

Documentación del Jenkins Pipeline

Descripción General

Este pipeline orquesta la construcción, prueba, empaquetado y despliegue de múltiples microservicios para una aplicación eCommerce.

Incluye pruebas unitarias, generación de cobertura, pruebas de carga, escaneo de seguridad (OWASP ZAP), despliegue en contenedores y despliegue final en Kubernetes.

Herramientas utilizadas

- Jenkins Declarative Pipeline
- Maven para construcción de artefactos y pruebas.
- Docker para crear y ejecutar contenedores.
- Docker Hub como repositorio de imágenes.
- Locust para pruebas de carga y estrés.
- OWASP ZAP para escaneo de seguridad.
- Kubernetes para despliegue en producción.
- Zipkin, Prometheus y Grafana para observabilidad.
- JUnit y Jacoco para pruebas y cobertura.

Variables de Entorno

- DOCKERHUB_USER: usuario de Docker Hub.
- DOCKER_CREDENTIALS_ID: ID de las credenciales en Jenkins.
- KUBECONFIG: ruta del archivo de configuración de Kubernetes.
- SERVICES: lista de microservicios usados en el pipeline.

Etapas del Pipeline

1. Compiling Services

Se ejecuta en ramas dev, stage y master. Ejecuta mvn clean package -DskipTests para compilar sin pruebas.

2. Unit Tests & Coverage

Solo en rama dev. Ejecuta pruebas unitarias y genera cobertura (Jacoco) para user-service y product-service.

3. Build Docker Images

Construye imágenes Docker para todos los servicios en SERVICES.

4. Push Docker Images

Autentica y sube imágenes a Docker Hub con credenciales de Jenkins.

5. Levantar contenedores para pruebas

Documentación del Jenkins Pipeline

Crea red Docker y levanta servicios auxiliares + microservicios para pruebas funcionales.

6. Load Tests con Locust

Pruebas de carga en order-service y payment-service con 10 usuarios.

7. Stress Tests con Locust

Pruebas de estrés en los mismos servicios con 50 usuarios.

8. OWASP ZAP Scan

Ejecuta escaneos de seguridad sobre 6 servicios y publica reportes en HTML.

9. Stop and Remove Containers

Limpieza: elimina todos los contenedores y la red Docker de pruebas.

10. Waiting approval

Pide aprobación manual para desplegar en producción (solo en master).

11. Deploy Core Services

Despliega zipkin, service-discovery y cloud-config en Kubernetes.

12. Deploy Microservices

Despliega microservicios de aplicación principal en Kubernetes.

13. Mostrar URLs de Monitorización

Imprime en consola las URLs locales de Prometheus y Grafana.

Post-Ejecucion

- En caso de éxito: imprime mensaje en consola.
- En caso de fallo: envía correo a camilobueno05@gmail.com usando emailtext.