



Actividad 2

JHARVY JONAS CADILO TARAZONA
ALEM CINVER ESPINOZA VALERA
LUIS ALBERTO ALANYA CAMPOS

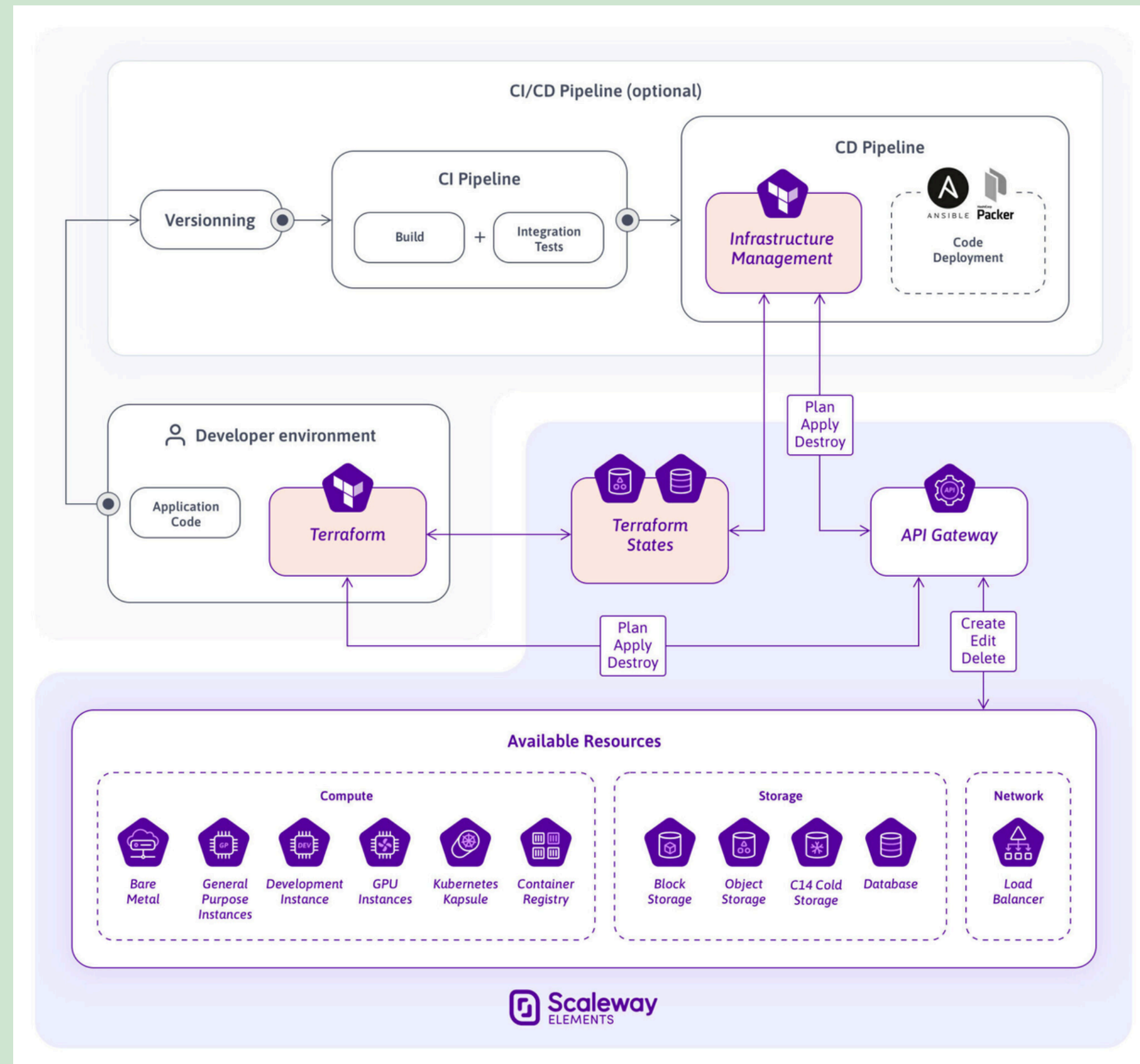
Hashicorp Terraform

Con Terraform, los usuarios
pueden definir su
infraestructura en la nube en
un lenguaje de configuración
declarativo denominado
Lenguaje de configuración de
HashiCorp (HCL)

Estructura basico de un
modulo de Terraforms

```
.  
├── main.tf  
├── outputs.tf  
├── README.md  
└── variables.tf
```

Terraform



Docker y Kubernetes

Ventajas de escalar una aplicación en un evento

de alto tráfico en kubernetes:

1. Escalamiento automático

2. Balanceo de carga

3. Alta disponibilidad

4. Uso eficiente de recursos

5. Flexibilidad



kubernetes

Docker y Kubernetes

Flujo simple de despliegue donde un desarrollador hace un cambio en el código, se construye una nueva imagen Docker y se actualiza un Deployment de Kubernetes:

1. El desarrollador hace un cambio en el código
2. El desarrollador actualiza el código de la aplicación en su computadora.
3. El desarrollador guarda y sube los cambios
4. Después de hacer los cambios, el desarrollador guarda esos cambios y los sube a un repositorio en línea, como GitHub.
5. El proceso de despliegue se activa automáticamente
6. Si hay una herramienta configurada para hacer despliegues automáticos, como Jenkins o GitHub Actions, se inicia cuando el código es subido al repositorio.

Docker y Kubernetes

Flujo simple de despliegue donde un desarrollador hace un cambio en el código, se construye una nueva imagen Docker y se actualiza un Deployment de Kubernetes:

7. Se construye una nueva imagen de Docker
8. La herramienta de despliegue construye una nueva imagen Docker con el código actualizado. Esto es como un “paquete” que contiene todo el código y las dependencias necesarias para ejecutar la aplicación.
9. El Deployment de Kubernetes se actualiza
10. Una vez que se tiene la nueva imagen Docker, se le dice a Kubernetes (el sistema que maneja los servidores) que actualice la aplicación con esta nueva versión.
11. Esto se hace con un comando simple que indica a Kubernetes que use la nueva imagen.
12. Kubernetes despliega la nueva versión

Docker y Kubernetes

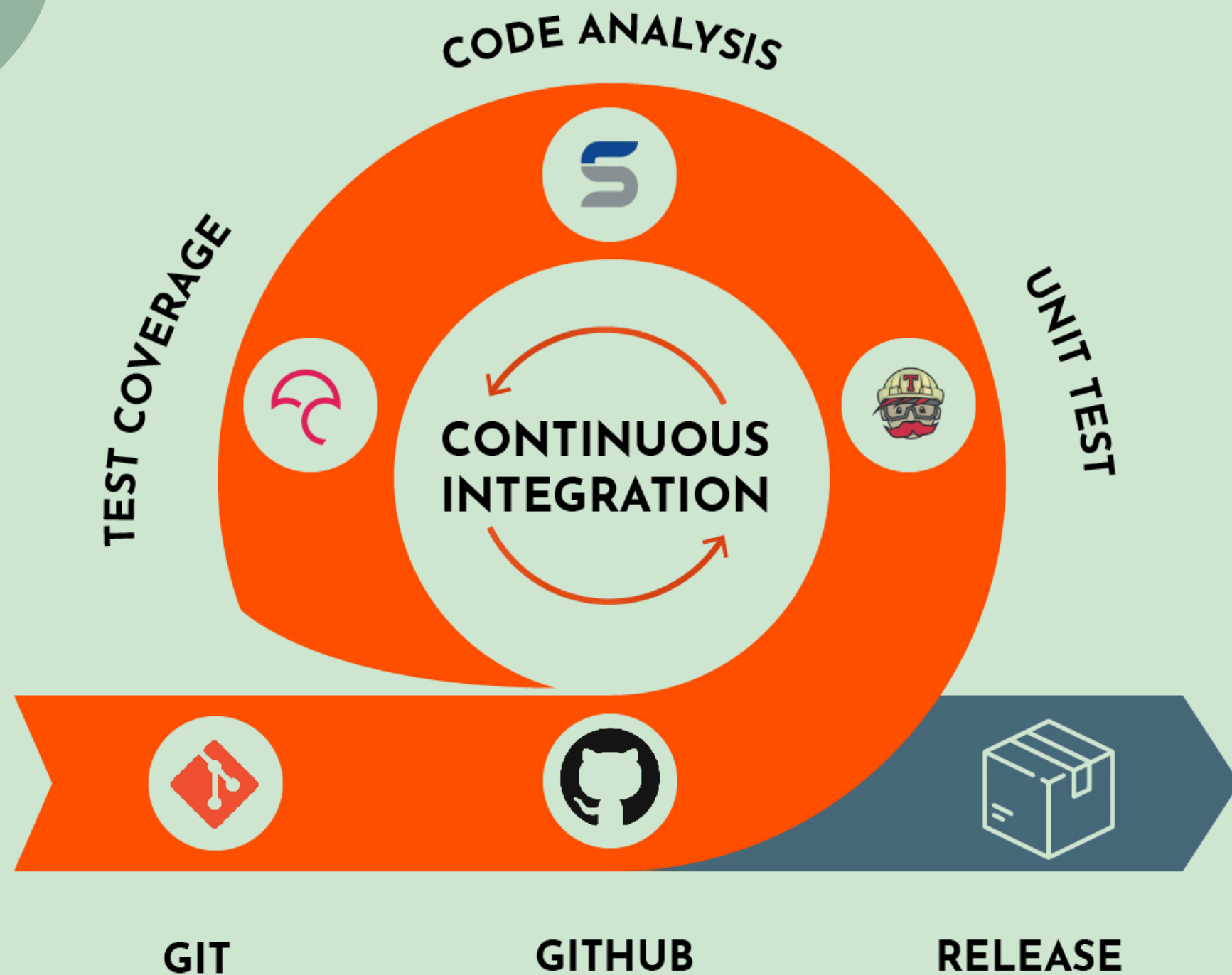
Flujo simple de despliegue donde un desarrollador hace un cambio en el código, se construye una nueva imagen Docker y se actualiza un Deployment de Kubernetes:

13. Kubernetes reemplaza los “servidores” viejos con la nueva versión de la aplicación, sin interrumpir el servicio.

14. Verificación

15. Finalmente, se verifican los resultados para asegurarse de que la nueva versión esté funcionando bien.

CI (Integración continua)



Facilita la integración de cambios de código en repositorios compartidos.

Proceso de automatización de testeo para a la hora de mergear algún cambio en el código no rompa la aplicación.

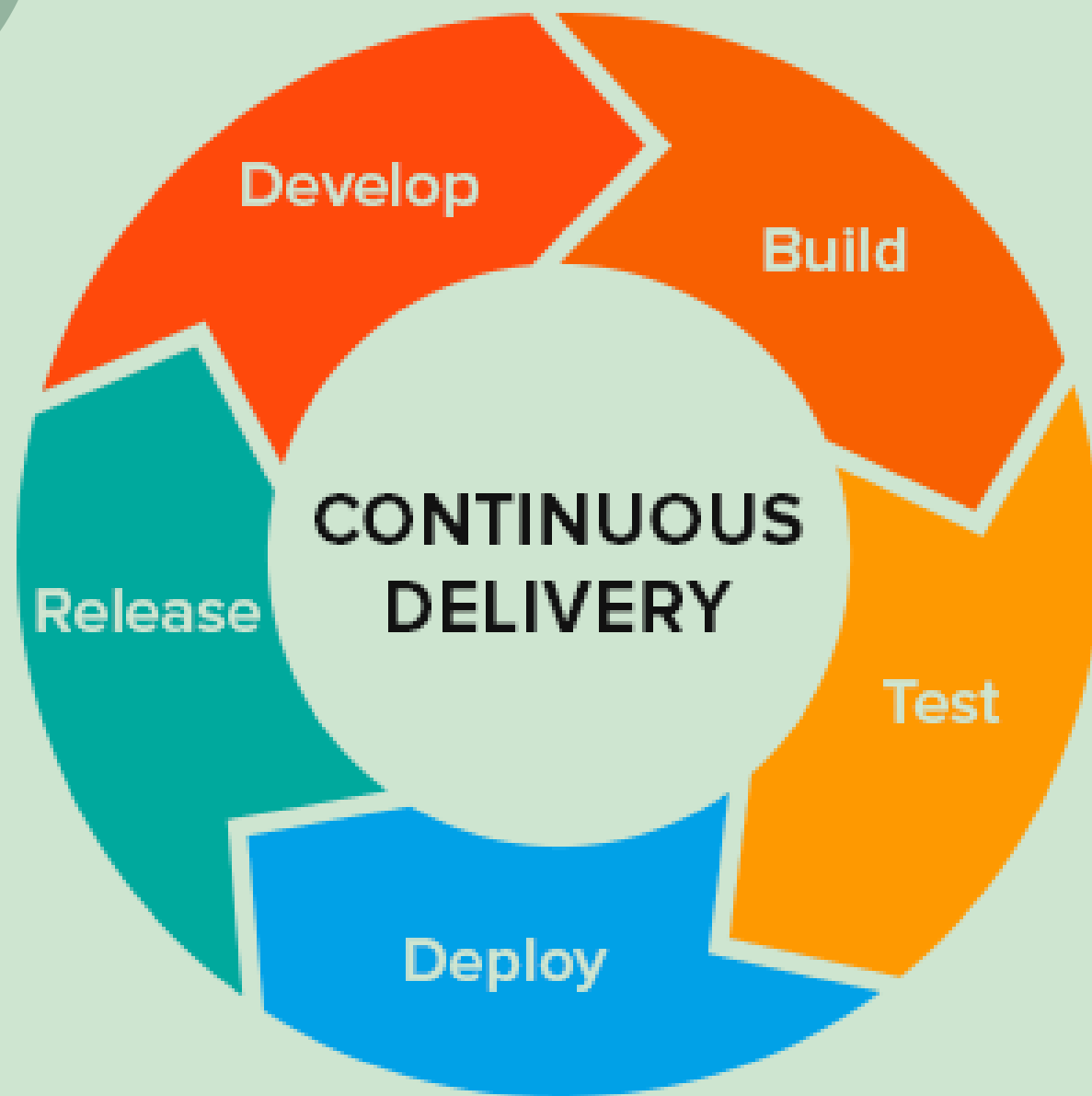


Bug detection



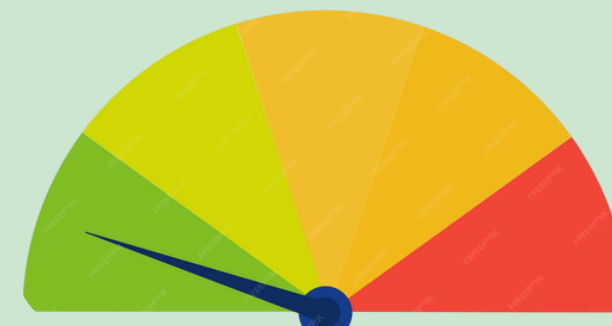
Faster Release Cycles

CD (Despliegue continuo)



El proceso de despliegue continuo revisa y valida en un repositorio el código tras la automatización de la integración por el proceso CI.

La meta del uso de CD es asegurar que el código base esté siempre disponible para su ejecución, permitiendo que el equipo de operaciones mande la aplicación a producción.



Bajo riesgo en las releases



incremento del time-to-market

Herramientas de integración



Jenkins

Herramienta de automatización que habilita el desarrollo de builds, testeos y deployment de nuestros softwares.



GitLab

plataforma de gestión de repositorios Git que permite a los equipos de desarrollo colaborar en proyectos de software.