

Módulo OpenTelemetry - Prueba Final

Objetivos

Evaluar la capacidad de aplicar los conocimientos adquiridos sobre la configuración necesaria para realizar operaciones de monitorización y observabilidad de métricas, trazas y logs.

Evaluar la capacidad de desenvolverse de manera autónoma realizando las tareas de configuración con el sistema.

Evaluar la capacidad de incorporar código básico para ampliar las capacidades de monitorización.

Descripción

Descargar como código de aplicación Python desde la siguiente dirección de repositorio GitHub

<https://github.com/SigNoz/sample-flask-app>

Práctica 1

Crear la configuración necesaria para la ejecución de la aplicación con instrumentación automática (zero config) por medio de OpenTelemetry Collector, Prometheus, Jaeger, Loki, Grafana definiendo toda la infraestructura por medio de contenedores Docker en un fichero docker-compose.yaml

(Tened en cuenta que la aplicación necesita MongoDB para funcionar por lo que debería estar también como contenedor en este fichero con la configuración:

```
nombre: mongodb
imagen: mongo:latest
puertos: 27017:27017)
```

Práctica 2

Comprobad que llegan las señales de métricas, trazas y logs a Grafana, una vez configurados como Data Sources: Prometheus, Jaeger, Loki.

Realizad un captura de pantalla con datos de cada origen para demostrarlo e incorporar la captura en una carpeta del proyecto (ej: capturas)

Práctica 3

Incorporad código en la función action

```
@app.route("/action", methods=['POST'])  
def action ():
```

De tal manera que se generen mensajes de log de nivel informativo para comprobar la ejecución, una traza de ejecución donde se incorporen como atributos los parámetros que llegan como: name, desc y una métrica de tipo contador que se llame total_altas_counter que acumule el número de veces que se invoca esta función.

Modificad la configuración necesaria y volved a ejecutar la aplicación para comprobar que llegan las nuevas señales de los logs, trazas y métricas.

Comprobad que llegan las señales de métricas, trazas y logs a Grafana, una vez configurados como Data Sources: Prometheus, Jaeger, Loki.

Realizad un captura de pantalla con datos de cada origen para demostrarlo e incorporar la captura en una carpeta del proyecto (ej: capturas)