

# CI/CD en Jenkins con nodeJS y un webhook de GitHub para realizar pruebas de building automatizado

- El objetivo principal es automatizar los Pull Requests o git push de los desarrolladores para que el webhook de un repositorio ejecute un buid en Jenkins y testee un proyecto de nodeJS cada vez que ocurra alguno de los dos eventos.

## Configuración en Jenkins:

- Instalar Jenkins como controlador en el entorno que más se adapte a las necesidades del proyecto. Ver documentación de referencia.
- Corroborar que esté instalado o instalar el plugin "GitHub Branch Source".

Name ⓘ	Enabled
<a href="#">GitHub Branch Source Plugin</a> 1807.v50351eb_7dd13 Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc. <a href="#">Report an issue with this plugin</a>	<div><div></div><div></div></div>

- Crear un nuevo ítem, darle un nombre y elegir "Multibranch Pipeline".
- En la configuración del nuevo job, en "Branch Sources" agregar la URL del repositorio de GitHub con el que Jenkins se comunicará. Si tiene credenciales, agregarlas también.
- En la sección "Build Configuration" asegurarse que la ruta del Jenkinsfile coincida con la ubicación en el repositorio de GitHub.
- Importante: Permisos para Pull requests. Se debe especificar el comportamiento a la hora de descubrir los Pull Requests, sobretodo en los forks. En este caso se confía en los colaboradores del repositorio para aceptar los PR.

Behaviors

Discover branches

?

Strategy ?

Exclude branches that are also filed as PRs

Discover pull requests from origin

?

Strategy ?

The current pull request revision

Discover pull requests from forks

?

Strategy ?

The current pull request revision

Trust ?

Collaborators

- Aplicar y guardar.

## Configuraciones adicionales:

Al ejecutar una aplicación de nodeJS se debe instalar el plugin "NodeJS":

Name ⓘ	Enabled
<a href="#">NodeJS Plugin</a> 1.6.2 NodeJS Plugin executes <a href="#">NodeJS</a> script as a build step. <a href="#">Report an issue with this plugin</a>	<div><div></div><div></div></div>

- Una vez instalado dirigirse a System Configuration >> Tools >> NodeJS Installations >> Add NodeJS
- El nombre de la nueva instalación debe coincidir con el nombre ingresado en el Jenkinsfile en el bloque tools {}
- En este caso sería "nodejs21"
- Elegir la versión de nodeJS que se requiera para el entorno, en este caso se elige "NodeJS 21.7.3"

NodeJS installations

NodeJS installations ^ Edited

Add NodeJS

NodeJS

Name

nodejs21

☒ Install automatically ?

Install from nodejs.org

Version

NodeJS 21.7.3

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail

☐ Force 32bit architecture

Global npm packages to install

Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax `packageName@version`

Global npm packages refresh hours

Duration, in hours, before 2 npm cache update. Note that 0 will always update npm cache

72

Add installer

- Aplicar y guardar.

## Configuración en ngrok:

- Instalar ngrok en la plataforma donde está instalado el controlador de Jenkins. Ver instalación en la documentación de referencia.
- Exponer el servicio de Jenkins a internet para que pueda ser accedido por GitHub.

## Configuración en GitHub:

- Crear un webhook en el repositorio que se comunicará con Jenkins especificando el "Payload URL" brindado por ngrok.
- Seleccionar los eventos individuales "Pull requests" y "Pushes":

☐ Pull request reviews

Pull request review submitted, edited, or dismissed.

☒ Pull requests

Pull request assigned, auto merge disabled, auto merge enabled, closed, converted to draft, demilestoned, dequeued, edited, enqueued, labeled, locked, milestoned, opened, ready for review, reopened, review request removed, review requested, synchronized, unassigned, unlabeled, or unlocked.

☒ Pushes

Git push to a repository.

☐ Registry packages

Registry package published or updated in a repository.

# Comandos a ejecutar en cada build especificado en el Jenkinsfile:

## Instalar dependencias:

```
npm install
```

## Ejecutar las pruebas:

```
npm test
```

- En caso que falle la prueba, no se ejecutará la aplicación.

## Ejecutar la aplicación:

```
npm start
```


## Realizar un requerimiento para que devuelva el mensaje:

```
curl http://localhost:3000
```

## Parar la aplicación:

```
pkill -f "node"
```

- Con esta configuración, Jenkins ejecutará el job de manera correcta cada vez que haga un git push o un pull request.













MultiBranch-Pipeline-NodeJS-HellowWorld-API

Folder name: Webhooks/NodeJS-HelloWorld-API-Test  
simple demo helloworld api

Branches (2)

Pull Requests (1)

S	W	Name	Last Success	Last Failure	Last Duration
		branch-main-2	1 hr 19 min 	2 days 9 hr 	17 sec 
		main	1 hr 5 min 	2 days 8 hr 	16 sec 

## Referencias:

- Instalación de Jenkins: <https://www.jenkins.io/doc/book/installing/>
- Branch Source Plugin: <https://docs.cloudbees.com/docs/cloudbees-ci/latest/cloud-admin-guide/github-branch-source-plugin>
- Instalación de ngrok: <https://ngrok.com/docs/getting-started/>