



# Openshift desde cero

---

ESTRATEGIAS EN EL BUILD DE CODIGO FUENTE

# Openshift desde Cero

## □ESTRATEGIAS POSIBLES PARA EL BUILD DE CÓDIGO FUENTE

- **Source:** esto utiliza Source-to-Image para producir imágenes listas para ser ejecutadas mediante la inyección del código fuente (u otros activos) en una Image Builder.
- **Docker:** Esto utiliza docker build desde un Dockerfile y los archivos fuente asociados y crear una imagen ejecutable.
- **Pipeline** Esto usa Jenkins y un flujo de trabajo definido por un Jenkinsfile (versión 3) para crear un pipeline para crear una imagen ejecutable. En version 4 se usa el framework Tekton
- **Custom:** Esto utiliza su propia imagen personalizada para controlar el proceso de compilación para crear la imagen ejecutable.

# Openshift desde Cero

- Se usa la siguiente opción para determinar el tipo de build.

```
oc new app apli1 - -strategy=valor
```

- Puede tomar el valor de
  - Docker
  - source
  - custom

# Openshift desde Cero

- También lo podemos usar dentro del fichero YAML del BuildConfig

```
strategy:  
  dockerStrategy:  
    from:  
      kind: "ImageStreamTag"  
      name: "debian:latest"
```

```
strategy:  
  sourceStrategy:  
    from:  
      kind: "ImageStreamTag"  
      name: "incremental-image:latest"
```

```
strategy:  
  customStrategy:  
    from:  
      kind: "DockerImage"  
      name: "openshift/sti-image-builder"
```

```
strategy:  
  jenkinsPipelineStrategy:  
    jenkinsfile: |-  
      node('agent') {  
        stage 'build'  
        openshiftBuild(buildConfig: 'ruby-sample-build', showBuildLogs: 'true')  
        stage 'deploy'  
        openshiftDeploy(deploymentConfig: 'frontend')  
      }
```