

Automatización de Despliegue con Ansible y Jenkins

1- Configuración de los servidores	2
1.1 Creación de servidores.	2
1.2 Instalación de herramientas necesarias	2
1.3 Generación de claves SSH necesarias	2
1.3.1 En el servidor Jenkins (192.168.1.20)	2
1.3.2 En el servidor Ansible (192.168.1.21)	2
1.3.3 Validación de acceso	3
2- Configuración de Ansible	3
2.1 Configuración del inventario	3
2.2 Creación del archivo ansible.cfg	3
2.3 Desarrollo del playbook	3
Código del Playbook	4
3- Instalar Ngrok	5
3.1 Instalación de ngrok	5
3.2 Configuración inicial	6
4. Configuración de Jenkins	6
4.1 Instalación y configuración inicial	6
4.2 Crear las credenciales necesarias	6
1. SSH para Ansible:	6
2. Acceso a GitHub:	6
4.3 Crear el pipeline	7
5. Configuración de Git	8
5.1 Configurar el repositorio	8
5.2 Configurar el webhook	8
5.3 Pruebas del webhook	8
6. Ejecución	9

1- Configuración de los servidores

1.1 Creación de servidores.

1. Asignaremos las siguientes direcciones IP a cada servidor:
 - Jenkins: 192.168.1.20 - 2 CPUs, 4GB RAM, 20GB Disco
 - Ansible: 192.168.1.21 - 2 CPUs, 2GB RAM, 10GB Disco
 - Nginx: 192.168.1.23 - 1 CPU, 1GB RAM, 8GB Disco
 - Tomcat: 192.168.1.22 - 1 CPU, 1GB RAM, 8GB Disco

1.2 Instalación de herramientas necesarias

1. En el servidor Ansible (192.168.1.21):
 - **Instala Ansible:**
sudo apt update
sudo apt install ansible -y
2. En el servidor Jenkins (192.168.1.20):
 - **Instalar Jenkins:**
sudo apt update
sudo apt install openjdk-17-jdk -y
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key
add -
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ >
/etc/apt/sources.list.d/jenkins.list'
sudo apt update
sudo apt install jenkins -y

1.3 Generación de claves SSH necesarias

1.3.1 En el servidor Jenkins (192.168.1.20)

Genera las claves SSH para Jenkins:

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

Copia la clave pública al servidor Ansible (192.168.1.21):

```
ssh-copy-id -i ~/.ssh/id_rsa.pub asurraco@192.168.1.21
```

1.3.2 En el servidor Ansible (192.168.1.21)

Genera las claves SSH para Ansible:

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

Copia la clave pública a los servidores web:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub asurraco@192.168.1.22  
ssh-copy-id -i ~/.ssh/id_rsa.pub asurraco@192.168.1.23
```

1.3.3 Validación de acceso

Desde Jenkins, verifica el acceso SSH al servidor Ansible:

```
ssh asurraco@192.168.1.21
```

Desde Ansible, verifica el acceso SSH a los servidores web:

```
ssh asurraco@192.168.1.22
```

```
ssh asurraco@192.168.1.23
```

2- Configuración de Ansible

2.1 Configuración del inventario

Crea el archivo hosts.ini con los detalles de los servidores:

```
[webservers]
```

```
server-nginx ansible_host=192.168.1.23 ansible_user=usuario_ansible
```

```
server-tomcat ansible_host=192.168.1.22 ansible_user=usuario_ansible
```

2.2 Creación del archivo ansible.cfg

Define parámetros predeterminados:

```
[defaults]
```

```
inventory = /ruta/al/hosts.ini
```

```
remote_user = usuario_ansible
```

```
host_key_checking = False
```

2.3 Desarrollo del playbook

El playbook realiza las siguientes tareas:

1. Crea el usuario usuario_ansible.
2. Configurar SSH.
3. Instala servicios (Nginx, Java, y Tomcat).
4. Despliega el archivo index.html.

Código del Playbook

- name: Configurar usuario ansible y desplegar archivo en servidores
Nginx y Apache

hosts: webservers

become: true

tasks:

- name: Crear usuario ansible

user:

name: usuario_ansible

state: present

groups: sudo

create_home: yes

- name: Configurar permisos SSH

file:

path: /home/usuario_ansible/.ssh

state: directory

mode: '0700'

- name: Copiar clave pública

copy:

content: "{{ lookup('file', '~/ssh/id_rsa.pub') }}"

dest: /home/usuario_ansible/.ssh/authorized_keys

mode: '0600'

- name: Instalar Nginx

apt:

name: nginx

state: present

when: "'server-nginx' in inventory_hostname"

- name: Instalar Java para Tomcat

apt:

name: default-jdk

state: present

when: "'server-tomcat' in inventory_hostname"

- name: Instalar Tomcat

get_url:

```

        url:
"https://downloads.apache.org/tomcat/tomcat-11/v11.0.1/bin/apache-to
mcat-11.0.1.tar.gz"
        dest: /tmp/apache-tomcat.tar.gz
        when: "'server-tomcat' in inventory_hostname"

- name: Extraer Tomcat
  unarchive:
    src: /tmp/apache-tomcat.tar.gz
    dest: /opt/
    remote_src: yes
    when: "'server-tomcat' in inventory_hostname"

- name: Desplegar archivo en Nginx
  get_url:
    url:
'https://raw.githubusercontent.com/usuario/repo/index.html'
    dest: /var/www/html/index.html
    when: "'server-nginx' in inventory_hostname"

- name: Desplegar archivo en Tomcat
  get_url:
    url:
'https://raw.githubusercontent.com/usuario/repo/index.html'
    dest: /opt/apache-tomcat-11.0.1/webapps/ROOT/index.html
    when: "'server-tomcat' in inventory_hostname"

```

3- Instalar Ngrok

3.1 Instalación de ngrok

1. Ve al servidor Jenkins (192.168.1.20).

Descarga ngrok ejecutando:

```

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

```

3.2 Configuración inicial

1. Autentica ngrok en el servidor Jenkins:

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

2. Inicia ngrok:

Ejecuta el siguiente comando para exponer Jenkins en el puerto 8080:

```
ngrok http 8080
```

3. Obtén la URL pública:

- ngrok generará una URL como **https://<SUBDOMINIO>.ngrok.io**. Esta URL será utilizada en el webhook de GitHub.

4. Configuración de Jenkins

4.1 Instalación y configuración inicial

1. Accede a Jenkins desde el navegador con la url de ngrok:
 - URL: `https://<SUBDOMINIO>.ngrok.io`
2. Instala los plugins necesarios desde Manage Jenkins > Plugin Manager:
 - Pipeline
 - Git Plugin
 - SSH Agent Plugin

4.2 Crear las credenciales necesarias

1. SSH para Ansible:
 - Ve a Manage Jenkins > Credentials > System > Global credentials (unrestricted) y selecciona Add Credentials.
 - Tipo: SSH Username with private key.
 - Configura:
 - Username: `usuario_ansible`.
 - Private Key: Copia la clave privada (`id_rsa_ansible`) generada previamente.
 - ID: `ansible-ssh-credentials`.
2. Acceso a GitHub:
 - Si usas SSH:
 - Crea una credencial de tipo SSH Username with private key.
 - Configura:
 - Username: tu usuario de GitHub.
 - Private Key: la clave privada generada en el servidor Jenkins.
 - ID: `github-ssh`.

4.3 Crear el pipeline

1. Ve a New Item y selecciona Pipeline.
2. Asigna un nombre como deploy_ansible.

Configura el pipeline con este script:

```
pipeline {
  agent any
  stages {
    stage('Clonar Repositorio') {
      steps {
        git credentialsId: 'github-ssh', url: 'https://github.com/usuario/repo.git'
      }
    }
    stage('Ejecutar Playbook') {
      steps {
        sshagent(['ansible-ssh-credentials']) {
          sh 'ansible-playbook -i
/home/asurraco/project-entrega11/ineventories/devs/hosts.ini
/home/asurraco/project-entrega11/playbooks/deploy.yml'
        }
      }
    }
  }
}
```

3. Guarda y prueba el pipeline manualmente para confirmar que funciona.

5. Configuración de Git

5.1 Configurar el repositorio

Ve al servidor Jenkins y clona tu repositorio en local para probar el acceso:

```
git clone git@github.com:usuario/repo.git
```

1. Si puedes clonar sin ingresar credenciales, el acceso está configurado correctamente.
2. Asegúrate de que el archivo index.html esté en el repositorio.

5.2 Configurar el webhook

1. En GitHub, ve al repositorio correspondiente y accede a:
 - Settings > Webhooks > Add Webhook.
2. Configura:
 - **Payload URL**: La URL de Jenkins (por ejemplo, `http://ngrok-url/github-webhook/`).
 - **Content Type**: application/json.
 - **Events**: Selecciona Push events para que el pipeline se dispare cuando se haga un push.

5.3 Pruebas del webhook

1. Haz un cambio en el repositorio (por ejemplo, modifica el archivo index.html).
2. Haz un git push y verifica en GitHub que el webhook se haya activado.
3. En Jenkins, verifica que el pipeline se haya disparado automáticamente.

6. Ejecución

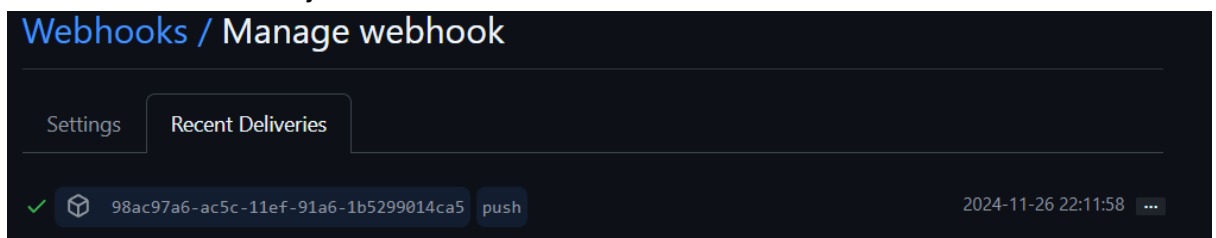
1. En nuestro servidor Jenkins editamos el archivo index.html

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Desafío 11</title>
6 </head>
7 <body>
8   <h1>Ejecucion de Jenkins con Webhook - Integración con Ansible - Instalación de Nginx y Tomcat-¡Despliegue automático exitoso!</h1>
9 </body>
10 </html>
```

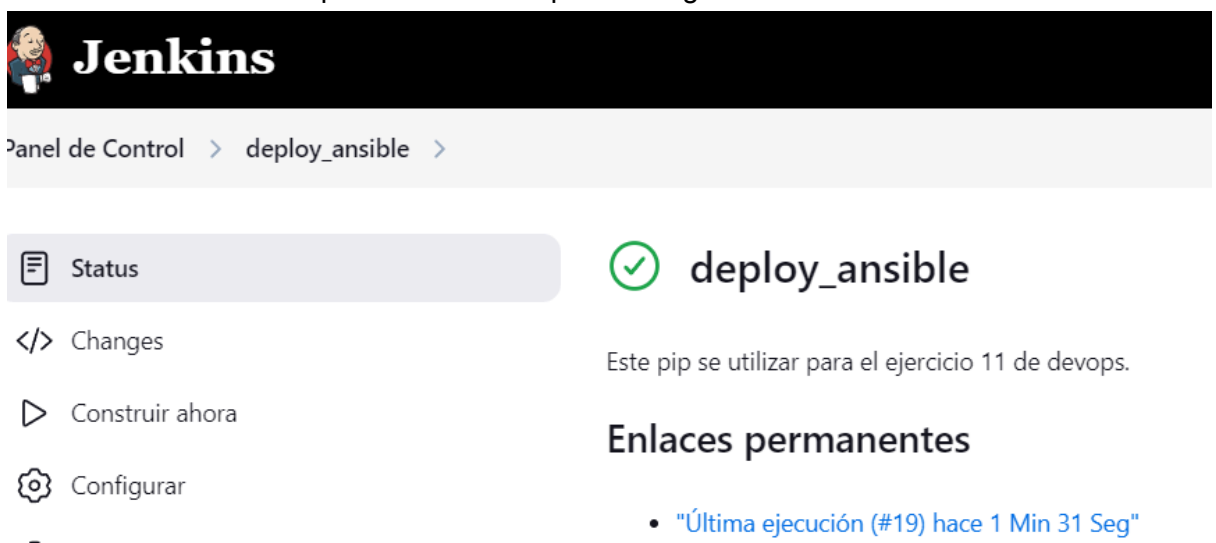
Le agrego un 2 al final del texto y salvamos el cambio.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Desafío 11</title>
6 </head>
7 <body>
8   <h1>Ejecucion de Jenkins con Webhook - Integración con Ansible - Instalación de Nginx y Tomcat-¡Despliegue automático exitoso!2</h1>
9 </body>
10 </html>
```

2. Ejecutamos los siguientes comando:
 - a. git add index.html
 - b. git commit -m "Ejecución con webhooks"
 - c. git push origin master
3. Verificamos en GIT la ejecución del webhook



4. Verificamos en Jenkins que la tarea se disparara luego de actualizar index.html



5. Verificamos el output de la ejecución

```
PLAY RECAP *****
server-nginx      : ok=10   changed=1   unreachable=0   failed=0   skipped=9   rescued=0   ignored=0
server-tomcat     : ok=14   changed=5   unreachable=0   failed=0   skipped=5   rescued=0   ignored=0

[Pipeline] }
$ ssh-agent -k
unset SSH_AUTH_SOCK;
unset SSH_AGENT_PID;
echo Agent pid 4687 killed;
[ssh-agent] Stopped.
[Pipeline] // sshagent
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] sh
+ echo Despliegue exitoso: index.html se completó correctamente.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```