

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**  
**FACULTAD DE INGENIERIA**  
**LAB. SOFTWARE AVANZADO**



**MANUAL DE USO Y CONFIGURACIÓN DE ANSIBLE**

**HECHO POR:**

**MADLINE ARIANA PÉREZ QUIÑONEZ**  
**ERICK ALEXANDER LEMUS MORALES**



## Tabla de contenido

<b>Requisitos .....</b>	<b>3</b>
<b>Configuración de servidor de producción.....</b>	<b>3</b>
<b>Configuración de entorno de Ansible.....</b>	<b>5</b>
Generación de clave SSH y copia en servidor de producción .....	5
Instalar Ansible .....	6
Configuración de hosts .....	6
<b>Prueba con Ansible.....</b>	<b>7</b>
<b>Ejemplo con ansible.cfg, inventarios y playbooks.....</b>	<b>8</b>
Configuración de ansible.cfg .....	8
Crear playbook .....	9
<b>Módulos de ansible.builtin.....</b>	<b>9</b>

## Requisitos

1. Crear una máquina virtual que será utilizada para ansible
2. Crear una máquina virtual que será utilizada para producción

<input type="checkbox"/>	productiontestlab	i-04e9cd7426051b891	Running	🔍🔍	t2.micro	2/2 checks passed
<input type="checkbox"/>	ansiblelabtest	i-08b96e03bbb2ba1fb	Running	🔍🔍	t2.micro	2/2 checks passed

## Configuración de servidor de producción

1. Ingresar a la máquina que será utilizada para producción
  - a. `ssh -i archivo.pem ubuntu@ip_servidor`
2. Crear un nuevo usuario llamado **ansible**
  - a. `sudo adduser ansible`
  - b. Agregar contraseña y llenar los campos de valores si se desea.

```
ubuntu@ip-10.10.10.10:~$ sudo adduser ansible
Adding user `ansible' ...
Adding new group `ansible' (1003) ...
Adding new user `ansible' (1003) with group `ansible' ...
Creating home directory `/home/ansible' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ansible
Enter the new value, or press ENTER for the default
[Full Name []:
[Room Number []:
[Work Phone []:
[Home Phone []:
[Other []:
[Is the information correct? [Y/n] Y
```

3. Editar el archivo `/etc/ssh/sshd_config` (utilizar el editor de su preferencia: nano, vi, vim)
  - a. `sudo vi /etc/ssh/sshd_config`
  - b. Verificar los siguientes campos esten de la siguiente manera
    - i. AuthorizedKeysFile

```
# Expect .ssh/authorized_keys2 to be disregarded by default in future.
AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
```

- ii. PasswordAuthentication yes

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
```

4. Editar **sudoers**

- a. sudo visudo
- b. Agregar el usuario creado con los permisos ALL

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

5. Reiniciar servicio sshd

- a. sudo systemctl restart sshd

6. Verificar si se puede iniciar sesion con el usuario creado

```
eclem1998@MacBook-Pro-de-Erick Software Avanzado % ssh ansible@34.201.137.30
ansible@34.201.137.30's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

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

System information as of Tue Sep 19 01:42:06 UTC 2023

System load:  0.0               Processes:            99
Usage of /:   16.2% of 14.36GB   Users logged in:     0
Memory usage: 30%              IPv4 address for docker0: 172.17.0.1
Swap usage:   0%                IPv4 address for eth0:  172.31.42.117

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

128 updates can be applied immediately.
68 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

Last login: Tue Sep 19 01:42:07 2023 from 190.104.126.96
ansible@ip-172-31-42-117:~$
```

7. Si hasta este paso no se ha conseguido el resultado se debe revisar el proceso, sino se puede continuar a la configuración del servidor de ansible.

## Configuración de entorno de Ansible

Iniciar sesión en el servidor en el cual se instalará ansible y continuar con los siguientes grupos de instrucciones.

### Generación de clave SSH y copia en servidor de producción

1. Generar la clave SSH en el servidor de ansible (se almacena en la ruta ~/.ssh).
  - a. ssh-keygen

```
ubuntu@ip-172-31-40-12:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
/home/ubuntu/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:rcSM095+ca70Mqr6ij4m/4EjU03WkZ0j5m9aPJF4L8 ubuntu@ip-172-31-40-12
The key's randomart image is:
+----[RSA 3072]-----+
|  o .                    |
| + B                     |
| . B +                   |
| + = = .                 |
| + * = S .               |
| . X . + . . .           |
| o + o E . . +           |
| o . o o . . + .         |
| =+o++o+o+.             |
+----[SHA256]-----+
```

2. Copiar la clave SSH al servidor de producción
  - a. ssh-copy-id **ansible@ip\_maquina\_prod**
  - b. Ingresar la clave del usuario de ansible.

```
ubuntu@ip-172-31-40-12:~$ ssh-copy-id ansible@172.31.42.117
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ubuntu/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansible@172.31.42.117's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ansible@172.31.42.117'"
and check to make sure that only the key(s) you wanted were added.

ubuntu@ip-172-31-40-12:~$
```

**Nota:** como AWS maneja las IP privadas, preferiblemente utilizarla ya que si se apaga la maquina virtual la IP pública cambia, y eso significaría cambiar otros archivos mas adelante.

3. Si existe un error, verificar los pasos de preparación de maquina de produccion.



## Instalar Ansible

Ansible puede ser instalado por medio de la paquetería del sistema operativo, o por medio de python. En este documento se hará por medio del repositorio de ubuntu (todo es extraído de la documentación oficial de ansible).

### PAQUETERIA DE UBUNTU

1. sudo apt update
2. sudo apt install software-properties-common
3. sudo add-apt-repository --yes --update ppa:ansible/ansible
4. sudo apt install ansible
5. ansible --version

```
[ubuntu@ip-172-31-40-12:~]$ ansible --version
ansible [core 2.15.4]
  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 = 3.10.6 (main, Mar 10 2023, 10:55:28) [GCC 11.3.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
[ubuntu@ip-172-31-40-12:~]$
```

### Configuración de hosts

Ansible es una herramienta que se conecta por medio de ssh, por lo tanto es importante configurar los hosts para hacer una referencia corta en los inventarios.

1. Editar el archivo /etc/hosts (se puede con vi, vim, nano, etc)
  - a. sudo vi /etc/hosts
  - b. Agregar la siguiente línea al final
    - i. IP\_maquina nombreReferencia
  - c. Guardar los cambios

```
127.0.0.1 localhost
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
172.31.42.117 prod
```

2. Realizar un ping a la maquina de produccion para verificar la efectividad de los pasos realizados
  - a. ping prod1

```
ubuntu@ip-172-31-40-12:~$ ping prod1
PING prod1 (172.31.42.117) 56(84) bytes of data:
64 bytes from prod1 (172.31.42.117): icmp_seq=1 ttl=64 time=0.534 ms
64 bytes from prod1 (172.31.42.117): icmp_seq=2 ttl=64 time=1.96 ms
64 bytes from prod1 (172.31.42.117): icmp_seq=3 ttl=64 time=0.498 ms
64 bytes from prod1 (172.31.42.117): icmp_seq=4 ttl=64 time=0.530 ms
64 bytes from prod1 (172.31.42.117): icmp_seq=5 ttl=64 time=0.659 ms
64 bytes from prod1 (172.31.42.117): icmp_seq=6 ttl=64 time=5.71 ms
^C
--- prod1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5075ms
rtt min/avg/max/mdev = 0.498/1.648/5.709/1.887 ms
```

3. Si hasta este paso todo es correcto, se puede pasar a realizar la prueba ejecutando ansible.

### Prueba con Ansible

El objetivo de los pasos anteriores es dejar listo el camino para ejecutar ansible. Para este caso se ejecutará una prueba sencilla utilizando el módulo de ping que ya es un módulo que ofrece Ansible.

1. Crear una carpeta para trabajar
  - a. mkdir ansibleEjemplo
  - b. cd ansibleEjemplo
2. Crear un inventario que contiene todos los grupos y subgrupos de maquina a los que nos queremos conectar y ejecutar instrucciones automatizadas con Ansible.
  - a. Ejemplo utilizando hosts
    - i. sudo vi hosts

```
[produccion]
prod1

[produccion:vars]
ansible_become=true
ansible_become_pass=ansible
ansible_user=ansible
```

- b. Ejemplo utilizando hosts.yaml
    - i. sudo vi hosts.yaml

```
produccion:
  hosts:
    prod1:
  vars:
    ansible_become: true
    ansible_become_pass: ansible
    ansible_user: ansible
...
```

### 3. Ejecutar el modulo ping utilizando el Ansible CLI

- a. `ansible -i hosts all -m ping`

```
ubuntu@ip-172-31-40-12:~$ ansible -i hosts all -m ping
prod1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

- b. `ansible -i hosts.yaml all -m ping`

```
ubuntu@ip-172-31-40-12:~$ ansible -i hosts.yaml all -m ping
prod1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

### 4. Si las pruebas anteriores dan los mismos resultados, es un indicador que ya está listo ansible.

#### Ejemplo con ansible.cfg, inventarios y playbooks

En este ejemplo haremos el mismo ejercicio anterior, pero orientado a los elementos del título. Reutilizaremos el archivo hosts.yaml del ejemplo anterior:

#### Configuración de ansible.cfg

El objetivo de este archivo es especificar configuraciones que puedan ser útiles en la ejecución de ansible.

1. Crear un archivo **ansible.cfg**
  - a. `vi ansible.cfg`
  - b. Insertar el siguiente contenido

```
[defaults]
# Indica que inventario se va a utilizar
inventory = hosts.yaml
# Indica que no verifique hosts
host_key_checking = False
```

2. Guardar el contenido



## Crear playbook

El objetivo de este archivo es indicar una serie de instrucciones que se ejecutarán por medio de módulos que ofrece ansible, entre estos estan: ping, file, instalacion de software, entre muchos más.

Un playbook está compuesto de uno o más plays que no son mas que instrucciones.

1. Crear un playbook.
  - a. vi miPlaybook.yaml
  - b. Insertar el siguiente contenido

```
hosts: produccion
tasks:
  - name: hacer ping
    ansible.builtin.ping:
```

2. Ejecutar el playbook
  - a. ansible-playbook miPlaybook.yaml

```
ubuntu@ip-172-31-40-12:~$ ansible-playbook miPlaybook.yaml
PLAY [produccion] *****
TASK [Gathering Facts] *****
ok: [prod1]
TASK [hacer ping] *****
ok: [prod1]
PLAY RECAP *****
prod1 : ok=2  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

3. Si el resultado es satisfactorio, ya se puede utilizar cualquier modulo.

## Módulos de ansible.builtin

<https://docs.ansible.com/ansible/latest/collections/ansible/builtin/index.html>