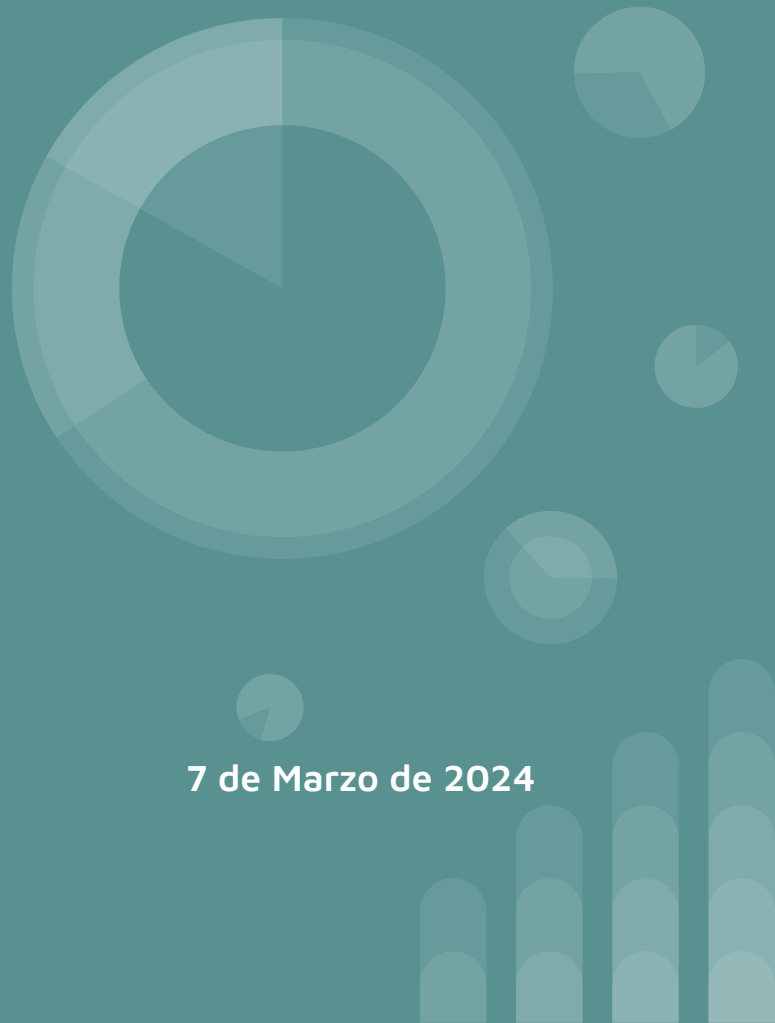


# Introducción a Tekton CI/CD

Andrés Torres  
Ingeniero de software  
Broadcom

7 de Marzo de 2024





## Sobre mí

- Desarrollador web por 10+ años
- Msc Ciencias de la Computación
- Ruby + Go + Kubernetes
- CKAD - CKA
- @elfotografo007





# Programa

- Introducción a CI/CD
- Introducción a Tekton
- Conceptos de Tekton
- Demo
- Casos de uso

The background is a solid orange color. In the top-left corner, there are three vertical bars of varying heights, each composed of three overlapping circles. In the bottom-right corner, there are four vertical bars of varying heights, each composed of three overlapping circles.

# **CI/CD: Continuous Integration / Continuous Delivery-Deployment**



# Continuous Integration

Integrar los cambios de código a las ramas principales de manera frecuente, compilando, empaquetando y probando dicho código de manera automática.

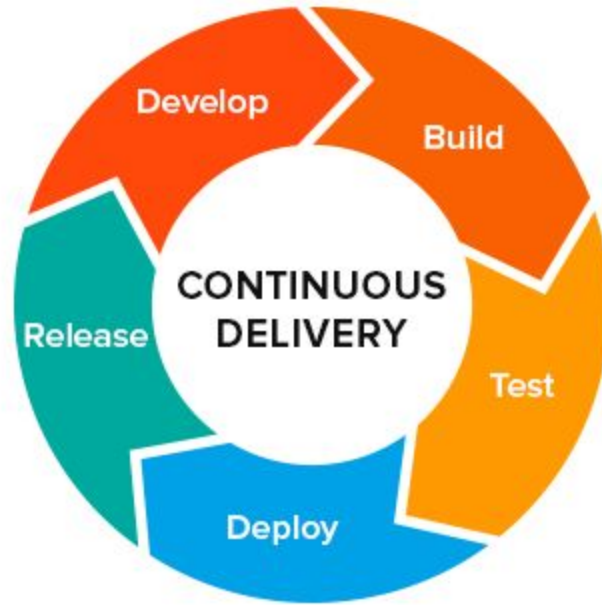
# CI



# **Continuous Delivery/Deployment**

Automatizar el despliegue de dichos cambios a producción para hacerlo también de manera frecuente

CD



Fuente: <https://openpracticelibrary.com/practice/continuous-delivery/>



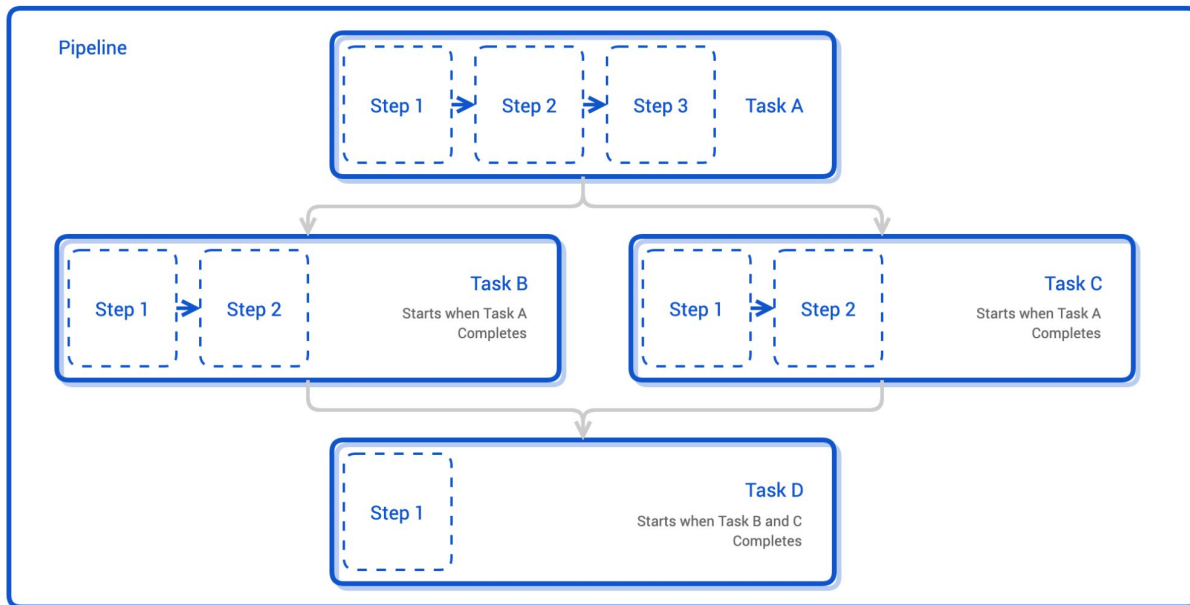
Framework Open Source de Kubernetes para crear sistemas CI/CD, permitiendo a los desarrolladores compilar, probar y desplegar software en diferentes proveedores cloud y sistemas on-prem.

- Estandariza el CI/CD
- Se integra con otras herramientas
- Supremamente flexible
- Solución de facto en Kubernetes
- Puede generar metadatos firmados para garantizar la seguridad en la cadena de suministro
- Tekton Hub: Biblioteca de tareas





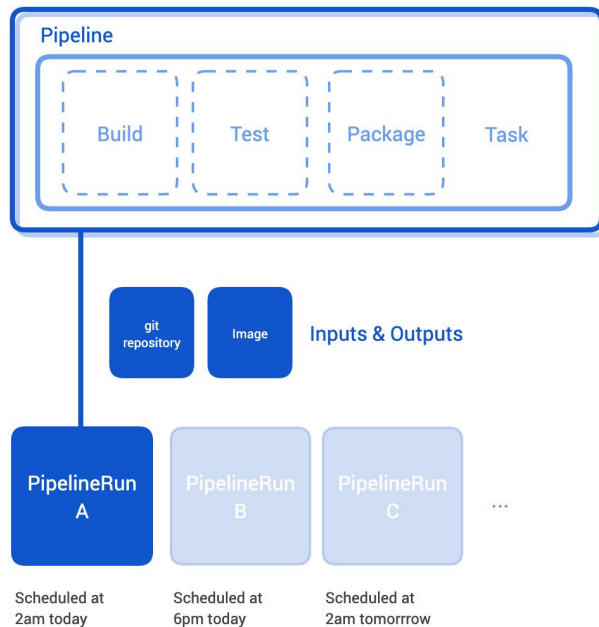
# Conceptos



Fuente: <https://tekton.dev/docs/concepts/concept-model/>



# PipelineRun y TaskRun



Fuente: <https://tekton.dev/docs/concepts/concept-model/>



```
apiVersion: tekton.dev/v1
kind: Pipeline
metadata:
  name: node-example
spec:
  params:
    - name: repository
  workspaces:
    - name: shared-data
  tasks:
    - name: git-clone
      taskRef:
        name: git-clone
      params:
        - name: url
          value: ${params.repository}
      workspaces:
        - name: output
          workspace: shared-data
    - name: test
      workspaces:
        - name: source
          workspace: shared-data
      runAfter:
        - git-clone
      taskSpec:
        steps:
          - name: lint
            image: node
            script: |
              #!/usr/bin/env bash
              cd ${workspaces.source.path}
              npm ci
              npm run test:eslint
```



# Demo

## Casos de uso

- Open Source
- Evitar “vendor lock-in”
- Costos
- Clúster de Kubernetes ya disponible
- Necesidad de correr on-prem o edge
- Ambientes regulados o “air-gapped”





## Material de apoyo





# Referencias

- The State of Continuous Delivery - The Evolution of Software Delivery Performance | Q1 2023, SlashData
- <https://openpracticelibrary.com/practice/continuous-delivery/>
- DevOps Culture and Practice with OpenShift, Beattie Tim et al.
- <https://tekton.dev/docs>
- <https://hub.tekton.dev/>