CONF42 SRE | 2021

Improve observability and operational performance for your container workloads

Suraj Muraleedharan

Senior Consultant, Amazon Web Services

Agenda

Introduction

Understand the state and behavior of the application

Importance of SLIs, SLOs and SLAs

Summary

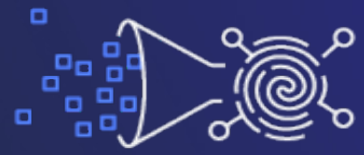
Introduction

What is observability?

- Observability describes how well you can understand what is happening in a system
  - Is my system up or down? Is it fast or slow as experienced by my end users?
  - What KPIs and SLAs should we establish, and how do we know if they're being met?
- Let's start with three primary signals

Logs

Metrics



Traces

Is observability new for software systems?

- Logs are used for debugging and identifying the root cause of issues
- Metrics are used during application or infrastructure issues
- Traces are used to understand path traversed by a request
- All these signals, together or individually were being used during development, testing ,operation and maintenance of software systems

What are logs?

- Logs can be split into different categories -
  - Application logs
  - System logs
  - Audit logs
  - Infrastructure logs
- Use the right logging levels – TRACE, DEBUG,WARN, INFO and ERROR
- Avoid printing sensitive information in logs
- Define a consistent logging pattern which allows log analytics
- Output logs to standard output for container-based workloads

What are metrics?

- Metrics are data about the performance of your systems
- These can be aggregate type data, numeric representations or point in time observation of a system
- Metrics can be used in two ways
  - Real-time monitoring and alerting
  - Trend analysis over time and long-term planning

What are traces?

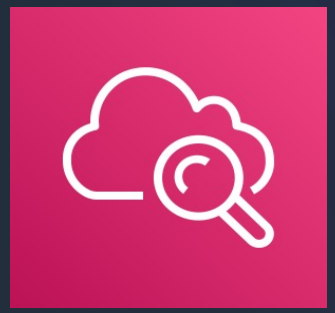
- Distributed tracing helps understand what happened during a distributed transaction. For example, request initiated by an end-user and its effects across all downstream services

Understand the state and behavior of the application

Use case – Monitor a microservices architecture

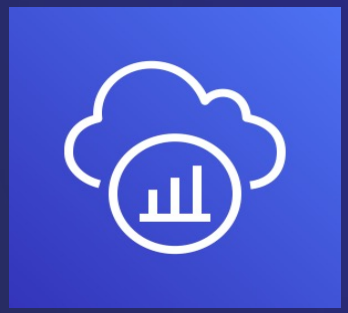
- A microservices architecture consists of many different distributed parts that have to be monitored. You can use Amazon CloudWatch to collect and track metrics, centralize and monitor log files, set alarms, and automatically react to changes in your AWS environment
- In many cases, a set of microservices work together to handle a request. Imagine a complex system consisting of tens of microservices in which an error occurs in one of the services in the call chain. Even if every microservice is logging properly and logs are consolidated in a central system, it can be difficult to find all relevant log messages
- The central idea behind AWS X-Ray is the use of correlation IDs, which are unique identifiers attached to all requests and messages related to a specific event chain.

What is Amazon CloudWatch and AWS X-Ray?



Amazon CloudWatch

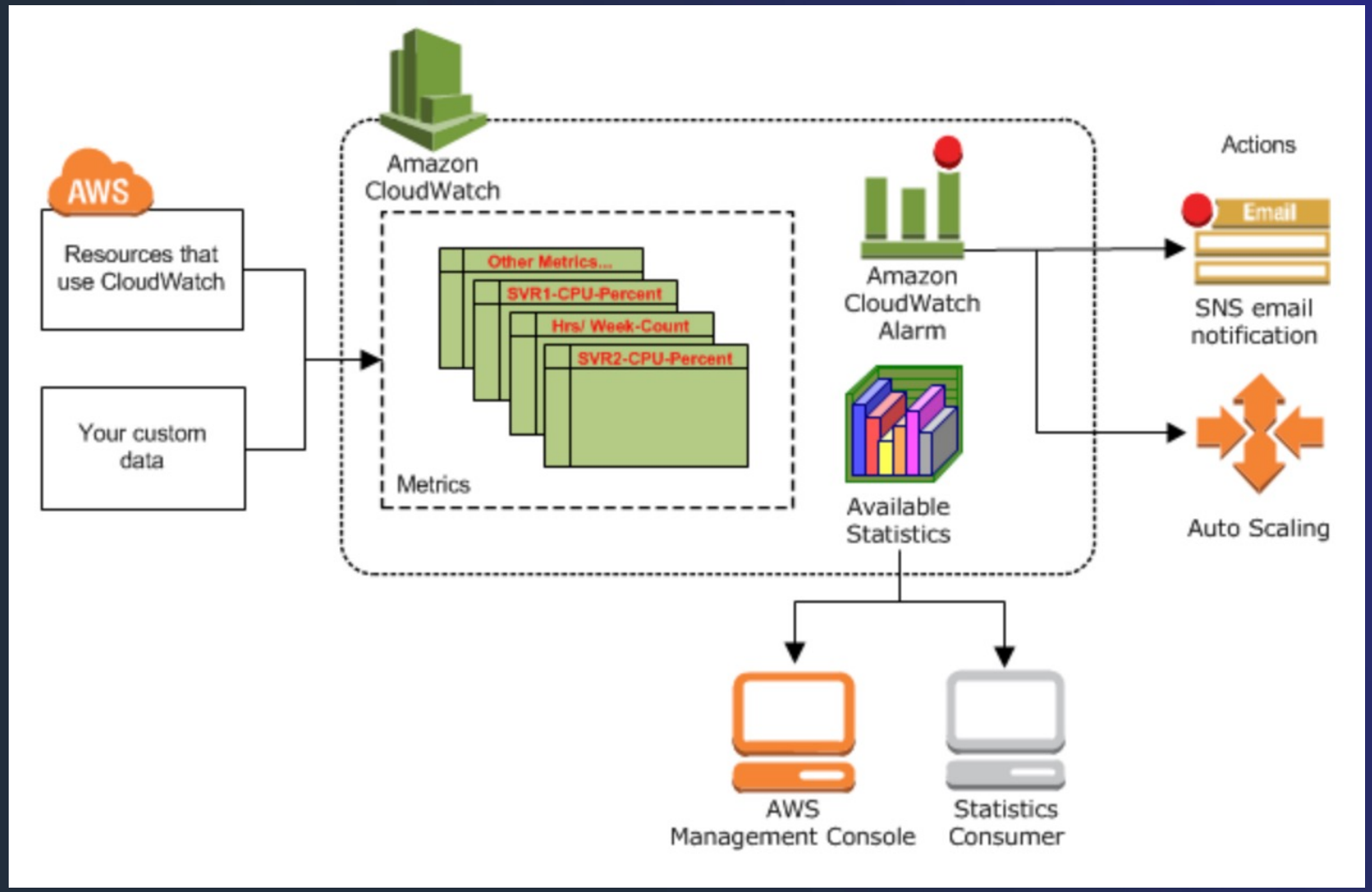
- Dashboards
- Logs
- Insights
- Service Lens
- Metrics
- Alarms
- Anomaly Detection
- Synthetics



AWS X-Ray

- Traces
- Analytics
- Service map

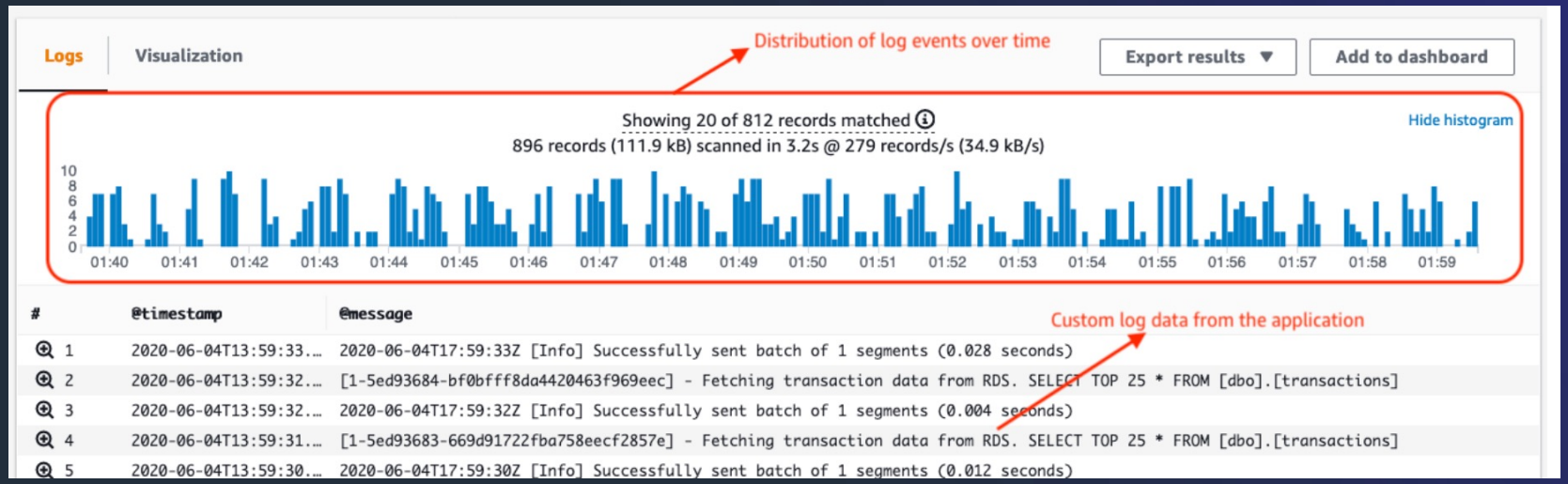
How does Amazon CloudWatch work?



Logs insights

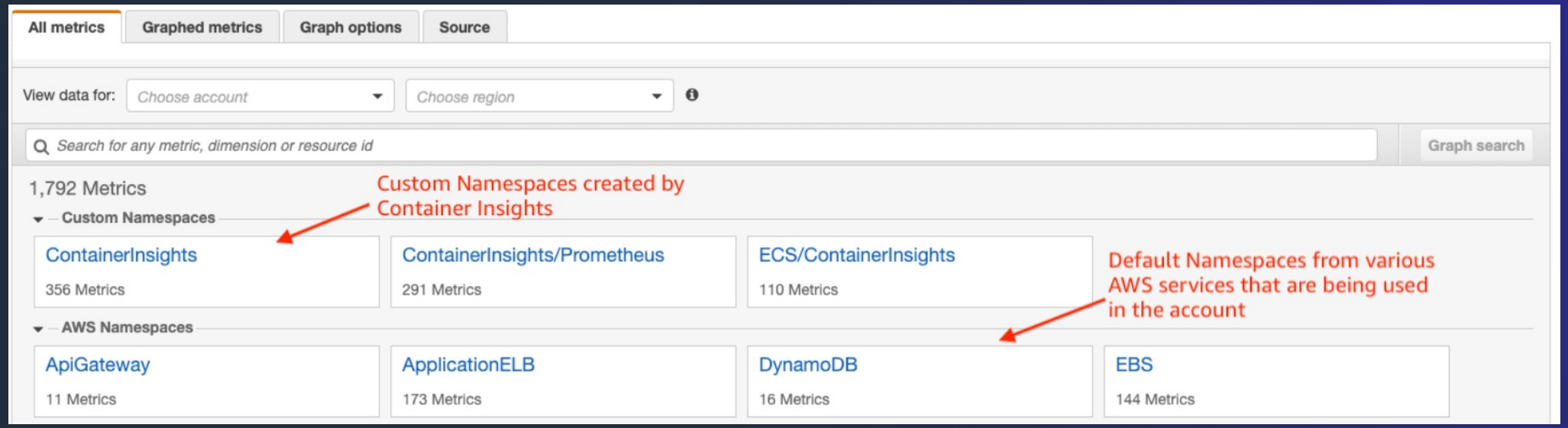
- Amazon CloudWatch Logs Insights enables you to interactively search and analyse your log data in Amazon CloudWatch Logs. You can perform queries to help you more efficiently and effectively respond to operational issues

```
fields @message
    | parse @message "[*] *" as loggingType, loggingMessage
    | filter loggingType = "ERROR"
    | display loggingMessage
```



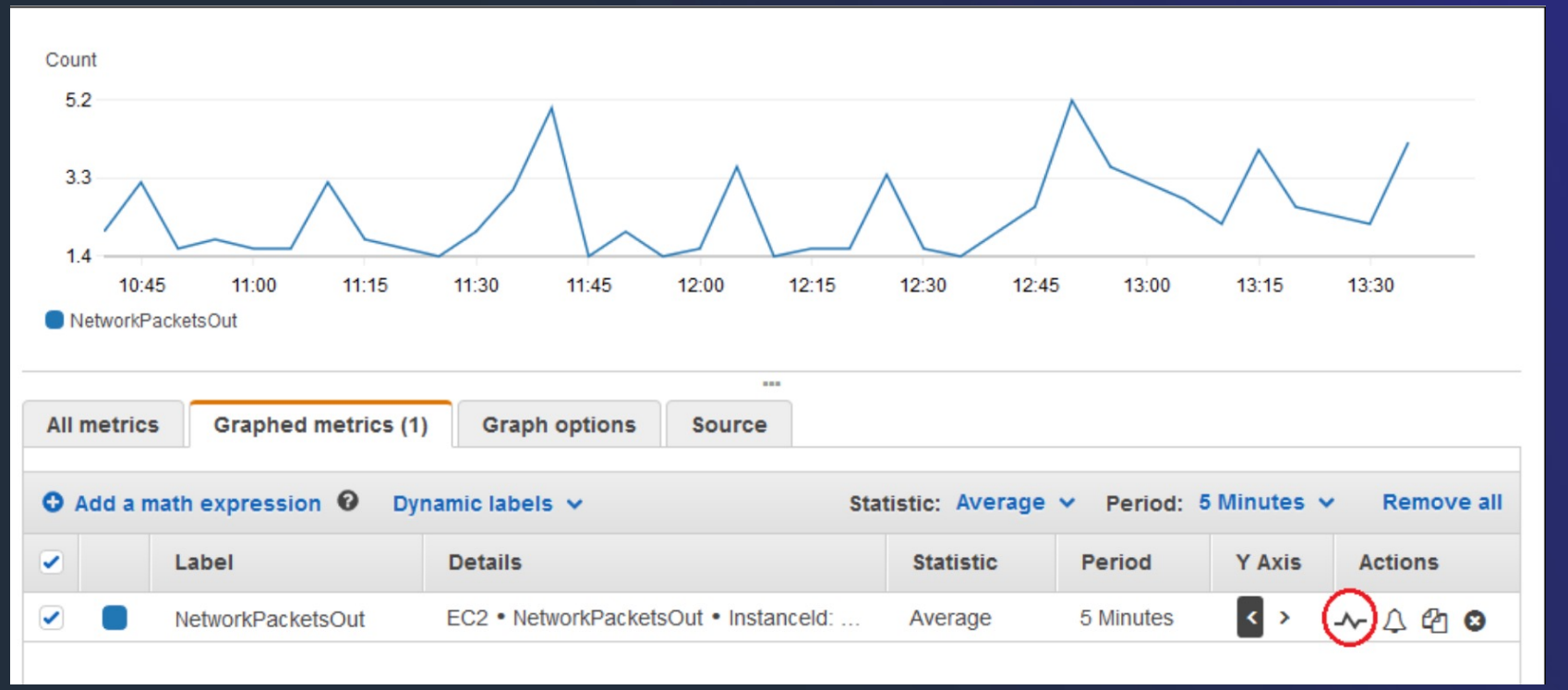
Metrics

- Metrics are grouped first by namespace, and then by the various dimension combinations within each namespace. For example, you can view all EC2 metrics, EC2 metrics grouped by instance, or EC2 metrics grouped by Auto Scaling group. Only the AWS services that you're using send metrics to Amazon CloudWatch.



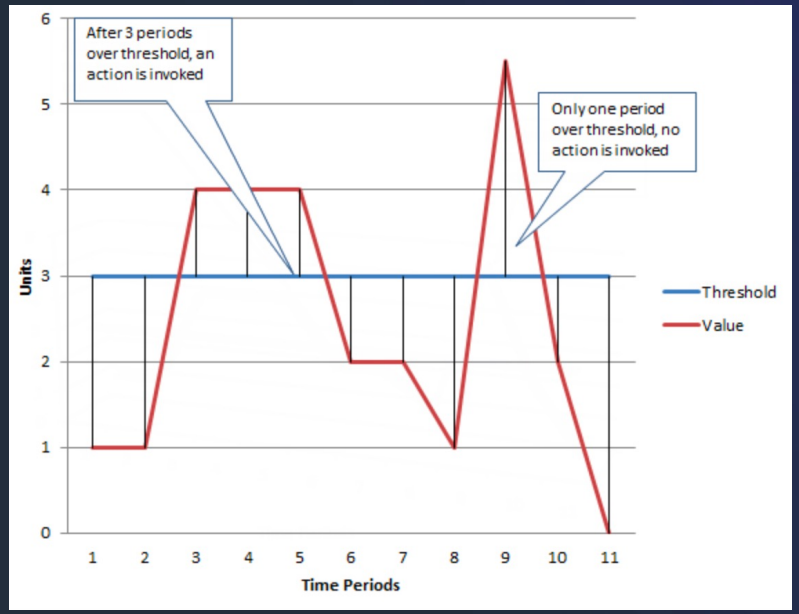
Graphed metrics

- CloudWatch supports the following statistics on metrics: Average, Minimum, Maximum, Sum, and SampleCount. To add an anomaly detection band that shows expected values for the metric, choose the anomaly detection icon under **Actions** next to the metric

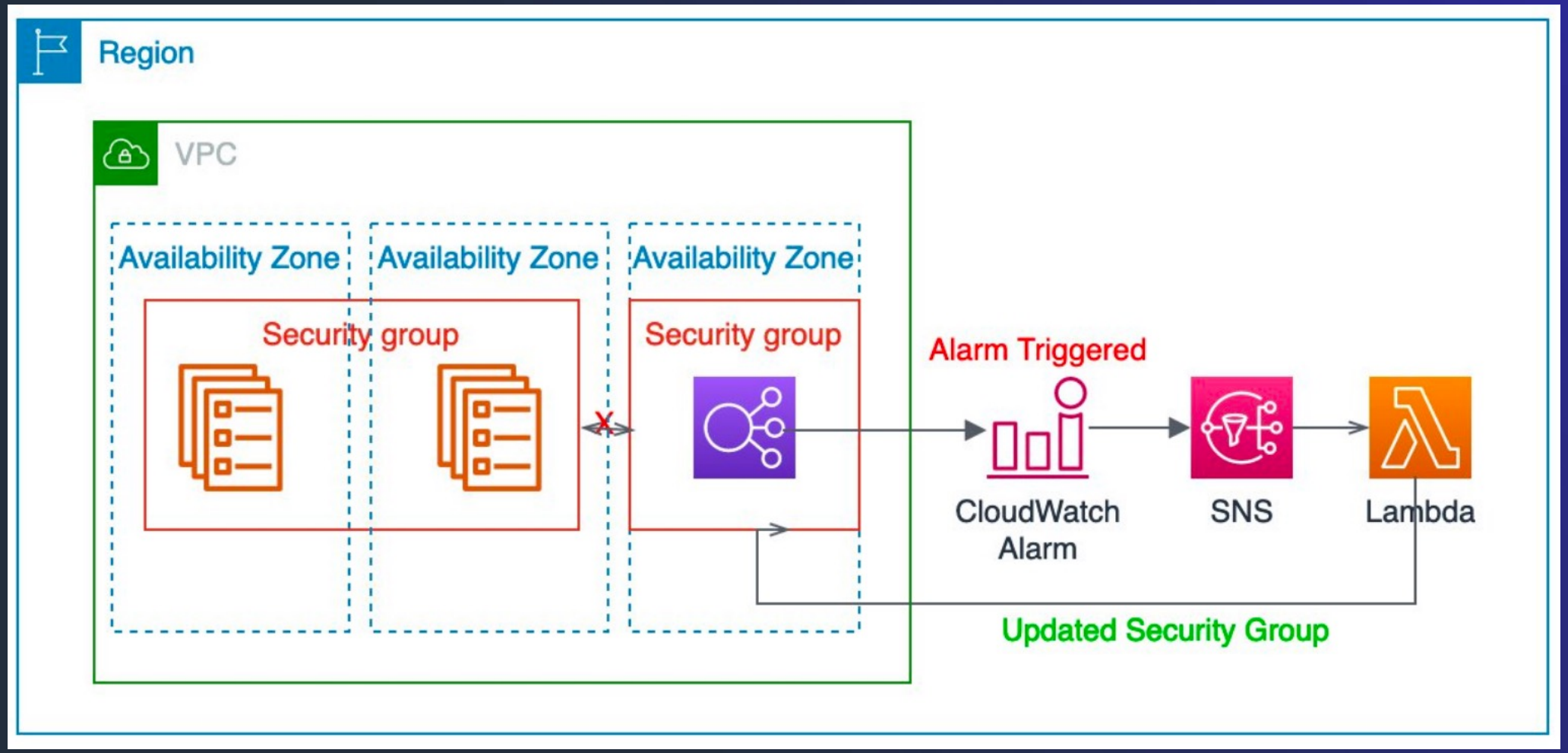


Alarms

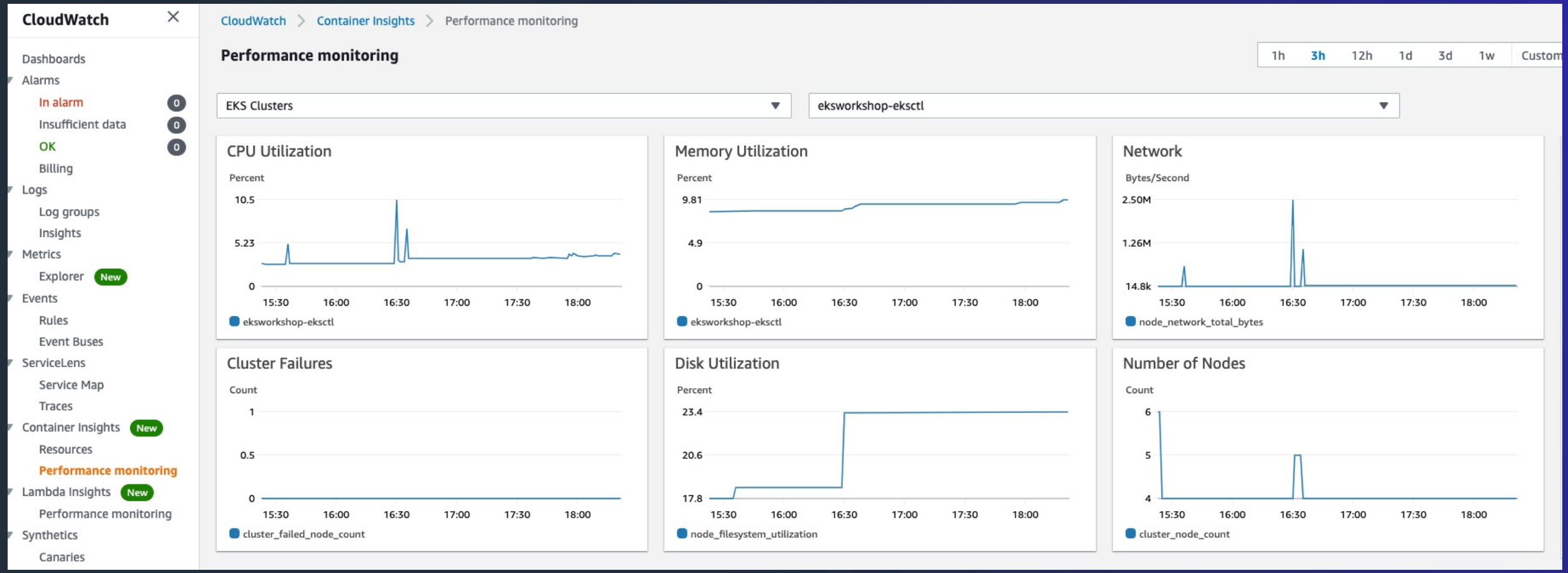
- You can create both *metric alarms* and *composite alarms* in Amazon CloudWatch. A *metric alarm* watches a single Amazon CloudWatch metric or the result of a math expression based on Amazon CloudWatch metrics. A *composite alarm* includes a rule expression that takes into account the alarm states of other alarms that you have created



Automated remediation using AWS Lambda

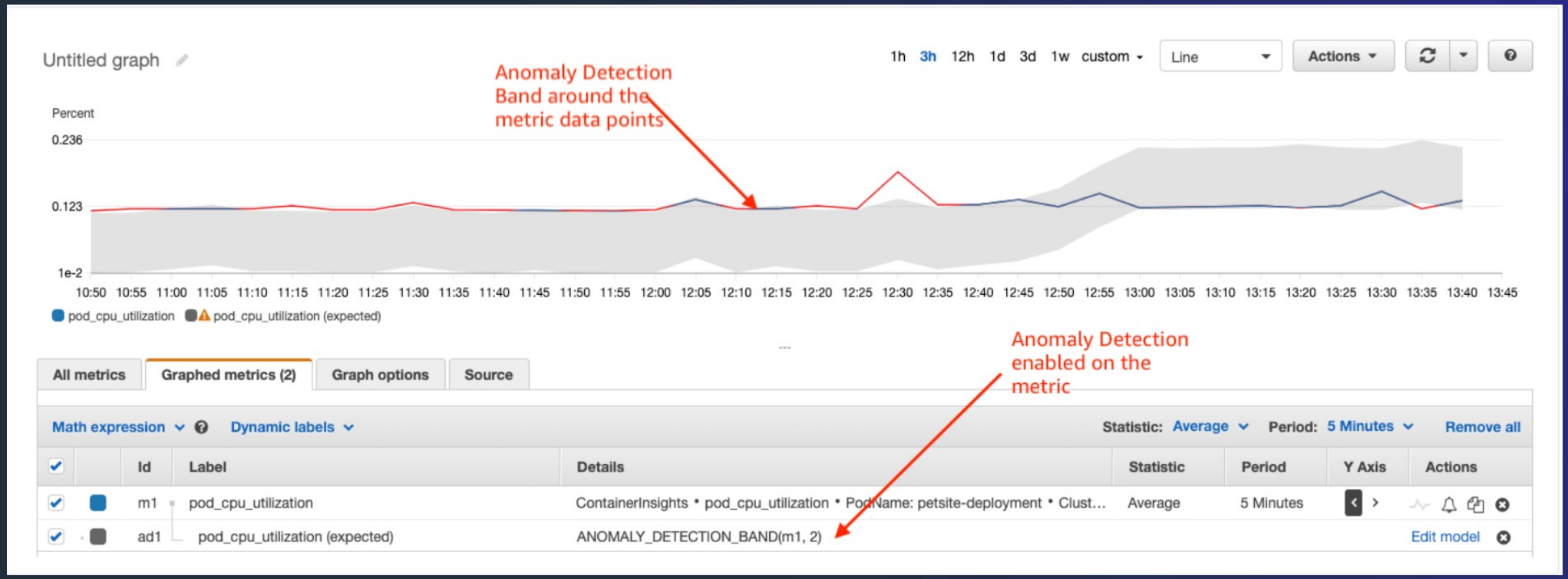


Container insights

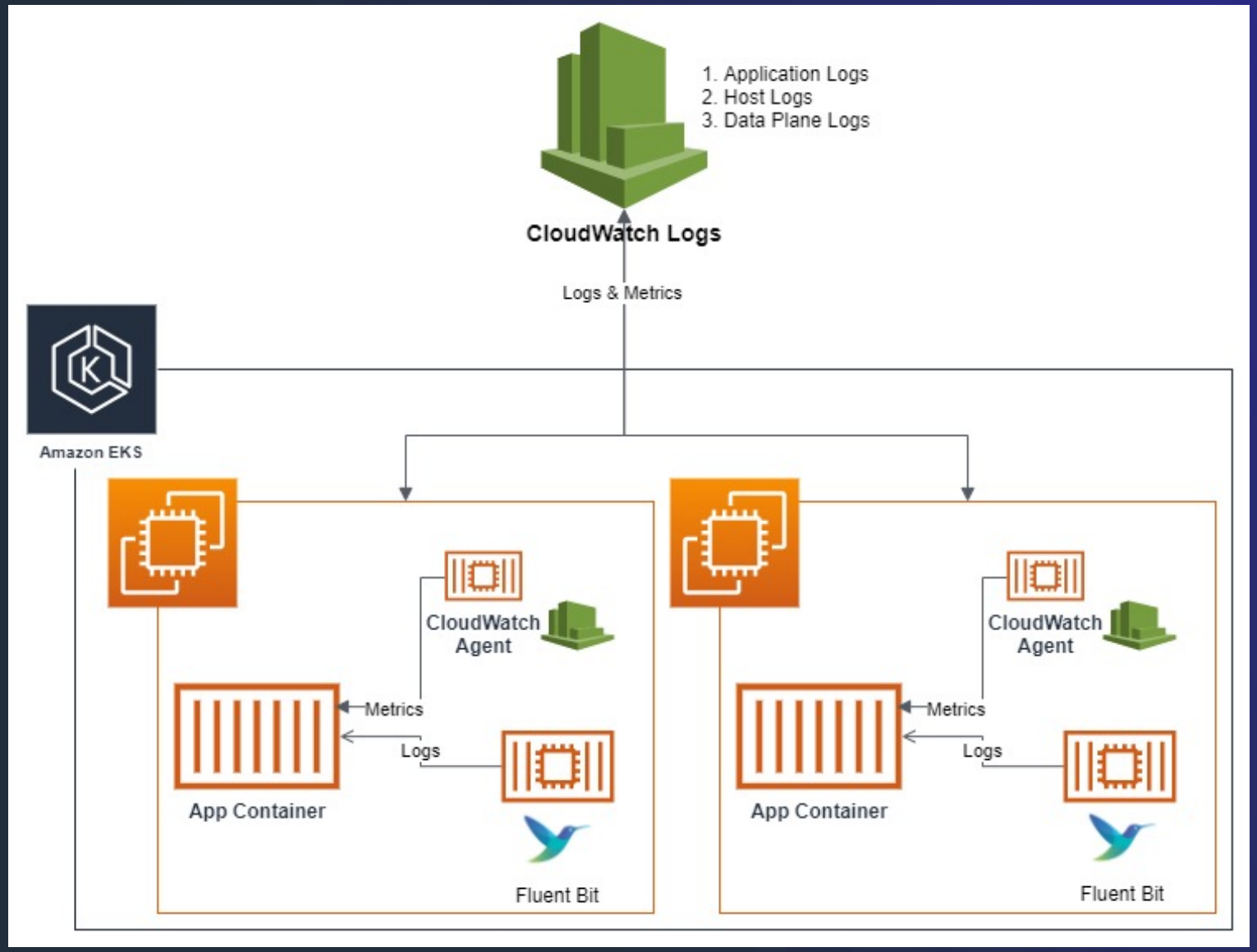


Anomaly detection

- When you enable *anomaly detection* for a metric, Amazon CloudWatch applies statistical and machine learning algorithms. These algorithms continuously analyse metrics of systems and applications, determine normal baselines, and surface anomalies with minimal user intervention.

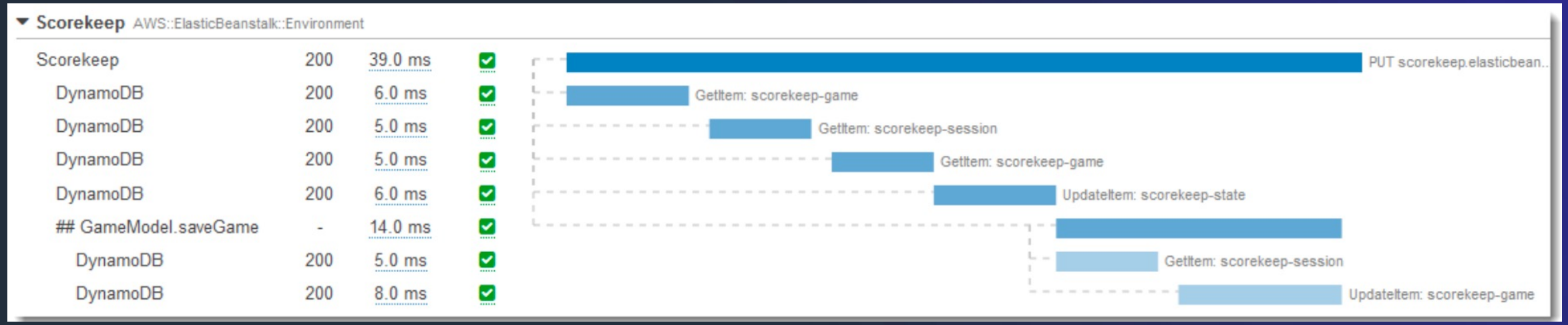


Fluent Bit integration with Amazon CloudWatch

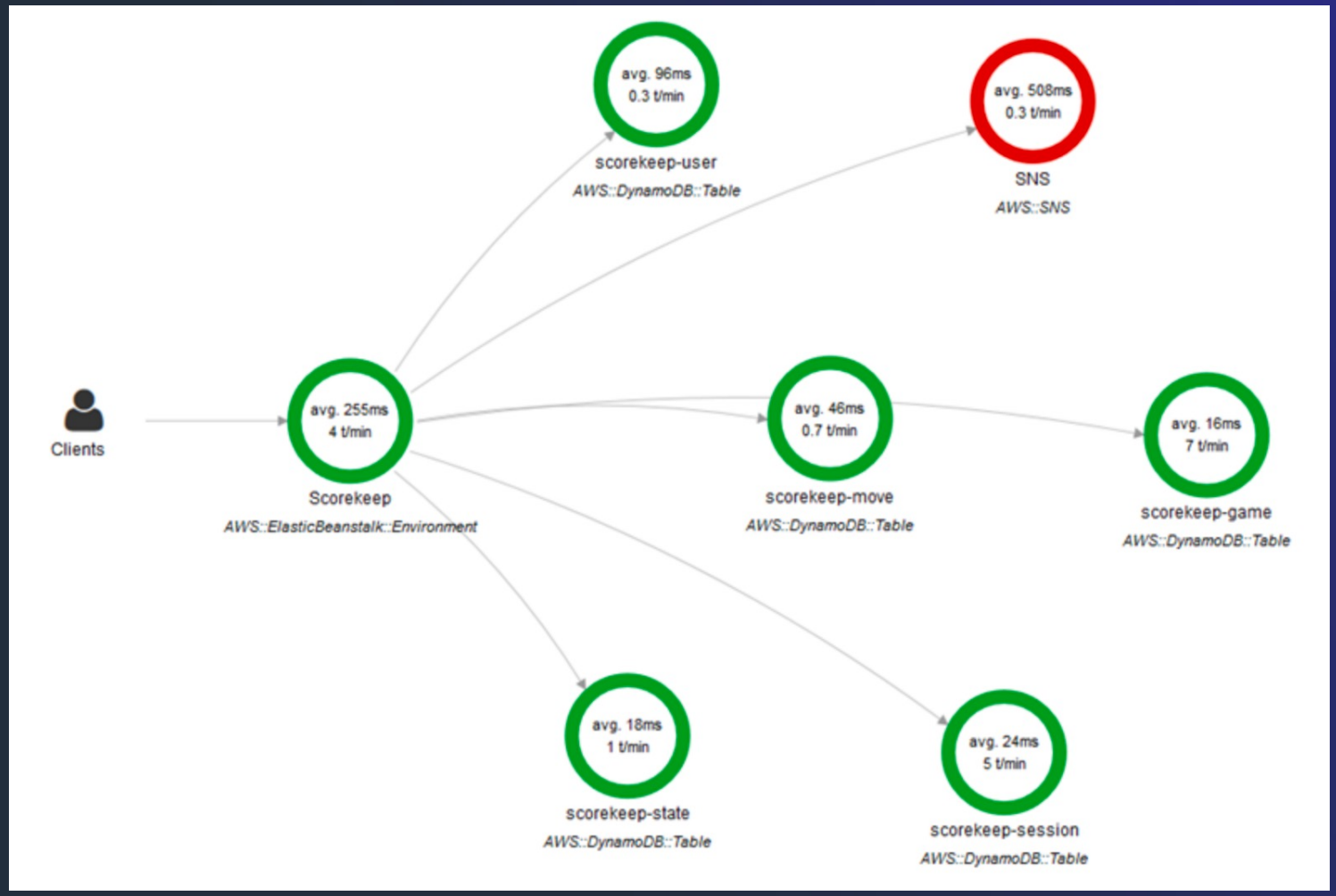


AWS X-Ray

- The central idea behind AWS X-Ray is the use of correlation IDs, which are unique identifiers attached to all requests and messages related to a specific event chain. The trace ID is added to HTTP requests in specific tracing headers named X-Amzn-Trace-Id when the request hits the first X-Ray-integrated service (for example, AWS Application Load Balancer or Amazon API Gateway) and included in the response. Via the AWS X-Ray SDK, any microservice can read but can also add or update this header.



AWS X-Ray service map



Importance of SLIs, SLOs and SLAs

Definitions

- Service Level Indicator (SLI) is a carefully defined quantitative measure of some aspect of the level of service that is provided.
- Service Level Objective (SLO) specifies a target level for the reliability of your service. Because SLOs are key to making data-driven decisions about reliability, they're at the core of SRE practices.
- Service Level Agreement (SLA) is an explicit or implicit contract with your users that includes impact of meeting (or missing) the SLOs they contain. The impacts are most easily recognized when they are financial—a rebate or a penalty—but they can take other forms

Guidance for SRE and product development

- You must not use every metric you can track in your monitoring system as an SLI. An understanding of what your users want from the system will inform the judicious selection of a few indicators
- Have as few SLOs as possible and get all stakeholders to agree to it. For clarity, SLOs should specify how they're measured and the conditions under which they're valid
- Instrument and create dashboards to represent SLIs
- Define a quarterly error budget based on the service's service level objective. The error budget provides a clear, objective metric that determines how unreliable the service is allowed to be within a single quarter. The main benefit of an error budget is that it provides a common incentive that allows both product development and SRE to focus on finding the right balance between innovation and reliability

Summary

What did we learn?

- Correlate logs, metrics and traces for deeper insights
- Implement SLIs, SLOs and SLAs to understand which behaviours really matter for the service, how to measure and evaluate those behaviours
- Leverage AWS services for monitoring, logging, alarming, and dashboards with Amazon CloudWatch and tracing through AWS X-Ray
- Visit https://observability.workshop.aws/ to get a hands-on experience on all AWS observability features

Thank you!