

Requisitos:

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

Ruta al Fork <https://github.com/jcavaiuolo/nodejs-helloworld-api>

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

ver punto 4

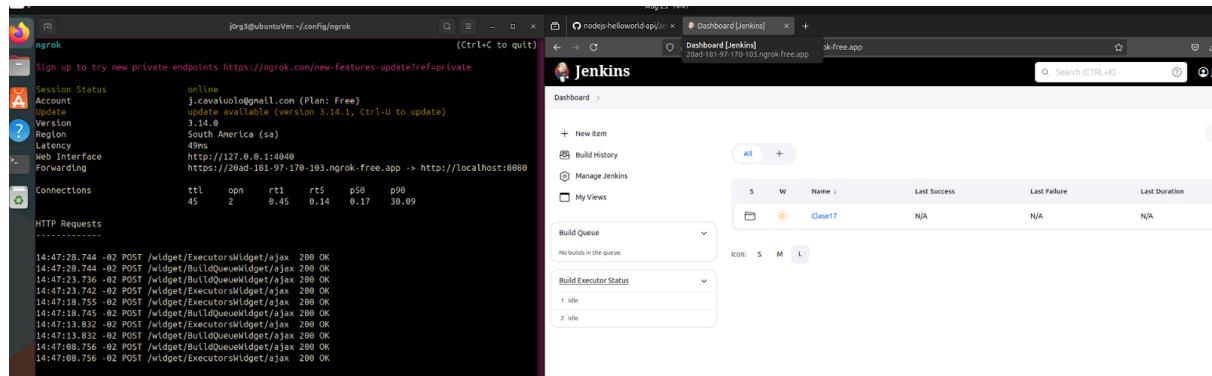
3. Exponer mediante ngrok el servicio de Jenkins.

luego de instalar ngrok creé un perfil para exponer la instancia local de jenkins:

```
j0rg3@ubuntuVm:~/Desktop/devopsBC_content$ cat ~/.config/ngrok/ngrok.yml
version: "2"
authtoken: 2l0oSkgBQzMf7EXWVZjwOMX9DUq_3E2KPtHh5KhnPu3kz68mU

tunnels:
  jenkins:
    proto: http
    addr: localhost:8080
#   basic_auth:
#     - "jorge:jenkinspass"
```

y una vez activado con “ngrok start jenkins” (siendo jenkins el nombre del perfil) se puede acceder desde el tunel.



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

CREACIÓN DEL PIPELINE

Voy a recrear el Jenkinsfile utilizando el plugin de BlueOcean para Jenkins y replicando las instrucciones del archivo README

1. Create Pipeline

https://95c0-181-97-170-103.ngrok-free.app/blue/create-pipeline

Jenkins

Pipelines Administration

Create Pipeline

✓

Where do you store your code?

Bitbucket Cloud

Bitbucket Server

GitHub

GitHub Enterprise

Git

✓

Which organization does the repository belong to?

jcavaiuolo

○

Choose a repository

Loaded 22 repositories

hell

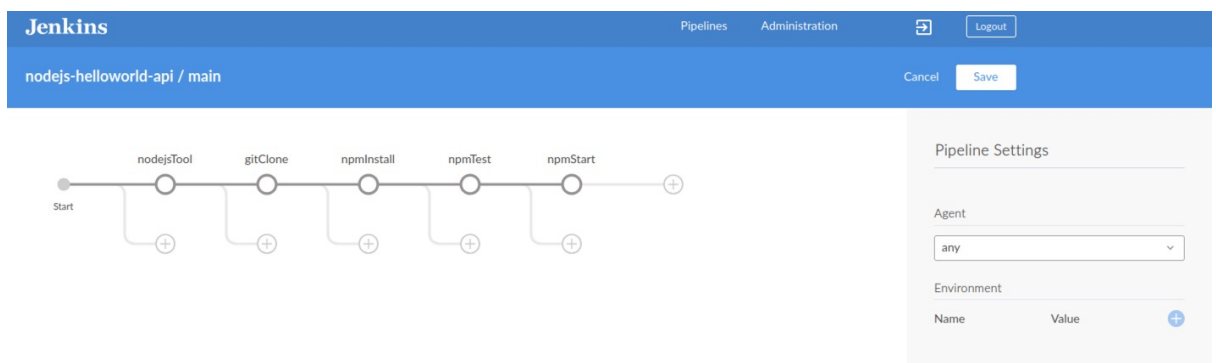
nodejs-helloworld-api

Create Pipeline

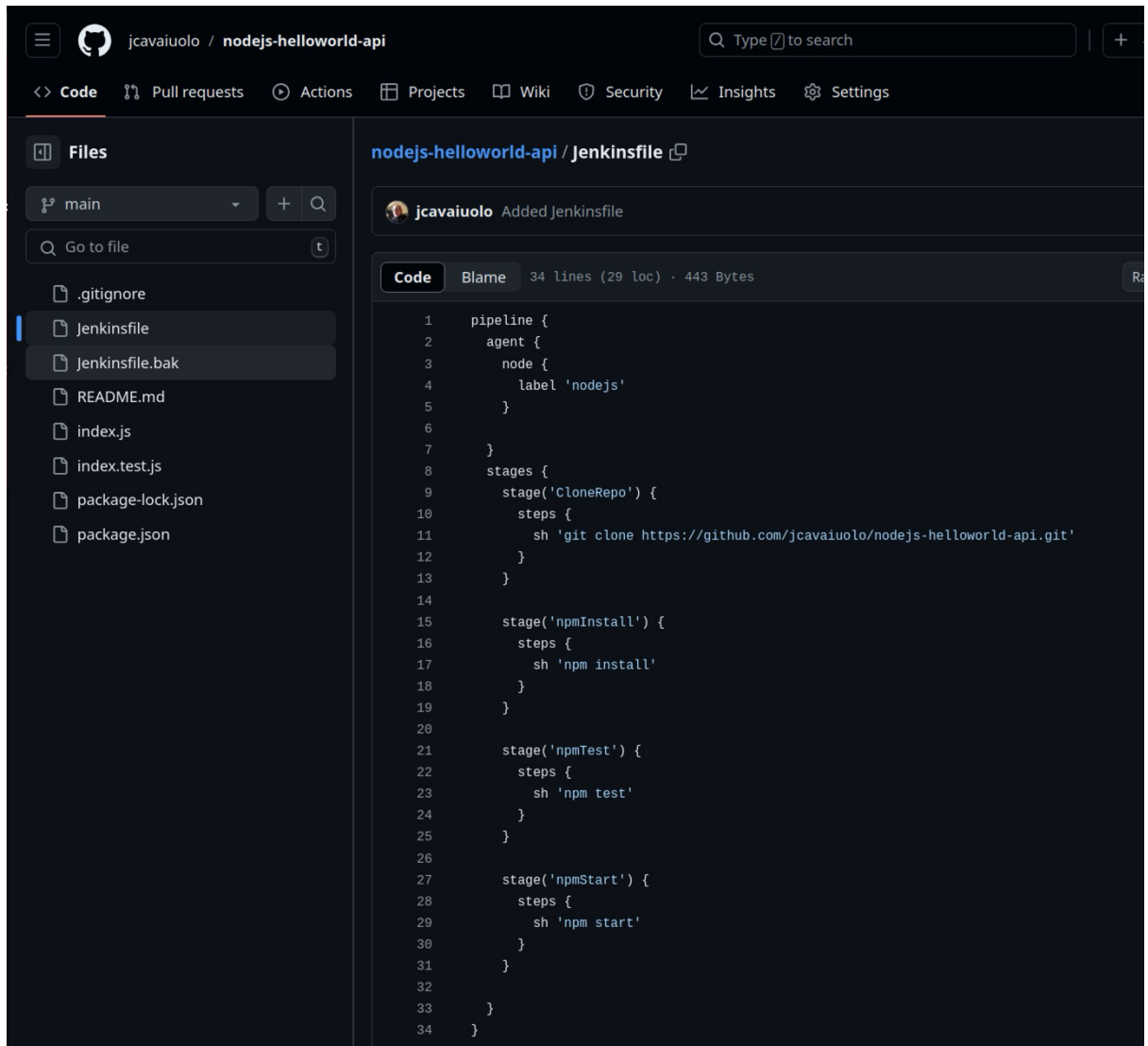
○

Complete

2. Se crea un pipeline con los pasos del Readme (git clone, npm install, npm test, npm build)



BlueOcean ha creado el Jenkinsfile en el repositorio
<https://github.com/jcavaiuolo/nodejs-helloworld-api/blob/main/Jenkinsfile>



PRUEBAS

3. Al completar la ejecución me di cuenta que se traba en el paso de “npm start”, aunque el servicio está activo y se puede completar la prueba con curl, asumo que Jenkins está esperando que finalice el proceso de “start”, al detener el pipeline el servicio deja de escuchar en el puerto 3000.

nodejs-helloworld-api < 15

Branch: main | 1m 45s | No changes | Branch indexing

Commit: cf85a57

Start nodejsTool gitClone npmInstall npmTest npmStart End

npmStart - 1m 22s

```

1 + npm start
2
3 > nodejs-helloworld-api@1.0.0 start
4 > node index.js
5
6 Server is listening on port 3000

```

- Al usar `nohup npm start &` (para ejecutar en segundo plano) el proceso finaliza de forma correcta ya que Jenkins no queda en espera pero el servicio queda fuera de línea.

nodejs-helloworld-api < 14

Branch: main | 24s | Changes by j.cavaiuolo | Branch indexing

Commit: 85a68e2

Start nodejsTool gitClone npmInstall npmTest npmStart End

npmStart - <1s

```

1 + nohup npm start

```

Entregables:

Los entregables establecidos para este proyecto son:

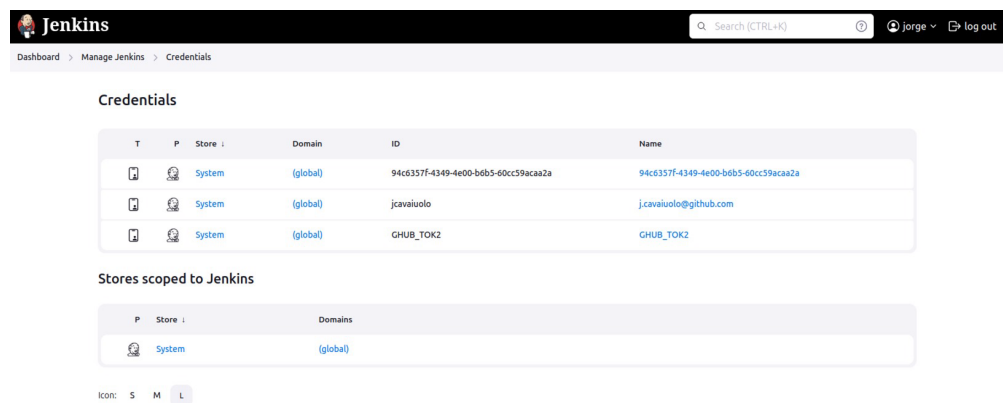
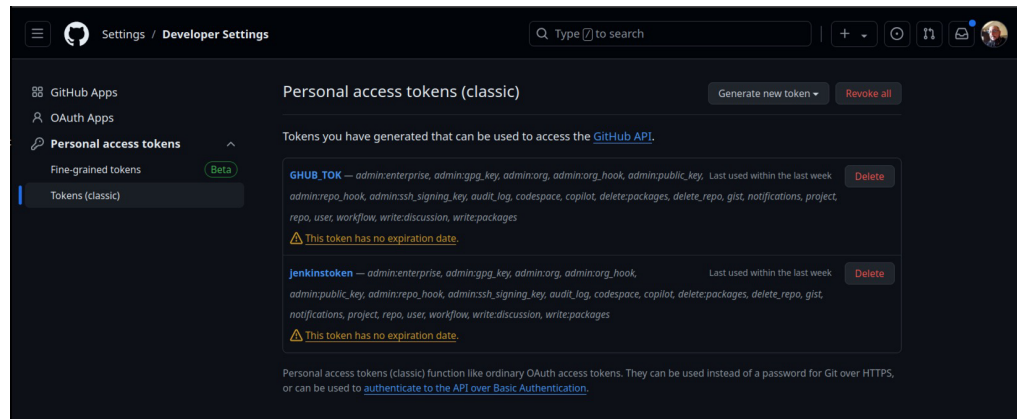
- Código fuente del pipeline de Jenkins publicado en un repositorio de Github.

<https://github.com/jcavaiuolo/nodejs-helloworld-api/blob/main/Jenkinsfile>

- Guía detallada de cómo utilizar el Job publicado en el archivo README del repositorio.

BlueOcean se encargó de mucho del setup la segunda vez, pero de entrada creé la configuración de forma manual. Los grandes pasos son:

- Instalación de Jenkins >> yo seguí las instrucciones de <https://www.jenkins.io/doc/book/installing/linux/#long-term-support-release> para instalar en Ubuntu
- Configuración del token de acceso a GitHub

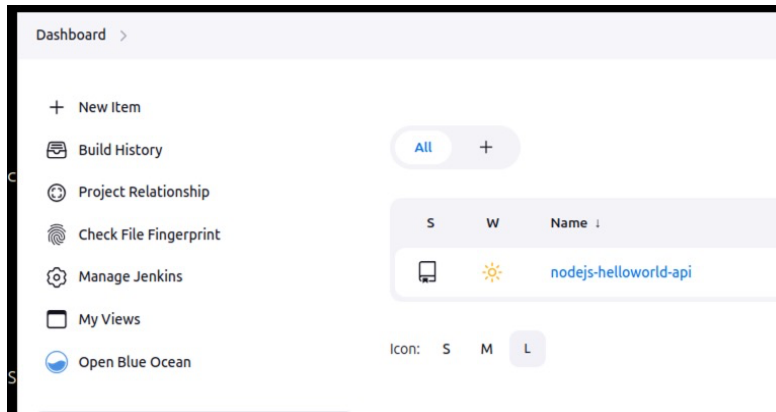


- Configuración del Pipeline (ver sección [CREACION DEL PIPELINE](#))
- Ejecución del Pipeline

Cómo Utilizar un `Jenkinsfile` desde un Repo en GitHub

Crear un Nuevo Pipeline:

- En la página principal, click en “Nuevo Item” o “New Item”.



- Dale un nombre a tu proyecto.
- Seleccioná “Pipeline” y click en “OK”.

Pipeline
 Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

- En la configuración del pipeline, en la sección “Pipeline script from SCM”, seleccioná "Git". En el campo "Credentials", click en "Add" y seleccioná "Jenkins" > "GitHub Personal Access Token". Agregar las credenciales como “secret text”

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/jcavauiolo/nodejs-helloworld-api.git

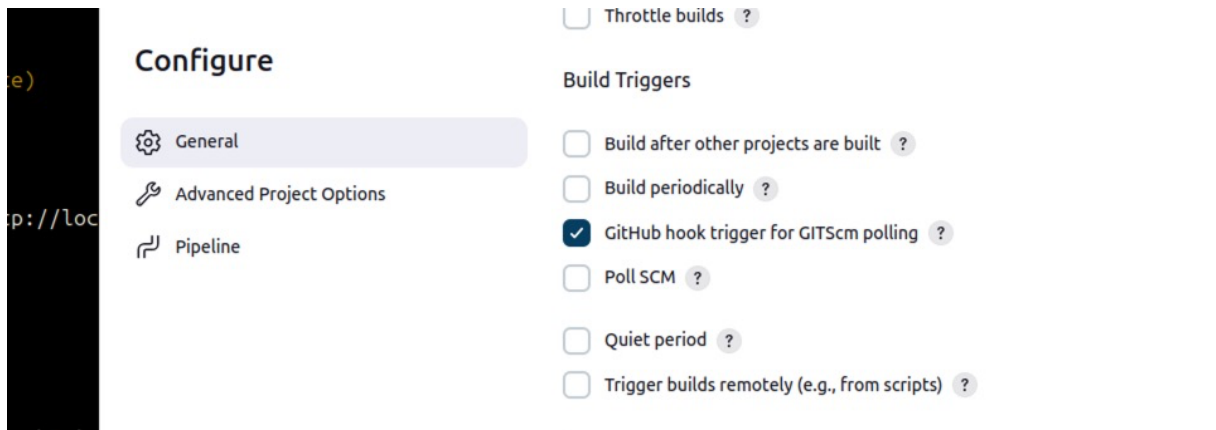
Credentials ?

94c6357f-4349-4e00-b6b5-60cc59acaa2a

+ Add

Advanced

- Para poder disparar el pipeline automaticamente se debe seleccionar la opcion “GitHub hook trigger for GITScm polling”



- En "Script Path", especifica la ruta al `Jenkinsfile`

Script Path ?

Guardar y Ejecutar

Ahora se puede ejecutar el pipeline desde la interfaz de Jenkins con "Build Now" o configurarlo para que se ejecute automáticamente cuando haya cambios en el repositorio.

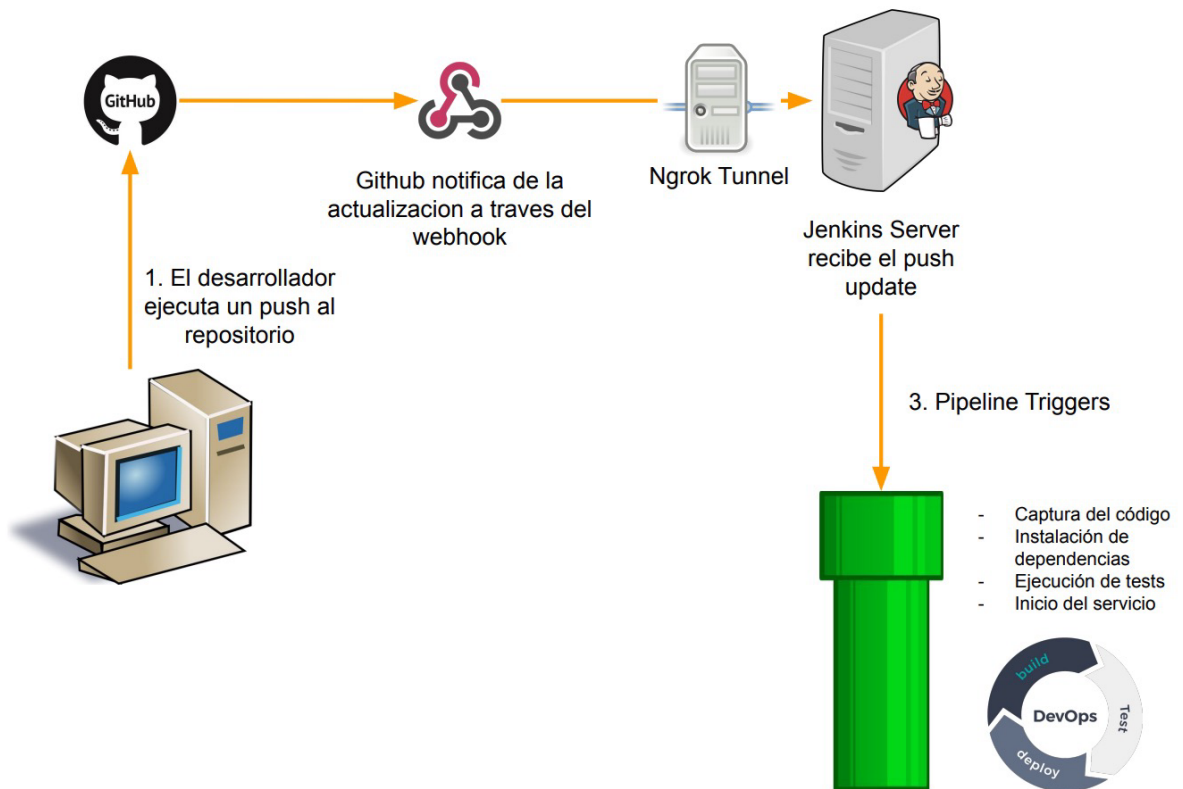
3. Evidencia de las pruebas con ejecución exitosa.

ver punto 3 y 4 de la sección [PRUEBAS](#)

Workflow general:

En general el flujo de información será el siguiente.

1. El desarrollador empuja una actualización al repositorio de código.
2. El repositorio registra la actualización y por medio de un webhook notificará al servidor de Jenkins de este hecho. (En el caso de esta implementación el servidor de Jenkins está expuesto hacia la internet con ngrok a través de un túnel.)
3. Al recibir el push notification, Jenkins ejecutará el pipeline definido para ese trabajo. (En el caso de esta implementación se capturará el código, instalarán las dependencias, se ejecutarán los tests y se iniciará el servicio)



Evaluación:

- Entrega en fecha.
 - Redactar documentación legible y que sea comprendida por terceros
 - Añada material de soporte adicional
- Ejemplo: diagrama de alto nivel
- Cumple con las consignas solicitadas.
 - El entregable es funcional.
 - Ejemplo: el script bash al ejecutarse funciona sin errores y realiza lo solicitado.