

# Infraestructura como código

Gerson Zuñiga

Akira García

Joel Seminario



# Parte A{

## Infraestructura como código

Investigar una herramienta IaC y describe cómo organiza sus módulos

Los módulos en Terraform son contenedores de múltiples recursos almacenados y usados como un grupo.

Pueden ser de dos tipos:

- El módulo raíz consiste de los recursos definidos en los archivos .tf en el directorio principal.
- Los módulos hijo se entienden como los módulos llamados por otros módulos, usualmente el módulo raíz



Estructura de archivos y directorios para “network”, “database” y “applications”

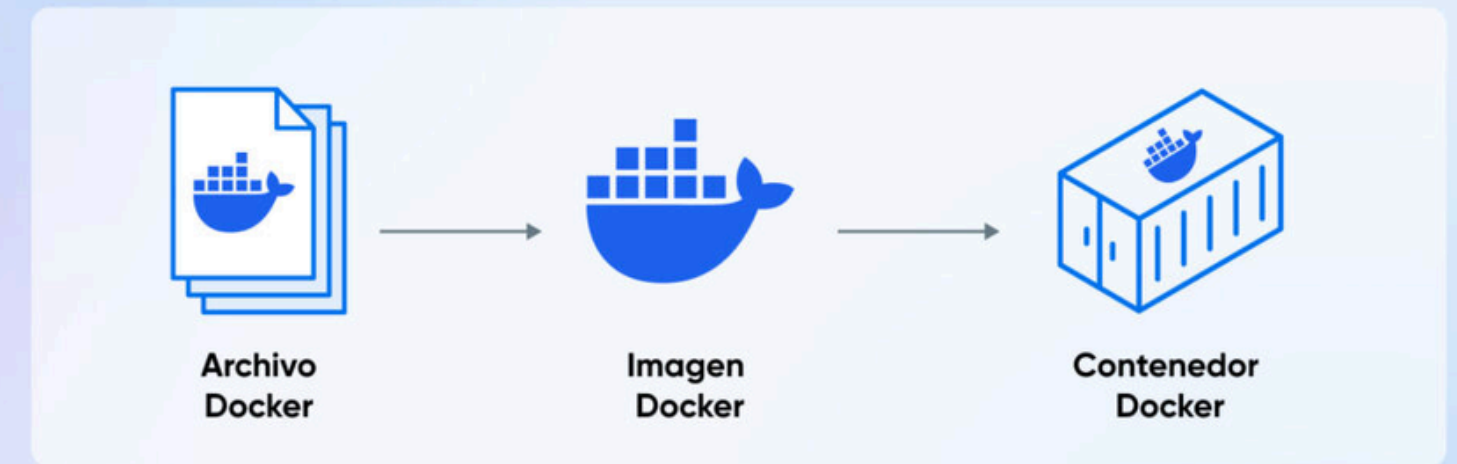
```
$ tree complete-module/
.
├── README.md
├── main.tf
├── variables.tf
├── outputs.tf
├── ...
└── modules/
    ├── network/
    │   ├── README.md
    │   ├── variables.tf
    │   ├── main.tf
    │   └── outputs.tf
    ├── database/
    │   ├── README.md
    │   ├── variables.tf
    │   ├── main.tf
    │   └── outputs.tf
    └── applications/
        ├── README.md
        ├── variables.tf
        ├── main.tf
        └── outputs.tf
```

}

## Parte B{

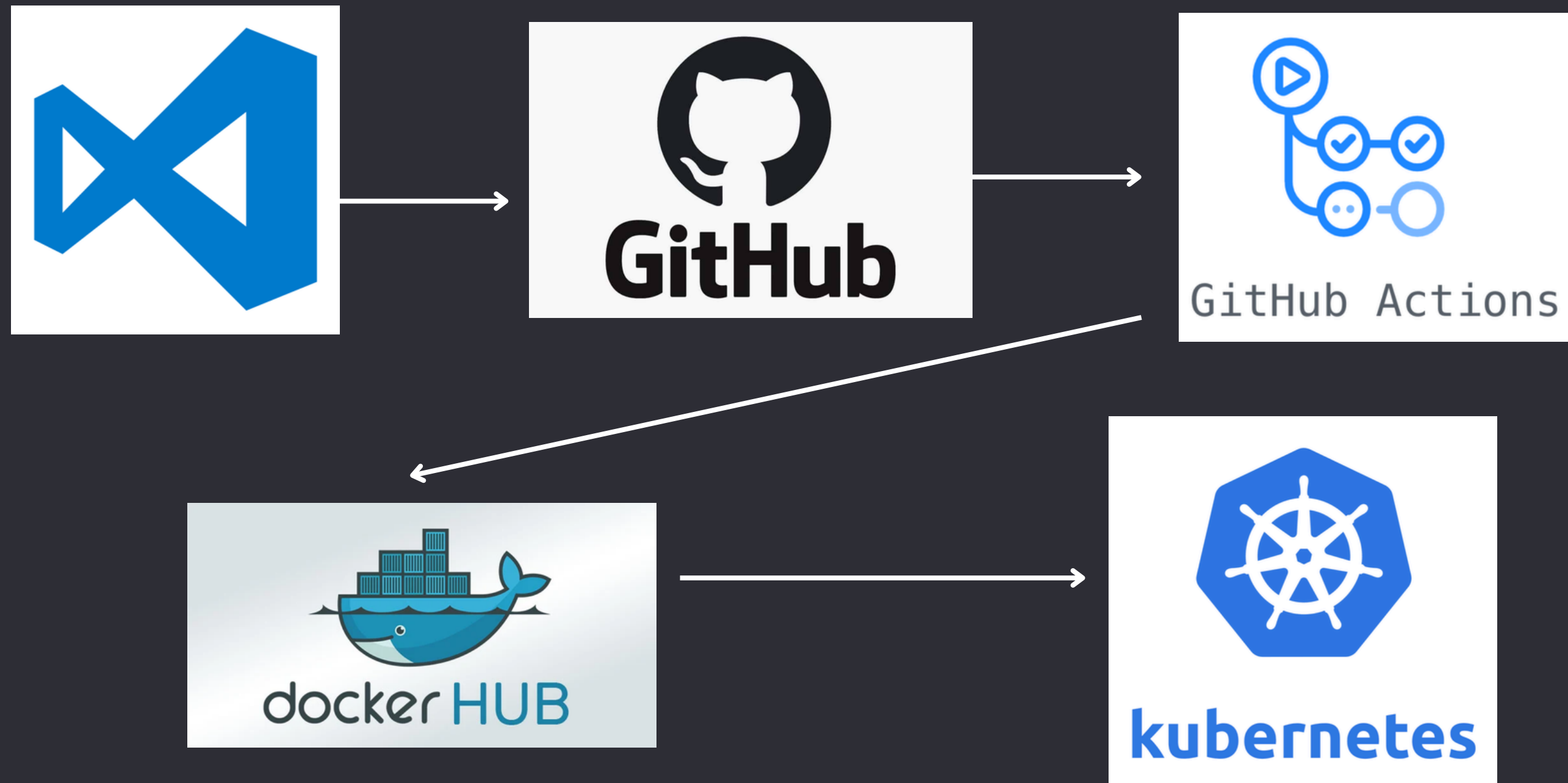
Contenerización y despliegue  
de aplicaciones modernas

### Componentes Clave de Docker



}

Describir un flujo simple de despliegue {



}

# Ventajas de Kubernetes para Escala {

## Escalado Automático

```
kubectl autoscale deployment mi-aplicacion --cpu-percent=50 --min=2 --max=10
```

## Balanceo de Carga

Service con tipo LoadBalancer.

## Alta Disponibilidad

Si un nodo o pod falla, Kubernetes detecta el problema y crea nuevos

## Despliegues y Rollbacks Seguros

en caso de falla, se puede revertir a la versión anterior fácilmente con  
`kubectl rollout undo deployment mi-aplicacion.`

}

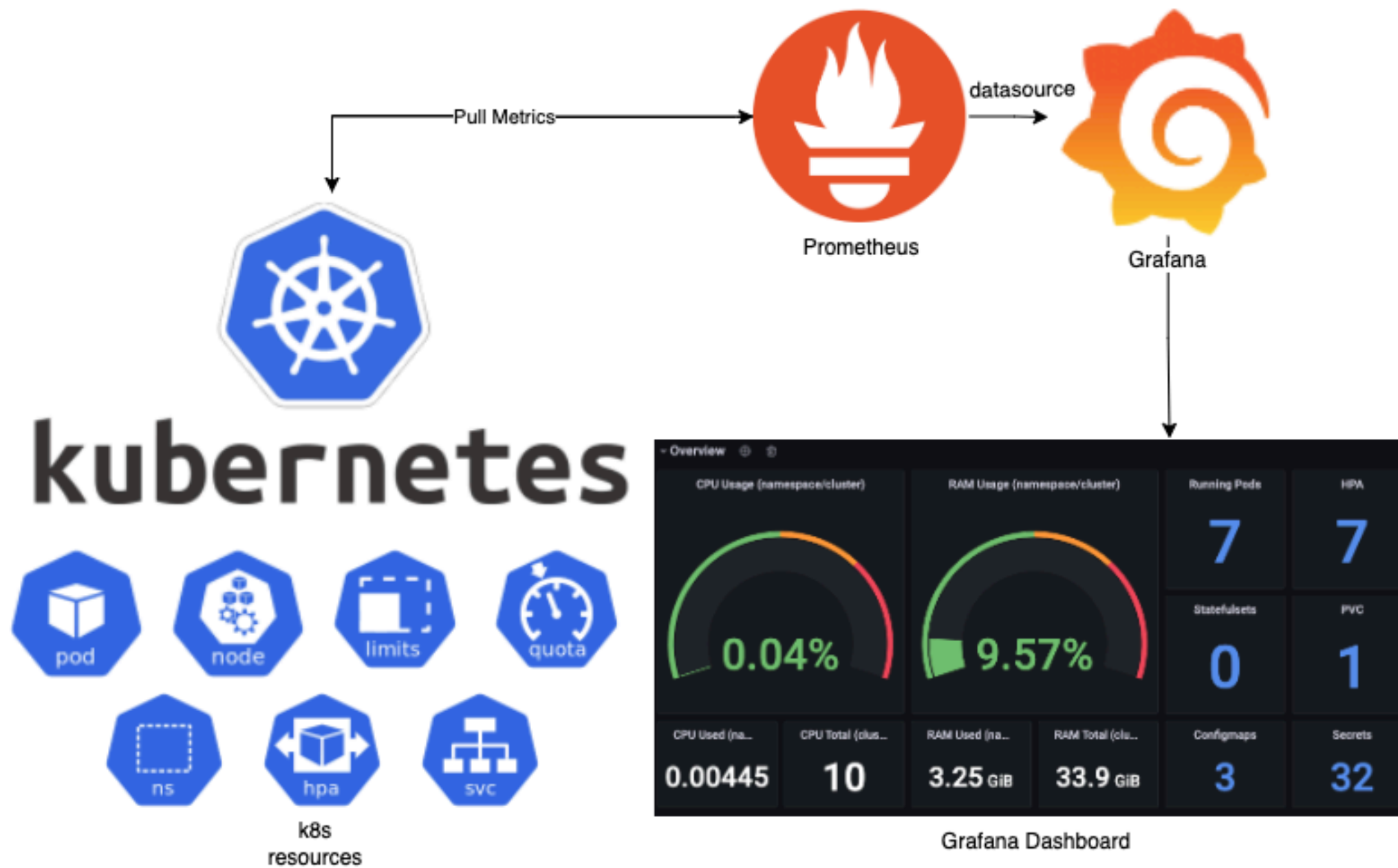
# Parte C{

Observabilidad y  
Troubleshooting



}

# Prometheus, Grafana y k8s {





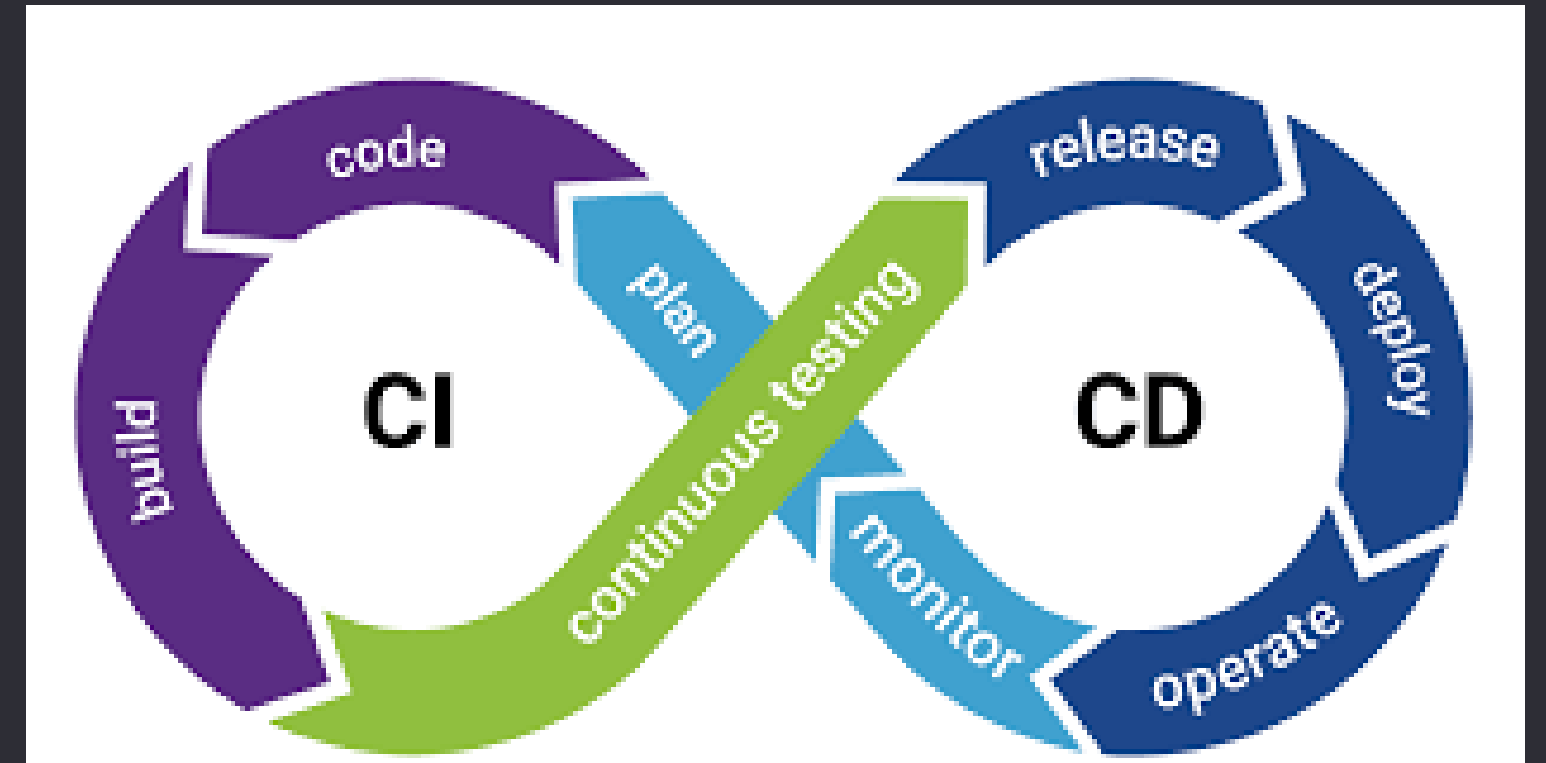
# Métricas {



}

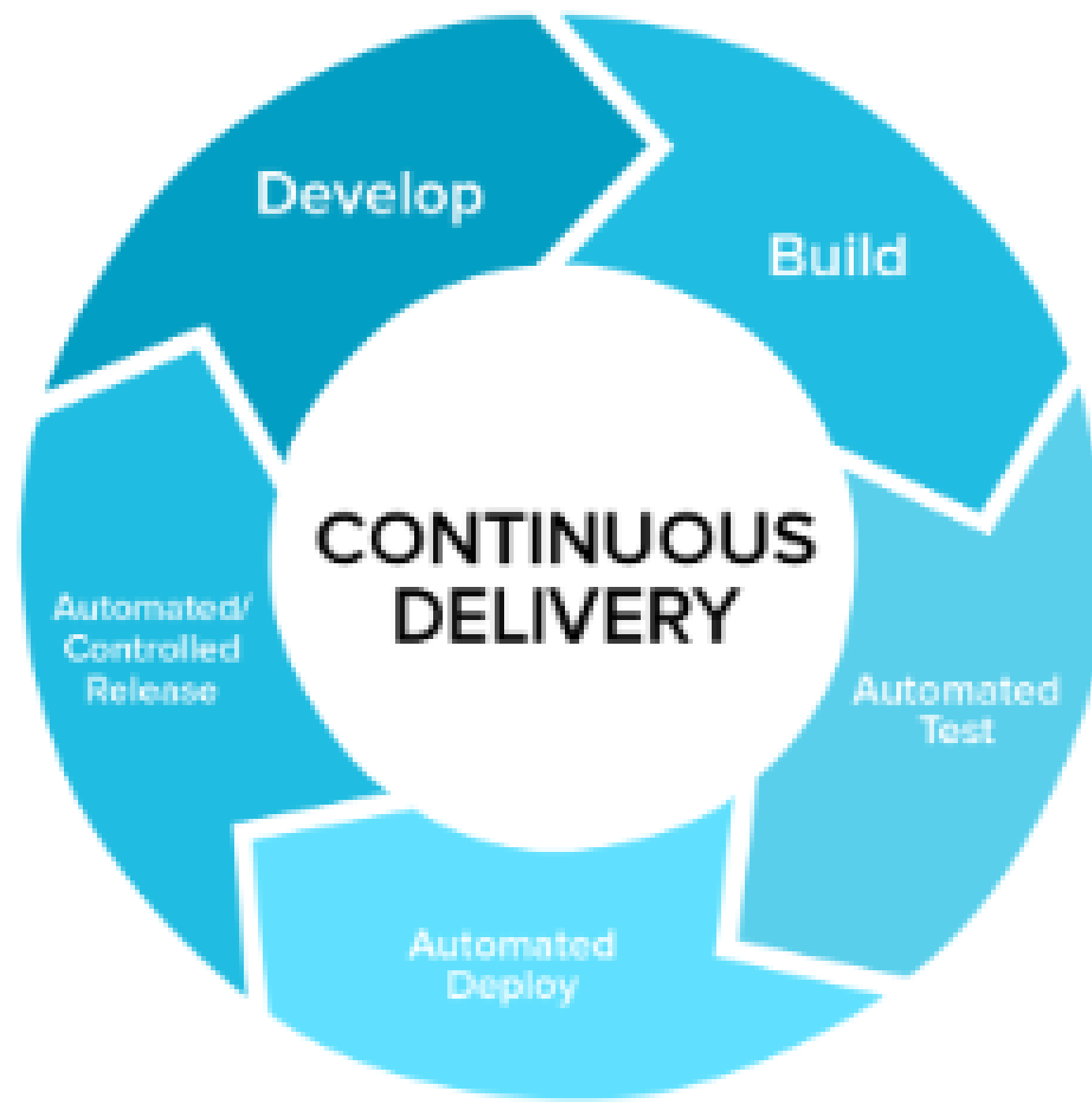
Parte D{

CI/CD (Integración continua /  
Despliegue continuo)



}

# Continuous Delivery



# continuous deployment

