

Curso: Introducción a DevOps

Juan Camilo Escobar Naranjo

Administrador de Sistemas Informáticos

Estudiante de Maestría en Administración de Sistemas
Informáticos

TPM – DevOps Services – Banco de Bogotá

jcescobarn@unal.edu.co

Automatización con Pipelines

Taller

el objetivo de este taller es que en grupos de máximo 2 estudiantes construyan una pequeña aplicación que exponga métricas en formato Prometheus, despliegue Prometheus y Grafana usando Docker Compose, y visualice sus métricas en un dashboard. Además, automatizará el despliegue completo usando GitHub Actions.

Paso 1

Crear un directorio app/ y dentro colocar un archivo app.py con el siguiente código mínimo:

```
from flask import Flask
from prometheus_client import Counter, generate_latest

app = Flask(__name__)
visits = Counter('webapp_visits_total', 'Total visits to the homepage')

@app.route("/")
def hello():
    visits.inc()
    return "Hello, Prometheus!"

@app.route("/metrics")
def metrics():
    return generate_latest()

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=8000)
```

Paso 2

Crear un Dockerfile que ejecute la aplicación creada anteriormente

Paso 3

Crear el prometheus.yml

```
global:
  scrape_interval: 5s

scrape_configs:
  - job_name: 'flask-app'
    static_configs:
      - targets: ['app:8000']
```

Paso 4

Crear un docker-compose.yaml que permita levantar un entorno completo de observabilidad compuesto por:

- Una aplicación en Python que expone métricas personalizadas en el endpoint /metrics.
- Un servicio de Prometheus que recolecte dichas métricas cada 5 segundos.
- Un servicio de Grafana que permita visualizar las métricas expuestas, accediendo desde un navegador.

Paso 5

Configurar un flujo de trabajo de GitHub Actions que:

- Compile una imagen Docker de la aplicación ubicada en app/.
- La suba a Github Packages con la etiqueta latest.

Paso 6

Crea un pipeline que:

- Se active por medio de un workflow dispatch
- Se conecte por SSH a una instancia EC2.
- Realice git pull del proyecto.
- Ejecute docker compose up -d app para levantar solo la aplicación.

Paso 7

Implementar un tercer pipeline que:

- Se active al hacer cambios en el directorio monitoring/.
- Se conecte por SSH a la instancia EC2.
- Haga pull del código actualizado.
- Despliegue los servicios de monitoreo (prometheus y grafana) vía docker compose.