



# Obligatorio [Curso Devops]

---

Repositorio de Devops

CICD GitHub Actions

Terraform

Helm

Cucumber

[Resumen](#) •

[CI](#) •

[CD](#) •

[Infraestructura](#) •

[QA Automation](#) •

## Resumen

---

Nuestro entorno cuenta con cuatro microservicios con tres ambientes separados por rama para cada microservicio (develop, test, production) y un repositorio donde se encuentra la infraestructura como código correspondiente a la entrega.

Para lograr el CI/CD utilizamos principalmente la herramienta de Github Actions.

En una primer instancia cuando un microservicio recibe cambios en una rama de las mencionadas anteriormente comienza el proceso de liberación para actualizar la versión en el ambiente correspondiente. Una vez liberada la versión, el mismo proceso actualiza la misma para el ambiente correspondiente en el repositorio donde se encuentra el código de CD. Esto inicia el proceso de despliegue utilizando Terraform y Helm sobre EKS.

Todos los repositorios se encuentran públicos para la revisión de la entrega.

## Repositorios

- Orders Service: <https://github.com/ignaciolopez409/orders-service>
- Products Service: <https://github.com/ignaciolopez409/products-service>
- Payment Service: <https://github.com/ignaciolopez409/payments-service>
- Shipping Service: <https://github.com/ignaciolopez409/shipping-service>
- Infra Obligatorio DevOPS: <https://github.com/ignaciolopez409/infra-obligatorio-devops>

## CI

---

Para el ciclo de integración se corren tests unitarios y se hace un análisis del código con la plataforma de sonarcube.io. Por cuestiones de la versión Community Edition solamente podemos correr análisis sobre la rama por defecto, dado esto por razones de practicidad esta rama es develop.

Para empaquetar y liberar los microservicios utilizamos Maven. Nos apoyamos principalmente en el plugin de release e implementamos el plugin Jib que se encarga de crear imágenes docker basados en estándares teniendo en cuenta el lifecycle de Maven, autentica contra el registry DockerHub y pushea la imagen con su respectiva versión.

Una vez terminado este proceso, actualizamos sobre el repositorio de Devops el valor de la versión en su respectivo ambiente y pusheamos un commit indicando el ambiente y de esta forma se encadena el proceso de CD

## Análisis de Código

---

En el proceso de CI se ejecuta el analisis de código utilizando la herramienta SonarCloud.

## Informe en base a análisis y recomendaciones

---

### Orders Service

<https://sonarcloud.io/project/overview?id=orders-service-obligatorio>

- Resolver Bug reportado
- Resolver Vulnerabilidades
- Resolver duplicado de código
- Cubrir el código realizando las pruebas unitarias correspondientes (de momento está en 0.0%)

### Payments Service

<https://sonarcloud.io/project/overview?id=payments-service-obligatorio>

- Resolver Bug reportado
- Resolver Security hotspot reportado
- Cubrir el código realizando las pruebas unitarias correspondientes (de momento está en 0.0%)

## Products Service

<https://sonarcloud.io/project/overview?id=products-service-obligatorio>

- Cubrir el código realizando las pruebas unitarias correspondientes (de momento está en 0.0%)

## Shipping Service

<https://sonarcloud.io/project/overview?id=shipping-service-obligatorio>

- Cubrir el código realizando las pruebas unitarias correspondientes (de momento está en 0.0%)

## CD

---

Una vez que el repositorio recibe los cambios para un ambiente determinado comienza el proceso de CD utilizando Infraestructura como código.

El mismo cuenta de tres etapas:

### Deploy de Infraestructura

Utilizando manifest de Terraform se crean todos los componentes necesarios para crear el cluster de Kubernetes en EKS y el cluster propiamente dicho.

### Deploy de aplicaciones

Se creó un template de Helm el cual despliega los microservicios y es ejecutado por otro manifest de Terraform.

## QA Automation

Una vez desplegada la infraestructura y las aplicaciones se ejecutan pruebas API Testing utilizando Karate Framework sobre Cucumber. El reporte se envía a los stakeholders deseados via email.

## Nota

---

En la defensa se realizará una demo para despejar cualquier duda al respecto.