Cloud Native Architecture Nanodegree Glossary

Course 4: Observability

Black box monitoring

Historically, we have used black box monitoring, which involves getting indirect data about the system (such as disk errors, hypervisor resource alerts, or hardware uptime) from the outside and then using this to attempt to infer the cause of a problem (without being able to view the source of the problem directly).

Distributed tracing

Distributed tracing is tracing for microservices. With distributed tracing, we are able follow the execution of a request as it goes through all of the relevant services.

ELK

ELK stands for Elasticsearch, Logstash, Kibana. This stack serves as a popular observability alternative to the one we use in this course (Prometheus, Jaeger, and Grafana).

Error budget

A buffer in an SLO that allows for a small amount of error.

Errors

The number of requests that are failing (such as number of HTTP 500 responses). One of the Four Golden Signals.

Four Golden Signals

Four core properties that are used to define service or performance: Latency, Traffic, Saturation,

Grafana

A visualization platform that allows you to build open source dashboards. Supports time-series databases as a backend and for that reason is often bundled with Prometheus.

Jaeger

An open source distributed tracing system.

Latency

The time taken to serve a request (usually measured in ms). One of the Four Golden Signals.

Microservices

In contrast to monolithic applications, with microservices, all of the functions are separated into standalone services that work asynchronously and are distributed in nature. If one service has issues, this does not necessarily impact the rest of the application. Also, due to their small size and portability, microservices are great for Cloud Native development.

Monolithic application

Monolithic applications have all the services bundled together. The user interface, business logic and data layer are all stacked on top of each other, such that making changes to one service can cause all the others to have issues as well.

Observability

Observability allows us to look inside an application and even trace through the execution of the application to gather detailed performance data on its individual components.

OpenTracing

The data model used to structure spans and traces. Note that OpenTelemetry is the future of OpenTracing. It was created when OpenCensus merged with OpenTracing.

Panel

Panels are essentially containers for charts and graphs within our Grafana dashboards.

Prometheus

An open source monitoring and altering application that records system information in a time-series database.

PromQL

The querying language used by Prometheus.

Saturation

The overall capacity of a service (such as the percentage of memory or CPU used). One of the Four Golden Signals.

Service level

The degree (i.e. level) of performance that an application (or service) provides to the user.

Service-Level Agreement (SLA)

A legally binding contract that guarantees a service level to the customer. If the SLA is not met, typically the company will provide some form of compensation to the customer.

Service-Level Indicator (SLI)

A measurement that indicates the level of performance a service is achieving.

Service-Level Objective (SLO)

A measurable goal set by the SRE team to ensure a standard level of performance during a specified period of time.

Spans

The units of work that make up a trace; spans are objects that represent a single part of the process being traced.

Trace

A mock request that goes through services and records its performance as it executes.

Traffic

The amount of stress on a system from demand (such as the number of HTTP requests/second). One of the Four Golden Signals.

White box monitoring

In white box monitoring, we look inside and get direct data on the application (such as user utilization, HTTP status errors, and SQL queries); we can directly observe the root cause of the issue.