

Desafío 2 - Bootcamp Devops Engineer

Alumno: Ignacio Peretti

Objetivo:

Este desafío tiene como objetivo realizar la configuración de un webhook en un repositorio de Github y también crear un simple pipeline para ejecutar un build y testear un proyecto nodejs.

Escenario:

Nuestra organización nos encomendó la tarea de realizar una prueba de concepto para crear un CI/CD de nodejs, necesitan entender cómo se debe realizar el proceso de build para una API hecha en nodejs.

Por este motivo, nos entregaron una aplicación de ejemplo hecha en esta tecnología, este proyecto cuenta con una prueba integrada hecha en la herramienta Jest propia de este framework de desarrollo.

El referente de desarrollo, nos encargó realizar la configuración de un webhook que permita hacer un build automático cada vez que se produzca un push o un pull request, esto puede ayudar para automatizar la ejecución del job para este CI/CD.

Requisitos:

1. Crear un fork del siguiente proyecto en nuestra cuenta de Github.

<https://github.com/edgaregonzalez/nodejs-helloworld-api>

2. Realizar la configuración del github webhook para inicializar el job cada vez que se produce un push o un se crea un PR.

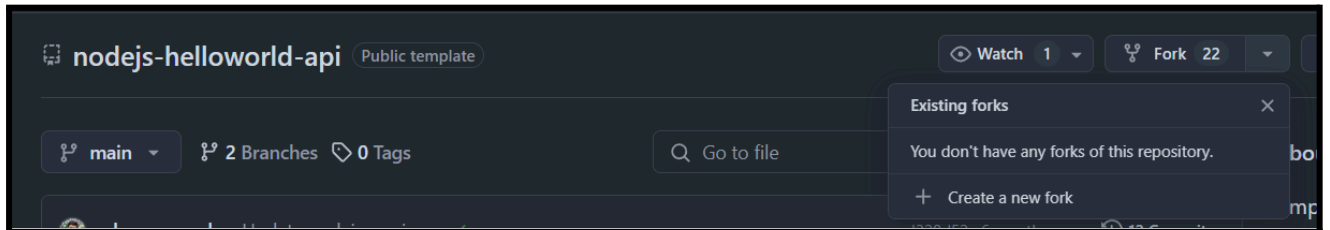
3. Exponer mediante ngrok el servicio de Jenkins.

4. Elaborar un jenkins pipeline para que ejecute los pasos para desarrollo, tomar las instrucciones del README.md.

1. Crear un fork del siguiente proyecto en nuestra cuenta de Github.

Accede al Repositorio Original: [nodejs-helloworld-api](#).

Forkea el Repositorio: Haz clic en el botón "Fork" en la parte superior derecha de la página para crear una copia del repositorio en tu cuenta de GitHub.



2. Configurar el Webhook de GitHub

En tu cuenta de GitHub, nos dirigimos al repositorio que acabamos de forkear.

Configuración: Ve a la pestaña "Settings".

Webhooks: En el menú lateral, selecciona "Webhooks".

Agregar Webhook: Haz clic en "Add webhook".

Payload URL: Ingresa la URL que te proporcionará ngrok (una vez que esté expuesto el servicio de Jenkins).

Content type: Selecciona application/json.

Eventos: Marca "Let me select individual events." y selecciona "Pushes" y "Pull requests".

Guardar: Haz clic en "Add webhook" para finalizar.

3. Exponer Jenkins con Ngrok

Instalar Ngrok: Si no lo tienes instalado, descarga Ngrok desde [aquí](#).

Ejecutar Jenkins: Asegúrate de que Jenkins esté corriendo en tu máquina (generalmente en el puerto 8080).

Iniciar Ngrok: Abre una terminal y ejecuta: ngrok http 8080

4. Crear un Jenkins Pipeline

Accede a Jenkins: Ve a tu instancia de Jenkins usando la URL proporcionada por Ngrok.
Crear un Nuevo Proyecto:

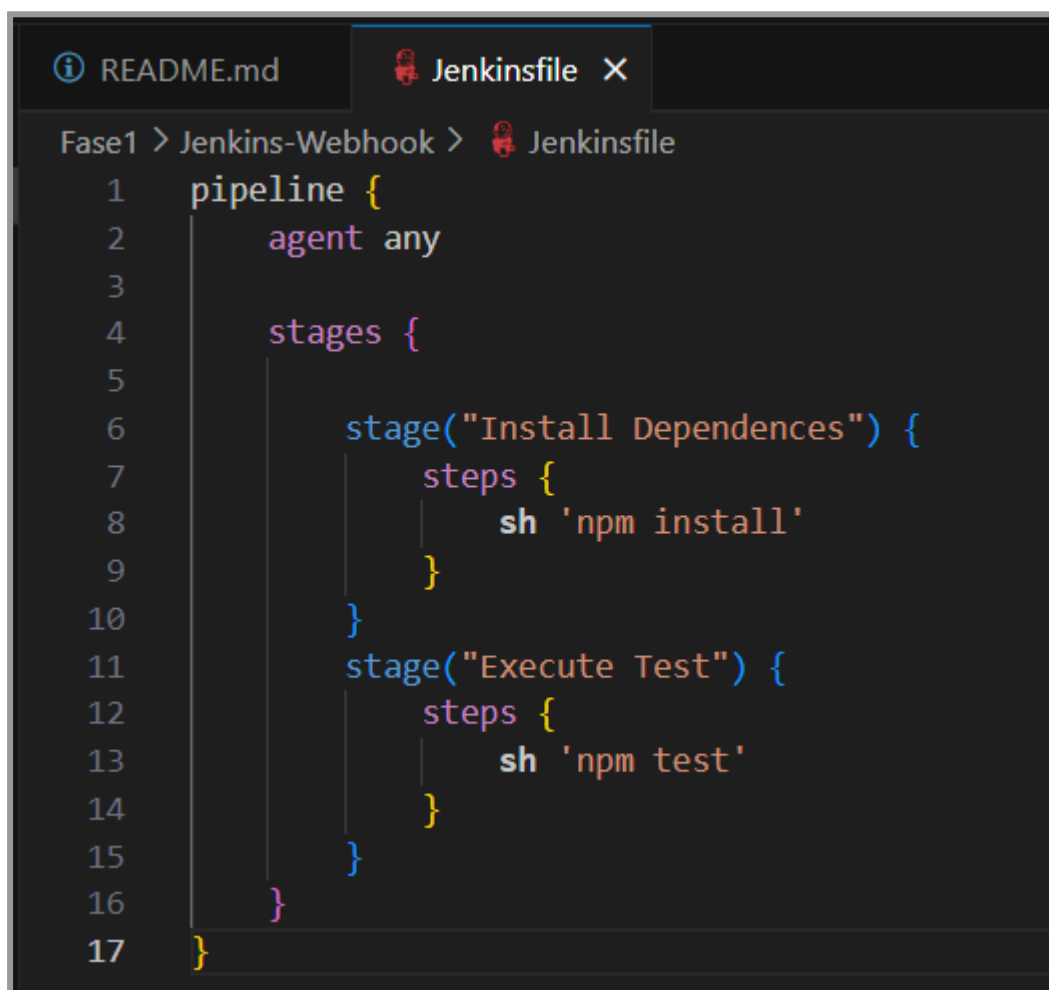
Haz clic en **"New Item"**.

Ingresa un nombre para tu pipeline y selecciona **"Pipeline"**. Luego, haz clic en "OK".

Configurar el Pipeline:

En la sección "Pipeline", selecciona "Pipeline script".

Usa el siguiente ejemplo de script de pipeline para que ejecute los pasos descritos en el README.md de tu proyecto:

A screenshot of a code editor with a dark theme. At the top, there are two tabs: 'README.md' and 'Jenkinsfile'. The 'Jenkinsfile' tab is active. Below the tabs, the breadcrumb navigation shows 'Fase1 > Jenkins-Webhook > Jenkinsfile'. The main area displays a Jenkinsfile script with line numbers 1 through 17 on the left. The script defines a pipeline with an agent 'any' and two stages: 'Install Dependencies' and 'Execute Test'. The first stage has a step 'sh' to run 'npm install'. The second stage has a step 'sh' to run 'npm test'.

```
1 pipeline {  
2     agent any  
3  
4     stages {  
5  
6         stage("Install Dependencies") {  
7             steps {  
8                 sh 'npm install'  
9             }  
10        }  
11        stage("Execute Test") {  
12            steps {  
13                sh 'npm test'  
14            }  
15        }  
16    }  
17 }
```

Guardar y Ejecutar: Guardamos el pipeline y nos aseguramos de que todo funcione correctamente.

Imágenes del correcto funcionamiento.

⚙️ Configurar

🗑️ Borrar Pipeline

🔍 Full Stage View

🌐 GitHub

✎ Rename

❓ Pipeline Syntax

📄 GitHub Hook Log

Stage View

Average stage times:
(Average full run time: ~9s)

#12
abr 09 19:33
1 commit

Declarative: Checkout SCM	Install Dependencies	Execute Test
1s	2s	2s
1s	2s	2s

Historia de tareas Tendencia ▼

🔍 Filter... /

✅ #12 9 abr 2024 19:33

✅ #11 9 abr 2024 19:24

✅ #10 9 abr 2024 16:54

📡 Atom feed Para Todos

📡 Atom feed para los errores

Enlaces permanentes

- "Última ejecución (#12) hace 3 Hor 51 Min"
- "Última ejecución estable (#12) hace 3 Hor 51 Min"
- "Última ejecución correcta (#12) hace 3 Hor 51 Min"
- "Last completed build (#12) hace 3 Hor 51 Min"

✓ Salida de consola

```
Started by GitHub push by IgnacioPeretti
Obtained Jenkinsfile from git https://github.com/IgnacioPeretti/nodejs-helloworld-api.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/desafio2
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github-jenkins-server
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/desafio2/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/IgnacioPeretti/nodejs-helloworld-api.git # timeout=10
Fetching upstream changes from https://github.com/IgnacioPeretti/nodejs-helloworld-api.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/IgnacioPeretti/nodejs-helloworld-api.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 4854f5a1537a8cff9e5774452a5087019114f4ab (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
```

```

using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/IgnacioPeretti/nodejs-helloworld-api.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 4854f5a1537a8cff9e5774452a5087019114f4ab (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 4854f5a1537a8cff9e5774452a5087019114f4ab # timeout=10
Commit message: "Update Final"
> git rev-list --no-walk 7f3bd90cc49cfa9f396ff1213f81d6a6d81c5dd5 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Install Dependencies)
[Pipeline] sh
+ npm install

up to date, audited 368 packages in 2s

45 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Execute Test)

```

```

[Pipeline] { (Execute Test)
[Pipeline] sh
+ npm test

> nodejs-helloworld-api@1.0.0 test
> jest --forceExit

console.log
  Server is listening on port 3000

    at Server.log (index.js:10:13)

PASS ./index.test.js
  GET /
    â responds with hello world message in JSON format (141 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.932 s, estimated 2 s
Ran all test suites.
Force exiting Jest: Have you considered using `--detectOpenHandles` to detect async operations that kept running after all tests finished?
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Esta página muestra los logs de polling que lanzaron esta tarea.

Mostrar como texto plano=

```
Started on Apr 9, 2024, 7:33:44 PM
Started by event from 140.82.115.43 ? https://f59e-149-87-91-129.ngrok-free.app:8080/github-webhook/ on Tue Apr 09 19:33:44 ART 2024
Using strategy: Default
[poll] Last Built Revision: Revision 7f3bd90cc49cfa9f396ff1213f81d6a6d81c5dd5 (refs/remotes/origin/main)
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github-jenkins-server
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials
> git ls-remote -h -- https://github.com/IgnacioPeretti/nodejs-helloworld-api.git # timeout=10
Found 1 remote heads on https://github.com/IgnacioPeretti/nodejs-helloworld-api.git
[poll] Latest remote head revision on refs/heads/main is: 4854f5a1537a8cff9e5774452a5087019114f4ab
Done. Took 0.76 sec
Changes found
```

Last GitHub Push

```
Started on Apr 9, 2024, 7:33:44 PM
Started by event from 140.82.115.43 ? https://f59e-149-87-91-129.ngrok-free.app:8080/github-webhook/ on Tue Apr 09 19:33:44 ART 2024
Using strategy: Default
[poll] Last Built Revision: Revision 7f3bd90cc49cfa9f396ff1213f81d6a6d81c5dd5 (refs/remotes/origin/main)
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github-jenkins-server
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials
> git ls-remote -h -- https://github.com/IgnacioPeretti/nodejs-helloworld-api.git # timeout=10
Found 1 remote heads on https://github.com/IgnacioPeretti/nodejs-helloworld-api.git
[poll] Latest remote head revision on refs/heads/main is: 4854f5a1537a8cff9e5774452a5087019114f4ab
Done. Took 0.76 sec
Changes found
```