

Paso 1: Configurar las Máquinas Virtuales

VM1: Jenkins y Git

Actualizar los paquetes y sistema operativo:

```
sudo apt update  
sudo apt upgrade -y
```

Instalar Git:

```
sudo apt install git -y
```

VM2: Apache web server y OpenSSH

Actualizar los paquetes y sistema operativo:

```
sudo apt update  
sudo apt upgrade -y
```

Instalar Apache y OpenSSH en VM1 y VM2:

```
sudo apt install apache2 openssh-server -y  
sudo systemctl start ssh  
sudo systemctl enable ssh
```

Paso 2: Instalar Jenkins en VM1

Agregar la clave del repositorio Jenkins:

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo  
apt-key add -
```

```
asurrao@ubuntu-server-template:~$ wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -  
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).  
OK
```

Agregar el repositorio Jenkins al archivo sources.list:

```
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ >  
/etc/apt/sources.list.d/jenkins.list'
```

```
asurrao@ubuntu-server-template:~$ sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
```

Actualizar la lista de paquetes y luego instalar Jenkins:

```
sudo apt update  
sudo apt install jenkins -y
```

Iniciar el servicio Jenkins:

```
sudo systemctl start jenkins  
sudo systemctl enable jenkins
```

Verificar el estado de Jenkins:

```
sudo systemctl status jenkins
```

Paso 3: Exponer Jenkins con Ngrok

Instalar Ngrok:

```
curl -s https://ngrok-agent.s3.amazonaws.com/ngrok.asc | sudo tee /etc/apt/trusted.gpg.d/ngrok.asc >/dev/null && echo "deb https://ngrok-agent.s3.amazonaws.com buster main" | sudo tee /etc/apt/sources.list.d/ngrok.list && sudo apt update && sudo apt install ngrok
```

Autenticar Ngrok (reemplaza `<YOUR_NGROK_AUTHTOKEN>` con tu token de autenticación):

```
ngrok config add-authtoken <YOUR_NGROK_AUTHTOKEN>
```

Exponer Jenkins con Ngrok:

```
ngrok http 8080
```

```
ngrok
Found a bug? Let us know: https://github.com/ngrok/ngrok

Session Status      online
Account             Casper (Plan: Free)
Version             3.14.0
Region              South America (sa)
Web Interface       http://127.0.0.1:4040
Forwarding           https://7f73-2800-a4-1458-9800-a00-27ff-fe7d-24bc.ngrok-free.app -> http://localhost:8080

Connections         ttl    opn    rt1    rt5    p50    p90
0                  0      0.00   0.00   0.00   0.00
```

1. Anota la URL proporcionada por Ngrok

```
Forwarding           https://7f73-2800-a4-1458-9800-a00-27ff-fe7d-24bc.ngrok-free.app -> http://localhost:8080
```

Paso 4: Acceder a Jenkins

1. Abrir un navegador y navegar a la URL de Ngrok.

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (**not sure where to find it?**) and this file on the server:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

2. Desbloquear Jenkins y crear un usuario administrador.

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Getting Started

Bienvenido a Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Paso 5: Generar y Configurar Llaves SSH

Generar una clave SSH en VM1:

```
ssh-keygen -t rsa -b 4096
```

Copiar la clave pública a VM2:

```
ssh-copy-id user@vm2-ip
```

Paso 6: Configurar Credenciales Globales en Jenkins

1. Ir a **Manage Jenkins > Manage Credentials > (global) > Add Credentials**.
2. Agregar las credenciales SSH necesarias para la conexión con VM2.

Paso 7: Configurar un Agent en Jenkins

1. Ir a **Manage Jenkins > Manage Nodes and Clouds > New Node**.
2. Configurar un nuevo nodo usando la conexión SSH a VM2.

Nodos

+ New Node

Configure Monitors



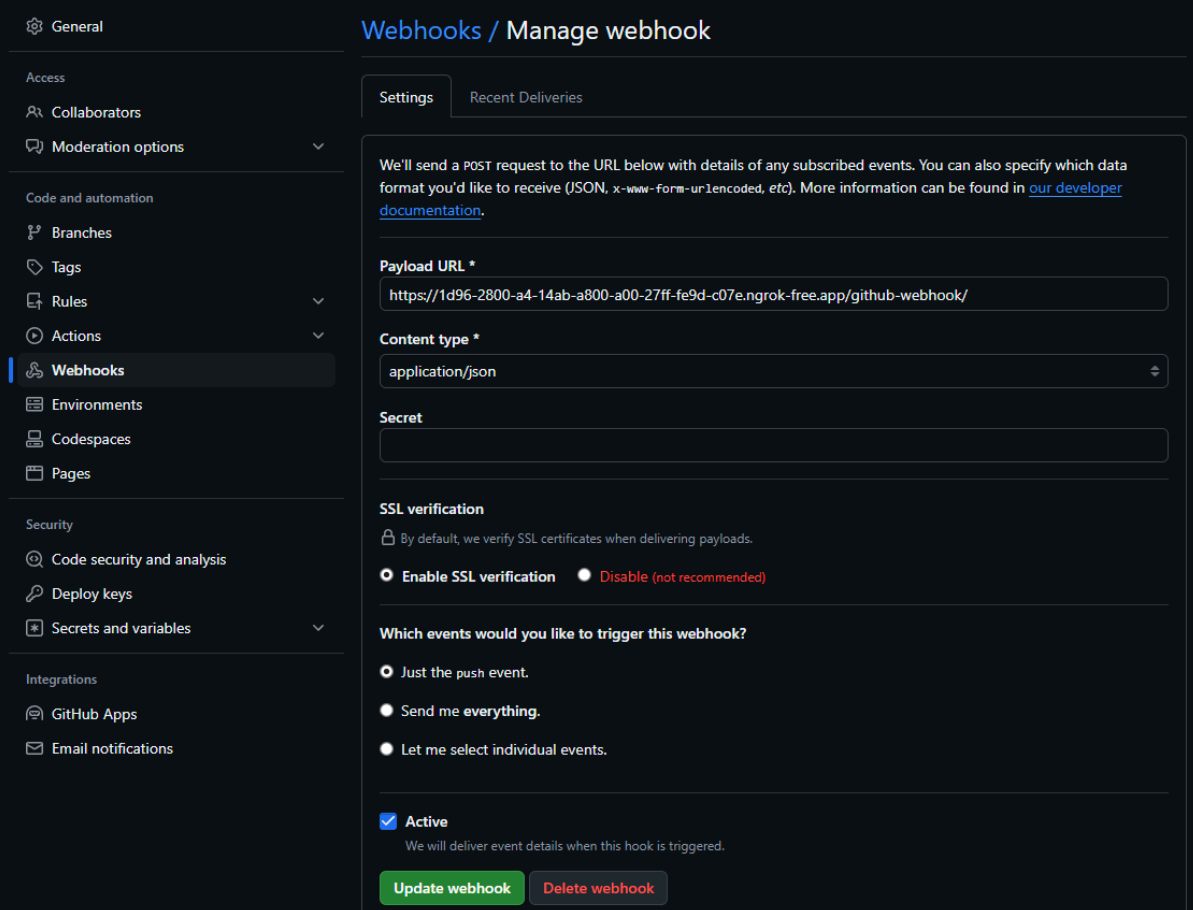
S	Nombre ↓	Arquitectura	Diferencia entre los relojes	Espacio de disco libre	Espacio de intercambio libre	Espacio temporal libre	Tiempo de respuesta
	principal	Linux (amd64)	Sincronizados	20,06 GiB	4,00 GiB	20,06 GiB	0ms
	Slave-Jenkins	Linux (amd64)	Sincronizados	20,29 GiB	4,00 GiB	20,29 GiB	20ms
Data obtained		46 Min	46 Min	46 Min	46 Min	46 Min	46 Min

Paso 8: Crear un Repositorio en GitHub

1. Crear un nuevo repositorio en GitHub.
2. Clonarlo en tu máquina local

Paso 9: Configurar el Webhook en GitHub

1. Ir a **Settings** > en tu repositorio de GitHub.
2. Agregar un webhook que se dispare en eventos de **push**.
 - **Payload URL:** URL de Jenkins proporcionada por Ngrok.
 - **Content type:** **application/json**.
 - **Which events would you like to trigger this webhook?:** **Just the push event**.



The screenshot shows the GitHub 'Webhooks / Manage webhook' settings page. The left sidebar contains navigation links for General, Access (Collaborators, Moderation options), Code and automation (Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces, Pages), Security (Code security and analysis, Deploy keys, Secrets and variables), and Integrations (GitHub Apps, Email notifications). The main content area has two tabs: 'Settings' (active) and 'Recent Deliveries'. The 'Settings' tab contains the following fields and options:

- Settings** / Recent Deliveries
- Introductory text: "We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#)."
- Payload URL ***: `https://1d96-2800-a4-14ab-a800-a00-27ff-fe9d-c07e.ngrok-free.app/github-webhook/`
- Content type ***: `application/json`
- Secret**: (Empty text field)
- SSL verification**:
 - By default, we verify SSL certificates when delivering payloads.
 - ☒ Enable SSL verification ☐ Disable (not recommended)
- Which events would you like to trigger this webhook?**
 - ☒ Just the push event.
 - ☐ Send me everything.
 - ☐ Let me select individual events.
- ☒ **Active**
We will deliver event details when this hook is triggered.
- Update webhook** (green button) **Delete webhook** (red button)

Paso 10: Crear un Jenkinsfile en el Repositorio

Crear un archivo llamado **Jenkinsfile** en tu repositorio local:

```
pipeline {
  agent { label 'Slave-Jenkins' }
  stages {
    stage('Check Apache') {
      steps {
        sh 'systemctl is-active apache2 || (echo "Apache is not running" && exit 1)'
      }
    }
    stage('Deploy to Apache') {
      steps {
        sh 'echo "<h1>Hello from Jenkins Pipeline</h1>" | sudo tee /var/www/html/index.html'
      }
    }
  }
}
```

Paso 11: Hacer Commit y Push del Jenkinsfile

Realizar el commit y push del Jenkinsfile:

```
git add Jenkinsfile
git commit -m "Add Jenkinsfile"
git push origin master
```

Paso 12: Realizar un Commit en GitHub

1. Modificar un archivo en el repositorio local (por ejemplo, **Jenkinsfile**).

Hacer commit y push de los cambios a GitHub.

```
git add Jenkinsfile
git commit -m "Update Jenkins"
git push origin master
```

Paso 13: Verificar el Despliegue en Apache

1. Jenkins debería detectar el commit y ejecutar el pipeline.
2. Verificar que el contenido se haya desplegado correctamente en el servidor Apache navegando a la dirección IP de VM2.

```
asurraco@Slave-Jenkins:/var/www/html$ cat index.html
<h1>Hello from Jenkins Pipeline 31/08/2024 22:46</h1>
```