



Escuela profesional de Ciencia de la Computación

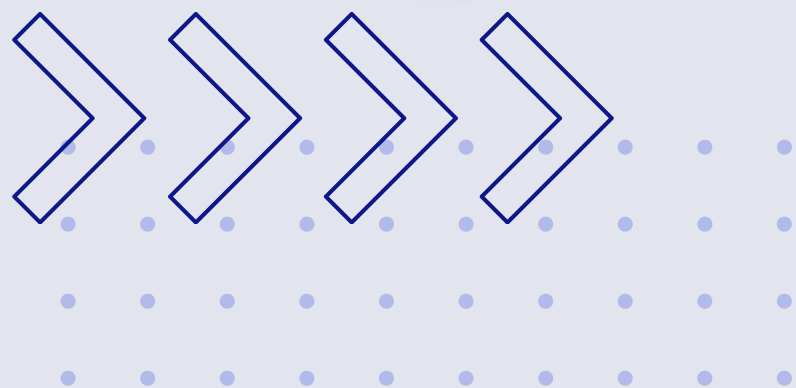


DEL CÓDIGO A LA PRODUCCIÓN: INFRAESTRUCTURA, CONTENEDORES, DESPLIEGUE Y OBSERVABILIDAD

Actividad 2

Integrantes:

- Quispe Villena Renzo
- Cruz Tairo Jhon Emdilson

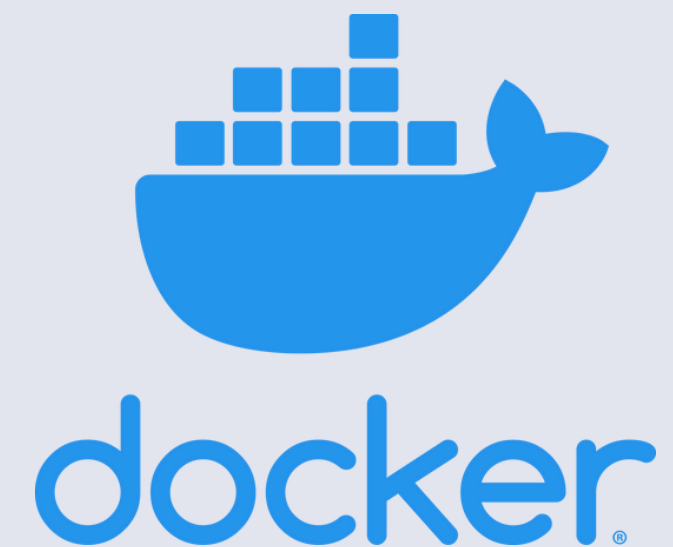


Para entregar software de forma rápida y segura se utilizan herramientas modernas que nos ayudan automatizar tareas, escalar fácilmente(escalabilidad) y monitorear en tiempo real(observabilidad). Se detallara la importancia de las herramientas estudiadas:

```
each: function(e, t, n) {
  var r, i = 0;
  n = e.length;
  n = N(n);
  if (n) {
    if (a) {
      for (; i < n; i++)
        if (r = t.apply(e[i], n), r === !1) break;
    } else
      for (i in e)
        if (r = t.apply(e[i], n), r === !1) break;
    } else if (a) {
      for (; i < n; i++)
        if (r = t.call(e[i], i, e[i]), r === !1) break;
    } else
      for (i in e)
        if (r = t.call(e[i], i, e[i]), r === !1) break;
    return e;
  },
  trim: b && !b.call("\uffff\u00a0") ? function(e) {
    return null == e ? "" : b.call(e)
  } : function(e) {
    return null == e ? "" : (e + "").replace(C, "")
  },
  makeArray: function(e, t) {
    var n = t || [];
    return null != e && (N(Object(e)) ? x.merge(n, "string" == typeof e ? [e] : e) : b.call(n, e), n
  ),
  isArray: function(e, t, n) {
    var r;
    if (t) {
      if (n) return e.call(t, n, n);
      for (r = t.length, n = n > 0 ? Math.max(0, r + n) : 0; r > n; r++)
        if (n in t && t[n] === e) return e;
    }
  }
```

laC: Permite definir y configurar servidores, redes y otros recursos con archivos de texto. Terraform es un ejemplo de herramienta laC que organiza la infraestructura en módulos reutilizables. Esto mejora el orden, la automatización y facilita la colaboración entre equipos.

Contenedores: Los contenedores empaquetan todo lo que necesita una aplicación para funcionar. Con ellos las aplicaciones se ejecutan en cualquier entorno, esto evita errores y hace mas facil el despliegue.



Kubernetes: Es un orquestador de contenedores. Permite distribuir el tráfico, escalar automáticamente cuando hay más usuarios y recuperar contenedores si fallan. Todo esto sin intervención manual, haciendo que la aplicación esté siempre disponible.

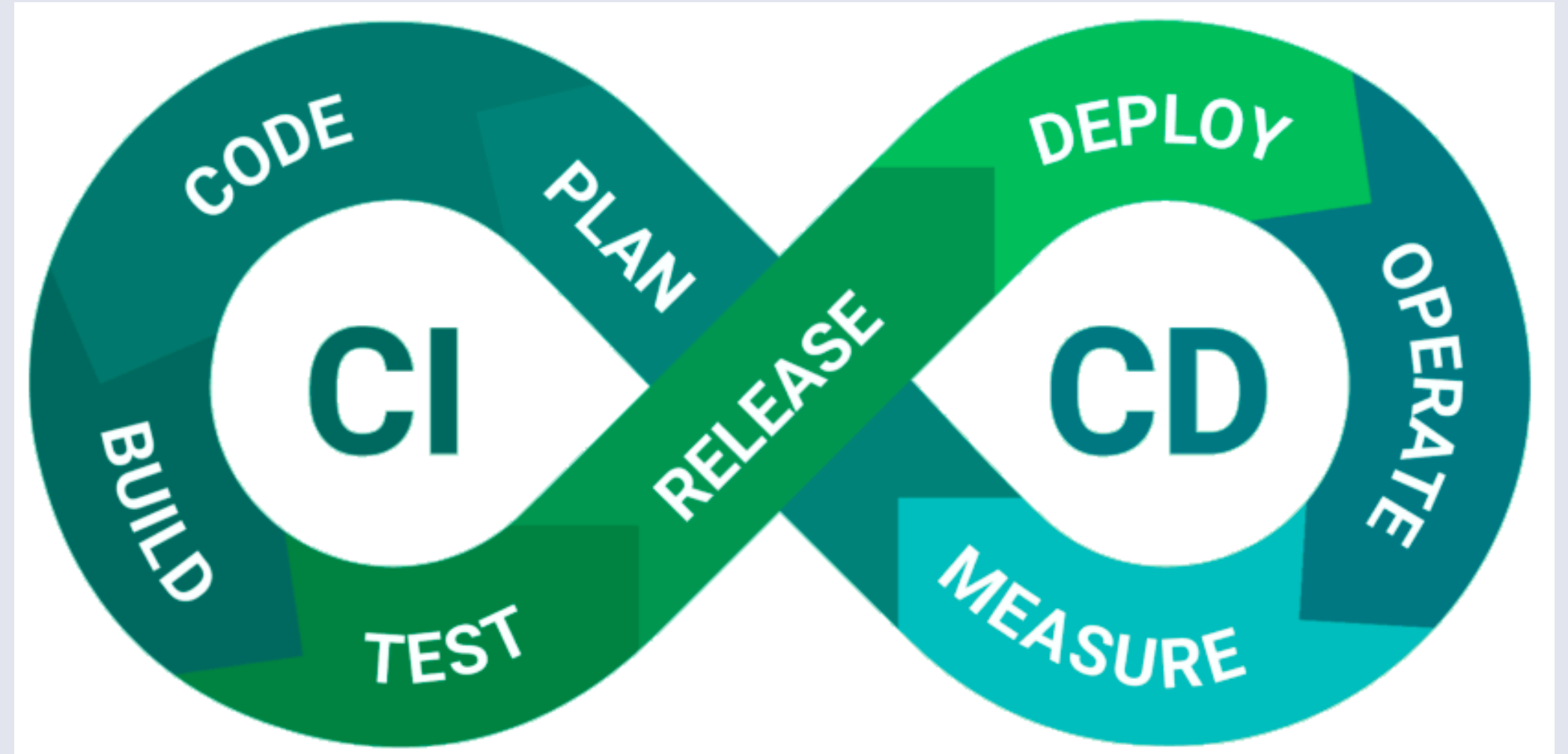


Observabilidad: Prometheus recolecta métricas del sistema, como uso de CPU, memoria y errores y con Grafana se muestra dicha información en gráficos en tiempo real. Juntos permiten monitorear el estado de la aplicación y activar alertas cuando hay errores.



CI/CD:

Automatiza desde que el desarrollador sube un cambio hasta que se despliega en producción. Incluye pruebas automáticas (unitarias, integración y seguridad), creación de imágenes Docker y despliegue en Kubernetes. Esto reduce errores, acelera las entregas y mejora la calidad del software.



FLUJO TEORICO

1. COMMIT DEL CODIGO

Los desarrolladores desarrollan cambios en el código

Hacen commit en el repositorio

Se ejecutan pipeline CI/CD

2. PIPELINE DE CI BUILD Y PRUEBAS

Integración continua donde se compila el código, se ejecutan pruebas unitarias y se verifica la calidad del software

Asegura que los cambios no rompen la funcionalidad existente.

3. Construcción y publicación de la imagen contenedor

Una vez que las pruebas son aprobadas, se construye la imagen del contenedor

Se publica en un registro de contenedores

4. PROVISIONAMIENTO DE RECURSOS IaC

Se utiliza IaC para crear y configurar de forma automatizada los recursos en la nube necesarios para el despliegue

5. DESPLIEGUE EN KUBERNETS

El pipeline de entrega continua despliega la imagen publicada en el clúster de Kubernetes.

Se utilizan manifiestos YAML o herramientas como Helm para orquestar el despliegue

7. Monitoreo con Prometheus

Prometheus se configura para recolectar métricas del clúster y de las aplicaciones. Recopilan datos como uso de CPU, memoria, estado de los pods, entre otros, a través de endpoints expuestos por Kubernetes y los contenedores

6. ORQUESTACION Y GESTION DE APLICACIONES

Kubernetes se encarga de gestionar los contenedores desplegados

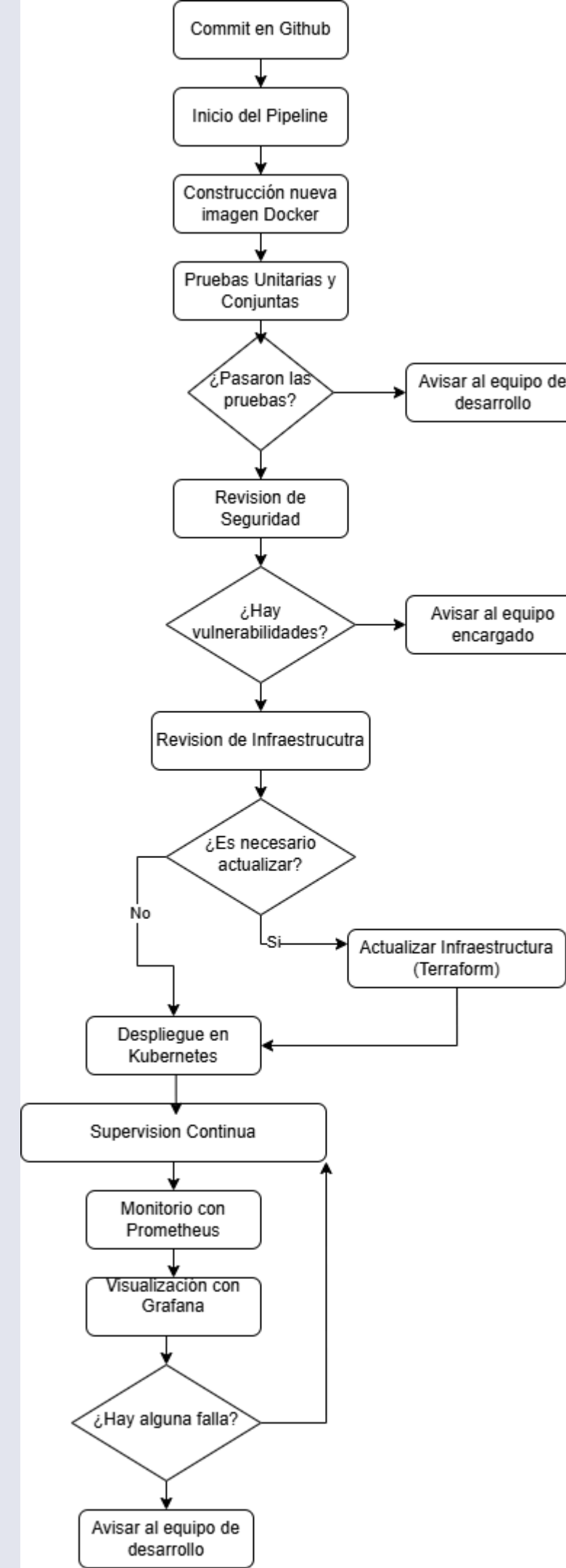
Se asegura que estén siempre en ejecución y escalado según la demanda del momento

8. Visualización en Grafana

Grafana se integra con Prometheus para mostrar las métricas en dashboards interactivos y en tiempo real.

9. Monitoreo constante y aviso de fallas al equipo correspondiente

DIAGRAMA DE FLUJO



GRACIAS