9223372036854775808 to 9223372036854775807	Field for mapping a date and time value (timestamp). The time is specified in UTC0. In the KUMA web interface, the value is displayed based in the timezone of the user's browser.
DeviceCustomDate1Label	String	1023 characters	Field for describing the purpose of the DeviceCustomDate1 field.
DeviceCustomDate2	Number, timestamp	From -9223372036854775808 to 9223372036854775807	Field for mapping a date and time value (timestamp). The time is specified in UTC0. In the KUMA web interface, the value is displayed based in the timezone of the user's browser.
DeviceCustomDate2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomDate2 field.
DeviceCustomFloatingPoint1	Floating point number	+/- 1.7E-308 to 1.7E+308	Field for mapping floating point numbers.
DeviceCustomFloatingPoint1Label	String	1023 characters	Field for describing the purpose of the DeviceCustomFloatingPoint1 field.
DeviceCustomFloatingPoint2	Floating point number	+/- 1.7E-308 to 1.7E+308	Field for mapping floating point numbers.
DeviceCustomFloatingPoint2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomFloatingPoint2 field.
DeviceCustomFloatingPoint3	Floating point number	+/- 1.7E-308 to 1.7E+308	Field for mapping floating point numbers.

DeviceCustomFloatingPoint3Label	String	1023 characters	Field for describing the purpose of the DeviceCustomFloatingPoint3 field.
DeviceCustomFloatingPoint4	Floating point number	+/- 1.7E-308 to 1.7E+308	Field for mapping floating point numbers.
DeviceCustomFloatingPoint4Label	String	1023 characters	Field for describing the purpose of the DeviceCustomFloatingPoint4 field.
DeviceCustomIPv6Address1	String	45 characters	Field for mapping an IPv6 address value. Format example: y:y:y:y:y:y:y:y
DeviceCustomIPv6Address1Label	String	1023 characters	Field for describing the purpose of the DeviceCustomIPv6Address1 field.
DeviceCustomIPv6Address2	String	45 characters	Field for mapping an IPv6 address value. Format example: y:y:y:y:y:y:y:y
DeviceCustomIPv6Address2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomIPv6Address2 field.
DeviceCustomIPv6Address3	String	45 characters	Field for mapping an IPv6 address value. Format example: y:y:y:y:y:y:y:y
DeviceCustomIPv6Address3Label	String	1023 characters	Field for describing the purpose of the DeviceCustomIPv6Address3 field.
DeviceCustomIPv6Address4	String	45 characters	Field for mapping an IPv6 address value. For example, y:y:y:y:y:y:y:y
DeviceCustomIPv6Address4Label	String	1023 characters	Field for describing the purpose of the DeviceCustomIPv6Address4 field.
DeviceCustomNumber1	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer value.
DeviceCustomNumber1Label	String	1023 characters	Field for describing the purpose of the DeviceCustomNumber1 field.
DeviceCustomNumber2	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer value.
DeviceCustomNumber2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomNumber2 field.
DeviceCustomNumber3	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer value.
DeviceCustomNumber3Label	String	1023 characters	Field for describing the purpose of the DeviceCustomNumber3 field.
DeviceCustomString1	String	4,000 characters	Field for mapping a string value.
DeviceCustomString1Label	String	1,023 characters	Field for describing the purpose of the DeviceCustomString1 field.
DeviceCustomString2	String	4,000 characters	Field for mapping a string value.
DeviceCustomString2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString2 field.
DeviceCustomString3	String	4,000 characters	Field for mapping a string value.
DeviceCustomString3Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString3 field.
DeviceCustomString4	String	4,000 characters	Field for mapping a string value.
DeviceCustomString4Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString4 field.
DeviceCustomString5	String	4,000 characters	Field for mapping a string value.
DeviceCustomString5Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString5 field.
DeviceCustomString6	String	4,000 characters	Field for mapping a string value.

DeviceCustomString6Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString6 field.
DeviceDirection	Number	From -9223372036854775808 to 9223372036854775807	Field for describing the direction of connection for an event. "0" - incoming connection, "1" - outgoing connection.
DeviceEventCategory	String	1023 characters	Event category assigned by the device that sent the event to SIEM.
FlexDate1	Number, timestamp	From -9223372036854775808 to 9223372036854775807	Field for mapping a date and time value (timestamp). The time is specified in UTC0. In the KUMA web interface, the value is displayed based in the timezone of the user's browser.
FlexDate1Label	String	128 characters	Field for describing the purpose of the FlexDate1Label field.
FlexNumber1	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer value.
FlexNumber1Label	String	128 characters	Field for describing the purpose of the FlexNumber1Label field.
FlexNumber2	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer value.
FlexNumber2Label	String	128 characters	Field for describing the purpose of the FlexNumber2Label field.
FlexString1	String	1023 characters	Field for mapping a string value.
FlexString1Label	String	128 characters	Field for describing the purpose of the FlexString1Label field.
FlexString2	String	1023 characters	Field for mapping a string value.
FlexString2Label	String	128 characters	Field for describing the purpose of the FlexString2Label field.
Service fields. Cannot be edited.			
AffectedAssets	Nested [Affected] structure	-	Nested structure from which you can query alert-related assets and user accounts, and find out the number of times they appear in alert events.
AggregationRuleID	String	-	Aggregation rule ID.
AggregationRuleName	String	-	The name of the aggregation rule that processed the event.
BaseEventCount	Number	-	For an aggregated base event, this is the number of base events that were processed by the aggregation rule.
BaseEvents	Nested [Event] list	-	Nested structure containing a list of base events.
Code	String	-	In a base event, this is the code of a process, function or operation return from the source.
DestinationAccountID	String	-	This field stores the user ID.
DestinationAssetID	String	-	This field stores the asset ID of the destination.
DeviceAssetID	String	-	This field stores the ID of the asset that sent the event to SIEM.
Extra	Nested [string:string] dictionary	-	During normalization of a raw event, this field can be used to place those fields that have not been mapped to KUMA event fields. This field can be filled in only for base events. The maximum size of the field is 4 MB.
ID	String	-	Unique event ID of UUID type. For a base event that is generated on the collector, the ID is generated by the collector. The ID never changes its value.
Raw	String	-	Non-normalized text of the original raw event. Maximum field size is 16,384 bytes.

ReplayID	String	-	ID of the retroscan that generated the event.
ServiceID	String	-	ID of the service instance: collector, storage.
ServiceName	String	-	Name of the microservice instance that the KUMA administrator assigns when creating the microservice.
SourceAccountID	String	-	This field stores the user ID.
SourceAssetID	String	-	This field stores the asset ID of the event source.
SpaceID	String	-	ID of the space.
TenantID	String	-	This field stores the ID of the tenant.
TI	Nested [string:string] dictionary	-	Field that contains categories in a dictionary format received from an external Threat Intelligence source based on indicators from an event.
Timestamp	Number	-	Timestamp of the base event created in the collector. The time is specified in UTC0. In the KUMA web interface, the value is displayed based in the timezone of the user's browser.

Nested Affected structure

Field	Data type	Description
Assets	Nested [AffectedRecord] list	List and number of assets associated with the alert.
Accounts	Nested [AffectedRecord] list	List and number of user accounts associated with the alert.

Nested AffectedRecord structure

Field	Data type	Description
Value	String	ID of the asset or user account.
Count	Number	How many times an asset or user account appears in alert-related events.

Fields generated by LogAnalyzer

LogAnalyzer generates the following fields that cannot be modified: BranchID, BranchName, DestinationAccountName, DestinationAssetName, DeviceAssetName, SourceAccountName, SourceAssetName, TenantID (the field displays the name of the tenant, an enriched value, while the tenant ID is used for searching the database).

Configuring the data model of a normalized event from KATA EDR

When investigating incidents, it can be useful to build a chain of processes. For this purpose, it is important that certain fields of the normalized event contain KATA/EDR process IDs. After correctly configuring normalization, you can build a process chain for events coming from KATA/EDR using an SQL query.

Example of building a process chain using an SQL query

Example of an SQL query for selecting events containing a process chain

```
SELECT
    b.DeviceHostName AS HostName,
    b.SourceAccountID AS AccountID,
    b.SourceUserName AS UserName,
    a.DeviceCustomString3 AS GrandParentProcessID,
    a.SourceProcessName AS GrandParentProcessName,
    a.DeviceCustomString5 AS ParentProcessID,
    a.DestinationProcessName AS ParentProcessName,
    b.DeviceCustomString5 AS ProcessID,
    b.DestinationProcessName AS ProcessName,
    concat(a.SourceProcessName, ' -> ', a.DestinationProcessName, ' -> ',
b.DestinationProcessName) AS ProcessTree
FROM `events` AS a
INNER JOIN
    (
        SELECT *
        FROM `events`
        WHERE DeviceEventClassID='4688'
    ) AS b
ON a.DeviceCustomString5 = b.DeviceCustomString3
WHERE DeviceEventClassID='4688'
```

Result

As a result of the query execution, events that meet the query conditions are returned in the **Events** section. In the event card, the `ProcessTree` field lists the process chain: process A → process B → ... → process N.

For example:

```
C:\Windows\System32\svchost.exe -> C:\Windows\WinSxS\amd64_microsoft-windows-
servicingstack_31bf3856ad364e35_10.0.17763.164_none_7e114a3d4d0589d4\TIWorker.exe ->
C:\Windows\System32\conhost.exe
```

To build a process chain, you need to configure the copying of data from the fields of the raw events to the fields of the normalized event in KUMA normalizers as follows:

1. For any KATA/EDR events, you must configure normalization with copying of the following fields:

   - The `EventType` field of the KATA/EDR event must be copied to the `DeviceEventCategory` field of the normalized KUMA event.

   - The `HostName` field of the KATA/EDR event must be copied to the `DeviceHostName` field of the normalized KUMA event.

2. For any event where `DeviceProduct = 'KATA'`, normalization must be configured in accordance with the table below.

Normalization of event fields from KATA/EDR

KATA/EDR event field	Normalized event field
IOATag	DeviceCustomIPv6Address2
	IOATag
IOAImportance	DeviceCustomIPv6Address1
	IOAImportance
FilePath	FilePath
FileName	FileName
Md5	FileHash
FileSize	FileSize

3. For events listed in the table below, additional normalization with field copying must be configured in accordance with the table.

Additional normalization with copying of event fields from KATA/EDR

Event	Raw event field	Normalized event field
Process	UniqueParentPid	FlexString1
	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
AppLock	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
BlockedDocument	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
Module	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
FileChange	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName

Event	Raw event field	Normalized event field
Driver	HostName	DeviceHostName
	FileName	FileName
	ProductName	DeviceCustomString5,
		ProductName
	ProductVendor	DeviceCustomString6
		ProductVendor
Connection	UniquePid	FlexString2
	HostName	DeviceHostName
	URI	RequestURL
	RemoteIP	DestinationAddress
	RemotePort	DestinationPort
PortListen	UniquePid	FlexString2
	HostName	DeviceHostName
	LocalIP	SourceAddress
	LocalPort	SourcePort
Registry	UniquePid	FlexString2
	HostName	DeviceHostName
	ValueName	DeviceCustomString5
		New Value Name
	KeyName	DeviceCustomString4
		New Key Name
	PreviousKeyName	FlexString2
		Old Key Name
	ValueData	DeviceCustomString6
		New Value Data
	PreviousValueData	FlexString1
		Old Value Data
	ValueType	FlexNumber1
		Value Type
	PreviousValueType	FlexNumber2
		Previous Value Type
SystemEventLog	UniquePid	FlexString2
	HostName	DeviceHostName
	OperationResult	EventOutcome
	EventId	DeviceCustomNumber3
		EventId
	EventRecordId	DeviceCustomNumber2
		EventRecordId
	Channel	DeviceCustomString6
		Channel
	ProviderName	SourceUserID

Event	Raw event field	Normalized event field
ThreatDetect	UniquePid	FlexString2
	HostName	DeviceHostName
	VerdictName	EventOutcome
	DetectedObjectType	OldFileType
	isSilent	FlexString1
		Is Silent
	RecordId	DeviceCustomString5
		Record ID
	DatabaseTimestamp	DeviceCustomDate2
		Database Timestamp
ThreatDetectProcessingResult	UniquePid	FlexString2
	HostName	DeviceHostName
	ThreatStatus	DeviceCustomString5
		Threat Status
PROCESS_INTERPRET_FILE_RUN	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
	InterpretedFilePath	OldFilePath
	InterpretedFileSize	OldFileSize
	InterpretedFileHash	OldFileHash
PROCESS_CONSOLE_INTERACTIVE_INPUT	UniquePid	FlexString2
	HostName	DeviceHostName
	InteractiveInputText	DeviceCustomString4
		Command Line
AMSI SCAN	UniquePid	FlexString2
	HostName	DeviceHostName
	ObjectContent	DeviceCustomString5
		Object Content

User account data model

User account fields can be referenced from email templates.

Field	Value type	Description
`ID`	String	User account ID.
`ObjectGUID`	String	Active Directory attribute. User account ID in Active Directory.
`TenantID`	String	Tenant ID.
`TenantName`	String	Tenant name.
`UpdatedAt`	Number	Last update of user account.
`Domain`	String	Domain.

CN	String	Active Directory attribute. User name.
displayName	String	Active Directory attribute. Displayed user name.
DistinguishedName	String	Active Directory attribute. LDAP object name.
employeeID	String	Active Directory attribute. Employee ID.
Mail	String	Active Directory attribute. User email address.
mailNickname	String	Active Directory attribute. Alternate email address.
Mobile	String	Active Directory attribute. Mobile phone number.
ObjectSID	String	Active Directory attribute. Security ID.
SAMAccountName	String	Active Directory attribute. Login.
TelephoneNumber	String	Active Directory attribute. Phone number.
UserPrincipalName	String	Active Directory attribute. User principal name (UPN).
Archived	TRUE/FALSE string	Indicator that determines whether a user account is obsolete.
MemberOf	List of strings	Active Directory attribute. AD groups joined by the user.
PreliminarilyArchived	TRUE/FALSE string	Indicator that determines whether a user account should be designated as obsolete.
CreatedAt	Number	User account creation date.
SN	String	Active Directory attribute. Last name of the user.
SAMAccountType	String	Active Directory attribute. User account type.
Title	String	Active Directory attribute. Job title of the user.
Division	String	Active Directory attribute. User's department.
Department	String	Active Directory attribute. User's division.
Manager	String	Active Directory attribute. User's supervisor.
Location	String	Active Directory attribute. User's location.
Company	String	Active Directory attribute. User's company.
StreetAddress	String	Active Directory attribute. Company address.
PhysicalDeliveryOfficeName	String	Active Directory attribute. Delivery address.
managedObjects	List of strings	Active Directory attribute. Objects under control of the user.
UserAccountControl	Number	Active Directory attribute. AD account type.
WhenCreated	Number	Active Directory attribute. User account creation date.
WhenChanged	Number	Active Directory attribute. User account modification date.
AccountExpires	Number	Active Directory attribute. User account expiration date.
BadPasswordTime	Number	Active Directory attribute. Date of last unsuccessful login attempt.

LogAnalyzer audit events

Audit events are created when certain security-related actions are completed in LogAnalyzer. These events are used to ensure system integrity.

To view audit events, you need to go to the **Events** section in LogAnalyzer and add `SELECT * FROM 'events' WHERE Type=4` to the query.

As a result of executing the query, audit events are displayed if the user role allows viewing audit events.

Event fields with general information

Every audit event has the event fields described in the table below.

Event field name	Field value
`Timestamp`	Event time.
`DeviceHostName`	The event source host. For audit events, the hostname is the hostname of the host where the kuma-core service is installed because the service is the source of events.
`DeviceTimeZone`	Timezone of the system time of the server hosting the LogAnalyzer Core in the format `+-hh:mm`.
`Type`	Type of the audit event. `Audit` or `Base`.
`TenantID`	Tenant ID.
`DeviceVendor`	`Kaspersky`
`DeviceProduct`	`KUMA`
`EndTime`	Event creation time.
`SpaceID`	`KUMA Audit`

Event fields with general information

Every audit event has the event fields described below.

Event field name	Field value
`Timestamp`	Event time.
`DeviceHostName`	The event source host. For audit events, it is the hostname where kuma-core is installed, because it is the source of events.
`DeviceTimeZone`	Timezone of the system time of the server hosting the LogAnalyzer Core in the format `+-hh:mm`.
`Type`	Type of the audit event. `Audit` or `Base`.
`TenantID`	ID of the main tenant.
`DeviceVendor`	`Kaspersky`
`DeviceProduct`	`KUMA`
`EndTime`	Event creation time.
`SpaceID`	`KUMA Audit`

User was successfully signed in or failed to sign in

Event field name	Field value
`DeviceAction`	`user login`
`EventOutcome`	`succeeded` or `failed`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.

SourceUserName	User login.
SourceUserID	User ID.
Message	Description of the error. Appears only if an error occurred during login. Otherwise, the field will be empty.

User login changed

Event field name	Field value
DeviceAction	user login changed
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User name of the user that modified the data.
SourceUserID	ID of the user that modified the data.
DestinationUserName	User login whose data was changed.
DestinationUserID	ID of the user whose data was changed.
DeviceCustomString1	Current value of the login.
DeviceCustomString1Label	new login
DeviceCustomString2	Value of the login before it was changed.
DeviceCustomString2Label	old login

User role was changed

Event field name	Field value
DeviceAction	user role changed
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to change data.
SourceUserID	User ID that was used to change the data.
DestinationUserName	User login whose data was changed.
DestinationUserID	ID of the user whose data was changed.
DeviceCustomString1	Current value of the role.
DeviceCustomString1Label	new role

DeviceCustomString2	Value of the role before it was changed.
DeviceCustomString2Label	old role

Other data of the user was changed

Event field name	Field value
DeviceAction	user other info changed
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User name of the user that modified the data.
SourceUserID	ID of the user that modified the data.
DestinationUserName	User login whose data was changed.
DestinationUserID	ID of the user whose data was changed.

User logged out

This event appears only when the user pressed the logout button.

This event will not appear if the user is logged out due to the end of the session or if the user logs in again from another browser.

Field values of audit event User logged out

Event field name	Field value
DeviceAction	user logout
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login.
SourceUserID	User ID.

User password was changed

Event field name	Field value
DeviceAction	user password changed
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to change data.
SourceUserID	User ID that was used to change data.
DestinationUserName	User login whose data was changed.
DestinationUserID	User ID whose data was changed.

User was created

Event field name	Field value
DeviceAction	user created
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to create the user account.
SourceUserID	User ID that was used to create the user account.
DestinationUserName	User login for which the user account was created.
DestinationUserID	User ID for which the user account was created.
DeviceCustomString1	Role of the created user.
DeviceCustomString1Label	role

User role was assigned

Event field name	Field value
DeviceAction	granted access
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.

SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	Login of the user for whom the data changes were made.
SourceUserID	ID of the user for whom the data changes were made.
DestinationUserPrivileges	Role name. Available values: general admin, admin, analyst, operator.
DeviceCustomString5	ID of the tenant used to assign the role.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

User role was revoked

Event field name	Field value
DeviceAction	revoked access
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	Login of the user who makes the changes.
SourceUserID	ID of the user who makes the changes.
DestinationUserName	Login of the user for whom the changes are made.
DestinationUserID	ID of the user for whom the changes are made.
DestinationUserPrivileges	Role name. Available values: general admin, admin, analyst, operator.
DeviceCustomString5	ID of the tenant used to assign the role.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

The user has edited the set of fields settings to define sources

Event field name	Field value
DeviceAction	settings updated
DeviceFacility	eventSourceIdentity
EventOutcome	succeeded
SourceUserName	Login of the user who makes the changes.

SourceUserID	ID of the user who makes the changes.
DeviceCustomString5	Updated set of fields, | is used as the delimiter.

User access token was changed

Event field name	Field value
DeviceAction	user access token changed
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to change data.
SourceUserID	User ID that was used to change the data.
DestinationUserName	User login whose data was changed.
DestinationUserID	ID of the user whose data was changed.

Changed the set of spaces to differentiate access to events

Event field name	Field value
DeviceAction	• space set was created • space set was deleted • space set was updated (including granting and revoking access permissions)
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to edit settings.
SourceUserID	User ID that was used to change the settings.
DeviceCustomString5	ID of the space set.
DeviceCustomString5Label	set ID
DeviceCustomString3	Name of the space set.
DeviceCustomString3Label	space name

Service was created

Event field name	Field value
DeviceAction	service created
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to create the service.
SourceUserID	User ID that was used to create the service.
DeviceExternalID	Service ID.
DeviceProcessName	Service name.
DeviceFacility	Service type.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Service was deleted

Event field name	Field value
DeviceAction	service deleted
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to delete the service.
SourceUserID	User ID that was used to delete the service.
DeviceExternalID	Service ID.
DeviceProcessName	Service name.
DeviceFacility	Service type.
DestinationAddress	Address of the device that was used to start the service. If the service has never been started before, the field will be empty.
DestinationHostName	The FQDN of the machine that was used to start the service. If the service has never been started before, the field will be empty.
DeviceCustomString5	Tenant ID.

DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Service was reloaded

Event field name	Field value
DeviceAction	service reloaded
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to reset the service.
SourceUserID	User ID that was used to restart the service.
DeviceExternalID	Service ID.
DeviceProcessName	Service name.
DeviceFacility	Service type.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Service was restarted

Event field name	Field value
DeviceAction	service restarted
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to restart the service.
SourceUserID	User ID that was used to restart the service.
DeviceExternalID	Service ID.
DeviceProcessName	Service name.
DeviceFacility	Service type.
DeviceCustomString5	Tenant ID.

DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Service was started

Event field name	Field value
DeviceAction	service started
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
DeviceExternalID	Service ID.
DeviceProcessName	Service name.
DeviceFacility	Service type.
DestinationAddress	Address of the device where the service was started.
DestinationHostName	FQDN of the device where the service was started.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Service was paired

Event field name	Field value
DeviceAction	service paired
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
DeviceExternalID	Service ID.
DeviceProcessName	Service name.
DeviceFacility	Service type.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID

DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Service status was changed

Event field name	Field value
DeviceAction	service status changed
DeviceExternalID	Service ID.
DeviceProcessName	Service name.
DeviceFacility	Service type.
DestinationAddress	Address of the device where the service was started.
DestinationHostName	FQDN of the device where the service was started.
DeviceCustomString1	green, yellow, or red
DeviceCustomString1Label	new status
DeviceCustomString2	green, yellow, or red
DeviceCustomString2Label	old status
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Partition index was deleted by user

Event field name	Field value
DeviceAction	partition deleted
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to delete partition.
SourceUserID	User ID that was used to delete partition.
Name	Storage name | Tenant of the partition being moved or deleted | Name of the space to which the partition belongs.
DeviceCustomDate1	Data modification time (UTC). The date that is displayed in the storage partitions in the 'date' column.
DeviceCustomDate1Label	date of event
Message	deleted by user

Storage partition was deleted automatically due to expiration

Event field name	Field value
DeviceAction	applied retention policy by days.
EventOutcome	succeeded or failed
Name	Storage name | Tenant of the partition being moved or deleted | Name of the space to which the partition belongs.
DeviceCustomDate1	Partition creation date.
DeviceCustomDate1Label	date of partition
SourceServiceName	scheduler
DeviceCustomString1	Node ID.
DeviceCustomString1Label	nodeID
Message	If moved: • If EventOutcome = succeeded, the moved partition data to cold storage message is displayed. • If EventOutcome = failed, the move partition data to cold storage failed: error description > error message is displayed. If deleted: • If EventOutcome = succeeded , the deleted partition data message is displayed. • If EventOutcome = failed, the delete partition data failed: error description > error message is displayed.
DeviceCustomNumber1	Storage partition size in bytes.
DeviceCustomNumber1Label	size
DeviceCustomNumber2	Number of events in the storage partition.
DeviceCustomNumber2Label	events

Storage partition was deleted automatically or moved due to exceeding the storage capacity.

In LogAnalyzer, you can set a condition for storage partitions to have the disk partition automatically moved or deleted if the configured maximum size of such a partition is reached or exceeded. The maximum storage capacity can be configured in one of the following ways:

- Specify a percentage of the entire disk partition where the storage is installed.
- Set the size of the disk partition in GB.

If the condition is checked and it is found that the size of the storage is equal to or greater than the configured size in percent or GB, an audit event is generated and the storage partition is moved or deleted.

Event field name	Field value
DeviceAction	If the storage partition size is specified as a percentage of the disk size, the value is applied retention policy by percent. If the storage partition size is specified in GB, the value is applied retention policy by GB.
EventOutcome	succeeded or failed

Name	Storage name | Tenant of the partition being moved or deleted | Name of the space to which the partition belongs.
DeviceCustomDate1	Partition creation date.
DeviceCustomDate1Label	date of partition
SourceServiceName	scheduler
DeviceCustomString1	ID of the node that started the storage cleanup due to the maximum space usage threshold being reached.
DeviceCustomString1Label	retention policy trigger node
Message	If moved: • If EventOutcome = succeeded, the moved partition data to cold storage message is displayed. • If EventOutcome = failed, the move partition data to cold storage failed: error description > error message is displayed. If deleted: • If EventOutcome = succeeded , the deleted partition data message is displayed. • If EventOutcome = failed, the delete partition data failed: error description > error message is displayed.
DeviceCustomNumber1	Storage partition size in bytes.
DeviceCustomNumber1Label	size
DeviceCustomNumber2	Number of events in the storage partition.
DeviceCustomNumber2Label	events

Resource was added

Event field name	Field value
DeviceAction	resource added
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to add the resource.
SourceUserID	User ID that was used to add the resource.
DeviceExternalID	Resource ID.
DeviceProcessName	Resource name.

DeviceFacility	Resource type: • activeList • agent • aggregationRule • collector • connection • connector • correlationRule • correlator • destination • dictionary • enrichmentRule • filter • normalizer • proxy • responseRule • storage
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Resource was deleted

Event field name	Field value
DeviceAction	resource deleted
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to delete the resource.
SourceUserID	User ID that was used to delete the resource.
DeviceExternalID	Resource ID.
DeviceProcessName	Resource name.

DeviceFacility	Resource type: • activeList • agent • aggregationRule • collector • connection • connector • correlationRule • correlator • destination • dictionary • enrichmentRule • filter • normalizer • proxy • responseRule • storage
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Resource was updated

Event field name	Field value
DeviceAction	resource updated
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to update the resource.
SourceUserID	User ID that was used to update the resource.
DeviceExternalID	Resource ID.
DeviceProcessName	Resource name.

DeviceFacility	Resource type: • activeList • agent • aggregationRule • collector • connection • connector • correlationRule • correlator • destination • dictionary • enrichmentRule • filter • normalizer • proxy • responseRule • storage
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Settings were updated

Event field name	Field value
DeviceAction	settings updated
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to update the settings.
SourceUserID	User ID that was used to update the settings.
DeviceFacility	Type of settings.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Tenant was created

Event field name	Field value
DeviceAction	tenant created
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login used to create the tenant.
SourceUserID	User ID used to create the tenant.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Tenant was enabled

Event field name	Field value
DeviceAction	tenant enabled
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
SourcePort	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
SourceUserName	User login used to enable the tenant.
SourceUserID	User ID used to enable the tenant.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Tenant was disabled

Event field name	Field value
DeviceAction	tenant disabled
EventOutcome	succeeded

SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
SourcePort	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
SourceUserName	User login used to disable the tenant.
SourceUserID	User ID used to disable the tenant.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Other tenant data was changed

Event field name	Field value
DeviceAction	tenant other info changed
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
SourcePort	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
SourceUserName	User login that was used to change the tenant data.
SourceUserID	User ID that was used to change the tenant data.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Updated data retention policy after changing drives or the operation ended with an error

Event field name	Field value
DeviceAction	storage policy modified
EventOutcome	succeeded or failed
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
SourcePort	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
SourceUserName	User login that was used to change the tenant data.
SourceUserID	User ID that was used to change the tenant data.

The dictionary was successfully updated on the service or operation was unsuccessful

Event field name	Field value
`DeviceAction`	`dictionary update succeeded`
`EventOutcome`	`succeeded` or `failed`
`SourceTranslatedAddress`	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
`SourceUserName`	User login that was used to create the service.
`SourceUserID`	User ID that was used to create the service.
`DeviceExternalID`	Service ID.
`ExternalID`	Dictionary ID.
`DeviceProcessName`	Service name.
`DeviceFacility`	Service type.
`DeviceCustomString1`	Dictionary type: table, dictionary.
`DeviceCustomString1Label`	`dictionary type`
`DeviceCustomString5`	Tenant ID.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`
`Message`	If EventOutcome = `failed`, an error message can be found here.

Sending test events to LogAnalyzer

LogAnalyzer allows sending test events to the system. Use the option of sending test events to LogAnalyzer to test rules, reports, dashboards, and also to check the resource consumption of the collector with different event streams. Events can only be sent to a collector that receives events over TCP or HTTP.

To send test events, you need:

- The 'kuma' file started with certain options.

  In the following instructions, the file with raw events is named send_test_events.txt as an example. You can use your own file name.

- A configuration file in which you define the parameters for running the executable file.

  In the following instructions, the configuration file is named config_for_test_events as an example. You can use your own file name.

To send test events:

1. Get sample events to send to LogAnalyzer:

   a. In the LogAnalyzer web interface, in the **Events** section, in the upper-right corner, click the ⚙ icon and in the displayed window, on the **Event fields columns** tab, select the check box for the **Raw** field. The 'Raw' column is displayed in the **Events** window.

   b. Search for events.

   c. Export your search results: in the **Events** window, in the upper right corner, click ... and select **Export TSV**.

   d. Go to the **Task manager** section of LogAnalyzer and click the **Events export** task; in the context menu, select **Download**.

   The <name of file with exported events>.tsv file is displayed in the Downloads section.

   If you are not collecting raw events, enable collection for a short time by setting the **Keep raw event** setting of the normalizer to **Always**. After the collection is completed, restore the previous value of the **Keep raw event** setting.

   e. Create a text file named send_test_events.txt and copy the contents of the Raw field from <name of file with exported events>.tsv to a text file named send_test_events.txt.

   f. Save send_test_events.txt.

2. Create a config_for_test_events configuration file and add the following lines to the file:

```
{
"kind": "<tcp or http>",
"name": "-",
"connection": {
"name": "-",
"kind": "<tcp or http>",
"urls": ["<IP address of the KUMA collector for receiving events over TCP>:<port of the KUMA collector for receiving event over TCP>"]
}
}
```

   Save the config_for_test_events configuration file.

3. Ensure that network connectivity exists between the server sending events and the server on which the collector is installed.

4. To send the contents of the test event file to the LogAnalyzer collector, run the following command:

   ```
   /opt/kaspersky/kuma/kuma tools load --raw --events /home/events/send_test_events.txt --cfg home/events/config_for_test_events --limit 1500 --replay 100000
   ```

   Available settings

Setting	Description
`--events`	Full path to the file containing raw events. Required setting. If the full path is not specified, the command does not run.
`--cfg`	Path to the configuration file. Required setting. If the full path is not specified, the command does not run.
`--limit`	Stream to be sent to the collector, in events per second (EPS). Required setting. If no value is specified, the command does not run.
`--replay`	Number of events to send. Required setting. If no value is specified, the command does not run. The step for --replay is 10000. The minimum value is 10000. --replay 16 sends 10000 events. --replay 16000 sends 20000 events.

As a result of running the command, test events are successfully sent to the LogAnalyzer collector. You can verify the arrival of test events by searching for related events in the LogAnalyzer web interface.

Time format

LogAnalyzer supports processing information passed to the fields of the event data model with the timestamp type (EndTime, StartTime, DeviceCustomDate1, etc) in the following formats:

- "May 8, 2009 5:57:51 PM".
- "oct 7, 1970".
- "oct 7, '70".
- "oct. 7, 1970".
- "oct. 7, 70".
- "Mon Jan 2 15:04:05 2006".
- "Mon Jan 2 15:04:05 MST 2006".
- "Mon Jan 02 15:04:05 -0700 2006".
- "Monday, 02-Jan-06 15:04:05 MST".
- "Mon, 02 Jan 2006 15:04:05 MST".
- "Tue, 11 Jul 2017 16:28:13 +0200 (CEST)".
- "Mon, 02 Jan 2006 15:04:05 -0700".

- "Mon 30 Sep 2018 09:09:09 PM UTC".
- "Mon Aug 10 15:44:11 UTC+0100 2015".
- "Thu, 4 Jan 2018 17:53:36 +0000".
- "Fri Jul 03 2015 18:04:07 GMT+0100 (GMT Daylight Time)".
- "Sun, 3 Jan 2021 00:12:23 +0800 (GMT+08:00)".
- "September 17, 2012 10:09am".
- "September 17, 2012 at 10:09am PST-08".
- "September 17, 2012, 10:10:09".
- "October 7, 1970".
- "October 7th, 1970".
- "12 Feb 2006, 19:17".
- "12 Feb 2006 19:17".
- "14 May 2019 19:11:40.164".
- "7 oct 70".
- "7 oct 1970".
- "03 February 2013".
- "1 July 2013".
- "2013-Feb-03".

dd/Mon/yyyy format

- "06/Jan/2008:15:04:05 -0700".
- "06/Jan/2008 15:04:05 -0700".

mm/dd/yyyy format

- "3/31/2014".
- "03/31/2014".
- "08/21/71".
- "8/1/71".
- "4/8/2014 22:05".

- "04/08/2014 22:05".
- "4/8/14 22:05".
- "04/2/2014 03:00:51".
- "8/8/1965 12:00:00 AM".
- "8/8/1965 01:00:01 PM".
- "8/8/1965 01:00 PM".
- "8/8/1965 1:00 PM".
- "8/8/1965 12:00 AM".
- "4/02/2014 03:00:51".
- "03/19/2012 10:11:59".
- "03/19/2012 10:11:59.3186369".

yyyy/mm/dd format

- "2014/3/31".
- "2014/03/31".
- "2014/4/8 22:05".
- "2014/04/08 22:05".
- "2014/04/2 03:00:51".
- "2014/4/02 03:00:51".
- "2012/03/19 10:11:59".
- "2012/03/19 10:11:59.3186369".

yyyy:mm:dd format

- "2014:3:31".
- "2014:03:31".
- "2014:4:8 22:05".
- "2014:04:08 22:05".
- "2014:04:2 03:00:51".
- "2014:4:02 03:00:51".

- "2012:03:19 10:11:59".
- "2012:03:19 10:11:59.3186369".

Format containing Chinese characters

"2014年04月08日"

yyyy-mm-ddThh format

- "2006-01-02T15:04:05+0000".
- "2009-08-12T22:15:09-07:00".
- "2009-08-12T22:15:09".
- "2009-08-12T22:15:09.988".
- "2009-08-12T22:15:09Z".
- "2017-07-19T03:21:51:897+0100".
- "2019-05-29T08:41-04" without seconds, 2-character TZ.

yyyy-mm-dd hh:mm:ss format

- "2014-04-26 17:24:37.3186369".
- "2012-08-03 18:31:59.257000000".
- "2014-04-26 17:24:37.123".
- "2013-04-01 22:43".
- "2013-04-01 22:43:22".
- "2014-12-16 06:20:00 UTC".
- "2014-12-16 06:20:00 GMT".
- "2014-04-26 05:24:37 PM".
- "2014-04-26 13:13:43 +0800".
- "2014-04-26 13:13:43 +0800 +08".
- "2014-04-26 13:13:44 +09:00".
- "2012-08-03 18:31:59.257000000 +0000 UTC".
- "2015-09-30 18:48:56.35272715 +0000 UTC".

- "2015-02-18 00:12:00 +0000 GMT".
- "2015-02-18 00:12:00 +0000 UTC".
- "2015-02-08 03:02:00 +0300 MSK m=+0.000000001".
- "2015-02-08 03:02:00.001 +0300 MSK m=+0.000000001".
- "2017-07-19 03:21:51+00:00".
- "2014-04-26".
- "2014-04".
- "2014".
- "2014-05-11 08:20:13,787".

yyyy-mm-dd-07:00 format

"2020-07-20+08:00"

mm.dd.yyyy format

- "3.31.2014".
- "03.31.2014".
- "08.21.71".

yyyy.mm.dd format

- "2014.03.30"

yyyymmdd format and similar

- "20140601".
- "20140722105203".

yymmdd hh:mm:yy format

"171113 14:14:20"

Unix timestamp format

- "1332151919".
- "1384216367189".
- "1384216367111222".
- "13842163671111222333".

Mapping fields of predefined normalizers

The file available via the download link contains a description of the field mapping of preset normalizers.

Download Description of field mapping of preset normalizers.

Deprecated resources

List of deprecated resources

Name	Resource type	Description
[Deprecated][OOTB] Microsoft SQL Server xml	Normalizer	This normalizer was removed from the resource set in LogAnalyzer 3.2. If you were using this normalizer, you must migrate to the [OOTB] Microsoft Products for KUMA 3 normalizer.
[Deprecated][OOTB] Windows Basic	Normalizer	This normalizer was removed from the resource set in LogAnalyzer 3.2. If you were using this normalizer, you must migrate to the [OOTB] Microsoft Products for KUMA 3 normalizer.
[Deprecated][OOTB] Windows Extended v.0.3	Normalizer	This normalizer was removed from the resource set in KUMA 3.2. If you were using this normalizer, you must migrate to the [OOTB] Microsoft Products for KUMA 3 normalizer.
[Deprecated][OOTB] Cisco ASA Extended v 0.1	Normalizer	This normalizer was removed from the resource set in KUMA 3.2. If you were using this normalizer, you must migrate to the [OOTB] Cisco ASA and IOS syslog normalizer.
[Deprecated][OOTB] Cisco Basic	Normalizer	This normalizer was removed from the resource set in KUMA 3.2. If you were using this normalizer, you must migrate to the [OOTB] Cisco ASA and IOS syslog normalizer.
[Deprecated][OOTB] Linux audit and iptables syslog	Normalizer	This normalizer was removed from the resource set in KUMA 3.4. In KUMA 3.2, we recommend using the [OOTB] Linux auditd syslog for KUMA 3.2 normalizer.
[Deprecated][OOTB] Linux audit.log file	Normalizer	This normalizer was removed from the resource set in KUMA 3.4. In KUMA 3.2, we recommend using the [OOTB] Linux auditd file for KUMA 3.2 normalizer.
[OOTB] Checkpoint Syslog CEF by CheckPoint	Normalizer	This normalizer was removed from the resource set in KUMA 3.4. In KUMA 3.4 and newer versions, we recommend using the [OOTB] Checkpoint syslog normalizer.
[OOTB] Eltex MES Switches	Normalizer	This normalizer was removed from the resource set in KUMA 3.4. In KUMA 3.4 and newer versions, we recommend using the [OOTB] Eltex MES syslog, [OOTB] Eltex ESR syslog normalizers.

Name	Resource type	Description
[OOTB] PTsecurity NAD	Normalizer	This normalizer was removed from the resource set in KUMA 3.4. In KUMA 3.4 and newer versions, we recommend using the [OOTB] PTsecurity NAD json normalizer.
[OOTB][AD] Granted TGS without TGT (Golden Ticket)	Rule	The rule was removed from the KUMA 3.4 resource set.
[OOTB][AD] Possible Kerberoasting attack	Rule	The rule was removed from the KUMA 3.4 resource set.
[OOTB][AD][Technical] 4768. TGT Requested	Rule	The rule was removed from the KUMA 3.4 resource set.
[OOTB][AD] List of requested TGT. EventID 4768	Active list	The list was removed from the KUMA 3.4 resource set.

Generating events for testing a normalizer

If necessary, you can generate your own example events to test your normalizer. Such testing makes it easier to write regular expressions and lets you see which values end up in the LogAnalyzer event fields.

Keep in mind the following special considerations:

- This tests simulates event processing. Example events in the **Example event** field are intended for displaying examples in the **Field mapping** section. Examples of the parent normalizer are used to generate examples of child normalizers, taking into account the **Field to pass into normalizer** setting.
- Mutations cannot be applied.

To test the normalizer, you need to add an example event to the **Event examples** field in the selected normalizer and start generating events by using the relevant command. As a result of running the command, KUMA takes the example event from the **Example event** field and sends events to the normalizer with the specified interval. If necessary, you can specify multiple examples to get events for multiple examples.

To test the normalizer:

1. Select the collector that you want to use for testing:
   - If the collector is installed on the server and running, stop the collector service:

     ```
     sudo systemctl stop kuma-collector-<collector service ID copied from the KUMA web interface>.service
     ```

   - If the collector is not running, or is in the process of being created or edited, proceed to the next step.
2. In the collector creation wizard, if necessary, fill in or edit the required fields at the **Connect event sources** step and at the **Transport** step, then proceed to the **Parsing** step:
   a. Link a normalizer by selecting it from the drop-down list, or create a normalizer.
   b. In the **Event examples** field, add example events. For example, for a json normalizer, you can add the following value: `{"name": "test_events", "address": "10.12.12.31"}`. You can specify multiple examples if you want to receive events for multiple examples in the same normalizer. Events are generated for each example.

3. In the Collector Installation Wizard, go to the **Routing** step and specify the storage where you want to save test events.

4. Review the collector settings and click **Save**.

5. Go to the **Active services** section in LogAnalyzer and click **Add** to add a collector. This opens the **Choose a service** window; in that window, select the collector and click **Create service**. The collector is displayed in the **Active services** list.

6. Check the status of the collector to which events are being sent. The collector status should be red.

7. Run the event generation command with the necessary parameters:

   - If the collector is not installed on the server, but only added in the **Active services** section:

     ```
     sudo /opt/kaspersky/kuma/kuma collector --core <FQDN of the KUMA Core server>:<port used by the KUMA Core for internal communication (port 7210 is used by default)> --generator.interval <interval in seconds for generating and sending events> --id <collector service ID copied from the KUMA web interface> --api.port <number of a free, unused API port>
     ```

     If the value of the event generation and sending interval is not specified or it is set to zero, events are not generated.

   - If the collector is installed on the server:

     ```
     sudo /opt/kaspersky/kuma/kuma collector --generator.interval <value of the event generation and sending interval in seconds> --id <collector service ID copied from the KUMA web interface> --api.port <number of a free, unused API port>
     ```

     If the value of the event generation and sending interval is not specified or it is set to zero, events are not generated.

   As a result, LogAnalyzer generates events and sends them to the normalizer, observing the specified interval.

   You can verify that events have been created and satisfy your expectations in the **Events** section. For additional information about the check, see the /etc/systemd/system/multi-user.target.wants/kuma-collector-<collector service ID copied from the KUMA web interface>.service file.

If the result does not meet expectations, modify the example event:

- If the collector is not installed on the server and has only been in the **Active services** section, edit the **Event examples** field in the normalizer of the collector and save the collector settings.

- If the collector is installed on the server and stopped as a service, edit the **Event examples** field in the normalizer of the collector, save the collector settings, go to the **Active services** section, select the collector, and refresh the collector settings by clicking **Refresh**.

If the result meets expectations:

1. Disable event generation, for example, by pressing **Ctrl+C** on the command line.

2. Start the collector service; if the service is already installed on the server, but has been stopped:

   ```
   sudo systemctl start kuma-collector-<collector service ID copied from the KUMA web interface>.service
   ```

3. If the collector has only been added in the **Active services** section, but has not been installed on the server yet, install the collector on the server using the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma collector --core <FQDN of the KUMA Core server>:<port used by KUMA Core server for internal communication (port 7210 by default)> --id <collector service ID copied from the KUMA web interface> --api.port <port used for communication with the installed component> --install
   ```

Glossary

Aggregation

Combining several messages of the same type from the event source into a single event.

Cluster

A group of servers on which the LogAnalyzer application has been installed and that have been clustered together for centralized management using the application's web interface.

Collector

LogAnalyzer component that receives messages from event sources, processes them, and transmits them to a storage and/or third-party services to identify suspected information security incidents.

Connector

A LogAnalyzer component that ensures transport for receiving data from external systems.

Dashboard

Component of the LogAnalyzer system that performs data visualization.

Enrichment

The conversion of the textual representation of an event using dictionaries, constants, calls to the DNS service, and other tools.

Event

An instance of activity of network devices, application software, information security tools, operating systems, and other devices that can be detected and recorded. For example, events include: successful user logon events, log clear events, anti-virus software disable event.

Filter

The set of conditions the application uses to select events for further processing.

LogAnalyzer web interface

A LogAnalyzer service that provides a user interface to configure and track LogAnalyzer operations.

Network port

A TCP and UDP protocol setting that defines the destination of IP-format data packets that are transmitted to a host over a network and allows various applications running on the same host to receive the data independently of each other. Each application processes the data sent to a specific port (sometimes it is said that the application listens to this port number).

It's standard practice to assign standard port numbers to certain common network protocols (for example, web servers usually receive data over HTTP on TCP port 80), although in general an application can use any protocol on any port. Possible values: from 1 to 65,535.

Normalization

A process that formats data received from an event in accordance with the fields of the LogAnalyzer event data model. During normalization, the data may be modified in accordance with certain rules (for example, changing upper case characters to lower case, replacing certain sequences of characters with others, etc.).

Normalizer

System component responsible for processing raw events from event sources. One normalizer processes events from one device or software of one specific version.

Parsing

The process of organizing data and converting incoming events into LogAnalyzer format.

Raw event

An event that has not passed the normalization stage in LogAnalyzer.

Report

LogAnalyzer resource that is used to generate a dataset based on user-defined filter criteria.

Role

A set of access privileges established to grant the LogAnalyzer web interface user the authority to perform tasks.

SELinux (Security-Enhanced Linux)

A system for controlling process access to operating system resources based on the use of security policies.

SIEM

Security Information and Event Management system. A solution for managing information and events in a company's security system.

STARTTLS

Text exchange protocol enhancement that lets you create an encrypted connection (TLS or SSL) directly over an ordinary TCP connection instead of opening a separate port for the encrypted connection.

UserPrincipalName

UserPrincipalName (UPN)—user name in email address format, such as `username@domain.com`.

The UPN must match the actual email address of the user. In this example, `username` is the user name in the Active Directory domain (user logon name), and `domain.com` is the UPN suffix. They are separated by the `@` character. The DNS name of the Active Directory domain is used as the default UPN suffix in Active Directory.

Information about third-party code

Information about third-party code is in the LEGAL_NOTICES file located in the /opt/kaspersky/kuma/LEGAL_NOTICES folder.

Trademark notices

Registered trademarks and service marks are the property of their respective owners.

AMD is a trademark or registered trademark of Advanced Micro Devices, Inc.

Apache and Apache Cassandra are registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries.

Ubuntu, LTS are registered trademarks of Canonical Ltd.

Cisco, IOS, Snort are registered trademarks or trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

Citrix, Citrix NetScaler are either a registered trademark or a trademark of Cloud Software Group, Inc., and/or its subsidiaries in the United States and/or other countries.

Dameware is the exclusive property of SolarWinds Worldwide, LLC or its affiliates, is registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries.

Dell Technologies, Dell, and other trademarks are trademarks of Dell Inc or its subsidiaries.

The Grafana word mark and the Grafana logo are either registered trademarks/service marks or trademarks/service marks of Coding Instinct AB, in the United States and other countries and are used with Coding Instinct's permission. We are not affiliated with, endorsed or sponsored by Coding Instinct, or the Grafana community. We are not affiliated with, endorsed, or sponsored by Coding Instinct, or the Grafana community.

Elasticsearch is a trademark of Elasticsearch BV, registered in the U.S. and in other countries.

ESET is trademark or registered trademark of ESET spol. s r.o. or respective ESET entity.

F5 is a trademark of F5 Networks, Inc. in the U.S. and in certain other countries.

Firebird is a registered trademark of the Firebird Foundation.

Fortinet, FortiGate, FortiMail, FortiSOAR are either registered trademarks or trademarks of Fortinet, Inc. in the United States and/or other countries.

The FreeBSD mark is a registered trademark of The FreeBSD Foundation.

Google, Chrome, Google Chrome are trademarks of Google LLC.

HUAWEI, Huawei Eudemon are trademarks of Huawei Technologies Co., Ltd.

IBM, Guardium, InfoSphere are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide.

Intel, Core are trademarks of Intel Corporation or its subsidiaries.

Juniper Networks and JUNOS are trademarks or registered trademarks of Juniper Networks, Inc. in the United States and other countries.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft, Active Directory, Azure, Excel, Halo, Hyper-V, Lync, Office 365, Outlook, PowerShell, SharePoint, Skype, SQL Server, Windows, Windows PowerShell, and Windows Server are trademarks of the Microsoft group of companies.

Mozilla and Firefox are trademarks of the Mozilla Foundation in the U.S. and other countries.

NetApp is a trademark or registered trademark of NetApp, Inc. in the United States and/or other countries.

Netskope, the Netskope logo, and other Netskope product names referenced herein are trademarks of Netskope, Inc. and/or one of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries.

OpenSSL is a trademark owned by the OpenSSL Software Foundation.

OpenVPN is a registered trademark of OpenVPN, Inc.

Oracle is a registered trademark of Oracle and/or its affiliates.

Python is a trademark or registered trademark of the Python Software Foundation.

Red Hat, Red Hat Enterprise Linux, CentOS are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

Ansible is a registered trademark of Red Hat, Inc. in the United States and other countries.

Sendmail, and other product designations or names are trademarks or registered trademarks of Sendmail, Inc.

Debian is a registered trademark of Software in the Public Interest, Inc.

The CommuniGate Pro name is a trademark or registered trademark of Stalker Software, Inc.

Symantec is a trademark or registered trademark of Symantec Corporation or its affiliates in the U.S. and other countries.

OpenAPI is a trademark of The Linux Foundation.

Kubernetes is a registered trademark of The Linux Foundation in the United States and other countries.

Rocky Linux is a trademark of The Rocky Enterprise Software Foundation.

Trend Micro is a trademark or registered trademark of Trend Micro Incorporated.

The names, images, logos and pictures identifying UserGate's products and services are proprietary marks of UserGate and/or its subsidiaries or affiliates, and the products themselves are proprietary to UserGate.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

ClickHouse is a trademark of YANDEX LLC.

Zabbix is a registered trademark of Zabbix SIA.

ViPNet is a registered trademark of Infotecs.

The MITRE Corporation Terms of Use

LICENSE

The MITRE Corporation (MITRE) hereby grants you a non-exclusive, royalty-free license to use ATT&CK® for research, development, and commercial purposes. Any copy you make for such purposes is authorized provided that you reproduce MITRE's copyright designation and this license in any such copy.

"© 2025 The MITRE Corporation. This work is reproduced and distributed with the permission of The MITRE Corporation."

DISCLAIMERS

MITRE does not claim ATT&CK enumerates all possibilities for the types of actions and behaviors documented as part of its adversary model and framework of techniques. Using the information contained within ATT&CK to address or cover full categories of techniques will not guarantee full defensive coverage as there may be undisclosed techniques or variations on existing techniques not documented by ATT&CK.

ALL DOCUMENTS AND THE INFORMATION CONTAINED THEREIN ARE PROVIDED ON AN "AS IS" BASIS AND THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE MITRE CORPORATION, ITS BOARD OF TRUSTEES, OFFICERS, AGENTS, AND EMPLOYEES, DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

List of limitations

LogAnalyzer has some limitations that are not critical for the operation of the application:

- After upgrading LogAnalyzer 3.4 to version 4.0, some widgets on the dashboard are displayed with an error. To fix the error, please manually refresh the data on the dashboard or wait for it to be refreshed automatically.
- The widget form can be opened below the **Dashboards** settings form when you follow links from the **Used in widgets** column if you have opened the settings of any widget before opening the **Dashboards** settings form.