INSTALACIÓN Y DESPLIEGUE DE ANSIBLE

PABLO HORCAJADA GONZALEZ y PABLO BÉJAR THOMAS

Profesor: Guillermo Bellettini

Clase : Despliegue Web

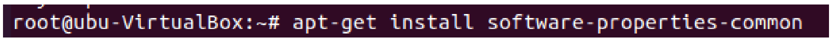
Fecha 13/01/2022

Contenido

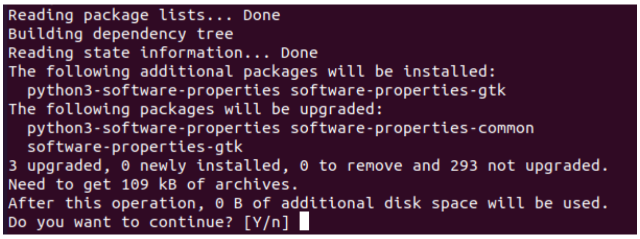
Instalación Ansible

Ubuntu

Instalar el paquete “software-properties-common” para crear repositorios externos



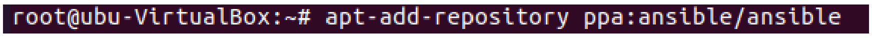
```
root@ubu-VirtualBox:~# apt-get install software-properties-common
```



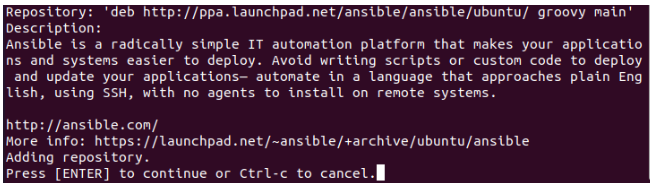
```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-software-properties software-properties-gtk
The following packages will be upgraded:
  python3-software-properties software-properties-common
  software-properties-gtk
3 upgraded, 0 newly installed, 0 to remove and 293 not upgraded.
Need to get 109 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] 
```

-Pulsaremos la tecla “y” e intro para continuar y terminar la instalación.

Creamos el repositorio para Ansible

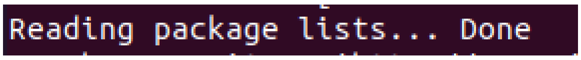


```
root@ubu-VirtualBox:~# apt-add-repository ppa:ansible/ansible
```



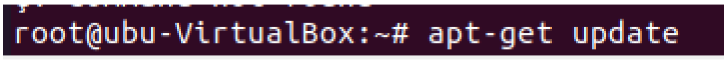
```
Repository: 'deb http://ppa.launchpad.net/ansible/ansible/ubuntu/ groovy main'
Description:
Ansible is a radically simple IT automation platform that makes your applicatio
ns and systems easier to deploy. Avoid writing scripts or custom code to deploy
 and update your applications— automate in a language that approaches plain Eng
lish, using SSH, with no agents to install on remote systems.

http://ansible.com/
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
```

-Presionamos “enter” para continuar y terminar la creación de repositorios.



```
Reading package lists... Done
```

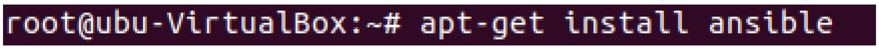
Actualizamos la lista de paquetes *



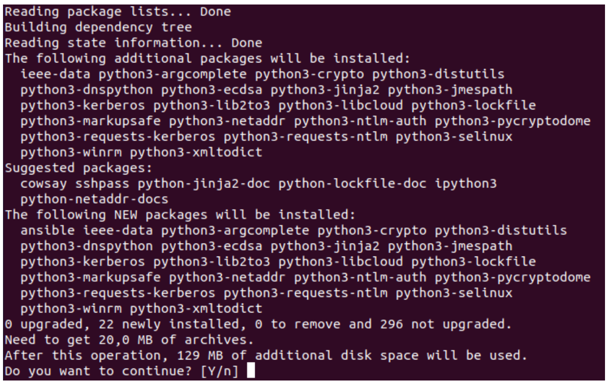
```
root@ubu-VirtualBox:~# apt-get update
```

-En CentOS se actualiza solo.

Instalamos Ansible

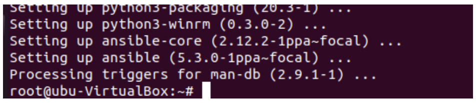


```
root@ubu-VirtualBox:~# apt-get install ansible
```



```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ieee-data python3-argcomplete python3-crypto python3-distutils
  python3-dnspython python3-ecdsa python3-jinja2 python3-jmespath
  python3-kerberos python3-lib2to3 python3-libcloud python3-lockfile
  python3-markupsafe python3-netaddr python3-ntlm-auth python3-pycryptodome
  python3-requests-kerberos python3-requests-ntlm python3-selinux
  python3-winrm python3-xmltodict
Suggested packages:
  cowsay sshpass python-jinja2-doc python-lockfile-doc ipython3
  python-netaddr-docs
The following NEW packages will be installed:
  ansible ieee-data python3-argcomplete python3-crypto python3-distutils
  python3-dnspython python3-ecdsa python3-jinja2 python3-jmespath
  python3-kerberos python3-lib2to3 python3-libcloud python3-lockfile
  python3-markupsafe python3-netaddr python3-ntlm-auth python3-pycryptodome
  python3-requests-kerberos python3-requests-ntlm python3-selinux
  python3-winrm python3-xmltodict
0 upgraded, 22 newly installed, 0 to remove and 296 not upgraded.
Need to get 20,0 MB of archives.
After this operation, 129 MB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

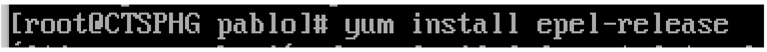
-Presionamos la tecla “y” para continuar y acabar la instalación



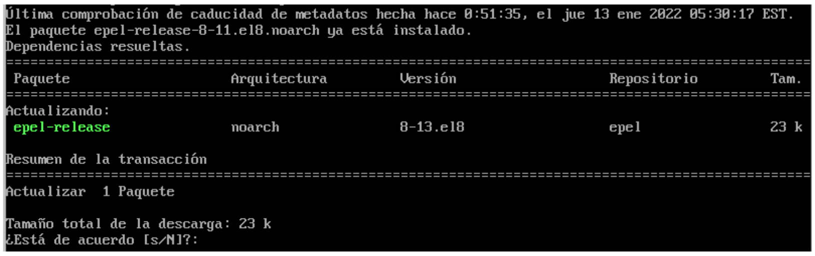
```
Setting up python3-packaging (20.3-1) ...
Setting up python3-winrm (0.3.0-2) ...
Setting up ansible-core (2.12.2-1ppa~focal) ...
Setting up ansible (5.3.0-1ppa~focal) ...
Processing triggers for man-db (2.9.1-1) ...
root@ubu-VirtualBox:~# 
```

CentOS

Instalamos el repositorio epel-release

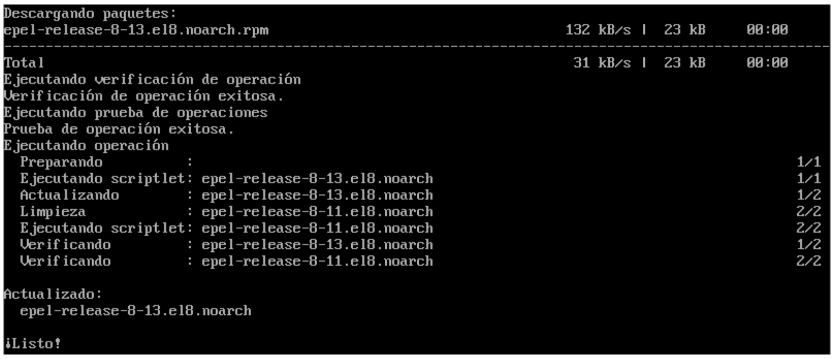


```
[root@CTSPHG pablo]# yum install epel-release
```



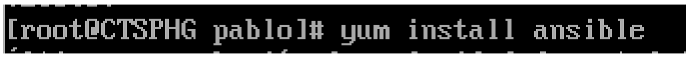
```
Última comprobación de caducidad de metadatos hecha hace 0:51:35, el jue 13 ene 2022 05:30:17 EST.
El paquete epel-release-8-11.el8.noarch ya está instalado.
Dependencias resueltas.
================================================================================================
 Paquete                    Arquitectura         Versión                   Repositorio        Tam.
================================================================================================
Actualizando:
 epel-release               noarch               8-13.el8                  epel               23 k

Resumen de la transacción
================================================================================================
Actualizar  1 Paquete

Tamaño total de la descarga: 23 k
¿Está de acuerdo [s/N]?:
```

-Presionamos la tecla “s” para continuar y terminar la instalación.

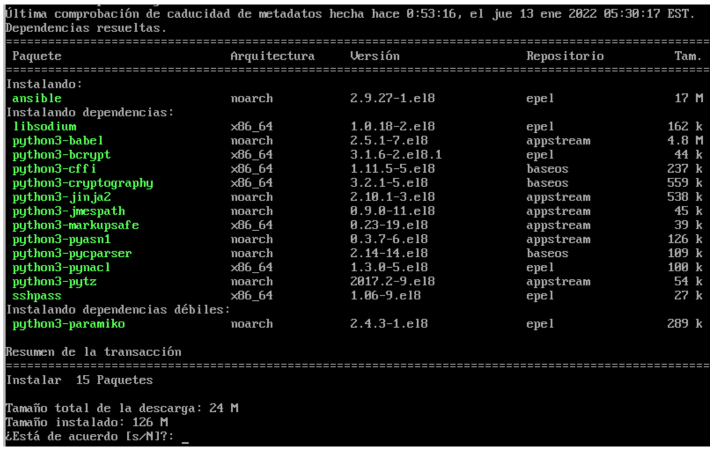


```
Descargando paquetes:
epel-release-8-13.el8.noarch.rpm                                   132 kB/s |  23 kB     00:00
--------------------------------------------------------------------------------------------------
Total                                                               31 kB/s |  23 kB     00:00
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
  Preparando           :                                                                      1/1
  Ejecutando scriptlet: epel-release-8-13.el8.noarch                                         1/1
  Actualizando         : epel-release-8-13.el8.noarch                                         1/2
  Limpieza             : epel-release-8-11.el8.noarch                                         2/2
  Ejecutando scriptlet: epel-release-8-11.el8.noarch                                         2/2
  Verificando          : epel-release-8-13.el8.noarch                                         1/2
  Verificando          : epel-release-8-11.el8.noarch                                         2/2

Actualizado:
  epel-release-8-13.el8.noarch

¡Listo!
```

Instalamos el Ansible

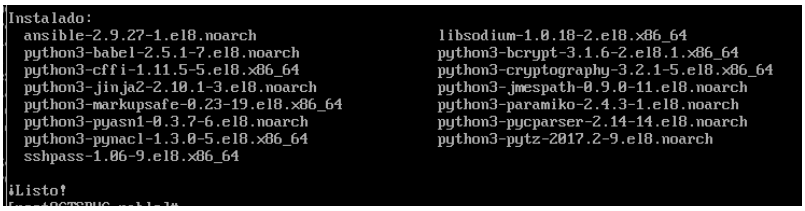


```
[root@CTSPHG pablo]# yum install ansible
```



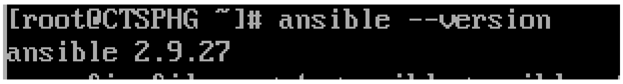
```
Última comprobación de caducidad de metadatos hecha hace 0:53:16, el jue 13 ene 2022 05:30:17 EST.
Dependencias resueltas.
=========================================================================================================
 Paquete                       Arquitectura      Versión                   Repositorio            Tam.
=========================================================================================================
Instalando:
 ansible                       noarch            2.9.27-1.el8              epel                   17 M
Instalando dependencias:
 libsodium                     x86_64            1.0.18-2.el8              epel                  162 k
 python3-babel                 noarch            2.5.1-7.el8               appstream             4.8 M
 python3-bcrypt                x86_64            3.1.6-2.el8.1             epel                   44 k
 python3-cffi                  x86_64            1.11.5-5.el8              baseos                237 k
 python3-cryptography          x86_64            3.2.1-5.el8               baseos                559 k
 python3-jinja2                noarch            2.10.1-3.el8              appstream             538 k
 python3-jmespath              noarch            0.9.0-11.el8              appstream              45 k
 python3-markupsafe            x86_64            0.23-19.el8               appstream              39 k
 python3-pyasn1                noarch            0.3.7-6.el8               appstream             126 k
 python3-pycparser             noarch            2.14-14.el8               baseos                109 k
 python3-pynacl                x86_64            1.3.0-5.el8               epel                  100 k
 python3-pytz                  noarch            2017.2-9.el8              appstream              54 k
 sshpass                       x86_64            1.06-9.el8                epel                   27 k
Instalando dependencias débiles:
 python3-paramiko              noarch            2.4.3-1.el8               epel                  289 k

Resumen de la transacción
=========================================================================================================
Instalar  15 Paquetes

Tamaño total de la descarga: 24 M
Tamaño instalado: 126 M
¿Está de acuerdo [s/N]?: _
```

-Presionamos la tecla “s” para continuar y esperamos que termine la instalación.



```
Instalado:
  ansible-2.9.27-1.el8.noarch                          libsodium-1.0.18-2.el8.x86_64
  python3-babel-2.5.1-7.el8.noarch                     python3-bcrypt-3.1.6-2.el8.1.x86_64
  python3-cffi-1.11.5-5.el8.x86_64                     python3-cryptography-3.2.1-5.el8.x86_64
  python3-jinja2-2.10.1-3.el8.noarch                   python3-jmespath-0.9.0-11.el8.noarch
  python3-markupsafe-0.23-19.el8.x86_64                python3-paramiko-2.4.3-1.el8.noarch
  python3-pyasn1-0.3.7-6.el8.noarch                    python3-pycparser-2.14-14.el8.noarch
  python3-pynacl-1.3.0-5.el8.x86_64                    python3-pytz-2017.2-9.el8.noarch
  sshpass-1.06-9.el8.x86_64

¡Listo!
```

Comprobamos el contenido instalado

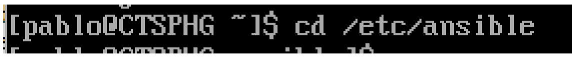


```
[root@CTSPHG ~]# ansible --version
ansible 2.9.27
```

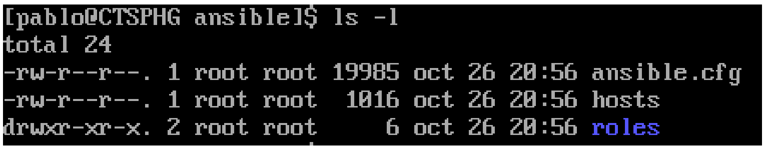
Despliegue con Ansible

Configuración archivo host de manera local

-Entramos al directorio de Ansible y listamos el contenido

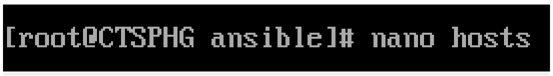


```
[pablo@CTSPHG ~]$ cd /etc/ansible
```

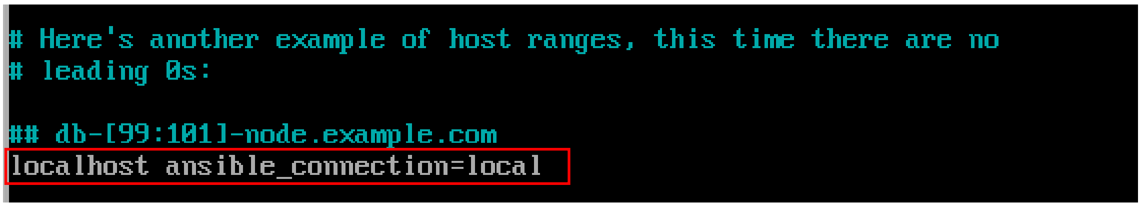


```
[pablo@CTSPHG ansible]$ ls -l
total 24
-rw-r--r--. 1 root root 19985 oct 26 20:56 ansible.cfg
-rw-r--r--. 1 root root  1016 oct 26 20:56 hosts
drwxr-xr-x. 2 root root     6 oct 26 20:56 roles
```

-Editamos el archivo "hosts" y añadimos la siguiente línea al final del archivo.

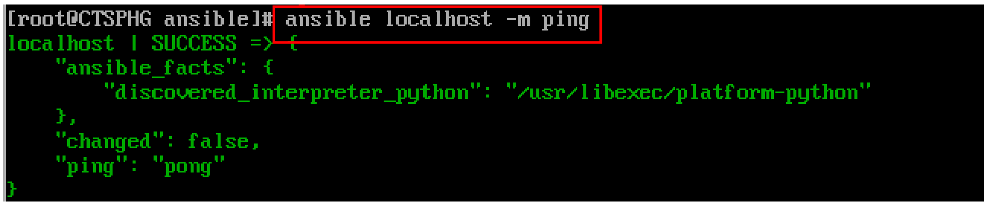


```
[root@CTSPHG ansible]# nano hosts
```



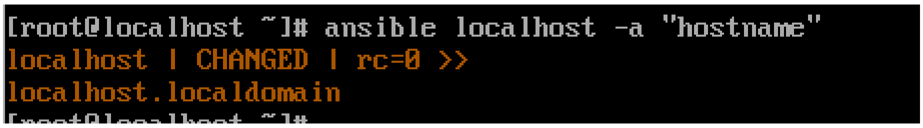
```
# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com
localhost ansible_connection=local
```

-Comprobamos que hemos añadido lo anterior bien haciendo un ping (al salir verde indica que funciona correctamente, se ha conectado y nos ha respondido con un "pong").



```
[root@CTSPHG ansible]# ansible localhost -m ping
localhost | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
```

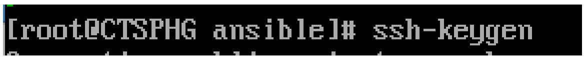
-Ejecutamos un comando con ansible



```
[root@localhost ~]# ansible localhost -a "hostname"
localhost | CHANGED | rc=0 >>
localhost.localdomain
[root@localhost ~]#
```

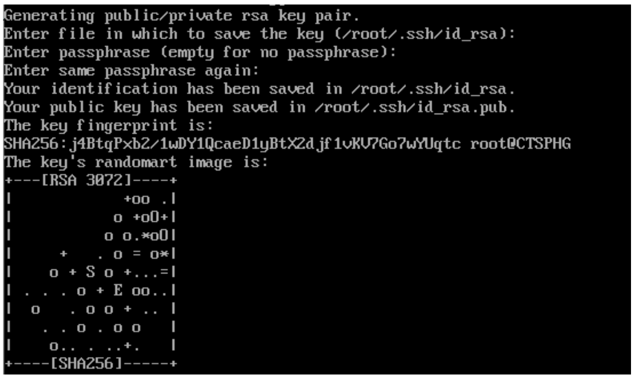
Conexión a los servidores de manera remota

Generamos la clave pública



```
[root@CTSPHG ansible]# ssh-keygen
```

-Pulsaremos a “enter” para confirmar todo y crear la clave

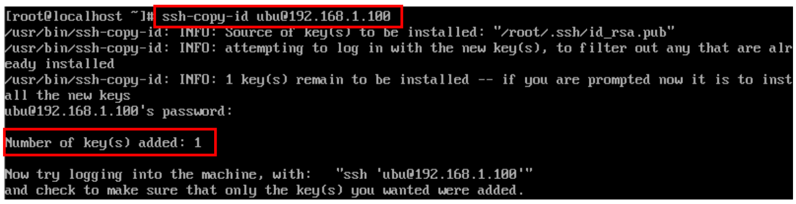


```
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:j4BtqPxb2/1wDY1QcaeD1yBtX2djf1vKV7Go7wYUqtc root@CTSPHG
The key's randomart image is:
+---[RSA 3072]----+
|            +oo .|
|           o +oO+|
|          o o.*oO|
|     +    . o = o*|
|    o + S o +...=|
| . . . o + E oo..|
|  o    . o o + .. |
|   . . o . o o   |
|    o.. . ..+.   |
+----[SHA256]-----+
```

Copiamos la clave a los servidores

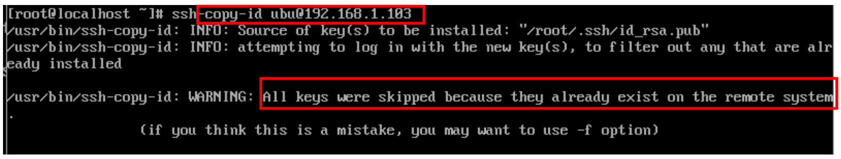
-Usuario1 de Ubuntu es ubu y la ip es 192.168.1.100

-Nos pedirá la contraseña del servidor



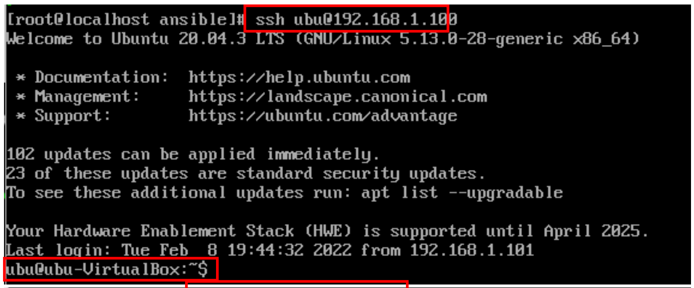
```
[root@localhost ~]# ssh-copy-id ubu@192.168.1.100
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
ubu@192.168.1.100's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ubu@192.168.1.100'"
and check to make sure that only the key(s) you wanted were added.
```

-Repetimos los mismos pasos, pero con el usuario de Ubuntu y su ip

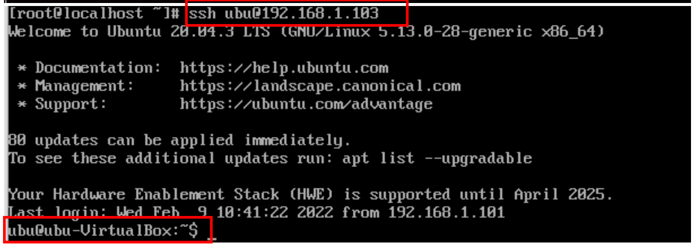


```
[root@localhost ~]# ssh-copy-id ubu@192.168.1.103
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alr
eady installed

/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote system
.
                (if you think this is a mistake, you may want to use -f option)
```

Conexión a los servidores con ssh



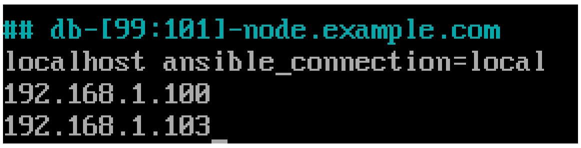
```
[root@localhost ansible]# ssh ubu@192.168.1.100
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.13.0-28-generic x86_64)

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

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

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Tue Feb  8 19:44:32 2022 from 192.168.1.101
ubu@ubu-VirtualBox:~$
```



```
[root@localhost ~]# ssh ubu@192.168.1.103
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.13.0-28-generic x86_64)

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

80 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Wed Feb  9 10:41:22 2022 from 192.168.1.101
ubu@ubu-VirtualBox:~$ _
```

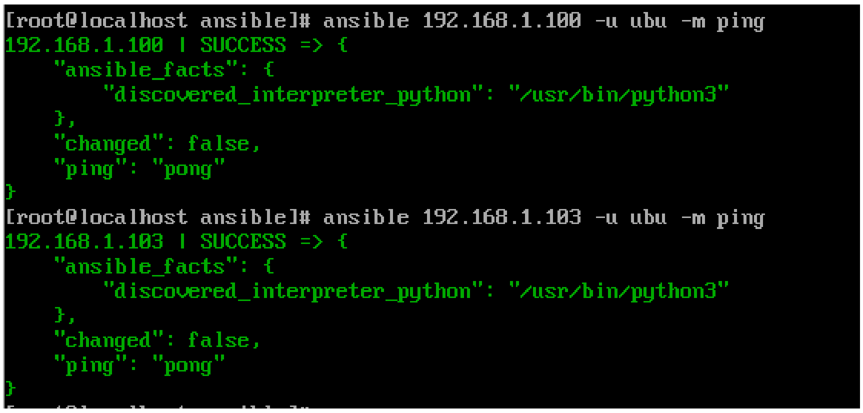
Conexión ping a los servidores

-Añadimos la ip de los servidores al archivo hosts



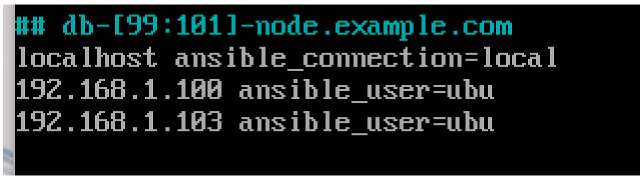
```
## db-[99:101]-node.example.com
localhost ansible_connection=local
192.168.1.100
192.168.1.103_
```

-probamos el ping pong



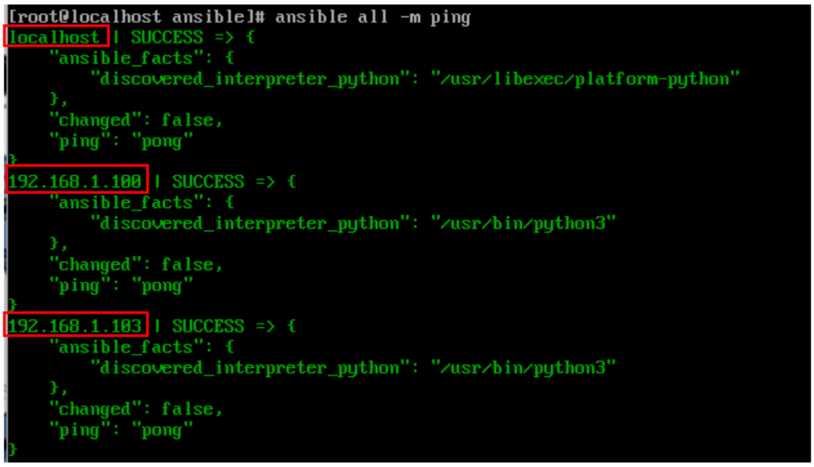
```
[root@localhost ansible]# ansible 192.168.1.100 -u ubu -m ping
192.168.1.100 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
[root@localhost ansible]# ansible 192.168.1.103 -u ubu -m ping
192.168.1.103 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

-añadimos el usuario en el archivo



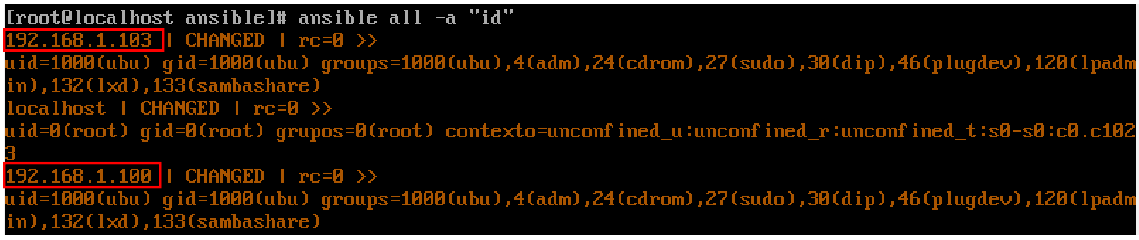
```
## db-[99:101]-node.example.com
localhost ansible_connection=local
192.168.1.100 ansible_user=ubu
192.168.1.103 ansible_user=ubu
```

-Hacemos ping a todas las maquinas



```
[root@localhost ansible]# ansible all -m ping
localhost | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
192.168.1.100 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
192.168.1.103 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

-vemos el id de las maquinas

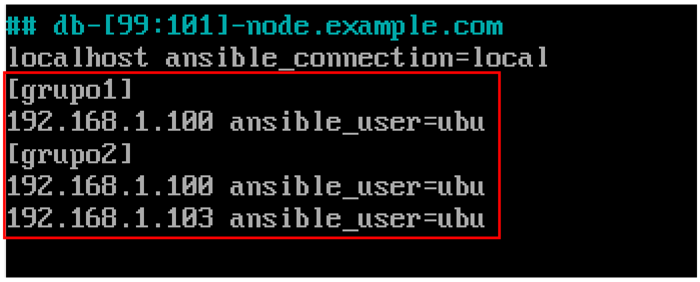


```
[root@localhost ansible]# ansible all -a "id"
192.168.1.103 | CHANGED | rc=0 >>
uid=1000(ubu) gid=1000(ubu) groups=1000(ubu),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadm
in),132(lxd),133(sambashare)
localhost | CHANGED | rc=0 >>
uid=0(root) gid=0(root) grupos=0(root) contexto=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c102
3
192.168.1.100 | CHANGED | rc=0 >>
uid=1000(ubu) gid=1000(ubu) groups=1000(ubu),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadm
in),132(lxd),133(sambashare)
```

Pruebas con los servidores

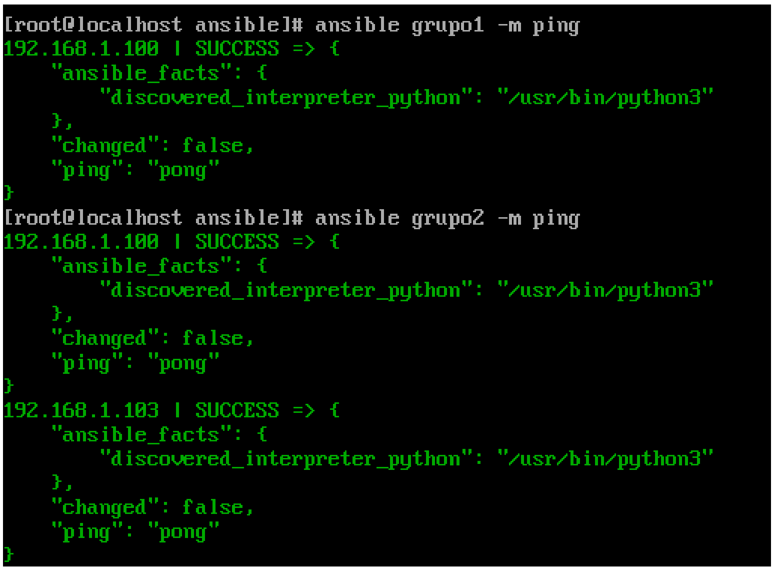
Creación grupos e hijo con servidores

-creamos dos grupos con los servidores en el archivo hosts



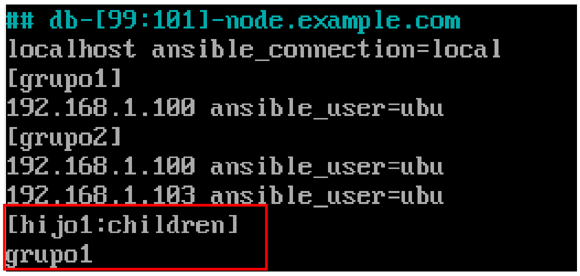
```
## db-[99:101]-node.example.com
localhost ansible_connection=local
[grupo1]
192.168.1.100 ansible_user=ubu
[grupo2]
192.168.1.100 ansible_user=ubu
192.168.1.103 ansible_user=ubu
```

-hacemos ping a los grupos



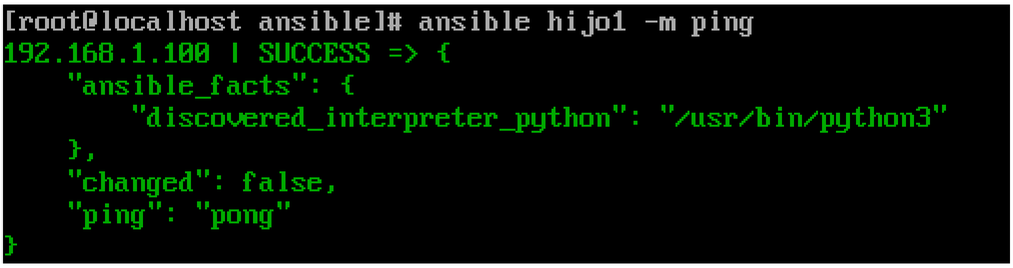
```
[root@localhost ansible]# ansible grupo1 -m ping
192.168.1.100 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
[root@localhost ansible]# ansible grupo2 -m ping
192.168.1.100 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
192.168.1.103 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

-creamos un hijo que contenga el grupo 1



```
## db-[99:101]-node.example.com
localhost ansible_connection=local
[grupo1]
192.168.1.100 ansible_user=ubu
[grupo2]
192.168.1.100 ansible_user=ubu
192.168.1.103 ansible_user=ubu
[hijo1:children]
grupo1
```

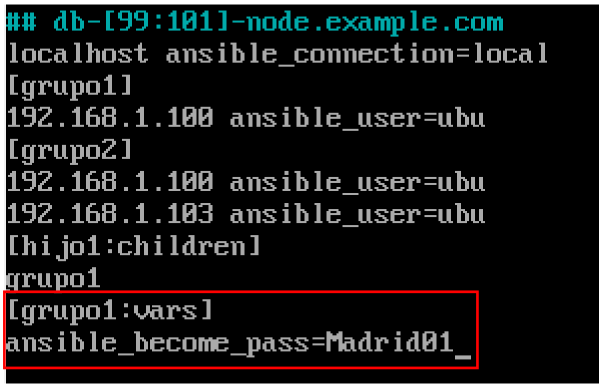
-probamos el ping al hijo



```
[root@localhost ansible]# ansible hijo1 -m ping
192.168.1.100 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

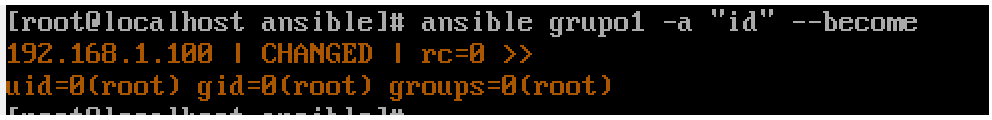
Conexión a un servidor mediante usuario root

-creamos una variable con la contraseña root

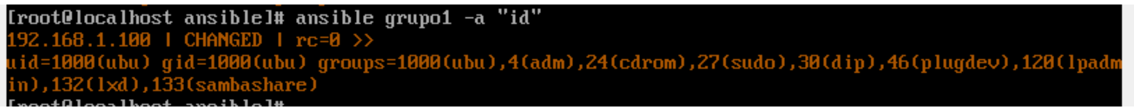


```
## db-[99:101]-node.example.com
localhost ansible_connection=local
[grupo1]
192.168.1.100 ansible_user=ubu
[grupo2]
192.168.1.100 ansible_user=ubu
192.168.1.103 ansible_user=ubu
[hijo1:children]
grupo1
[grupo1:vars]
ansible_become_pass=Madrid01_
```

Nos conectamos al hijo (--become sirve para modo root)



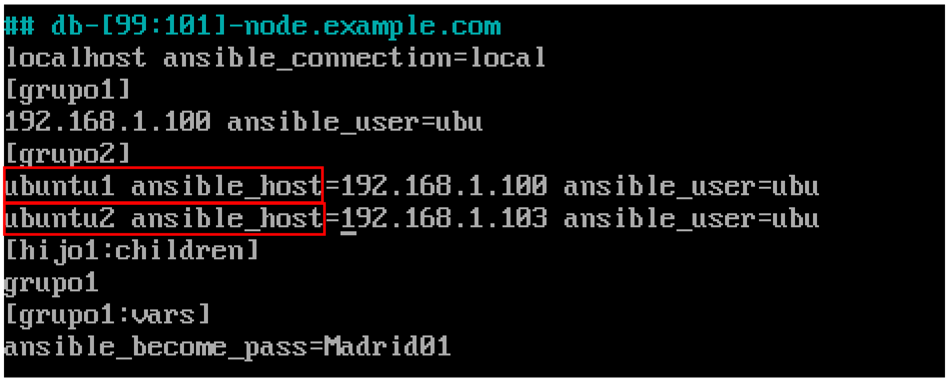
```
[root@localhost ansible]# ansible grupo1 -a "id" --become
192.168.1.100 | CHANGED | rc=0 >>
uid=0(root) gid=0(root) groups=0(root)
```



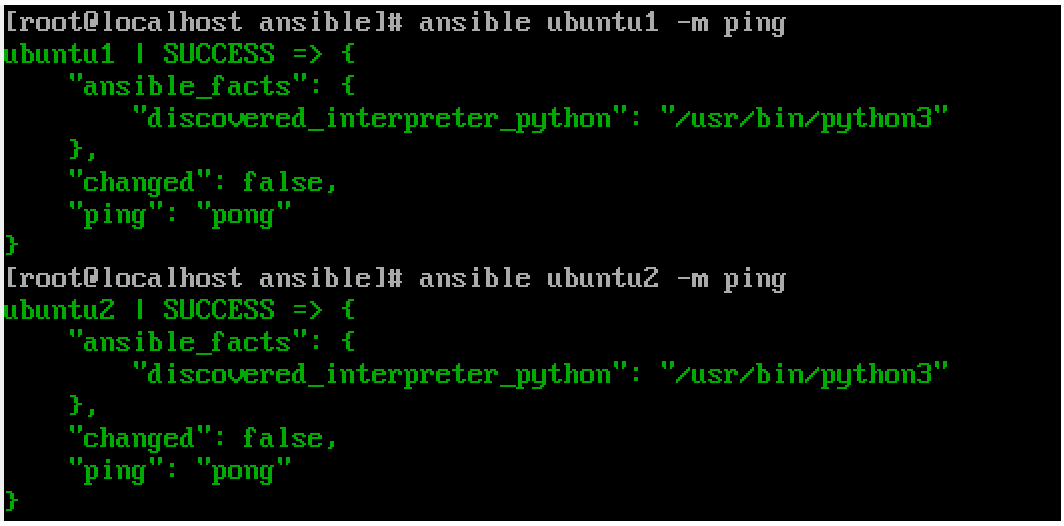
```
[root@localhost ansible]# ansible grupo1 -a "id"
192.168.1.100 | CHANGED | rc=0 >>
uid=1000(ubu) gid=1000(ubu) groups=1000(ubu),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadm
in),132(lxd),133(sambashare)
```

Nombres virtuales para los servidores

-ponemos un nombre virtual a los servidores modificando el archivo hosts



```
## db-[99:101]-node.example.com
localhost ansible_connection=local
[grupo1]
192.168.1.100 ansible_user=ubu
[grupo2]
ubuntu1 ansible_host=192.168.1.100 ansible_user=ubu
ubuntu2 ansible_host=192.168.1.103 ansible_user=ubu
[hijo1:children]
grupo1
[grupo1:vars]
ansible_become_pass=Madrid01
```



```
[root@localhost ansible]# ansible ubuntu1 -m ping
ubuntu1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
[root@localhost ansible]# ansible ubuntu2 -m ping
ubuntu2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```