



StackOps Workshop: 102

Kick start your observability journey today

Oct 2024

StackOps x Elastic

Agenda

Introduction to StackOps

Four Golden Signals

Synthetics (Hands-on)

-Break-

What is OpenTelemetry?

Application Performance Monitoring (APM) (Hands-on)

- Lunch -

What is SLO?

Service Level Objectives (SLO) (Hands-on)

Universal Profiling (Hands-on)

- Break -

Dashboarding with Kibana (Hands-on)

Teams

<https://go.gov.sg/stackop-102-teams>



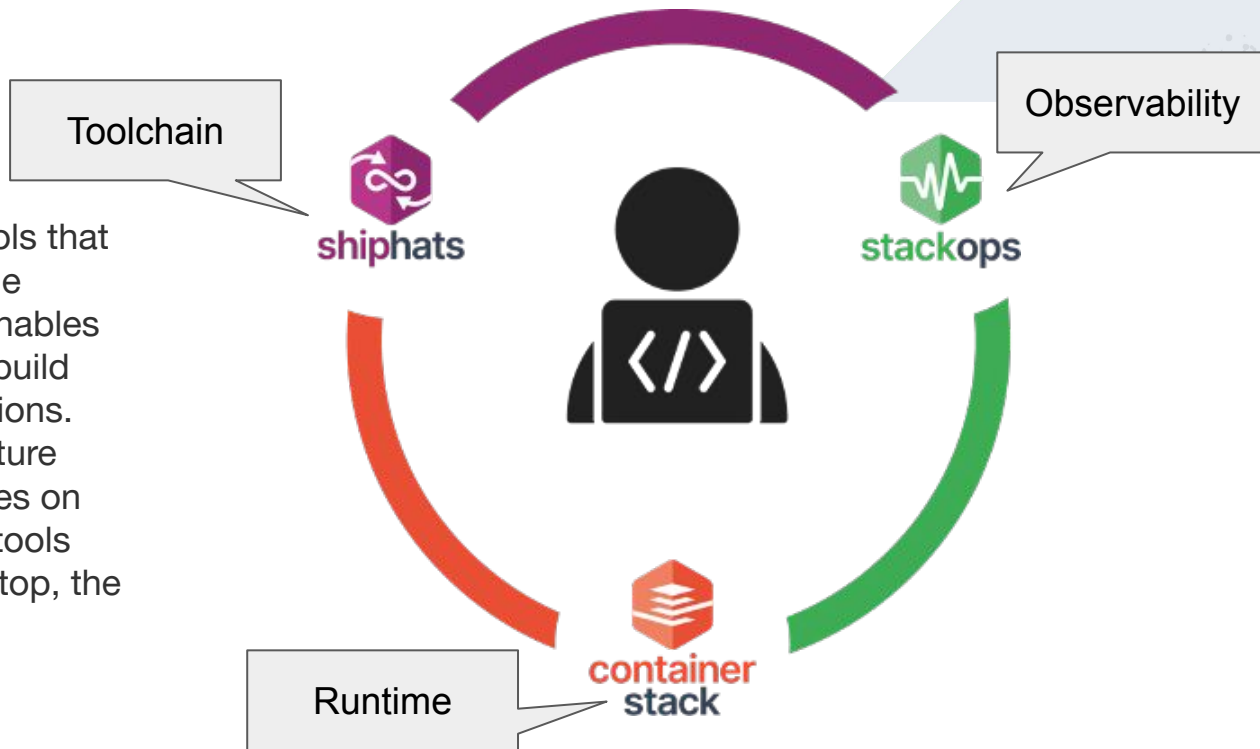
Q & A

<https://pigeonhole.at/STACKOPS102>

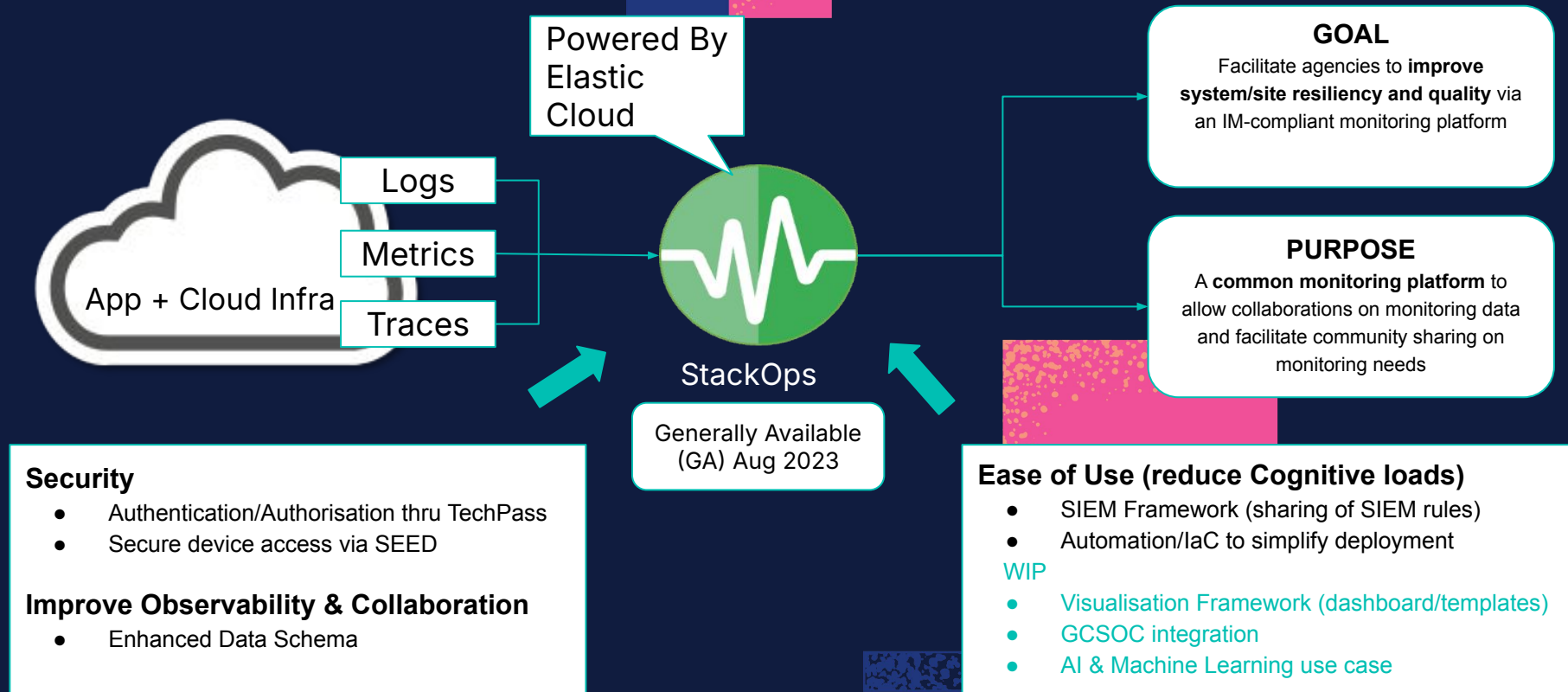


SG Tech Stack (SGTS) Base Layer

SGTS is a set of platform tools that streamlines and simplifies the development process and enables code reuse across WOG to build secure, high quality applications. SGTS has a 2-layer architecture where the Base Layer focuses on standardising development tools and environment and at the top, the Services Layer drives code reusability within WOG.



StackOps - SGTs Observability Solution





https://en.wikipedia.org/wiki/OODA_loop



Define: Observability

Observability pertains to how a system's internal state can be understood by examining its external outputs, especially its data.

In the context of modern application development, observability refers to the collection and analysis of data — logs, metrics and traces — from a wide variety of sources, to provide detailed insight into the behavior of applications running in your environments. It can be applied to any system that you build and want to monitor.

Observability is crucial for today's dynamic architectures and multi-cloud computing environments. It allows software engineers, IT, DevOps, and Site Reliability Engineering (SRE) teams to decode telemetry data. This is done with the help of visualizations — dashboards, service dependency maps, and distributed traces — as well as AIOps and machine learning approaches. With the right observability solution, you can understand how your applications, services, and infrastructure are performing to track and respond to issues.

Why: Observability

Observability is important because it allows teams to evaluate, monitor, and improve the performance of distributed IT systems. It is far more effective than traditional monitoring methods. An end-to-end observability platform can help breach silos and foster collaboration. Issues can be proactively diagnosed, analyzed, and traced back to their origins.

Observability gives IT organizations true visibility across the many data sources in cloud-native environments. The three pillars of observability are logs, metrics, and traces. Full-stack observability allows you to track the performance of your multi-cloud ecosystem in real-time and historically. You can synthesize data from endpoints and services generated by hardware, software, cloud infrastructure components, containers, open-source, microservices, and more.

Observability helps organizations:

- Discover and analyze the significance of performance incidents to their business
- Increase the efficiency of software development life cycles
- Accelerate problem resolution and root cause analysis
- Improve end-user experiences
- Fortify application security

What: Google SRE Chapter 6

Monitoring Distributed Systems - Four Golden Signals

01	Latency	• The time it takes to service a request • important to distinguish between the latency of successful requests and the latency of failed requests
02	Traffic	• Measure of how much demand is being placed on your system • For a web service, this measurement is usually HTTP requests per second
03	Errors	• The rate of requests that fail, either explicitly (e.g., HTTP 500s) • Implicitly (for example, an HTTP 200 success response, but coupled with the wrong content)
04	Saturation	• How "full" your service is. • A measure of your system fraction, emphasizing the resources that are most constrained • e.g., in a memory-constrained system, show memory; in an I/O-constrained system, show I/O

How: Observability

Observability works by continuously collecting performance data to create a complete, correlated record of every user request and transaction. The more observable your system, the more swiftly and accurately you can identify and trace a performance issue to its origins.

Key concepts for observability include:

Why?

Log data

Logs offer timestamped records of application events. A large organization can produce billions of logs in a day!

What?

Metrics

Metrics are time series data that measure application and system health and performance over time.

How?

Distributed tracing

Distributed tracing provides an end-to-end code-level record of every user request through the entire distributed architecture.

Where?

Dependency mapping

Dependency maps show how applications, services, and infrastructure components are connected.

Unknowns

Anomaly detection

Anomaly detection uses baselining, statistical analysis, or machine learning to provide an early warning of abnormal behavior in a system.

Google SRE Chapter 6 – Monitoring Distributed Systems

Black-box Monitoring

Testing externally visible behavior as a user would see it.

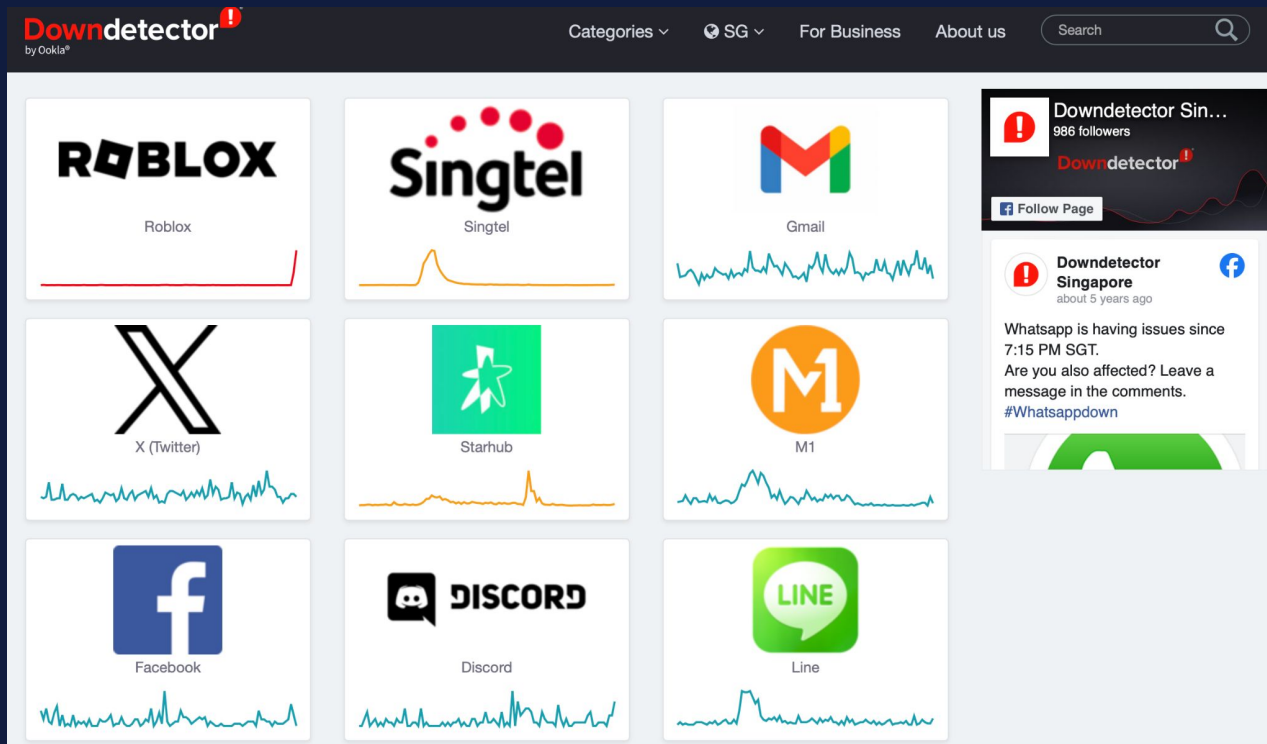
Easiest outside-in using Synthetics Monitoring

Synthetic monitoring is a method to test and monitor applications and systems by simulating user interactions. It uses scripts designed to mimic a real user's behavior, then uses the data to evaluate response times, error rates, and other performance metrics.

Considered a proactive monitoring solution, synthetic monitoring assesses how well an application will respond to user requests before a real user interacts with it. This allows developers to troubleshoot issues before their application is released and for IT operations teams to monitor critical user journeys in a consistent and repeatable way, once in production.

Unlike Real User Monitoring (RUM), a monitoring solution that requires human users, synthetic monitoring can be completely automated and run on a schedule. This allows you to detect issues, trigger alerts, and optimize system performance before real users run into issues. The result is a better end product and improved experience for users.

Black Box Monitoring = Downtetector :)



User Journeys

Synthetics Recorder

<https://www.elastic.co/guide/en/observability/current/synthetics-recorder.html>

<https://www.elastic.co/guide/en/observability/current/synthetics-journeys.html>

Make API Requests

<https://www.elastic.co/guide/en/observability/current/synthetics-create-test.html#synthetics-request-param>

Params and secrets

<https://www.elastic.co/guide/en/observability/current/synthetics-params-secrets.html>

Don't use actual user credentials or secrets, Scan with SingPass QR Code, PKI, MFA :)

<https://www.elastic.co/guide/en/observability/current/synthetics-params-secrets.html#synthetics-secrets-sensitive>

Hands-on Labs

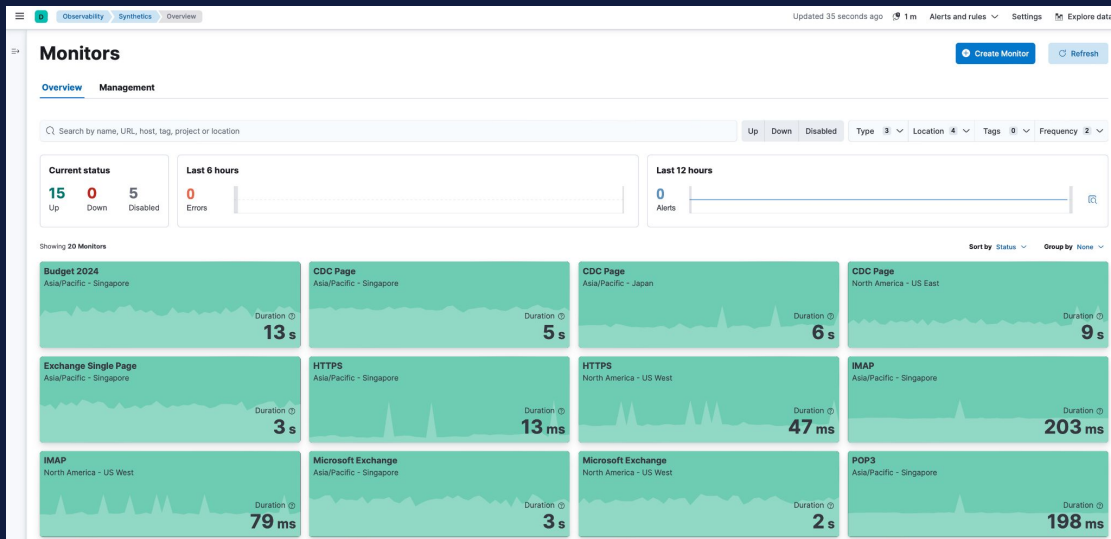
Lab 1

- Lab UI & Elastic Cloud UI Walkthrough
- Synthetic Monitoring

<https://ela.st/stackops-102-am>

Where to instrument the 4 golden signals?

Starting with Synthetic Monitoring Outside-in Monitoring



Observability - Synthetic Monitoring - 8.14

Learn how to use Elastic Observability for APM

Time limit: 1 hour

Learn how to use Elastic Observability for APM

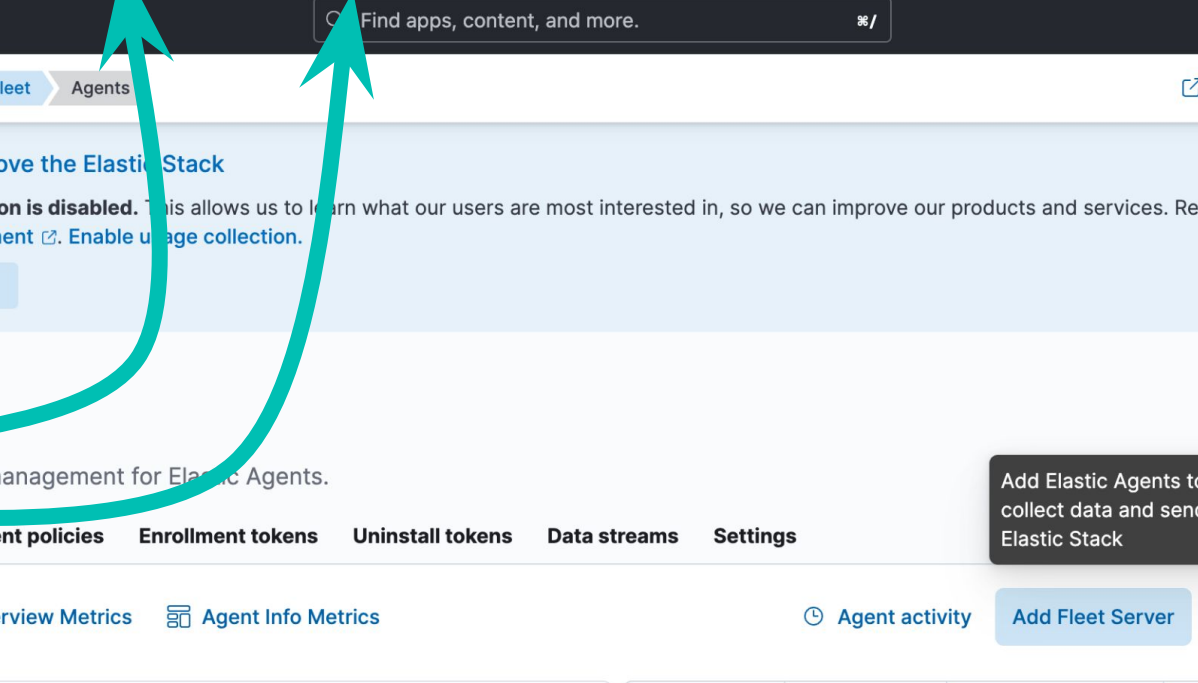
6 Single Page Monitor

Start

6 Multi Step Monitor

6 Catching Errors

Note: the different tabs



The screenshot shows the Kibana Fleet management interface. At the top, there is a dark navigation bar with the 'Kibana - Fleet' tab selected. Below this is the Elastic header with a search bar containing the text 'Find apps, content, and more.'. The main content area has a light blue background and features a 'Dismiss' button for a notification about usage collection. The 'Fleet' section title is prominently displayed. Below the title, there is a navigation bar with tabs for 'Agents', 'Agent policies', 'Enrollment tokens', 'Uninstall tokens', 'Data streams', and 'Settings'. The 'Agents' tab is currently active. On the right side of the 'Agents' tab, there is a tooltip that reads: 'Add Elastic Agents to your hosts to collect data and send it to the Elastic Stack'. At the bottom of the 'Agents' tab, there is a search bar with the placeholder text 'Filter your data using KQL syntax'. To the right of the search bar, there are status filters: 'Status 4', 'Tags 0', 'Agent policy 3', and 'Upgrade available'. The bottom of the screen shows a table with columns for 'Healthy', 'Unhealthy', 'Updating', and 'Offline', each with a count of 0.

B R E A K

Resume 10:50

What is OpenTelemetry?

Why Now?

OpenTelemetry is:

- An Observability framework and toolkit designed to create and manage telemetry data such as traces, metrics, and logs.
- Vendor- and tool-agnostic, meaning that it can be used with a broad variety of Observability backends, including open source tools like Jaeger and Prometheus, as well as commercial offerings.
- Not an observability backend like Jaeger, Prometheus, or other commercial vendors.
- Focused on the generation, collection, management, and export of telemetry. A major goal of OpenTelemetry is that you can easily instrument your applications or systems, no matter their language, infrastructure, or runtime environment. The storage and visualization of telemetry is intentionally left to other tools.

Language	Traces	Metrics	Logs
C++	Stable	Stable	Stable
C#/.NET	Stable	Stable	Stable
Erlang/Elixir	Stable	Development	Development
Go	Stable	Stable	Beta
Java	Stable	Stable	Stable
JavaScript	Stable	Stable	Development
PHP	Stable	Stable	Stable
Python	Stable	Stable	Development
Ruby	Stable	Development	Development
Rust	Beta	Alpha	Alpha
Swift	Stable	Development	Development

Why OpenTelemetry?

With the rise of cloud computing, microservices architectures, and increasingly complex business requirements, the need for software and infrastructure observability is greater than ever.

OpenTelemetry satisfies the need for observability while following two key principles:

- You own the data that you generate. There's no vendor lock-in.
- You only have to learn a single set of APIs and conventions.

Both principles combined grant teams and organizations the flexibility they need in today's modern computing world.

If you want to learn more, take a look at OpenTelemetry's [mission, vision, and values](#).

Main OpenTelemetry Components

OpenTelemetry consists of the following major components:

- A specification for all components
- A standard protocol that defines the shape of telemetry data
- Semantic conventions that define a standard naming scheme for common telemetry data types
- APIs that define how to generate telemetry data
- Language SDKs that implement the specification, APIs, and export of telemetry data
- A library ecosystem that implements instrumentation for common libraries and frameworks
- Automatic instrumentation components that generate telemetry data without requiring code changes
- The OpenTelemetry Collector, a proxy that receives, processes, and exports telemetry data
- Various other tools, such as the OpenTelemetry Operator for Kubernetes, OpenTelemetry Helm Charts, and community assets for FaaS

OpenTelemetry is used by a wide variety of libraries, services and apps that have OpenTelemetry integrated to provide observability by default.

OpenTelemetry is supported by numerous vendors, many of whom provide commercial support for OpenTelemetry and contribute to the project directly.

Observations About Observability



Data collection becomes commoditized via OpenTelemetry

- More services **expose OTel directly** or contribute Integrations to Collector
- APM Vendors begin to **drop proprietary APM agents and build upon OTel**

Observations About Observability

Vendor Differentiation Comes Through:

- Multi-signal analysis in unified User Experience
- ML-based single and multi-signal analytics for RCA
- Simple tools (BI for all!) for end users to analyze data
- Flexible deployment patterns (on-prem, Cloud, hybrid)
- Intelligent cost controls
- Workflow management (SLO, case management)
- Crowd-sourced (stackoverflow), LLM/chat-based AI Assistance

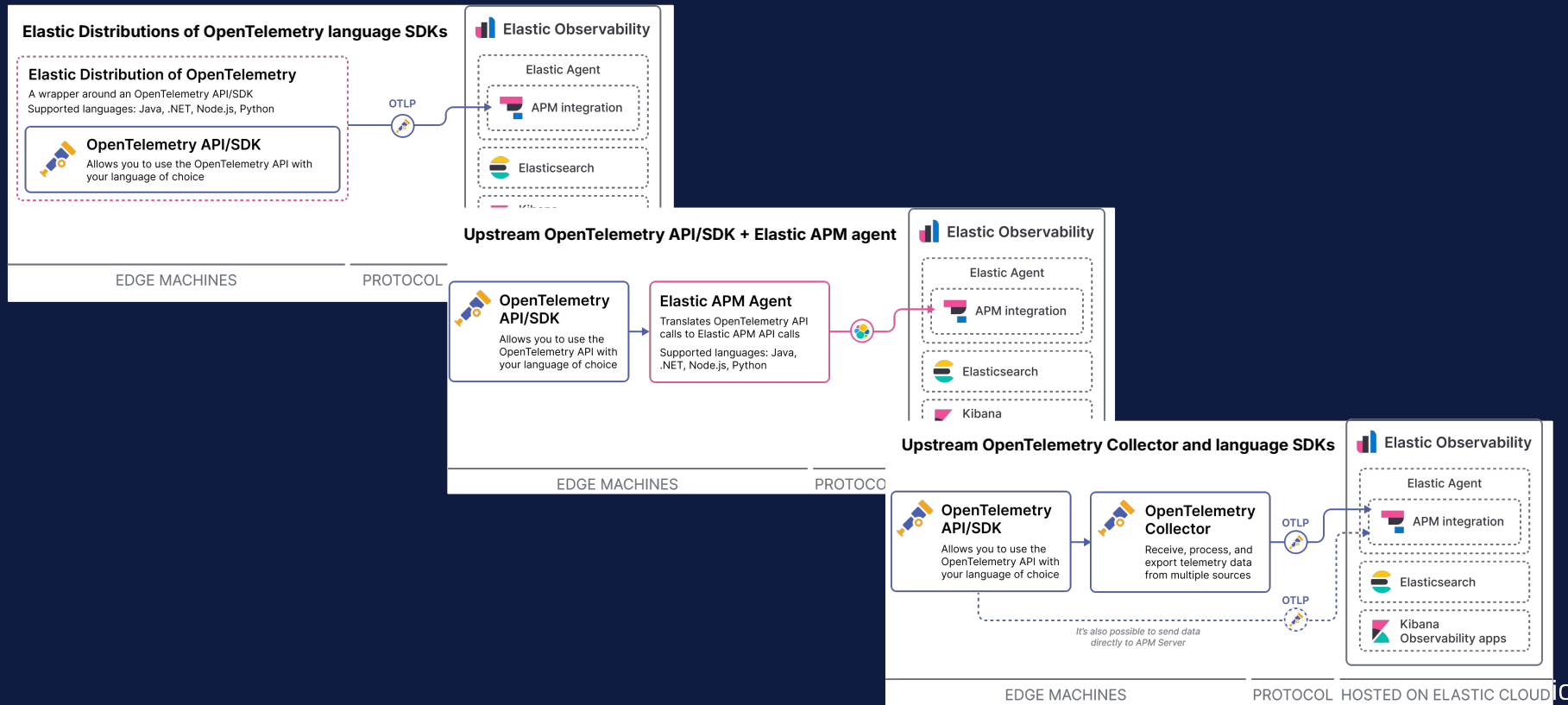
We are standardizing and investing on OpenTelemetry (OTel)

Make **OTel** the best data collection and processing framework

Base Elastic's data collection architecture for Observability and Security in **OTel**

Contribute to the success of **OTel** and help shape its future vs reacting

Elastic x OpenTelemetry



OpenTelemetry Distro

List of open source OpenTelemetry distributions maintained by third parties.

OpenTelemetry distributions are a way of customizing OpenTelemetry components so that they're easier to deploy and use with specific observability backends.

Any third-party can customize OpenTelemetry components with backend, vendor, or end-user specific changes. You don't have to use a distributions in order to use OpenTelemetry components, though distributions might facilitate usage under certain circumstances, such as specific vendor requirements.

<https://opentelemetry.io/ecosystem/distributions/>

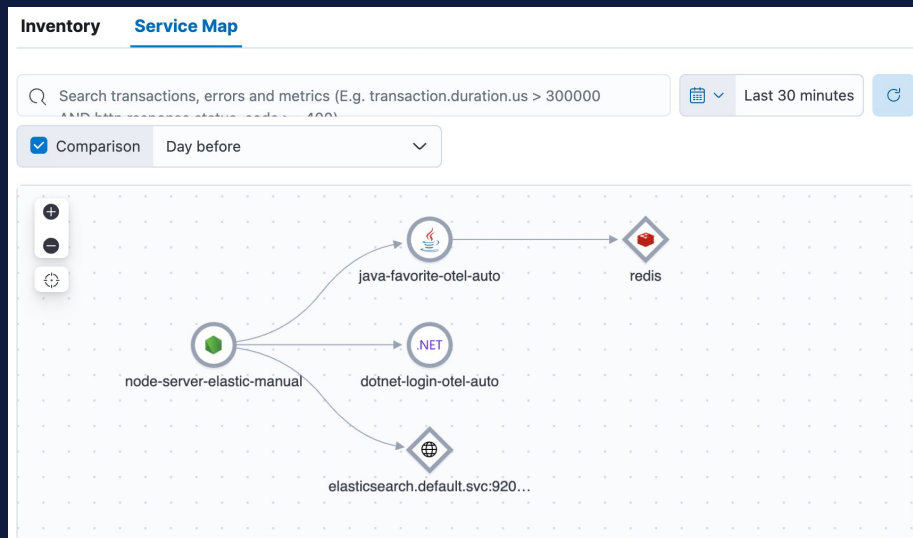
<https://www.elastic.co/observability-labs/blog/tag/opentelemetry>

<https://www.elastic.co/observability-labs/blog/elastic-opentelemetry-sdk-distributions>

<https://www.elastic.co/observability-labs/blog/opentelemetry-demo-with-the-elastic-distributions-of-opentelemetry>

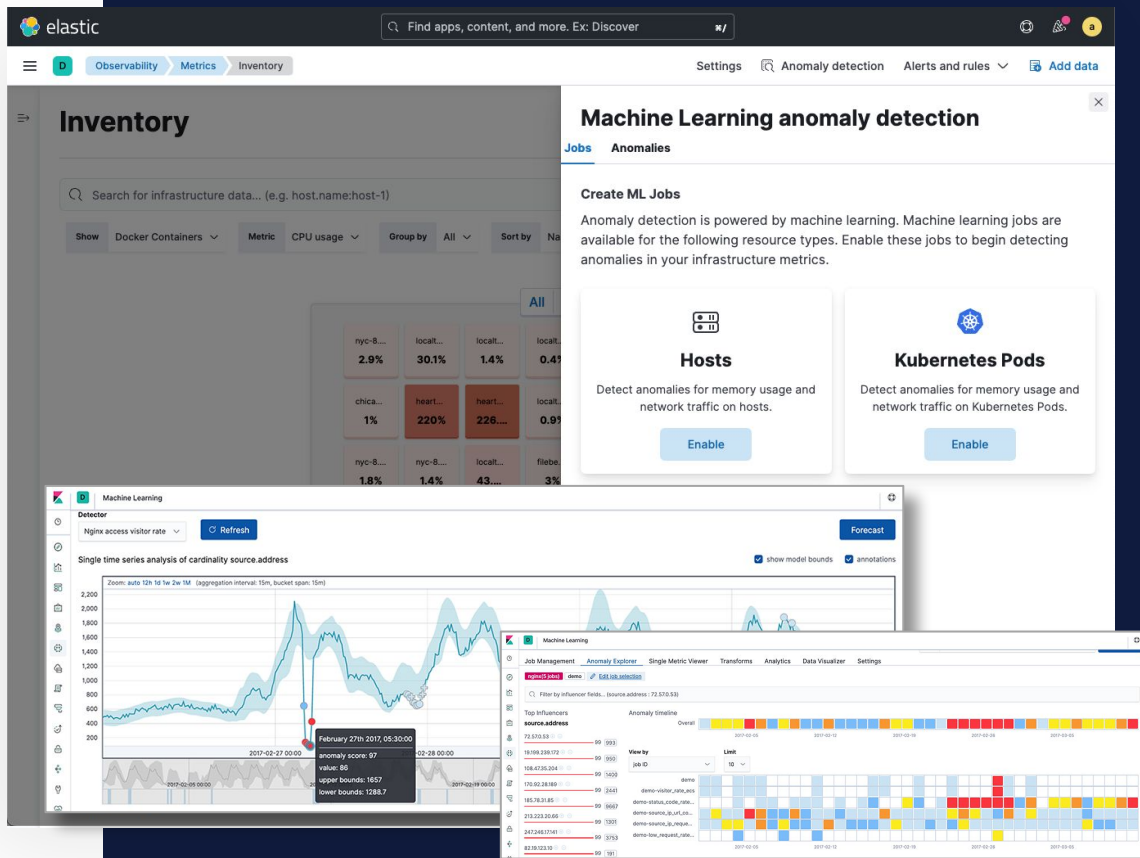
Application Performance Monitoring

- Improve code quality with end-to-end distributed tracing
- Quickly troubleshoot problems with ML-powered health indicators and anomaly detection
- Identify root cause of slowness and errors with on-demand correlations
- Out-of-the box support for OpenTelemetry and native agents



Actionable Insights

- Zero configuration (built-in) machine learning
- AI-driven anomaly detection and root cause analysis across all observability data
- Automatic APM correlations to find root causes
- Powerful search for "unknown unknowns"
- Reduces MTTD and MTTR



Where to instrument the 4 golden signals?

Observability = Logs, Traces and Metrics

Traces with Application Performance Monitoring

Elastic APM is an application performance monitoring system built on the Elastic Stack. It allows you to monitor software services and applications in real time, by collecting detailed performance information on response time for incoming requests, database queries, calls to caches, external HTTP requests, and more.

Elastic APM also automatically collects unhandled errors and exceptions. Errors are grouped based primarily on the stack trace, so you can identify new errors as they appear and keep an eye on how many times specific errors happen.

What is an APM agent?

The Elastic APM Python agent sends performance metrics and error logs to the APM Server. It has built-in support for Django and Flask performance metrics and error logging, as well as generic support of other WSGI frameworks for error logging.

<https://www.elastic.co/guide/en/apm/agent/python/current/getting-started.html>

APM Data Model

Elastic APM agents capture different types of information from within their instrumented applications. These are known as events, and can be spans, transactions, errors, or metrics.

HTTP request

Response



Do I need capture manually?

The Python Agent instruments your application to collect APM events in a few different ways:

- To collect data about incoming requests and background tasks, the Agent integrates with supported technologies like Django, Flask to make use of hooks and signals provided by the framework. These framework integrations require limited code changes in your application.
- To collect data from database drivers, HTTP libraries etc., we instrument certain functions and methods in these libraries. Instrumentations are set up automatically and do not require any code changes.
- In addition to APM and error data, the Python agent also collects system and application metrics in regular intervals. This collection happens in a background thread that is started by the agent.

How does the Agent work?

To gather APM events (called transactions and spans), errors and metrics, the Python agent instruments your application in a few different ways. These events, are then sent to the APM Server.

The APM Server converts them to a format suitable for Elasticsearch, and sends them to an Elasticsearch cluster. You can then use the APM app in Kibana to gain insight into latency issues and error culprits within your application.

```
$ pip install "elastic-apm[flask]"
```

```
from elasticapm.contrib.flask import ElasticAPM
```

```
app = Flask(__name__)
```

```
apm = ElasticAPM(app)
```

Full Streamlit Example:

<https://github.com/elastic/genai-workshop-codespaces/blob/main/streamlit/app.py>

Hands-on Labs

Lab 2

- Application Performance Monitoring

<https://ela.st/stackops-102-am>

APM

Challenges:

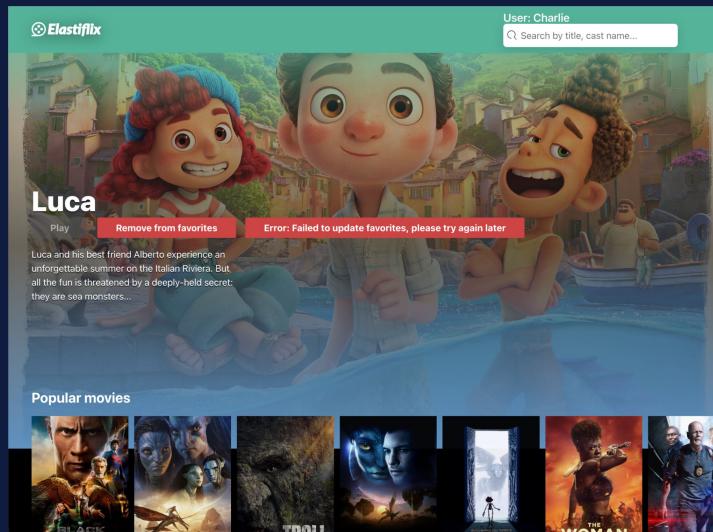
- Elastiflix - our demo application
- Elastic APM
- Extra Lab setting up Elastic APM agents
- <https://github.com/jiehonglim/apm-workshop>
- A broken application - 1
- Quiz - 1
- A healthy application - 1
- A broken application - 2
- Quiz - 2
- A healthy application - 2
- A broken application - 3
- Quiz - 3

Outline:

Learn how to use Elastic Observability for APM.

This track demonstrates how to:

- Use APM to gain additional insight
- Troubleshoot errors with APM
- Finding correlations in your data

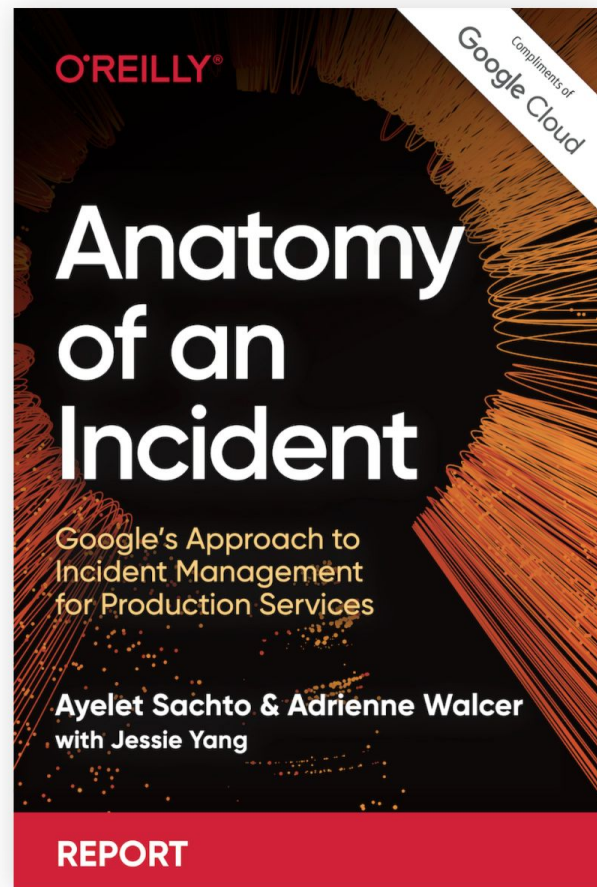


LUNCH : 12:30 - 13:30

Start 13:45

APM lab explanation after lunch :)

Anatomy of an Incident - Google - Site Reliability Engineering



When it comes to system design, failure is inevitable. Scientists and engineers implement solutions based on the available information, without a complete knowledge of the future. You can't always anticipate the next zero-day event, viral media trend, weather disaster, or shift in technology. But you can be prepared to respond when incidents like these affect your systems.

With this report, SRE and DevOps practitioners, IT managers, and engineering leaders will explore methods to help your organization prepare for, respond to, and recover from incidents. With advice from Ayelet Sachto, Adrienne Walcer, and Jessie Yang, you'll learn how to be prepared to handle failure if and when it happens.

Learn the stages of the incident management lifecycle: preparedness, response, recovery, and mitigation

- Deal proactively with incidents: issues that escalate beyond metrics and alerts
- Be prepared: practice disaster role playing and incident response exercises
- Learn the characteristics of the incident-response organizational structure
- Examine steps to recovery and mitigation after an incident has occurred
- Conduct postmortems to analyze what went wrong
- Explore a real-world example from Google: The Mayan Apocalypse
- Learn how to measure and reduce incidents impact
- Use postmortems as a tool for prevention and psychological safety

[PDF](#)[EPUB](#)[MOBI](#)

<https://sre.google/resources/practices-and-processes/anatomy-of-an-incident/>

Alert Fatigue ? Or Manage by SLO

Use effective alerts to avoid alert fatigue. Use pages when you need immediate action. Only the right responders—the specific team and owners—should get the alerts.

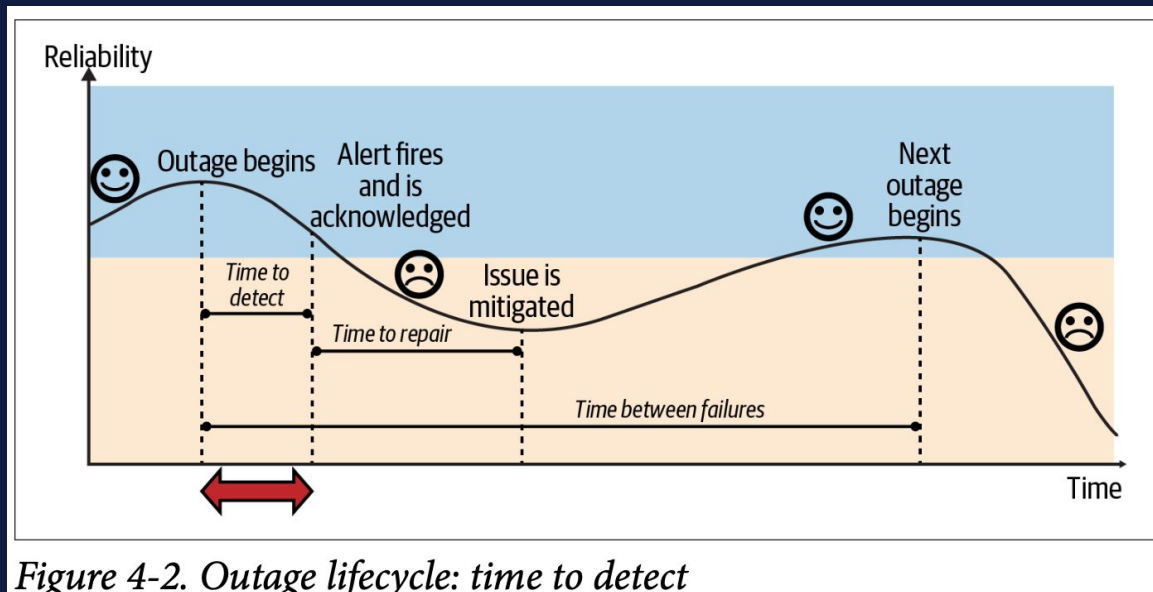


Figure 4-2. Outage lifecycle: time to detect

What matters to your users?

Service Level Indicator

The proportion of successful requests, as measured from the load balancer metrics.

Any HTTP status other than 500–599 is considered successful.

count of "api" http_requests which
do not have a 5XX status code
divided by
count of all "api" http_requests

<https://sre.google/workbook/slo-document/>

Create new SLO

1 Define SLI

Choose the SLI type

Custom Query

Index

Select a Data view

Timestamp field

Select a timestamp field

Query filter

Optional

Custom filter to apply on the index

Good query

Define the good events

Total query

Optional

Define the total events

Group by

Optional

Select an optional field to group by

QR Check In – Sports/Entertainment Venue

SLO: 1 failed out of 1000 successful check-ins = 99.9% successful check-ins

Error Budget: In this case: $100\% - 99.9\% = 0.1\%$

The error budget is the allowed number of errors or failures within a given time period. It's calculated as $100\% - \text{SLO}\%$. This means you have a 0.1% error budget. For every 1000 check-ins, you can tolerate 1 failure without violating your SLO.

Error Budget Calculation (using Occurrences method): $100,000 * 0.1\% = 100$ failures

Over a 30-day period, and you have 100,000 total check-ins during this time. You can have up to 100 failed check-ins over 30 days before you violate your SLO.

QR Check In – Sports/Entertainment Venue

Burn Rate:

The burn rate is how quickly you're using up your error budget compared to the rate you've allotted. A burn rate of 1 means you're using your error budget at exactly the expected rate. Higher than 1 means you're burning through your error budget too quickly.

Burn Rate = (Errors occurring / Time period) / (Total error budget / Total time)

For example, if you've had 20 failures in the first 3 days:

Burn Rate = (20 / 3) / (100 / 30) = 6.67

This burn rate of 6.67 means you're using your error budget 6.67 times faster than you should be.

Hands-on Labs

Lab 3

- Service Level Objectives (SLO)

<https://ela.st/stackops-102-pm>

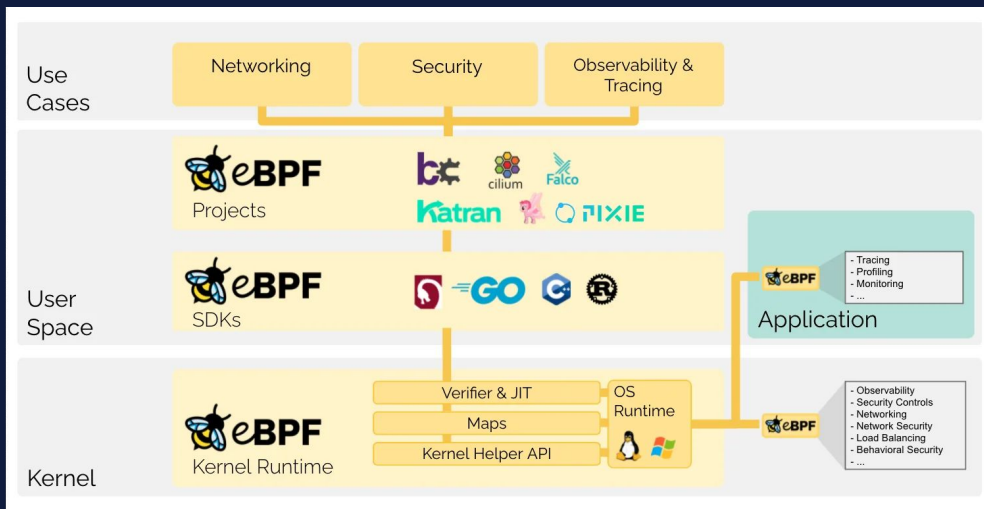
Universal Profiling

Next level of depth with Profiling

Why profile?

Elastic Universal Profiling is a whole-system, always-on, continuous profiling solution that eliminates the need for code instrumentation, recompilation, on-host debug symbols and service restarts.

<https://www.elastic.co/customers/appomni>



Observability - Profiling - 8.14

Learn how to use Elastic Observability for APM

Time limit: **1 hour**

Learn how to use Elastic Observability for APM

- 6 Setting up Universal Profiling
- 6 Investigating a Performance Problem
- 6 Fixing the Performance Problem

Start

Hands-on Labs

Lab 4

- Universal Profiling

<https://ela.st/stackops-102-pm>

B R E A K
Resumes 3:40

Real User Monitoring

Real User Monitoring

Why profile?

User Experience provides a way to quantify and analyze the perceived performance of your web application.

User Experience: Page load duration, Core Web Vitals, Load/view Distribution, Error Breakdown

Repo: <https://github.com/elastic/apm-agent-rum-js>

Docs: <https://www.elastic.co/guide/en/apm/agent/rum-js/current/index.html>

Demo page: <https://jiehonglim.github.io/apm-workshop/>



Install the Agent

Script Tags

Synchronous / Blocking Pattern

```
<script src="https://<your-cdn-host>.com/path/to/elastic-apm-rum.umd.min-<version>.js"
crossorigin></script>
<script>
  elasticApm.init({
    serviceName: '<instrumented-app>',
    serverUrl: '<apm-server-url>',
  })
</script>
```

Asynchronous / Non-Blocking Pattern

```
<script>
;(function(d, s, c) {
  var j = d.createElement(s),
      t = d.getElementsByTagName(s)[0]

  j.src = 'https://<your-cdn-host>.com/path/to/elastic-apm-rum.umd.min-<version>.js'
  j.onload = function() {elasticApm.init(c)}
  t.parentNode.insertBefore(j, t)
})(document, 'script', {serviceName: '<instrumented-app>', serverUrl: '<apm-server-url>'})
</script>
```

Hands-on Labs

Lab 5

- Dashboarding with Kibana

<https://ela.st/stackops-102-pm>

Agenda

-  Quick Intro to Elastic Stack and Kibana
-  Discover & AI Assistant
-  Visualizing Data With Kibana Lens
-  Dashboards
-  Alerting
-  Machine Learning
-  Challenges
-  Wrap-up

Lab 0: Setup Your Environment



Quick Intro: The Elastic Platform and Kibana

Data Exploration & Visualization



Who is the target audience?

- **Anyone** trying to make sense of data
- **Business** Analysts / Users
- **Data** Scientists
- Logs / Metrics **Analysts**
- **Security** Analysts
- NOC / SOC **Engineers**
- Data Service **Providers**





Simple. Intuitive. Flexible.



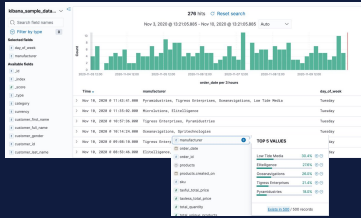
Time to dashboard



Make it beautiful ...

and actionable

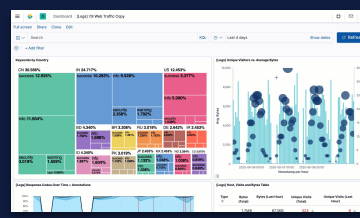
Kibana In A Nutshell



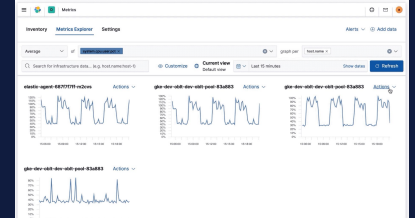
Discover



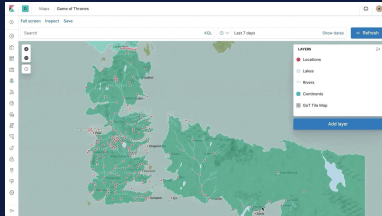
Lens



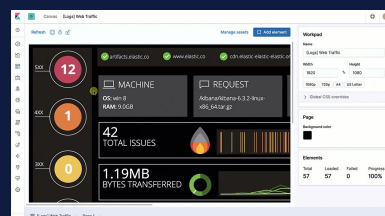
Dashboards & DrillDown



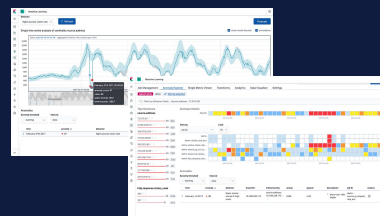
Alerting & Actions



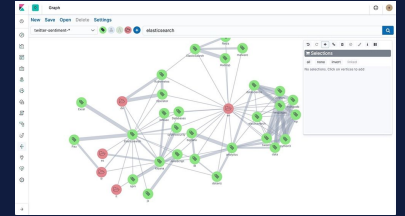
Maps & Visualization



Canvas



Machine Learning



Graph

and many more ...

Agenda

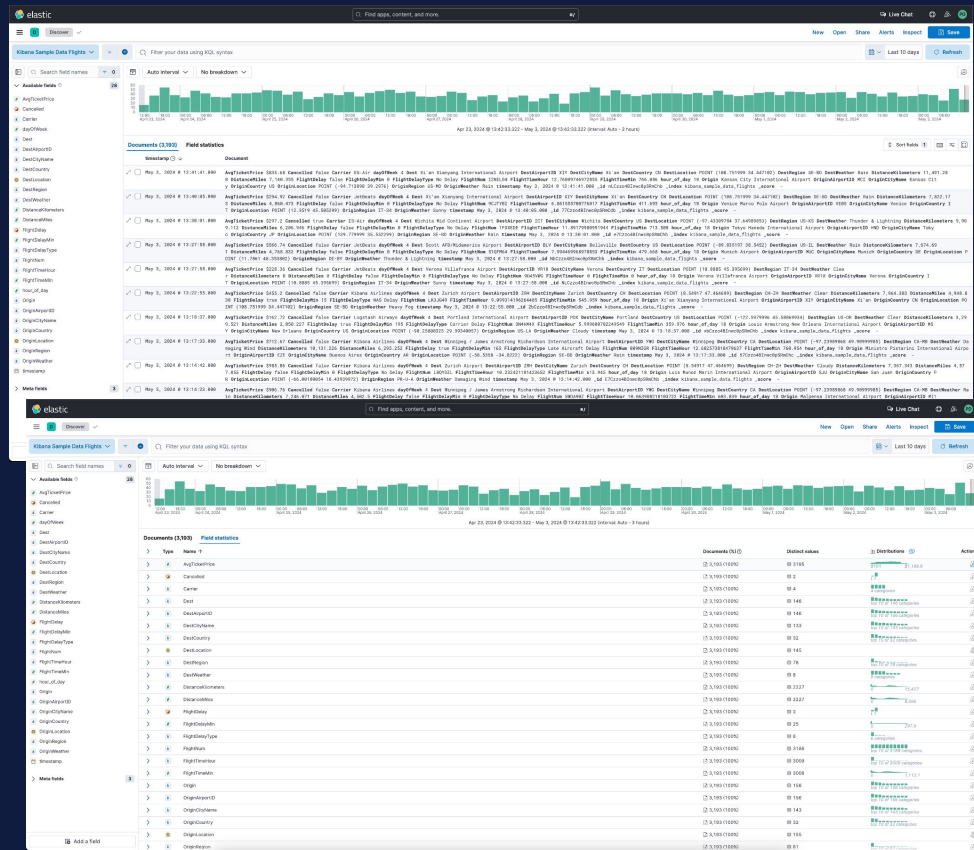
- Quick Intro to Elastic Stack and Kibana
- Discover & AI Assistant**
- Visualizing Data With Kibana Lens
- Dashboards
- Alerting
- Machine Learning
- Challenges
- Wrap-up

Explore Your Data With Kibana Discover

Kibana Discover

Enables you to explore your data

- Discover allows you to quickly browse your data
 - What does my data look like?
 - What is the distribution of data over time?
 - What values are most common?
- The Kibana Discover table can be customized and exported to CSV
- The data can be queried and filtered



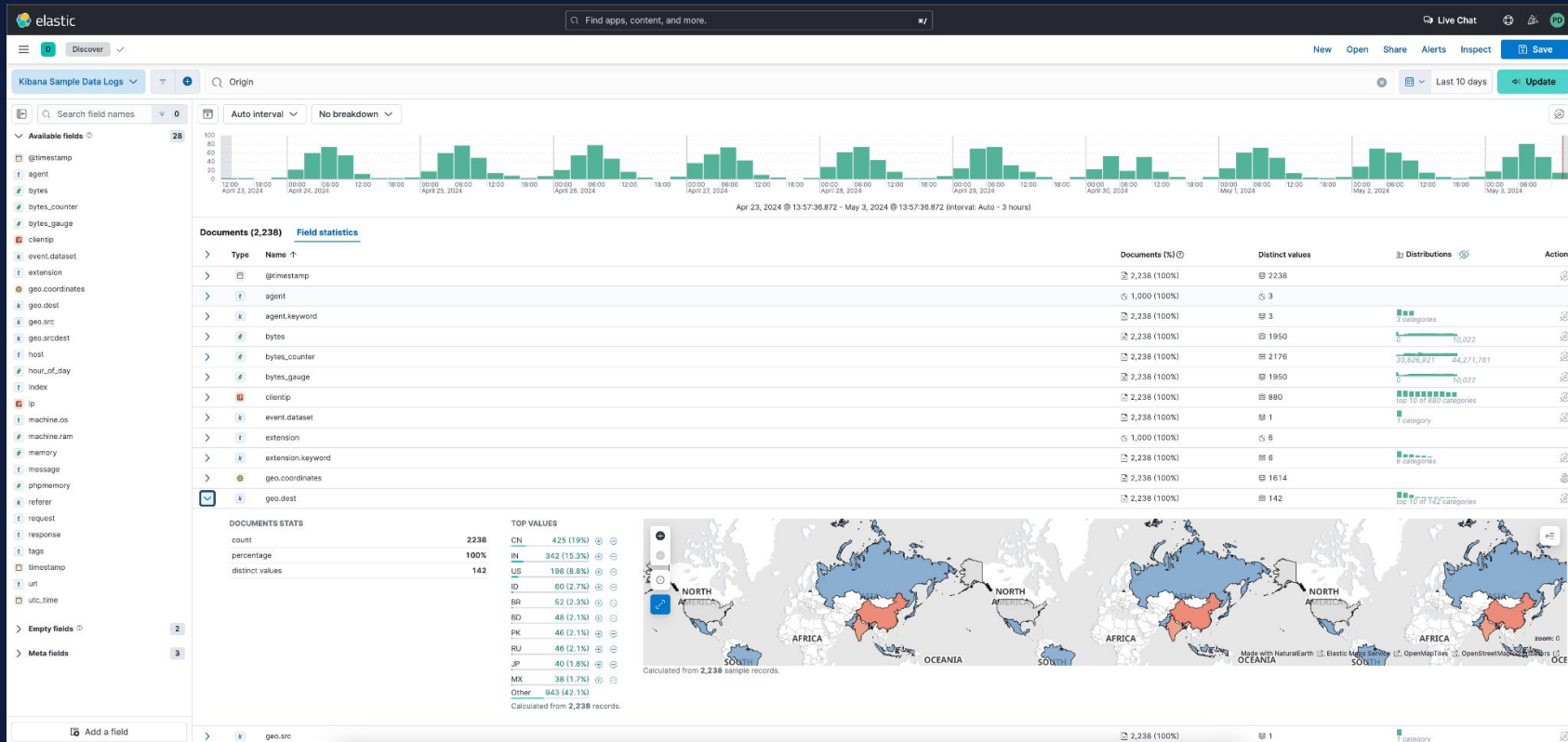
KQL

Query Your Data

- The default query language in Kibana is called KQL (Kibana Query Language)
- KQL supports:
 - searching for one or more terms, e.g. **response:200**
 - boolean queries using **or**, **and**, and **not**
 - ranges using **>**, **>=**, **<**, and **<=**
 - wildcards, e.g. **machine.os:win***



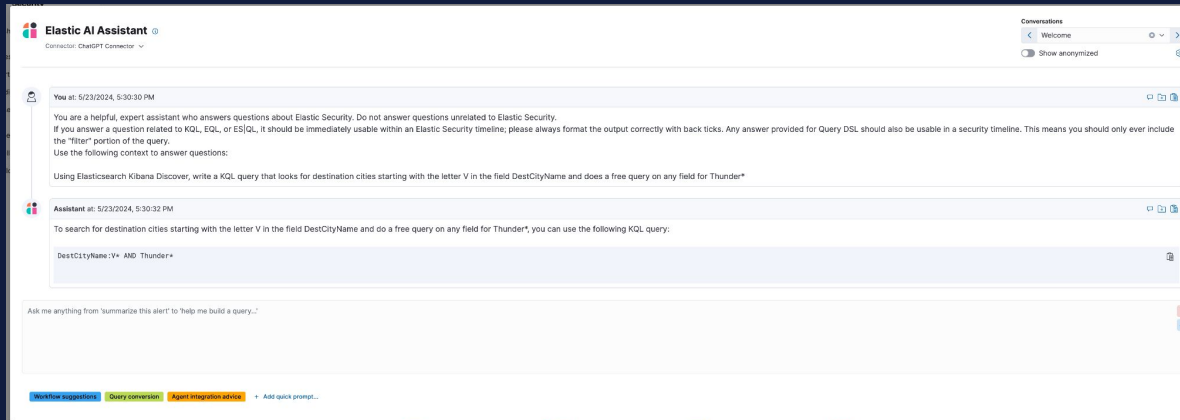
Field Statistics



Get Assisted with Elastic AI Assistant

Elastic AI Assistant

- Elastic AI Assistant is an interactive natural language chat interface that allows you to zero in on the most relevant information, faster.
- Uses Generative AI of your choice: OpenAI, Azure OpenAI Service, and Amazon Bedrock
- Select system prompts and add a custom system prompt
- Get help with query creation, triaging alerts, or understanding alerts



We make two types of AI investments

Each addresses different aspects of our customer-base and use cases



**Foundational technical
building blocks within
the Elastic Stack**



**AI-powered features
that enhance our
Solutions**

AI-powered Features in Elastic Solutions

Across Security, Search, and Observability



Security

Threat Intelligence
Automated Alerting
Security chatbot



Search

Semantic Search
Image Search
Sentiment Analysis
Recommendations
Named Entity Recognition



Observability

Anomaly Detection
AIOps
Interactive Root Cause Analysis
Integrated Case Management

Lab 1: Discover Kibana Discover



Agenda



Quick Intro to Elastic Stack and Kibana



Discover & AI Assistant



Visualizing Data With Kibana Lens



Dashboards



Alerting



Machine Learning



Challenges

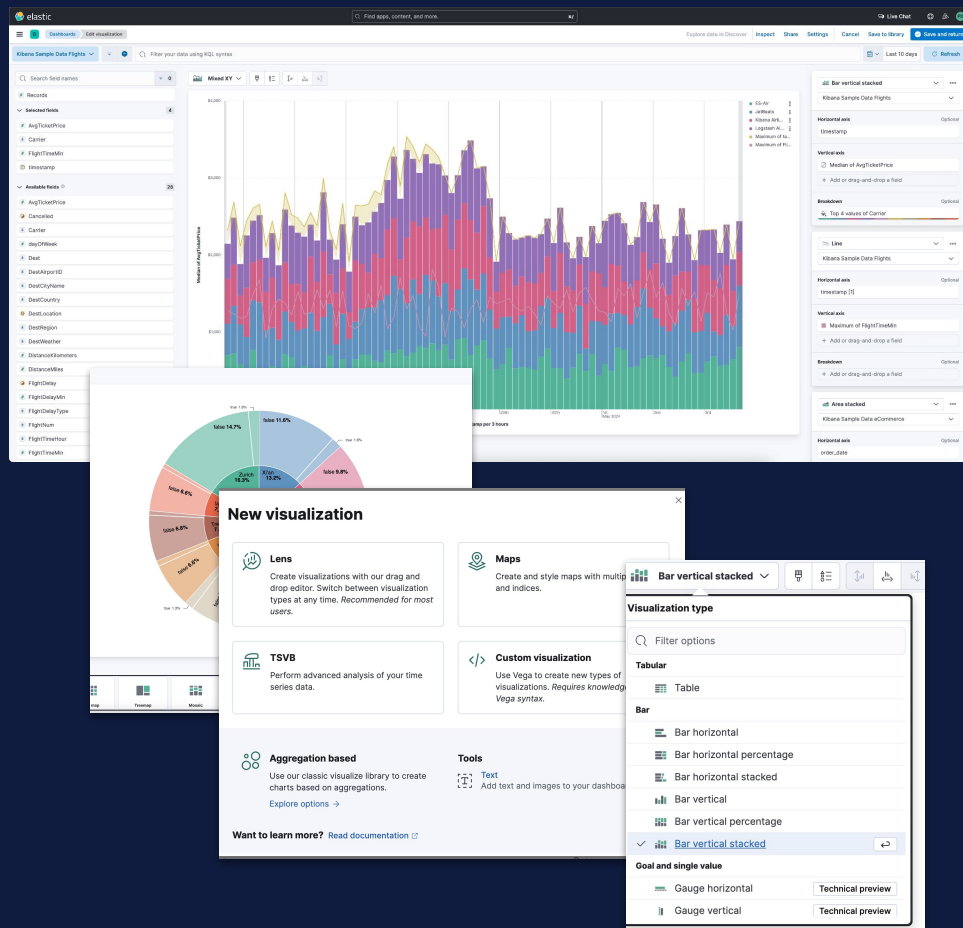


Wrap-up

Visualizing Data With Kibana Lens / Visualizations

Kibana Lens – *Explore Your Data*

- Kibana Lens provides an **easy-to-use** way of creating visualizations:
 - Simply drag and drop the fields you wish to visualize
 - Explore different types of visualizations
 - Create a visualization with just a few clicks
- Visualizations gets you from data to insights in just a **few clicks** — no prior **experience** in Kibana required.
- All visualizations created using Kibana Lens can be used on a **Kibana Dashboard**



Lab 2: Kibana Lens for Visualizations



Agenda

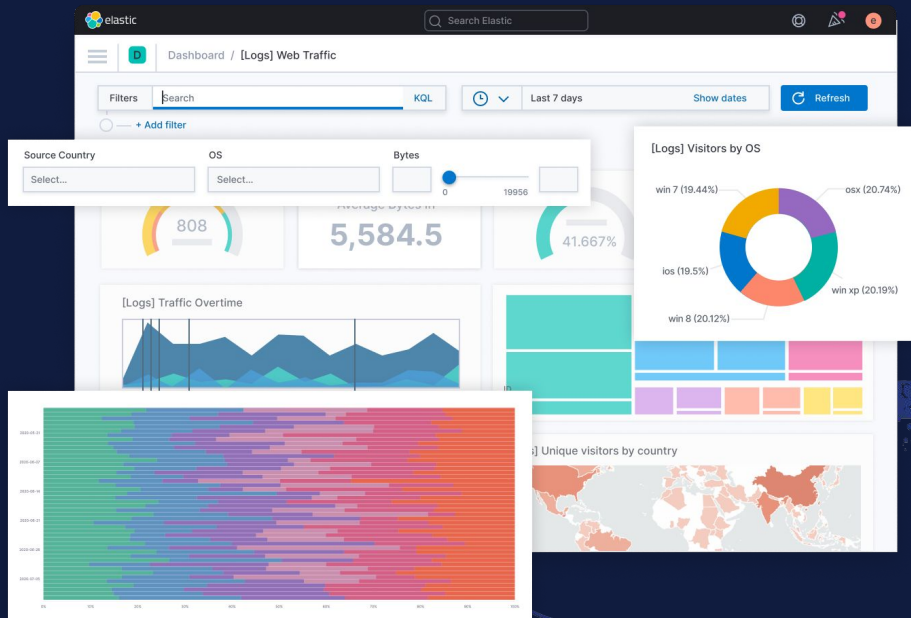
- Quick Intro to Elastic Stack and Kibana
- Discover & AI Assistant
- Visualizing Data With Kibana Lens
- Dashboards**
- Alerting
- Machine Learning
- Challenges
- Wrap-up

Put All Of Your
Visualizations On
Dashboards and Create
Reports

Dashboard

Collection of panels that you use to analyze your data

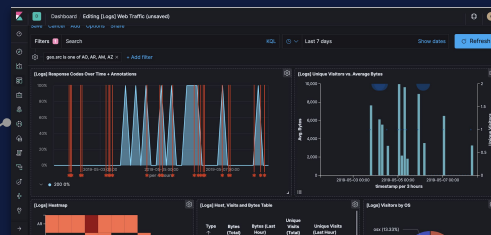
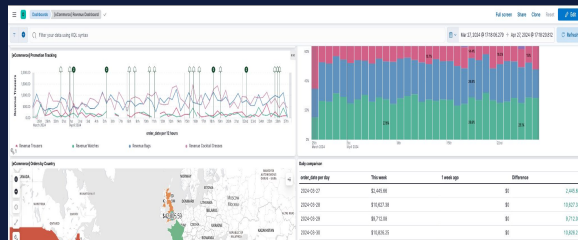
- Add **multiple panels** to see many aspects and views of your data in one place.
- **Arrange** panels for analysis and comparison.
- Add **text** and **images** to provide context to the panels and make them easy to consume.
- Create and **apply filters** to focus on the data you want to display.
- Control who can **use** your data, and **share** the dashboard with a small or large audience.
- **Generate reports** based on your findings.



Dashboard Drilldowns

Customize Kibana Flow and Take Actions

- Drill from one dashboard to another
- Drilldown to solutions: see the tracing behind a spike
- Drilldown and navigate to 3rd parties
- Initiate a flow from a panel








Host Name:

Agent ID (aid):

Time range:

Last 7 days

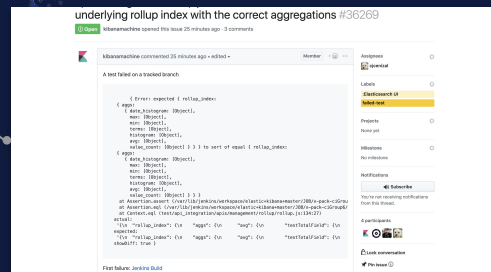
Submit

Host Info

Host Name	Local IP	External IP	Version	Device Type	Manufacturer	Model
CS-SE-CD-WIN7	43.130.141.237	107.77.196.135	Windows 7	Workstation	VMware, Inc.	VMware Virtu
CS-SE-CD-WIN7	43.130.141.237	65.25.95.175	Windows 7	Workstation	VMware, Inc.	VMware Virtu
CS-SE-CD-WIN7	192.168.186.130	98.255.110.28	Windows 7	Workstation	VMware, Inc.	VMware Virtu

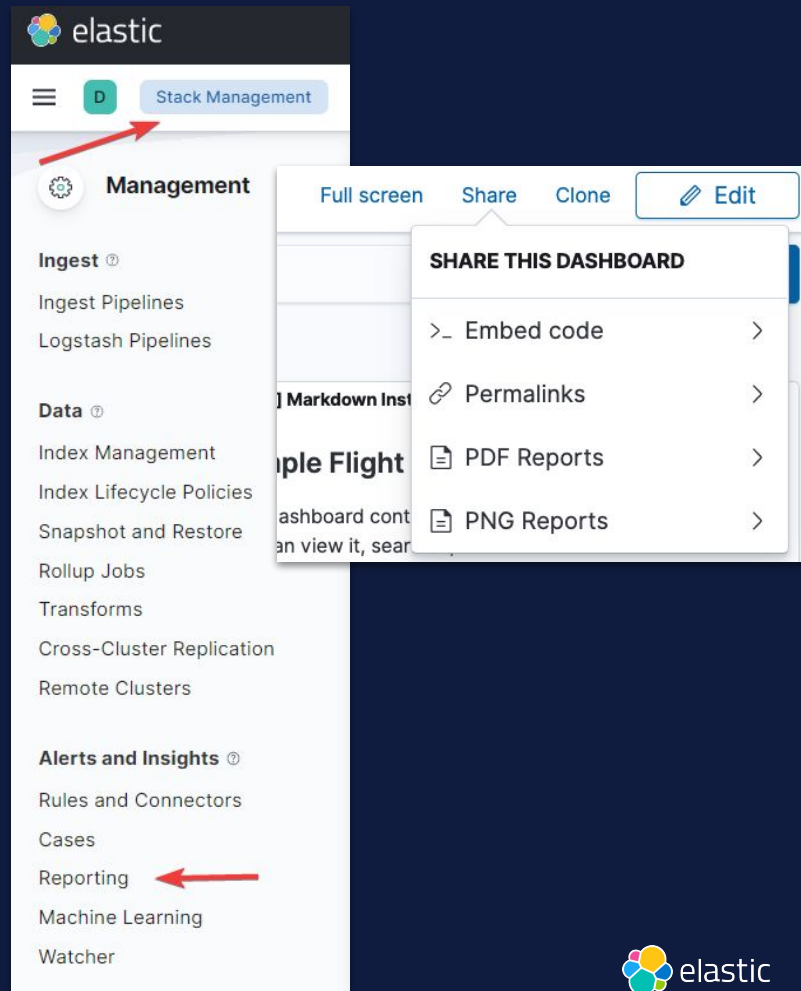
Detect History (last 14 days)

File Name	Scenario	Description
Activity Monitors	Known Malware	Activated an attempt to detect backups, which is often indicative of ransomware activity.
explorer.exe	Known Malware	Detecting the deletion of backups often indicative of ransomware activity.
explorer.exe	Process Terminated	Terminated a process related to the deletion of backups, which is often indicative of ransomware activity.



Reporting & Sharing

- The resulting reports can be found under "Alerts and Insights", accessible via the "Stack Management" menu option.
- Share your dashboard outside of Kibana:
 - Embed the code in a web page. Users must have Kibana access to view an embedded dashboard.
 - Share a direct link to a Kibana dashboard
 - Generate a PDF report
 - Generate a PNG report



Lab 3: Kibana Dashboards



Agenda

- Quick Intro to Elastic Stack and Kibana
- Discover & AI Assistant
- Visualizing Data With Kibana Lens
- Dashboards
- Alerting**
- Machine Learning
- Challenges
- Wrap-up

Alerting In The Elastic Platform

Alerting In The Elastic Platform

Choose Your Use Case And Take Action

- **Alerting Everywhere:**

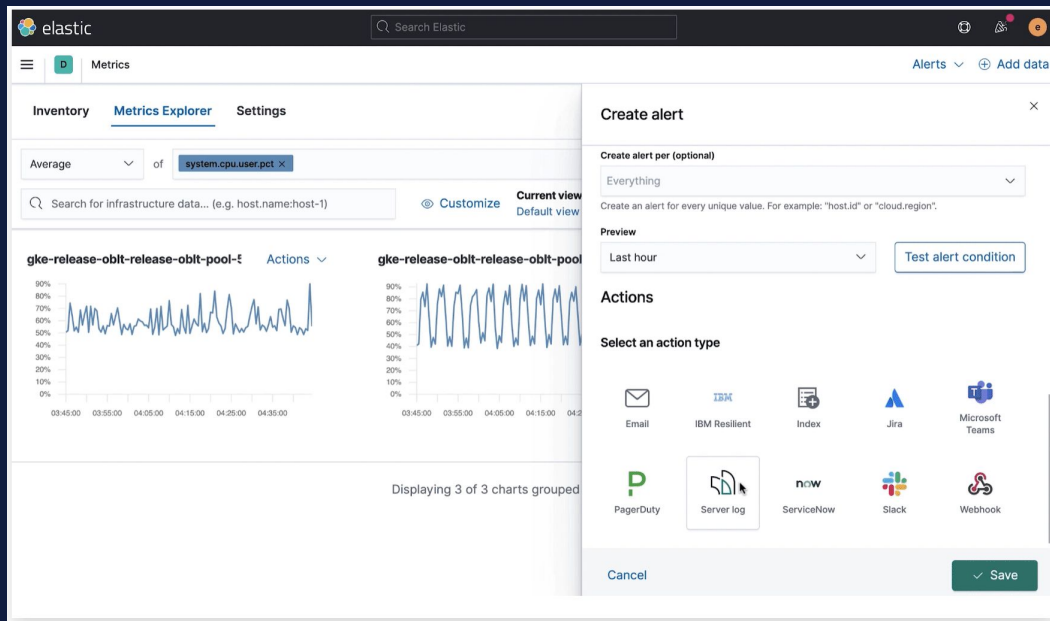
- Find alerts everywhere, from Elastic Logs, SIEM, APM, Uptime, Infrastructure, and Maps, to features like monitoring and machine learning to a multitude of Kibana dashboards
- Alerts and notifications play a critical role, yet each has unique needs for detecting conditions, expressing them, and showing them in context

- **Making Sense of Alerts**

- Uptime alerts may show a service outage
- APM alerts explain which transaction caused it
- Monitoring alerts pinpoint why it happened.

- **Detection and Action**

- Make alerts actionable
- Leverage machine learning
- Connect the alerts
- Look for fully autonomous actions
 - Auto-scaling
 - self-healing



Lab 4: Kibana Alerting

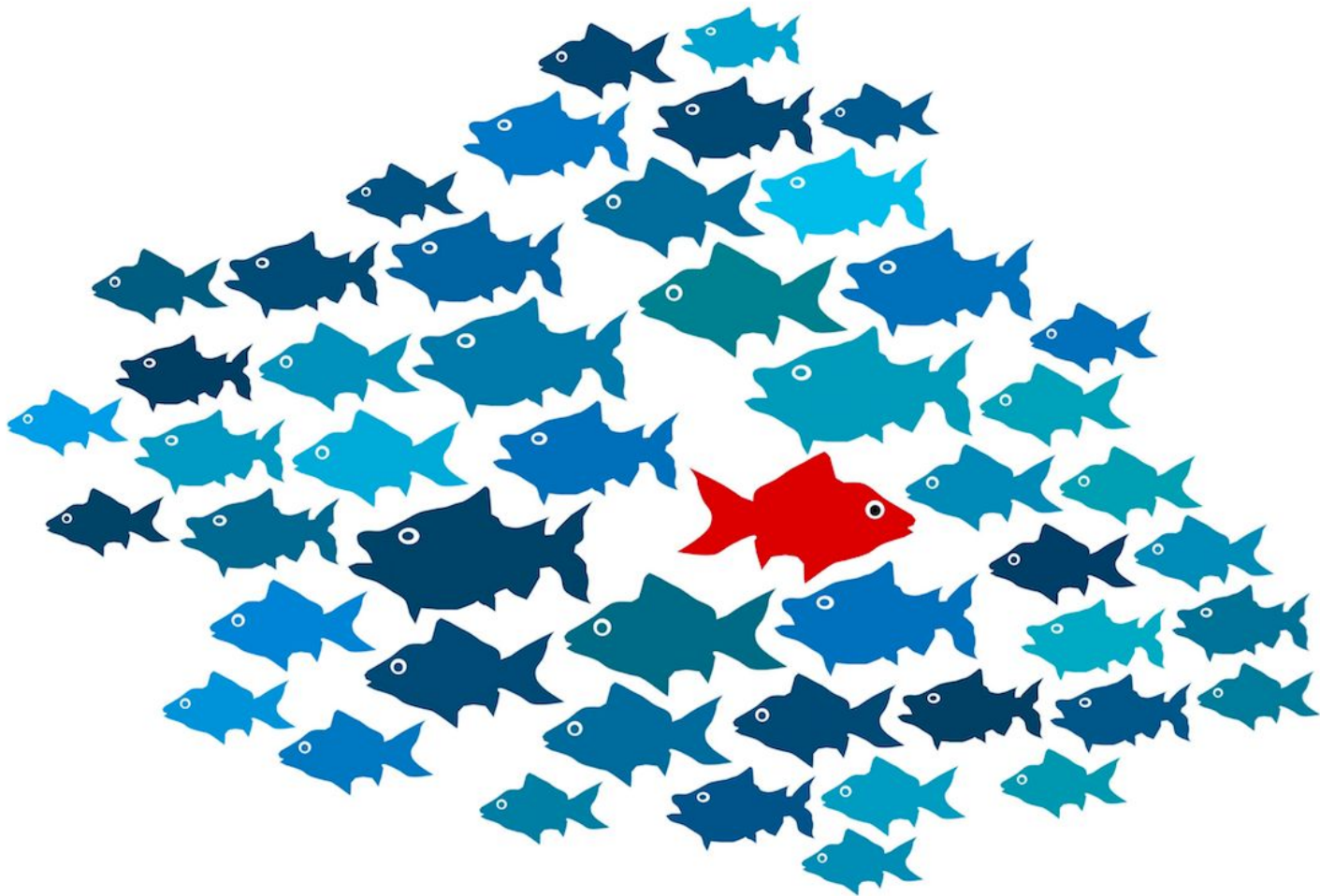


Agenda

- Quick Intro to Elastic Stack and Kibana
- Discover & AI Assistant
- Visualizing Data With Kibana Lens
- Dashboards
- Alerting
- Machine Learning**
- Challenges
- Wrap-up

Let The Machine Learn and Bring You Insights





Machine Learning Anomaly Detection

10 Years Development & Industry Leading Technology

Unsupervised Machine Learning(ML)

Automatically detect anomalies, outliers from group, and rare events

Sophisticated ML Job User Interface

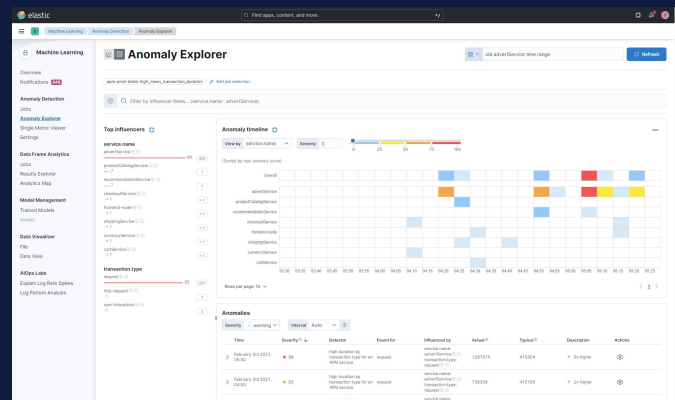
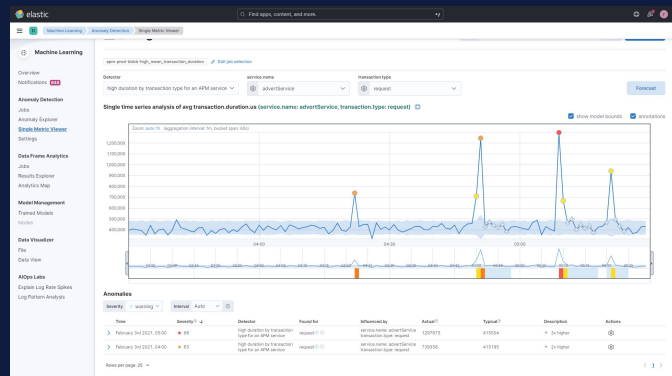
Interactive views of model and anomaly scoring

Root Cause Analysis

Report on factors influencing anomalies

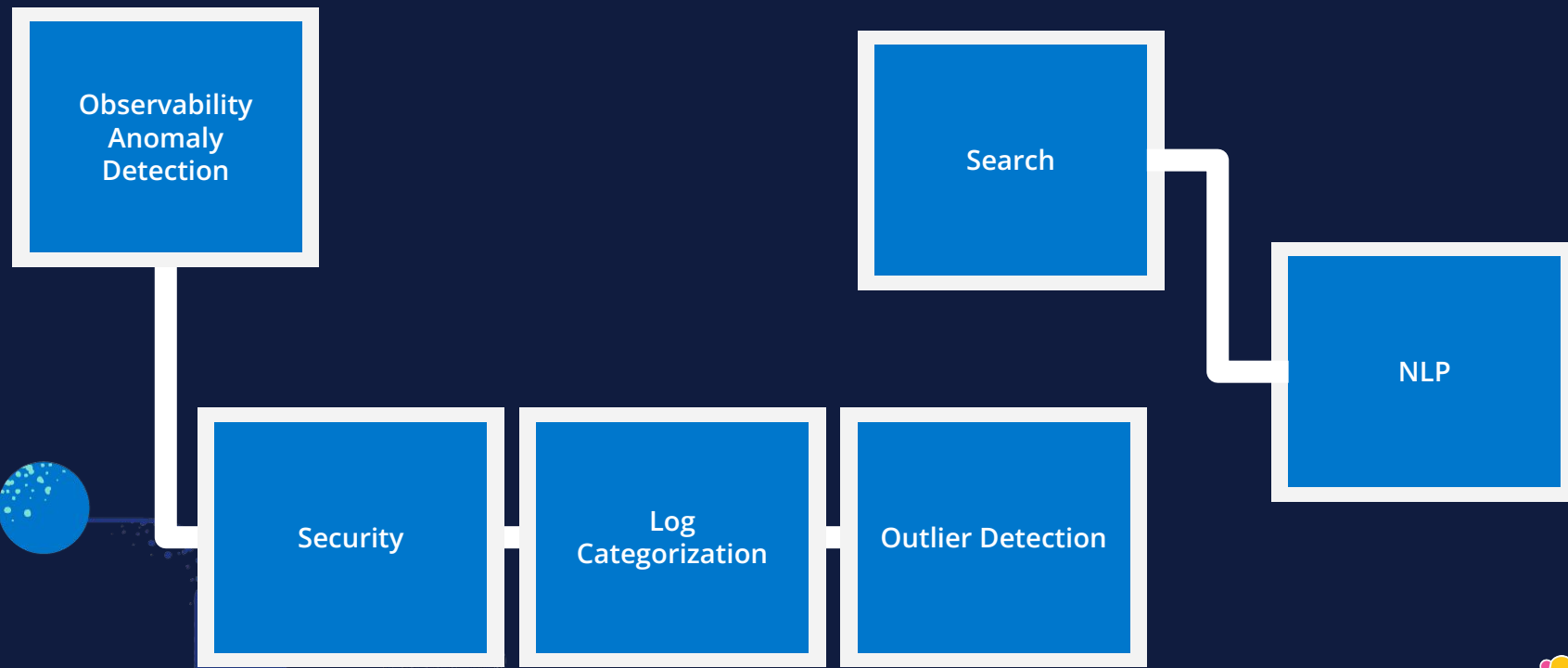
On-Demand Forecasting

Forecast out time series metrics



Some Machine Learning Scenarios

Out of the Box Examples.



Lab 5: Kibana Machine Learning



Agenda

- Quick Intro to Elastic Stack and Kibana
- Discover & AI Assistant
- Visualizing Data With Kibana Lens
- Dashboards
- Alerting
- Machine Learning
- Challenges**
- Wrap-up

Agenda

- Quick Intro to Elastic Stack and Kibana
- Discover & AI Assistant
- Visualizing Data With Kibana Lens
- Dashboards
- Alerting
- Machine Learning
- Challenges
- **Wrap-up**

Post Course Survey for StackOps App Monitoring

Only applicable for Singapore Public Service officers or participants from Government-linked companies (GLCs)



<https://go.gov.sg/d300-20241009>

<https://go.gov.sg/d300-20241009>

Call to Actions

Contact Us	enquiries_cep@tech.gov.sg
StackOps Docs	https://go.gov.sg/stackops-docs/
Future Workshops	https://go.gov.sg/stackops-learning-events
Telegram Community	Scan the code https://go.gov.sg/stackops-community



Future Workshops

Level	Stage	Topics Covered
101	Overview and Ingestion <u>Sign up</u>	Introduction to StackOps StackOps Architecture Host/Log Monitoring (Hands-on) Short form Kibana (Hands-on)
103	SIEM	Creation of Rules Alerts integration Exception Handlings
201	Automation, Integration & Customising	Log/ingest Anomaly Detection (ML) Integration ML based Rules

Q & A

<https://pigeonhole.at/STACKOPS102>



Thank You