

Elastic APM Workshop Handbook

Requirements:

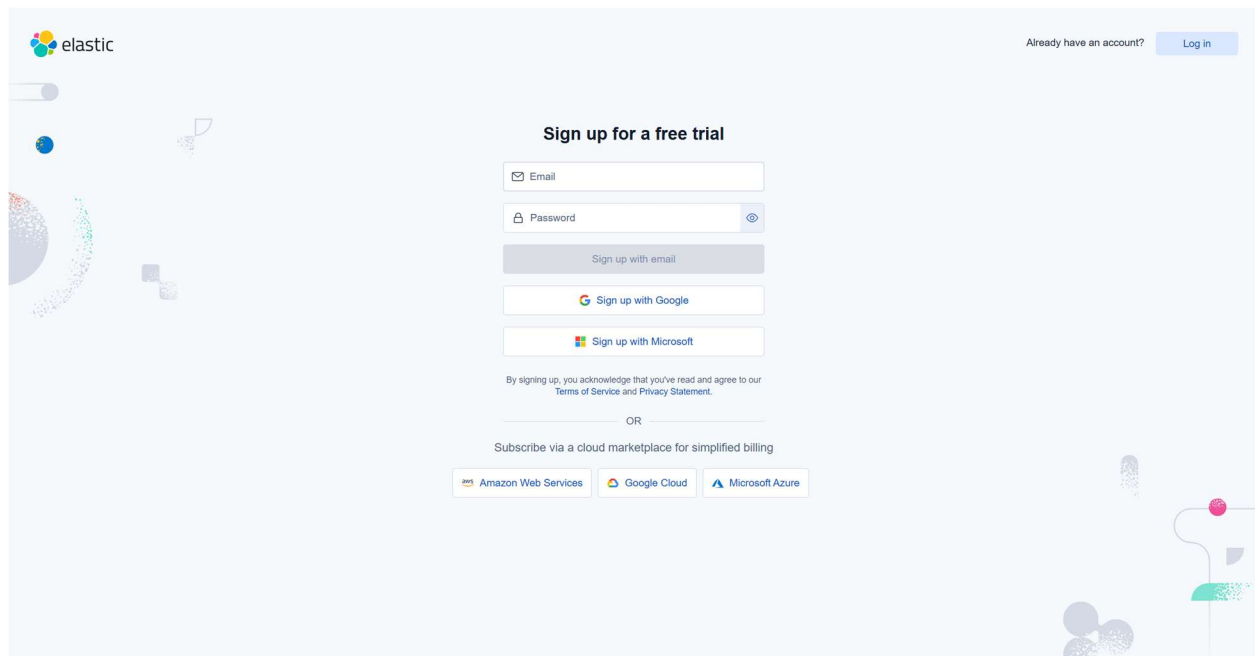
- Internet connection is required
- JDK 17 or later and configured JAVA_HOME
<https://www.oracle.com/id/java/technologies/downloads/#java24>
- Download material dan demo app pada link berikut
<https://github.com/xyperia/elastic-apm-workshop>

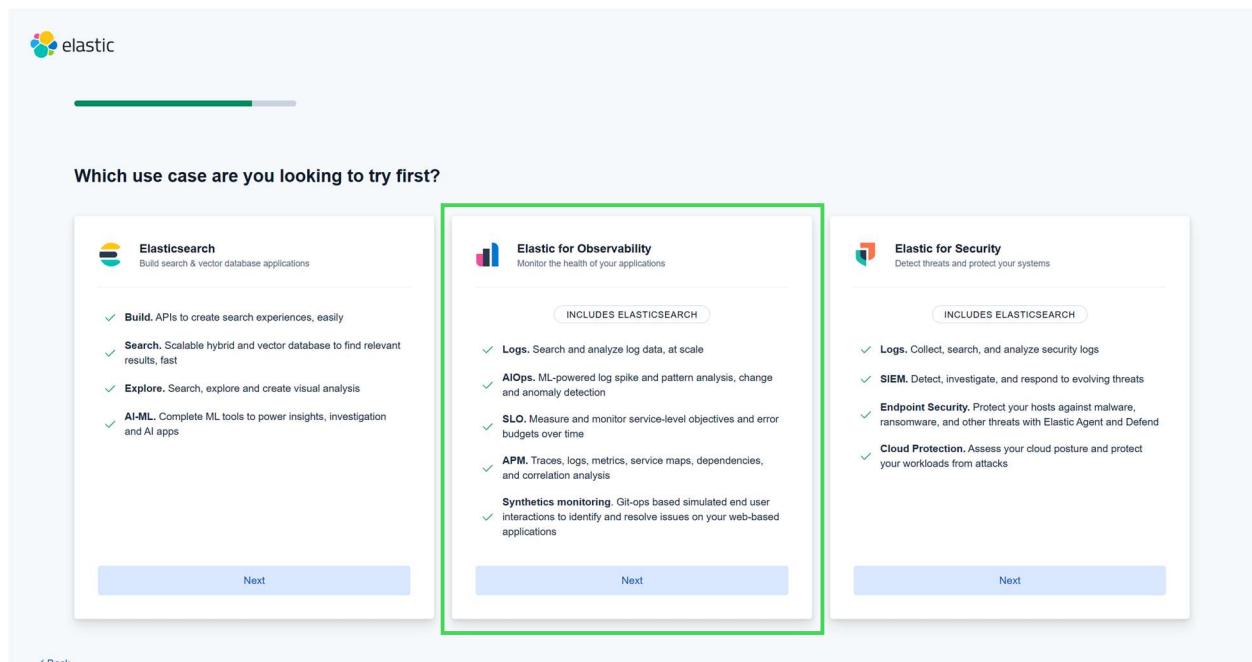
OpenAI Token

```
sk-proj-  
47BgwsGfqw7p6TsCzTCH2XrYuVy_bAczWN_hTuGiaCAGvL_S4v64G68i6LQIKUBGsJEC5GSUWkT3BL  
bkFJZP4_gSTLARJC6_rXzj6UZq5KBm_hVSMh3Bx0NI5XIRSU5ejA96HqzvafFZbC1T651judbctLoA
```

Create Deployment

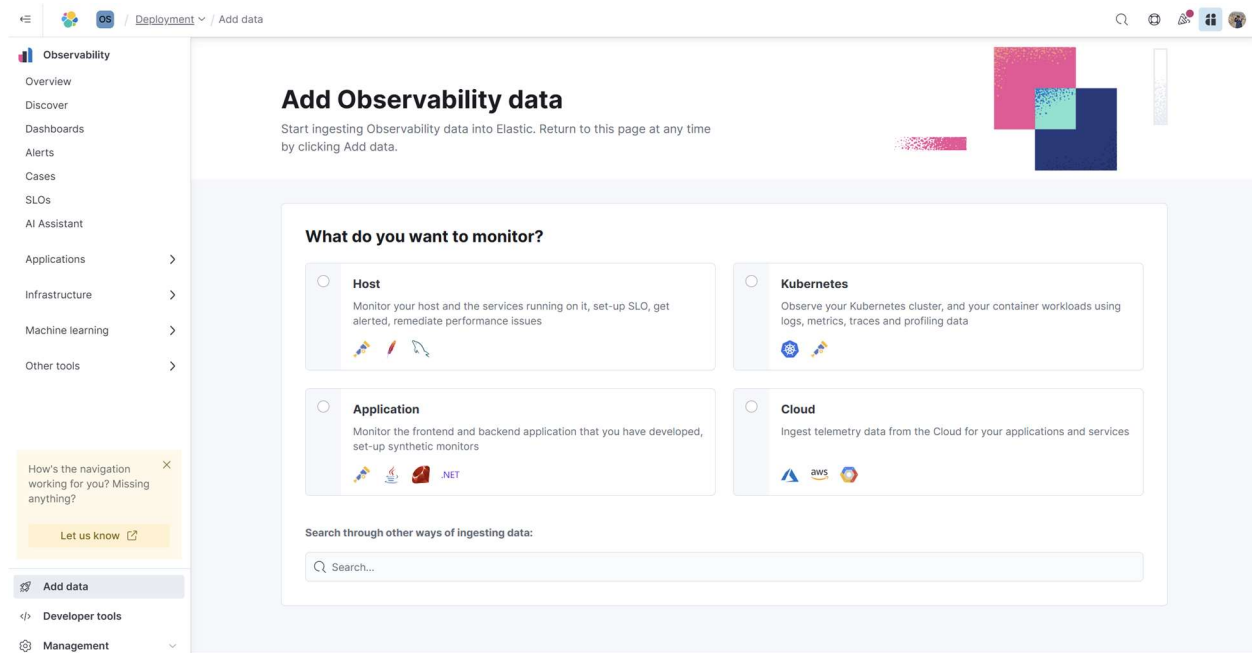
Akses Elastic Cloud pada cloud.elastic.co menggunakan email untuk membuat akun baru. Sesuaikan dengan kebutuhan anda. *Screenshot terlampir*





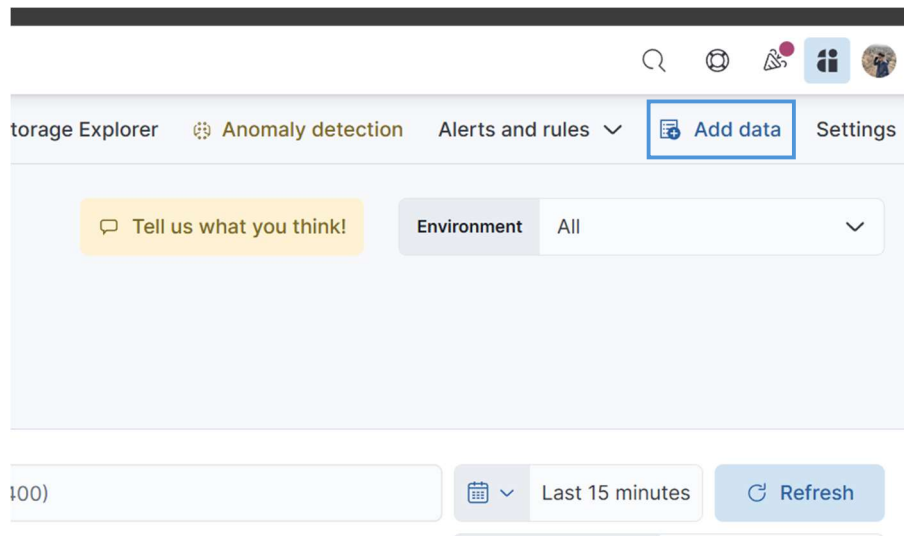
APM Agent Onboarding

Berikut adalah tampilan awal jika cluster Elastic sudah berhasil di-setup

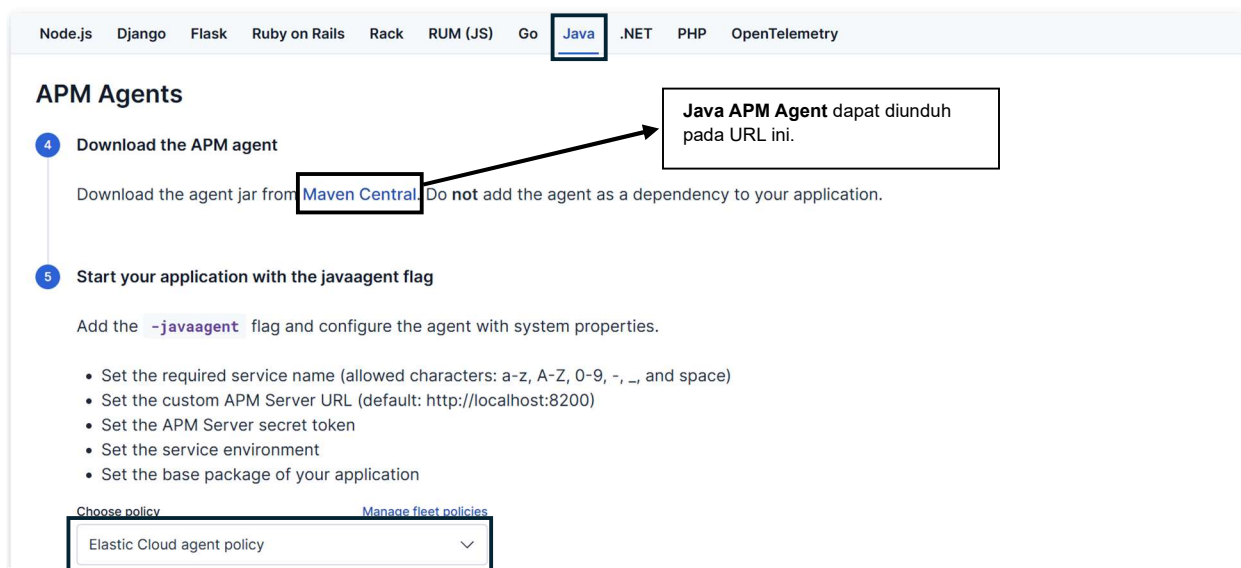


Navigasi ke **Applications > Service Inventory**

Klik **Add data** untuk menambahkan aplikasi yang akan dimonitor



Demo kali ini akan menggunakan Java App.



Berikut merupakan sample command yang bisa digunakan untuk implementasi APM agent

Choose policy [Manage fleet policies](#)

Elastic Cloud agent policy

Adds the selected policy configuration to the snippet below.

Configuration setting	Configuration value
Delastic.apm.service_name	my-service-name
Delastic.apm.secret_token	3rjm7a3ybIK5t4U5oL
Delastic.apm.server_url	https://c130387521b94b5dbbfb856aeb11ca90.apm.asia-southeast1.gcp.elastic-cloud.com:443
Delastic.apm.environment	my-environment

```
1 java -javaagent:/path/to/elastic-apm-agent-<version>.jar \  
2 -Delastic.apm.service_name=my-service-name \  
3 -Delastic.apm.secret_token=3rjm7a3ybIK5t4U5oL \  
4 -Delastic.apm.server_url=https://c130387521b94b5dbbfb856aeb11ca90.apm.asia-southeast1.gcp.elastic-cloud.com:443 \  
5 -Delastic.apm.environment=my-environment \  
6 -Delastic.apm.application_packages=org.example \  
7 -jar my-service-name.jar
```

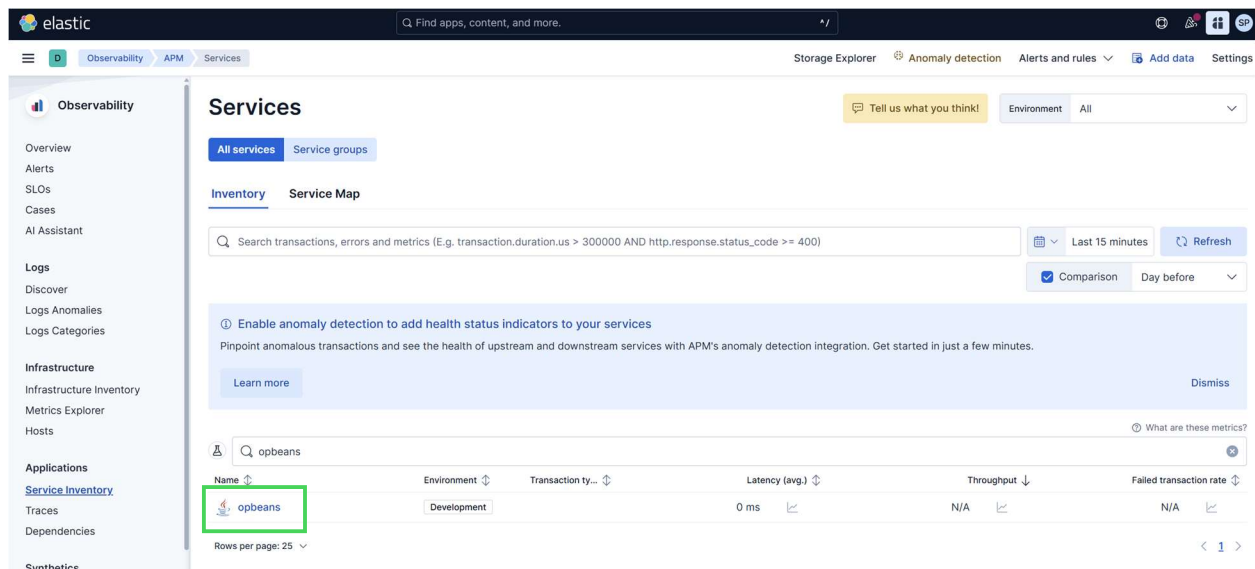
See the [documentation](#) for configuration options and advanced usage.

Path, environment, dan service name bisa disesuaikan dengan aplikasi yang ingin dimonitor.

Contoh command:

```
java -javaagent:"C:\Users\xyperia\OneDrive\Elastic\elastic-apm-agent-1.54.0.jar" -Delastic.apm.service_name=opbeans -Delastic.apm.secret_token=3rjm7a3ybIK5t4U5oL -Delastic.apm.server_url=https://c130387521b94b5dbbfb856aeb11ca90.apm.asia-southeast1.gcp.elastic-cloud.com:443 -Delastic.apm.environment=Development -jar "C:\Users\xyperia\OneDrive\Elastic\opbeans-0.0.1.jar"
```

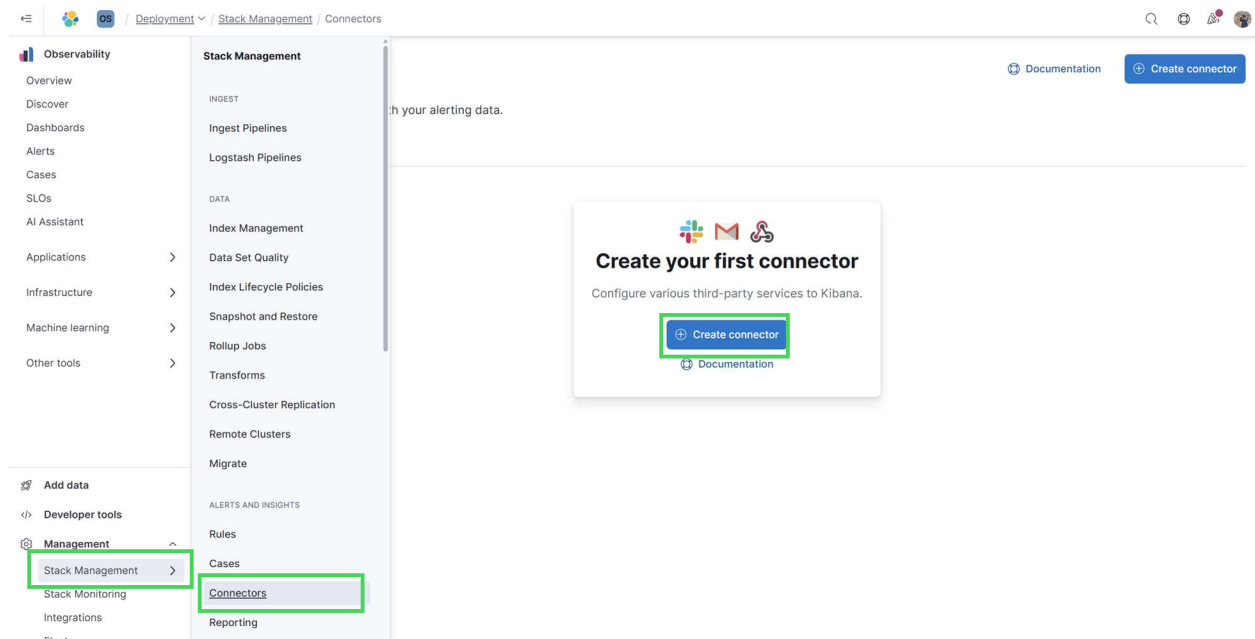
Entry Opbeans akan muncul apabila integrasi sukses dilakukan



Aplikasi Opbeans dapat diakses melalui <http://localhost:8080>. Silahkan akses app tersebut dan monitor aplikasi melalui Kibana


AI Assistant Integration

Navigasi ke menu **Stack Management > Connectors > Create connector** untuk menambahkan integrasi baru



Tambahkan integrasi OpenAI dengan memilih OpenAI

Selanjutnya masukan Connector name dan API Key. API Key dapat dilihat pada halaman 1.

**OpenAI connector**×

Send a request to an OpenAI or Azure OpenAI service.

Compatibility: Generative AI for Security Generative AI for Observability Generative AI for Search

Connector name


Connector settings
Select an OpenAI provider

OpenAI ▼

URL

The OpenAI API endpoint URL. For more information on the URL, refer to the [OpenAI documentation](#).
Default model

If a request does not include a model, it uses the default.
OpenAI Organization
Optional
For users who belong to multiple organizations. Organization IDs can be found on your Organization settings page.
OpenAI Project
Optional
For users who are accessing their projects through their legacy user API key. Project IDs can be found on your General settings page by selecting the specific project.
Authentication
API key



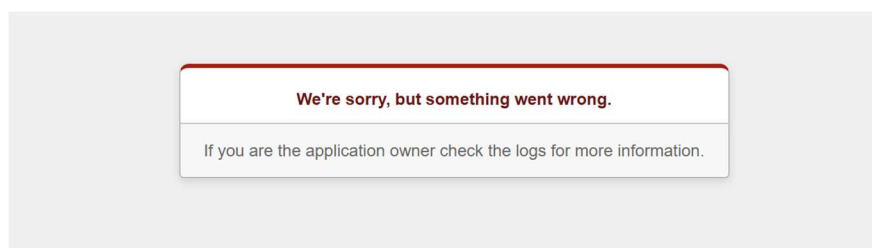
The OpenAI API authentication key for HTTP Basic authentication. For more details about generating OpenAI API keys, refer to the [OpenAI documentation](#).

Setelah itu klik Save.

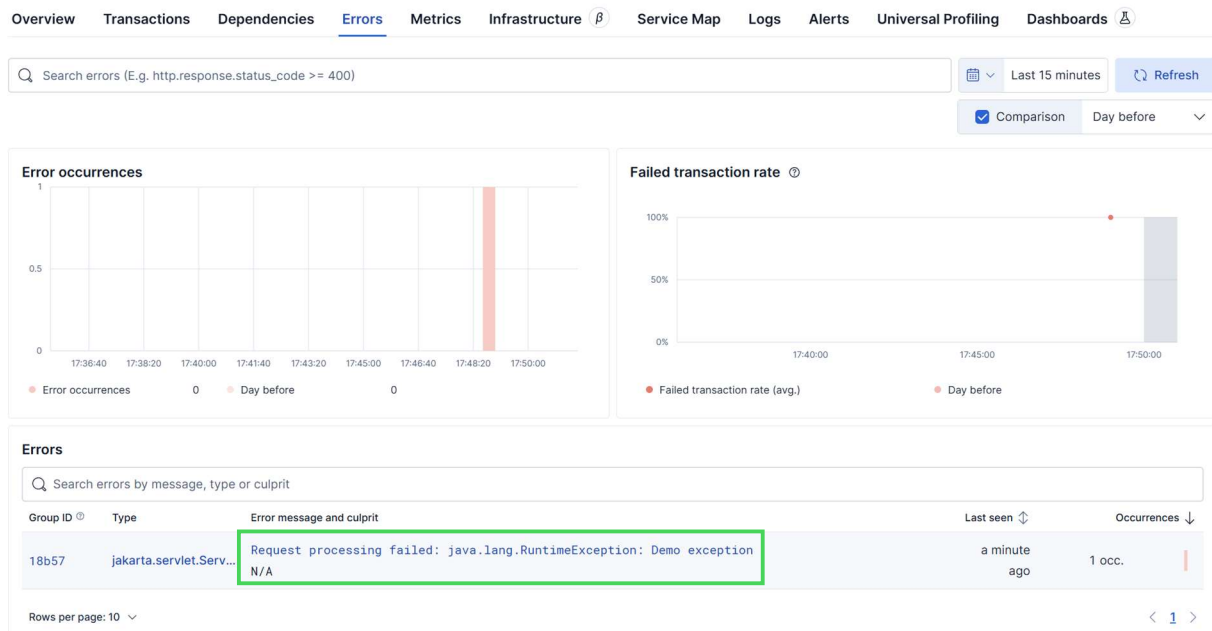
Testing AI Assistant

Navigasi ke app Opbeans, lalu akses url berikut <http://localhost:8080/is-it-coffee-time>

Endpoint tersebut akan men-trigger error pada aplikasi yang akan dibaca oleh Elastic APM.



Kembali ke Opbeans pada menu Elastic APM, lalu navigasi ke tab **Errors**



Elastic APM mendeteksi error yang terjadi pada aplikasi. Silahkan klik pada pesan error
Kita dapat memanfaatkan OpenAI untuk melakukan analisa terhadap error log yang muncul pada menu APM.

4 minutes ago | GET http://localhost:8080/is-it-coffee-time | 500 Internal Server Error | Chrome (137.0.0.0) | IndexController#error

Exception message

Request processing failed: java.lang.RuntimeException: Demo exception

Culprit

N/A

> **What's this error?** Get helpful insights from our Elastic AI Assistant.

[Exception stack trace](#) [Metadata](#)

jakarta.servlet.ServletException: Request processing failed: java.lang.RuntimeException: Demo exception

```
at org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:1022)
at org.springframework.web.servlet.FrameworkServlet.doGet(FrameworkServlet.java:903)
at jakarta.servlet.http.HttpServlet.service(HttpServlet.java:564)
at org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:885)
at jakarta.servlet.http.HttpServlet.service(HttpServlet.java:658)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:205)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:149)
```

4 minutes ago | GET http://localhost:8080/is-it-coffee-time | 500 Internal Server Error | Chrome (137.0.0.0) | [IndexController#error](#)

Exception message

Request processing failed: java.lang.RuntimeException: Demo exception

Culprit

N/A

What's this error? Get helpful insights from our Elastic AI Assistant.

The exception you're seeing, `java.lang.RuntimeException: Demo exception`, appears to be a deliberately thrown exception likely used for testing or demonstration purposes within the opbeans Java service.

The stack trace shows the exception originating from the `IndexController#error` endpoint, suggesting this endpoint is specifically designed to trigger this error. The exception is caught and handled by the Spring framework, ultimately resulting in a `jakarta.servlet.ServletException`.

Since this is a demo exception, it might not indicate a real problem in your application. However, if you're encountering this unexpectedly outside of a testing context, it could suggest an issue in your code where a demo exception is being thrown unintentionally. You should review the code within the `IndexController#error` endpoint and any related components to determine why this exception is being thrown.

Was this helpful? Yes No [Regenerate](#) [Start conversation](#)

[Exception stack trace](#) [Metadata](#)

jakarta.servlet.ServletException: Request processing failed: java.lang.RuntimeException: Demo exception

Machine Learning Integration

Elastic dilengkapi dengan kemampuan machine learning. Pada workshop ini, kita akan memanfaatkan kemampuan machine learning Elastic untuk mendeteksi jika ada anomaly dari sisi metrics yang dibaca oleh Elastic APM

Silahkan menambahkan sample data pada menu berikut

Deployment | Add data

Find apps, content, and more.

Observability

- Overview
- Discover
- Dashboards
- Alerts
- Cases
- SLOs
- AI Assistant
- Applications >
- Infrastructure >
- Machine learning >
- Other tools >

Add data

- Developer tools
- Management
 - Stack Management >
 - Stack Monitoring
 - Integrations
 - Fleet

Add Observability data

Start ingesting Observability data into Elastic. Return to this page at any time by clicking Add data.

What do you want to monitor?

- ☐ Host
Monitor your host and the services running on it, set-up SLO, get alerted, remediate performance issues
- ☐ Kubernetes
Observe your Kubernetes cluster, and your container workloads using logs, metrics, traces and profiling data
- ☐ Application
Monitor the frontend and backend application that you have developed, set-up synthetic monitors
- ☐ Cloud
Ingest telemetry data from the Cloud for your applications and services

Search through other ways of ingesting data:

Upload a file

Upload a file
Upload data from a CSV, TSV, JSON or other log file to Elasticsearch for analysis.

Klik add data pada sample data yang tersedia

More ways to add data

In addition to adding [integrations](#), you can try our sample data or upload your own data.

[Sample data](#) [Upload file](#)

Explore our live demo environment

Browse real-world data in a demo environment where you can explore search, observability, and security use cases like yours.

[Start exploring](#)



Other sample data sets

Sample eCommerce orders

Sample data, visualizations, and dashboards for tracking eCommerce orders.

[Add data](#)

Sample flight data

Sample data, visualizations, and dashboards for monitoring flight routes.

[Add data](#)

Sample web logs

Sample data, visualizations, and dashboards for monitoring web logs.

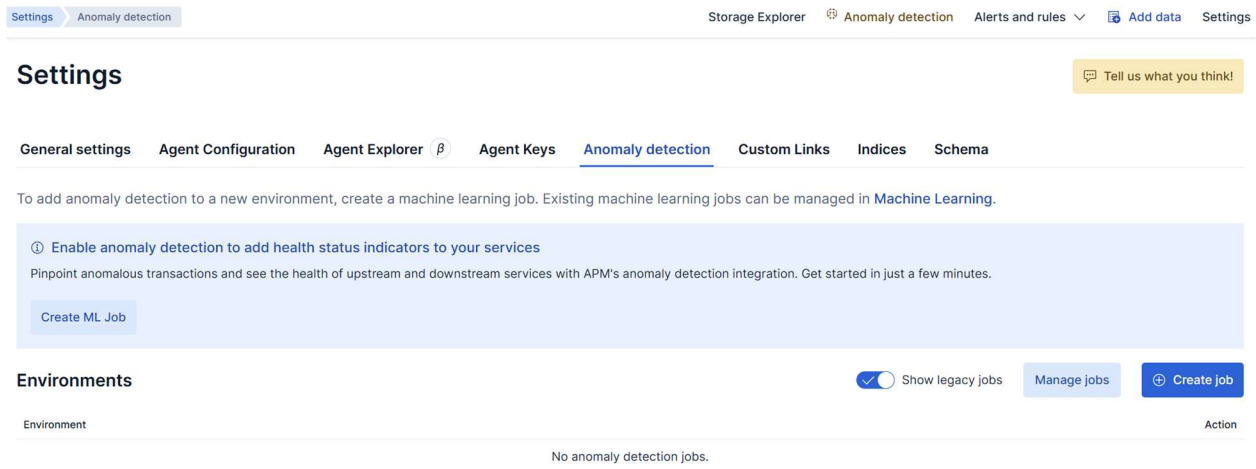
[Add data](#)

Deployment machine learning pada Elastic untuk mendeteksi anomaly cukup mudah, bisa klik menu **Anomaly Detection**

The screenshot shows the Elastic Observability interface. The top navigation bar includes 'Observability', 'APM', and 'Services'. The 'Anomaly detection' menu item is highlighted. The main content area is titled 'Services' and shows a list of services with columns for Name, Environment, Transaction type, Latency (avg.), Throughput, and Failed transaction rate. The services listed are frontend, load-generator, product-catalog, frontend-proxy, .NET cart, and currency. A search bar is present at the top of the services list. A notification banner at the top of the services section states: 'Enable anomaly detection to add health status indicators to your services. Pinpoint anomalous transactions and see the health of upstream and downstream services with APM's anomaly detection integration. Get started in just a few minutes.' There is a 'Learn more' button and a 'Dismiss' button for this notification.

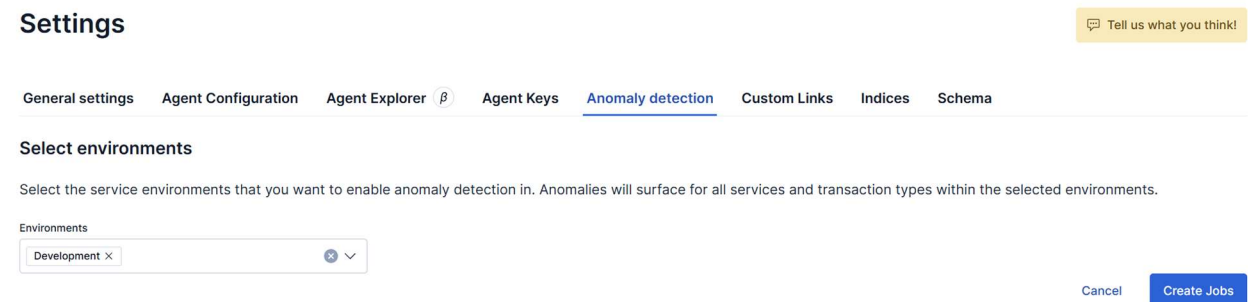
Name	Environment	Transaction ty...	Latency (avg.)	Throughput	Failed transaction rate
frontend	Not defined	request	459 ms	221.9 tpm	0%
load-generator	Not defined	external.http	372 ms	123.3 tpm	1.0%
product-catalog	Not defined	request	0.2 ms	122.2 tpm	0%
frontend-proxy	Not defined	request	395 ms	115.2 tpm	0.9%
.NET cart	Not defined	request	2.1 ms	55.6 tpm	0%
currency	Not defined	request	18 ms	16.1 tpm	0%

Kemudian pilih Create job



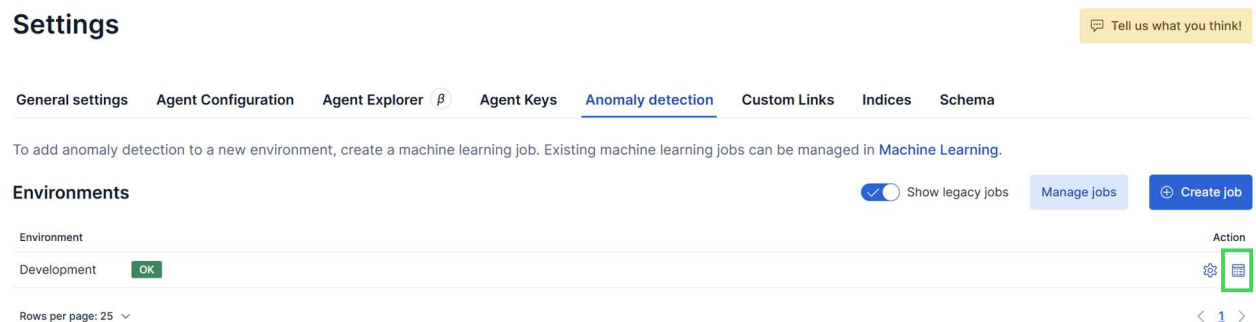
The screenshot shows the 'Settings' page for 'Anomaly detection'. The top navigation bar includes 'Settings', 'Storage Explorer', 'Anomaly detection', 'Alerts and rules', 'Add data', and 'Settings'. The main header is 'Settings'. Below it, there are tabs for 'General settings', 'Agent Configuration', 'Agent Explorer', 'Agent Keys', 'Anomaly detection' (selected), 'Custom Links', 'Indices', and 'Schema'. A message states: 'To add anomaly detection to a new environment, create a machine learning job. Existing machine learning jobs can be managed in [Machine Learning](#).' Below this, there is a section titled 'Enable anomaly detection to add health status indicators to your services' with a 'Create ML Job' button. At the bottom, there is a table titled 'Environments' with columns 'Environment' and 'Action'. The table is currently empty, showing 'No anomaly detection jobs.'.

Pilih Environment yang diinginkan. Ini mengacu pada variable environment yang kita define saat integrasi APM Agent dengan aplikasi Java. Setelah itu klik Create Jobs



The screenshot shows the 'Settings' page for 'Anomaly detection'. The top navigation bar includes 'Settings', 'Storage Explorer', 'Anomaly detection', 'Alerts and rules', 'Add data', and 'Settings'. The main header is 'Settings'. Below it, there are tabs for 'General settings', 'Agent Configuration', 'Agent Explorer', 'Agent Keys', 'Anomaly detection' (selected), 'Custom Links', 'Indices', and 'Schema'. A message states: 'To add anomaly detection to a new environment, create a machine learning job. Existing machine learning jobs can be managed in [Machine Learning](#).' Below this, there is a section titled 'Select environments' with a sub-header 'Select the service environments that you want to enable anomaly detection in. Anomalies will surface for all services and transaction types within the selected environments.' Below the sub-header, there is a dropdown menu labeled 'Environments' with 'Development' selected. To the right of the dropdown are 'Cancel' and 'Create Jobs' buttons.

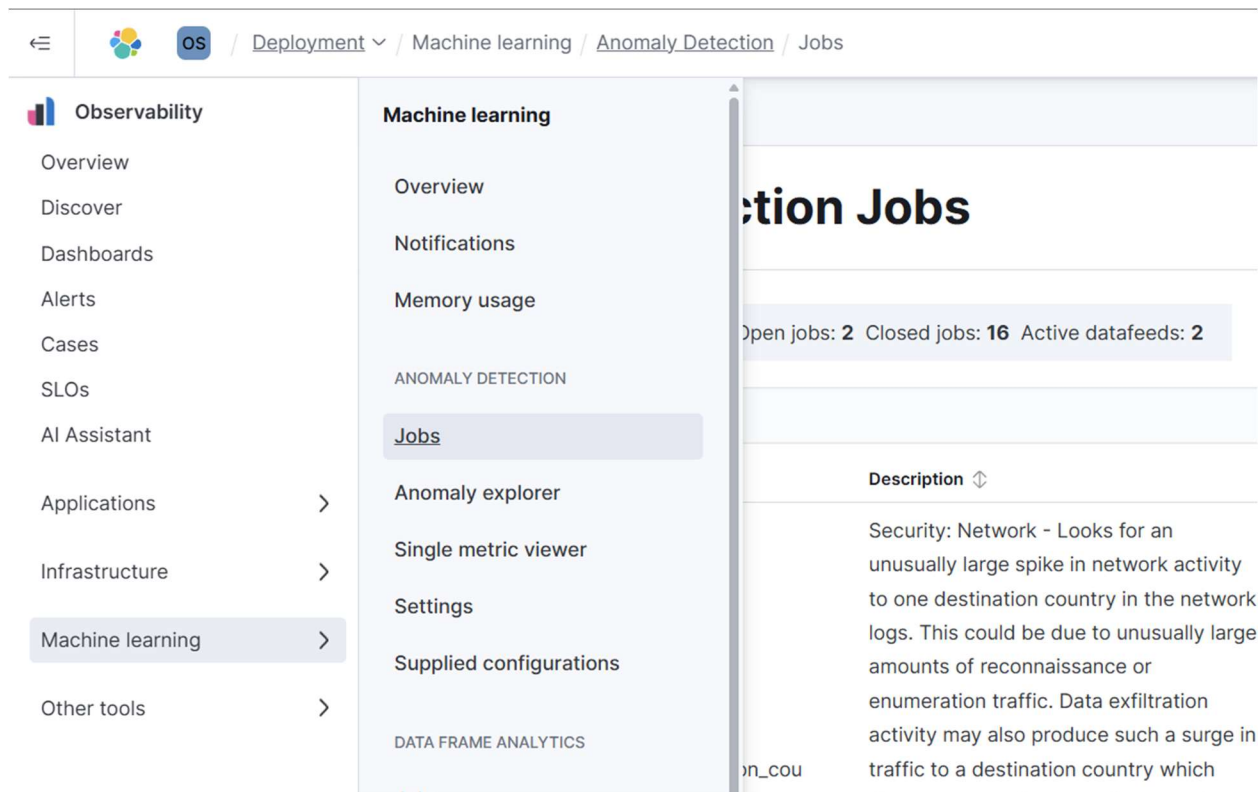
Klik tombol berikut untuk menampilkan Anomaly yang ada



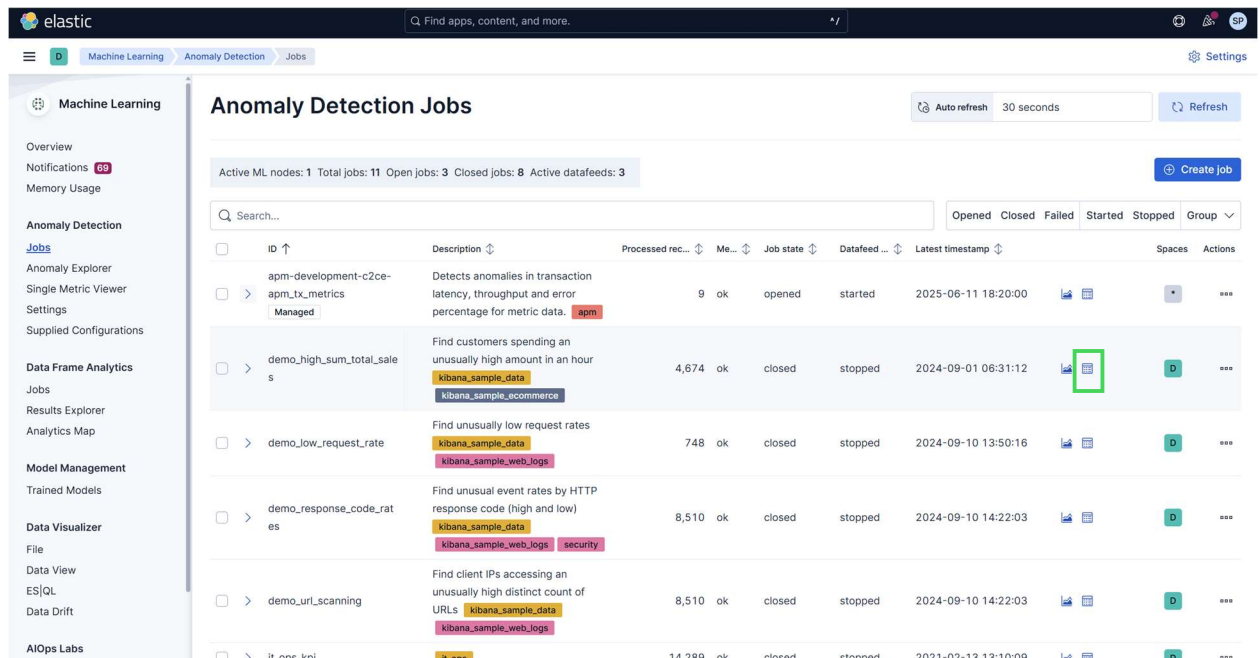
The screenshot shows the 'Settings' page for 'Anomaly detection'. The top navigation bar includes 'Settings', 'Storage Explorer', 'Anomaly detection', 'Alerts and rules', 'Add data', and 'Settings'. The main header is 'Settings'. Below it, there are tabs for 'General settings', 'Agent Configuration', 'Agent Explorer', 'Agent Keys', 'Anomaly detection' (selected), 'Custom Links', 'Indices', and 'Schema'. A message states: 'To add anomaly detection to a new environment, create a machine learning job. Existing machine learning jobs can be managed in [Machine Learning](#).' Below this, there is a section titled 'Environments' with a sub-header 'Select the service environments that you want to enable anomaly detection in. Anomalies will surface for all services and transaction types within the selected environments.' Below the sub-header, there is a dropdown menu labeled 'Environments' with 'Development' selected. To the right of the dropdown are 'Cancel' and 'Create Jobs' buttons. Below the dropdown, there is a table titled 'Environments' with columns 'Environment' and 'Action'. The table has one row with 'Development' in the 'Environment' column and a green 'OK' button in the 'Action' column. At the bottom, there is a 'Rows per page: 25' dropdown and a pagination bar showing '< 1 >'.

Elastic akan melakukan analisa secara berkala terhadap log yang masuk (Unsupervised ML)

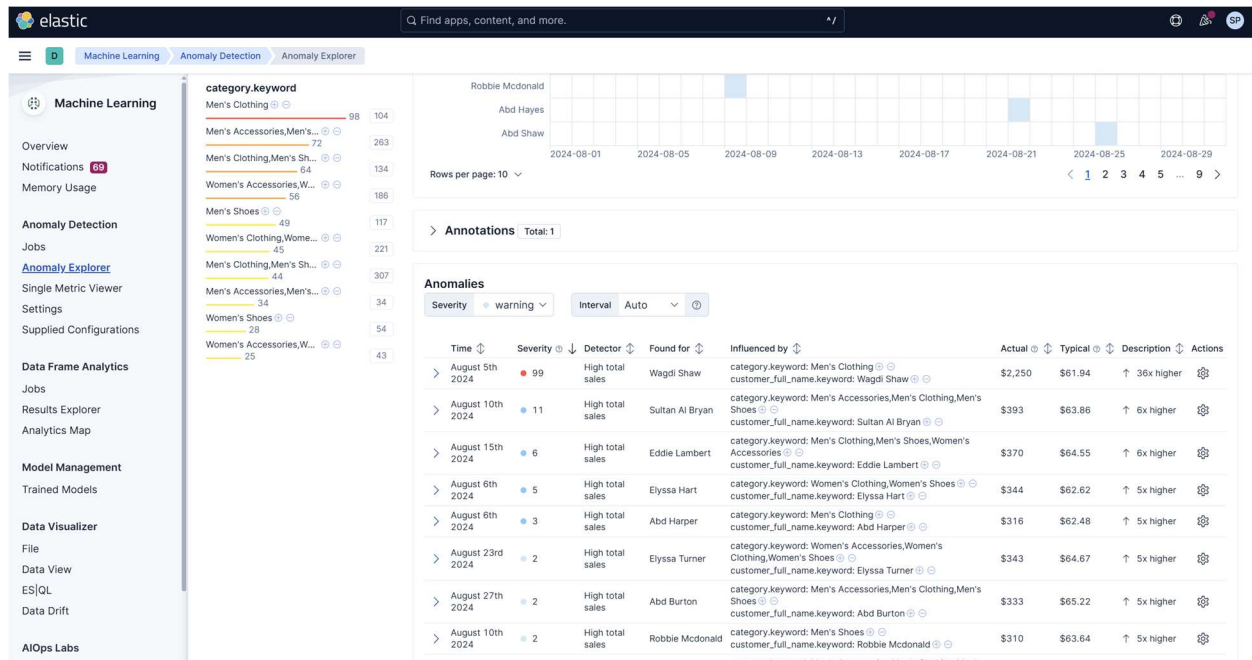
Untuk menampilkan job machine learning yang sudah di-deploy, navigasi ke menu **Machine learning > Jobs**



Silahkan pilih salah satu machine learning jobs yang sudah di-deploy



Kita dapat melihat detail dari anomali yang dideteksi

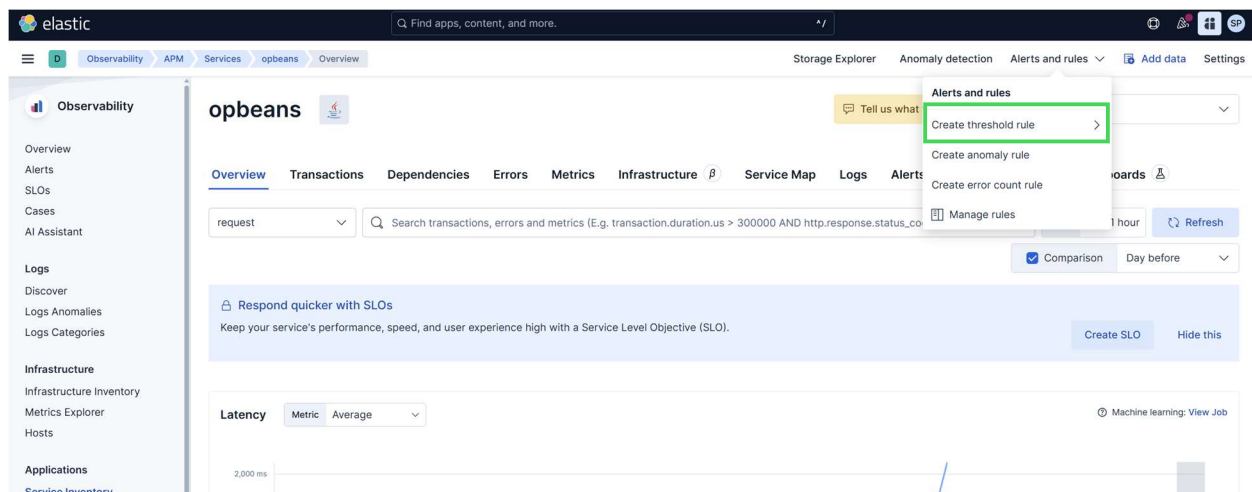


Alerting

Kibana memiliki fitur untuk mengirimkan alert apabila sebuah threshold, indicator, metrics, dll cocok dengan variable yang sudah kita konfigurasi.

Contohnya kita akan melakukan pengiriman alert melalui email apabila aplikasi Opbeans memiliki error

Navigasi ke menu APM, lalu klik **Alert and rules > Create threshold rule > Failed transaction rate**



Buat alert rule dengan kondisi berikut

Create rule

Name

Failed transaction rate threshold | opbeans

Tags

Optional

apm × service.name:opbeans ×

Failed transaction rate threshold

Alert when the rate of transaction errors in a service exceeds a defined threshold. [Learn more](#)

×

Use KQL Filter

SERVICE All

TYPE All

ENVIRONMENT All

NAME All

IS ABOVE 2%

FOR THE LAST 5 minutes

2.0%

1.5%

1.0%

0.5%

0%

Cancel

Show API request

✓ Save

Alert akan dikirimkan melalui email

Create rule

Check every1minute

> Advanced options

Actions

Select a connector type

Cases

D3 Security

Email

IBM Resilient

Index

Jira

Microsoft Teams

Observability AI Assistant

Opsgenie

PagerDuty

Server log

ServiceNow ITOM

ServiceNow ITSM

ServiceNow SecOps

Slack

Swimlane

TheHive

Torq

Webhook

xMatters

Cancel

Show API request

Save

Sesuaikan email penerima yang akan digunakan untuk menerima alert. Setelah itu klik Save

