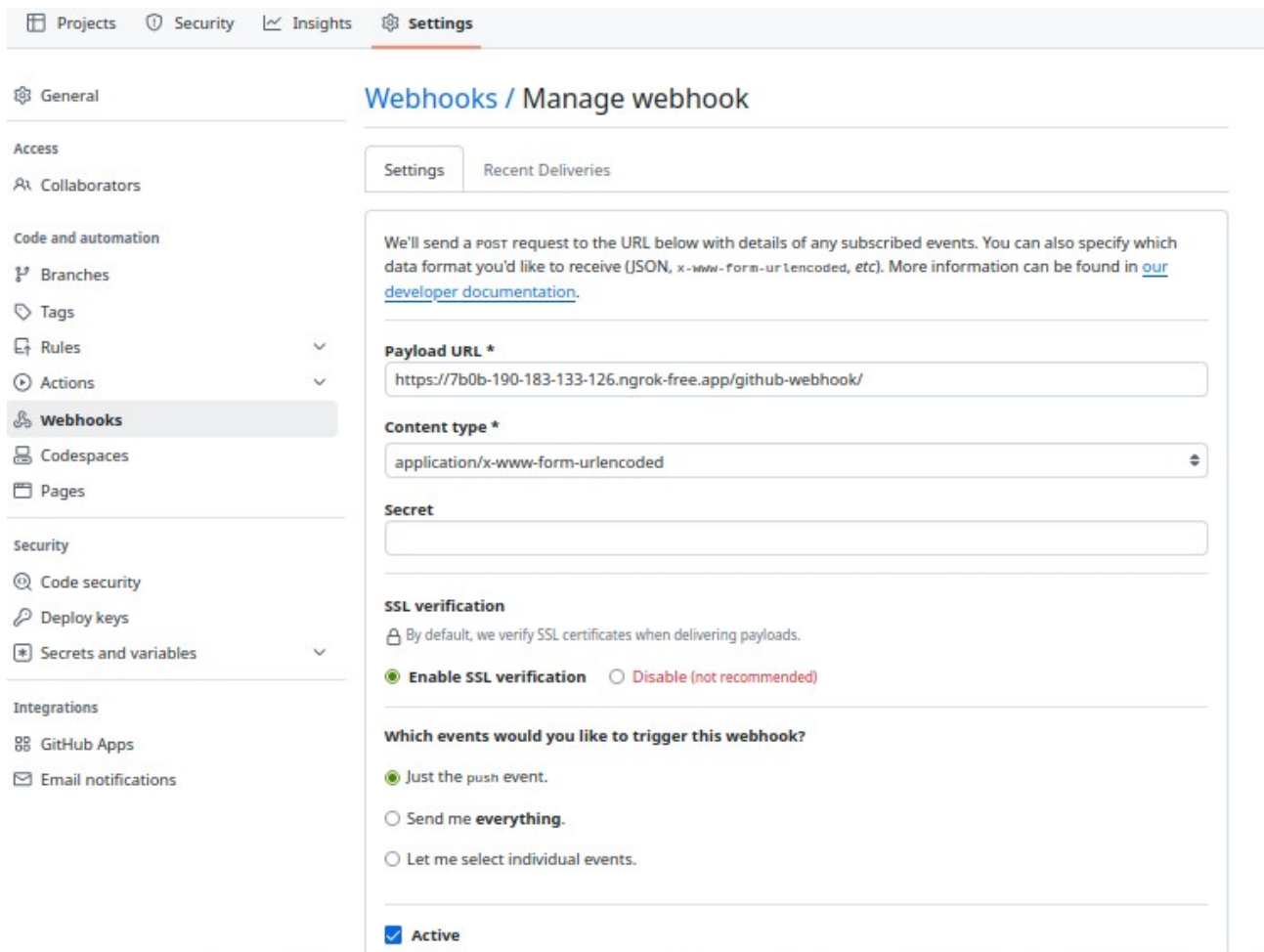


- Para cumplimentar con lo solicitado por el desafío numero dos de éste curso, se procedió de la siguiente manera: se configuro el webhook en un repositorio clonado según las instrucciones en Github el cual contiene un pipeline simple.

## Configuración de webhook

Nos ubicamos sobre repositorio en el que queremos realizar el webhook, cliqueamos en la pestaña de settings del repositorio, donde buscamos la opción de webhook, éste nos abre una especie de formulario que debemos configurar según se observa en la captura de pantalla.



The screenshot shows the GitHub repository settings page for a repository named 'Webhooks / Manage webhook'. The left sidebar contains navigation links: General, Access, Collaborators, Code and automation (Branches, Tags, Rules, Actions, Webhooks, Codespaces, Pages), Security (Code security, Deploy keys, Secrets and variables), and Integrations (GitHub Apps, Email notifications). The main content area is titled 'Webhooks / Manage webhook' and has two tabs: 'Settings' and 'Recent Deliveries'. The 'Settings' tab is active, showing a form to configure a new webhook. The form includes a text input for 'Payload URL' with the value 'https://7b0b-190-183-133-126.ngrok-free.app/github-webhook/'. Below this is a dropdown for 'Content type' set to 'application/x-www-form-urlencoded'. There is a 'Secret' field with a password icon. The 'SSL verification' section has a checkbox for 'Enable SSL verification' which is checked. The 'Which events would you like to trigger this webhook?' section has three radio buttons: 'Just the push event.' (checked), 'Send me everything.', and 'Let me select individual events.'. At the bottom, there is a checkbox for 'Active' which is checked.

## Configuración de payload url:

para ello ante cualquier duda consultamos el plugin de git en jenkins donde nos indica la manera de utilizar las url.

En la sección de manual mode

ejemplos: de url estática y dinámica.

Esta url se utiliza generalmente si tengo una base estática cuando instalamos jenkins  
`$JENKINS_BASE_URL/github-webhook/`

como utilizamos una url dinámica que nos brinda ngrok debemos construir nuestra url para que funcione webhoob:

for example: <https://ci.example.com/jenkins/github-webhook/>

ejemplo de creación url para nuestro webhook.

`https://084c-190-7-36-128.ngrok-free.app/github-webhook/`

### Manual Mode

In this mode, you'll be responsible for registering the hook URLs to GitHub. Click the  icon (under Manage Jenkins > Configure System > GitHub) to see the URL in Jenkins that receives the post-commit POSTs — but in general the URL is of the form `$JENKINS_BASE_URL/github-webhook/` — for example: `https://ci.example.com/jenkins/github-webhook/`.

Posteriormente debemos verificar que github se puede comunicar jenkins, o por lo menos llega al puerto.

## Webhooks

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

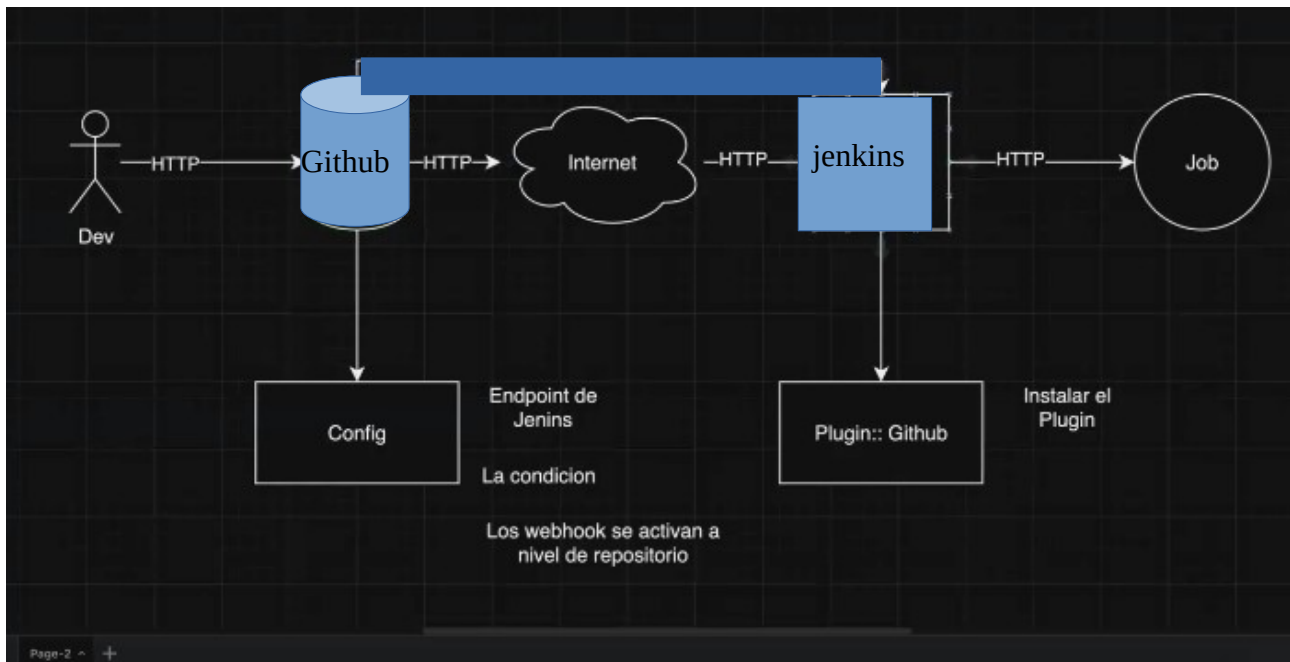
✓ `https://084c-190-7-36-128.ngrok-f...` (push)

Last delivery was successful.

Edit

Delete

Si lo queremos ver de forma gráfica podemos observar que github se puede comunicar con jenkins



## A continuación creamos nuestro pipeline declarativo

1. Debemos configurar el job informando que el plugin de github dispere a través de Bulid triggers opción Github hook trigger for GITScm polling

The screenshot shows the 'Build Triggers' section of a Jenkins configuration page. It contains several checkboxes with corresponding labels and help icons (question marks):

- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☒ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?
- ☐ Quiet period ?
- ☐ Trigger builds remotely (e.g., from scripts) ?

Below this section is the 'Advanced Project Options' section, which is currently collapsed. At the bottom of the form are two buttons: 'Save' (in a blue box) and 'Apply' (in a grey box).

2. seguidamente configuramos el pipeline con el sistema de diversionado

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

<https://github.com/hernan130/nodejs-helloworld-api.git>

Credentials ?

- none -

+ Add

Avanzado

al ser un repositorio publico no nos va solicitar credenciales, que en el caso de se repo privado debemos generar un token en github para utilizarlo como password al nombre que le demos a la credencial.

Seguidamente debemos habilitar nodejs en la configuración de (tool)

previa instalación desde plugin.

## Administrar Jenkins

Q Search settings

Building on the built-in node can be a security issue. You should set up distributed builds. See the [documentation](#).

Set up agent

Set up cloud

Dismiss

### System Configuration



#### System

Configurar variables globales y rutas.



#### Tools

Configure tools, their locations and automatic installers.



#### Plugins

Añadir, borrar, desactivar y activar plugins que extienden la funcionalidad de Jenkins.



#### Nodes

Añadir, borrar, gestionar y monitorizar los nodos sobre los que Jenkins ejecuta tareas.



#### Clouds

Add, remove, and configure cloud instances to provision agents on-demand.



#### Appearance

Configure the look and feel of Jenkins

### instalaciones de NodeJS

instalaciones de NodeJS ^

✎ Edited

Añadir NodeJS

#### NodeJS

Nombre

nodejs21

☒ Instalar automáticamente ?

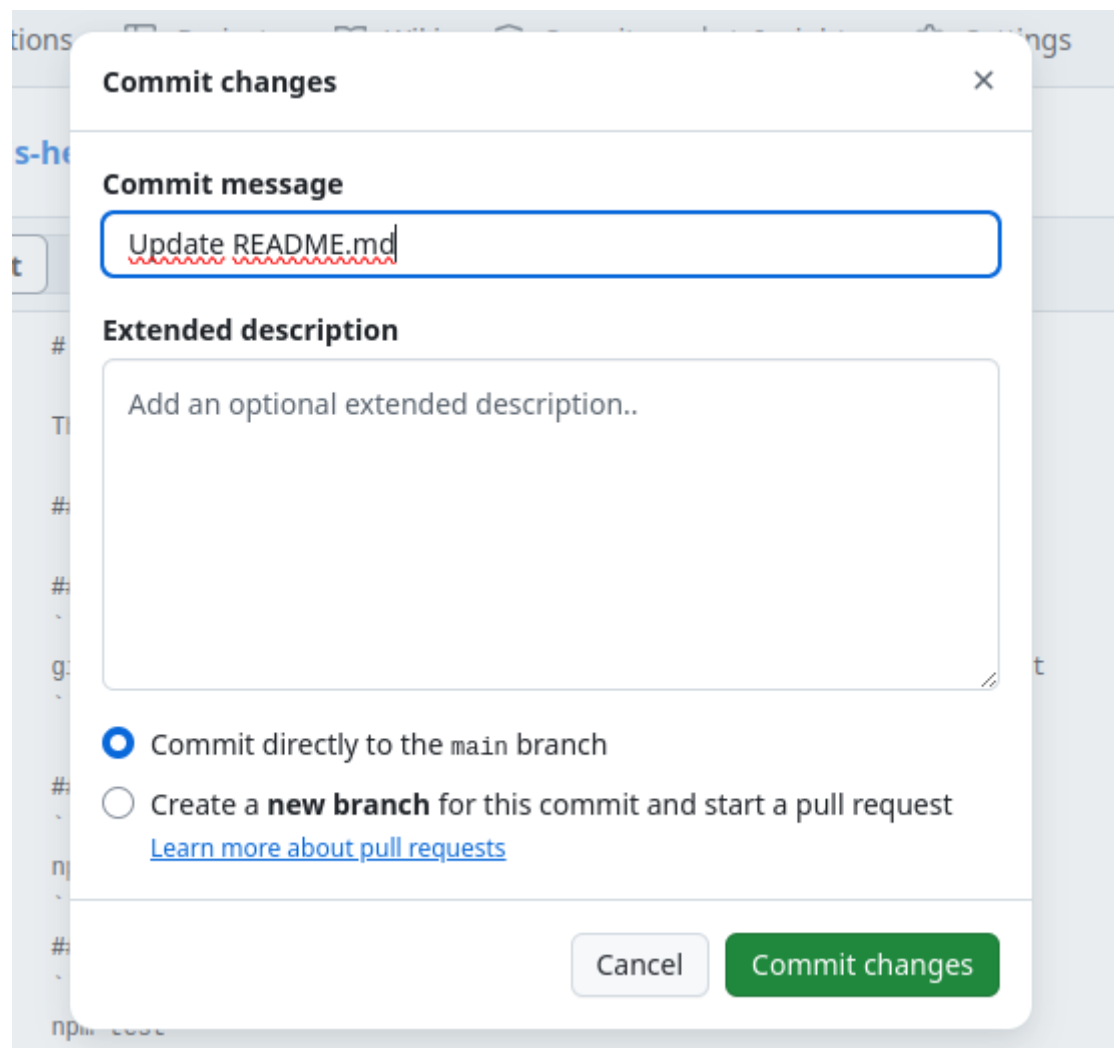
#### Install from nodejs.org

Versión

NodeJS 21.7.3

Una vez instalado debemos reiniciar jenkins.

Por ultimo comprobamos que todo funcione de manera correcta, realizamos una modificación ejemplo el (README.md) y guardamos los cambios en github, obteniendo como resultado lo que se observa en la captura de pantalla.



The image shows a screenshot of the GitHub 'Commit changes' dialog box. The dialog has a title bar with a close button (X). It contains the following elements:

- Commit message:** A text input field containing the text 'Update README.md'. The text is underlined with a red squiggly line, indicating a warning or error.
- Extended description:** A large text area with the placeholder text 'Add an optional extended description..'. There is a small icon in the bottom right corner of the text area.
- Commit options:** Two radio buttons are present:
  - The first option is selected (indicated by a blue dot) and is labeled 'Commit directly to the main branch'.
  - The second option is unselected and is labeled 'Create a **new branch** for this commit and start a pull request'. Below this label is a blue link that says 'Learn more about pull requests'.
- Buttons:** At the bottom right, there are two buttons: a light gray 'Cancel' button and a green 'Commit changes' button.



3a63f2b4-ae70-11ef-99b4-a35685f7ddb2

push

Request

Response

200

Redeliver

## Headers

**Request URL:** https://084c-190-7-36-128.ngrok-free.app/github-webhook/

**Request method:** POST

**Accept:** \*/\*

**Content-Type:** application/x-www-form-urlencoded

**User-Agent:** GitHub-Hookshot/922ffd2

**X-GitHub-Delivery:** 3a63f2b4-ae70-11ef-99b4-a35685f7ddb2

**X-GitHub-Event:** push

**X-GitHub-Hook-ID:** 515944793

**X-GitHub-Hook-Installation-Target-ID:** 895799733

**X-GitHub-Hook-Installation-Target-Type:** repository

## Payload

```
{
  "ref": "refs/heads/main",
  "before": "062af4ac4313011b6b7ee4eda16113f3793da838",
  "after": "c2834aef047acd5bef630398d6442ab99f9c413a",
  "repository": {
    "id": 895799733,
    "node_id": "R_kgDONWTRtQ",
    "name": "nodejs-helloworld-api",
    "full_name": "hernan130/nodejs-helloworld-api",
```

De ésta manera, se cumplimento con lo solicitado en el desafío numero dos de este curso.

The screenshot shows the Jenkins web interface. At the top, there's a header with the Jenkins logo, a search bar, and user information (hernan Acosta). Below the header, the breadcrumb trail reads "Panel de Control > jenkinsfile > #3". On the left sidebar, the "Console Output" tab is selected. The main area displays the console output for a pipeline build, titled "Salida de consola" with a green checkmark icon. The output text is as follows:

```
Started by GitHub push by hernan130
Obtained Jenkinsfile from git https://github.com/hernan130/nodejs-helloworld-api.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/jenkinsfile
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/jenkinsfile/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/hernan130/nodejs-helloworld-api.git # timeout=10
Fetching upstream changes from https://github.com/hernan130/nodejs-helloworld-api.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/hernan130/nodejs-helloworld-api.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision c2834aef047acd5bef630398d6442ab99f9c413a (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f c2834aef047acd5bef630398d6442ab99f9c413a # timeout=10
Commit message: "Update README.md"
> git rev-list --no-walk 062af4ac4313011b6b7ee4eda16113f3793da838 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
```