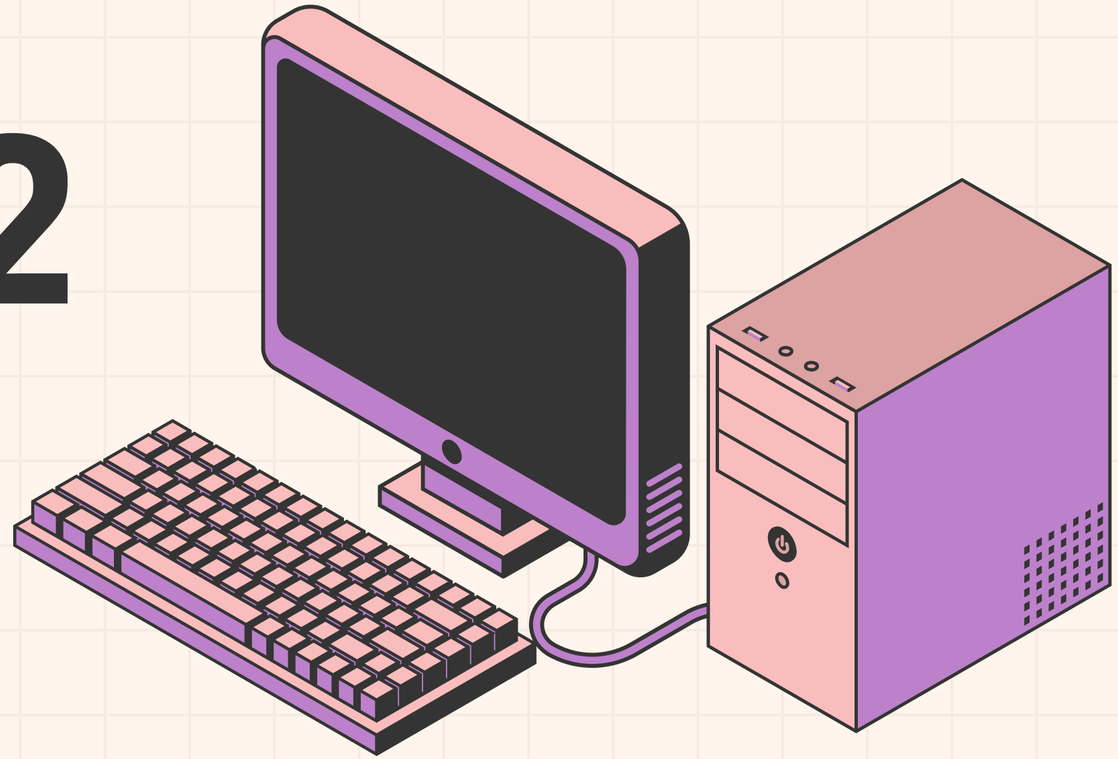
ACTIVIDAD 2

Bianca Merchan Torres
Jesus Diego Santa Cruz Basilio
Daren Herrera Romo



TERRAFORM

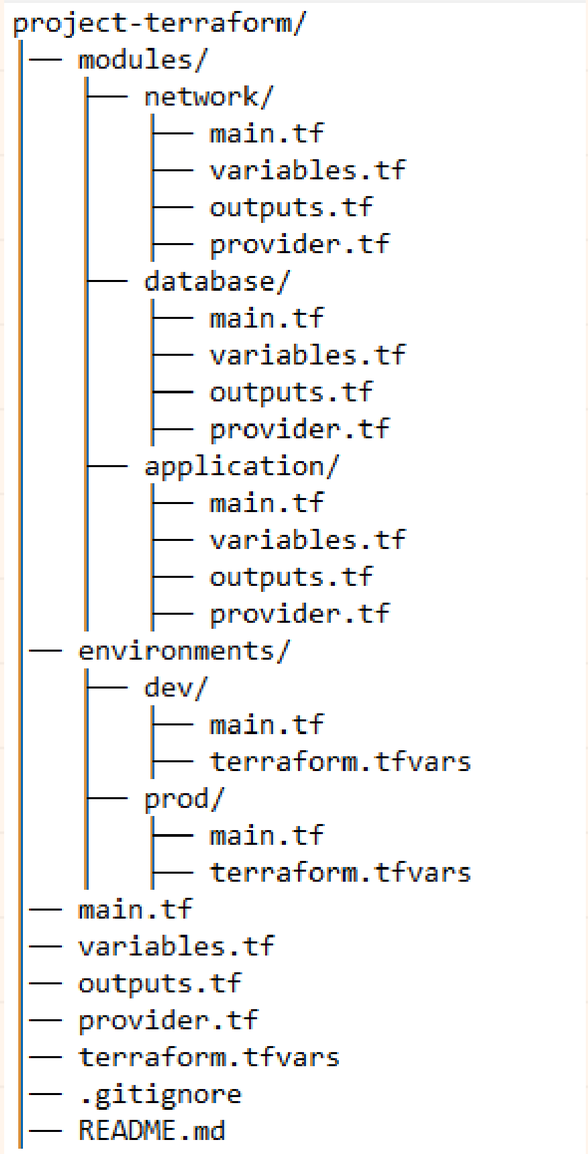
Terraform es una herramienta que permite a los usuarios gestionar infraestructura de un proyecto mediante archivos de configuración. Para organizar mejor los proyectos, Terraform usa módulos, los cuales agrupan archivos de configuración relacionados para facilitar su reutilización y mantenimiento.

Un modulo consiste de una colección de archivos .tf y/o .tf.json que se mantienen juntas en un directorio.

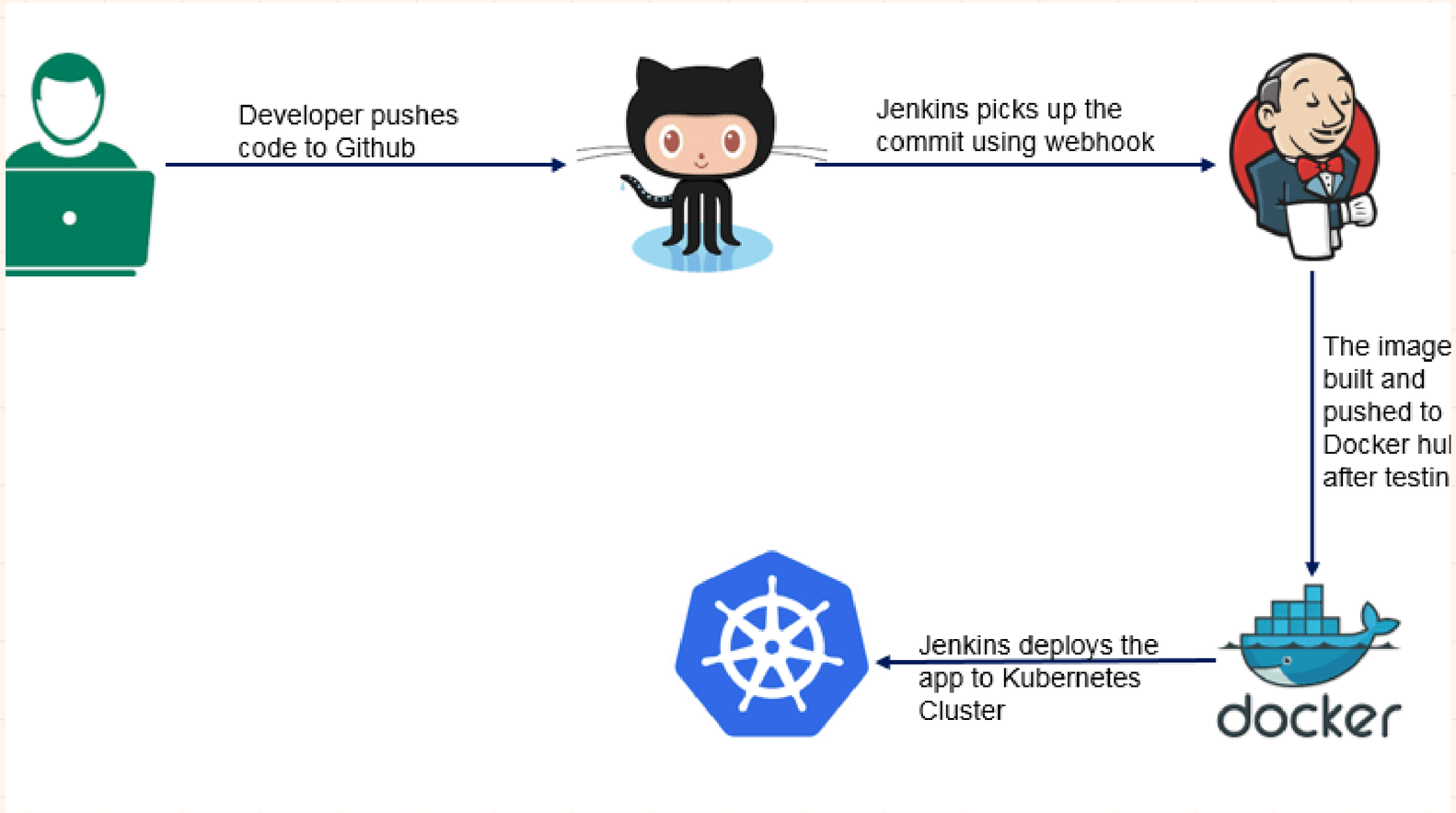
Toda configuración de Terraform tiene al menos un módulo, cono cido como el **modulo raíz** que consiste en los recursos definidos en los archivos .tf en el directorio de trabajo principal.

EJEMPLO DE ESTRUCTURA DE UN PROYECTO

- **main.tf:** Define los recursos principales del módulo.
- **variables.tf:** Contiene los parámetros de entrada
- **outputs.tf:** Define las salidas del módulo
- **providers.tf** (opcional): Declara los proveedores de nube o infraestructura usados.
- **terraform.tfvars** (opcional): Permite definir valores por defecto para las variables.

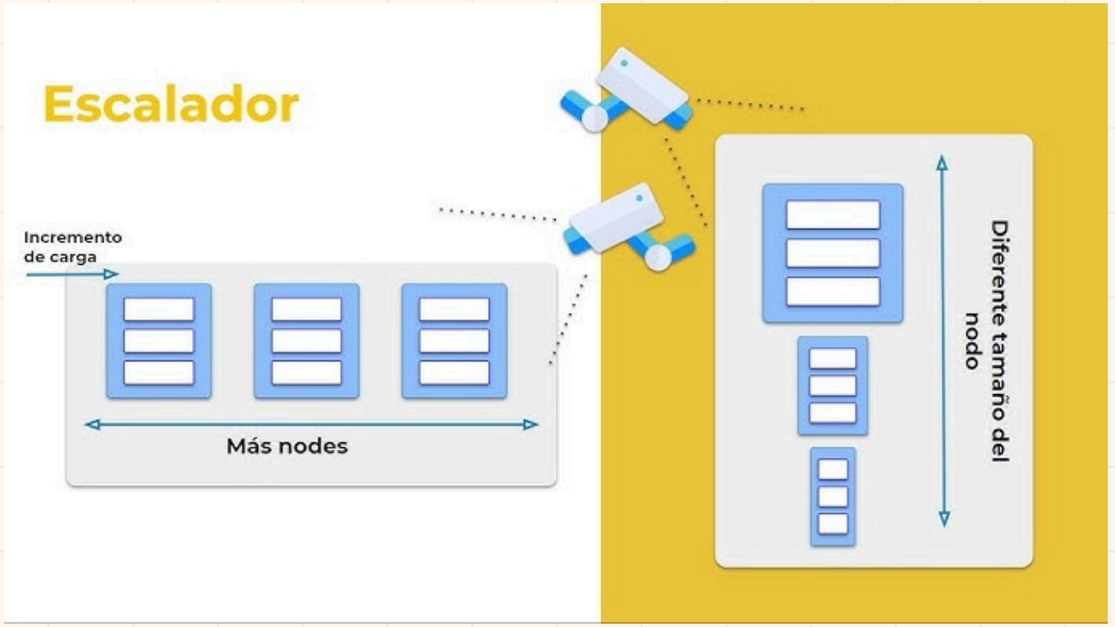


FLUJO SIMPLE DE DESPLIEGUE



VENTAJAS DE USAR KUBERNETES PARA ESCALAR UNA APLICACIÓN EN UN EVENTO DE ALTO TRÁFICO

Kubernetes posee una característica importante llamada autoescalado. Este permite que las apicaciones se ajusten automáticamente a los cambios den la demanda eficientemente. Para esto, se emplea el escalado horizontal de pods (HPA). Este es un mecanismo que ajusta automáticamente la cantidad de copias de los Pods en función al uso de la CPU u otras métricas.



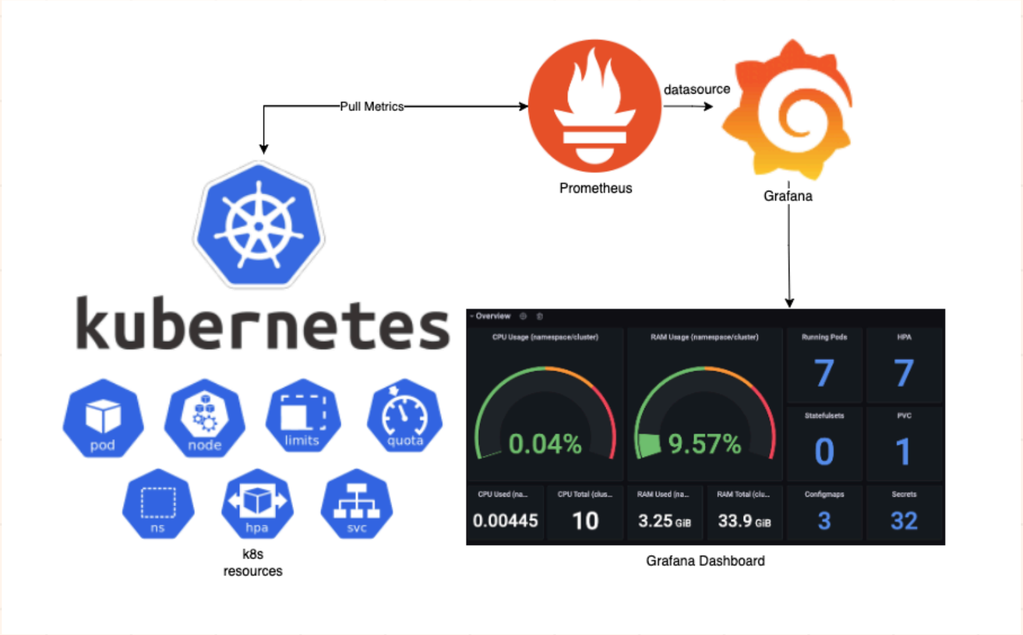
Esto es útil ya que **maneja picos de tráfico sin que la aplicación caiga**, **evita problemas de hardware distribuyendo la carga** y también **se adapta a las interrupciones de red** asegurando que la aplicación siempre esté disponible.

Tarea teorica 3

INTEGRACION DE PROMETHEUS Y GRAFANA CON KUBERNETES PARA EL MONITOREO DE CONTENEDORES Y CLUSTERES

Prometheus es un sistema de monitoreo y alerta de código abierto. Se integra con Kubernetes para extraer datos sobre el rendimiento de los nodos, pods y contenedores. Utiliza lo que se conoce como pull para consultar periodicamente las metricas de los servicios mediante su lenguaje PromQL. Grafana se utiliza para representar gráficamente las métricas recolectadas por Prometheus.

1. Necesitamos un servidor de Grafana, corriendo
2. Necesitamos que en el mismo servidor donde se encuentre Grafana haya un servidor de Prometheus corriendo como servicio de Linux
3. Necesitamos que haya un servidor Prometheus en cada uno de los clusters de Kubernetes



SET DE METRICAS Y ALERTAS MÍNIMAS PARA UNA APLICACION WEB

- **Latencia de peticiones:** Medirlas en ms y establecer una alerta en el caso de se supere N ms en un intervalo determinado de tiempo.
- **Uso del CPU:** El porcentaje de CPU utilizado en cada nodo o pod y establecer una alerta si el uso supera cierto porcentaje.
- **Tiempo de respuesta de la base de datos:** Tiempo de respuesta en consultas muy importantes y establecer una alerta en el caso de que se supere el umbral de tiempo.
- **Uso de memoria:** Medir cuanta memoria se usa respecto a la memoria que no se usa y establecer una alerta en el caso de que se use mas de la que no se use.

DIFERENCIA ENTRE LA ENTREGA CONTINUA Y DESPLIEGUE CONTINUO

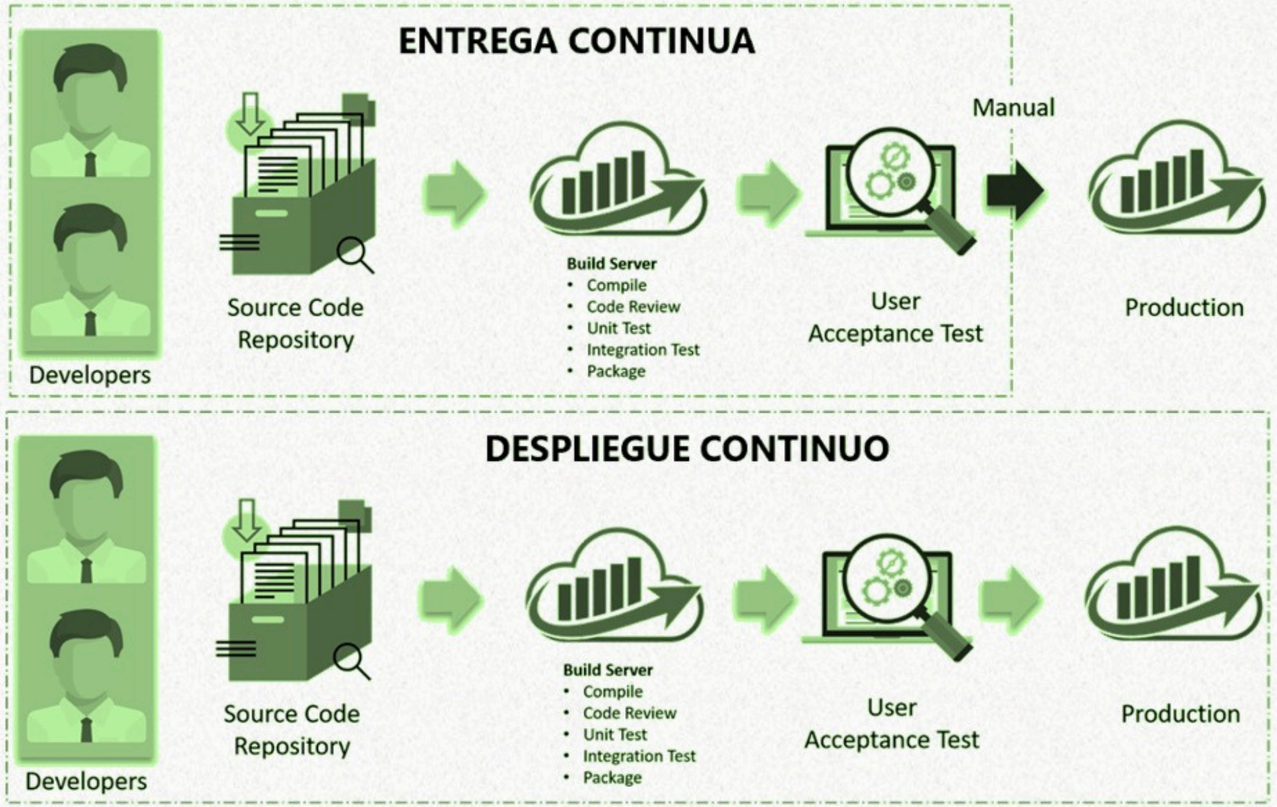
La entrega continua (CD) cierra el ciclo mediante la automatización de aspectos de la entrega de software. A medida que se abordan los comentarios y se implementan correcciones, estos cambios se cargan automáticamente hasta que el equipo tome la decisión de enviar la aplicación a producción. La CD da lugar a un producto que se puede desplegar, pero depende de la autorización humana para implantarlo, lo que permite a los equipos decidir qué se debe lanzar y cuándo. Los desarrolladores pueden seguir perfeccionando la aplicación antes de entregarla al usuario final.

El despliegue continuo es similar a la entrega continua. La principal diferencia con el despliegue continuo es que, en lugar de requerir una autorización humana para lanzar un producto, el despliegue continuo envía cada cambio de forma automatizada, que luego se envía inmediatamente a producción. No se espera un ciclo de aprobación manual, lo que significa que el código en sí mismo debe haberse probado lo suficiente antes de pasar a producción.

Tarea teorica 4

Entrega Continua vs Despliegue Continuo

La Entrega Continua tiene una decisión explícita (manual) para pasar a producción. Con el Despliegue Continuo, las liberaciones son promovidas automáticamente a producción.

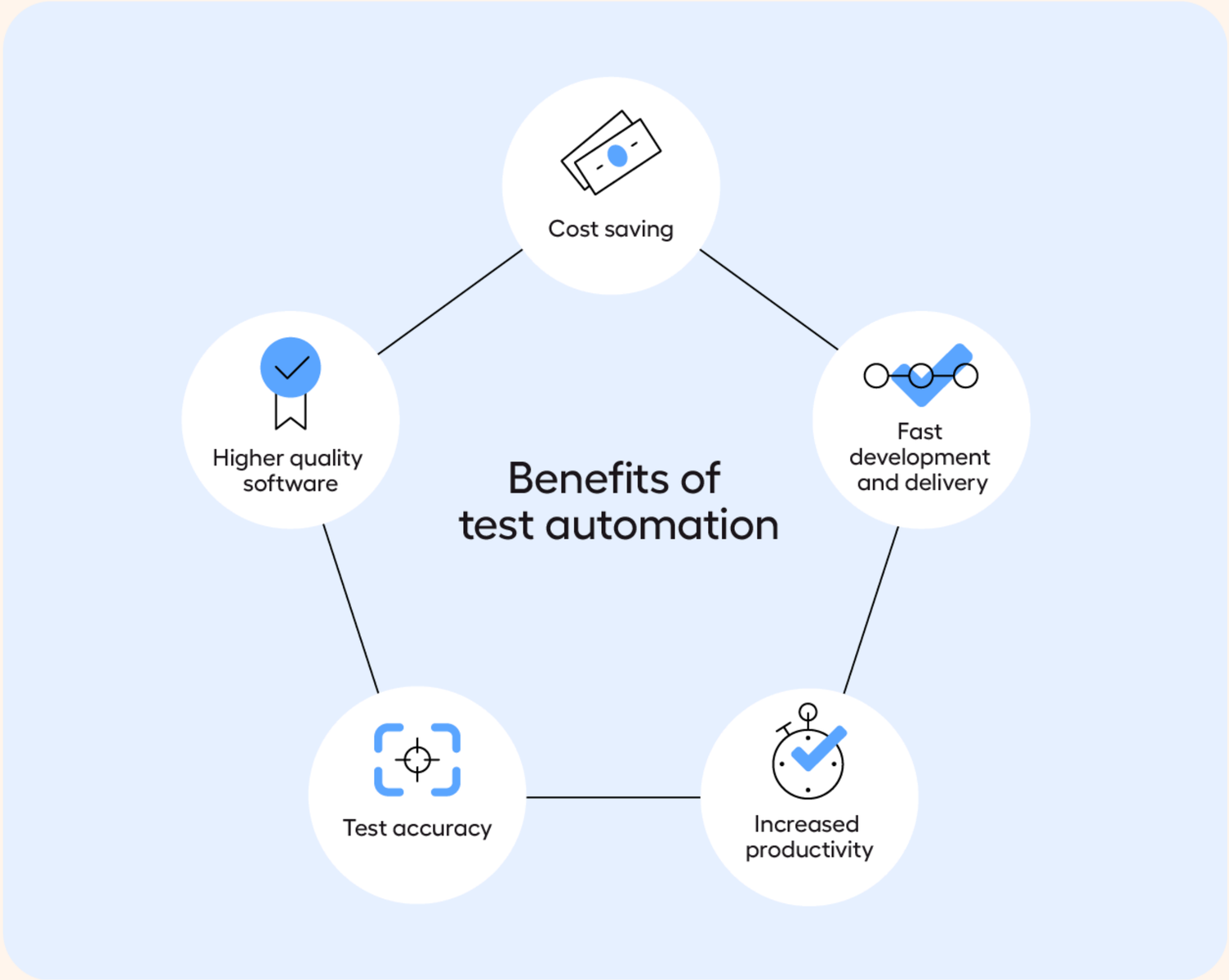


DOu DevOps united

Qualizens

¿POR QUÉ ES IMPORTANTE IMPLEMENTAR PRUEBAS AUTOMÁTICAS (UNITARIAS, DE INTEGRACIÓN, DE SEGURIDAD) DENTRO DEL PIPELINE?

Porque aumentan la confiabilidad, consistencia y efciencia tanto del equipo como del producto final. Ayuda a ahorrar tiempo para que el equipo pueda centrarse en otras tareas. Además, reduce la posibilidad de que ocurran errores humana. La automatizacion de pruebas tambien ofrece flexibilidad, ya que los equipos de desarrollo puede reutilizar sus scripts de prueba para cualquier conjunto de pruebas relacionado.


Cost saving
Higher quality software
Fast development and delivery
Benefits of test automation
Test accuracy
Increased productivity