

# Deploy a Servidor Linux Usando Github Actions

By Freddy Alcarazo

20 de Junio del 2025 - Perú

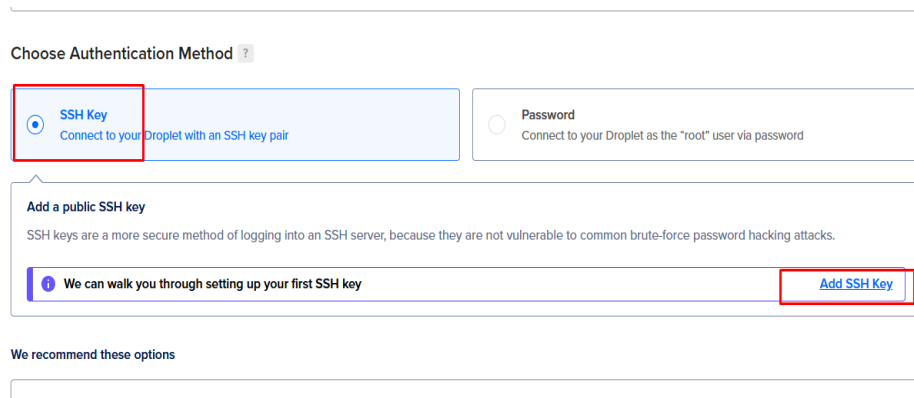
Video Tutorial: <https://youtu.be/zdpOSGLxZEK>

## Pasos:

### 1. Crear maquina virtual en la nube de su preferencia.

En este caso se creará una máquina virtual en DigitalOcean.com

Al momento de crear el droplet, elegir el método de autenticación usando SSH Key.



Choose Authentication Method ?

☒ **SSH Key**  
Connect to your Droplet with an SSH key pair

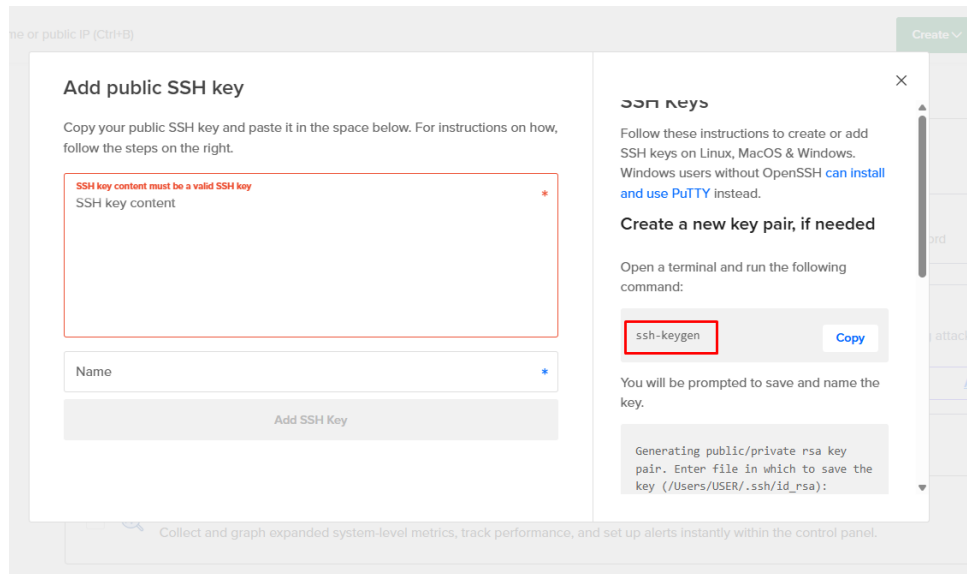
☐ **Password**  
Connect to your Droplet as the "root" user via password

**Add a public SSH key**  
SSH keys are a more secure method of logging into an SSH server, because they are not vulnerable to common brute-force password hacking attacks.

*i* We can walk you through setting up your first SSH key [Add SSH Key](#)

We recommend these options

Luego de clickear en “**Add SSH Key**” se apertura el siguiente diálogo. Aquí tenemos que copiar el comando “**ssh-keygen**” y ejecutarlo en una terminal **CMD** en Windows, esto nos va a generar una llave pública y privada. **Se recomienda generar la llave en un directorio seguro.**

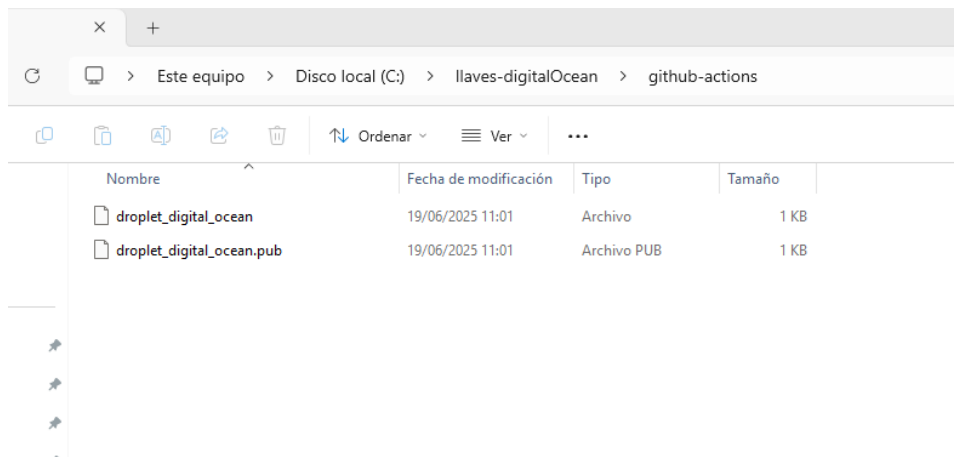


Nota Importante: ¡Cuando le pida “Enter passphrase: ” no escriban nada solo den enter!  
 Más del 90% de todas las claves SSH en la mayoría de las grandes empresas no tienen frase de contraseña (passphrase).

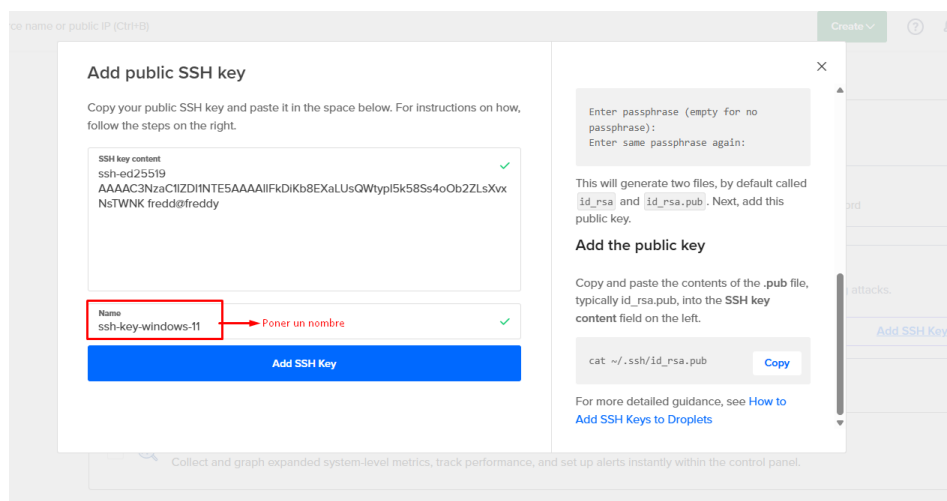
```
C:\llaves-digitalOcean\github-actions>ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\fredd/.ssh/id_ed25519): droplet_digital_ocean
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in droplet_digital_ocean
Your public key has been saved in droplet_digital_ocean.pub
The key fingerprint is:
SHA256:wThCIZ6M0S7GzpHsYLUnjs7ZIHx2I3wHiZH/q9o9L6I fredd@freddy
The key's randomart image is:
+--[ED25519 256]--+
|... oo          |
| =.=o o         |
|+.* o+o.o       |
|oBoo+o+. .     |
|Bo+++.oS        |
|. * .o + o      |
|+ + o o .       |
|+ . ...+        |
|   Eooo.+       |
+---[SHA256]-----+

C:\llaves-digitalOcean\github-actions>ls
"ls" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
```

Como se puede ver yo generé las llaves dentro de esta ruta:  
 C:\llaves-digitalOcean\github-actions



El archivo que no tiene extensión “.pub” es la llave privada. Ahora tenemos que abrir el archivo con extensión “.pub” y copiar su contenido y pegarlo en digitalocean.



¡Luego en este caso click en “Add SSH Key” y finalizamos creando el droplet!

## 2. Prueba de Conexión SSH a la máquina virtual usando llave privada.

Abrimos una terminal dentro de la carpeta que contiene las llaves publica y privada que se generó antes para hacer una conexión de prueba vía SSH.

```
$ ssh -i droplet_digital_ocean root@67.205.136.246
```

Nota: droplet\_digital\_ocean es el nombre del archivo de **la llave privada**, con este seremos capaces de hacer la conexión.

```
C:\laves-digitalOcean\github-actions>ssh -i droplet_digital_ocean root@67.205.136.246
The authenticity of host '67.205.136.246 (67.205.136.246)' can't be established.
ED25519 key fingerprint is SHA256:1YkgJnECzv3PW9cb6KpXe8cDo8FW3XD8AN7MnWB1TgQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '67.205.136.246' (ED25519) to the list of known hosts.
Enter passphrase for key 'droplet_digital_ocean':
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-51-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Jun 19 16:14:28 UTC 2025

System load:  0.07               Processes:    101
Usage of /:   3.8% of 47.39GB    Users logged in: 0
Memory usage: 9%               IPv4 address for eth0: 67.205.136.246
Swap usage:  0%                IPv4 address for eth0: 10.10.0.5

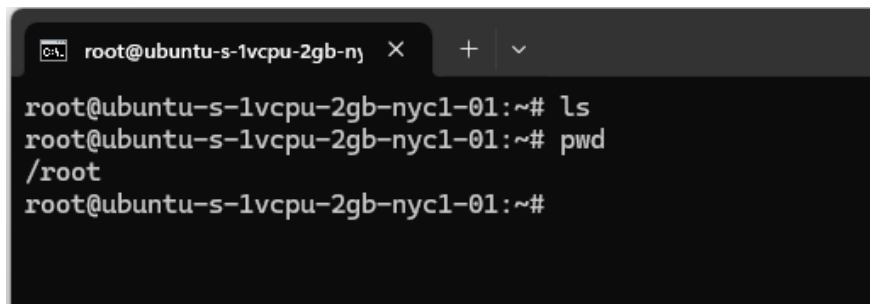
Expanded Security Maintenance for Applications is not enabled.

225 updates can be applied immediately.
114 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

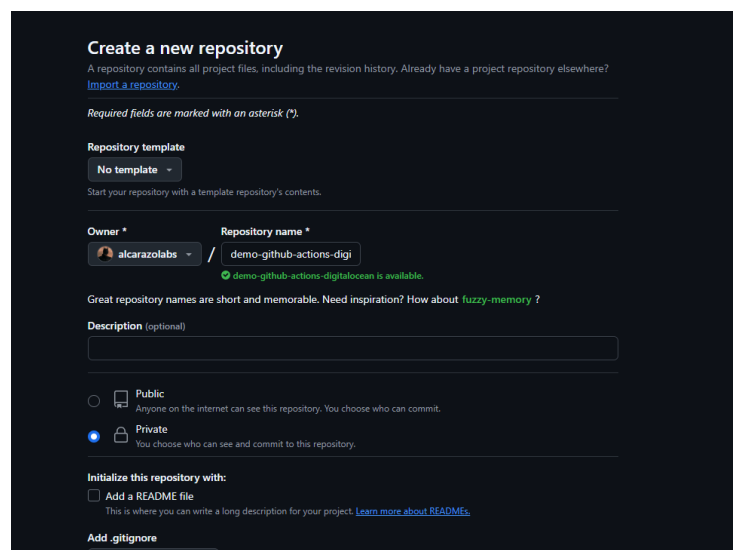
¡Y listo ya probamos la conexión!



```
root@ubuntu-s-1vcpu-2gb-ny X + v
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# ls
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# pwd
/root
root@ubuntu-s-1vcpu-2gb-nyc1-01:~#
```

### 3. Crear repositorio privado y configuración

Creemos un repositorio privado.



**Create a new repository**  
 A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository](#)

*Required fields are marked with an asterisk (\*).*

**Repository template**  
 No template

Start your repository with a template repository's contents.

**Owner \*** alcarazolabs **Repository name \*** demo-github-actions-digi  
 demo-github-actions-digitalocean is available.

Great repository names are short and memorable. Need inspiration? How about fuzzy-memory?

**Description (optional)**

☐ **Public**  
 Anyone on the internet can see this repository. You choose who can commit.

☒ **Private**  
 You choose who can see and commit to this repository.

**Initialize this repository with:**  
☐ Add a README file  
 This is where you can write a long description for your project. [Learn more about READMEs](#).

**Add .gitignore**

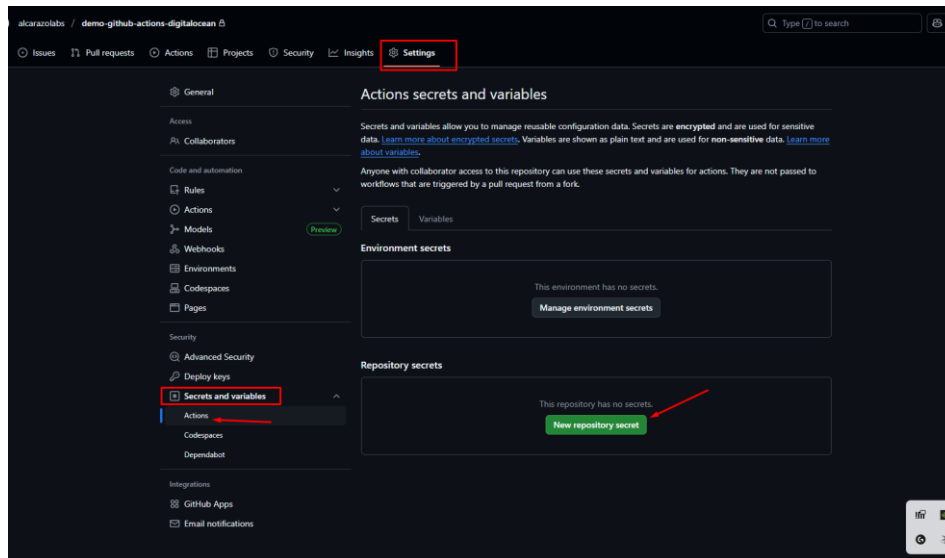
Luego inicializar el repositorio en la PC, pero antes veamos el contenido del workflow que vamos a probar.

```
1 name: Deploy
2
3 on: push
4
5 jobs:
6   build:
7     runs-on: ubuntu-latest
8
9     steps:
10      - name: Checkout
11        uses: actions/checkout@v4
12
13      - name: Setup SSH
14        uses: webfactory/ssh-agent@v0.9.0
15        with:
16          ssh-private-key: ${ secrets.SSH_KEY }
17
18      - name: Create file
19        run: ssh -o StrictHostKeyChecking=no root@142.93.1.231 touch hola.txt
20
21 # - name: Deploy to server
22 #   run: bash deploy.sh
```

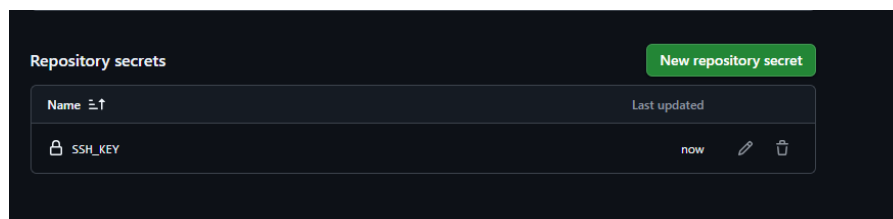
Este workflow tiene como objetivo crear un archivo llamado “hola.txt” dentro del servidor como ven en la línea 19, al final pueden ver el comando “touch.txt”.

Además, como se puede ver los comentarios con letra roja en la imagen anterior debemos de crear una variable llamada “SSH\_KEY” o secreto dentro de los settings del repositorio y esta variable en mi caso le voy a poner SSH\_KEY y su contenido será el contenido la llave privada generada anteriormente (ver paso 1). Para crear el secreto nos vamos a:

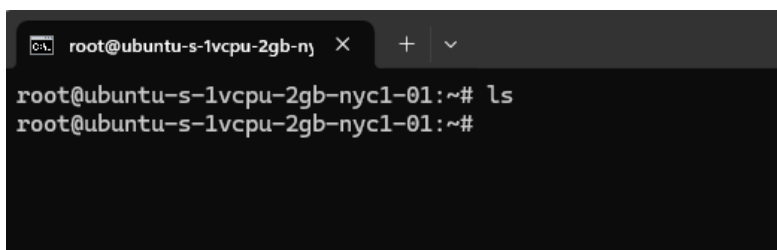
Settings->Secrets and Variables->Actions y en Repository secrets hacemos click en “New Repository Secret”



Ponemos un nombre al secreto y pegamos el contenido de la llave privada de conexión ssh que se creó antes.

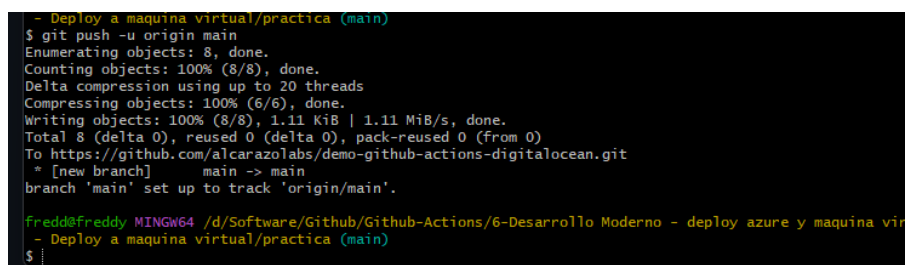


¡Ya estamos listos para hacer push! Como se puede ver en la imagen de abajo, no existe ningún archivo dentro del directorio root de la VM.

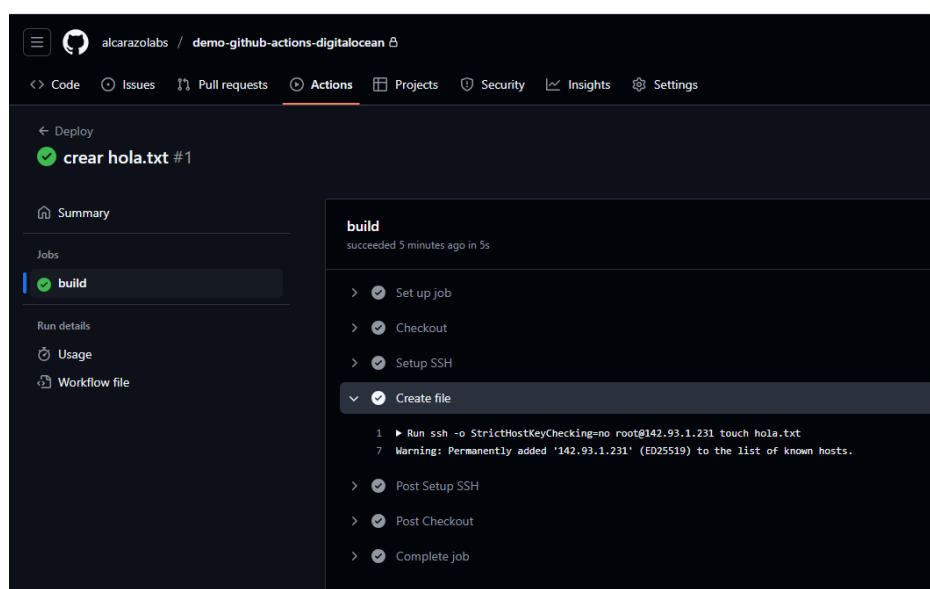
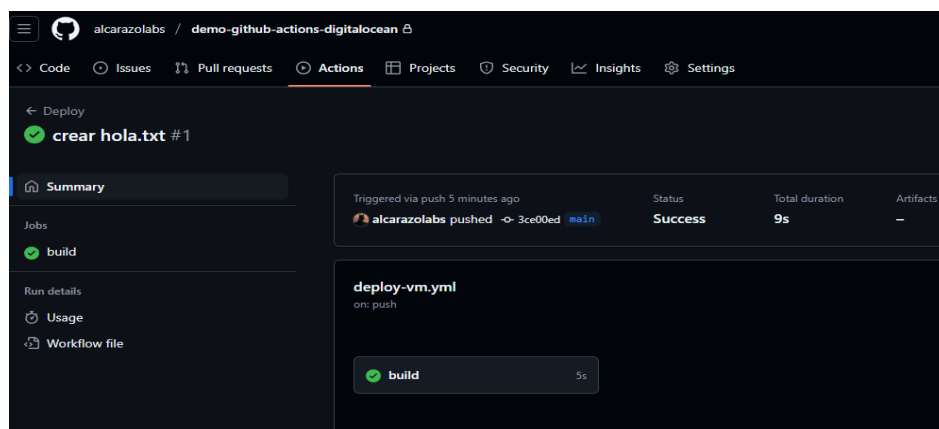
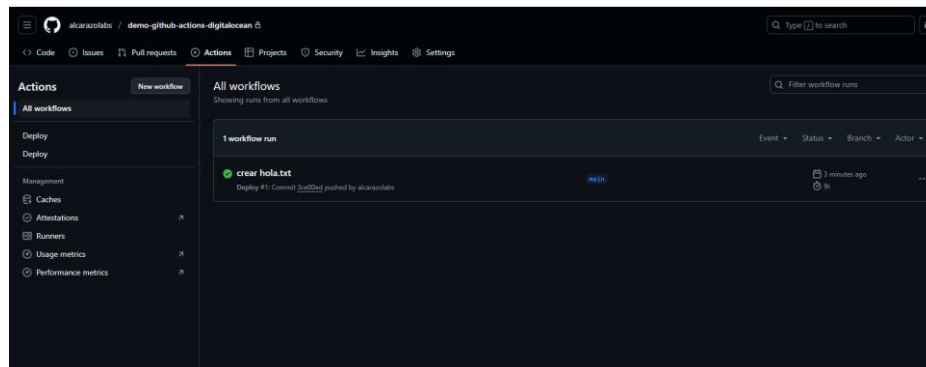


A continuación, se muestran pantallazos de que todo salió bien:

Haciendo push:

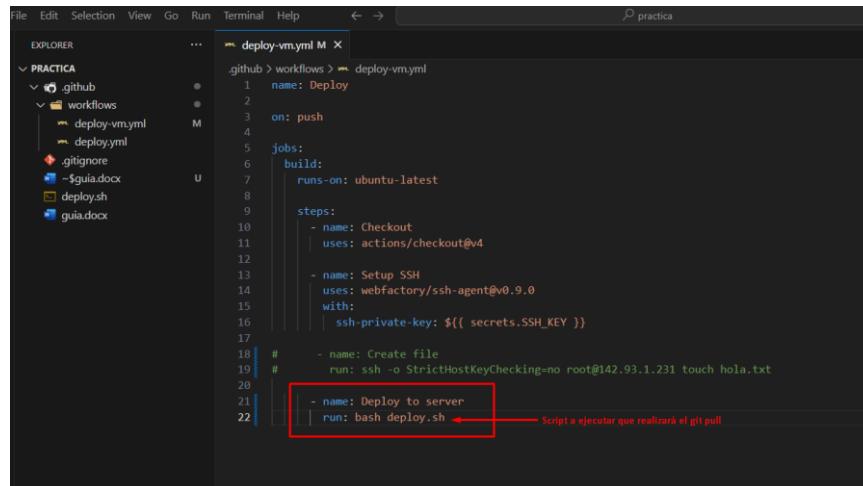


¡Nos vamos a “Actions” del repositorio y vemos ahí el action con el nombre del commit y en efecto se ejecutó bien ya que tiene el check verde!

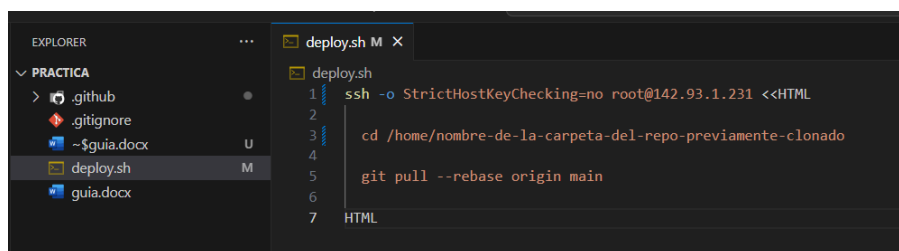


#### 4. Hacer git pull del repositorio en la maquina virtual

Con nuestro workflow anterior solo creamos un archivo llamado “hola.txt” lo cual es algo básico para comprobar las conexiones entre github y nuestro VPS/VM. Lo que haremos ahora es ejecutar un paso extra en nuestro job actual del workflow que haga un “git pull” de nuestro repositorio en el servidor remoto, obviamente esto se hará luego de hacer git push desde nuestra máquina.



Como se puede ver en la imagen anterior vamos a correr un paso llamado “Deploy to server”, este va a ejecutar un archivo llamado “deploy.sh” este archivo se encuentra en la raíz del repositorio fuera de la carpeta .github:



Este archivo.sh se conectara al servidor por ssh (ustedes deben de actualizar el usuario y la ip del servidor) y luego ejecutara el comando “cd” para moverse a un directorio en especifico en este caso se moverá al /home y se ubicará dentro de la carpeta del repositorio clonado previamente y finaliza haciendo “git pull” de la rama “main”.

Antes de ejecutar o hacer push para que se ejecute el workflow lo que realizaremos es hacer un “git clone” del repositorio actual dentro del servidor VPS/VM. Para hacer ello realizaremos una buena práctica que es crear nuestras llaves publica y privada para que así github pueda permitir hacer git clone del repositorio bien sea privada o público, pero



lo más obvio de esto de generar las llaves es porque estaremos tratando con repositorios privados. Exactamente generaremos las llaves y solo vamos a copiar el contenido de la llave publica y la agregaremos al repositorio actual o bien usarla también en otros repositorios.

Lanzaré el comando “ssh-keygen” dentro de la carpeta root, **todo le damos Enter** si no quieren que se genere las llaves con un nombre custom, en este caso evitaremos darle nombres y dejaremos que el servidor les asigne nombres aleatorios.

```
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# ls
hola.txt
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:6UexD4vouTgcpg6dnhOF3uIFdRNg9dVkpzv8mXwYVU root@ubuntu-s-1vcpu-2gb-nyc1-01
The key's randomart image is:
+--[ED25519 256]--+
|    .oo. .o+  E|
| .. o. o =.   |
| o . . =      |
| o . . + .    |
| .+  S + o o   |
| .+ = o o = + .|
| ..oB .. o + o +|
| o+.oo . . o o |
| .+...+.      |
+-----[SHA256]-----+
root@ubuntu-s-1vcpu-2gb-nyc1-01:~#
```

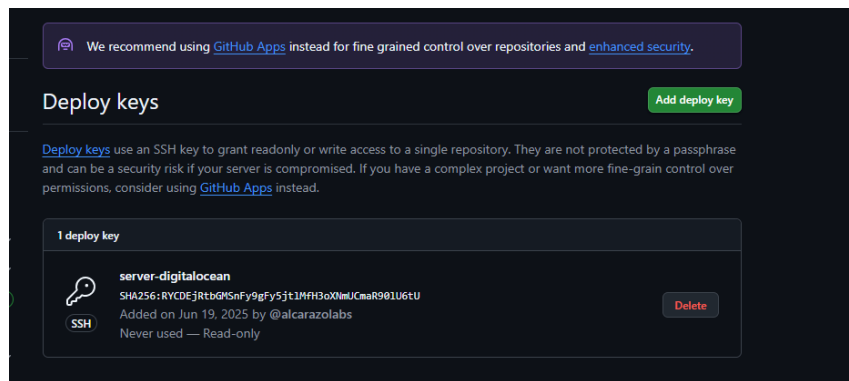
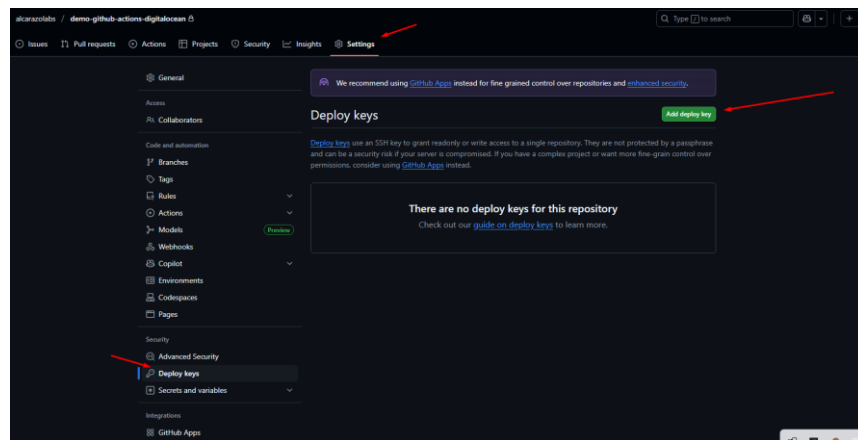
A continuación, accedemos a la carpeta “.ssh” y podrán ver que se generaron las 2 llaves:

```
root@ubuntu-s-1vcpu-2gb-nyc1-01:~/.ssh# ls
authorized_keys id_ed25519 id_ed25519.pub
root@ubuntu-s-1vcpu-2gb-nyc1-01:~/.ssh#
```

Hacemos un cat de la llave que tiene “.pub” y copiamos la salida:

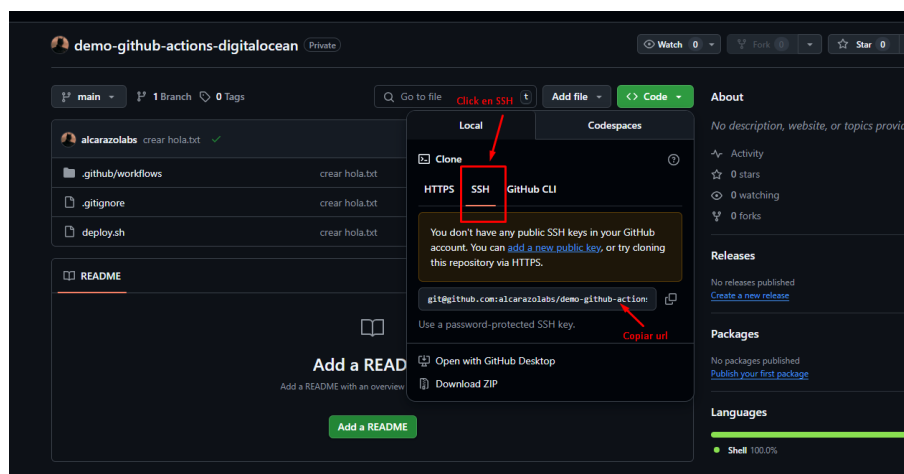
```
root@ubuntu-s-1vcpu-2gb-nyc1-01:~/.ssh# ls
server server.pub
root@ubuntu-s-1vcpu-2gb-nyc1-01:~/.ssh# cat server.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFdh5m0YurmlJXZu0u21hk3RqI4Br0jbyZAuyPi/WMJ root@ubuntu-s-1vcpu-2gb-nyc1-01
root@ubuntu-s-1vcpu-2gb-nyc1-01:~/.ssh#
```

Ahora con esto nos vamos al repositorio github a la sección de “settings->Deploy keys y click en el botón “Add deploy key”:



A esta “deploy key” le llamé “server-digitalocean”. Esta llave permitirá a cualquier persona que tenga la llave privada pueda hacer “git pull” del repositorio.

Listo ahora en el servidor hacemos **git clone** del repositorio, pero usando la URL de **SSH**:



esto lo realizaré dentro de la carpeta /home.

```
root@ubuntu-s-1vcpu-2gb-nyc1-01:~# cd /home/
root@ubuntu-s-1vcpu-2gb-nyc1-01:/home# git clone git@github.com:alcarazolabs/demo-github-actions-digitalocean.git
Cloning into 'demo-github-actions-digitalocean'...
The authenticity of host 'github.com (140.82.113.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3mVV6TuuJJhbpZisF/zLDA0zPMSvHdk4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 8 (delta 0), reused 8 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (8/8), done.
root@ubuntu-s-1vcpu-2gb-nyc1-01:/home# ls
demo-github-actions-digitalocean
root@ubuntu-s-1vcpu-2gb-nyc1-01:/home#
```

¡Y listo se clonó!

Accedemos a la carpeta del repositorio clonado y **copiaremos su ruta**:

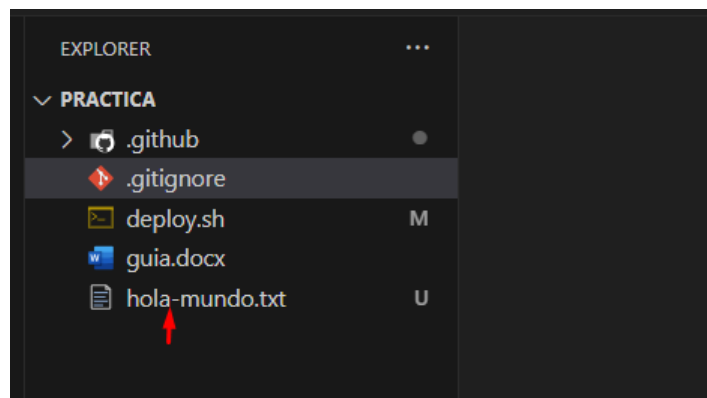
```
root@ubuntu-s-1vcpu-2gb-nyc1-01:/home/demo-github-actions-digitalocean# pwd
/home/demo-github-actions-digitalocean
root@ubuntu-s-1vcpu-2gb-nyc1-01:/home/demo-github-actions-digitalocean#
```

Editamos el archivo “**deploy.sh**” y actualizaremos la ruta del “cd” :

```
deploy.sh M X
1 ssh -o StrictHostKeyChecking=no root@142.93.1.231 <<HTML
2
3 cd /home/nombre-de-la-carpeta-del-repo-previamente-clonado
4
5 git pull --rebase origin main
6
7 HTML
```

```
deploy.sh M X
deploy.sh
1 ssh -o StrictHostKeyChecking=no root@142.93.1.231 <<HTML
2
3 cd /home/demo-github-actions-digitalocean
4
5 git pull --rebase origin main
6
7 HTML
```

¡Y listo! Ahora solo **crearé un archivo llamado “hola-mundo.txt”** en el directorio raíz del repositorio en mi pc:

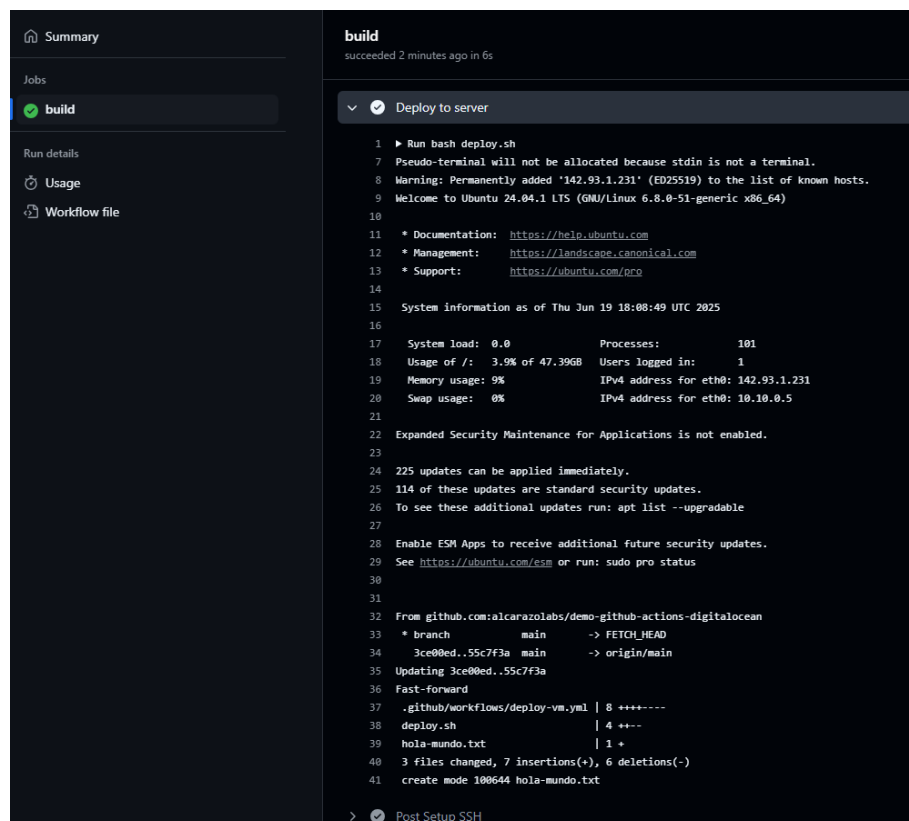
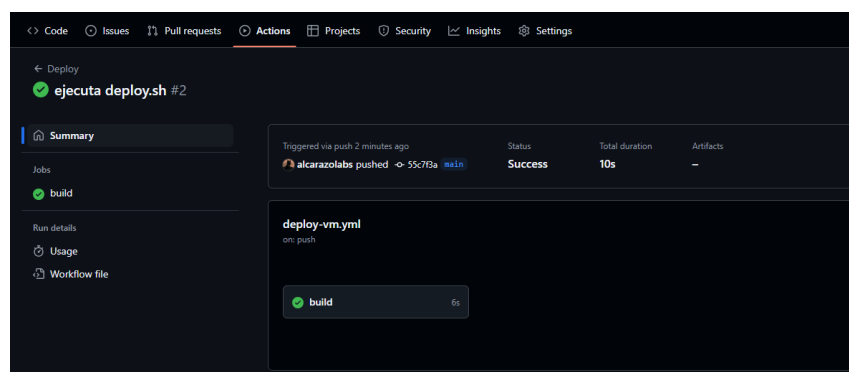
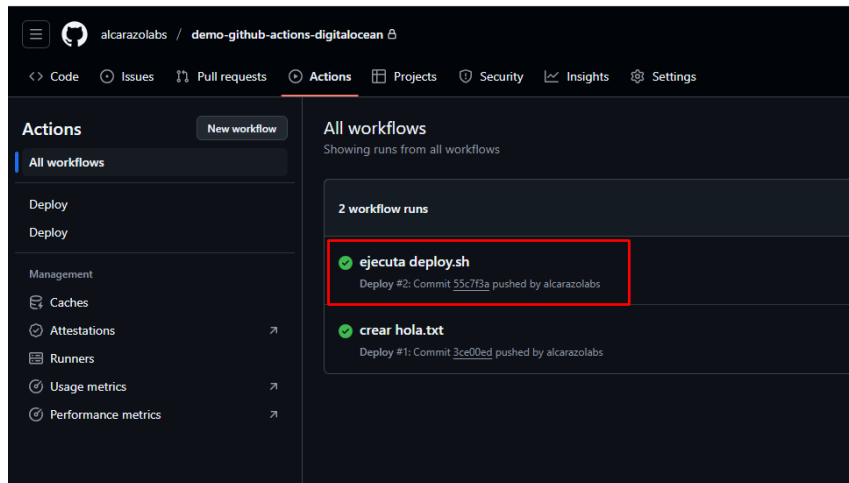


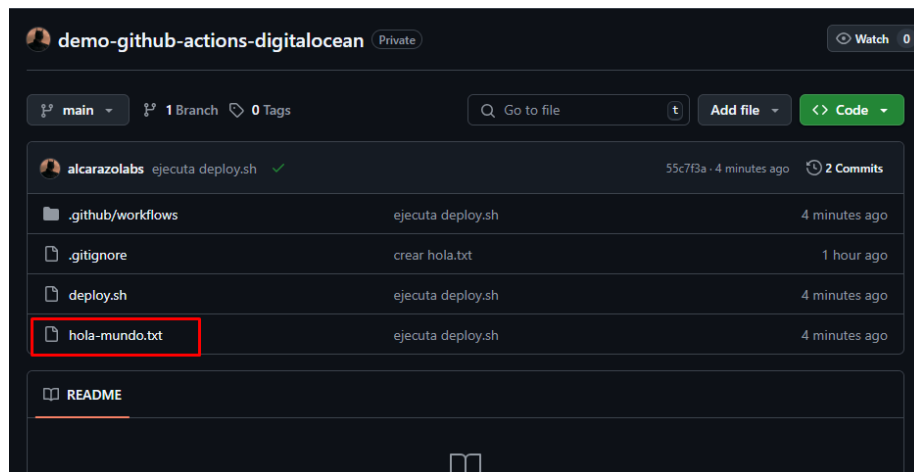
y realizaré un **commit** y **git push**.

```
Freddy@Freddy MINGW64 /d/Software/Github/Github-Actions/6-Desarrollo Moderno - deploy azure y maquina virtual/2
- Deploy a maquina virtual/practica (main)
$ git commit -m "ejecuta deploy.sh"
[main 55c7f3a] ejecuta deploy.sh
3 files changed, 7 insertions(+), 6 deletions(-)
create mode 100644 hola-mundo.txt

Freddy@Freddy MINGW64 /d/Software/Github/Github-Actions/6-Desarrollo Moderno - deploy azure y maquina virtual/2
- Deploy a maquina virtual/practica (main)
$ git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 20 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 714 bytes | 714.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/alcarazolabs/demo-github-actions-digitalocean.git
3ce00ed..55c7f3a main -> main

Freddy@Freddy MINGW64 /d/Software/Github/Github-Actions/6-Desarrollo Moderno - deploy azure y maquina virtual/2
- Deploy a maquina virtual/practica (main)
$
```





Verificando en el servidor que se crearon los nuevos archivos:

```
root@ubuntu-s-1vcpu-2gb-nyc-01: /home/demo-github-actions-digitalocean# pwd
/home/demo-github-actions-digitalocean
root@ubuntu-s-1vcpu-2gb-nyc1-01: /home/demo-github-actions-digitalocean# ls
deploy.sh
root@ubuntu-s-1vcpu-2gb-nyc1-01: /home/demo-github-actions-digitalocean# ls -a
.  ..  .git  .github  .gitignore  deploy.sh
root@ubuntu-s-1vcpu-2gb-nyc1-01: /home/demo-github-actions-digitalocean# ls
deploy.sh  hola-mundo.txt
root@ubuntu-s-1vcpu-2gb-nyc1-01: /home/demo-github-actions-digitalocean#
```