Elastic Stack

Reference Guide

Hyperscalers with Elastic Stack

Thursday, 14 September 2023

1 CONTENTS

2 INTRODUCTION

Hyperscalers identifies the Enterprise requirement as well as IT administrators to have the data of any format from various sources to search, analyse and visuals in real time. Elastic stack along with core products helps to achieve this in a single place.

The Elastic Stack, often known as the ELK Stack, is utilized in a wide range of use cases, including debugging faults in application metrics, looking into security concerns in logs, and powering search boxes on websites and apps. The Elastic Stack, which consists of Elasticsearch, Kibana, Beats, and Logstash, offers a flexible and adaptable platform for search and analysis for various types of data.

This can be very useful for companies working with large datasets, any complex search application requirements along with infrastructure metrics and container monitoring, logging and application performance monitoring, visualization and analysis of geographical data as well as business and security analytics.

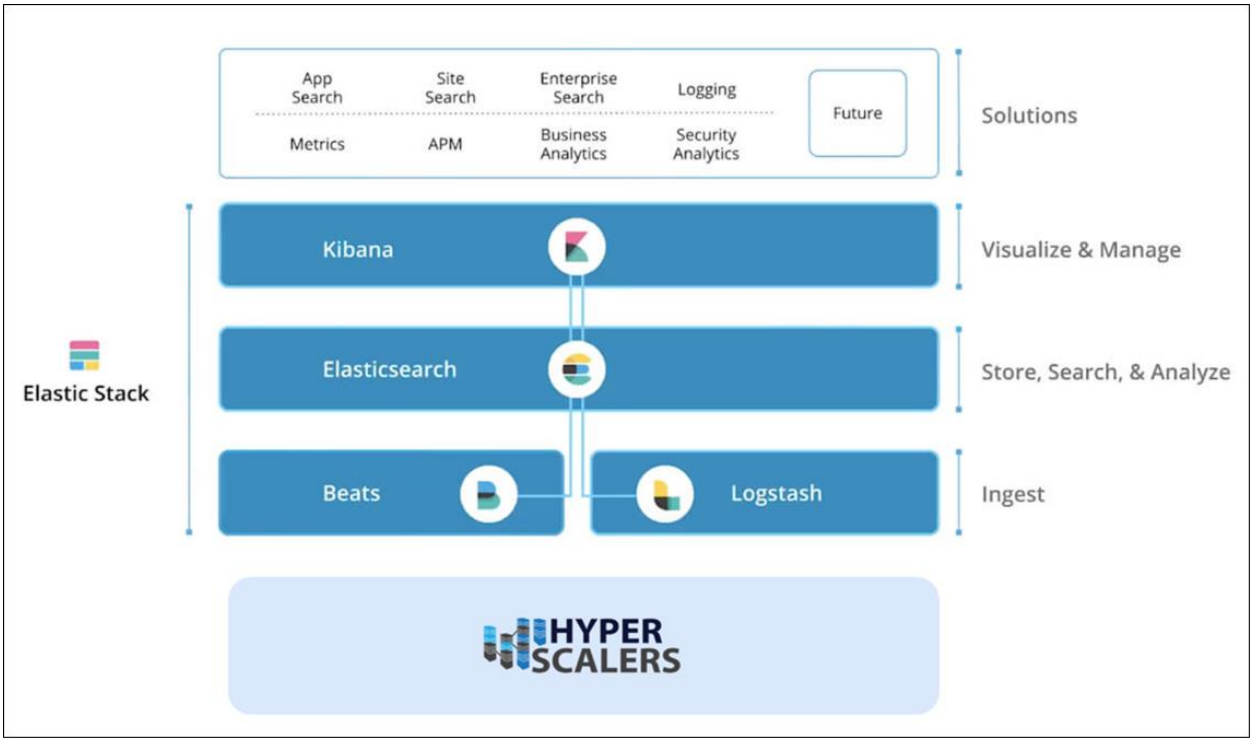


Figure 1 System Functional black diagram for Elastic Stack

Hyperscalers [1] is the world's first open supply chain Original Equipment Manufacturer- OEM, solving Information Technology challenges through standardization of best practices and hyperscale inspired practices and efficiencies. Hyperscalers offers choice across two open hardware architectures:

- Hyperscale - high efficiency open compute equipment as used by macro service providers
- Tier 1 Original – conventional equipment as per established Tier 1 OEM suppliers.

Each architecture is complete with network, compute, storage, and converged GP GPU infrastructure elements, and is open / free from vendor lock-in.

Hyperscalers' appliance solutions are packaged complete with hardware, software and pre-built (customisable) configurations. These were all pre-engineered using an in-house IP Appliance Design Process and validated in

partnership with associated major software manufacturers. Many can be "test-driven" using Hyperscalers Lab as a Service (LaaS). Hyperscalers appliance solutions are ideally suited to IaaS PaaS and SaaS providers looking to implement their services from anywhere.

Figure 2 Physical architecture design for Elastic Stack

The Elastic stack appliance by Hyperscalers is a complete package including the high-performance CPU, memory and network resources coupled with ELK stack core products to provide corporate search, observability, and security solutions that can be deployed anywhere and are based on a single, adaptable technology stack.

In this reference guide, we are trying to provide security to the company data by monitoring their logs, application and analysing their geographical data. Elasticsearch allows to store, search and analyse huge volumes of data quickly and in near real time and give back answers in milliseconds. It gives quick search responses by searching an index. So highly recommended for those companies, who are looking for data protection of their companies.

Audience and Purpose

Engineers, Enthusiasts, Executives, and IT professionals with background in Computer Science/ Electronics/ Information Technology with understanding in in Linux commands, Java language and basic electronics who intend to study, explore, deploy ELK Stack can be benefitted from this reference guide.

The purpose of this documentation is to provide in depth knowledge about the basic overview, appliance requirements and steps to deploy into your network.

Documents, Knowledge Base, and Technical Support

Hyperscalers reference architectures and appliance / solutions demonstrations are available at: https://www.hyperscalers.com/OCP-hyperscale-rack-solutions

For technical queries regarding this document and for managing virtualized, mobile, and cloud technologies, you can contact Hyperscalers technical support at support@hyperscalers.com.

Additional reference to the Elasticsearch, Kibana and security dashboard can be found in Hyperscalers lab as a service (LaaS) page and reference architecture section link – https://elastic.hyperscale2.com

Readers are recommended to have a prior knowledge and expertise with Kibana, Logstash, beats, Linux programming to better understand the following documentation.

Contact info@hyperscalers.com for more information.

Features of Elasticsearch [1]

The Elastic Stack comes with numerous tools (some originally bundled as X-Pack) to let you ingest, analyse, search, and display all forms of data at scale. These features range from enterprise-grade security and developer-friendly APIs to machine learning and graph analytics. The core security features Elastic stack integrates all the security features to the platform are listed below.

a) Authentication: sign on, securely.
b) Authorization: manage users and roles.
c) Encryption: prevent snooping, tampering, and sniffing.
d) Layered security: secure all the way down to the field level.
e) Audit logging: record which user activity.
f) Compliance: meeting security standards
g) Alerting: Highly available, Scalable alerting
h) Monitoring: monitor all types of devices, collect data
i) Scalability and Resiliency: Elasticsearch operates in a distributed environment designed from the ground up for perpetual peace of mind. Clusters grow with your needs — just add another node.
j) Management: The Elastic Stack comes with a variety of management tools, UIs, and APIs to allow full control over data, users, cluster operations, and more.
k) Alerting: The alerting features of the elastic stack give you the full power of the Elasticsearch query language to identify changes in your data that are interesting to you. In other words, if you can query something in Elasticsearch, you can alert on it.
l) Stack Security: The security features of the Elastic Stack give the right access to the right people. IT, operations, and application teams rely on these features to manage well-intentioned users and keep malicious actors at bay, while executives and customers can rest easy knowing data stored in the Elastic Stack is safe and secure.
m) Deployment: Public cloud, private cloud, or somewhere in between — we make it easy for you to run and manage the Elastic Stack.
n) Clients: The Elastic Stack allows you to work with data in whatever way you're most comfortable. With its RESTful APIs, language clients, robust DSL, and more (even SQL), we're flexible so you don't get stuck.

Important Considerations

The following documentation gives a detailed step by step deployment of Elastic Stack and the products that are installed offers a flexible and adaptable platform for search and analysis for various types of data. The Elasticsearch architecture is specific and designed to support the retrieval of documents, which helps handle complex data and queries. To track information, Elasticsearch uses keys prepended with an underscore, which represents metadata.

The Elasticsearch architecture is built for scalability and flexibility. The core components are Elasticsearch clusters, nodes, shards, and analyzers. Hyperscalers recommends the below important considerations before proceeding to the deployment phase.

a. ElasticStack required minimum three nodes to deployed, on each one will be master, and one be data. These nodes can be installed in both bare metal and can be virtualized and on the top of it, we install Elastic Stack products like Kibana, logstash, beats and Elastic Search.
b. Keep the hardware configuration consistent across all the nodes to ensure replication and high availability.
c. The Elastic stack requires minimum of at least 1 installation of Kibana applications for data visualization and exploration tool for log and time series analytics, applications monitoring and operational intelligence.
d. Elastic search required minimum of 2 or more cores intel processor with 32 GB memory and 3 hard disks on each node. It supports Linux and MacOS, windows, Debian, Ubuntu and is suitable for Red Hat, Centos SLES, OpenSuSE and other RPM based systems.

Digital IP Appliance Design Process

Hyperscalers has developed a Digital- IP-Appliance Design Process and associated Appliance Optimizer Utility which can enable the productization of IT-appliances for Digital-IP owners needing to hyperscale their services very quickly, reliably and at a fraction of traditional costs.

Appliance Optimizer Utility AOU

The Appliance Optimizer Utility (AOU) automates the discovery of appliance bottlenecks by pinging all layers in the proposed solution stack. A live dashboard unifies all key performance characteristics to provide a head-to-head performance assessment between all data-path layers in the appliance, as well as a comparison between holistic appliances.

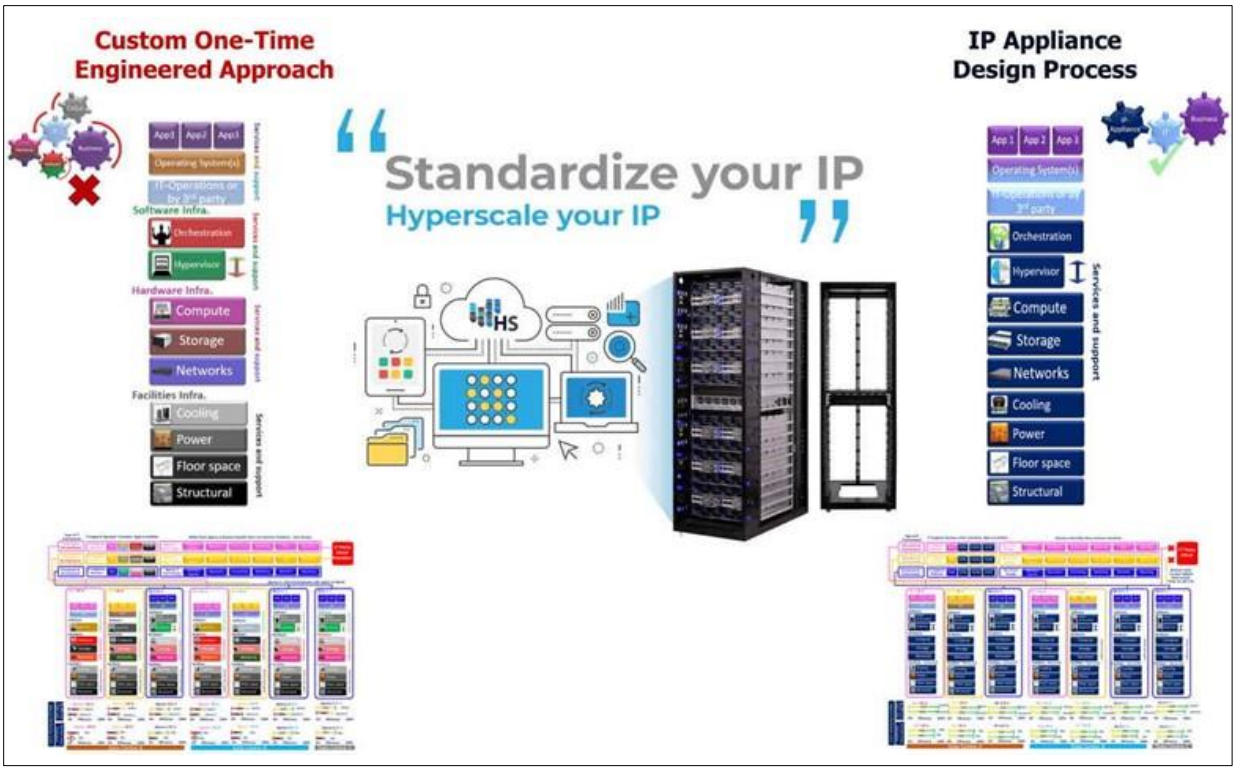


Figure 3 Digital IP-Appliance Design Process

Infrastructure Setup

To demonstrate a scalable and resilient Elastic Stack, we have used 3 nodes all as master and data.

Hardware Requirements

The **hardware** configuration for the build is listed below:

ELK 1 NODE	ELK 2 NODE	ELK 3 NODE
CPU: 2X INTEL(R) XEON(R) SILVER 4316 CPU @ 2.30GHZ SSD OS DISK: 2X 250G M.2 NVME DATA DISK: 4X 7.68TB MEMORY: 16X 32 GB 3200MT/S	CPU: 2X INTEL(R) XEON(R) SILVER 4316 CPU @ 2.30GHZ SSD OS DISK: 2X 250G M.2 NVME DATA DISK: 4X 7.68TB MEMORY: 16X 32 GB 3200MT/S	CPU: 2X INTEL(R) XEON(R) SILVER 4316 CPU @ 2.30GHZ SSD OS DISK: 2X 250G M.2 NVME DATA DISK: 4X 7.68TB MEMORY: 16X 32 GB 3200MT/S

Software Requirements:

Application: Elastic Stack (8.6)

Product used: Kibana, Logstash, Beats, APM and Elasticsearch Hadoop

Data visualization: Events, Flows

Ticketing system: Cases

Building Blocks:

S5Z | T43Z-2U

The ELK1 node is QuantaPlex T43Z-2U 20S5ZCU0050 with high performance multi node server and has high density server optimized for extreme compute performance and space efficiency. Featuring all-NVMe with high memory footprint and additional expansibility. Supports top-bin 3rd Generation Intel® Xeon® Scalable processors

in compact chassis. Two (2) CPU Sockets for up to 80 cores using Intel® Xeon® Silver 4316 Processor 40cores each. 62.3 Gib of Memory slots of 32 GB each. OS type 64 bit with 16 Front Storage drive bays, 4 for each node.

Access and Default Credentials

To access the Elastic Stack portal, first need to access the Hyperscalers lab as a service (LAAS) portal, go to https://www.hyperscale2.com which is a repository of enterprise appliances that can be used to test drive the use cases before deploying on a mass scale.

Elastic stack portal can be access from the link: https://elastic.hyperscale2.com

3 BASE PRODUCT DEPLOYMENT

Elasticsearch directly deploy on machines in their local data centre, it is increasingly common to deploy Elasticsearch in the public cloud or using container orchestrators. We can deploy Elasticsearch on the Amazon and Azure public clouds and via Kubernetes.

Elastic Cloud on Kubernetes (ECK) supports the deployment of the ELK stack on Kubernetes (including Elasticsearch, Logstash, Kibana and Beats). ECK takes advantage of Kubernetes orchestration capabilities.

ECK allows you to streamline critical operations, including managing and scaling clusters and storage, monitoring multiple clusters, securing clusters and using rolling upgrades for safe configuration. To distribute Elasticsearch resources across availability zones in the cloud, you can enable zone awareness.

You can also set up hot-warm-cold architectures for data storage. ECK lets you tier your data to meet different needs and conserve costs. Hot data is frequently accessed, warm data is infrequently accessed, and cold data is archival or backup storage—you can use lower-cost archive cloud storage tiers for warm and cold data.

Preinstallation Requirements

The Preinstallation Requirements are listed below:

Hardware Requirements:

Before installation of the Elastic stack, you need to setup hardware, the minimum requirements of the hardware chassis are listed below.

ELK 1 NODE	ELK 2 NODE	ELK 3 NODE
CPU: 2X INTEL(R) XEON(R) SILVER 4316 CPU @ 2.30GHZ SSD OS DISK: 2X 250G M.2 NVME DATA DISK: 4X 7.68TB MEMORY: 16X 32 GB 3200MT/S	CPU: 2X INTEL(R) XEON(R) SILVER 4316 CPU @ 2.30GHZ SSD OS DISK: 2X 250G M.2 NVME DATA DISK: 4X 7.68TB MEMORY: 16X 32 GB 3200MT/S	CPU: 2X INTEL(R) XEON(R) SILVER 4316 CPU @ .230GHZ SSD OS DISK: 2X 250G M.2 NVME DATA DISK: 4X 7.68TB MEMORY: 16X 32 GB 3200MT/S

Software requirements

To install Elastic search:

1. First need to install Java.
2. Install Elastic Search
3. Install ssh, certificate
4. Install Kibana
5. Logstash
6. Beats
7. APM
8. Elasticsearch Hadoop

Installation Components

The installation components are listed below:

1. Production level software product requirements

When installing the Elastic Stack, you must use the same version across the entire stack. For example, if you are using Elasticsearch 8.5.2, you install Beats 8.5.2, APM Server 8.5.2, Elasticsearch Hadoop 8.5.2, Kibana 8.5.2, and Logstash 8.5.2.

Installation Order

Install the Elastic Stacks products you want to use in the following order:

a. Elasticsearch
b. Kibana
c. Logstash
d. Beats
e. APM
f. Elasticsearch Hadoop

Installation of elastic search on Ubuntu

Follow the below process to install elastic search in Ubuntu.

1. Install Java[2]

In Linux, there are several ways to install java. Steps for setting the environment in the Linux operating system are as follows:

Step 1: Go to **Application -> Accessories -> Terminal**.
Step 2: Type command as below as follows:

```
sudo apt-get install openjdk-8-jdk
```

Step 3: For the "JAVA_HOME" (Environment Variable) type command as shown below, in "Terminal" using your installation path…(Note: the default path is as shown, but if you have to install OpenJDK at another location then set that path.)

```
export JAVA_HOME = /usr/lib/jvm/java-8-openjdk
```

Step 4: For "PATH" (Environment Value) type command as shown below, in "Terminal" using your installation path…Note: the default path is as shown, but if you have to install OpenJDK at another location then set that path.)

```
export PATH = $PATH:/usr/lib/jvm/java-8-openjdk/bin
```

Note: We are done setting up the environment in Java for Linux OS.

Note: *Now to check whether the installation is done correctly, type java -version in the Terminal. You will see that java is running on your machine.*

- ***Notepad/gedit*** *: They are simple* ***text-editors*** *for writing java programs. Notepad is available on Windows and gedit is available on Linux.*
- ***Eclipse IDE*** *: It is the most widely used IDE(Integrated Development Environment) for developing software in java. You can download Eclipse.*

2. Install Elastic Search[3]

1. Installing the Elasticsearch

- Self-Managed Elasticsearch options
  If you want to install and manage Elasticsearch yourself.

- Elasticsearch install packages

  Elasticsearch is provided in the following package formats:

Debian, Ubuntu, and other Debian-based systems	Deb	The deb package is suitable for Debian, Ubuntu, and other Debian-based systems. Debian packages may be downloaded from the Elasticsearch website or from our Debian repository. https://www.elastic.co/guide/en/elasticsearch/reference/8.6/deb.html

Step 1: First, update your system by using the following command:

```
$sudo apt install update
```

Step 2: Download *.deb file* for elasticsearch.

```
$wget
https://download.elastic.co/elasticsearch/release/org/elasticsearch/distribution/deb/elasticsearch/2.3.1/elasticsearch-2.3.1.deb
```

Step 3: Use dpkg commond to install the *.deb* file.

```
$sudo dpkg -i elasticsearch-2.3.1.deb
```

Step 4: Enable elasticsearch service

```
$sudo systemctl enable elasticsearch.service
```

Step 5: Setup network configuration for elasticsearch. Open file

```
$sudo nano /etc/elasticsearch/elasticsearch.yml
```

and set IP as localhost

```
...

network.host: 127.0.0.1

...
```

Step 6: Now, restart service.

```
$sudo systemctl restart elasticsearch
```

Step 7: Using and Testing Elasticsearch

```
$curl -X GET 'http://localhost:9200'
```

a. Kibana [4]

- Install Kibana yourself

Starting with version 6.0.0, Kibana only supports 64 bit operating systems. Kibana is provided in the following package formats:

deb	The deb package is suitable for Debian, Ubuntu, and other Debian-based systems. Debian packages may be downloaded from the Elastic website or from our Debian repository. https://www.elastic.co/guide/en/kibana/8.6/deb.html

Install Kibana with Debian package

1. Import the Elastic PGP key

Download and install the public signing key:

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o
/usr/share/keyrings/elasticsearch-keyring.gpg
```

2. Install from the APT repository

You may need to install the apt-transport-https package on Debian before proceeding:

```
sudo apt-get install apt-transport-https
```

Save the repository definition to /etc/apt/sources.list.d/elastic-8.x.list:

echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-8.x.list

#You can install the Kibana Debian package with:

```
sudo apt-get update && sudo apt-get install kibana
```

#Download and install the Debian package manually

The Debian package for Kibana v8.6.1 can be downloaded from the website and installed as follows:

```
wget https://artifacts.elastic.co/downloads/kibana/kibana-8.6.1-amd64.deb

shasum -a 512 kibana-8.6.1-amd64.deb

sudo dpkg -i kibana-8.6.1-amd64.deb
```

Start Elasticsearch and generate an enrolment token for Kibana

When you start Elasticsearch for the first time, the following security configuration occurs automatically:

- Authentication and authorization are enabled, and a password is generated for the elastic built-in superuser.
- Certificates and keys for TLS are generated for the transport and HTTP layer, and TLS is enabled and configured with these keys and certificates.

The password and certificate and keys are output to your terminal. You can then generate an enrollment token for Kibana with the elasticsearch-create-enrollment-token tool:

```
bin/elasticsearch-create-enrollment-token -s kibana
```

Start Kibana and enter the enrollment token to securely connect Kibana with Elasticsearch.

Run Kibana with *system*

To configure Kibana to start automatically when the system starts, run the following commands:

```
sudo /bin/systemctl daemon-reload

sudo /bin/systemctl enable kibana.service
```

Kibana can be started and stopped as follows:

```
sudo systemctl start kibana.service

sudo systemctl stop kibana.service
```

These commands provide no feedback as to whether Kibana was started successfully or not. Log information can be accessed via journalctl -u kibana.service..

b. Logstash [5]

- APT

Download and install the Public Signing Key:

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add –
```

#You may need to install the apt-transport-https package on Debian before proceeding:

```
sudo apt-get install apt-transport-https
```

Save the repository definition to /etc/apt/sources.list.d/elastic-8.x.list :

Run sudo apt-get update and the repository is ready for use. You can install it with:

```
echo "deb https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee
-a /etc/apt/sources.list.d/elastic-8.x.list
```

c. Beats [6]

Each Beat is a separately installable product.

- Auditbeat
- Filebeat
- Functionbeat
- Heartbeat
- Metricbeat
- Packetbeat
- Winlogbeat

- Auditbeat quick start: installation and configuration [7]

This guide describes how to get started quickly with audit data collection. You'll learn how to:

- install Auditbeat on each system you want to monitor
- specify the location of your audit data
- parse log data into fields and send it to Elasticsearch
- visualize the log data in Kibana

You need Elasticsearch for storing and searching your data, and Kibana for visualizing and managing it.

Step 1. Install Audibeat

Install Auditbeat on all the servers you want to monitor.
To download and install Auditbeat, use the commands that work with your system:

DEB

```
1. curl -L -O https://artifacts.elastic.co/downloads/beats/auditbeat/auditbeat-8.6.0-amd64.deb
2. sudo dpkg -i auditbeat-8.6.0-amd64.deb
```

Step 2: Connect to the Elastic Stack

Connections to Elasticsearch and Kibana are required to set up Auditbeat. Set the connection information in auditbeat.yml.

Self-Managed

i. Set the host and port where Auditbeat can find the Elasticsearch installation, and set the username and password of a user who is authorized to set up Auditbeat.

For example:

```
a) output.elasticsearch:
b) hosts: ["https://myEShost:9200"]
c) username: "auditbeat_internal"
d) password: "YOUR_PASSWORD"
e) ssl:
f) enabled: true
g) ca_trusted_fingerprint:
   "b9a10bbe64ee9826abeda6546fc988c8bf798b41957c33d05db736716513dc9c"
```

ii. If you plan to use our pre-built Kibana dashboards, configure the Kibana endpoint. Skip this step if Kibana is running on the same host as Elasticsearch.

```
1. setup.kibana:
2. host: "mykibanahost:5601"
3. username: "my_kibana_user"
4. password: "{pwd}"
```

Step 3: Configuration data collection modules

Auditbeat uses modules to collect audit information. By default, Auditbeat uses a configuration that's tailored to the operating system where Auditbeat is running. To use a different configuration, change the module settings in auditbeat.yml.

The following example shows the file_integrity module configured to generate events whenever a file in one of the specified paths changes on disk:

```
-auditbeat.modules:
- module: file_integrity
  paths:
  - /bin
  - /usr/bin
  - /sbin
```

- /usr/sbin
- /etc

Step 4: Set up assets

Auditbeat comes with predefined assets for parsing, indexing, and visualizing your data. To load these assets:

1. Make sure the user specified in auditbeat.yml is authorized to setup auditbeat.
2. From the installation directory, run:

DEB
auditbeat setup -e

-e is optional and sends output to standard error instead of the configured log output.

Step 5: Start Auditbeat

Before starting Auditbeat, modify the user credentials in auditbeat.yml and specify a user who is authorized to publish events.

To start Auditbeat, run:

DEB	sudo service auditbeat start

Step 6: View your data in Kibana

To make it easier for you to start auditing the activities of users and processes on your system, Auditbeat comes with pre-built Kibana dashboards and UIs for visualizing your data.

To open the dashboards:

1. Launch Kibana:

Elasticsearch Service	Self-managed
• Login to your Elastic Cloud account. • Navigate to the Kibana endpoint in your deployment.	Point your browser to http://localhost:5601, replacing localhost with the name of the Kibana host.

2. In the side navigation, click **Discover**. To see Auditbeat data, make sure the predefined auditbeat-* index pattern is selected.

3. In the side navigation, click **Dashboard**, then select the dashboard that you want to open.

- **FileBeat** [8]

a. Filebeat quick start: installation and configuration

This guide describes how to get started quickly with log collection.

- install Filebeat on each system you want to monitor
- specify the location of your log files
- parse log data into fields and send it to Elasticsearch
- visualize the log data in Kibana

Step 1. Install Filebeat

Install Filebeat on all the servers you want to monitor.

To download and install Filebeat, use the commands that work with your system:

DEB	curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-8.5.2-amd64.deb sudo dpkg -i filebeat-8.5.2-amd64.deb

Step 2: Connect to the Elastic Stack

Connections to Elasticsearch and Kibana are required to set up Filebeat.

Set the connection information in filebeat.yml.

Self-Managed	• Set the host and port where Filebeat can find the Elasticsearch installation, and set the username and password of a user who is authorized to set up Filebeat. For example: **output.elasticsearch:** **hosts: ["https://myEShost:9200"]** **username: "filebeat_internal"** **password: "YOUR_PASSWORD"** **ssl:** **enabled: true** **ca_trusted_fingerprint: "b9a10bbe64ee9826abeda6546fc988c8bf798b41957c33d05db736716513dc9c"** • If you plan to use our pre-built Kibana dashboards, configure the Kibana endpoint. Skip this step if Kibana is running on the same host as Elasticsearch. **setup.kibana:** **host: "mykibanahost:5601"** **username: "my_kibana_user"** **password: "{pwd}"**

Step 3. Collecting Log data

1. Identify the modules you need to enable.

DEB	filebeat modules list

2. From the installation directory, enable one or more modules. For example, the following command enables the nginx module config:

DEB	filebeat modules enable ngnix

3. In the module config under modules.d, change the module settings to match your environment. You must enable at least one fileset in the module. **Filesets are disabled by default.**
   For example, log locations are set based on the OS. If your logs aren't in default locations, set the paths variable:

```
- module: nginx
  access:
    enabled: true
    var.paths: ["/var/log/nginx/access.log*"]
```

Step 4: Set up assets

Filebeat comes with predefined assets for parsing, indexing, and visualizing your data. To load these assets:

1. Make sure the user specified in filebeat.yml is authorized to setup filebeat.
2. From the installation directory, run:

DEB	filebeat setup -e

Step 5: Start Filebeat

Before starting Filebeat, modify the user credentials in filebeat.yml

To start Filebeat, run:

DEB	sudo service filebeat start

Step 6: View your data in kibana

1. Launch Kibana

Elasticsearch Service	Self-managed
• Login to your Elastic Cloud account. • Navigate to the Kibana endpoint in your deployment.	Point your browser to http://localhost:5601, replacing localhost with the name of the Kibana host.

2. In the side navigation, click **Discover**. To see filebeat data, make sure the predefined filebeat-* index pattern is selected.
3. In the side navigation, click **Dashboard**, then select the dashboard that you want to open.

➢ **Function Beat** [9]

The Functionbeat distribution contains the command line tools, configuration file, and binary code required to run Functionbeat in your serverless environment. To download and extract the package, use the commands that work with your system.

Linux	curl -L -O https://artifacts.elastic.co/downloads/beats/functionbeat/functionbeat-8.5.2-linux-x86_64.tar.gz tar xzvf functionbeat-8.5.2-linux-x86_64.tar.gz

Step 2: Connect to the Elastic Stack

Connections to Elasticsearch and Kibana are required to set up Functionbeat. Set the connection information in functionbeat.yml.

Self-managed	• Set the host and port where Functionbeat can find the Elasticsearch installation, and set the username and password of a user who is authorized to set up Functionbeat. For example: • **output.elasticsearch:** • **hosts: ["https://myEShost:9200"]** • **username: "functionbeat_internal"** • **password: "YOUR_PASSWORD"** • **ssl:** • **enabled: true** • **ca_trusted_fingerprint: "b9a10bbe64ee9826abeda6546fc988c8bf798b41957c33d05db736716513dc9c"** If you plan to use our pre-built Kibana dashboards, configure the Kibana endpoint. Skip this step if Kibana is running on the same host as Elasticsearch. • **setup.kibana:** • **host: "mykibanahost:5601"** • **username: "my_kibana_user"** • **password: "{pwd}"**

Step 3: Configure functionbeat

To configure Function beat, edit the configuration file. The default configuration file is called functionbeat.yml.

Step 4: Set up assets

Functionbeat comes with predefined assets for parsing, indexing, and visualizing your data. To load these assets:

- Make sure the user specified in functionbeat.yml is authorized to set up Functionbeat.
- From the installation directory, run:

Linux	./functionbeat setup -e

-e is optional and sends output to standard error instead of the configured log output.

Step 5: View your data in Kibana

There are currently no example dashboards available for Function beat.

❖ Heartbeat [10]

This guide describes how to get started quickly collecting uptime data about your hosts. You'll learn how to:

- install Heartbeat
- specify the protocols to monitor
- send uptime data to Elasticsearch
- visualize the uptime data in Kibana

Before you begin

You need Elasticsearch for storing and searching your data, and Kibana for visualizing and managing it.

Self-managed

To install and run Elasticsearch and Kibana, see Installing the Elastic Stack.

Step 1: Install Heartbeat

Unlike most Beats, which you install on edge nodes, you typically install Heartbeat as part of a monitoring service that runs on a separate machine and possibly even outside of the network where the services that you want to monitor are running.

To download and install Heartbeat, use the commands that work with your system:

DEB

```
curl -L -O https://artifacts.elastic.co/downloads/beats/heartbeat/heartbeat-8.6.1-amd64.deb

sudo dpkg -i heartbeat-8.6.1-amd64.deb
```

P.S. C:\Program Files\Heartbeat> .\install-service-heartbeat.ps1

Step 2: Connect to the Elastic Stack

Connections to Elasticsearch and Kibana are required to set up Heartbeat. Set the connection information in heartbeat.yml. To locate this configuration file, see Directory layout.

Service Self-managed

a. Set the host and port where Heartbeat can find the Elasticsearch installation and set the username and password of a user who is authorized to set up Heartbeat.

For example:

```
output.elasticsearch:
  hosts: ["https://myEShost:9200"]
  username: "heartbeat_internal"
  password: "YOUR_PASSWORD"
  ssl:
    enabled: true

    ca_trusted_fingerprint: "b9a10bbe64ee9826abeda6546fc988c8bf798b41957c33d05db736716513dc9c"
```

b. If you plan to use our pre-built Kibana dashboards, configure the Kibana endpoint. Skip this step if Kibana is running on the same host as Elasticsearch.

```
setup.kibana:

  host: "mykibanahost:5601"

  username: "my_kibana_user"

  password: "{pwd}"
```

Step3: Configure Heartbeat monitors

Heartbeat provides monitors to check the status of hosts at set intervals. Heartbeat currently provides monitors for ICMP, TCP, and HTTP (see Heartbeat overview for more about these monitors).

You configure each monitor individually. In heartbeat.yml, specify the list of monitors that you want to enable. Each item in the list begins with a dash (-).

The following example configures Heartbeat to use three monitors: an icmp monitor, a tcp monitor, and an http monitor.

```
heartbeat.monitors:
- type: icmp
  schedule: '*/5 * * * * * *'
  hosts: ["myhost"]
  id: my-icmp-service
  name: My ICMP Service
- type: tcp
  schedule: '@every 5s'
  hosts: ["myhost:12345"]
  mode: any
```

```
id: my-tcp-service
- type: http
schedule: '@every 5s'
urls: ["http://example.net"]
service.name: apm-service-name
id: my-http-service
name: My HTTP Service
```

- The icmp monitor is scheduled to run exactly every 5 seconds (10:00:00, 10:00:05, and so on). The schedule option uses a cron-like syntax based on this cronexpr implementation.
- The tcp monitor is set to run every 5 seconds from the time when Heartbeat was started. Heartbeat adds the @every keyword to the syntax provided by the cronexpr package.
- The mode specifies whether to ping one IP (any) or all resolvable IPs
- The service.name field can be used to integrate heartbeat with elastic APM via the Uptime UI.

Step 4: Configure the Heartbeat location

Heartbeat can be deployed in multiple locations so that you can detect differences in availability and response times across those locations. Configure the Heartbeat location to allow Kibana to display location-specific information on Uptime maps and perform Uptime anomaly detection based on location.

To configure the location of a Heartbeat instance, modify the add_observer_metadata processor in heartbeat.yml. The following example specifies the geo.name of the add_observer_metadata processor as us-east-1a:

```
========================= Processors =========================
processors:
 - add_observer_metadata:
    # Optional, but recommended geo settings for the location Heartbeat is running in
    geo:
     # Token describing this location
     name: us-east-1a
     # Lat, Lon "
     #location: "37.926868, -78.024902"
```

- Uncomment the geo setting.
- Uncomment name and assign the name of the location of the Heartbeat server.
- Optionally uncomment location and assign the latitude and longitude.

Step 5: Set up assets

Heartbeat comes with predefined assets for parsing, indexing, and visualizing your data. To load these assets:

- Make sure the user specified in heartbeat.yml is authorized to set up Heartbeat.
- From the installation directory, run:

DEB

```
heartbeat setup -e
```

-e is optional and sends output to standard error instead of the configured log output.

Step 6: Start Heartbeat
Before starting Heartbeat, modify the user credentials in heartbeat.yml and specify a user who is authorized to publish events.

To start Heartbeat, run:

DEB
sudo service heartbeat-elastic start

Heartbeat is now ready to check the status of your services and send events to your defined output.

Step 7: View your data in Kibana

Heartbeat comes with pre-built Kibana dashboards and UIs for visualizing the status of your services. The dashboards are available in the uptime-contrib GitHub repository.

If you loaded the dashboards earlier, open them now.

To open the dashboards:

a. Launch Kibana:

   Self-managed

   Point your browser to http://localhost:5601, replacing localhost with the name of the Kibana host.

b. In the side navigation, click Discover. To see Heartbeat data, make sure the predefined heartbeat-* index pattern is selected.
c. In the side navigation, click Dashboard, then select the dashboard that you want to open.

d. **Elastic Search Metricsbeat** [11]

Metricbeat helps you monitor your servers and the services they host by collecting metrics from the operating system and services.

This guide describes how to get started quickly with metrics collection. You'll learn how to:

- install Metricbeat on each system you want to monitor
- specify the metrics you want to collect
- send the metrics to Elasticsearch
- visualize the metrics data in Kibana

Step 1: Install Metricbeat

Install Metricbeat as close as possible to the service you want to monitor. For example, if you have four servers with MySQL running, it's recommended that you run Metricbeat on each server. This allows Metricbeat to access your service from localhost and does not cause any additional network traffic or prevent Metricbeat from collecting metrics when there are network problems. Metrics from multiple Metricbeat instances will be combined on the Elasticsearch server. To download and install Metricbeat, use the commands that work with your system:

DEB

```
1. curl -L -O https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-8.5.2-amd64.deb
2. sudo dpkg -i metricbeat-8.5.2-amd64.deb
```

Step 2: Connect to the Elastic Stack

Connections to Elasticsearch and Kibana are required to set up Metricbeat.

Set the connection information in metricbeat.yml.

Self Managed	**a.** Set the host and port where Metricbeat can find the Elasticsearch installation, and set the username and password of a user who is authorized to set up Metricbeat. For example: **output.elasticsearch:** **hosts: ["https://myEShost:9200"]** **username: "metricbeat_internal"** **password: "YOUR_PASSWORD"** **ssl:** **enabled: true** **ca_trusted_fingerprint: "b9a10bbe64ee9826abeda6546fc988c8bf798b41957c33d05db736716513dc9c"** **b.** If you plan to use our pre-built Kibana dashboards, configure the Kibana endpoint. Skip this step if Kibana is running on the same host as Elasticsearch. **setup.kibana:** **host: "mykibanahost:5601"** **username: "my_kibana_user"** **password: "{pwd}"**

2. **Enable and configure metrics collection modules**

Metricbeat uses modules to collect metrics. Each module defines the basic logic for collecting data from a specific service, such as Redis or MySQL. A module consists of metricsets that fetch and structure the data.

1. Identify the modules you need to enable.

DEB	metricbeat modules list

2. From the installation directory, enable one or more modules. If you accept the default configuration without enabling additional modules, Metricbeat collects system metrics only. The following command enables the nginx config in the modules.d directory:

DEB	metricbeat module enable nginx

3. In the module config under modules.d, change the module settings to match your environment.

- Configure Metricbeat
- Config file format
- Metricbeat.reference.yml: This reference configuration file shows all non-deprecated options. You'll find it in the same location as **metricbeat.yml.**

Step 4: Set up assets

Metricbeat comes with predefined assets for parsing, indexing, and visualizing your data. To load these assets:

1. Make sure the user specified in metricbeat.yml is authorized to set up Metricbeat.
2. From the installation directory, run:

DEB	1. metricbeat setup -e

-e is optional and sends output to standard error instead of the configured log output.

Step 5: Start Metricbeat

Before starting Metricbeat, modify the user credentials in metricbeat.yml.

To start Metricbeat, run:

DEB	sudo service metricbeat start

Start 6: View your data in Kibana

Metricbeat comes with pre-built Kibana dashboards and UIs for visualizing log data. You loaded the dashboards earlier when you ran the setup command.

To open the dashboards:

1. Launch Kibana:
   - Self-Managed
     a. Point your browser to http://localhost:5601, replace localhost with the name of the kibana host.
2. In the side navigation, click **Discover**. To see Metricbeat data, make sure the predefined metricbeat-* index pattern is selected.
3. In the side navigation, click **Dashboard**, then select the dashboard that you want to open.

Packetbeat [12]

The best way to understand the value of a network packet analytics system like Packetbeat is to try it on your own traffic.

This guide describes how to get started quickly with network packets analytics. You'll learn how to:

- install Packetbeat on each system you want to monitor
- specify the network devices and protocols to sniff
- parse the packet data into fields and send it to Elasticsearch
- visualize the packet data in Kibana

Self-Managed: To install and run Elasticsearch and Kibana.

- On most platforms, Packetbeat requires the libpcap packet capture library. Depending on your OS, you might need to install it:

DEB	sudo apt-get install libpcap0.8

Step1: Install Packetbeat

You can install Packetbeat on dedicated servers, getting the traffic from mirror ports or tap devices, or you can install it on your existing application servers.

To download and install Packetbeat, use the commands that work with your system:

DEB	1. curl -L -O https://artifacts.elastic.co/downloads/beats/packetbeat/packetbeat-8.5.2-amd64.deb 2. sudo dpkg -i packetbeat-8.5.2-amd64.deb

Step2: Connect to the Elastic Stack

Connections to Elasticsearch and Kibana are required to set up Packetbeat. Set the connection information in packetbeat.yml.

- Self-Managed: Set the host and port where Packetbeat can find the Elasticsearch installation, and set the username and password of a user who is authorized to set up Packetbeat.

For example:

```
output.elasticsearch:
  hosts: ["https://myEShost:9200"]
  username: "packetbeat_internal"
  password: "YOUR_PASSWORD"
  ssl:
    enabled: true
    ca_trusted_fingerprint: "b9a10bbe64ee9826abeda6546fc988c8bf798b41957c33d05db736716513dc9c"
```

- If you plan to use our pre-built Kibana dashboards, configure the Kibana endpoint. Skip this step if Kibana is running on the same host as Elasticsearch.

```
setup.kibana:
  host: "mykibanahost:5601"
  username: "my_kibana_user"
  password: "{pwd}"
```

You can send data to other outputs, such as Logstash, but that requires additional configuration and setup.

Step 3: Configure Sniffing

In packetbeat.yml, configure the network devices and protocols to capture traffic from.

- Set the sniffer type. By default, Packetbeat uses pcap, which uses the libpcap library and works on most platforms.

- Specify the network device to capture traffic from. For example:

```
packetbeat.interfaces.device: eth0
```

Dev	packetbeat devices

- In the protocols section, configure the ports where Packetbeat can find each protocol. If you use any non-standard ports, add them here. Otherwise, use the default values.

packetbeat.protocols:

```
- type: dhcpv4
  ports: [67, 68]
- type: dns
  ports: [53]
- type: http
  ports: [80, 8080, 8081, 5000, 8002]
- type: mongodb
  ports: [27017]
- type: memcache
  ports: [11211]
- type: mysql
  ports: [3306,3307]
- type: pgsql
  ports: [5432]
- type: redis
  ports: [6379]
```

- type: cassandra ports: [9042] - type: tls ports: [443, 993, 995, 5223, 8443, 8883, 9243]	- type: thrift ports: [9090]

Step 4: Set up assets

Packetbeat comes with predefined assets for parsing, indexing, and visualizing your data. To load these assets:

Make sure the user specified in packetbeat.yml

From the installation directory, run:

DEB	1. packetbeat setup -e

-e is optional and sends output to standard error instead of the configured log output.

Step 5: Start Packetbeat

Before starting Packetbeat, modify the user credentials in packetbeat.yml.

To start Packetbeat, run:

DEB	1. sudo service packetbeat start

Step 6: View your data in Kibana

Packetbeat comes with pre-built Kibana dashboards and UIs for visualizing log data. You loaded the dashboards earlier when you ran the setup command.

To open the dashboards:

1. Launch Kibana

Self-Managed

Point your browser to http://localhost:5601, replacing localhost with the name of the Kibana host.

In the side navigation, click **Discover**. To see Packetbeat data, make sure the predefined packetbeat-* index pattern is selected.

❖ **Winlogbeat** [13]

This guide describes how to get started quickly with Windows log monitoring. You'll learn how to:

- install Winlogbeat on each system you want to monitor
- specify the location of your log files
- parse log data into fields and send it to Elasticsearch
- visualize the log data in Kibana

Before you begin

You need Elasticsearch for storing and searching your data, and Kibana for visualizing and managing it.

- Self-managed

  Install and run Elasticsearch and Kibana.

Step 1: Install Winlogbeat

1. Download the Winlogbeat zip file from the downloads page.
2. Extract the contents into C:\Program Files.
3. Rename the winlogbeat-<version> directory to Winlogbeat.
4. Open a PowerShell prompt as an Administrator (right-click on the PowerShell icon and select Run As Administrator).
5. From the PowerShell prompt, run the following commands to install the service.

```
PS C:\Users\Administrator> cd 'C:\Program Files\Winlogbeat'
PS C:\Program Files\Winlogbeat> .\install-service-winlogbeat.ps1
Security warning
Run only scripts that you trust. While scripts from the internet can be useful,
this script can potentially harm your computer. If you trust this script, use
the Unblock-File cmdlet to allow the script to run without this warning message.
Do you want to run C:\Program Files\Winlogbeat\install-service-winlogbeat.ps1?
[D] Do not run  [R] Run once  [S] Suspend  [?] Help (default is "D"): R
Status   Name            DisplayName
------   ----            -----------
Stopped  winlogbeat      winlogbeat
```

Note:
If script execution is disabled on your system, you need to set the execution policy for the current session to allow the script to run. For example: PowerShell.exe -ExecutionPolicy UnRestricted -File .\install-service-winlogbeat.ps1.

Step 2.: Connect to the Elastic Stack

Connections to Elasticsearch and Kibana are required to set up Winlogbeat.

Set the connection information in winlogbeat.yml. To locate this configuration file.

- Elasticsearch Self-Managed

  a. Set the host and port where Winlogbeat can find the Elasticsearch installation, and set the username and password of a user who is authorized to set up Winlogbeat.

  b. For example:

```
output.elasticsearch:
  hosts: ["https://myEShost:9200"]
```

```
username: "winlogbeat_internal"
password: "YOUR_PASSWORD"
ssl:
  enabled: true
  ca_trusted_fingerprint: "b9a10bbe64ee9826abeda6546fc988c8bf798b41957c33d05db736716513dc9c"
```

c. If you plan to use our pre-built Kibana dashboards, configure the Kibana endpoint. Skip this step if Kibana is running on the same host as Elasticsearch.

```
setup.kibana:

  host: "mykibanahost:5601"

  username: "my_kibana_user"

  password: "{pwd}"
```

Step 3: Configure Winlogbeat

In winlogbeat.yml, configure the event logs that you want to monitor.

1. Under winlogbeat.event_log, specify a list of event logs to monitor. By default, Winlogbeat monitors application, security, and system logs.

```
winlogbeat.event_logs:
  - name: Application
  - name: Security
  - name: System
```

To obtain a list of available event logs, run Get-EventLog * in PowerShell.

2. (Optional) Set logging options to write Winlogbeat logs to a file:

```
logging.to_files: true
logging.files:
  path: C:\ProgramData\winlogbeat\Logs
logging.level: info
```

3. After you save your configuration file, test it with the following command.

```
PS C:\Program Files\Winlogbeat> .\winlogbeat.exe test config -c .\winlogbeat.yml -e
```

Step 4: Set up assets

Winlogbeat comes with predefined assets for parsing, indexing, and visualizing your data. To load these assets:

1. Make sure the user specified in winlogbeat.yml is authorized to set up Winlogbeat.

2. From the installation directory, run:

```
PS > .\winlogbeat.exe setup -e
```

Step 5: Start Winlogbeat

Before starting Winlogbeat, modify the user credentials in winlogbeat.yml and specify a user who is authorized to publish events.

To start the Winlogbeat service, run:

```
PS C:\Program Files\Winlogbeat> Start-Service winlogbeat
```

Winlogbeat should now be running. If you used the logging configuration described here, you can view the log file at C:\ProgramData\winlogbeat\Logs\winlogbeat.

You can view the status of the service and control it from the Services management console in Windows. To launch the management console, run this command:

```
PS C:\Program Files\Winlogbeat> services.msc
```

Stop Winlogbeat

Stop the Winlogbeat service with the following command:

```
PS C:\Program Files\Winlogbeat> Stop-Service winlogbeat
```

Step 6: View your data in Kibana

Winlogbeat comes with pre-built Kibana dashboards and UIs for visualizing log data. You loaded the dashboards earlier when you ran the setup command.

To open the dashboards:

1. Launch Kibana:

Self-Managed

i. Point your browser to http://localhost:5601, replacing localhost with the name of the kibana host.

II. In the side navigation, click **Discover**. To see Winlogbeat data, make sure the predefined winlogbeat-* index pattern is selected.

III. In the side navigation, click **Dashboard**, then select the dashboard that you want to open.

e. **Application Performance Monitoring (APM)** [14]

APM describes how to:

- Collect Application Performance Monitoring (APM) data

- Send APM data to the Elastic Stack
- Explore and visualize the data in real-time

Prerequisites

You need Elasticsearch for storing and searching your data, and Kibana for visualizing and managing it. Also can use Hosted Elasticsearch service on elastic cloud or self-manage the elastic stack on your own hardware.

Self-Managed

- Elasticsearch cluster and Kibana (version 8.5) with a basic license or higher.
- Secure, encrypted connection between Kibana and Elasticsearch.
- Internet connection for Kibana to download integration packages from the Elastic Package Registry. Make sure the Kibana server can connect to https://epr.elastic.co on port 443.
- Kibana user with All privileges on Fleet and Integrations. Since many Integrations assets are shared across spaces, users need the Kibana privileges in all spaces.
- In the Elasticsearch configuration, the built-in API key service must be enabled. (xpack.security.authc.api_key.enabled: true).
- In the Kibana configuration, the saved objects encryption key must be set. Fleet requires this setting in order to save API keys and encrypt them in Kibana. You can either set xpack.encryptedSavedObjects.encryptionKey to an alphanumeric value of at least 32 characters, or run the kibana-encryption-keys command to generate the key.

Example Security settings

For testing purposes, you can use the following settings to get started quickly, but make sure you properly secure the Elastic Stack before sending real data.

elasticsearch.yml example:

- xpack.security.enabled: true
- xpack.security.authc.api_key.enabled: true

Kibana.yml example

- elasticsearch.username: "kibana_system"
- xpack.encryptedSavedObjects.encryptionKey: "something_at_least_32_characters"

The password should be stored in the Kibana keystore as described in the Elasticsearch security documentation.

Step1: Set up Fleet

Use Fleet in Kibana to get APM data into the Elastic Stack. The first time you use Fleet, you might need to set it up and add a Fleet Server:

Self-Managed

To deploy a self-managed Fleet server, you install an Elastic Agent and enroll it in an agent policy containing the fleet server integration.

Note: You can install only a single Elastic Agent per host, which means you cannot run Fleet Server and another Elastic Agent on the same host unless you deploy a containerized Fleet Server.

1. In Kibana, go to **Management > Fleet > Settings**. For more information about these settings, see Fleet settings.
2. Under **Fleet Server hosts**, click **Edit hosts** and specify one or more host URLs your Elastic Agents will use to connect to Fleet Server. For example, https://192.0.2.1:8220, where 192.0.2.1 is the host IP where you will install Fleet Server. Save and apply your settings.
3. In the **Elasticsearch hosts** field, specify the Elasticsearch URLs where Elastic Agents will send data. For example, https://192.0.2.0:9200. Skip this step if you've started the Elastic Stack with security enabled.
4. Save and apply the settings.
5. Click the **Agents** tab and follow the in-product instructions to add a Fleet server:

Note:

- Choose **Quick Start** if you want Fleet to generate a Fleet Server policy and enrollment token for you.
- Choose Advanced if you want either:
  - Use your own Fleet server policy.
  - Use your own TLS certificates to encrypt traffic between Elastic Agents and Fleet Server.

- It's recommended you generate a unique service token for each Fleet Server. For other ways to generate service tokens, see elasticsearch-service-tokens.
- If you are providing your own certificates:
  - Before running the install command, make sure you replace the values in angle brackets.
  - Note that the URL specified by --url must match the DNS name used to generate the certificate specified by --fleet-server-cert.
- The install command installs the Elastic Agent as a managed service and enrolls it in a Fleet Server policy.

If installation is successful, you'll see confirmation that Fleet Server connected. Click Continue enrolling Elastic Agent to begin enrolling your agents in Fleet Server.

Note:
If you're unable to add a Fleet-managed agent, click the Agents tab and confirm that the agent running Fleet Server is healthy.

The APM integration does not support running Elastic Agent in standalone mode; you must use Fleet to manage Elastic Agent.

Step 2: Configure the APM integration

Elastic Cloud runs a hosted version of Integrations Server that includes the APM integration. Self-managed users will need to add the APM integration before configuring it.

Self-Managed

I. In Kibana, select **Add integrations** > **Elastic APM**.
II. Click **Add Elastic APM**.
III. On the **Add Elastic APM integration** page, define the host and port where APM Server will listen. Make a note of this value—you'll need it later.
   a. Note: Using Docker or Kubernetes? Set the host to 0.0.0.0 to bind to all interfaces.

IV. Click **Save and continue**. This step takes a minute or two to complete. When it's done, you'll have an agent policy that contains an APM integration policy for the configuration you just specified.
V. To view the new policy, click **Agent policy 1**. Any Elastic Agents assigned to this policy will collect APM data from your instrumented services.

Step 3: Install and run an Elastic Agent on your machine

Elastic Agent is a single, unified way to add monitoring for logs, metrics, and other types of data to a host. It can also protect hosts from security threats, query data from operating systems, forward data from remote services or hardware, and more. A single agent makes it easier and faster to deploy monitoring across your infrastructure. Don't confuse Elastic Agent with APM agents—they are different components.

If you plan on enabling Real User Monitoring (RUM), you must run Elastic Agent centrally. If RUM is disabled, you should run Elastic Agent on edge machines.

To send APM data to the Elastic Stack:

1. In Kibana, go to **Fleet > Agents**, and click **Add agent**.
2. In the **Add agent** flyout, select an existing agent policy or create a new one. If you create a new policy, Fleet generates a new Fleet enrollment token.
3. Make sure **Enroll in Fleet** is selected.
4. Download, install, and enroll the Elastic Agent on your host by selecting your host operating system and following the **Install Elastic Agent on your host** step.

a. If you are enrolling the agent in a Fleet Server that uses your organization's certificate you *must* add the `--certificate-authorities` option to the command provided in the in-product instructions. If you do not include the certificate, you will see the following error: "x509: certificate signed by unknown authority".
After about a minute, the agent will enroll in Fleet, download the configuration specified in the agent policy, and start collecting data.

Note:

- *If you encounter an "x509: certificate signed by unknown authority" error, you might be trying to enroll in a Fleet Server that uses self-signed certs. To fix this problem in a non-production environment, pass the `--insecure` flag.*
- *Optionally, you can use the `--tag` flag to specify a comma-separated list of tags to apply to the enrolled Elastic Agent.*
- *Refer to Installation layout for the location of installed Elastic Agent files.*
- *Because Elastic Agent is installed as an auto-starting service, it will restart automatically if the system is rebooted.*

To confirm that Elastic Agent is installed and running, go to the **Agents** tab in Fleet.

If you run into problems:

- Check the Elastic Agent logs. If you use the default policy, agent logs and metrics are collected automatically unless you change the default settings.

Step 4: Install APM agents

APM agents are written in the same language as your service. To monitor a new service, you must install the agent and configure it with a service name, APM Server host, and Secret token.

- **Service name**: The APM integration maps an instrumented service's name–defined in each APM agent's configuration– to the index that its data is stored in Elasticsearch. Service names are case-insensitive and

must be unique. For example, you cannot have a service named Foo and another named foo. Special characters will be removed from service names and replaced with underscores (_).

- APM Server URL: The host and port that APM Server listens for events on. This should match the host and port defined when setting up the APM integration.
- Secret token: Authentication method for APM agent and APM Server communication. This should match the secret token defined when setting up the APM integration.

PHP

Install the agent

Install the PHP agent using one of the published packages.

To use the DEB package (Debian and Ubuntu):

```
• dpkg -i <package-file>.deb
```

Configure the agent

Configure your agent inside of the php.ini file:

```
• elastic_apm.server_url=http://localhost:8200
• elastic_apm.secret_token=SECRET_TOKEN
• elastic_apm.service_name="My-service"
```

Step 5: View your data

Back in Kibana, under Observability, select APM. You should see application performance monitoring data flowing into the Elastic Stack!

f. Elasticsearch Hadoop [15]

Installation

Elasticsearch-hadoop binaries can be obtained either by downloading them from the elastic.co site as a ZIP (containing project jars, sources and documentation) or by using any Maven-compatible tool with the following dependency:

```
<dependency>
  <groupId>org.elasticsearch</groupId>
  <artifactId>elasticsearch-hadoop</artifactId>
  <version>8.6.1</version>
</dependency>
```

The jar above contains all the features of elasticsearch-hadoop and does not require any other dependencies at runtime; in other words it can be used as is.elasticsearch-hadoop binary is suitable for Hadoop 2.x (also known as YARN) environments. Support for Hadoop 1.x environments are deprecated in 5.5 and will no longer be tested against in 6.0.

Minimalistic binaries

In addition to the uber jar, elasticsearch-hadoop provides minimalistic jars for each integration, tailored for those who use just one module (in all other situations the uber jar is recommended); the jars are smaller in size and use a dedicated pom, covering only the needed dependencies. These are available under the same groupId, using an artifactId with the pattern elasticsearch-hadoop-{integration}:

Map/Reduce.

```
<dependency>
  <groupId>org.elasticsearch</groupId>
  <artifactId>elasticsearch-hadoop-mr</artifactId>
  <version>8.6.1</version>
</dependency>
```

Apache Hive.

```
<dependency>
  <groupId>org.elasticsearch</groupId>
  <artifactId>elasticsearch-hadoop-hive</artifactId>
  <version>8.6.1</version>
</dependency>
```

Apache Pig.

```
<dependency>
  <groupId>org.elasticsearch</groupId>
  <artifactId>elasticsearch-hadoop-pig</artifactId>
  <version>8.6.1</version>
</dependency>
```

Apache Spark.

```
<dependency>
  <groupId>org.elasticsearch</groupId>
  <artifactId>elasticsearch-spark-30_2.12</artifactId>
  <version>8.6.1</version>
</dependency>
```

Storm

```
<dependency>
  <groupId>org.elasticsearch</groupId>
  <artifactId>elasticsearch-storm</artifactId>
```

```
 <version>8.6.1</version>
</dependency>
```

Development Builds

Development (or nightly or snapshots) builds are published daily at sonatype-oss repository (see below). Make sure to use snapshot versioning:

```
<dependency>
  <groupId>org.elasticsearch</groupId>
  <artifactId>elasticsearch-hadoop</artifactId>
  <version>{version}-SNAPSHOT</version>
</dependency>
```

but also enable the dedicated snapshots repository :

```
<repositories>
  <repository>
    <id>sonatype-oss</id>
    <url>http://oss.sonatype.org/content/repositories/snapshots</url>
    <snapshots><enabled>true</enabled></snapshots>
  </repository>
</repositories>\
```

4 CONFIGURE THE APPLIANCE [16]

When the base project deployment is complete, we need to configure the appliance to deploy the workload with good defaults and requires very little configuration. The configuration files should contain settings which are node-specific (such as node.name and paths), or settings which a node requires in order to be able to join a cluster, such as cluster.name and network. Host.

Config file's location

Elasticsearch has three configuration files, they are listed below:

- elasticsearch.yml for configuring Elasticsearch
- jvm.options for configuring Elasticsearch JVM settings
- log4j2.properties for configuring Elasticsearch logging

These files are in the config directory, whose default location depends on whether or not the installation is from an archive distribution (tar.gz or zip) or a package distribution (Debian or RPM packages).

For the archive distributions, the config directory location defaults to $ES_HOME/config. The location of the config directory can be changed via the ES_PATH_CONF environment variable as follows:

```
ES_PATH_CONF=/path/to/my/config ./bin/elasticsearch
```

Alternatively, you can export the ES_PATH_CONF environment variable via the command line or via your shell profile.

For the package distributions, the config directory location defaults to /etc/elasticsearch. The location of the config directory can also be changed via the ES_PATH_CONF environment variable, but note that setting this in your shell is not sufficient. Instead, this variable is sourced from /etc/default/elasticsearch (for the Debian package) and /etc/sysconfig/elasticsearch (for the RPM package). You will need to edit the ES_PATH_CONF=/etc/elasticsearch entry in one of these files accordingly to change the config directory location.

Config File Format

To configure the file format of Kibana, ELK 1, ELK2 and ELK 3 please go through the below documents which are listed below in Addendum.

I. Kibana Configuration
II. Elasticsearch-ELK 1Configuration
III. Elasticsearch-ELK 2 Configuration
IV. Elasticsearch-ELK 3 Configuration

The configuration format is YAML. Here is an example of changing the path of the data and logs directories:

```
path:
    data: /var/lib/elasticsearch
    logs: /var/log/elasticsearch
```

Settings can also be flattened as follows:

```
1. path.data: /var/lib/elasticsearch
2. path.logs: /var/log/elasticsearch
```

In YAML, you can format non-scalar values as sequences:

```
1. discovery.seed_hosts:
2.     - 192.168.1.10:9300
3.     - 192.168.1.11
4.     - seeds.mydomain.com
```

Though less common, you can also format non-scalar values as arrays:

```
1. discovery.seed_hosts: ["192.168.1.10:9300", "192.168.1.11", "seeds.mydomain.com"]
```

Environment variable substitution

Environment variables referenced with the `${...}` notation within the configuration file will be replaced with the value of the environment variable. For example:

```
1. node.name:    ${HOSTNAME}
2. network.host: ${ES_NETWORK_HOST}
```

Values for environment variables must be simple strings. Use a comma-separated string to provide values that Elasticsearch will parse as a list. For example, Elasticsearch will split the following string into a list of values for the `${HOSTNAME}` environment variable:

```
1. export HOSTNAME="host1,host2"
```

Important Elasticsearch configuration [17]

Elasticsearch requires very little configuration to get started, but there are a number of items which **must** be considered before using your cluster in production:

- Path settings
- Cluster name setting
- Node name setting
- Network host settings
- Discovery settings
- JVM fatal error log setting

In Elastic Cloud service configures these items automatically, making your cluster production-ready by default.

Path Settings

Elasticsearch writes the data you index to indices and data streams to a `data` directory. Elasticsearch writes its own application logs, which contain information about cluster health and operations, to a `logs` directory.

In production, we strongly recommend you set the path.data and path.logs in elasticsearch.yml to locations outside of $ES_HOME. Supported path.data and path.logs values vary by platform:

Unix-like systems

```
path:
  data: /var/data/elasticsearch
  logs: /var/log/elasticsearch
```

Cluster name setting

A node can only join a cluster when it shares its cluster.name with all the other nodes in the cluster. The default name is elasticsearch, but you should change it to an appropriate name that describes the purpose of the cluster.

```
cluster.name: logging-prod
```

Node name setting

Elasticsearch uses node.name as a human-readable identifier for a particular instance of Elasticsearch. This name is included in the response of many APIs. The node name defaults to the hostname of the machine when Elasticsearch starts, but can be configured explicitly in elasticsearch.yml:

```
node.name: prod-data-2
```

Network host setting

By default, Elasticsearch only binds to loopback addresses such as 127.0.0.1 and [::1]. This is sufficient to run a cluster of one or more nodes on a single server for development and testing, but a resilient production cluster must involve nodes on other servers. There are many network settings but usually all you need to configure is network.host:

```
network.host: 192.168.1.10
```

Discovery and cluster formation settings

Configure two important discovery and cluster formation settings before going to production so that nodes in the cluster can discover each other and elect a master node.

discovery.seed_hosts

When you want to form a cluster with nodes on other hosts, use the static discovery.seed_hosts setting. This setting provides a list of other nodes in the cluster that are master-eligible and likely to be live and contactable to seed the discovery process. This setting accepts a YAML sequence or array of the addresses of all the master-eligible nodes in the cluster. Each address can be either an IP address or a hostname that resolves to one or more IP addresses via DNS.

```
discovery.seed_hosts:
  - 192.168.1.10:9300
  - 192.168.1.11
  - seeds.mydomain.com
  - [0:0:0:0:0:ffff:c0a8:10c]:9301
```

cluster.initial_master_nodes

When you start an Elasticsearch cluster for the first time, a cluster bootstrapping step determines the set of master-eligible nodes whose votes are counted in the first election. In development mode, with no discovery settings configured, this step is performed automatically by the nodes themselves.

Because auto-bootstrapping is inherently unsafe, when starting a new cluster in production mode, you must explicitly list the master-eligible nodes whose votes should be counted in the very first election. You set this list using the cluster.initial_master_nodes setting.

```
discovery.seed_hosts:
  - 192.168.1.10:9300
  - 192.168.1.11
  - seeds.mydomain.com
  - [0:0:0:0:0:ffff:c0a8:10c]:9301
cluster.initial_master_nodes:
  - master-node-a
  - master-node-b
  - master-node-c
```

JVM fatal error log setting:

By default, Elasticsearch configures the JVM to write fatal error logs to the default logging directory. On RPM and Debian packages, this directory is /var/log/elasticsearch. On Linux and MacOS and Windows distributions, the logs directory is located under the root of the Elasticsearch installation.

These are logs produced by the JVM when it encounters a fatal error, such as a segmentation fault. If this path is not suitable for receiving logs, modify the -XX:ErrorFile=... entry in jvm.options.

5 TESTING THE APPLIANCE

For test the appliance, we have use Security app of the Elasticsearch which explain about the Elastic Security UI, equips teams to prevent, detect, and respond to threats at cloud speed and scale — securing business operations with a unified, open platform.

Elastic Security UI [18]

The Elastic Security app is a highly interactive workspace designed for security analysts that provides a clear overview of events and alerts from your environment. You can use the interactive UI to drill down into areas of interest.

Search

Filter for alerts, events, processes, and other important security data by entering Kibana Query language (KQL) queries in the search bar, which appears at the top of each page throughout the app. A date/time filter set to Today is enabled by default but can be changed to any time range.

Figure 4 Search bar

Navigation Menu

The navigation menu contains direct links and expandable groups, identified by the group icon.

Figure 5 Navigation Menu

- Click a top-level link to go directly to its landing page, which contains links and information for related pages.
- Click a group's icon ( ) to open its flyout menu, which displays links to related pages within that group. Click a link in the flyout to navigate to its landing page.
- Click the **Collapse side navigation** icon ( ) to collapse and expand the main navigation menu.

Elastic Security app pages

The Elastic Security app contains the following pages that enable analysts to view, analyze, and manage security data.

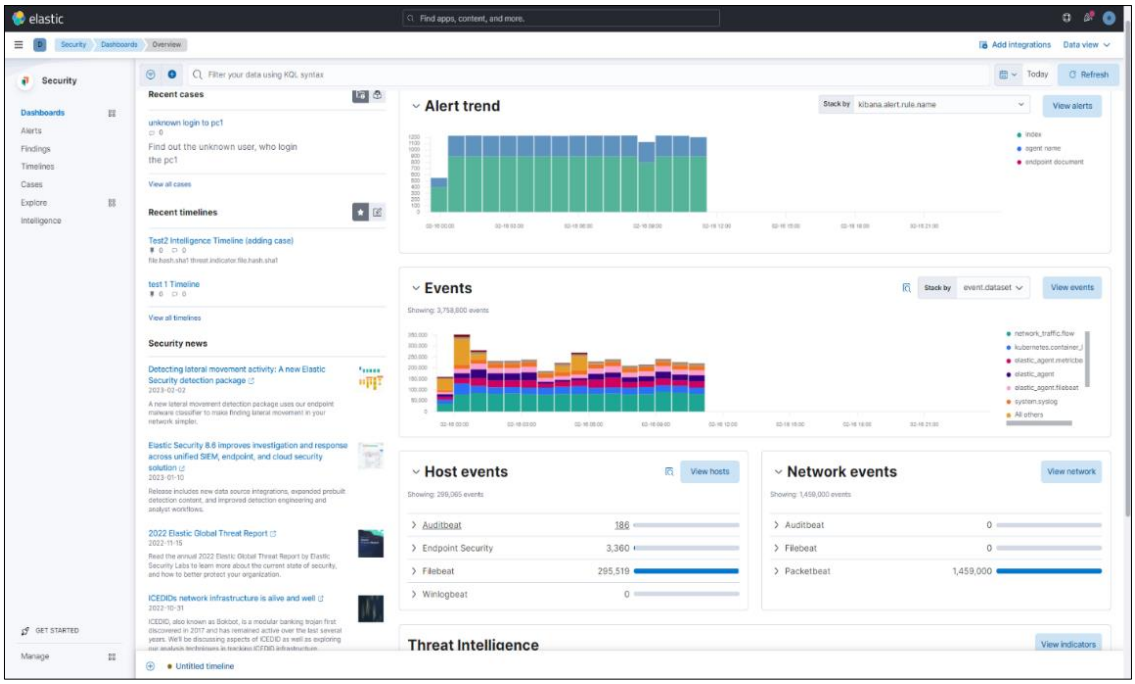


Figure 6 Security Dashboard

Dashboards

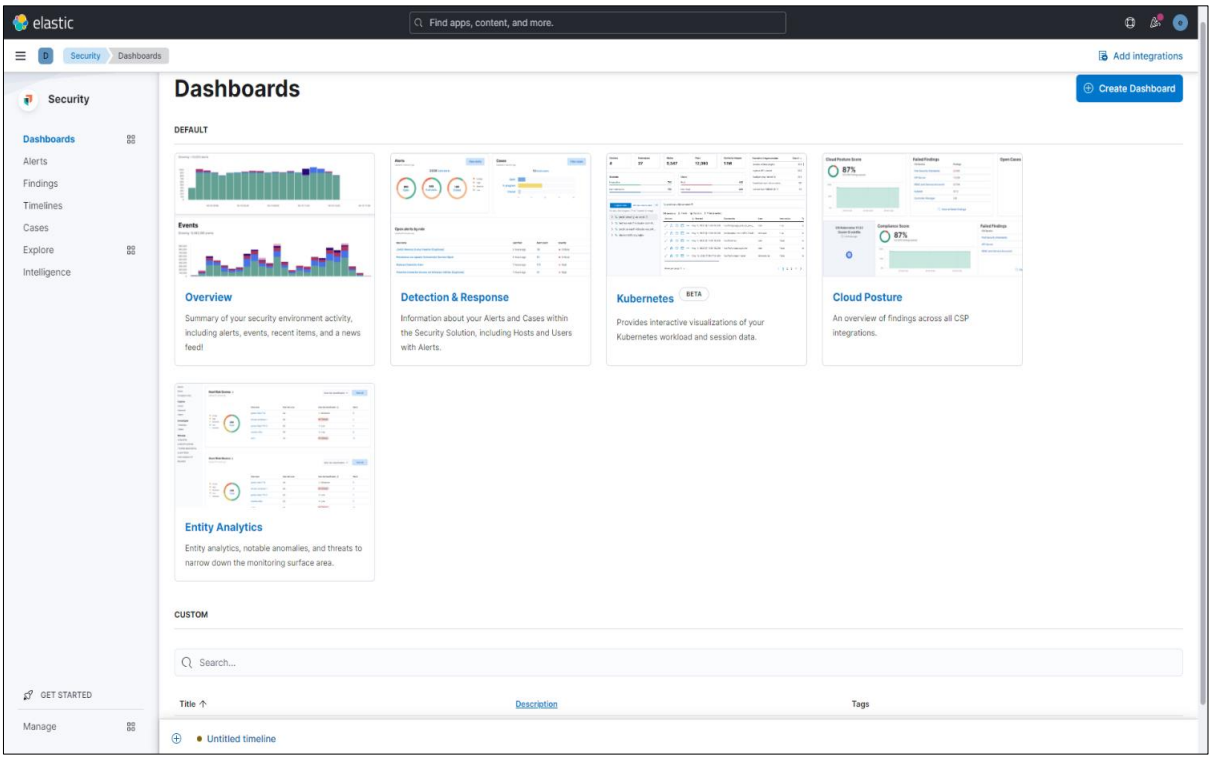


Figure 7 Overview Page

Expand this section to access the Overview, Detection & Response, Kubernetes, Cloud Posture, and Entity Analytics dashboards, which **provide interactive visualizations that summarize your data**

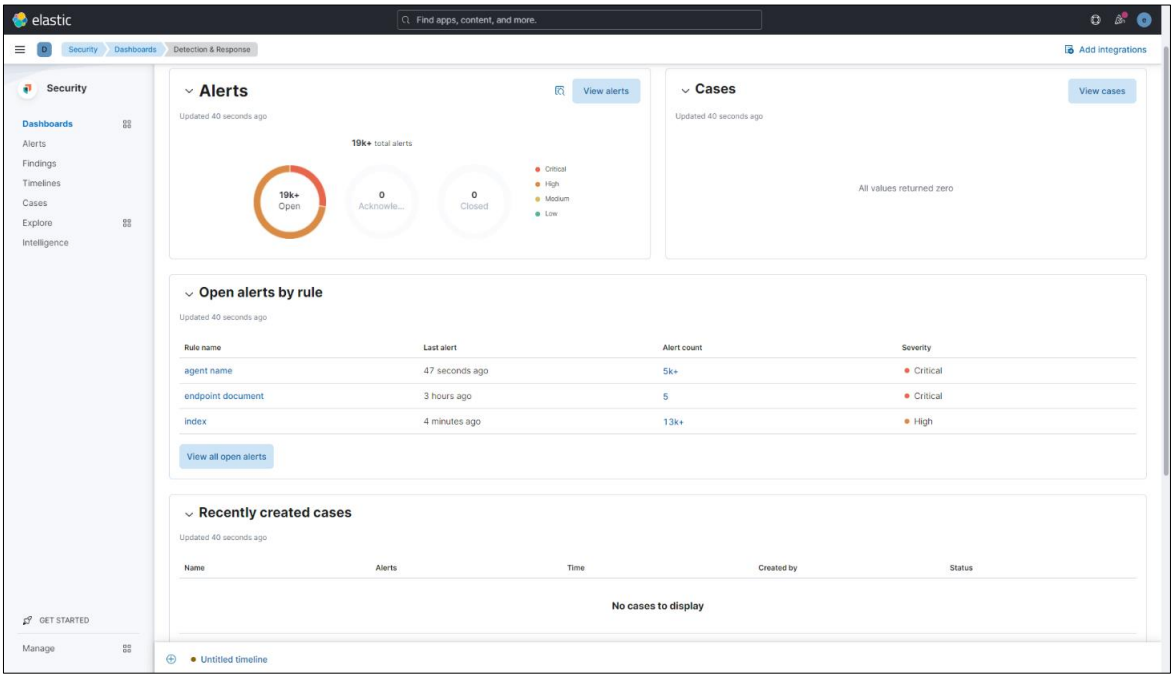
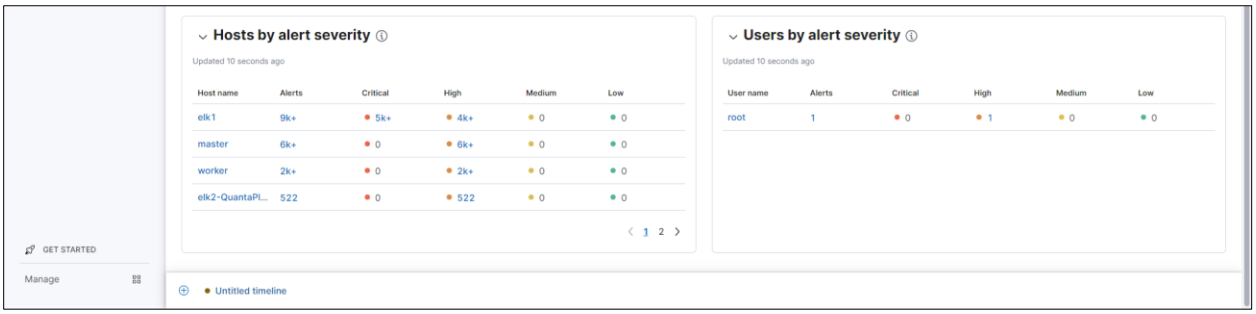


Figure 8 Detection and Response (1)



Error! Reference source not found. **Figure 9 Detection and Response (2)**

Alerts

View and manage alerts to monitor activity within your network.

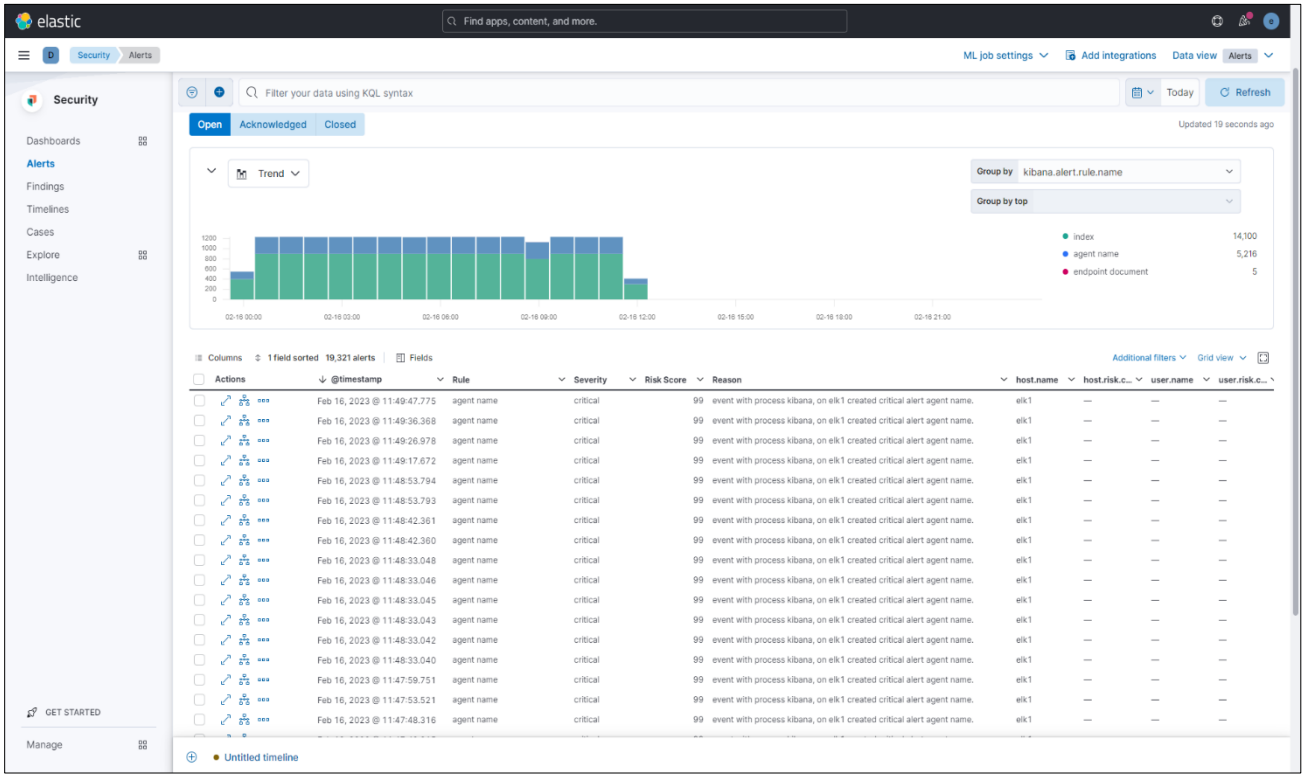


Figure 10 Security Alerts

Detections and alerts

Use the detection engine to create and manage rules and view the alerts these rules create. Rules periodically search indices (such as logs-* and filebeat-*) for suspicious source events and create alerts when a rule's conditions are met. When an alert is created, its status is Open. To help track investigations, an alert's status can be set as Open, Acknowledged, or Closed.

In addition to creating own rules, enable Elastic prebuilt rules to immediately start detecting suspicious activity. Once the prebuilt rules are loaded and running, Tune detection rules and add and manage exceptions explain how to modify the rules to reduce false positives and get a better set of actionable alerts. You can also use exceptions and value lists when creating or modifying your own rules.

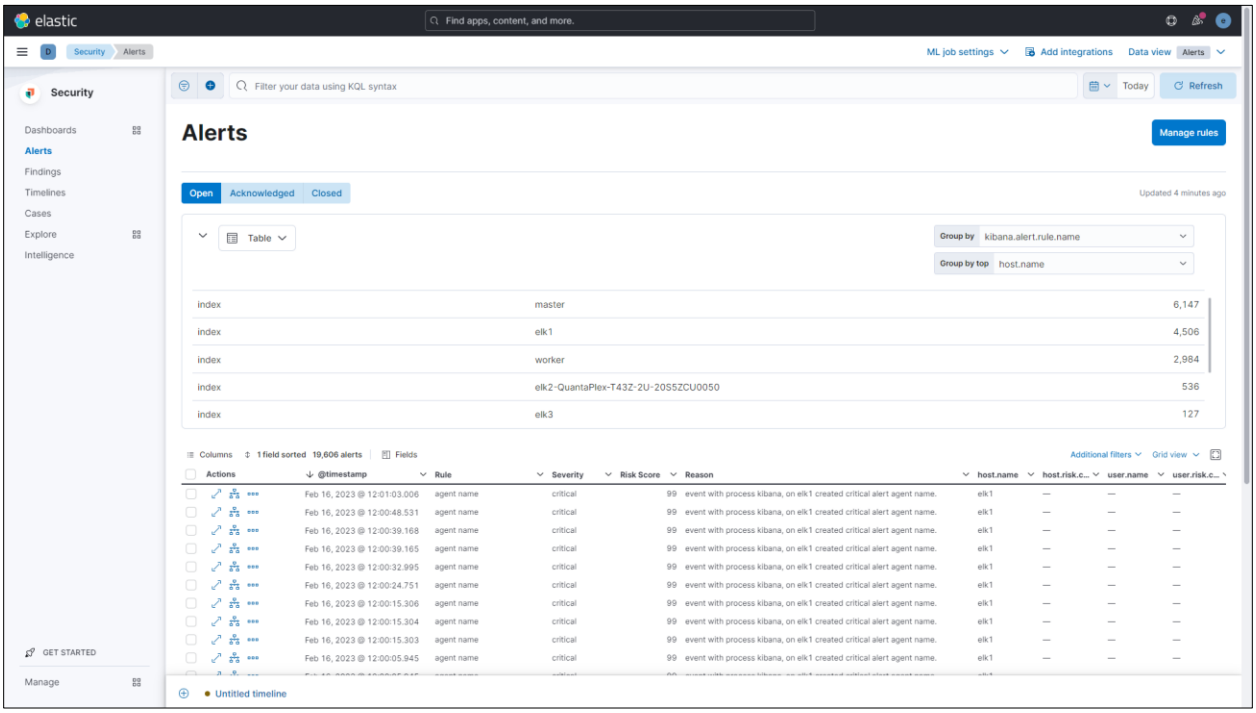


Figure 11 Alerts Status

There are two special prebuilt rules need to know about:

a. Endpoint Security: Automatically creates an alert from all incoming Elastic Endpoint alerts. To receive Elastic Endpoint alerts, you must install the Endpoint agent on your hosts.

   When this rule is enabled, the following Endpoint events are displayed as detection alerts:

- Malware Prevention Alert
- Malware Detection Alert

b. External Alerts: Automatically creates an alert for all incoming third-party system alerts (for example, Suricata alerts).

If you want to receive notifications via external systems, such as Slack or email, when alerts are created, use the Kibana Alerting and Actions framework. After rules have started running, you can monitor their executions to verify they are functioning correctly, as well as view, manage, and troubleshoot alerts. You can create and manage rules and alerts via the UI or the Detections API.

Findings

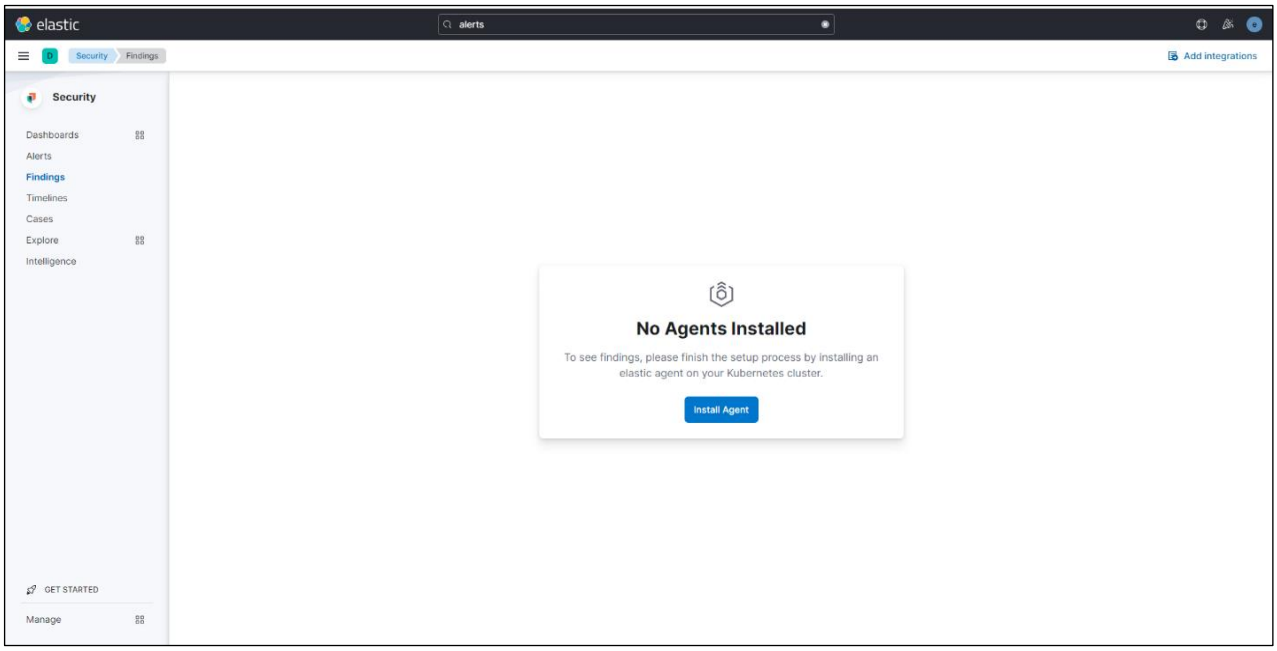


Figure 12 Finding Page

Compare your Kubernetes infrastructure against a variety of security benchmarks.

What are findings?

Findings indicate whether Kubernetes resources passed or failed evaluation against benchmark rules. Each finding includes metadata about the resource evaluated and the benchmark rule used to evaluate it. Each finding's result (pass or fail) indicates whether a particular part of your Kubernetes infrastructure meets a benchmark rule.

Group and filter findings

By default, the Findings page lists all findings, without grouping or filtering.

Group findings by resource

1. Select **Group by → Resource** to display a list of resources sorted by their total number of failed findings.
2. Click a resource ID to display the findings associated with that resource.

Filter findings

You can filter findings data in two ways:

- **The KQL search bar**: For example, search for `result.evaluation : failed` to view all failed findings.
- **In-table value filters**: Hover over a finding to display available inline actions. Use the Filter In (plus) and Filter Out (minus) buttons.

Remediate failed findings

To remediate failed findings and reduce your attack surface:

1. Navigate to the Findings page and filter for failed findings.
2. Click a failed finding to open the findings flyout.
3. Follow the steps under **Remediation**.

Timelines

Investigate alerts and complex threats — such as lateral movement — in your network. Timelines are interactive and allow you to share your findings with other team members.

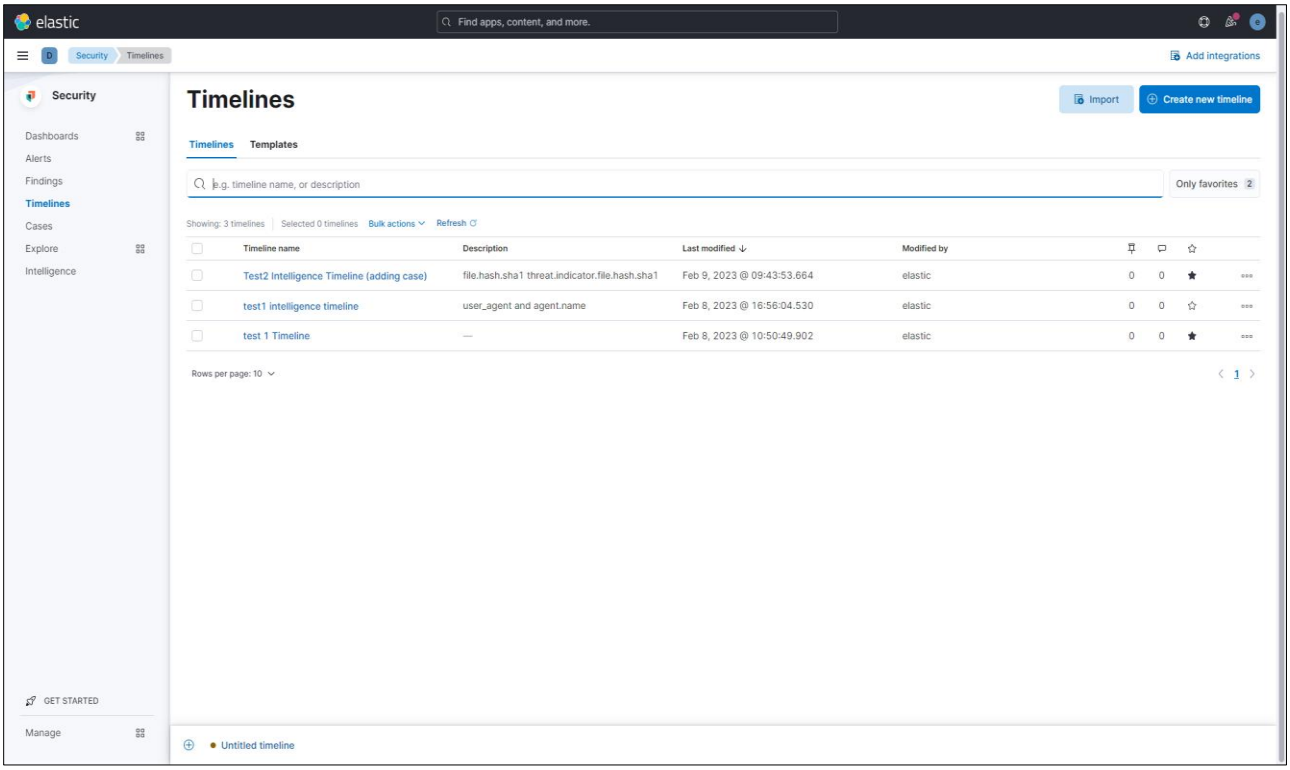


Figure 13 Timelines

You can drag or send fields of interest to a Timeline to create the desired query. For example, you can add fields from tables and histograms on the **Overview**, **Alerts**, **Hosts**, and **Network** pages, as well as from other Timelines. Alternatively, you can add a query directly in Timeline by clicking **+ Add field**

Timelines are responsive, and they persist as you move through the Elastic Security app collecting data. Auto-saving ensures that the results of your investigation are available for later review. To record and share your findings with others, attach your Timeline to a case.Timeline templates allow you to define the source event fields used when you investigate alerts in Timeline. You can select whether the fields use predefined values or values retrieved from the alert.

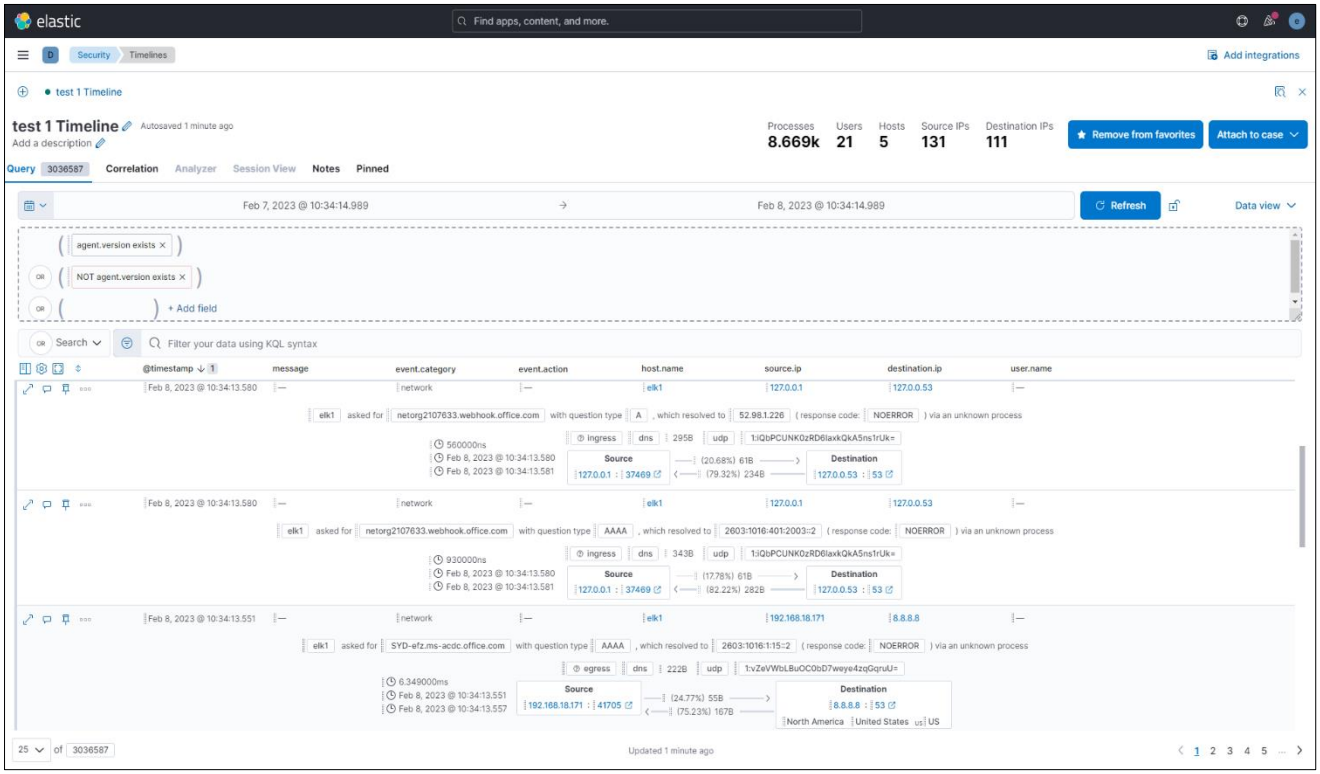


Figure 14 Generic Network Timeline (1)

View and refine Timeline results

You can select whether Timeline displays detection alerts and other raw events, or just alerts. By default, Timeline displays both raw events and alerts. To hide raw events and display alerts only, click Data view to the right of the date and time picker, then select Show only detection alerts.

Inspect an event or alert

To further inspect an event or detection alert, click the View details button. A flyout with event or alert details appears.

Configure Timeline event context and display

Many types of events automatically appear in preconfigured views that provide relevant contextual information, called Event Renderers. You can display and turn them on or off with the Settings menu in the upper left corner of the results pane:

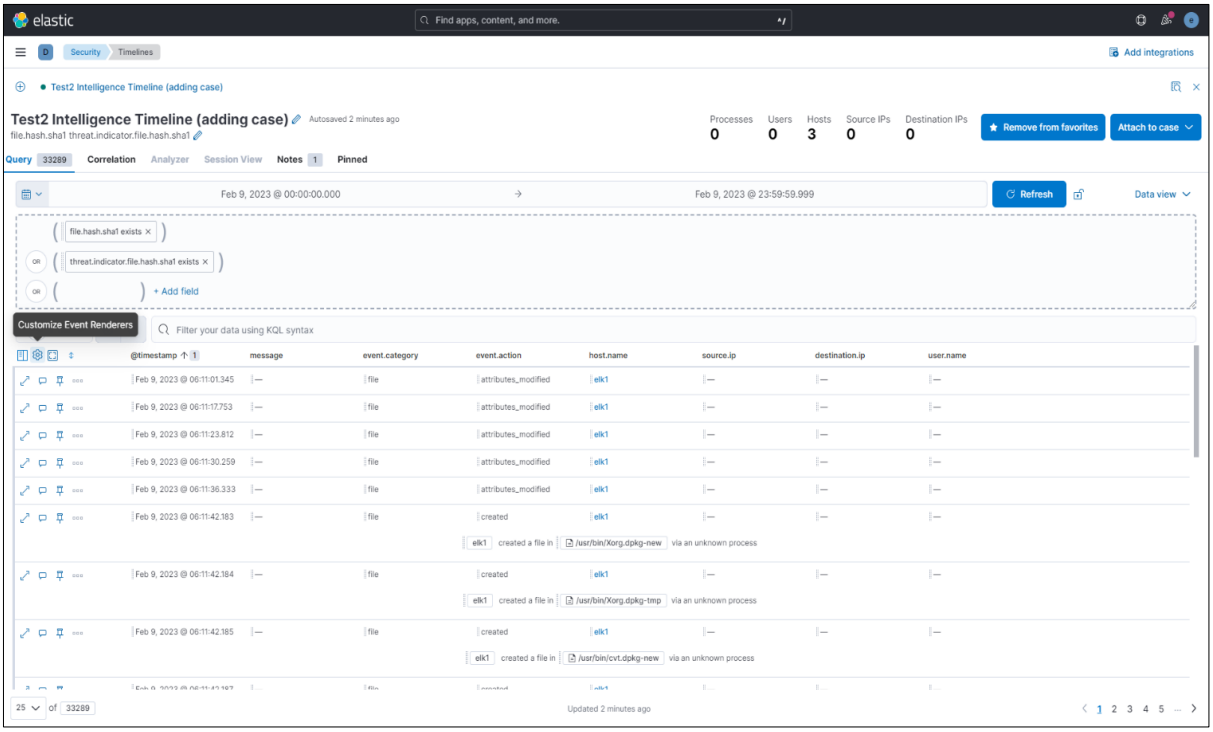


Figure 15 Timeline events (2)

Figure 16 Event renderers

The example above displays the Flow event renderer, which highlights the movement of data between its source and destination. If you see a particular part of the rendered event that interests you, you can drag it up to the drop zone below the query bar for further investigation.

You can also modify a Timeline's display in other ways:

- Add, remove, reorder, or resize columns
- Create runtime fields and display them in the Timeline
- View the Timeline in full screen mode
- Add notes to individual events
- Add investigation notes to the entire Timeline
- Pin interesting events to the Timeline

Narrow or expand your KQl query

By placing fields within the drop zone, you turn them into query filters. Their relative placement specifies their logical relationships: horizontally adjacent filters use AND, while vertically adjacent filters use OR.

Edit existing filters

Click a filter to access additional operations such as Add filter, Clear all, Load saved query, and more.

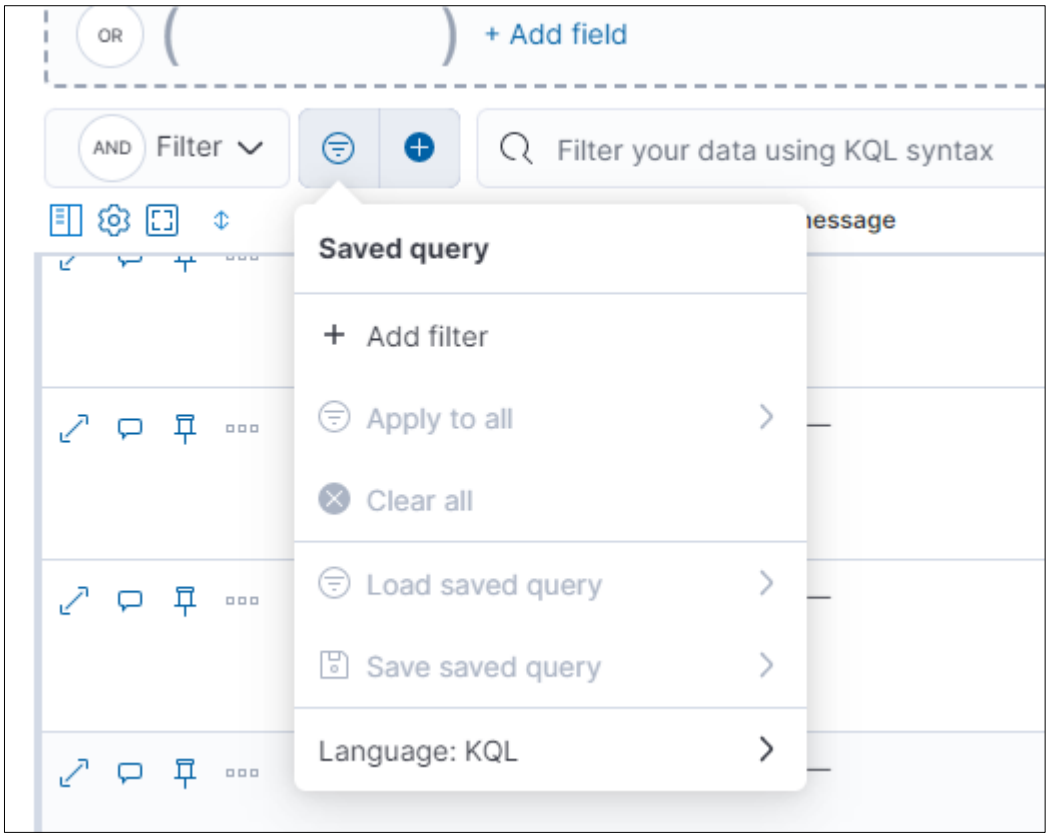


Figure 17 Edit existing filters

Attach Timeline to a case

To attach a Timeline to a new or existing case, open it, click Attach to case in the upper right corner, then select either **Attach to new case or Attach to existing case**.

Manage existing Timelines

You can view, duplicate, export, delete, and create templates from existing Timelines:

1. Go to Timelines.
2. Click the All actions menu in the desired row, then select an action:
    - Create template from timeline (refer to *About Timeline templates*)
    - Duplicate timeline
    - Export selected (refer to Export and import Timelines)
    - Delete selected

Export and import Timelines

You can export and import Timelines, which enables you to share Timelines from one Kibana space or instance to another. Exported Timelines are saved as .ndjson files.
To export Timelines:

- Go to **Timelines**.
- Either click the **All actions** menu in the relevant row and select **Export selected**, or select multiple Timelines and then click **Bulk actions** → **Export selected**.

To import Timelines:

- Click **Import**, then select or drag and drop the relevant .ndjson file.

Filter Timeline results with EQL

When forming EQL queries, you can write a basic query to return a list of events and alerts. Or, you can create sequences of EQL queries to view matched, ordered events across multiple event categories. Sequence queries are useful for identifying and predicting related events. They can also provide a more complete picture of potential adversary behavior in your environment, which you can use to create or update rules and detection alerts.

Figure 18 Filter Timeline results with EQL

From the **Correlation** tab, you can also do the following:

- Specify the date and time range that you want to investigate.
- Reorder the columns and choose which fields to display.
- Choose a data view and whether to show detection alerts only.

Cases

Collect and share information about security issues by opening a case in Elastic Security. Cases allow you to track key investigation details, collect alerts in a central location, and more. The Elastic Security UI provides several ways to create and manage cases. Alternatively, you can use the cases API to perform the same tasks. You can also send cases to these external systems by configuring external connectors:

- ServiceNow ITSM
- ServiceNow SecOps
- Jira (including Jira Service Desk)
- IBM Resilient
- Swimlane

- Webhook - Case Management

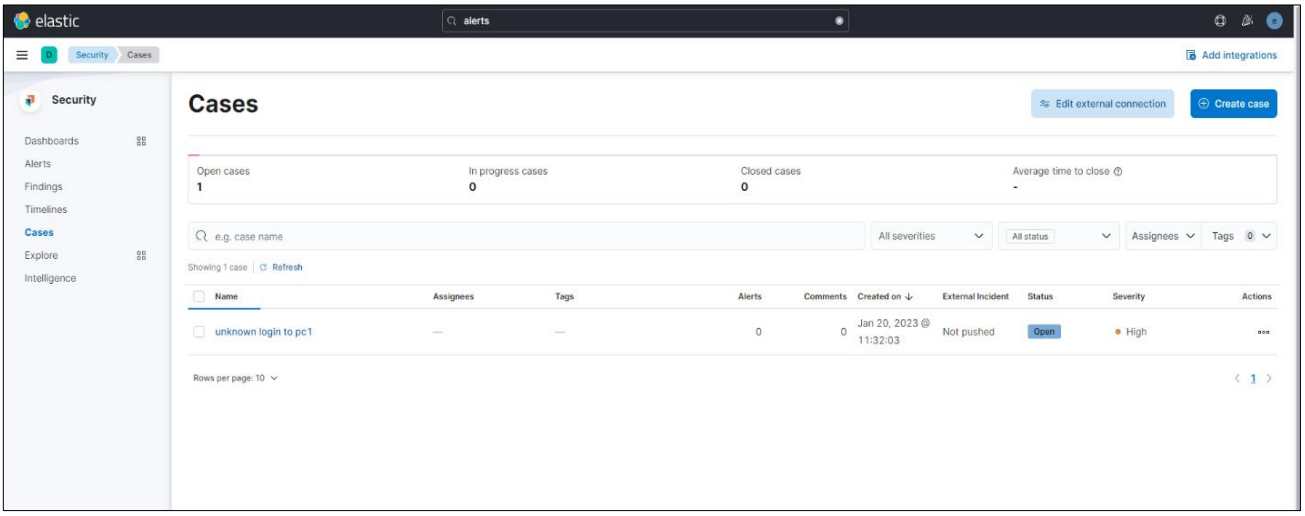


Figure 19 Cases Page

Explore

Expand the Explore page to view Hosts, Networks and Users pages.

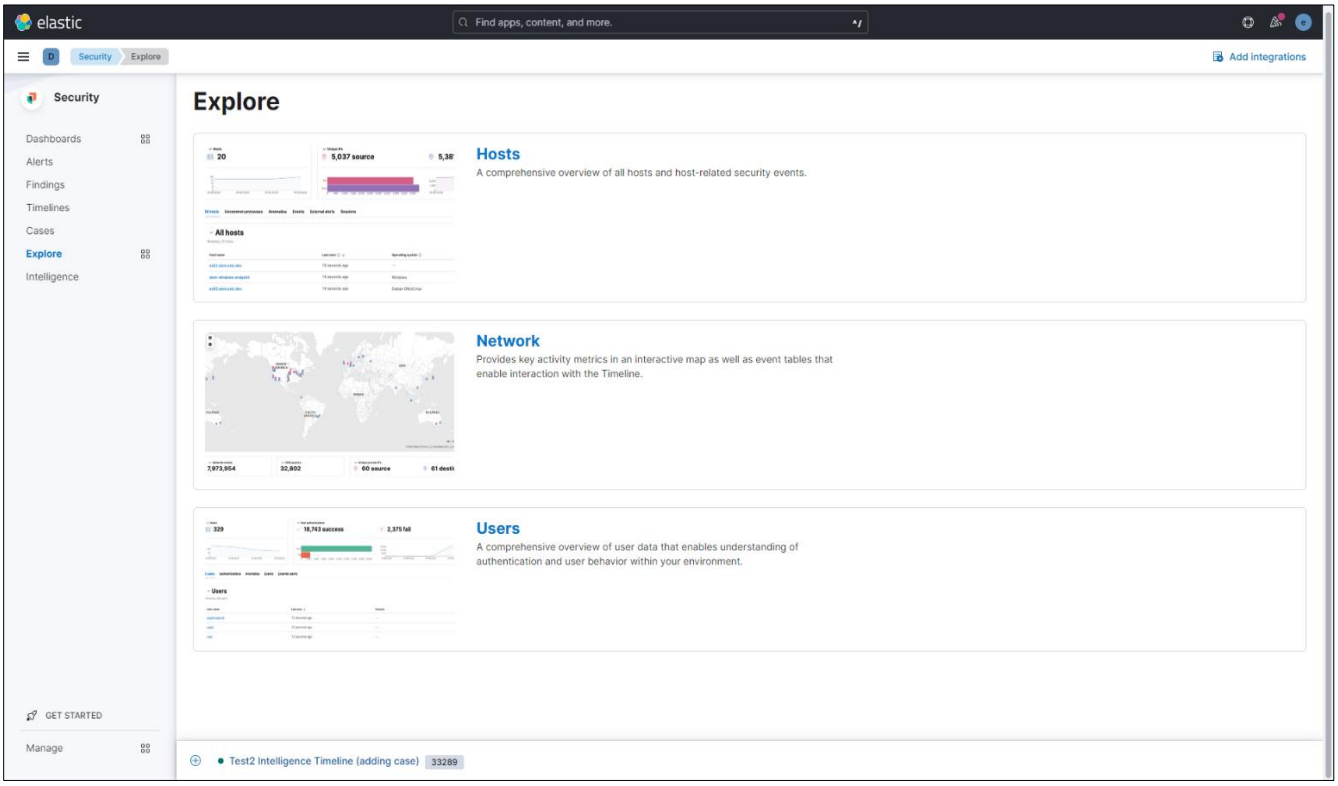


Figure 20 Explore Page

Hosts:

Examine key metrics for host-related security events using graphs, charts, and interactive data tables.

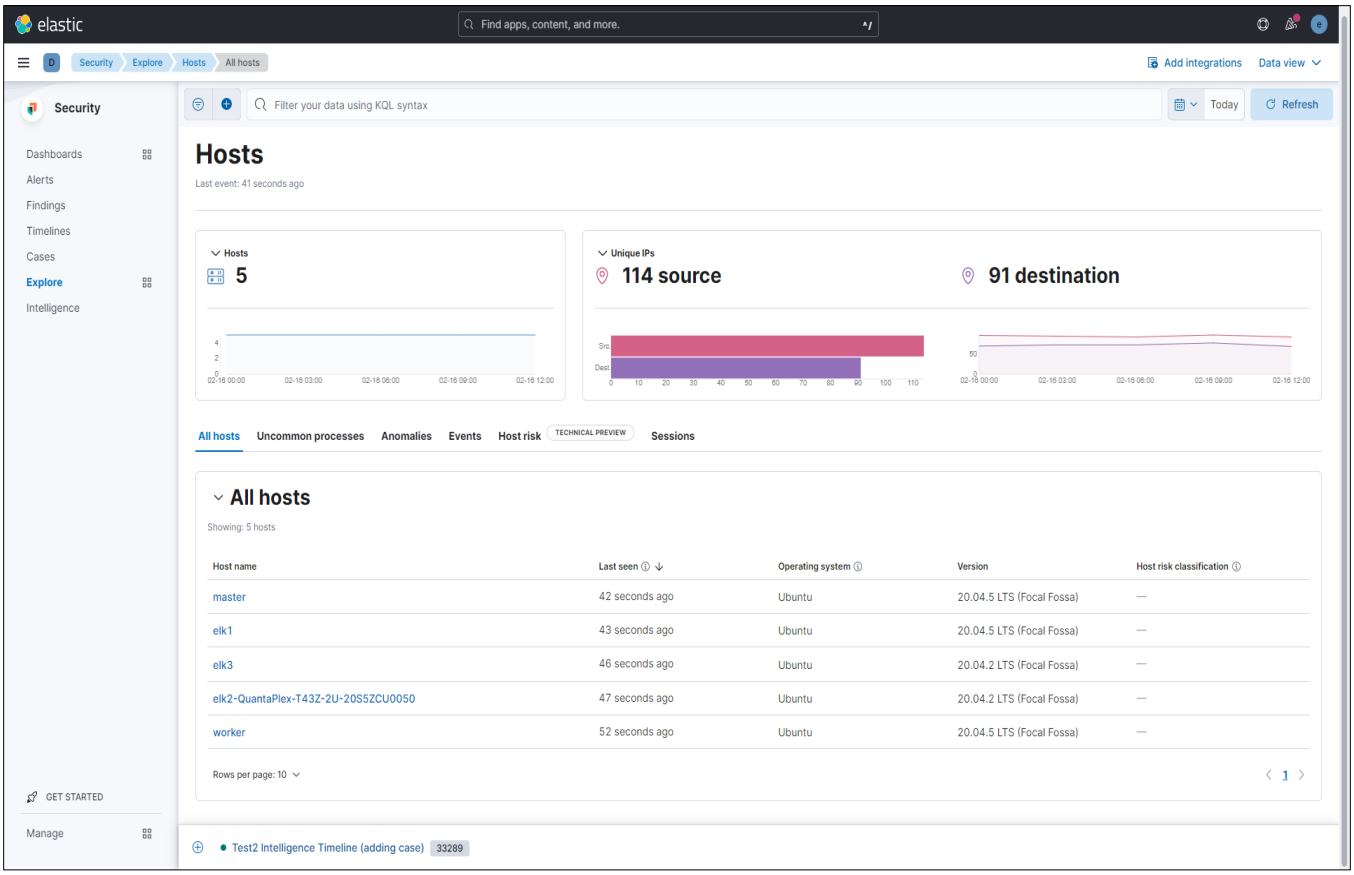


Figure 21 Hosts

Network

Explore the interactive map to discover key network activity metrics and investigate network events further in Timeline.

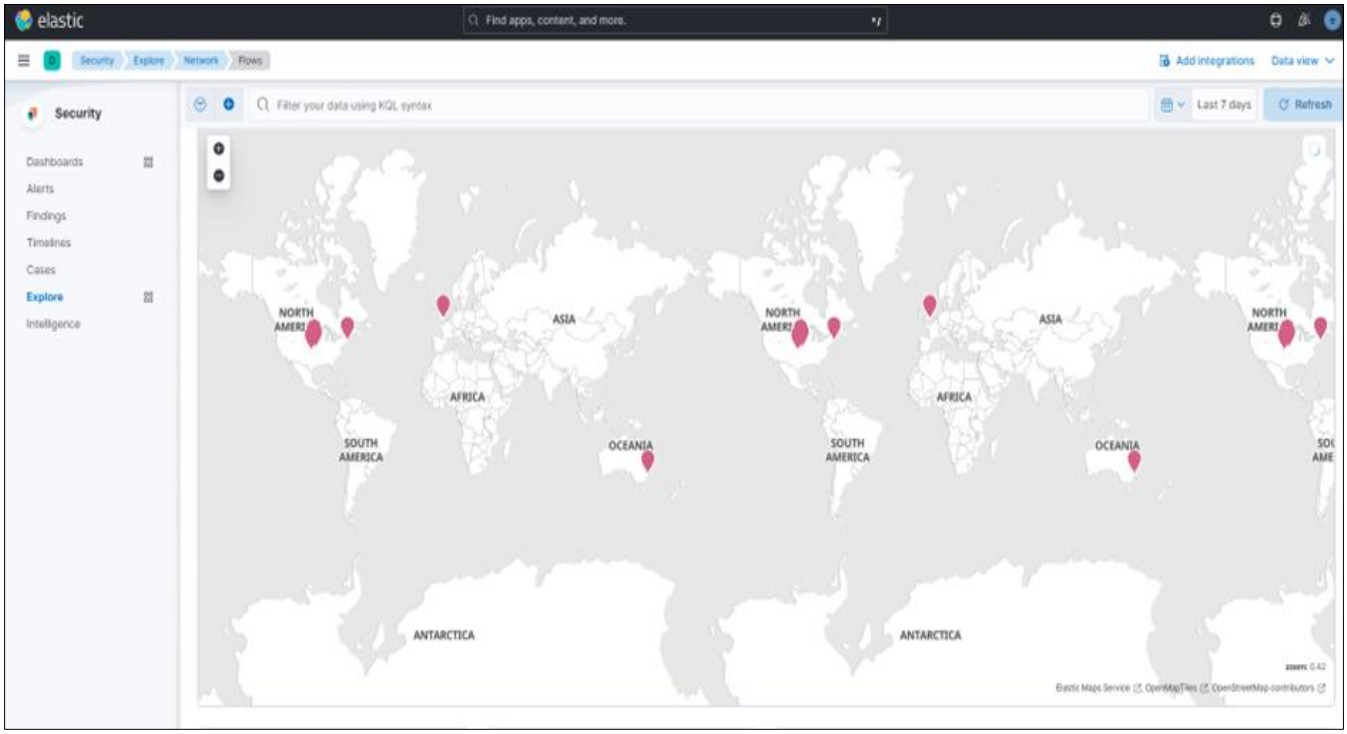


Figure 22 Network (1)

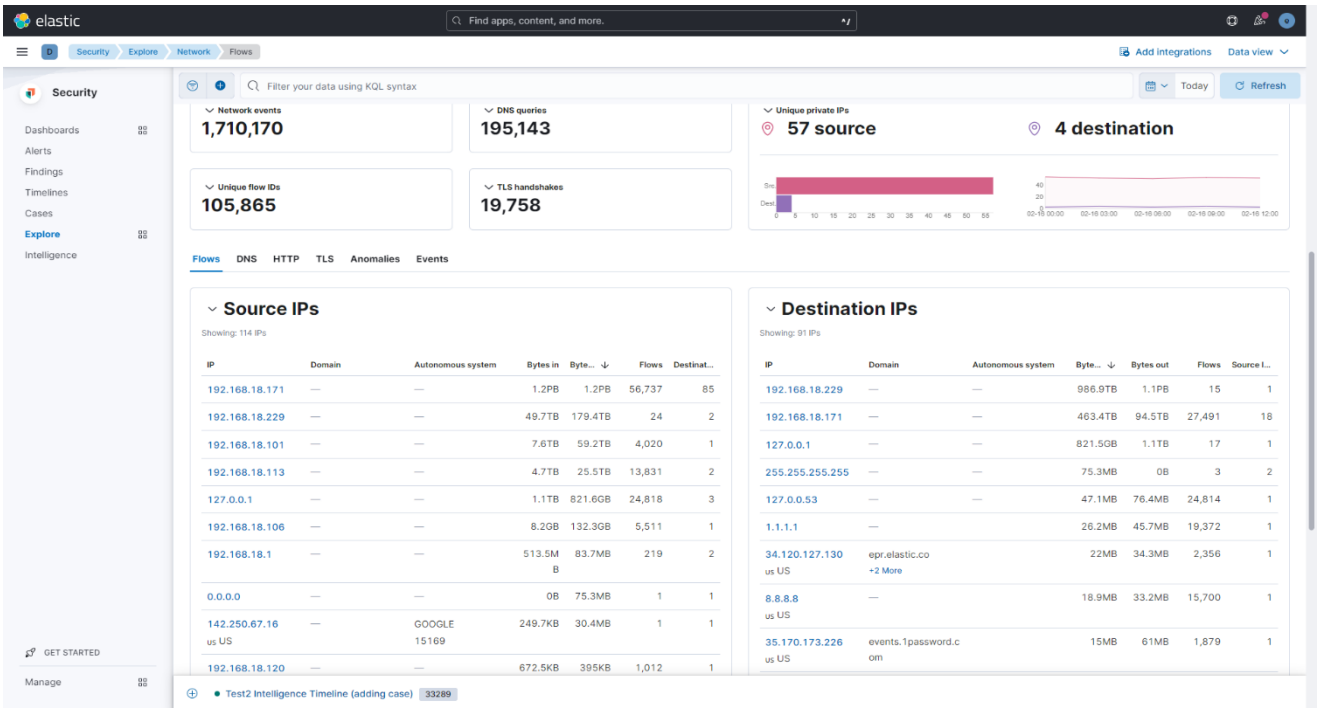


Figure 73 Network (2)

Users:

Access a comprehensive overview of user data to help you understand authentication and user behaviour within your environment.

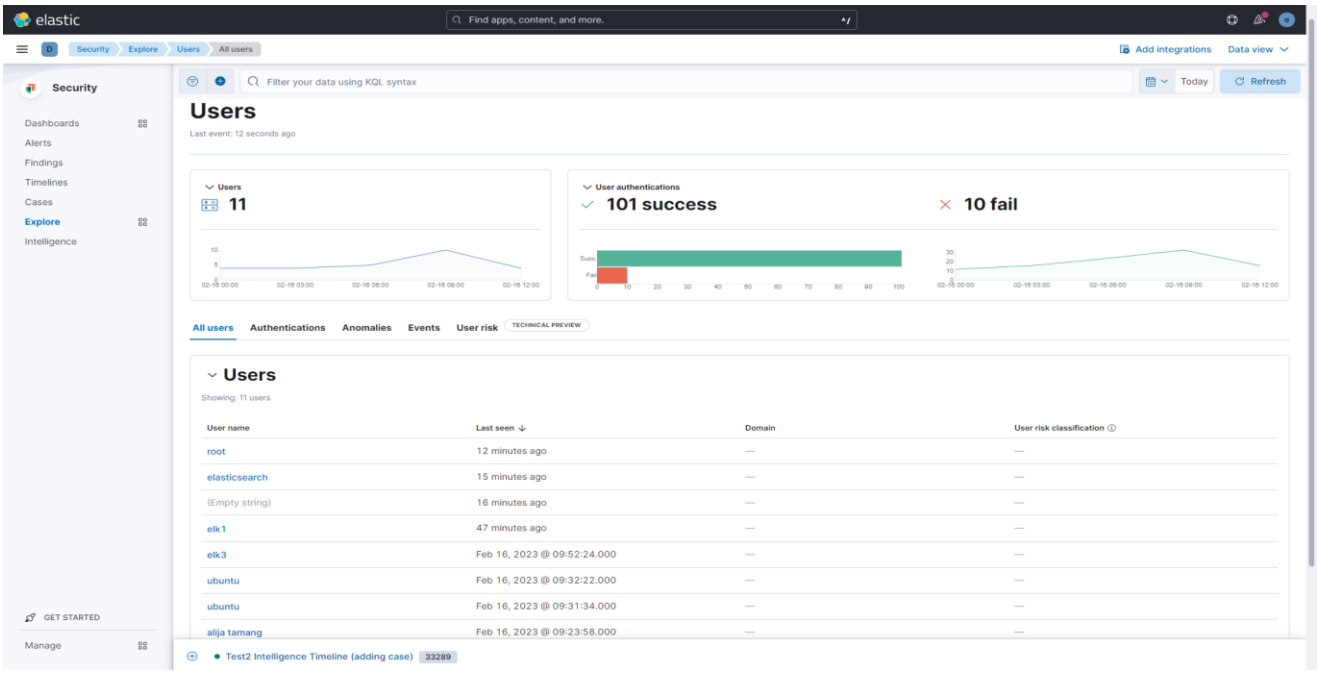


Figure 24 Users

Intelligence

The Intelligence section contains the indicators page, which collects data from enabled threat intelligence feeds and provides a centralized view of indicators of compromise (IoCs).

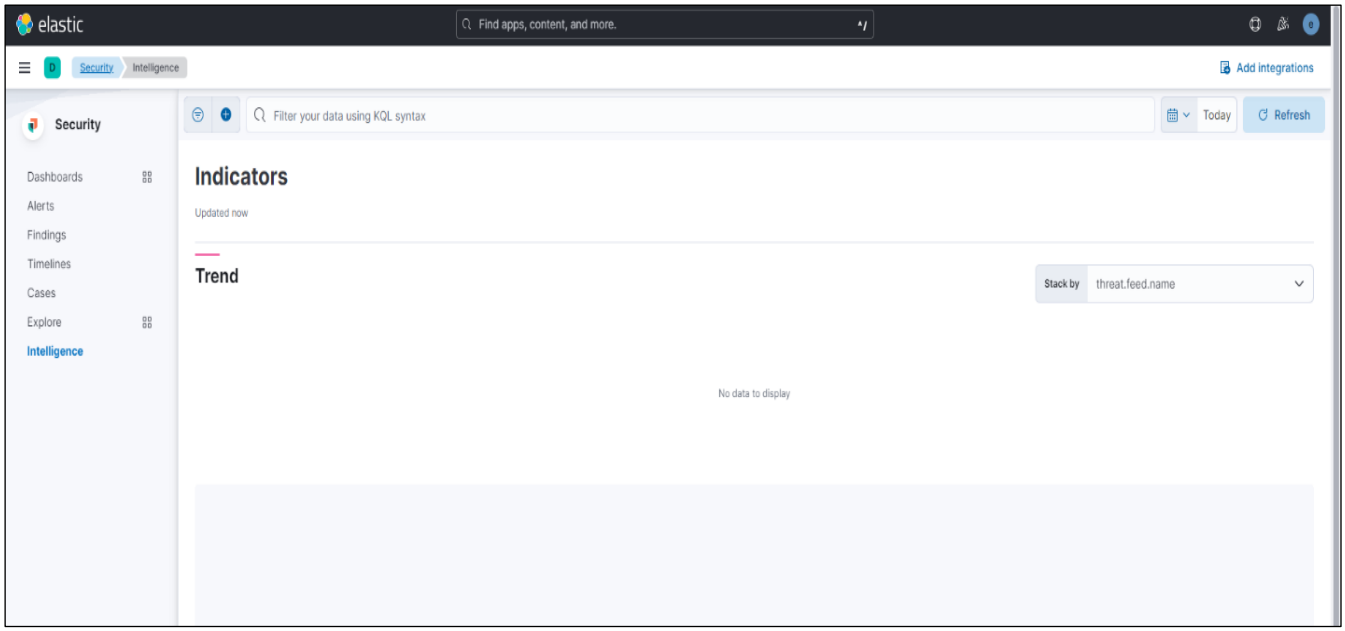


Figure 25 Intelligence Page

Threat intelligence and indicators

Threat intelligence is a research function that analyzes current and emerging threats and recommends appropriate actions to strengthen a company's security posture. Threat intelligence requires proactivity to be useful, such as gathering, analyzing, and investigating various threat and vulnerability data sources.

Set up the indicators page

Install a threat intelligence integration to add indicators to the indicators page.

1. Choose one of the following:

    - From the Elastic Security app main menu, go to **Intelligence** → **Indicators** → **Add Integrations**.
    - From the Kibana main menu, click **Add integrations**. Scroll down the list of integration categories and select **Threat Intelligence** to filter by threat intelligence integrations.
2. Select a threat intelligence integration, then complete the integration's guided installation.

3. Return to the Indicators page in Elastic Security. Refresh the page if indicator data isn't displaying.

Troubleshooting

If indicator data is not appearing in the indicators table after you installed a threat intelligence integration:

- Verify that the index storing indicator documents is included in the default Elastic Security indices (securitySolution:defaultIndex). The index storing indicator documents will differ based on the way you're collecting indicator data:

- **Elastic Agent integrations** - logs_ti*
- **Filebeat integrations** - filebeat-*

- Ensure the indicator data you're ingesting is mapped to Elastic Common Schema (ECS).

Indicators page UI

After you add indicators to the indicators page, you can examine, search, filter, and take action on indicator data. Indicators also appear in the Trend view, which shows the total values in the legend.

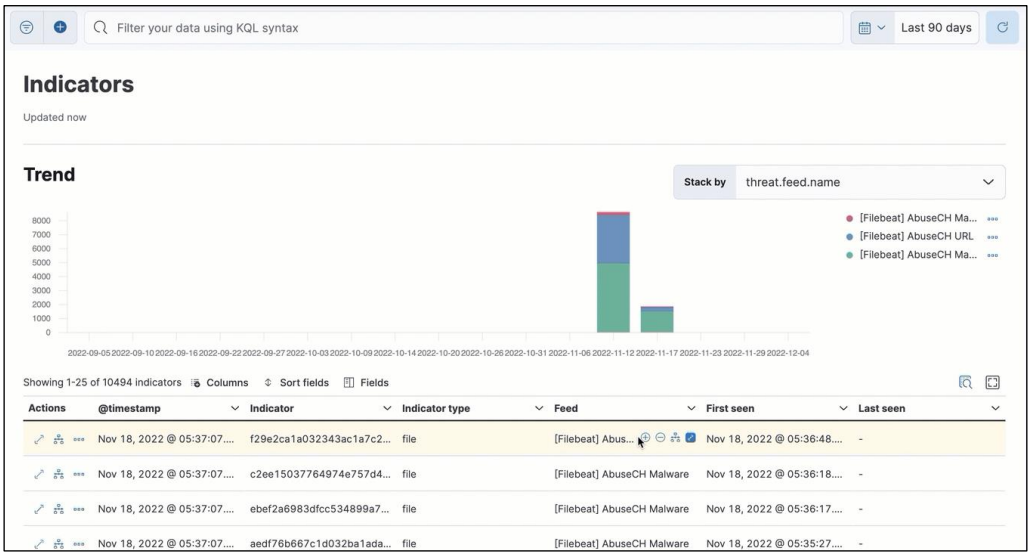


Figure 26 Indicators

Examine indicator details

The indicator page contains the informational like a summary of the indicator, including the indicator's name, the threat intelligence feed it came from, the indicator type, and additional relevant data.

Table: The indicator data in table format.

JSON: The indicator data in JSON format.

Figure 27 Indicator details

Find related security events

Investigating an indicator in Timeline helps you find related security events in your environment. You can add an indicator to Timeline from the Indicators table or the indicator details flyout.

When you add an indicator to Timeline, a new Timeline opens with a pre-populated KQL query. The query contains the indicator field-value pair that you selected plus the field-value pair of the mapped source event.

For example, imagine you've added this file hash indicator to Timeline:

threat.indicator.file.hash.sha256 : c207213257a63589b1e1bd2f459b47becd000c1af8ea7983dd9541aff145c3ba

A new Timeline opens with an automatically populated KQL query. The query contains the indicator field-value pair (mentioned previously) and the mapped source event field-value pair, which is:

file.hash.sha256 : c207213257a63589b1e1bd2f459b47becd000c1af8ea7983dd9541aff145c3ba.

Using a KQL query that includes both the indicator and source event allows Timeline to find all events and alerts that have matching field-value pairs.

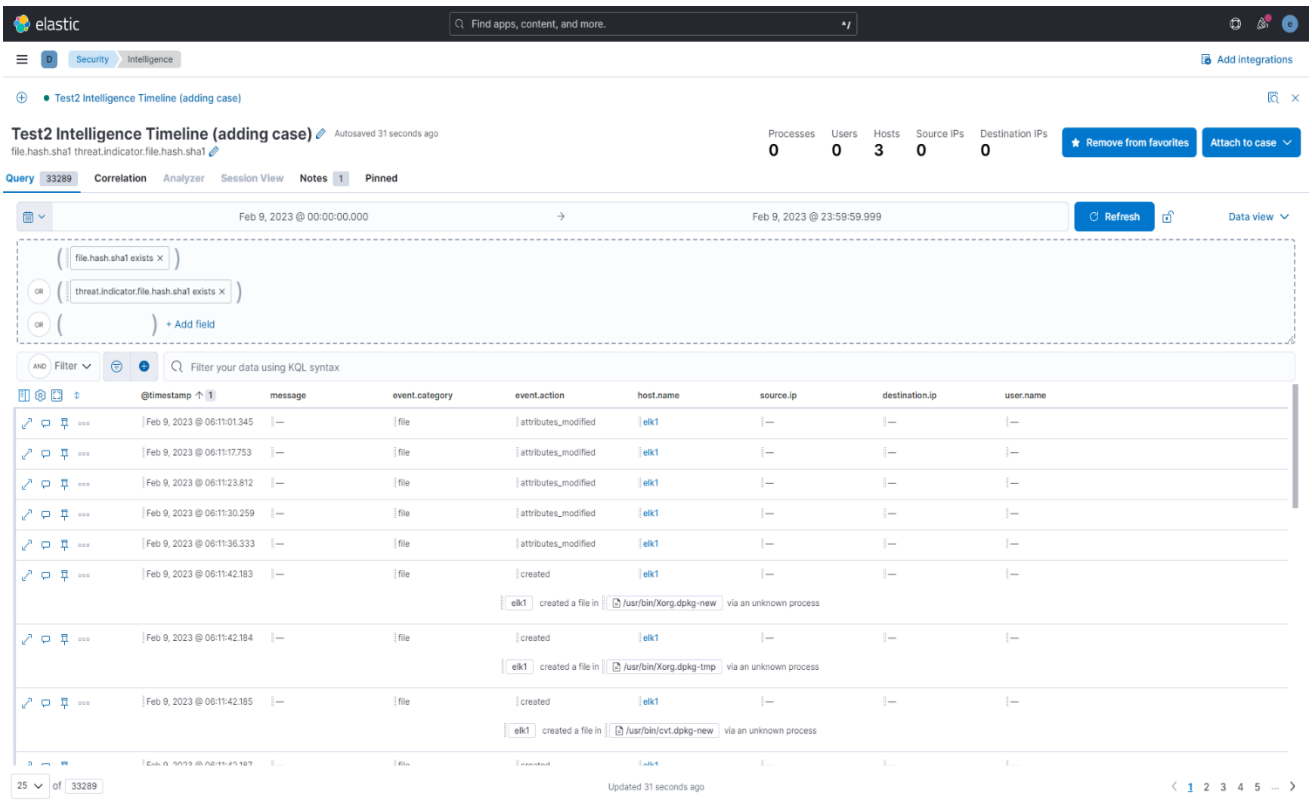


Figure 28 Investigating an Indicator in Timeline

Attach indicators to cases

Attaching indicators to cases provides more context and available actions for your investigations. This feature allows you to easily share or escalate threat intelligence to other teams.

To add indicators to cases:

1. From the Indicators table, click the **More actions menu** (…). Alternatively, open an indicator's details, then select **Take action**.
2. Select one of the following:

- **Add to existing case**: From the **Select case** dialog box, select the case to which you want to attach the indicator.
- **Add to new case**: Configure the case details.

The indicator is added to the case as a new comment.

Review indicator details in cases

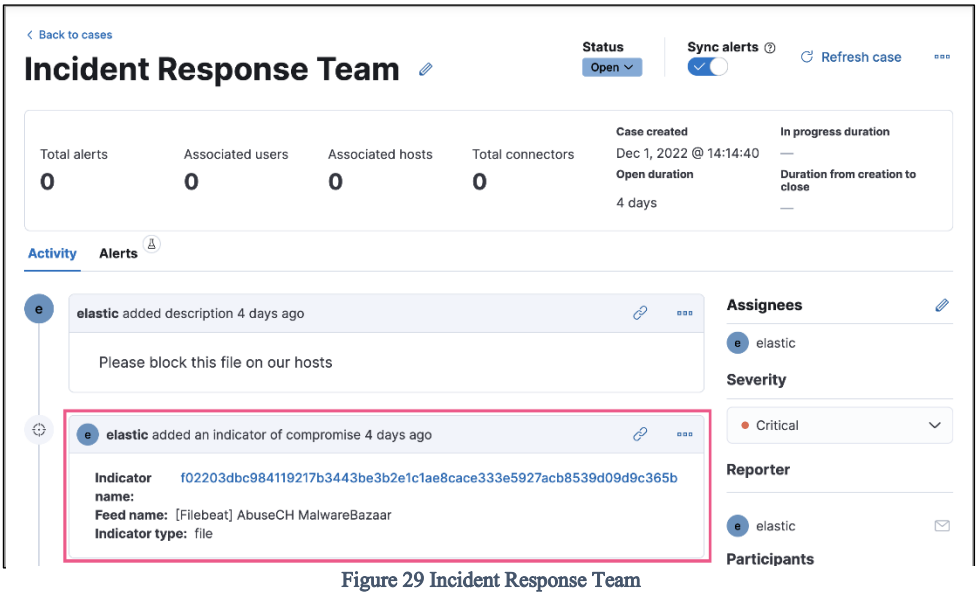


Figure 29 Incident Response Team

When you attach an indicator to a case, the indicator is added as a new comment with the following details:

- **Indicator name**: Click the linked name to open the Indicator details flyout, which contains additional information about the indicator. Indicator details are in JSON format.
- **Feed name**: The threat feed from which the indicator was ingested.
- **Indicator type**: The indicator type, for example, file or .exe.

Remove indicators from cases

To remove an indicator attached to a case, click **More actions** (…) → **Delete attachment** in the case comment.

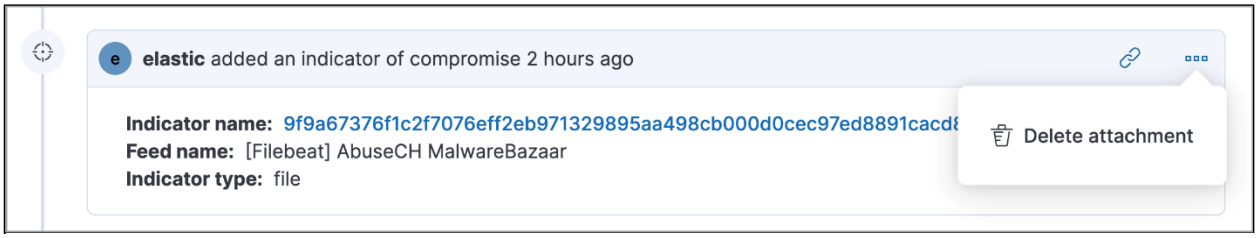


Figure 30 Remove indicators from cases

Manage

Expand this section to access and manage additional security features:

- **Rules**: Create and manage rules to monitor suspicious events.
- **Shared Exception Lists**: View and manage rule exceptions and shared exception lists.
- **Endpoints**: View and manage hosts running Elastic Defend.
- **Policies**: View and manage Elastic Defend integration policies.
- **Trusted applications**: View and manage trusted Windows, macOS, and Linux applications.

- Event filters: View and manage event filters, which allow you to filter endpoint events you don't need to want stored in Elasticsearch.
- Host isolation exceptions: View and manage host isolation exceptions, which specify IP addresses that can communicate with your hosts even when those hosts are blocked from your network.
- Blocklist: View and manage the blocklist, which allows you to prevent specified applications from running on hosts, extending the list of processes that Elastic Defend considers malicious.
- CSP Benchmarks: View, enable, or disable benchmark rules.

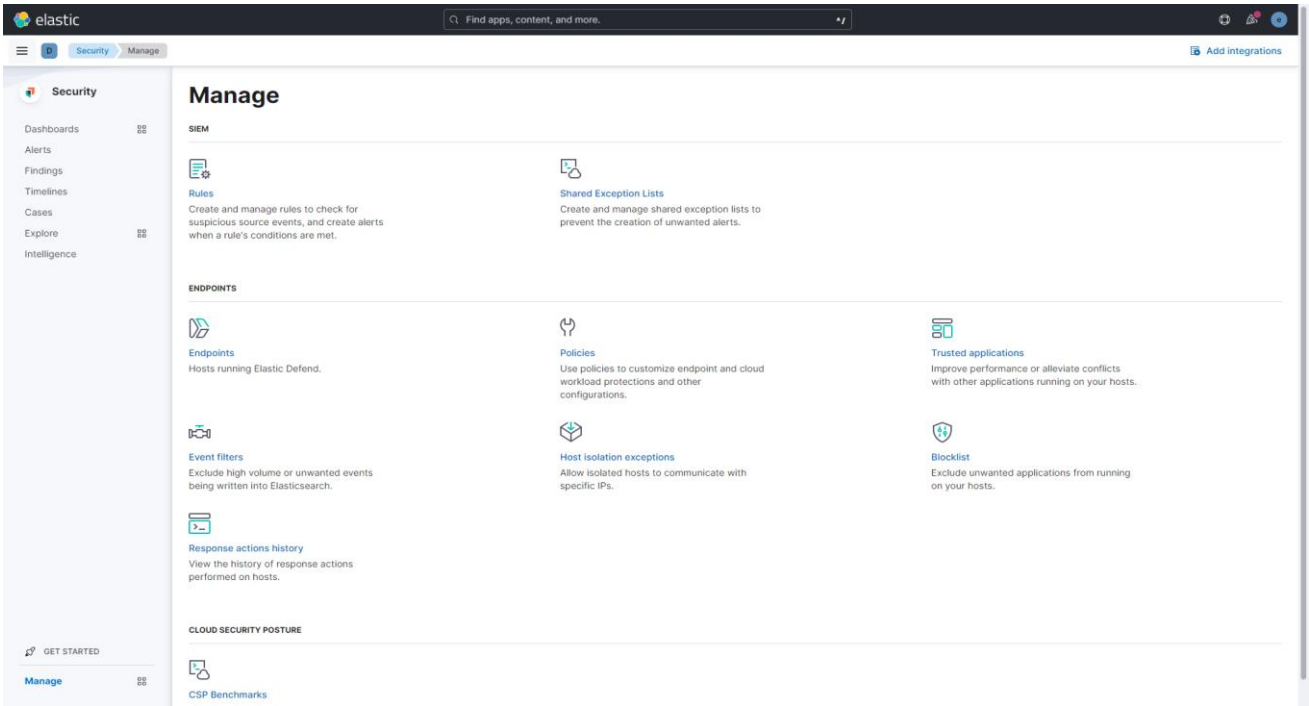


Figure 31 Manage page

6 ADDENDUM

Kibana Configuration

Below is the list of command and the configuration file used for Kibana server. This file specifies everything that are used in Kibana server during configuration.

For more configuration options see the configuration guide for Kibana in

https://www.elastic.co/guide/index.html

========== System: Kibana Server ==========

Kibana is served by a back end server. This setting specifies the port to use.

#server.port: 5601

Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.

```
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
server.host: "192.168.18.171"

# Enables you to specify a path to mount Kibana at if you are running behind a proxy.
# Use the `server.rewriteBasePath` setting to tell Kibana if it should remove the basePath
# from requests it receives, and to prevent a deprecation warning at startup.
# This setting cannot end in a slash.
#server.basePath: ""

# Specifies whether Kibana should rewrite requests that are prefixed with
# `server.basePath` or require that they are rewritten by your reverse proxy.
# Defaults to `false`.
#server.rewriteBasePath: false

# Specifies the public URL at which Kibana is available for end users. If
# `server.basePath` is configured this URL should end with the same basePath.
#server.publicBaseUrl: ""

# The maximum payload size in bytes for incoming server requests.
#server.maxPayload: 1048576

# The Kibana server's name. This is used for display purposes.
#server.name: "your-hostname"

# ==================== System: Kibana Server (Optional) ====================
# Enables SSL and paths to the PEM-format SSL certificate and SSL key files, respectively.
# These settings enable SSL for outgoing requests from the Kibana server to the browser.
#server.ssl.enabled: false
#server.ssl.certificate: /path/to/your/server.crt
#server.ssl.key: /path/to/your/server.key

# ==================== System: Elasticsearch ====================
# The URLs of the Elasticsearch instances to use for all your queries.
#elasticsearch.hosts: ["http://localhost:9200"]
```

```
# If your Elasticsearch is protected with basic authentication, these settings provide
# the username and password that the Kibana server uses to perform maintenance on the Kibana
# index at startup. Your Kibana users still need to authenticate with Elasticsearch, which
# is proxied through the Kibana server.
#elasticsearch.username: "kibana_system"
#elasticsearch.password: "pass"

# Kibana can also authenticate to Elasticsearch via "service account tokens".
# Service account tokens are Bearer style tokens that replace the traditional username/password based configuration.
# Use this token instead of a username/password.
# elasticsearch.serviceAccountToken: "my_token"

# Time in milliseconds to wait for Elasticsearch to respond to pings. Defaults to the value of
# the elasticsearch.requestTimeout setting.
# elasticsearch.pingTimeout: 1500

# Time in milliseconds to wait for responses from the back end or Elasticsearch. This value
# must be a positive integer.
#elasticsearch.requestTimeout: 30000

# The maximum number of sockets that can be used for communications with elasticsearch.
# Defaults to `Infinity`.
#elasticsearch.maxSockets: 1024

# Specifies whether Kibana should use compression for communications with elasticsearch
# Defaults to `false`.
#elasticsearch.compression: false

# List of Kibana client-side headers to send to Elasticsearch. To send *no* client-side
# headers, set this value to [] (an empty list).
#elasticsearch.requestHeadersWhitelist: [ authorization ]

# Header names and values that are sent to Elasticsearch. Any custom headers cannot be overwritten
# by client-side headers, regardless of the elasticsearch.requestHeadersWhitelist configuration.
```

```
#elasticsearch.customHeaders: {}

# Time in milliseconds for Elasticsearch to wait for responses from shards. Set to 0 to disable.
#elasticsearch.shardTimeout: 30000

# ==================== System: Elasticsearch (Optional) ====================
# These files are used to verify the identity of Kibana to Elasticsearch and are required when
# xpack.security.http.ssl.client_authentication in Elasticsearch is set to required.
#elasticsearch.ssl.certificate: /path/to/your/client.crt
#elasticsearch.ssl.key: /path/to/your/client.key

# Enables you to specify a path to the PEM file for the certificate

# authority for your Elasticsearch instance.
#elasticsearch.ssl.certificateAuthorities: [ "/path/to/your/CA.pem" ]

# To disregard the validity of SSL certificates, change this setting's value to 'none'.
#elasticsearch.ssl.verificationMode: full

# ==================== System: Logging ====================
# Set the value of this setting to off to suppress all logging output, or to debug to log everything. Defaults to 'info'
#logging.root.level: debug

# Enables you to specify a file where Kibana stores log output.
logging:
  appenders:
    file:
      type: file
      fileName: /var/log/kibana/kibana.log
      layout:
        type: json
  root:
    appenders:
      - default
      - file
```

```
#  layout:
#    type: json

# Logs queries sent to Elasticsearch.
#logging.loggers:
#  - name: elasticsearch.query
#    level: debug
# Logs http responses.
#logging.loggers:
#  - name: http.server.response
#    level: debug

# Logs system usage information.
#logging.loggers:
#  - name: metrics.ops
#    level: debug

# ==================== System: Other ====================
# The path where Kibana stores persistent data not saved in Elasticsearch. Defaults to data
#path.data: data

# Specifies the path where Kibana creates the process ID file.
pid.file: /run/kibana/kibana.pid

# Set the interval in milliseconds to sample system and process performance
# metrics. Minimum is 100ms. Defaults to 5000ms.
#ops.interval: 5000

# Specifies locale to be used for all localizable strings, dates and number formats.
# Supported languages are the following: English (default) "en", Chinese "zh-CN", Japanese "ja-JP", French "fr-FR".
#i18n.locale: "en"

# ==================== Frequently used (Optional)====================
```

========== Saved Objects: Migrations ==========

Saved object migrations run at startup. If you run into migration-related issues, you might need to adjust these settings.

The number of documents migrated at a time.

If Kibana can't start up or upgrade due to an Elasticsearch `circuit_breaking_exception`,

use a smaller batchSize value to reduce the memory pressure. Defaults to 1000 objects per batch.

#migrations.batchSize: 1000

The maximum payload size for indexing batches of upgraded saved objects.

To avoid migrations failing due to a 413 Request Entity Too Large response from Elasticsearch.

This value should be lower than or equal to your Elasticsearch cluster's `http.max_content_length`

configuration option. Default: 100mb

#migrations.maxBatchSizeBytes: 100mb

The number of times to retry temporary migration failures. Increase the setting

if migrations fail frequently with a message such as `Unable to complete the [...] step after

15 attempts, terminating`. Defaults to 15

#migrations.retryAttempts: 15

========== Search Autocomplete ==========

Time in milliseconds to wait for autocomplete suggestions from Elasticsearch.

This value must be a whole number greater than zero. Defaults to 1000ms

#unifiedSearch.autocomplete.valueSuggestions.timeout: 1000

Maximum number of documents loaded by each shard to generate autocomplete suggestions.

This value must be a whole number greater than zero. Defaults to 100_000

#unifiedSearch.autocomplete.valueSuggestions.terminateAfter: 100000

This section was automatically generated during setup.

elasticsearch.hosts: ['https://192.168.18.171:9200']

elasticsearch.username: kibana_system

elasticsearch.password: 9I*Zrw-Uvltc=CL7o4k_

elasticsearch.ssl.certificateAuthorities: [/var/lib/kibana/ca_1674107558849.crt]

xpack.fleet.outputs: [{id: fleet-default-output, name: default, is_default: true, is_default_monitoring: true, type: elasticsearch, hosts: ['https://192.168.18.171:9200'], ca_trusted_fingerprint: 379deb27bff02256f432f93c8ee48cc7f08252eb04c26bd44df76205ff4f7857}]

xpack.encryptedSavedObjects.encryptionKey: 2P-$(ASD>3My[~6)wJyTPz(ScH8k`~_M

Elasticsearch-ELK 1Configuration

Below is the list of command and the configuration file used for ELK1 server. This file specifies everything that are used in ELK1 server during configuration.

```
# ================ Elasticsearch Configuration =========================
#
# NOTE: Elasticsearch comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure you
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluster.
#
# Please consult the documentation for further information on configuration options:
# https://www.elastic.co/guide/en/elasticsearch/reference/index.html
#
# ---------------------------------- Cluster -----------------------------------
#
# Use a descriptive name for your cluster:
#
cluster.name: hs-elastic
#
# ------------------------------------ Node ------------------------------------
#
# Use a descriptive name for the node:
#
node.name: elk1
node.roles: [ master, data, ingest, ml ]
#
# Add custom attributes to the node:
#
#node.attr.rack: r1
#
```

```
# ---------------------------------- Paths ----------------------------------
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: /var/lib/elasticsearch
#
# Path to log files:
#
path.logs: /var/log/elasticsearch
#
# ---------------------------------- Memory -----------------------------------
#
# Lock the memory on startup:
#
#bootstrap.memory_lock: true
#
# Make sure that the heap size is set to about half the memory available
# on the system and that the owner of the process is allowed to use this
# limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ---------------------------------- Network ----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 192.168.18.171
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
# For more information, consult the network module documentation.
```

```
#
# ---------------------------------- Discovery ----------------------------------
#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
discovery.seed_hosts: ["elk1", "elk2", "elk3"]
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
cluster.initial_master_nodes: ["elk1", "elk2", "elk3"]
#
# For more information, consult the discovery and cluster formation module documentation.
#
# --------------------------------- Readiness ----------------------------------
#
# Enable an unauthenticated TCP readiness endpoint on localhost
#
#readiness.port: 9399
#
# ---------------------------------- Various -----------------------------------
#
# Allow wildcard deletion of indices:
#
#action.destructive_requires_name: false

#----------------------- BEGIN SECURITY AUTO CONFIGURATION -----------------------
#
# The following settings, TLS certificates, and keys have been automatically
# generated to configure Elasticsearch security features on 14-12-2022 23:44:37
#
# --------------------------------------------------------------------------------

# Enable security features
xpack.security.enabled: true
```

```
xpack.security.enrollment.enabled: true

# Enable encryption for HTTP API client connections, such as Kibana, Logstash, and Agents
xpack.security.http.ssl:
  enabled: true
  keystore.path: certs/http.p12

# Enable encryption and mutual authentication between cluster nodes
xpack.security.transport.ssl:
  enabled: true
  verification_mode: certificate
  keystore.path: certs/transport.p12
  truststore.path: certs/transport.p12
# Create a new cluster with the current node only
# Additional nodes can still join the cluster later
#cluster.initial_master_nodes: ["elk1"]

# Allow HTTP API connections from anywhere
# Connections are encrypted and require user authentication
#http.host: 0.0.0.0

# Allow other nodes to join the cluster from anywhere
# Connections are encrypted and mutually authenticated
#transport.host: 0.0.0.0
#----------------------- END SECURITY AUTO CONFIGURATION -------------------------
```

Elasticsearch-ELK 2 Configuration

Below is the list of command and the configuration file used for ELK2 server. This file specifies everything that are used in ELK2 server during configuration.

```
# ==================== ElasticsearchConfiguration ====================
#
# NOTE: Elasticsearch comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure you
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
```

```
# the most important settings you may want to configure for a production cluster.
#
# Please consult the documentation for further information on configuration options:
# https://www.elastic.co/guide/en/elasticsearch/reference/index.html
#
# ---------------------------------- Cluster -----------------------------------
#
# Use a descriptive name for your cluster:
#
cluster.name: hs-elastic
#
# ------------------------------------ Node ------------------------------------
#
# Use a descriptive name for the node:
#
node.name: elk2
node.roles: [ master,data ]
#
# Add custom attributes to the node:
#
#node.attr.rack: r1
#
# ---------------------------------- Paths -----------------------------------
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: /var/lib/elasticsearch
#
# Path to log files:
#
path.logs: /var/log/elasticsearch
#
# ---------------------------------- Memory ----------------------------------
#
# Lock the memory on startup:
```

```
#
#bootstrap.memory_lock: true
#
# Make sure that the heap size is set to about half the memory available
# on the system and that the owner of the process is allowed to use this
# limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ---------------------------------- Network -----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 192.168.18.229
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
# --------------------------------- Discovery ----------------------------------
#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
discovery.seed_hosts: ["elk1", "elk2", "elk3"]
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
cluster.initial_master_nodes: ["elk1", "elk2", "elk3"]
#
# For more information, consult the discovery and cluster formation module documentation.
```

```
#
# ---------------------------------- Readiness ----------------------------------
#
# Enable an unauthenticated TCP readiness endpoint on localhost
#
#readiness.port: 9399
#
# ---------------------------------- Various -----------------------------------
#
# Allow wildcard deletion of indices:
#
#action.destructive_requires_name: false

#----------------------- BEGIN SECURITY AUTO CONFIGURATION -----------------------
#
# The following settings, TLS certificates, and keys have been automatically
# generated to configure Elasticsearch security features on 14-12-2022 23:48:22
#
# ----------------------------------------------------------------------------

# Enable security features
xpack.security.enabled: true

xpack.security.enrollment.enabled: true

# Enable encryption for HTTP API client connections, such as Kibana, Logstash, and Agents
xpack.security.http.ssl:
  enabled: true
  keystore.path: certs/http.p12

# Enable encryption and mutual authentication between cluster nodes
xpack.security.transport.ssl:
  enabled: true
  verification_mode: certificate
  keystore.path: certs/transport.p12
```

```
  truststore.path: certs/transport.p12
# Discover existing nodes in the cluster
#discovery.seed_hosts: ["192.168.18.171:9300"]

# Allow HTTP API connections from anywhere
# Connections are encrypted and require user authentication
#http.host: 0.0.0.0

# Allow other nodes to join the cluster from anywhere
# Connections are encrypted and mutually authenticated
#transport.host: 0.0.0.0

#----------------------- END SECURITY AUTO CONFIGURATION -------------------------
```

Elasticsearch-ELK 3 Configuration

Below is the list of command and the configuration file used for ELK 3 server. This file specifies everything that are used in ELK3 server during configuration.

```
# ================ Elasticsearch Configuration =====================
#
# NOTE: Elasticsearch comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure you
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluster.
#
# Please consult the documentation for further information on configuration options:
# https://www.elastic.co/guide/en/elasticsearch/reference/index.html
#
# ---------------------------------- Cluster -----------------------------------
#
# Use a descriptive name for your cluster:
#
cluster.name: hs-elastic
#
# ------------------------------------ Node ------------------------------------
#
```

```
# Use a descriptive name for the node:
#
node.name: elk3
node.roles: [ master, data ]
#
# Add custom attributes to the node:
#
#node.attr.rack: r1
#
# ------------------------------------ Paths ------------------------------------
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: /var/lib/elasticsearch
#
# Path to log files:
#
path.logs: /var/log/elasticsearch
#
# ---------------------------------- Memory -----------------------------------
#
# Lock the memory on startup:
#
#bootstrap.memory_lock: true
#
# Make sure that the heap size is set to about half the memory available
# on the system and that the owner of the process is allowed to use this
# limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ---------------------------------- Network ----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
```

```
#
network.host: 192.168.18.113
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
#
# For more information, consult the network module documentation.
#
# --------------------------------- Discovery ----------------------------------
#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
discovery.seed_hosts: ["elk1", "elk2", "elk3"]
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
cluster.initial_master_nodes: ["elk1", "elk2", "elk3"]
#
# For more information, consult the discovery and cluster formation module documentation.
#
# --------------------------------- Readiness ----------------------------------
#
# Enable an unauthenticated TCP readiness endpoint on localhost
#
#readiness.port: 9399
#
# ---------------------------------- Various -----------------------------------
#
# Allow wildcard deletion of indices:
#
#action.destructive_requires_name: false
```

```
#----------------------- BEGIN SECURITY AUTO CONFIGURATION -----------------------
#
# The following settings, TLS certificates, and keys have been automatically
# generated to configure Elasticsearch security features on 15-12-2022 00:21:03
#
# ----------------------------------------------------------------------------

# Enable security features
xpack.security.enabled: true

xpack.security.enrollment.enabled: true

# Enable encryption for HTTP API client connections, such as Kibana, Logstash, and Agents
xpack.security.http.ssl:
  enabled: true
  keystore.path: certs/http.p12

# Enable encryption and mutual authentication between cluster nodes
xpack.security.transport.ssl:
  enabled: true
  verification_mode: certificate
  keystore.path: certs/transport.p12
  truststore.path: certs/transport.p12
# Discover existing nodes in the cluster
#discovery.seed_hosts: ["192.168.18.171:9300", "192.168.18.229:9300"]

# Allow HTTP API connections from anywhere
# Connections are encrypted and require user authentication
#http.host: 0.0.0.0

# Allow other nodes to join the cluster from anywhere
# Connections are encrypted and mutually authenticated
#transport.host: 0.0.0.0

#----------------------- END SECURITY AUTO CONFIGURATION -------------------------
```

7 Copyright and Licensing

8 References

[1] "Elastic Stack features," [Online]. Available: https://www.elastic.co/elastic-stack/features.

[2] "Setting up the environment in Java," GeeksforGeeks, 10 10 2022. [Online]. Available: https://www.geeksforgeeks.org/setting-environment-java/.

[3] "How to Install and Configure Elasticsearch on Ubuntu ?," GeeksforGeeks, 30 Sep 2022. [Online]. Available: https://www.geeksforgeeks.org/how-to-install-and-configure-elasticsearch-on-ubuntu/.

[4] "Install Kibana with Debian package," Elasticsearch B.V, 2023. [Online]. Available: https://www.elastic.co/guide/en/kibana/current/deb.html.

[5] "Installing Logstash," Elasticsearch B.C., 2023. [Online]. Available: https://www.elastic.co/guide/en/logstash/8.6/installing-logstash.html.

[6] "Get Started with Beats," Elasticsearch B.V., 2023. [Online]. Available: https://www.elastic.co/guide/en/beats/libbeat/8.6/getting-started.html.

[7] "Auditbeat quick start: Installation and Configuration," Elasticsearch B.V., 2023. [Online]. Available: https://www.elastic.co/guide/en/beats/auditbeat/8.6/auditbeat-installation-configuration.html.

[8] "Filebeat quick start: Installation and Configuration," Elasticsearch B.V, 2023. [Online]. Available: https://www.elastic.co/guide/en/beats/filebeat/8.6/filebeat-installation-configuration.html.

[9] "Functionbeat quick start: Installation and Configuration," Elasticsearch B.V., 2023. [Online]. Available: https://www.elastic.co/guide/en/beats/functionbeat/8.6/functionbeat-installation-configuration.html.

[10] "Heartbeat quick start: Installation and Configuration," Elasticsearch B.V., 2023. [Online]. Available: https://www.elastic.co/guide/en/beats/heartbeat/8.6/heartbeat-installation-configuration.html.

[11] "Metricbeat quick start: installation and configuration," Elasticsearch B.V., 2023. [Online]. Available: https://www.elastic.co/guide/en/beats/metricbeat/8.6/metricbeat-installation-configuration.html#metricbeat-installation-configuration.

[12] "Packetbeat quick start: installation and configuration," Elasticsearch B.V., 2023. [Online]. Available: https://www.elastic.co/guide/en/beats/packetbeat/8.6/packetbeat-installation-configuration.html.

[13] "Winlogbeat quick start: installation and configuration," Elasticsearch B.V., 2023. [Online]. Available: https://www.elastic.co/guide/en/beats/winlogbeat/8.6/winlogbeat-installation-configuration.html.

[14] "Application Performance Monitoring (APM)," Elasticsearch B.V., 2023. [Online]. Available: https://www.elastic.co/guide/en/apm/guide/8.6/apm-quick-start.html.

[15] "Elasticsearch for Apache Hadoop and Spark:," Elasticsearch B.V., 2023. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/hadoop/8.6/install.html.

[16] "Configuring Elasticsearch," Elasticsearch B.V., 2023. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/settings.html.

[17] "Important Elasticsearch Configuration," Elasticsearch B.V., 2023. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/important-settings.html.

18] "Elastic Security UI," Elasticsearch B.V, 2023. [Online]. Available: https://www.elastic.co/guide/en/security/current/es-ui-overview.html#_dashboards.

Index