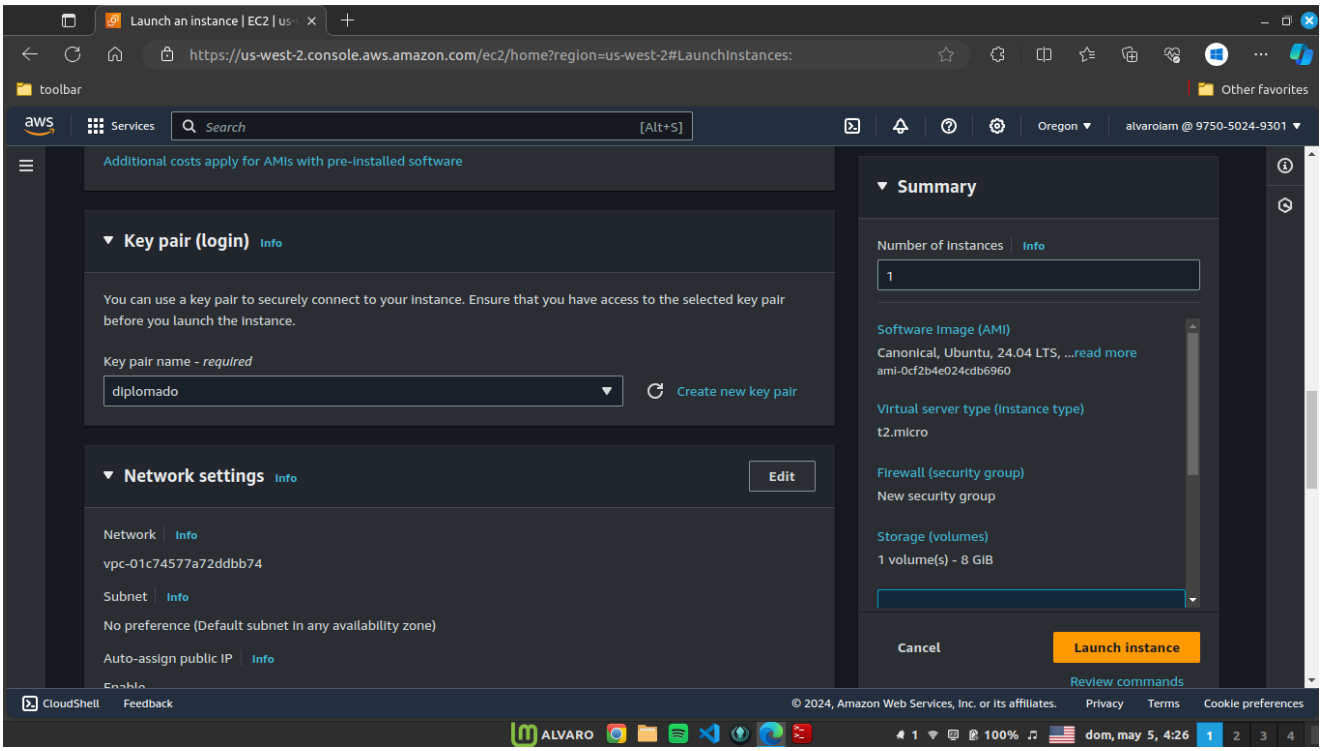
Diplomado_Cloud_Native

Se crearon las 3 VM con sus pares de llaves y se configuraron los puertos de seguridad para poder acceder a las VM desde el exterior.



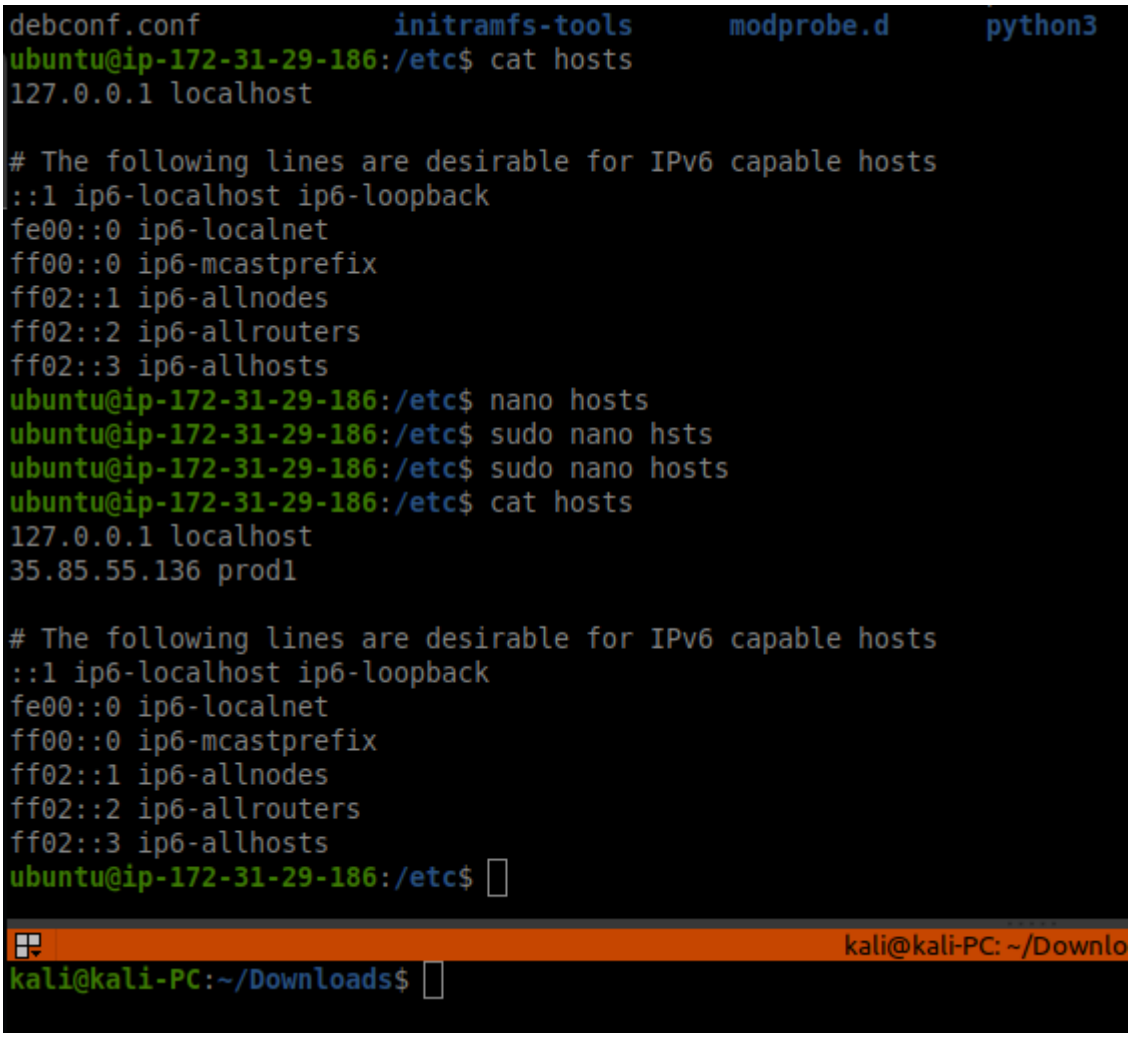
Aplicando Ansible

En el entorno de ejecución de ansible, será aquella máquina que se configura la instalación de ansible así como también el gitlab-runner para el acceso de gitlab.

Pasando por esta configuración se edita el archivo /etc/host
agregando las máquinas de aws que utilizaremos para agregar:

PROF

```
IP_MAQUINA [NOMBRE_REFERENCIA]
```



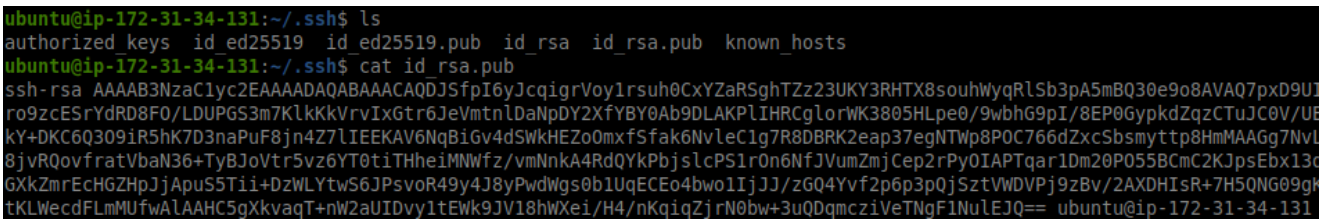
CONFIGURACION SSH

Para la configuración de ssh se debe de generar una llave ssh en la máquina de ansible y copiarla a las máquinas de aws para poder acceder a ellas sin necesidad de contraseña.

MAQUINA ANSIBLE CONTROLLER

PROF

```
ssh-keygen -t rsa -b 4096
cd ~/.ssh
cat id_rsa.pub
```



copiamos el contenido de la llave generada y la pegamos en el archivo ~/.ssh/authorized_keys de la maquina PRODUCCION.

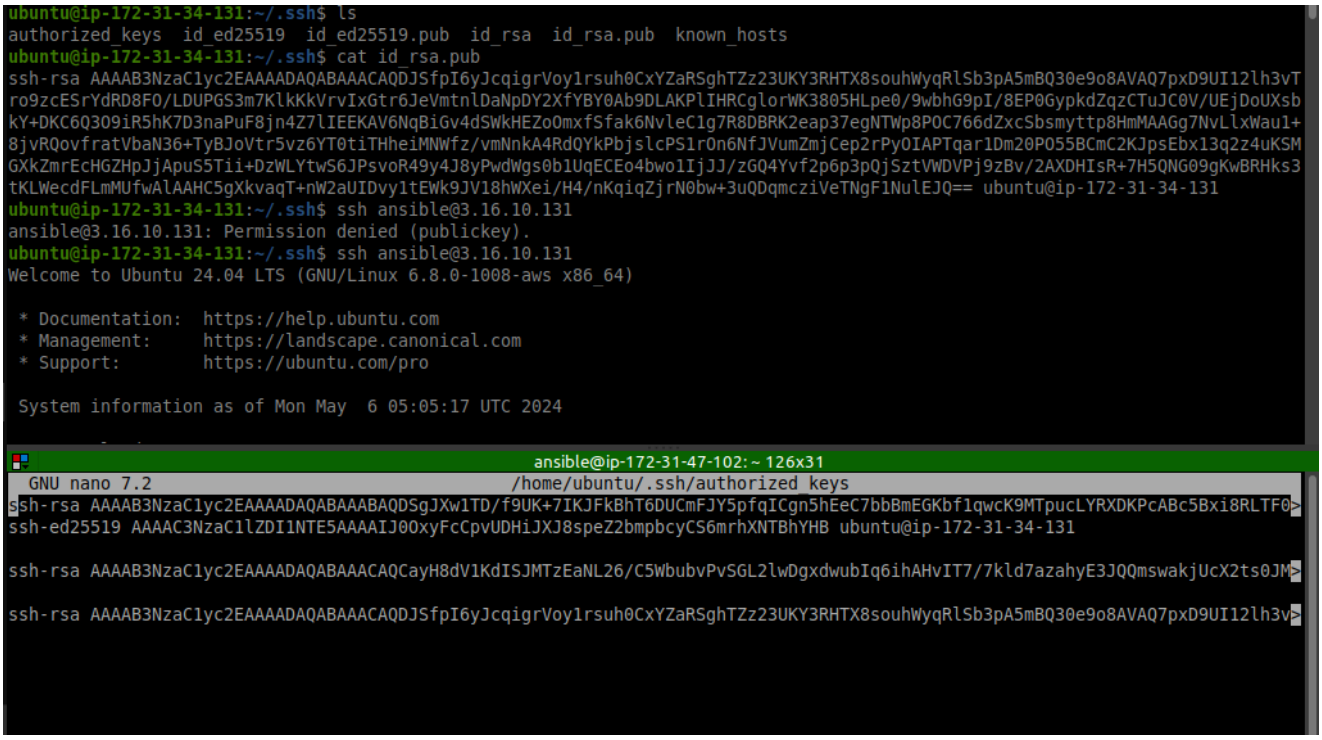
MAQUINA DE PRODUCCION

```
sudo adduser ansible
sudo nano /etc/ssh/sshd_config
# Descomentar lo siguiente:
#       PasswordAuthentication yes
#       AuthorizedKeysFile      .ssh/authorized_keys
.ssh/authorized_keys2
sudo visudo
# Agregar la siguiente linea en especificaciones de usuario
privilegiado:
#       ansible ALL=(ALL) ALL
# --luego...
```

```
# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
ansible ALL=(ALL)       ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

```
# si es ubuntu
sudo systemctl restart ssh
# si es alguna vm de GCP o basada en CentOS
sudo systemctl restart sshd

cd ~/.ssh
sudo nano authorized_keys
# Pegar la llave generada en la maquina de ansible
```

PROF

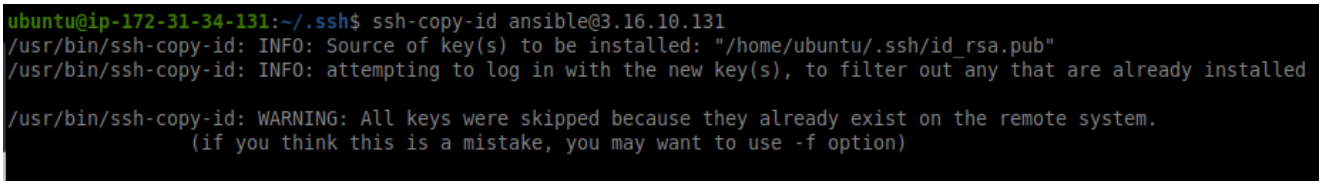



```
su ansible
mkdir /home/ansible/.ssh
sudo cp /home/ubuntu/.ssh/authorized_keys /home/ansible/.ssh/ # copiamos
la llave a la carpeta de ansible

# por ultimo dar los permisos para que ansible pueda acceder a la
maquina
sudo chown -R ansible:ansible /home/ansible/.ssh
sudo chmod 700 /home/ansible/.ssh
sudo chmod 600 /home/ansible/.ssh/authorized_keys
```

MAQUINA DE ANSIBLE

PROF

```
ssh-copy-id ansible@<ip-maquina-produccion>
```



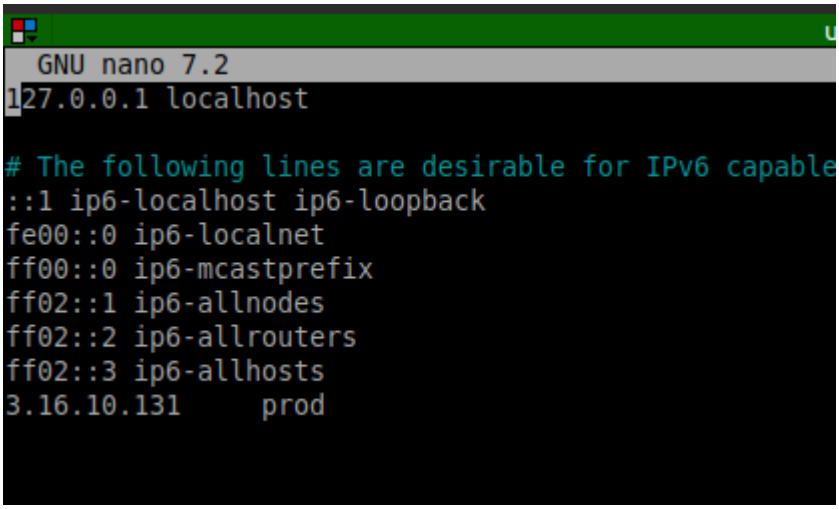

```
# INSTALACION DE ANSIBLE
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
```

```
sudo apt install ansible
ansible --version
```

```
ubuntu@ip-172-31-34-131:~/.ssh$ ansible --version
ansible [core 2.16.6]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Apr 10 2024, 05:33:47) [GCC 13.2.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
```

PING A MAQUINA DE PRODUCCION

```
# Agregar a /etc/hosts la ip de la maquina de produccion
sudo nano /etc/hosts
```




```
ping prod
```

PROF

```
ubuntu@ip-172-31-34-131:~$ ping prod
PING prod (3.16.10.131) 56(84) bytes of data.
64 bytes from prod (3.16.10.131): icmp_seq=1 ttl=63 time=0.506 ms
64 bytes from prod (3.16.10.131): icmp_seq=2 ttl=63 time=0.599 ms
64 bytes from prod (3.16.10.131): icmp_seq=3 ttl=63 time=0.543 ms
64 bytes from prod (3.16.10.131): icmp_seq=4 ttl=63 time=0.547 ms
64 bytes from prod (3.16.10.131): icmp_seq=5 ttl=63 time=0.572 ms
64 bytes from prod (3.16.10.131): icmp_seq=6 ttl=63 time=0.492 ms
64 bytes from prod (3.16.10.131): icmp_seq=7 ttl=63 time=0.548 ms
^C
```

Configuración de ANSIBLE

creamos el playbook y con el comando siguiente lo hacemos correr

```
```

```
ansible-playbook playBook.yaml
```