

Ejercicio: Crea un entorno de observabilidad completo con Docker

Objetivo

Crear desde cero un entorno Docker que permita observar una aplicación frontend (como Angular) mediante trazas, logs y métricas. Este stack incluirá **Grafana, Prometheus, Loki, Tempo, OpenTelemetry Collector**, y opcionalmente **Elasticsearch + Kibana**.

Requisitos

- Docker y Docker Compose instalados.
- Conocimientos básicos de observabilidad y contenedores.
- Familiaridad con herramientas como Grafana, Prometheus, etc.

Actividades

1. Crea un nuevo proyecto local con la siguiente estructura mínima:

```
observabilidad-stack/  
├── docker-compose.yml  
├── otel-collector-config.yml  
├── prometheus.yml  
└── grafana/  
    ├── provisioning/  
    │   ├── datasources.yml  
    │   └── dashboards.yml
```

2. Implementa un `docker-compose.yml` que levante los siguientes servicios:

Servicio	Puerto	Descripción
Grafana	3000	Visualización de métricas, logs y trazas (con datasources preconfigurados).
Prometheus	9090	Recolector de métricas desde OTEL Collector.
Loki	3100	Almacenador de logs estructurados.
Tempo	3200	Almacenador de trazas distribuidas.
OTEL Collector	4318	Punto de ingesta OTLP (HTTP) para logs, métricas y trazas.
Elasticsearch	9200	(Opcional) Almacenamiento de logs estructurados.
Kibana	5601	(Opcional) Visualización de logs vía Elasticsearch.

3. Crea y configura los siguientes archivos de configuración:

- `otel-collector-config.yml` con los pipelines para métricas, logs y trazas.
- `prometheus.yml` para que scrapee al OTEL Collector.
- `grafana/provisioning/datasources.yml` y `dashboards.yml` para tener dashboards precargados al iniciar Grafana.

4. Verifica que el stack funcione accediendo a los servicios desde el navegador:

- Grafana: <http://localhost:3000>
- Prometheus: <http://localhost:9090>
- Loki: <http://localhost:3100>
- Tempo: <http://localhost:3200>
- Elasticsearch: <http://localhost:9200> (opcional)
- Kibana: <http://localhost:5601> (opcional)

5. Integra una app Angular (puede ser una app de prueba mínima) que:

- Envíe trazas al endpoint OTLP del collector: `http://localhost:4318/v1/traces`
-

📁 Entregables

- Carpeta del proyecto con los archivos Docker y de configuración listos para ejecutarse.
 - `README.md` explicando cómo levantar el entorno y con el diagrama Mermaid incluido.
 - Una app Angular o simulador que envíe trazas (y opcionalmente logs/métricas).
 - Capturas de pantalla que muestren datos visibles en Grafana.
-

✔ Criterios de evaluación

- ✔ El entorno Docker funciona correctamente y los servicios están disponibles.
 - ✔ El flujo de datos (trazas, logs, métricas) está correctamente conectado.
 - ✔ El README contiene instrucciones claras y un diagrama explicativo.
 - ✔ La app frontend envía correctamente trazas al OTEL Collector.
 - ✔ (Opcional) Integración de Signoz.
-