

Entregable 6

En esta actividad práctica, se aprenderá a automatizar el proceso de despliegue de aplicaciones utilizando Jenkins y GitHub. La actividad está diseñada para proporcionar una primera experiencia práctica en la configuración de Jenkins, la exposición de Jenkins a través de Ngrok, la configuración de conexiones SSH entre máquinas virtuales, y la creación de un pipeline de Jenkins que despliegue automáticamente cambios a un servidor Apache. Esta actividad simula un entorno real de DevOps, donde la integración continua y la entrega continua (CI/CD) son fundamentales para la eficiencia y la fiabilidad del desarrollo de software.

Una vez instalado java y Jenkins en la maquina 1(Servidor) se procede a instalar Ngrok

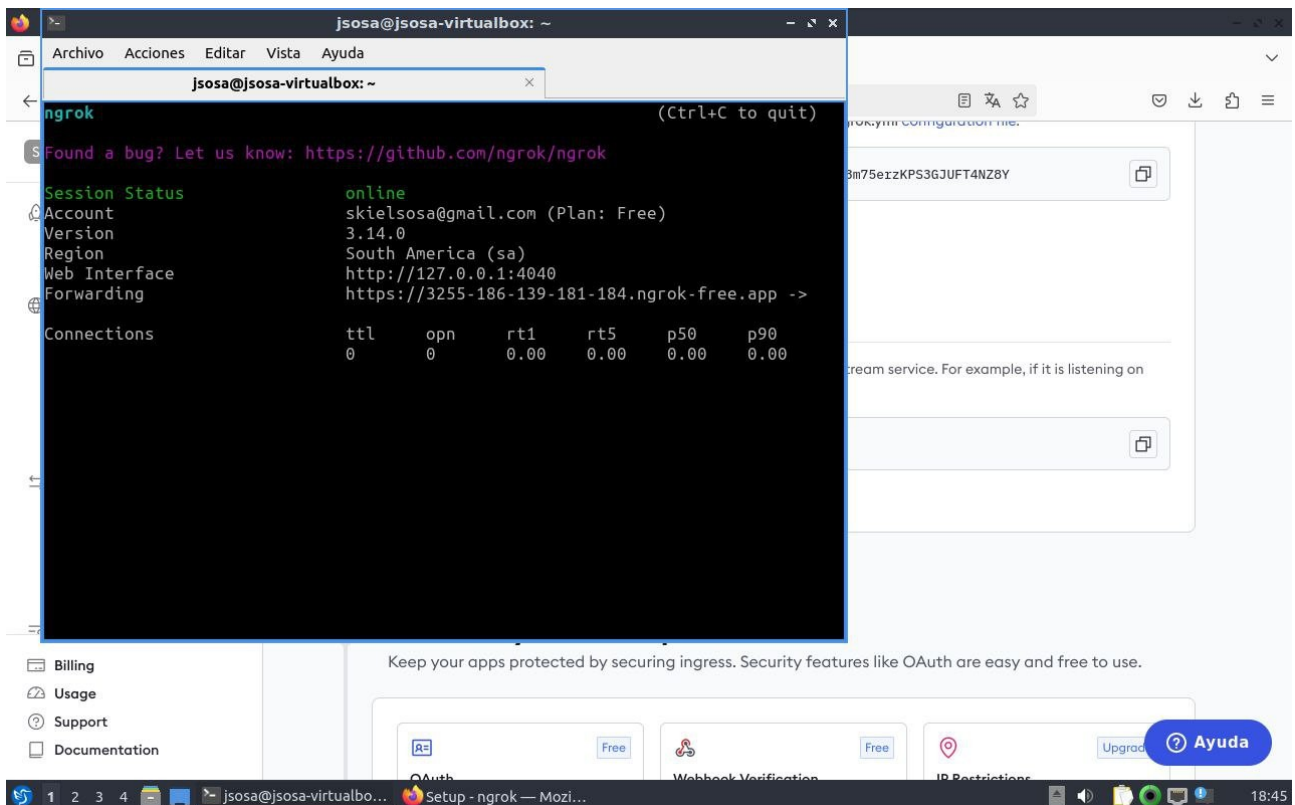
```
curl -sSL https://ngrok-agent.s3.amazonaws.com/ngrok.asc | sudo tee  
/etc/apt/trusted.gpg.d/ngrok.asc >/dev/null && echo "deb https://ngrok-  
agent.s3.amazonaws.com buster main" | sudo tee /etc/apt/sources.list.d/ngrok.list && sudo apt  
update && sudo apt install ngrok
```

Configuramos el token de Ngrok:

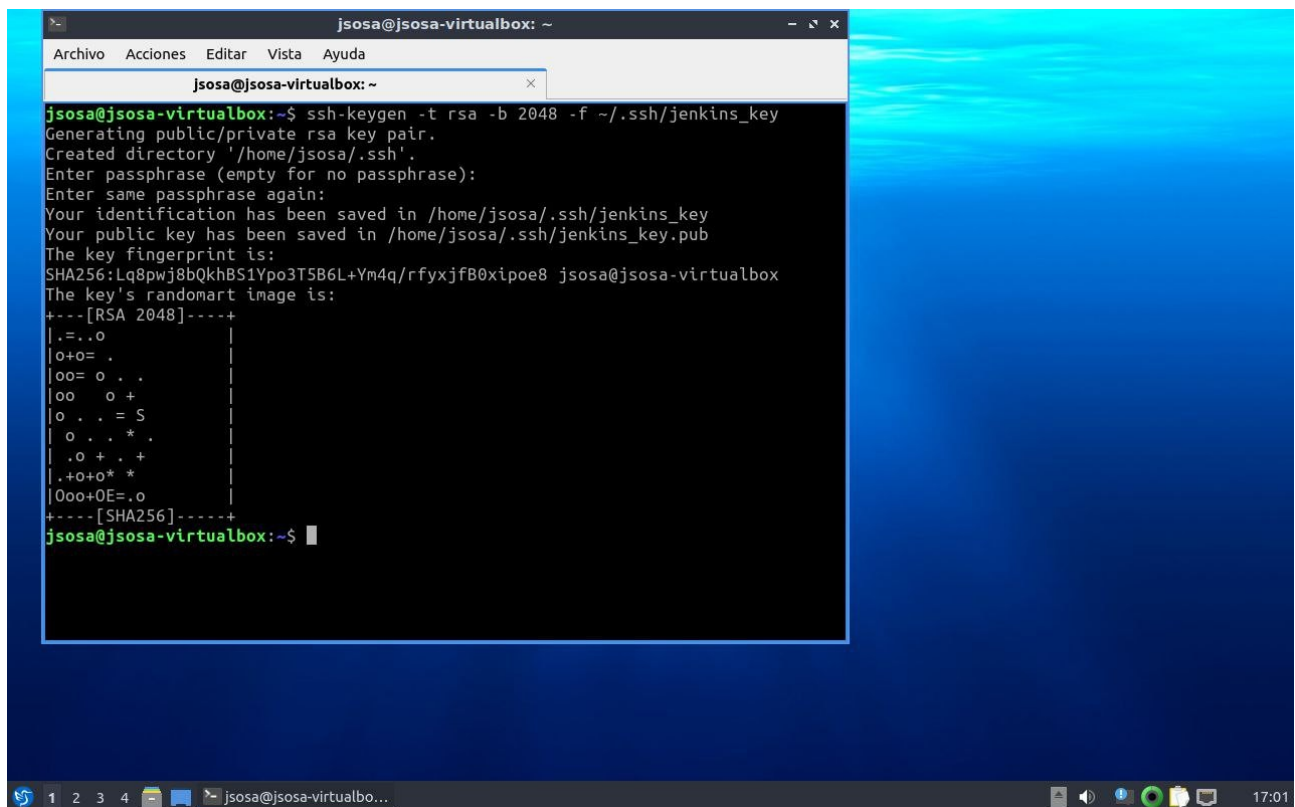
```
ngrok config add-authtoken <token>
```

y corremos Ngrok:

```
ngrok http 80
```



Creamos un par de llaves publicas/privadas



```
jsosa@jsosa-virtualbox: ~  
Archivo Acciones Editar Vista Ayuda  
jsosa@jsosa-virtualbox: ~  
jsosa@jsosa-virtualbox:~$ ssh-keygen -t rsa -b 2048 -f ~/.ssh/jenkins_key  
Generating public/private rsa key pair.  
Created directory '/home/jsosa/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/jsosa/.ssh/jenkins_key  
Your public key has been saved in /home/jsosa/.ssh/jenkins_key.pub  
The key fingerprint is:  
SHA256:Lq8pwj8bQkhBS1Ypo3T5B6L+Ym4q/rfyxjfB0xipoe8 jsosa@jsosa-virtualbox  
The key's randomart image is:  
+---[RSA 2048]-----+  
|.=.o  
|o+= .  
|oo= o . .  
|oo o +  
|o . . = S  
|o . . * .  
|.o + . +  
|.o+o* *  
|Ooo+OE=.o  
+---[SHA256]-----+  
jsosa@jsosa-virtualbox:~$
```

Y con el comando scp lo pasamos a la maquina 2 (servidor Apache)

```
jsosa@jsosa-virtualbox:~$ sudo scp ~/.ssh/jenkins_key.pub jsosa@192.168.0.151  
:/home/jsosa/.ssh/jenkins_key.pub  
jsosa@192.168.0.151's password:  
jenkins_key.pub 100% 404 109.7KB/s 00:00
```

Procedemos a configurar las credenciales globales en Jenkins

Panel de control > Administrar Jenkins > Credentials > System > Global credentials

Update credentials [Jenkins] — Mozilla Firefox

localhost:8080/manage/credentials/store/system/domain/_/credential/0b96a2df-f9a3-49c3-92fa-77684775fbcc

Panel de Control > Administrar Jenkins > Credentials > System > Global credentials (unrestricted) > jsosa (Credenciales para MV2 Bootcamp-Devops)

Delete

Move

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

ID ?

0b96a2df-f9a3-49c3-92fa-77684775fbcc

Description ?

Credenciales para MV2 Bootcamp-Devops

Username

jsosa

☐ Treat username as secret ?

Private Key

☒ Enter directly

Key

Concealed for Confidentiality

Replace

Save

System » Global credentials (unrestricted) [Jenkins] — Mozilla Firefox

localhost:8080/manage/credentials/store/system/domain/_/

Jenkins

Buscar (CTRL+K)

jonathan



Desconectar

Panel de Control > Administrar Jenkins > Credentials > System > Global credentials (unrestricted) >

Global credentials (unrestricted)

+ Add Credentials

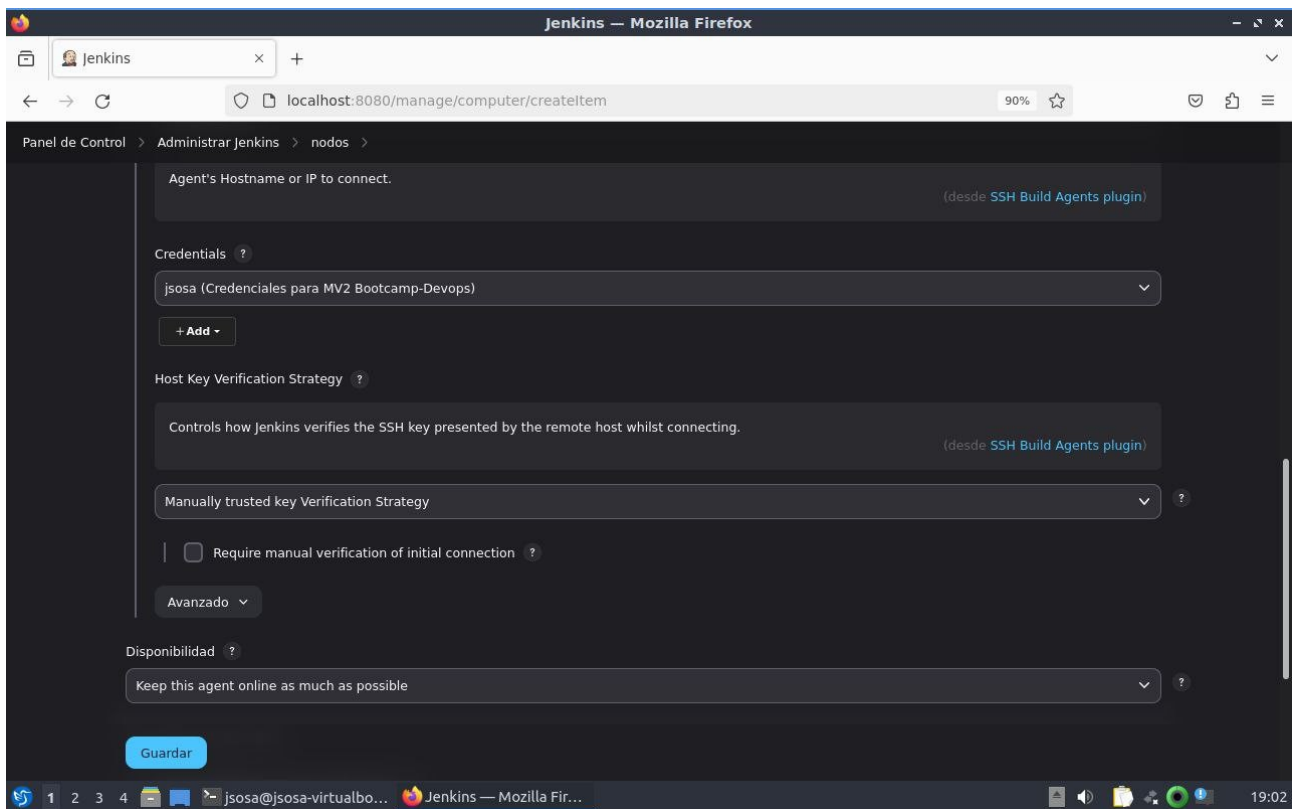
Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
 0b96a2df-f9a3-49c3-92fa-77684775fbcc	jsosa (Credenciales para MV2 Bootcamp-Devops)	SSH Username with private key	Credenciales para MV2 Bootcamp-Devops 

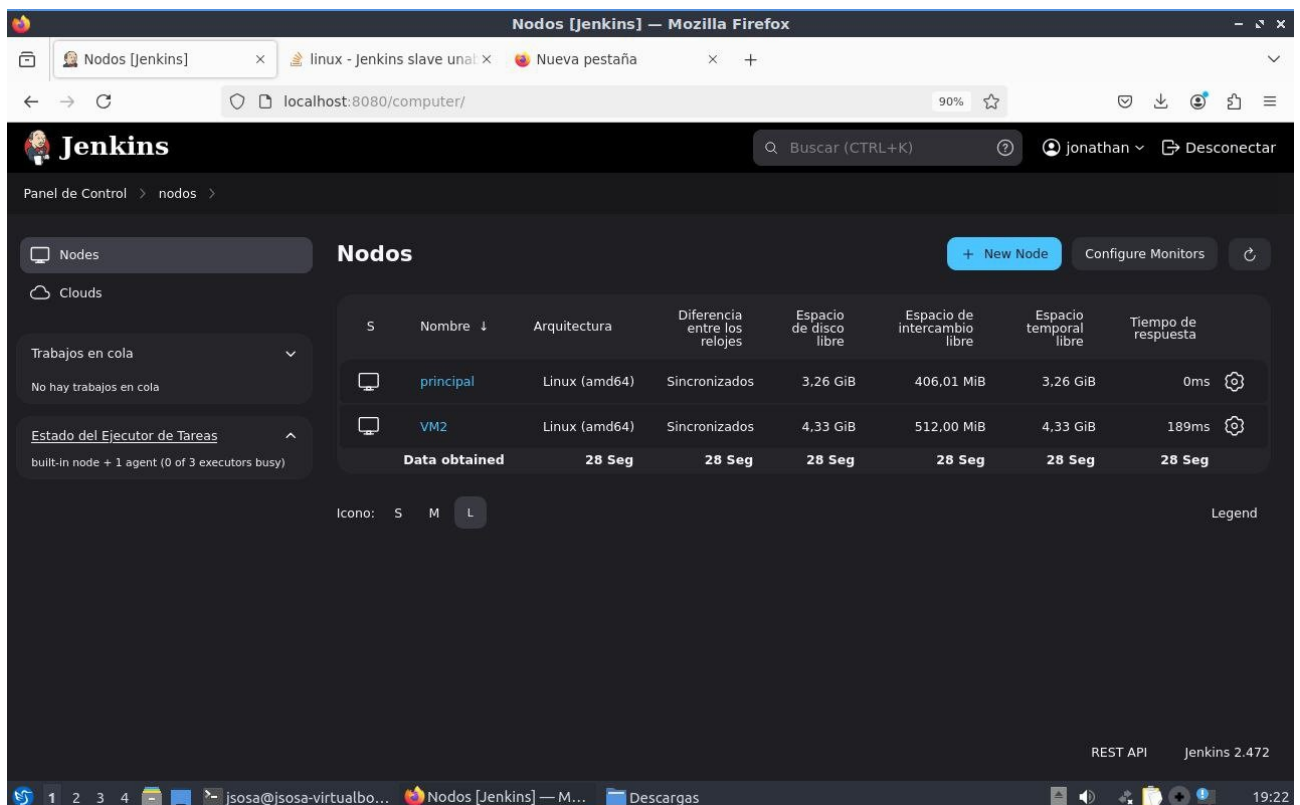
Icono: S M L

REST API Jenkins 2.470

Configuramos un Agent nuevo

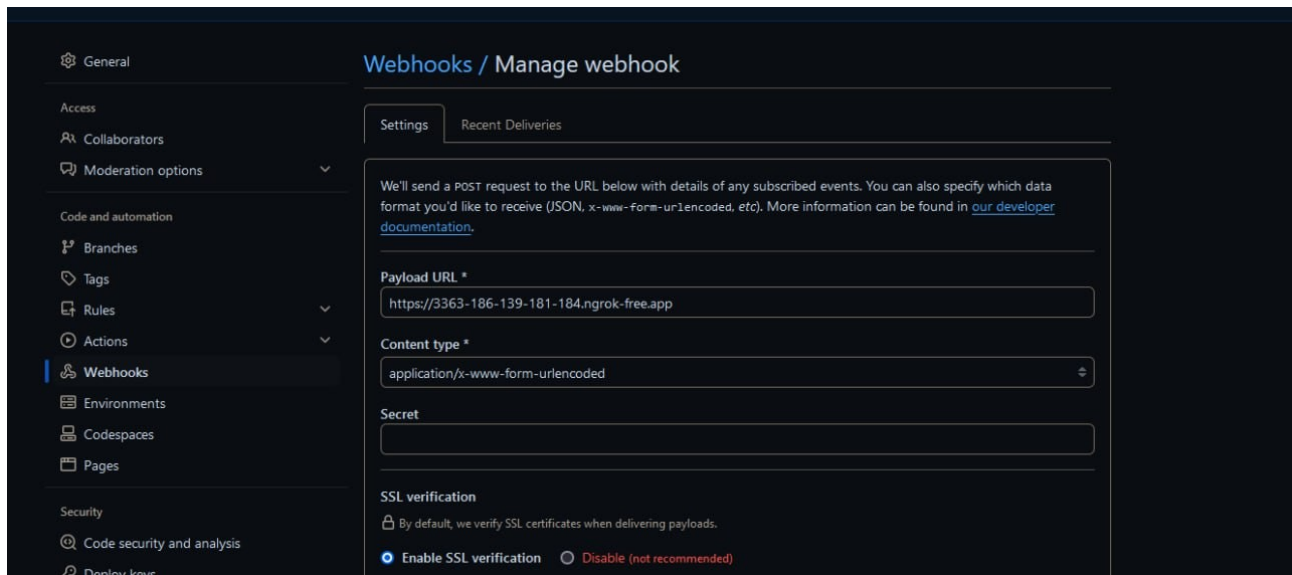


Al finalizar deberiamos ver 2 nodos, el principal y el Agente

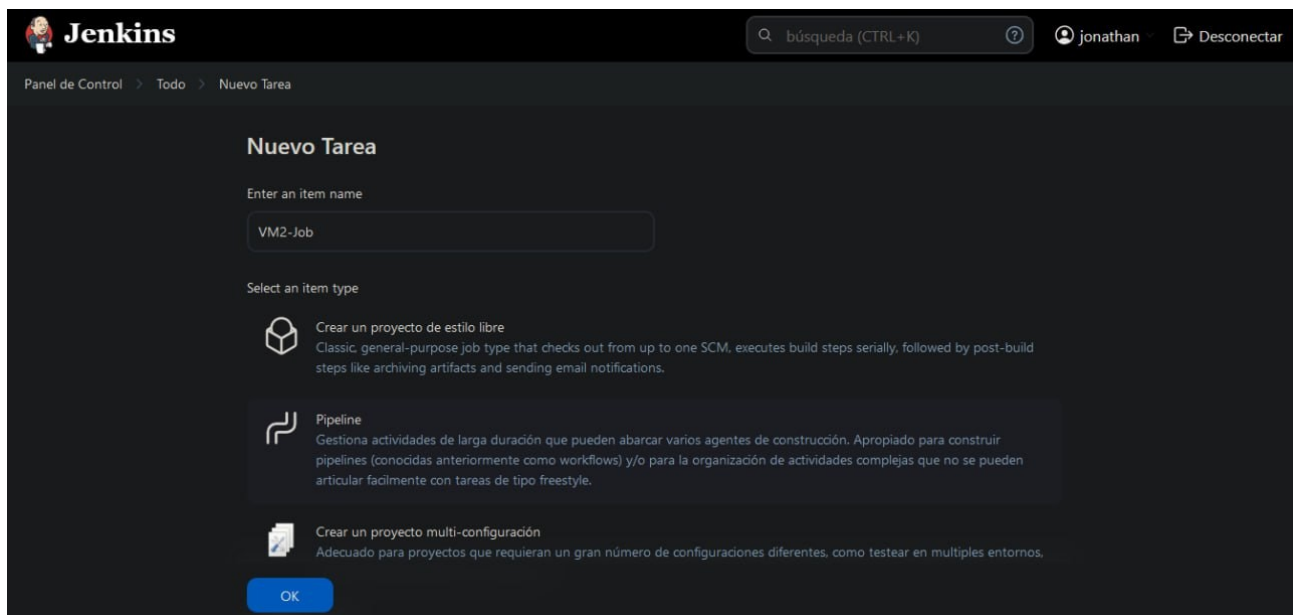


Configurar webhooks en GitHub para integrar con Jenkins. En el payload URL debemos poner la direccion que nos otorgo Ngrok + /github-webhook

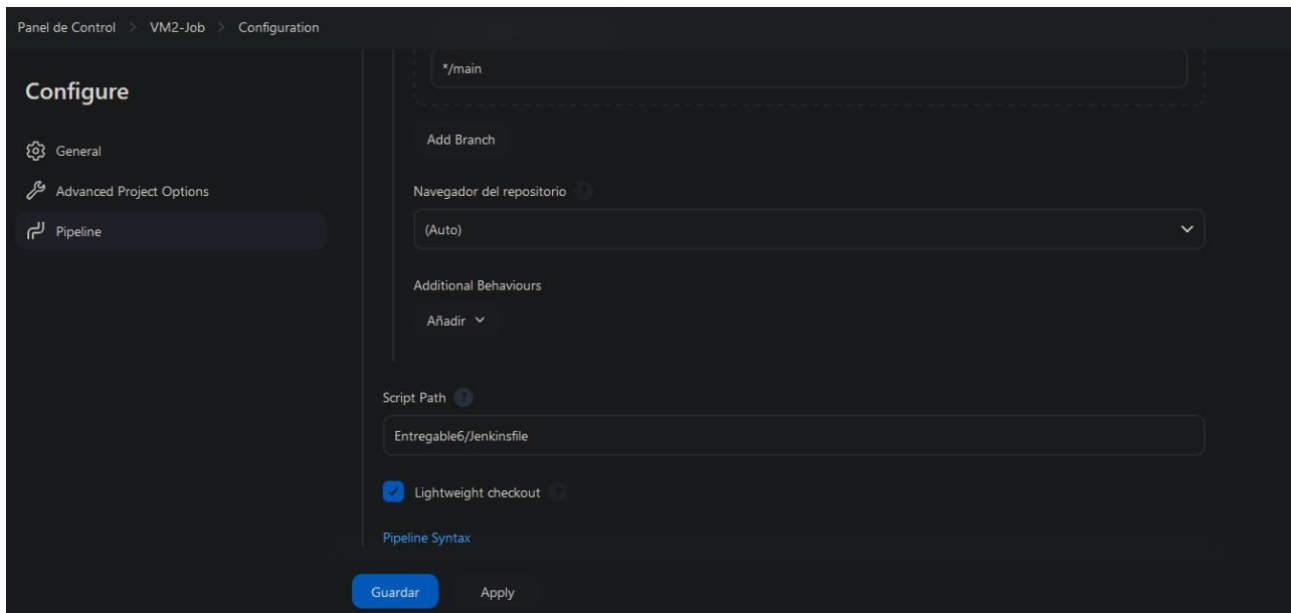
Ejemplo: <https://test.ngrok-free.app/github-webhook>



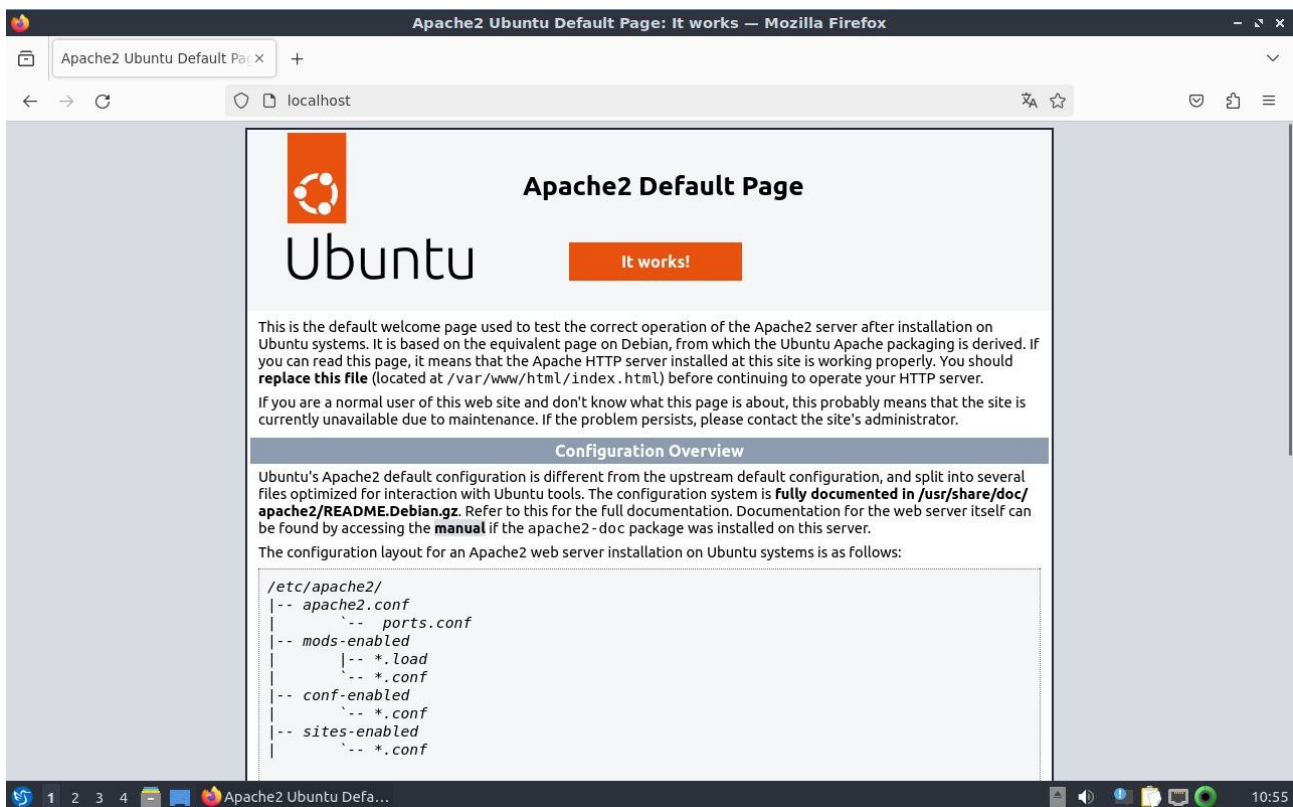
Teniendo el Nodo y las credenciales creadas en Jenkins nos falta el Job, seleccionamos el item type Pipeline y continuamos



Terminamos de configurar a gusto, seteamos la opcion de disparador por medio de Github Hook y le indicamos donde se encuentra el archivo Jenkinsfile dentro del proyecto



Si quedo todo configurado correctamente deberia estar funcionando, antes de enviar un cambio al repositorio vemos que el servidor apache nos devuelve el siguiente sitio web



Mandamos un archivo index.html al repositorio



```
index.html: Bloc de notas
Archivo Edición Formato Ver Ayuda
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Hola Mundo</title>
</head>
<body>
  <h1>Hola Mundo. Esto es apache luego del commit!</h1>
</body>
</html>
```

Luego, si en el Jenkinsfile seteamos los comandos correspondientes se debería reemplazar el index.html de Apache por el que mandamos

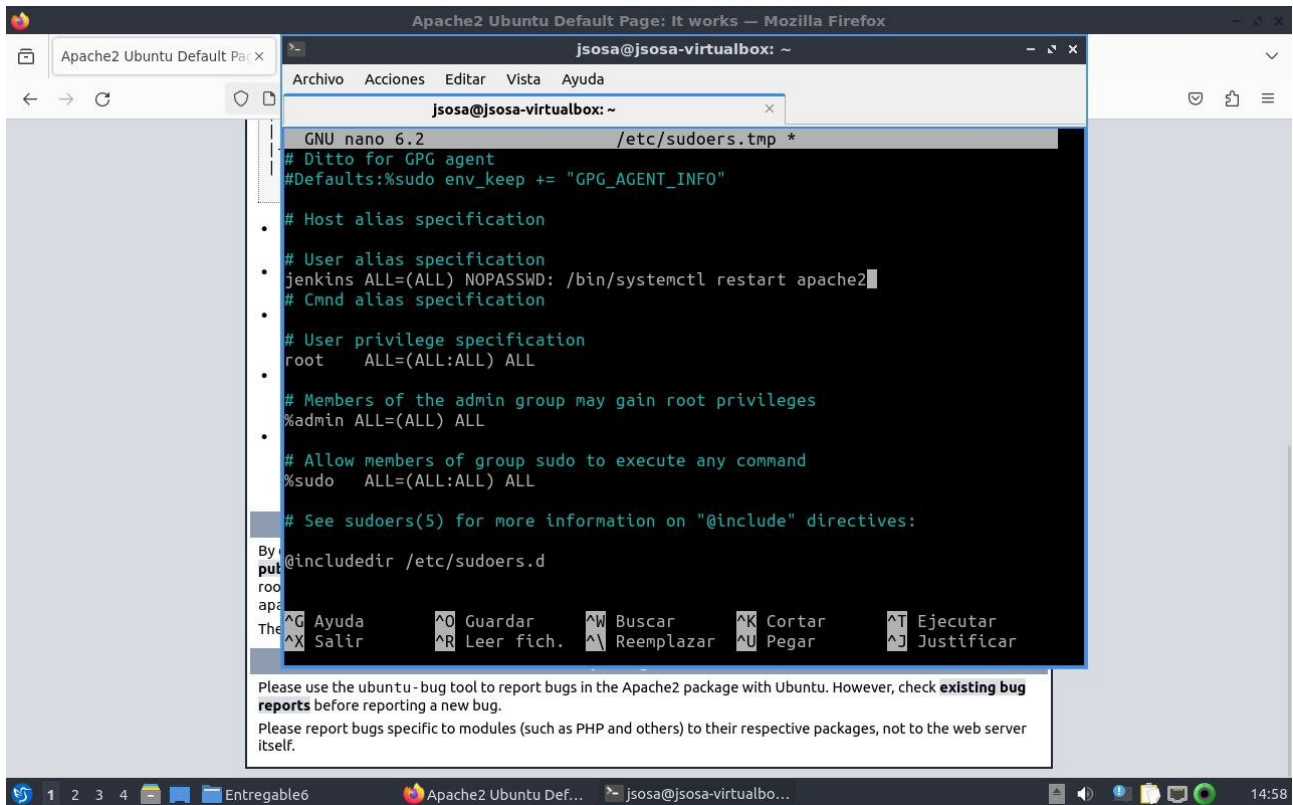
```
pipeline {
  agent { label 'apache-server' }

  stages {
    stage('Clone Repository') {
      steps {
        // Clonar el repositorio en el repositorio raiz de Jenkins
        sh '''
        cd /home/jsosa/jenkins
        git clone https://github.com/nohaynicksdisponibles/bootcamp-devops.git
        '''
      }
    }

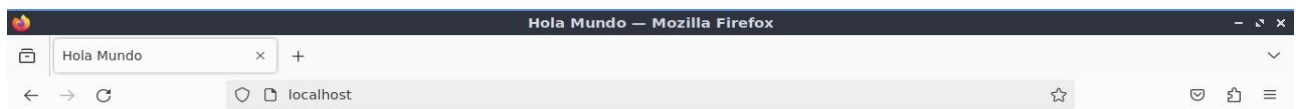
    stage('Update Code on Server') {
      steps {
        // Hacer pull de la última versión y reemplazar el archivo en el servidor Apache
        sh '''
        cd /home/jsosa/jenkins
        git pull origin main
        cp index.html /var/www/html/index.html
        '''
      }
    }

    stage('Clean Up') {
      steps {
        // Eliminar el directorio clonado
        sh '''
        rm -rf /home/jsosa/jenkins/bootcamp-devops
        '''
      }
    }
  }
}
```


de no poderse debemos modificar el archivo /etc/sudoers



Para finalizar corroboramos que se haya actualizado el servidor Apache



Hola Mundo. Esto es apache luego del commit!

