

LABORATORIO ANSIBLE

Nombre: Boris Vargas Paucara

Objetivo

El objetivo de esta práctica es el de automatizar el despliegue de un servidor web + una página sencilla en un servidor externo mediante la herramienta Ansible (utilizando ssh)

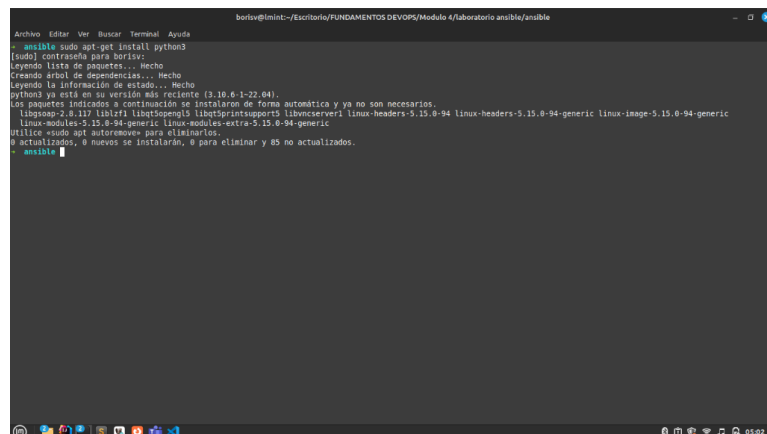
Resolución

Para esta práctica tendremos como requisito tener por un lado nuestro laboratorio principal (ya sea de forma local o una máquina virtual) y una segunda máquina Linux a la cual tengamos acceso por ssh desde nuestro laboratorio principal, esto es fundamental ya que ansible utilizar ssh de fondo para enviar las instrucciones. En nuestro caso, utilizaremos la máquina en AWS creada en el laboratorio de Terraform, pero es indistinta la ubicación de la máquina.

Empezaremos por la instalación de Ansible:

1) Primero, tenemos que asegurarnos de que tenemos python3 instalado, en caso de no tenerlo lo instalamos con

a) sudo apt-get install python3

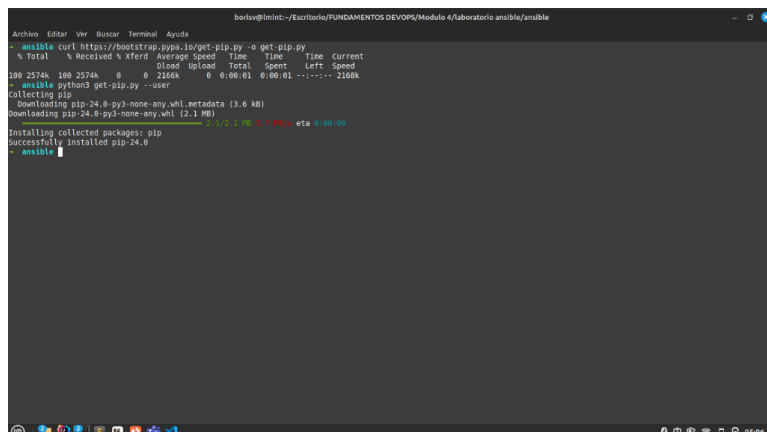


```
boris@imint:~/Escritorio/FUNDAMENTOS DEVOPS/Modulo 4/laboratorio ansible/ansible$
- ansible sudo apt-get install python3
[sudo] contraseña para boris:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
python3 ya está en su versión más reciente (3.10.6-1-22.04).
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
libasp-2.9.127 libelf1 libelf-dev libelf-dev:amd64 libelf-dev:arm64 libelf-dev:ppc64el libelf-dev:s390x
linux-modules-5.15.0-94-generic linux-modules-extra-5.15.0-94-generic
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 85 no actualizados.
- ansible
```

2) Una vez con python3 instalado, instalamos pip

a) curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py

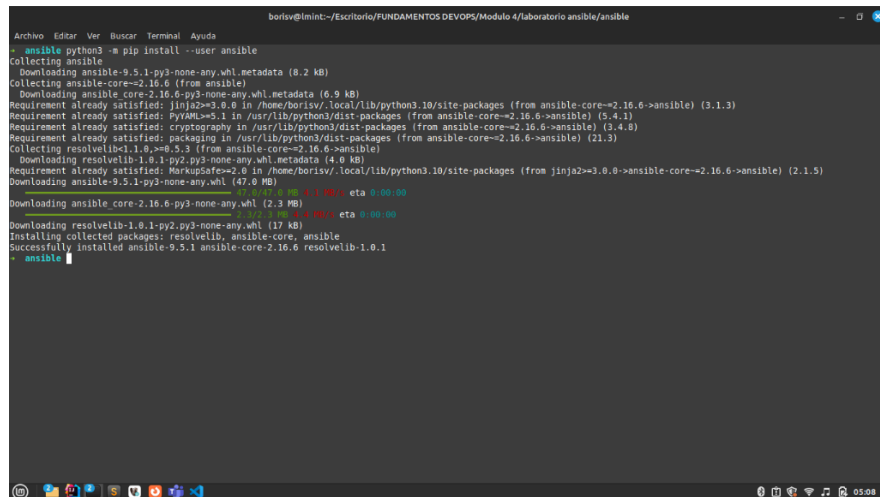
b) python3 get-pip.py --user



```
boris@imint:~/Escritorio/FUNDAMENTOS DEVOPS/Modulo 4/laboratorio ansible/ansible$
- ansible curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
% Total % Received % Xferd Average Speed Time Time Time Current
% Total % Received % Xferd Average Speed Time Time Time Current
100 2574k 100 2574k 0 0 2168k 0 0:00:01 0:00:01 --:--:-- 2168k
- ansible python3 get-pip.py --user
Collecting pip
  Downloading pip-24.0-py3-none-any.whl.metadata (3.6 kB)
  Downloading pip-24.0-py3-none-any.whl (2.1 MB)
    2.1 MB 4.7 MB/s eta 0:00:00
Installing collected packages: pip
Successfully installed pip-24.0
- ansible
```

3) Una vez que tengamos pip instalado, instalamos ansible

a) `python3 -m pip install --user ansible`



```
borisv@lmint:~/Escritorio/FUNDAMENTOS DEVOPS/Modulo 4/laboratorio ansible/ansible
- ansible python3 -m pip install --user ansible
Collecting ansible
  Downloading ansible-9.5.1-py3-none-any.whl.metadata (8.2 kB)
Collecting ansible-core==2.16.6 (from ansible)
  Downloading ansible-core-2.16.6-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: Jinja2>=3.0.0 in /home/borisv/.local/lib/python3.10/site-packages (from ansible-core==2.16.6->ansible) (3.1.3)
Requirement already satisfied: PyYAML>=5.1 in /usr/lib/python3/dist-packages (from ansible-core==2.16.6->ansible) (5.4.1)
Requirement already satisfied: cryptography in /usr/lib/python3/dist-packages (from ansible-core==2.16.6->ansible) (3.4.8)
Requirement already satisfied: packaging in /usr/lib/python3/dist-packages (from ansible-core==2.16.6->ansible) (21.3)
Collecting resolvelib<1.1.0,>=0.5.3 (from ansible-core==2.16.6->ansible)
  Downloading resolvelib-1.0.1-py2.py3-none-any.whl.metadata (4.0 kB)
Requirement already satisfied: MarkupSafe>=2.0 in /home/borisv/.local/lib/python3.10/site-packages (from Jinja2>=3.0.0->ansible-core==2.16.6->ansible) (2.1.5)
Downloading ansible-9.5.1-py3-none-any.whl (47.6 MB)
  1.7 MB/s eta 0:00:00
Downloading ansible-core-2.16.6-py3-none-any.whl (2.3 MB)
  3.3 MB/s eta 0:00:00
Downloading resolvelib-1.0.1-py2.py3-none-any.whl (17 kB)
Installing collected packages: resolvelib, ansible-core, ansible
Successfully installed ansible-9.5.1 ansible-core-2.16.6 resolvelib-1.0.1
- ansible
```

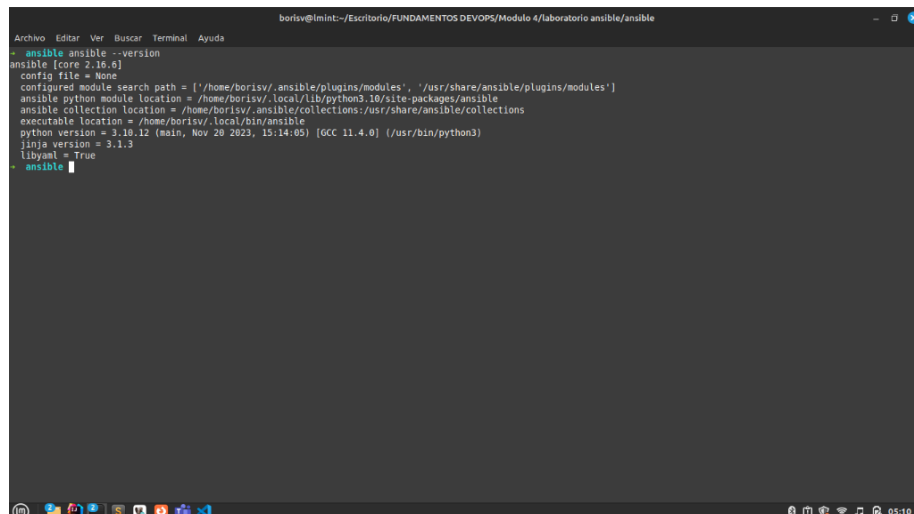
4) El último paso es añadir el path donde instalamos ansible a nuestro path de ejecutables, una forma de hacer esto es agregar el path que nos da el output de la instalación a nuestro `.bashrc` (o archivo de configuración de la shell que utilicemos)

Para esto, agregaremos la siguiente línea al final del archivo `.bashrc`

- a) `export PATH="/home/rapaza/.local/bin:$PATH"` (reemplazar el usuario)
- b) `source .bashrc`

5) Y listo, ya tendremos ansible listo para utilizar, para verificar su correcto funcionamiento corremos:

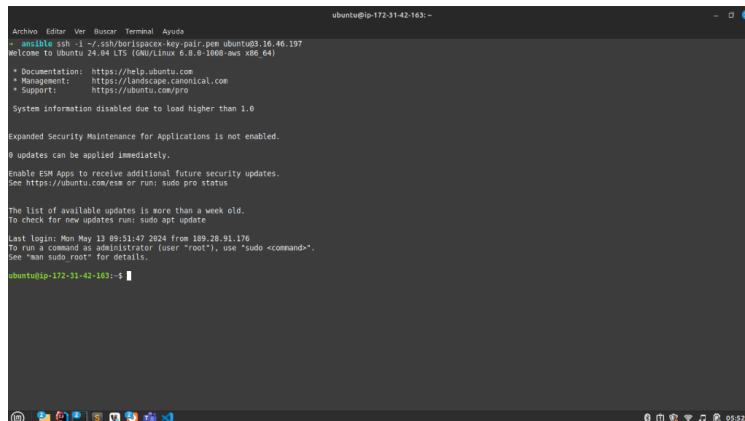
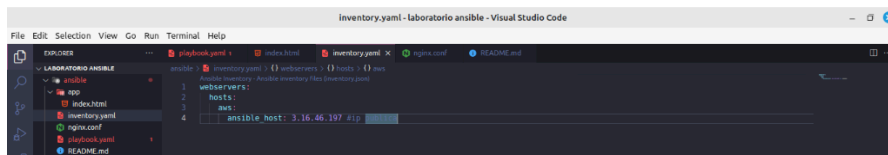
a) `ansible --version`



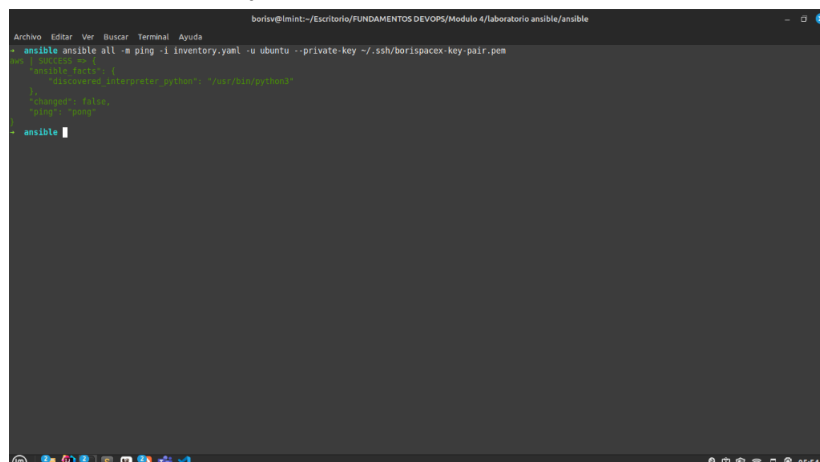
```
borisv@lmint:~/Escritorio/FUNDAMENTOS DEVOPS/Modulo 4/laboratorio ansible/ansible
- ansible ansible --version
ansible [core 2.16.6]
  config file = None
  configured module search path = ['/home/borisv/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /home/borisv/.local/lib/python3.10/site-packages/ansible
  ansible collection location = /home/borisv/.ansible/collectons:/usr/share/ansible/collectons
  executable location = /home/borisv/.local/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.1.3
  libyaml = True
- ansible
```

Ya con Ansible instalado, ahora crearemos un inventario donde configuraremos

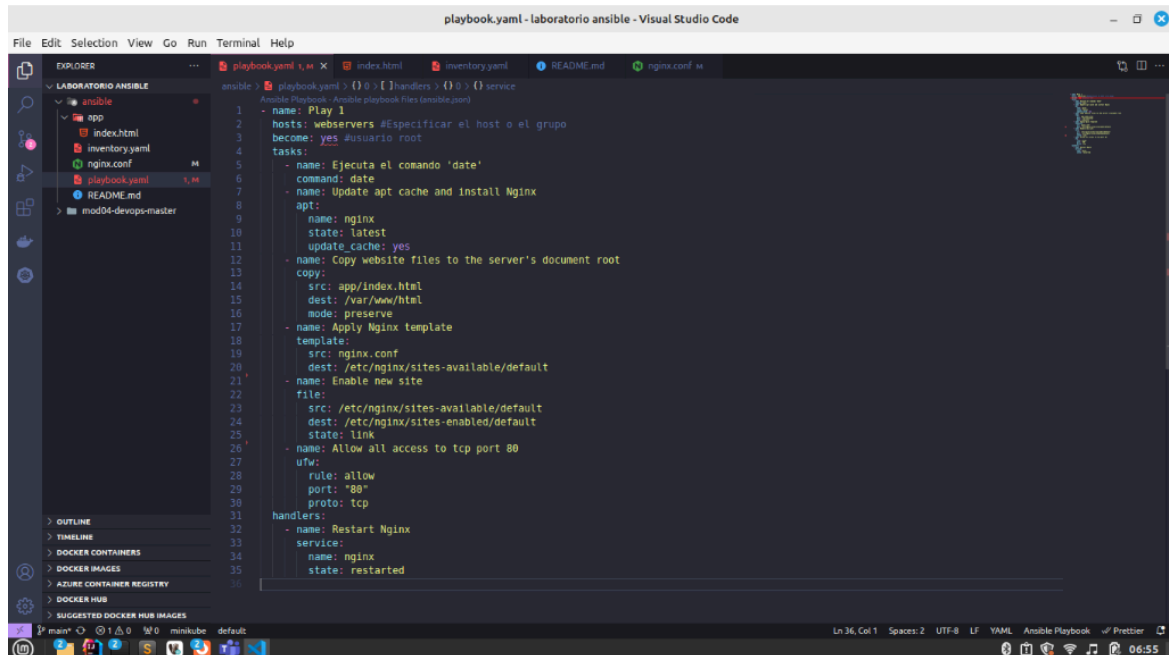
6) Para esto, crearemos un archivo de configuración con terminación `.yaml`, en nuestro caso lo llamamos `inventario.yaml` aunque el nombre es indistinto. En nuestro caso, de nuevo, estamos utilizando una máquina virtual que corre en AWS por eso la IP pública y no una IP privada de una red 192.168.x.x.



```
a) ansible all -m ping -i inventario.yaml -u ubuntu --private-key
../Terraform/AWS/terra-keys
```



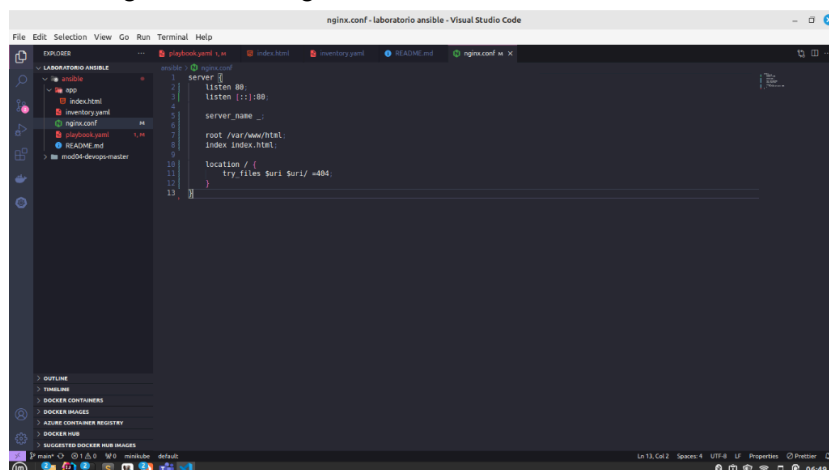
8) Ya creado el inventario y verificado el acceso a través de Ansible, pasaremos a crear el playbook.



```
1 - name: Play 1
2   hosts: webserver
3   become: yes
4   tasks:
5     - name: Ejecuta el comando 'date'
6       command: date
7     - name: Update apt cache and install Nginx
8       apt:
9         name: nginx
10        state: latest
11        update_cache: yes
12    - name: Copy website files to the server's document root
13      copy:
14        src: app/index.html
15        dest: /var/www/html
16        mode: preserve
17    - name: Apply Nginx template
18      template:
19        src: nginx.conf
20        dest: /etc/nginx/sites-available/default
21    - name: Enable new site
22      file:
23        src: /etc/nginx/sites-available/default
24        dest: /etc/nginx/sites-enabled/default
25        state: link
26    - name: Allow all access to tcp port 80
27      ufw:
28        rule: allow
29        port: "80"
30        proto: tcp
31    handlers:
32    - name: Restart Nginx
33      service:
34        name: nginx
35        state: restarted
```

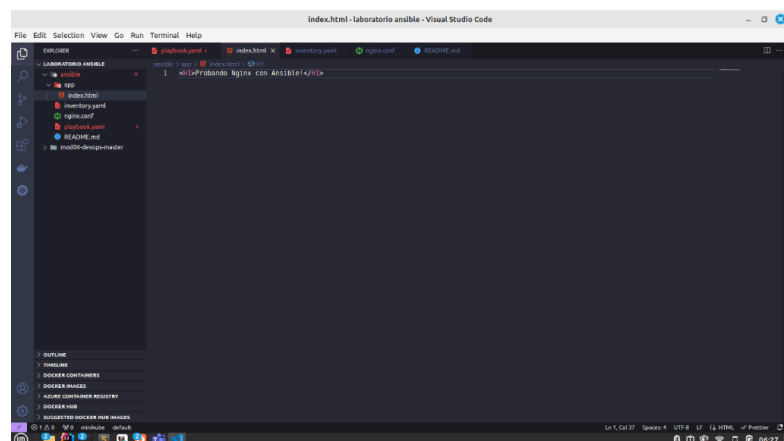
Adicional al playbook, necesitamos crear 2 archivos más:

- Un archivo de configuración de nginx



```
1 server {
2   listen 80;
3   listen :::80;
4
5   server_name _;
6
7   root /var/www/html;
8   index index.html;
9
10  location / {
11    try_files $uri $uri/ =404;
12  }
13 }
```

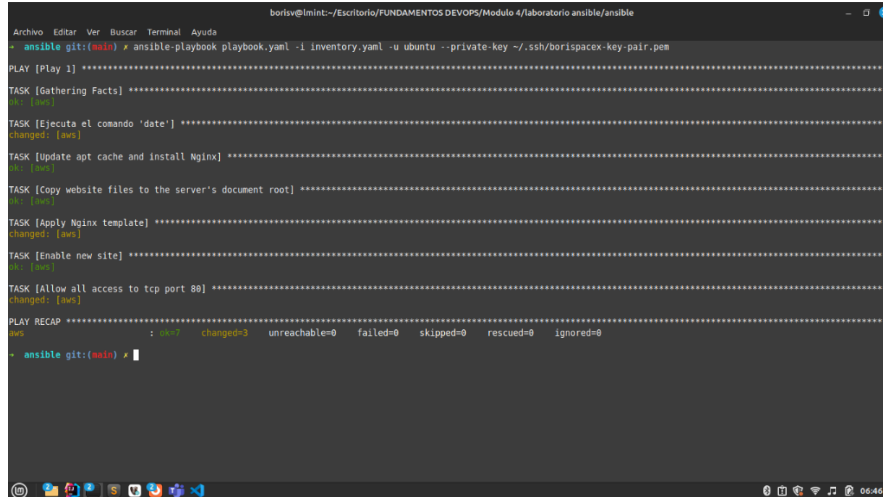
- Un directorio llamado app que dentro tenga los archivos para nuestro servidor web, en este caso solo vamos a usar un index.html bastante sencillo con un H1.



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Prueba Nginx con Ansible</title>
6 </head>
7 <body>
8 <h1>Prueba Nginx con Ansible</h1>
9 </body>
10</html>
```

9) Ya con el playbook y los archivos adicionales creados, pasaremos a ejecutar el playbook:

a) `ansible-playbook playbook.yaml -i inventario.yaml -u ubuntu --private-key ../Terraform/AWS/terra-keys`



```
boris@mint:~/Escritorio/FUNDAMENTOS DEVOPS/Modulo 4/laboratorio ansible/ansible
- ansible git:(main) x ansible-playbook playbook.yaml -i inventario.yaml -u ubuntu --private-key ../ssh/borispacex-key-pair.pem
PLAY [Play 1] *****
TASK [Gathering Facts] *****
ok: [aws]
changed: [aws]
TASK [Ejecuta el comando 'date'] *****
ok: [aws]
TASK [Update apt cache and install Nginx] *****
ok: [aws]
TASK [Copy website files to the server's document root] *****
ok: [aws]
changed: [aws]
TASK [Apply Nginx template] *****
changed: [aws]
TASK [Enable new site] *****
ok: [aws]
TASK [Allow all access to tcp port 88] *****
changed: [aws]
PLAY RECAP *****
aws : ok=1 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
- ansible git:(main) x
```

10) Una vez aplicado el playbook, solo es cuestión de acceder a la IP pública de nuestra máquina virtual en AWS para ver nuestra aplicación web corriendo



SOLUCIÓN

Link: <http://3.16.46.197/>

Repositorio: <https://github.com/borispacex/lab-ansible-udi>