

Abstract geometric lines in the top left corner, consisting of several overlapping, irregular polygons and lines in a light beige color.

PRÁCTICA 9 ANSIBLE

Paloma Pérez de Madrid

ÍNDICE

- 2) Preparación del entorno
- 3) Configuración de Ansible
- 4) Ejecución en modo ad-hoc
- 5) Libro de jugadas

ÍNDICE

2) Preparación del entorno

3) Configuración de Ansible

4) Ejecución en modo ad-hoc

5) Libro de jugadas

2) PREPARACIÓN DEL ENTORNO

Para la realización de la práctica construiremos un entorno de trabajo formado por una máquina de administración (con distribución Rocky 9.3) y cinco máquinas que serán gestionadas desde la anterior (dos con Rocky 9.3, una con Debian 11.2, otra con Ubuntu Server 22.4, y una última con OpenSUSE 15.3).

Nota: Estas instancias pueden arrancarse usando virtualización de escritorio (VMWare, VirtualBox, ...), en un proveedor de servicios cloud (AWS, ...), o en un entorno mixto (instancias locales y alojadas en el proveedor cloud). Si no es posible trabajar con las versiones del SO indicadas, se elegirán SO y versiones similares.

- a) Arrancar las seis máquinas virtuales e identificar sus direcciones IP.
- b) En los cinco equipos a gestionar comprobar que se cuenta con un servidor SSH arrancado. Crear o identificar un usuario con derechos administrativos (sudo) que usaremos para la administración (deberá tener Shell y directorio home, y podrá ejecutar comandos con cualquier otro perfil sin necesidad de introducir contraseña). Si es necesario, se pueden usar distintas cuentas de usuario para esta función, dependiendo del SO instalado.
- c) En el equipo de administración necesitamos una pareja de claves pública/privada sin frase de paso. Si el equipo administrado ya tiene inyectada la clave pública (instancias cloud), sólo será necesario contar con la clave privada correspondiente. En otro caso, crearemos una pareja de claves y copiaremos la clave pública al equipo administrado (utilizando la cuenta de usuario elegida para la administración remota).
- d) Comprobar que el usuario elegido puede acceder por SSH a las máquinas gestionadas utilizando la clave privada sin necesidad de introducir contraseña o frase de paso, y que el usuario puede ejecutar comandos con privilegios de root. Nota: Incluir en la memoria los problemas encontrados para configurar las máquinas, y el modo en que se han resuelto.

2) PREPARACIÓN DEL ENTORNO

Para la realización de la práctica construiremos un entorno de trabajo formado por **una máquina** de administración (con distribución Rocky 9.3) y **cinco máquinas** que serán gestionadas desde la anterior (**dos** con Rocky 9.3, **una** con Debian 11.2, **otra** con Ubuntu Server 22.4, y **una** última con OpenSUSE 15.3).

Nota: Estas instancias pueden arrancarse usando virtualización de escritorio (VMWare, VirtualBox, ...), en un proveedor de servicios cloud (AWS, ...), o en un entorno mixto (instancias locales y alojadas en el proveedor

a) Arrancar las seis máquinas virtuales e identificar sus direcciones IP.

1 Rocky 9.3

2 con Rocky 9.3

1 con Debian 11.2

1 con Ubuntu Server 22.4

1 OpenSUSE 15.3

Rocky Admin
192.168.119.144

Nodo Ubuntu
192.168.119.137

Nodo Debian
192.168.119.149

Nodo OpenSuse
192.168.119.146

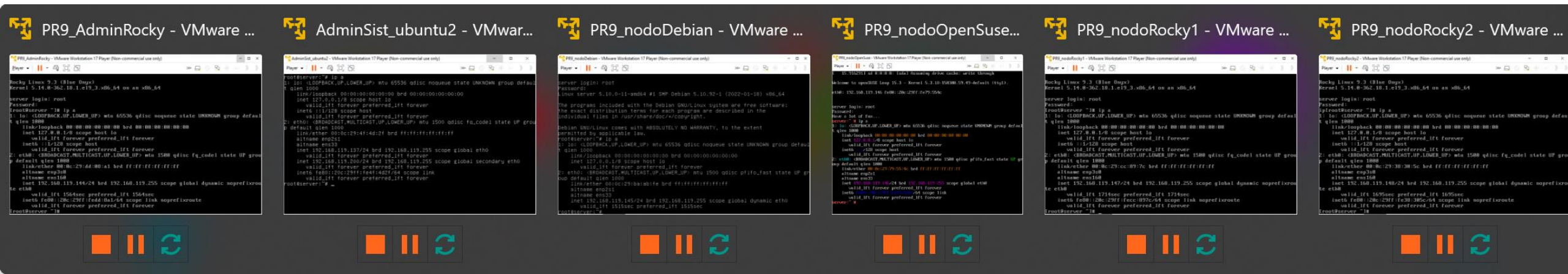
Nodo Rocky 1
192.168.119.147

Nodo Rocky 2
192.168.119.148

2) PREPARACIÓN DEL ENTORNO

Para la realización de la práctica construiremos un entorno de trabajo formado por **una máquina** de administración (con distribución Rocky 9.3) y **cinco máquinas** que serán gestionadas desde la anterior (**dos** con Rocky 9.3, **una** con Debian 11.2, **otra** con Ubuntu Server 22.4, y **una** última con OpenSUSE 15.3).

Nota: Estas instancias pueden arrancarse usando virtualización de escritorio (VMWare, VirtualBox, ...), en un proveedor de servicios cloud (AWS, ...), o en un entorno mixto (instancias locales y alojadas en el proveedor)



Nota: si falla el ssh puede ser que sea porque está activo pero no “enabled”

2) PREPARACIÓN DEL ENTORNO

b) En los cinco equipos a gestionar comprobar que se cuenta con un servidor SSH arrancado. Crear o identificar un usuario con derechos administrativos (sudo) que usaremos para la administración (deberá tener Shell y directorio home, y podrá ejecutar comandos con cualquier otro perfil sin necesidad de introducir contraseña). Si es necesario, se pueden usar distintas cuentas de usuario para esta función, dependiendo del SO instalado.

La máquina administradora cuenta con el usuario “root” y con los derechos administrativos

Rocky Admin
192.168.119.144

```

[root@server ~]# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-05-11 18:53:29 CEST; 1h 42min left
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 874 (sshd)
      Tasks: 1 (limit: 4385)
     Memory: 4.7M
        CPU: 151ms
    CGroup: /system.slice/sshd.service
            └─874 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

may 11 18:53:28 server systemd[1]: Starting OpenSSH server daemon...
may 11 18:53:29 server sshd[874]: Server listening on 0.0.0.0 port 22.
may 11 18:53:29 server sshd[874]: Server listening on :: port 22.
may 11 18:53:29 server systemd[1]: Started OpenSSH server daemon.
may 11 17:05:42 server sshd[1428]: Accepted password for root from 192.168.119.1 port 58165 ssh2
may 11 17:05:42 server sshd[1428]: pam_unix(sshd:session): session opened for user root(uid=0) by (uid=0)
[root@server ~]#
```

2) PREPARACIÓN DEL ENTORNO

b) En los cinco equipos a gestionar comprobar que se cuenta con un servidor SSH arrancado. Crear o identificar un usuario con derechos administrativos (sudo) que usaremos para la administración (deberá tener Shell y directorio home, y podrá ejecutar comandos con cualquier otro perfil sin necesidad de introducir contraseña). Si es necesario, se pueden usar distintas cuentas de usuario para esta función, dependiendo del SO instalado.

En el resto de los equipos se creará un usuario “ansible” con contraseña “ansible”

```
useradd -m ansible
passwd ansible
usermod -aG sudo ansible
visudo → ansible ALL=(ALL) NOPASSWD: ALL
su - ansible
```

En Rocky y Opensuse:

```
useradd -m ansible
passwd ansible
visudo → ansible ALL=(ALL) NOPASSWD: ALL
su - ansible
```

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

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

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
```

Nota: en debian tuve problemas editando el visudo (alternativa: vim /etc/sudoers)

Nota: en Debian 11.2 tuve que descargarme sudo (apt-get install sudo)

Nota: -m crea el directorio home, sin él no te puedes conectar por ssh

Nota: Para el futuro, añadir al usuario ansible al grupo ansible

Nota: Borrar un usuario de un grupo gpasswd -d nuevo_usuario

2) PREPARACIÓN DEL ENTORNO

Error que he tenido en Rocky:

```
##
## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL
ansible ALL=(ALL)        NOPASSWD: ALL
```

Correcto

```
## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL
ansible ALL=(ALL) NOPASSWD: ALL
```

Incorrecto, si haces sudo te pide contraseña

Error que he tenido en Debian y en Ubuntu

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

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL)    ALL
ansible ALL=(ALL:ALL)    NOPASSWD: ALL ←
# See sudoers(5) for more information on "@include" directives:
```

Si no lo añades y haces sudo te pide contraseña

En OpenSuse funciona sin problemas

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

2) PREPARACIÓN DEL ENTORNO

b) En los cinco equipos a gestionar comprobar que se cuenta con un servidor SSH arrancado. Crear o identificar un usuario con derechos administrativos (sudo) que usaremos para la administración (deberá tener Shell y directorio home, y podrá ejecutar comandos con cualquier otro perfil sin necesidad de introducir contraseña). Si es necesario, se pueden usar distintas cuentas de usuario para esta función, dependiendo del SO instalado.

Todas las máquinas cuentan con el usuario “ansible” y con los derechos administrativos

Nodo Ubuntu
192.168.119.137

```
root@server:~# systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-03-21 16:18:37 UTC; 1 month 20 days ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 990 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 1012 (sshd)
    Tasks: 1 (limit: 970)
   Memory: 8.7M
      CPU: 1.316s
   CGroup: /system.slice/ssh.service
           └─1012 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

2) PREPARACIÓN DEL ENTORNO

b) En los cinco equipos a gestionar comprobar que se cuenta con un servidor SSH arrancado. Crear o identificar un usuario con derechos administrativos (sudo) que usaremos para la administración (deberá tener Shell y directorio home, y podrá ejecutar comandos con cualquier otro perfil sin necesidad de introducir contraseña). Si es necesario, se pueden usar distintas cuentas de usuario para esta función, dependiendo del SO instalado.

Todas las máquinas cuentan con el usuario “ansible” y con los derechos administrativos

Nodo Debian
192.168.119.145

apt install ssh

```
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/s
systemd/system/ssh.service.
rescue-ssh.target is a disabled or a static unit, not starting it.
Configurando ssh (1:8.4p1-5+deb11u3) ...
Procesando disparadores para man-db (2.9.4-2) ...
Procesando disparadores para libc-bin (2.31-13+deb11u2) ...
root@server:~# systemctl status sshd
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: ena
   Active: active (running) since Sat 2024-05-11 17:10:19 CEST; 2min 44s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 1209 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 1211 (sshd)
     Tasks: 1 (limit: 486)
    Memory: 1.2M
       CPU: 33ms
   CGroup: /system.slice/ssh.service
           └─1211 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
```

2) PREPARACIÓN DEL ENTORNO

/etc/ssh/sshd_config

Error que he tenido: no estaba
confiurada correctamente la
conexión ssh

Nodo Debian
192.168.119.145

sudo apt get update
sudo apt get upgrade

vim /etc/passwd → comprobar
que está tu usuario

ansible:x:1001:1001::/home/ansible:/bin/sh

```
# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

-

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

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
-- INSERTAR --
```

39,1

26%

2) PREPARACIÓN DEL ENTORNO

b) En los cinco equipos a gestionar comprobar que se cuenta con un servidor SSH arrancado. Crear o identificar un usuario con derechos administrativos (sudo) que usaremos para la administración (deberá tener Shell y directorio home, y podrá ejecutar comandos con cualquier otro perfil sin necesidad de introducir contraseña). Si es necesario, se pueden usar distintas cuentas de usuario para esta función, dependiendo del SO instalado.

Todas las máquinas cuentan con el usuario “ansible” y con los derechos administrativos

Nodo OpenSuse
192.168.119.146

```
Last login: Sat May 11 16:58:05 2024
Have a lot of fun...
server:~ # systemctl status sshd
● sshd.service - OpenSSH Daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2024-05-11 16:57:06 CEST; 16min ago
     Process: 1434 ExecStartPre=/usr/sbin/sshd-gen-keys-start (code=exited, status=0/SUCCESS)
     Process: 1508 ExecStartPre=/usr/sbin/sshd -t $SSHD_OPTS (code=exited, status=0/SUCCESS)
    Main PID: 1524 (sshd)
       Tasks: 1
      CGroup: /system.slice/sshd.service
              └─1524 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

May 11 16:57:04 server systemd[1]: Starting OpenSSH Daemon...
May 11 16:57:04 server sshd-gen-keys-start[1434]: Checking for missing server keys in /etc/ssh
```

2) PREPARACIÓN DEL ENTORNO

b) En los cinco equipos a gestionar comprobar que se cuenta con un servidor SSH arrancado. Crear o identificar un usuario con derechos administrativos (sudo) que usaremos para la administración (deberá tener Shell y directorio home, y podrá ejecutar comandos con cualquier otro perfil sin necesidad de introducir contraseña). Si es necesario, se pueden usar distintas cuentas de usuario para esta función, dependiendo del SO instalado.

Nodo Rocky 1
192.168.119.147

```
warning: Permanently added 192.168.119.147 (192.168.119.147) to the list of known hosts.
root@192.168.119.147's password:
Last login: Sat May 11 16:59:56 2024
[root@server ~]# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-05-11 18:58:37 CEST; 1h 43min left
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 1077 (sshd)
     Tasks: 1 (limit: 4385)
```

Nodo Rocky 2
192.168.119.148

```
warning: Permanently added 192.168.119.148 (192.168.119.148) to the list of known hosts.
root@192.168.119.148's password:
Last login: Sat May 11 17:00:16 2024
[root@server ~]# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-05-11 18:58:35 CEST; 1h 43min left
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 957 (sshd)
     Tasks: 1 (limit: 4385)
```

2) PREPARACIÓN DEL ENTORNO

c) En el equipo de administración necesitamos una pareja de claves pública/privada sin frase de paso. Si el equipo administrado ya tiene inyectada la clave pública (instancias cloud), sólo será necesario contar con la clave privada correspondiente. En otro caso, crearemos una pareja de claves y copiaremos la clave pública al equipo administrado (utilizando la cuenta de usuario elegida para la administración remota).

`ssh-keygen` (no añadimos frase de paso)

Activar la autenticación basada en claves en el servidor
(`/etc/ssh/sshd_config`):

```
PubkeyAuthentication yes
```

`ssh-copy-id username@ip_nodo` → Copiar la clave pública en los nodos

Si falla → especificar el puerto: `ssh-copy-id -p 22 username@ip_nodo`

Rocky Admin 192.168.119.144

```
[root@server ~]# ssh-keygen
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:
SHA256:9PiywA04bM1LOLyVn8pqqdVKKADqcbbNAGrZM0ieSiSQ root@server
The key's randomart image is:
----[RSA 3072]-----+
E=
0++ .. .
0+++ = =
++ *+ * +..
.o X.o.S+
o o.++o.
o.=o.
.o o
..
-----[SHA256]-----+
```

2) PREPARACIÓN DEL ENTORNO

c) En el equipo de administración necesitamos una pareja de claves pública/privada sin frase de paso. Si el equipo administrado ya tiene inyectada la clave pública (instancias cloud), sólo será necesario contar con la clave privada correspondiente. En otro caso, crearemos una pareja de claves y copiaremos la clave pública al equipo administrado (utilizando la cuenta de usuario elegida para la administración remota).

`ssh-copy-id ansible@192.168.119.137`

```
[root@server group_vars]# ssh-copy-id ansible@192.168.119.137
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to f
dy installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
l the new keys
ansible@192.168.119.137's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'ansible@192.168.119.137'"
and check to make sure that only the key(s) you wanted were added.
```

Nodo Ubuntu 192.168.119.137

`ssh-copy-id ansible@192.168.119.146`

```
[root@server ~]# ssh-copy-id ansible@192.168.119.146
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to f
dy installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
l the new keys
(ansible@192.168.119.146) Password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'ansible@192.168.119.146'"
and check to make sure that only the key(s) you wanted were added.

[root@server ~]# |
```

Nodo OpenSuse 192.168.119.146

2) PREPARACIÓN DEL ENTORNO

c) En el equipo de administración necesitamos una pareja de claves pública/privada sin frase de paso. Si el equipo administrado ya tiene inyectada la clave pública (instancias cloud), sólo será necesario contar con la clave privada correspondiente. En otro caso, crearemos una pareja de claves y copiaremos la clave pública al equipo administrado (utilizando la cuenta de usuario elegida para la administración remota).

`ssh-copy-id ansible@192.168.119.149`

```
[root@server ~]# ssh-copy-id ansible@192.168.119.149
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to f
dy installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
l the new keys
ansible@192.168.119.149's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'ansible@192.168.119.149'"
and check to make sure that only the key(s) you wanted were added.
```

`ssh-copy-id ansible@192.168.119.147`

```
[root@server ~]# ssh-copy-id ansible@192.168.119.147
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to f
dy installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
l the new keys
ansible@192.168.119.147's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'ansible@192.168.119.147'"
and check to make sure that only the key(s) you wanted were added.
```

Nodo Debian 192.168.119.149

Nodo Rocky 1 192.168.119.147

2) PREPARACIÓN DEL ENTORNO

c) En el equipo de administración necesitamos una pareja de claves pública/privada sin frase de paso. Si el equipo administrado ya tiene inyectada la clave pública (instancias cloud), sólo será necesario contar con la clave privada correspondiente. En otro caso, crearemos una pareja de claves y copiaremos la clave pública al equipo administrado (utilizando la cuenta de usuario elegida para la administración remota).

`ssh-copy-id ansible@192.168.119.148`

```
[root@server ~]# ssh-copy-id ansible@192.168.119.148
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to f
dy installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
l the new keys
ansible@192.168.119.148's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'ansible@192.168.119.148'"
and check to make sure that only the key(s) you wanted were added.
```

```
[ansible@server ~]$ ls ~/.ssh
authorized_keys
```

Nodo Rocky 2 192.168.119.148

2) PREPARACIÓN DEL ENTORNO

Desde La máquina Rocky Admin:

```
[root@server ~]# ssh ansible@192.168.119.137
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-100-generic x86_64)

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

System information as of mié 15 may 2024 18:37:36 UTC

System load:  0.14501953125   Processes:            212
Usage of /:   43.7% of 18.53GB Users logged in:             1
Memory usage: 41%            IPv4 address for eth0: 192.168.119.137
Swap usage:   0%             IPv4 address for eth0: 192.168.119.137

El mantenimiento de seguridad expandido para Applications está desactivado.
Se pueden aplicar 17 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

1 actualización de seguridad adicional se puede aplicar con ESM Apps at https://ubuntu.com/esm
Aprenda más sobre cómo activar el servicio ESM Apps at https://ubuntu.com/esm

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-releases:
proxy settings

Last login: Wed May 15 17:39:49 2024 from 192.168.119.144
$ sudo ls /root
snap
```

Se conecta a Ubuntu (.137) y lista el directorio /root

d) Comprobar que el usuario elegido puede acceder por SSH a las máquinas gestionadas utilizando la clave privada **sin necesidad de introducir contraseña** o frase de paso, y que el usuario puede ejecutar comandos con privilegios de root. Nota: Incluir en la memoria los problemas encontrados para configurar las máquinas, y el modo en que se han resuelto

```
[root@server ~]# ssh ansible@192.168.119.146
Last login: Wed May 15 19:41:03 2024 from 192.168.119.144
Have a lot of fun...
ansible@server:~> |
```

```
ansible@server:~> sudo ls /root
bin  inst-sys
```

Se conecta a OpenSuse (.146) y lista el directorio /root

2) PREPARACIÓN DEL ENTORNO

d) Comprobar que el usuario elegido puede acceder por SSH a las máquinas gestionadas utilizando la clave privada sin necesidad de introducir contraseña o frase de paso, y que el usuario puede ejecutar comandos con privilegios de root. Nota: Incluir en la memoria los problemas encontrados para configurar las máquinas, y el modo en que se han resuelto

Desde La máquina Rocky Admin:

```
[root@server ~]# ssh ansible@192.168.119.149
Linux server 5.10.0-11-amd64 #1 SMP Debian 5.10.92-1 (2022-01-18) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May 15 20:34:05 2024 from 192.168.119.144
$ sudo apt-get update
Des:1 http://security.debian.org/debian-security bullseye-security InRelease [48,4 kB]
Obj:2 http://deb.debian.org/debian bullseye InRelease
Obj:3 http://deb.debian.org/debian bullseye-updates InRelease
Descargados 48,4 kB en 1s (58,3 kB/s)
Leyendo lista de paquetes... Hecho
```

Se conecta al nodo Debian (.149) y puede hacer un update sin meter la contraseña

```
[root@server ~]# ssh ansible@192.168.119.148
Last login: Wed May 15 20:11:39 2024 from 192.168.119.144
[ansible@server ~]$
[ansible@server ~]$ sudo ls /root
anaconda-ks.cfg
```

Se conecta al nodo Rocky 2(.148) y lista el directorio /root

```
[root@server ~]# ssh ansible@192.168.119.147
Last login: Wed May 15 20:17:40 2024 from 192.168.119.144
[ansible@server ~]$
[ansible@server ~]$ sudo ls /root
anaconda-ks.cfg
```

Se conecta al nodo Rocky 1 (.147) y lista el directorio /root

ÍNDICE

2) Preparación del entorno

3) Configuración de Ansible

4) Ejecución en modo ad-hoc

5) Libro de jugadas

3) CONFIGURACIÓN DE ANSIBLE

- a) Instalar Ansible en la máquina de administración.
- b) Editar el Inventario de Ansible y crear los siguientes grupos: WebServers (un equipo Rocky y los equipos Debian, Ubuntu y OpenSUSE), DNS_Maestro (un equipo Rocky), DNS_Esclavo (el otro equipo Rocky) y Servidores_DNS (los dos equipos Rocky). Es posible crear otros grupos adicionales, si lo considera necesario para agrupar instancias con características comunes.
- c) Establecer las variables globales `ansible_user` y `ansible_ssh_private_key_file` mediante uno o más archivos de variables (bajo el directorio `group_vars`).

Nota: Si todos los equipos administrados usan los mismos valores podemos usar el archivo `all`. En otro caso usaremos archivos específicos de cada grupo (el archivo tendrá el nombre del grupo).

- d) Comprobar la alcanzabilidad (capacidad de acceso y administración del equipo) usando el módulo `ping` sobre todos los equipos administrados (`all`). Probar también el módulo con la opción `-b` (acceso como root).

3) CONFIGURACIÓN DE ANSIBLE

a) Instalar Ansible en la máquina de administración.

`yum install ansible`

```
[root@server ~]# yum install ansible
Error al cargar el complemento "config_manager": '*prog'
Rocky Linux 9 - BaseOS                    5.9 kB/s | 4.1 kB      00:00
Rocky Linux 9 - AppStream                  13 kB/s | 4.5 kB      00:00
Rocky Linux 9 - Extras                     11 kB/s | 2.9 kB      00:00
Dependencias resueltas.
=====
Paquete                                Arquitectura  Versión                Repositorio            Tam.
=====
Instalando:
  ansible                               noarch        1:7.7.0-1.el9          epel                    34 M
Instalando dependencias:
```

3) CONFIGURACIÓN DE ANSIBLE

b) Editar el Inventario de Ansible y crear los siguientes grupos: WebServers (un equipo Rocky y los equipos Debian, Ubuntu y OpenSUSE), DNS_Maestro (un equipo Rocky), DNS_Esclavo (el otro equipo Rocky) y Servidores_DNS (los dos equipos Rocky). Es posible crear otros grupos adicionales, si lo considera necesario para agrupar instancias con características comunes.

archivo de inventario -- > vim /etc/ansible/hosts

```
[root@server ~]# cd /etc/ansible/  
[root@server ansible]# ls  
ansible.cfg  hosts  roles  
[root@server ansible]# vim hosts
```

```
[WebServers]  
nodo-rocky-1 ansible_host=192.168.119.147  
nodo-debian ansible_host=192.168.119.145  
nodo-ubuntu ansible_host=192.168.119.137  
nodo-opensuse ansible_host=192.168.119.146  
  
[DNS_Maestro]  
nodo-rocky-1 ansible_host=192.168.119.147  
  
[DNS_Esclavo]  
nodo-rocky-2 ansible_host=192.168.119.148  
  
[Servidores_DNS]  
nodo-rocky-1 ansible_host=192.168.119.147  
nodo-rocky-2 ansible_host=192.168.119.148
```

```
[WebServers]  
nodo-rocky-1 ansible_host=192.168.119.147  
nodo-debian ansible_host=192.168.119.149  
nodo-ubuntu ansible_host=192.168.119.137  
nodo-opensuse ansible_host=192.168.119.146
```

```
[DNS_Maestro]  
nodo-rocky-1 ansible_host=192.168.119.147
```

```
[DNS_Esclavo]  
nodo-rocky-2 ansible_host=192.168.119.148
```

```
[Servidores_DNS]  
nodo-rocky-1 ansible_host=192.168.119.147  
nodo-rocky-2 ansible_host=192.168.119.148
```


3) CONFIGURACIÓN DE ANSIBLE

b) Editar el Inventario de Ansible y crear los siguientes grupos: WebServers (un equipo Rocky y los equipos Debian, Ubuntu y OpenSUSE), DNS_Maestro (un equipo Rocky), DNS_Esclavo (el otro equipo Rocky) y Servidores_DNS (los dos equipos Rocky). Es posible crear otros grupos adicionales, si lo considera necesario para agrupar instancias con características comunes.

Verificar el inventario:

```
ansible-inventory --list -y
```

¿Por qué salen algunos nodos como {}?

No tienen ninguna configuración adicional. Si se observa, los nodos que aparecen como {} son los que ya se han especificado su ip en otros grupos

```
[root@server ansible]# ansible-inventory --list -y
all:
  children:
    DNS_Esclavo:
      hosts:
        nodo-rocky-2:
          ansible_host: 192.168.119.148
    DNS_Maestro:
      hosts:
        nodo-rocky-1: {}
    Servidores_DNS:
      hosts:
        nodo-rocky-1: {}
        nodo-rocky-2: {}
    WebServers:
      hosts:
        nodo-debian:
          ansible_host: 192.168.119.145
        nodo-opensuse:
          ansible_host: 192.168.119.146
        nodo-rocky-1:
          ansible_host: 192.168.119.147
        nodo-ubuntu:
          ansible_host: 192.168.119.137
    ungrouped: {}
```

3) CONFIGURACIÓN DE ANSIBLE

c) Establecer las variables globales `ansible_user` y `ansible_ssh_private_key_file` mediante uno o más archivos de variables (bajo el directorio `group_vars`).

- 1) Crear directorio `/etc/ansible/group_vars`
- 2) Crear el archivo de variables para cada grupo
- 3) Definir las variables en los archivos

`mkdir /etc/ansible/group_vars`

```
[root@server ansible]# mkdir /etc/ansible/group_vars
[root@server ansible]# ls
ansible.cfg  group_vars  hosts  roles
```

- 2) Crear el archivo de variables para cada grupo

```
vim /etc/ansible/group_vars/WebServers
vim /etc/ansible/group_vars/DNS_Maestro
vim /etc/ansible/group_vars/DNS_Esclavo
vim /etc/ansible/group_vars/Servidores_DNS
```

```
[root@server group_vars]# ls
DNS_Esclavo  DNS_Maestro  Servidores_DNS  WebServers
```

3) CONFIGURACIÓN DE ANSIBLE

c) Establecer las variables globales `ansible_user` y `ansible_ssh_private_key_file` mediante uno o más archivos de variables (bajo el directorio `group_vars`).

3) Definir las variables en los archivos

```
vim /etc/ansible/group_vars/WebServers
vim /etc/ansible/group_vars/DNS_Maestro
vim /etc/ansible/group_vars/DNS_Esclavo
vim /etc/ansible/group_vars/Servidores_DNS
```

Comprobar variable clave privada

```
[root@server group_vars]# ansible WebServers -m debug -a "var=ansible_ssh_private_key_file"
nodo-rocky-1 | SUCCESS => {
  "ansible_ssh_private_key_file": "~/.ssh/id_rsa"
}
nodo-debian | SUCCESS => {
  "ansible_ssh_private_key_file": "~/.ssh/id_rsa"
}
nodo-ubuntu | SUCCESS => {
  "ansible_ssh_private_key_file": "~/.ssh/id_rsa"
}
nodo-opensuse | SUCCESS => {
  "ansible_ssh_private_key_file": "~/.ssh/id_rsa"
}
[root@server group_vars]#
```

Usuario que administra los nodos

```
[root@server group_vars]# ls
DNS_Esclavo  DNS_Maestro  Servidores_DNS  WebServers
[root@server group_vars]# cat DNS_Esclavo
ansible_user: ansible
ansible_ssh_private_key_file: ~/.ssh/id_rsa
[root@server group_vars]# cat DNS_Maestro
ansible_user: ansible
ansible_ssh_private_key_file: ~/.ssh/id_rsa
[root@server group_vars]# cat Servidores_DNS
ansible_user: ansible
ansible_ssh_private_key_file: ~/.ssh/id_rsa
[root@server group_vars]# cat WebServers
ansible_user: ansible
ansible_ssh_private_key_file: ~/.ssh/id_rsa
```

3) CONFIGURACIÓN DE ANSIBLE

c) Establecer las variables globales `ansible_user` y `ansible_ssh_private_key_file` mediante uno o más archivos de variables (bajo el directorio `group_vars`).

`ansible all -m ping`

```
nodo-ubuntu | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
nodo-rocky-1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
nodo-rocky-2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
}
```

```
[root@server group_vars]# ansible all -m ping
nodo-debian | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform linux on host nodo-opensuse is using the discovered Python interpreter at
/usr/bin/python3.6, but future installation of another Python interpreter could change the meaning
of that path. See https://docs.ansible.com/ansible-
core/2.14/reference_appendices/interpreter_discovery.html for more information.
nodo-opensuse | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.6"
  },
  "changed": false,
  "ping": "pong"
}
}
```

ÍNDICE

2) Preparación del entorno

3) Configuración de Ansible

4) Ejecución en modo ad-hoc

5) Libro de jugadas

4) EJECUCIÓN EN MODO AD-HOC

Probar el funcionamiento de Ansible en **modo comando (ad-hoc)** realizando sobre los equipos las siguientes tareas de configuración:

- a) Instalar Apache y Bind en uno de los servidores Rocky. Configurar el arranque automático al inicio de ambos servicios.
- b) Crear un directorio /var/archivos en todos los servidores y copiar un archivo ejemplo.txt (con un contenido arbitrario) a esa ubicación.
- c) Actualizar el sistema operativo del equipo Debian (Nota: puede ser un proceso lento)
- d) Comprobar el funcionamiento de los módulos lineinfile y blockinfile, realizando distintas tareas sobre archivos ubicados en algún nodo administrado: Añadir líneas y bloques de texto a un archivo, borrar líneas o bloques de texto existentes, cambiar el valor de una directiva (por ejemplo, en un archivo .conf, cambiar un valor “yes” por “no” en una directiva, o activar una nueva directiva con el valor deseado). Comprobar el efecto de la directiva backup. En el caso del módulo blockinfile, estudiar también la propiedad Marker e indicar cuál es su función.
- e) Comprobar el funcionamiento del módulo **template**. Preparar una plantilla en formato jinja2 (.j2) que utilice alguna variable de prueba previamente definida en el archivo group_vars/all. Lanzar el módulo template sobre una máquina administrada y verificar el resultado.

4) EJECUCIÓN EN MODO AD-HOC

En rocky 1: `yum install bind | yum install httpd`

- a) Instalar Apache y Bind en uno de los servidores Rocky. Configurar el arranque automático al inicio de ambos servicios.

Nodo Rocky 1
192.168.119.147

```
[DNS_Maestro]
nodo-rocky-1 ansible_host=192.168.119.147
```

```
ansible -i archivo_de_inventario nombre_del_grupo -u USUARIO -b -m apt -a "name=SERVICIO state=present"
-m service -a "name=SERVICIO state=started enabled=yes"
```

Apache

```
ansible -i /etc/ansible/hosts DNS_Maestro -u ansible -b -m apt -a "name=httpd state=present" -m service -a
"name=httpd state=started enabled=yes"
```

Bind

```
ansible -i /etc/ansible/hosts DNS_Maestro -u ansible -b -m apt -a "name=named state=present" -m service -a
"name=named state=started enabled=yes"
```

4) EJECUCIÓN EN MODO AD-HOC

- a) Instalar Apache y Bind en uno de los servidores Rocky. Configurar el arranque automático al inicio de ambos servicios.

Apache

```
[root@server group_vars]# ansible -i /etc/ansible/hosts DNS_Maestro -u ansible -b -m apt -a "name=ht  
pd state=present" -m service -a "name=httpd state=started enabled=yes"  
nodo-rocky-1 | CHANGED => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "changed": true,  
  "enabled": true,  
  "name": "httpd",  
  "state": "started",  
  "status": {  
    "ActiveEnterTimestampMonotonic": "0",
```


4) EJECUCIÓN EN MODO AD-HOC

- a) Instalar Apache y Bind en uno de los servidores Rocky. Configurar el arranque automático al inicio de ambos servicios.

Bind

```
[root@server group_vars]# ansible -i /etc/ansible/hosts DNS_Maestro -u ansible -b -m apt -a "name=named state=present" -m service -a "name=named state=started enabled=yes"
nodo-rocky-1 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "enabled": true,
  "name": "named",
  "state": "started",
  "status": {
    "ActiveEnterTimestampMonotonic": "0",
```

4) EJECUCIÓN EN MODO AD-HOC

- a) Instalar Apache y Bind en uno de los servidores Rocky. Configurar el arranque automático al inicio de ambos servicios.

Nodo Rocky 1
192.168.119.147

[DNS_Maestro]
nodo-rocky-1 ansible_host=192.168.119.147

No se pueden activar a la vez → `ansible -i /etc/ansible/hosts DNS_Maestro -u ansible -b -m apt -a "name=apache2,bind9 state=present" -m service -a "name=httpd,bind9 state=started enabled=yes"`

```
[root@server group_vars]# ansible -i /etc/ansible/hosts DNS_Maestro -u ansible -b -m apt -a "name=apache2,bind9 state=present" -m service -a "name=httpd,bind9 state=started enabled=yes"
nodo-rocky-1 | FAILED! => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "msg": "Could not find the requested service httpd,bind9: host"
}
```

4) EJECUCIÓN EN MODO AD-HOC

b) Crear un directorio /var/archivos en todos los servidores y copiar un archivo ejemplo.txt (con un contenido arbitrario) a esa ubicación.

```
ansible all -m file -a "path=/var/archivos state=directory" -b
```

```
vim ejemplo.txt
```

```
ansible all -m copy -a "src=ejemplo.txt dest=/var/archivos/ejemplo.txt"
```

```
ansible all -m file -a "path=/var/archivos state=directory" -b
```

-b → ejecutar los comandos con permisos de usuario

```
[root@server group_vars]# ansible all -m file -a "path=/var/archivos state=directory" -b
nodo-debian | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "gid": 0,
  "group": "root",
  "mode": "0755",
  "owner": "root",
  "path": "/var/archivos",
  "size": 4096,
  "state": "directory",
  "uid": 0
}
```

```
nodo-opensuse | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.6"
  },
  "changed": false,
  "gid": 0,
  "group": "root",
  "mode": "0755",
  "owner": "root",
  "path": "/var/archivos",
  "size": 22,
  "state": "directory",
  "uid": 0
}
nodo-ubuntu | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "gid": 0,
  "group": "root",
  "mode": "0755",
  "owner": "root",
  "path": "/var/archivos",
  "size": 4096,
  "state": "directory",
  "uid": 0
}
```

4) EJECUCIÓN EN MODO AD-HOC

b) Crear un directorio /var/archivos en todos los servidores y copiar un archivo ejemplo.txt (con un contenido arbitrario) a esa ubicación.

`ansible all -m file -a "path=/var/archivos state=directory" -b`

```
nodo-rocky-1 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "gid": 0,
  "group": "root",
  "mode": "0755",
  "owner": "root",
  "path": "/var/archivos",
  "size": 6,
  "state": "directory",
  "uid": 0
}
```

```
nodo-rocky-2 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "gid": 0,
  "group": "root",
  "mode": "0755",
  "owner": "root",
  "path": "/var/archivos",
  "size": 6,
  "state": "directory",
  "uid": 0
}
```

4) EJECUCIÓN EN MODO AD-HOC

b) Crear un directorio /var/archivos en todos los servidores y copiar un archivo ejemplo.txt (con un contenido arbitrario) a esa ubicación.

vim ejemplo.txt

ansible all -m copy -a "src=ejemplo.txt dest=/var/archivos/ejemplo.txt" -b

```
[root@server group_vars]# ansible all -m copy -a "src=ejemplo.txt dest=/var/archivos/ejemplo.txt" -b
nodo-debian | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "checksum": "b4501670f5c843ca4a3e33316053aa4eb4daa70e",
  "dest": "/var/archivos/ejemplo.txt",
  "gid": 0,
  "group": "root",
  "md5sum": "a9a6841749b240d0624e9685fe90f7a3",
  "mode": "0644",
  "owner": "root",
  "size": 31,
  "src": "/home/ansible/.ansible/tmp/ansible-tmp-1715855165.659664-27583-187957829630002/source",
  "state": "file",
  "uid": 0
}
```

```
nodo-ubuntu | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "checksum": "b4501670f5c843ca4a3e33316053aa4eb4daa70e",
  "dest": "/var/archivos/ejemplo.txt",
  "gid": 0,
  "group": "root",
  "md5sum": "a9a6841749b240d0624e9685fe90f7a3",
  "mode": "0644",
  "owner": "root",
  "size": 31,
  "src": "/home/ansible/.ansible/tmp/ansible-tmp-1715855166.376985-27584-36690607547821/source",
  "state": "file",
  "uid": 0
}
```

4) EJECUCIÓN EN MODO AD-HOC

b) Crear un directorio /var/archivos en todos los servidores y copiar un archivo ejemplo.txt (con un contenido arbitrario) a esa ubicación.

vim ejemplo.txt

ansible all -m copy -a "src=ejemplo.txt dest=/var/archivos/ejemplo.txt" -b

```
nodo-opensuse | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.6"
  },
  "changed": true,
  "checksum": "b4501670f5c843ca4a3e33316053aa4eb4daa70e",
  "dest": "/var/archivos/ejemplo.txt",
  "gid": 0,
  "group": "root",
  "md5sum": "a9a6841749b240d0624e9685fe90f7a3",
  "mode": "0644",
  "owner": "root",
  "size": 31,
  "src": "/home/ansible/.ansible/tmp/ansible-tmp-1715855167",
  "state": "file",
  "uid": 0
}
```

```
nodo-rocky-2 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "checksum": "b4501670f5c843ca4a3e33316053aa4eb4daa70e",
  "dest": "/var/archivos/ejemplo.txt",
  "gid": 0,
  "group": "root",
  "md5sum": "a9a6841749b240d0624e9685fe90f7a3",
  "mode": "0644",
  "owner": "root",
  "size": 31,
  "src": "/home/ansible/.ansible/tmp/ansible-tmp-1715855168",
  "state": "file",
  "uid": 0
}
```

```
nodo-rocky-1 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "checksum": "b4501670f5c843ca4a3e33316053aa4eb4daa70e",
  "dest": "/var/archivos/ejemplo.txt",
  "gid": 0,
  "group": "root",
  "md5sum": "a9a6841749b240d0624e9685fe90f7a3",
  "mode": "0644",
  "owner": "root",
  "size": 31,
  "src": "/home/ansible/.ansible/tmp/ansible-tmp-1715855",
  "state": "file",
  "uid": 0
}
```

4) EJECUCIÓN EN MODO AD-HOC

b) Crear un directorio /var/archivos en todos los servidores y copiar un archivo ejemplo.txt (con un contenido arbitrario) a esa ubicación.

OpenSuse (.146)

```
ansible@server:~> ls /var/archivos  
ejemplo.txt
```

Ubuntu (.137)

```
root@server:~# ls /var/archivos  
ejemplo.txt
```

Rocky 1 (.147)

```
[ansible@server ~]$ ls /var/archivos  
ejemplo.txt
```

Rocky 2 (.148)

```
[ansible@server ~]$ ls /var/archivos  
ejemplo.txt
```

Debian (.149)

```
root@server:~# ls /var/archivos  
ejemplo.txt
```

4) EJECUCIÓN EN MODO AD-HOC

c) Actualizar el sistema operativo del equipo Debian (Nota: puede ser un proceso lento)

```
ansible Debian -i /etc/ansible/hosts -m apt -a "upgrade=dist" -b
```

archivo de inventario -- > vim /etc/ansible/hosts

```
[Debian]
nodo_debian| ansible_host=192.168.119.149
```

vim /etc/ansible/groups_var/ Debian

```
ansible_user: ansible
ansible_ssh_private_key_file: ~/.ssh/id_rsa
```

```
[root@server group_vars]# ansible Debian -i /etc/ansible/hosts -m apt -a "upgrade=dist" -b
nodo_debian | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "msg": "Reading package lists...\nBuilding dependency tree...\nReading state information
ulating upgrade...\nThe following NEW packages will be installed:\n  linux-image-5.10.0-29-a
following packages will be upgraded:\n  linux-image-amd64\n1 upgraded, 1 newly installed, 0
e and 0 not upgraded.\nNeed to get 55.7 MB of archives.\nAfter this operation, 318 MB of add
isk space will be used.\nGet:1 http://security.debian.org/debian-security bullseye-security/
4 linux-image-5.10.0-29-amd64 amd64 5.10.216-1 [55.7 MB]\nGet:2 http://security.debian.org/d
urity bullseye-security/main amd64 linux-image-amd64 amd64 5.10.216-1 [1484 B]\napt-listchan
ing changelogs...\nFetched 55.7 MB in 5s (11.5 MB/s)\nSelecting previously unselected packag
mage-5.10.0-29-amd64.\r\n(Reading database ... \r(Reading database ... 5%\r(Reading database
62%\r(Reading database ... 77%\r(Reading database ... 92%\r(Reading database ... 100%\r(Rea
```


4) EJECUCIÓN EN MODO AD-HOC

d) Comprobar el funcionamiento de los módulos lineinfile y blockinfile, realizando distintas tareas sobre archivos ubicados en algún nodo administrado: Añadir líneas y bloques de texto a un archivo, borrar líneas o bloques de texto existentes, cambiar el valor de una directiva (por ejemplo, en un archivo .conf, cambiar un valor “yes” por “no” en una directiva, o activar una nueva directiva con el valor deseado). Comprobar el efecto de la directiva backup. En el caso del módulo blockinfile, estudiar también la propiedad Marker e indicar cuál es su función.

- 1) Añadir líneas a un archivo de Rocky 1
- 2) Borrar un bloque a un archivo de OpenSuse
- 3) Cambiar el valor de una directiva de apache de la máquina Debian
- 4) Comprobar efecto de la directiva “backup”
- 5) Estudiar la propiedad Marker

- 1) Añadir líneas a un archivo de Rocky 1

```
ansible nodo-rocky-1 -m lineinfile -a "path=/home/ansible/poesia_becquer.txt  
line='Poesía de Gustavo Adolfo Bécquer' state=present"
```

```
[ansible@server ~]$ pwd  
/home/ansible  
[ansible@server ~]$ vim poesia_becquer.txt
```

```
nodo-rocky-1 | CHANGED => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "backup": "",  
  "changed": true,  
  "msg": "line added"  
}
```

```
cat poseia_becquer.txt
```

```
y en cada humano corazón resuena.  
Poesía de Gustavo Adolfo Bécquer  
[ansible@server ~]$
```

4) EJECUCIÓN EN MODO AD-HOC

d) Comprobar el funcionamiento de los módulos lineinfile y blockinfile, realizando distintas tareas sobre archivos ubicados en algún nodo administrado: Añadir líneas y bloques de texto a un archivo, borrar líneas o bloques de texto existentes, cambiar el valor de una directiva (por ejemplo, en un archivo .conf, cambiar un valor “yes” por “no” en una directiva, o activar una nueva directiva con el valor deseado). Comprobar el efecto de la directiva backup. En el caso del módulo blockinfile, estudiar también la propiedad Marker e indicar cuál es su función.

2) Borrar un bloque a un archivo de OpenSuse

```
ansible@server:~> pwd
/home/ansible
ansible@server:~> vim poesia_becquer.txt
```

```
como yo te he querido... desengáñate,
nadie te querrá.

#INICIO-BLOQUE|
Cuando me lo contaron sentí el frío
de una hoja de acero en las entrañas,
me apoyé contra el muro, y un instante
la conciencia perdí de dónde estaba.
Cayó sobre mi espíritu la noche
en ira y en piedad se anegó el alma ¡y entonces
¡y entonces comprendí por qué se mata!
Pasó la nube de dolor... con pena
logré balbucear breves palabras...
¿Quién me dio la noticia?... Un fiel amigo...
Me hacía un gran favor... Le di las gracias.

Podrá nublarse el sol eternamente;
```

Vamos a
poner un
marcador en
el bloque de
la poesía de
Bécquer que
se va a
borrar

```
ansible nodo-opensuse -m blockinfile -a
"path=/home/ansible/poesia_becquer.txt marker='#INICIO-BLOQUE'
state=absent"
```

Nota: No hace falta el marcador, se puede pegar todo el bloque pero es bastante largo

```
nodo-opensuse | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.6"
  },
  "changed": true,
  "msg": "Block removed"
}
```

```
como yo te he querido... desengáñate,
nadie te querrá.
```

```
Cuando me lo contaron sentí el frío
de una hoja de acero en las entrañas,
me apoyé contra el muro, y un instante
```

4) EJECUCIÓN EN MODO AD-HOC

d) Comprobar el funcionamiento de los módulos lineinfile y blockinfile, realizando distintas tareas sobre archivos ubicados en algún nodo administrado: Añadir líneas y bloques de texto a un archivo, borrar líneas o bloques de texto existentes, cambiar el valor de una directiva (por ejemplo, en un archivo .conf, cambiar un valor “yes” por “no” en una directiva, o activar una nueva directiva con el valor deseado). Comprobar el efecto de la directiva backup. En el caso del módulo blockinfile, estudiar también la propiedad Marker e indicar cuál es su función.

3) Cambiar el valor de una directiva de apache de la máquina Debian

4) Comprobar efecto de la directiva “backup”

En Rocky Admin; lynx direcc_ip_debian

```
Debian Logo Apache2 Debian Default Page
It works!

This is the default welcome page used to test the correct operation of the
after installation on Debian systems. If you can read this page, it means
HTTP server installed at this site is working properly. You should replace
at /var/www/html/index.html before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page
probably means that the site is currently unavailable due to maintenance.
persists, please contact the site's administrator.
Configuration Overview

Debian's Apache2 default configuration is different from the upstream default
and split into several files optimized for interaction with Debian tools.
system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Read
full documentation. Documentation for the web server itself can be found
manual if the apache2-doc package was installed on this server.
```

Vamos a cambiar la directiva DirectoryIndex a paloma.html:

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
    DirectoryIndex index.html
</Directory>
```

4) EJECUCIÓN EN MODO AD-HOC

d) Comprobar el funcionamiento de los módulos lineinfile y blockinfile, realizando distintas tareas sobre archivos ubicados en algún nodo administrado: Añadir líneas y bloques de texto a un archivo, borrar líneas o bloques de texto existentes, cambiar el valor de una directiva (por ejemplo, en un archivo .conf, cambiar un valor “yes” por “no” en una directiva, o activar una nueva directiva con el valor deseado). Comprobar el efecto de la directiva backup. En el caso del módulo blockinfile, estudiar también la propiedad Marker e indicar cuál es su función.

3) Cambiar el valor de una directiva de apache de la máquina Debian

4) Comprobar efecto de la directiva “backup”

```
ansible nodo-debian -m replace -a "path=/etc/apache2/apache2.conf regexp='(DirectoryIndex\s+)index.html'
replace='\1paloma.html' backup=yes" -b
```

- Módulo replace (**-m replace**)
- **regexp** es la expresión regular para encontrar la línea que contiene DirectoryIndex index.html
- **replace** es la cadena de reemplazo que sustituirá index.html por paloma.html
- **backup=yes** indica que deseas crear una copia de seguridad del archivo antes de realizar los cambios

```
[root@server ~]# ansible nodo-debian -m replace -a "path=/etc/apache2/ap
Index\s+)index.html' replace='\1paloma.html' backup=yes" -b
nodo-debian | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "backup_file": "/etc/apache2/apache2.conf.772.2024-05-16@15:44:35~",
  "changed": true,
  "msg": "1 replacements made",
  "rc": 0
}
```

4) EJECUCIÓN EN MODO AD-HOC

d) Comprobar el funcionamiento de los módulos lineinfile y blockinfile, realizando distintas tareas sobre archivos ubicados en algún nodo administrado: Añadir líneas y bloques de texto a un archivo, borrar líneas o bloques de texto existentes, cambiar el valor de una directiva (por ejemplo, en un archivo .conf, cambiar un valor “yes” por “no” en una directiva, o activar una nueva directiva con el valor deseado). Comprobar el efecto de la directiva backup. En el caso del módulo blockinfile, estudiar también la propiedad Marker e indicar cuál es su función.

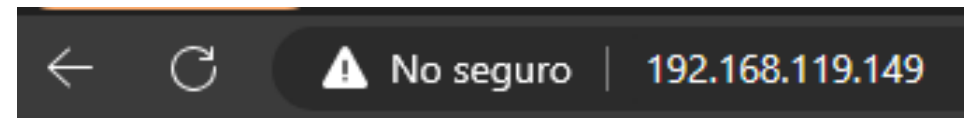
3) Cambiar el valor de una directiva de apache de la máquina Debian

4) Comprobar efecto de la directiva “backup”

cat /etc/apache2/apache2.conf en Debian


```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
    DirectoryIndex paloma.html
</Directory>
```

systemctl restart apache2



Paloma

¡Hola! Soy una paloma.

 Imagen de una paloma

4) EJECUCIÓN EN MODO AD-HOC

d) Comprobar el funcionamiento de los módulos lineinfile y blockinfile, realizando distintas tareas sobre archivos ubicados en algún nodo administrado: Añadir líneas y bloques de texto a un archivo, borrar líneas o bloques de texto existentes, cambiar el valor de una directiva (por ejemplo, en un archivo .conf, cambiar un valor “yes” por “no” en una directiva, o activar una nueva directiva con el valor deseado). Comprobar el efecto de la directiva backup. En el caso del módulo blockinfile, estudiar también la propiedad Marker e indicar cuál es su función.

3) Cambiar el valor de una directiva de apache de la máquina Debian

4) Comprobar efecto de la directiva “backup”

Comprobamos que se ha guardado el backup en la Debian:

```
root@server:/etc/apache2# ls
apache2.conf          conf-enabled  mods-available  sites-available
apache2.conf.772.2024-05-16@15:44:35~ envvars      mods-enabled    sites-enabled
conf-available        magic        ports.conf
root@server:/etc/apache2#
```

Si lo repetimos sin el backup, no nos saldrá ese archivo:

```
ansible nodo-debian -m replace -a
"path=/etc/apache2/apache2.conf
regex='(DirectoryIndex\s+)paloma.html'
replace='\1index.html' " -b
```

```
nodo-debian | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "msg": "1 replacements made",
  "rc": 0
}
```

4) EJECUCIÓN EN MODO AD-HOC

d) Comprobar el funcionamiento de los módulos `lineinfile` y `blockinfile`, realizando distintas tareas sobre archivos ubicados en algún nodo administrado: Añadir líneas y bloques de texto a un archivo, borrar líneas o bloques de texto existentes, cambiar el valor de una directiva (por ejemplo, en un archivo `.conf`, cambiar un valor “yes” por “no” en una directiva, o activar una nueva directiva con el valor deseado). Comprobar el efecto de la directiva `backup`. En el caso del módulo `blockinfile`, estudiar también la propiedad `Marker` e indicar cuál es su función.

5) Estudiar la propiedad `Marker`

La propiedad `Marker` es una característica específica de este módulo que se utiliza para indicar el inicio y el final del bloque de contenido que se va a modificar. Ya la hemos utilizado en el apartado “2) Borrar un bloque a un archivo de OpenSuse” .

La propiedad `Marker` tiene dos partes:

- `begin`: Indica el texto que marca el inicio del bloque de contenido.
- `end`: Indica el texto que marca el final del bloque de contenido.

En el caso de la poesía de Bécquer podríamos hacer:

```
ansible nodo-opensuse -m blockinfile -a "path=/home/ansible/poesia_becquer.txt marker='#INICIO-BLOQUE'
block='Paloma ha esto aquí :)' marker_end='#FIN-BLOQUE'"
```

4) EJECUCIÓN EN MODO AD-HOC

d) Comprobar el funcionamiento de los módulos lineinfile y blockinfile, realizando distintas tareas sobre archivos ubicados en algún nodo administrado: Añadir líneas y bloques de texto a un archivo, borrar líneas o bloques de texto existentes, cambiar el valor de una directiva (por ejemplo, en un archivo .conf, cambiar un valor “yes” por “no” en una directiva, o activar una nueva directiva con el valor deseado). Comprobar el efecto de la directiva backup. En el caso del módulo blockinfile, estudiar también la propiedad Marker e indicar cuál es su función.

5) Estudiar la propiedad Marker

```
ansible nodo-opensuse -m blockinfile -a "path=/home/ansible/poesia_becquer.txt marker='#INICIO-BLOQUE'
block='Paloma ha esto aquí :)' marker_end='#FIN-BLOQUE'"
```

```
logré balbucear breves palabras...
¿Quién me dio la noticia?... Un fiel amigo...
Me hacía un gran favor... Le di las gracias.

#INICIO-BLOQUE
#FIN-BLOQUE

Podrá nublarse el sol eternamente;
Podrá secarse en un instante el mar;
Podrá romperse el eje de la tierra
Como un débil cristal.
```

```
nodo-opensuse | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.6"
  },
  "changed": true,
  "msg": "Block inserted"
}
```

```
logré balbucear breves palabras...
¿Quién me dio la noticia?... Un fiel an
Me hacía un gran favor... Le di las gra

#INICIO-BLOQUE
Paloma ha esto aquí :)
#INICIO-BLOQUE
#FIN-BLOQUE

Podrá nublarse el sol eternamente;
Podrá secarse en un instante el mar;
Podrá romperse el eje de la tierra
Como un débil cristal.
```


4) EJECUCIÓN EN MODO AD-HOC

e) Comprobar el funcionamiento del módulo **template**. Preparar una plantilla en formato jinja2 (.j2) que utilice alguna variable de prueba previamente definida en el archivo group_vars/all. Lanzar el módulo template sobre una máquina administrada y verificar el resultado

Vamos a pegar ese mensaje en prueba.txt en Rocky 1

La estructura de búsqueda de variables en Ansible es la siguiente:

1. Variables definidas a nivel de host (en los archivos host_vars/hostname.yml o host_vars/hostname.yml).
2. Variables definidas a nivel de grupo (en los archivos group_vars/groupname.yml o group_vars/groupname.yml).
3. Variables definidas en el archivo group_vars/all.yml o group_vars/all.yml. Estas variables se aplican a todos los grupos de hosts.

Vamos a definir la variable en /etc/ansible/group_vars/all.yml

```
mensaje_paloma: "Paloma ha estado aquí"
```

Creamos una plantilla: template_prueba.j2

```
[root@server group_vars]# vim template_prueba.j2
[root@server group_vars]# cat template_prueba.j2
Este es un mensaje de prueba: {{ mensaje_paloma }}
```

4) EJECUCIÓN EN MODO AD-HOC

e) Comprobar el funcionamiento del módulo **template**. Preparar una plantilla en formato jinja2 (.j2) que utilice alguna variable de prueba previamente definida en el archivo group_vars/all. Lanzar el módulo template sobre una máquina administrada y verificar el resultado

```
ansible nodo-rocky-1 -m template -a "src=mensaje_template.j2 dest=/home/ansible/prueba.txt" -b
```

```
all.yaml  dns_esclavo  ejemplo.txt  template_prueba.j2
[root@server group_vars]# ansible nodo-rocky-1 -m template -a "src=template_prueba.j2 dest=/home/ansible/prueba.txt" -b
nodo-rocky-1 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "checksum": "fdc35c24b7de0ecbe86930bc0f530ad56d8af72d",
  "dest": "/home/ansible/prueba.txt",
  "gid": 1000,
  "group": "ansible",
  "md5sum": "0af7166b1b4a382be2cb256f82bcd8b4",
  "mode": "0644",
  "owner": "ansible",
  "size": 53,
  "src": "/home/ansible/.ansible/tmp/ansible-tmp-1715871333.3394682-28212-235311526113199/source",
  "state": "file",
  "uid": 1000
}
```

```
[ansible@server ~]$ cat prueba.txt
Este es un mensaje de prueba: Paloma ha estado aquí
[ansible@server ~]$ |
```

En Rocky 1

ÍNDICE

2) Preparación del entorno

3) Configuración de Ansible

4) Ejecución en modo ad-hoc

5) Libro de jugadas

5) LIBROS DE JUGADAS

- a) Crear un libro de jugadas (servidoresweb.yml) para instalar la última versión del servidor Apache (paquete httpd o apache2, dependiendo de la distribución) y configurar un nuevo proyecto web, como servidor virtual, en las máquinas administradas. Esta jugada se aplicará sobre el grupo WebServers, cuyos miembros tienen instaladas distintas distribuciones Linux: Rocky, Ubuntu, Debian y OpenSUSE. El nombre de dominio del nuevo servidor (por ejemplo, prueba.com) será una variable pasada como parámetro en la llamada al playbook (usando --extra-vars). El proyecto web se ubicará en un directorio /var/projects/ del servidor administrado, y contendrá una página de inicio con el mensaje “Página de inicio del dominio”. Para facilitar la construcción se recomienda crear en la máquina administrada un archivo de configuración independiente para cada servidor virtual de Apache, con extensión .conf, y ubicarlo en un directorio de configuración de Apache (enlazado mediante un include, como los directorios conf.d o sites-enabled). El playbook deberá recoger en variables las características diferenciadas de las distribuciones, para llevar a cabo la misma tarea sobre todos los equipos, independientemente de su distribución. Por ejemplo, la instalación de Apache podría hacerse de la siguiente forma con el paquete package (la variable nombre_servicio contendrá la cadena httpd o apache2)

```
- name: Instalar Apache con el modulo package
  package:
    name={{nombre_servicio}}
    state=latest
```

- b) Crear un libro de jugadas (zonadns.yml) para instalar el servidor de DNS Bind y una zona de resolución directa (cuyo nombre se pasará como parámetro con --extra-vars en la llamada al playbook) en los servidores DNS_Maestro y DNS_Esclavo (como su nombre indica, el primero mantendrá la zona maestra y el segundo una esclava de la misma zona). Ambos servidores tendrán que configurar el servidor Bind para aceptar consultas desde cualquier origen y por cualquier interface, y no permitirán consultas recursivas. Las zonas creadas serán muy sencillas, con un registro SOA, dos registros NS (ns1 y ns2 de la misma zona, uno para cada servidor de DNS y con las direcciones IP que figuran en el inventario o en las propias máquinas Maestra y Esclava), dos registros A para ns1 y ns2, y otros dos registros A para los servicios web de el dominio que damos de alta (con y sin www, y ambos apuntando a una dirección ficticia 10.10.10.10). Se valorará el diseño utilizado: su estructura, originalidad, generalidad y simplicidad. Nota: los aspectos significativos de la configuración realizada en los apartados 2 a 5 tendrán que reflejarse en la memoria de la práctica.

5) LIBROS DE JUGADAS

- a) Crear un libro de jugadas (servidoresweb.yml) para instalar la última versión del servidor Apache (paquete httpd o apache2, dependiendo de la distribución) y configurar un nuevo proyecto web, como servidor virtual, en las máquinas administradas. Esta jugada se aplicará sobre el grupo WebServers, cuyos miembros tienen instaladas distintas distribuciones Linux: Rocky, Ubuntu, Debian y OpenSUSE. El nombre de dominio del nuevo servidor (por ejemplo, prueba.com) será una variable pasada como parámetro en la llamada al playbook (usando --extra-vars). El proyecto web se ubicará en un directorio /var/projects/ del servidor administrado, y contendrá una página de inicio con el mensaje “Página de inicio del dominio”. Para facilitar la construcción se recomienda crear en la máquina administrada un archivo de configuración independiente para cada servidor virtual de Apache, con extensión .conf, y ubicarlo en un directorio de configuración de Apache (enlazado mediante un include, como los directorios conf.d o sites-enabled). El playbook deberá recoger en variables las características diferenciadas de las distribuciones, para llevar a cabo la misma tarea sobre todos los equipos, independientemente de su distribución. Por ejemplo, la instalación de Apache podría hacerse de la siguiente forma con el paquete package (la variable nombre_servicio contendrá la cadena httpd o apache2)

```
- name: Instalar Apache con el modulo package
  package:
    name={{nombre_servicio}}
    state=latest
```

```
[WebServers]
nodo-rocky-1 ansible_host=192.168.119.147
nodo-debian ansible_host=192.168.119.149
nodo-ubuntu ansible_host=192.168.119.137
nodo-opensuse ansible_host=192.168.119.146
```

5) LIBROS DE JUGADAS

Playbook

Instalar Apache y configurar un servidor virtual, incluyendo tareas y variables globales.

Variables Específicas

Especifica las rutas y nombres del servicio Apache para diferentes máquinas administradas (con diferentes distribuciones Linux) .

Plantilla de Configuración

Define la configuración del servidor virtual Apache usando una plantilla Jinja2.

```
etc/
├── ansible/
│   ├── group_vars/
│   │   └── WebServers
│   ├── hosts
│   ├── roles/
│   ├── ansible.cfg/
│   ├── templates/
│   │   └── vhost.conf.j2
│   ├── hosts_vars/
│   │   └── apache_vars.yml
│   └── servidoresweb.yml
```

```
[WebServers]
nodo-rocky-1 ansible_host=192.168.119.147
nodo-debian ansible_host=192.168.119.149
nodo-ubuntu ansible_host=192.168.119.137
nodo-opensuse ansible_host=192.168.119.146
```

5) LIBROS DE JUGADAS

Playbook

Instalar Apache y configurar un servidor virtual, incluyendo tareas y variables globales.

Variables Específicas

Especifica las rutas y nombres del servicio Apache para diferentes máquinas administradas (con diferentes distribuciones Linux) .

Plantilla de Configuración

Define la configuración del servidor virtual Apache usando una plantilla Jinja2.

5) LIBROS DE JUGADAS

Playbook

Configuración Inicial

```
1 ---
2 - name: Configurar servidores web
3   hosts: WebServers
4   become: yes # Permisos de superusuario
5   vars:
6     apache_conf_dir: "{{ apache_conf_dirs[ansible_os_family] }}" # Directorio config apache (según distrib)
7     servicio_apache: "{{ apache_servicios[ansible_os_family] }}" # Nombre del servicio (según distrib)
8     apache_conf_file: "{{ apache_conf_files[ansible_os_family] }}" # Archivo principal de config (según distrib)
9     proyecto_dir: "{{ proyecto_dir_index[ansible_os_family] }}" # Directorio base del proyecto
10    dominio: ""
11    vars_files:
12      - hosts_vars/apache_vars.yml
```

- Variable dominio vacía → se espera que su valor sea proporcionado dinámicamente al ejecutar el playbook mediante --extra-vars
- Vars_files: hosts_vars/apache_vars.yml → Importa variables específicas

Variables
Específicas /hos

5) LIBROS DE JUGADAS

Playbook

Tareas (tasks)

Módulos a utilizar:

- File: crear directorio (archivo)
- Copy: copiar desde Admin Rocky
- Template: generar en base a la plantilla de servidores virtuales
- Command: ejecuta un comando

- state: present → asegura que el paquete está instalado
- state: directory → Especifica que el estado deseado es un directorio.
- when: condicional, el archivo “000-default.conf” solo existe en Ubuntu y Debian. Luego quitaremos el link en dichos casos

```
tasks:
  - name: Instalar Apache
    package:
      name: "{{ servicio_apache }}"
      state: present
    notify:
      - Reiniciar Apache

  - name: Habilitar Apache
    service:
      name: "{{ servicio_apache }}"
      enabled: yes
      state: started
    notify:
      - Apache habilitado

  - name: Deshabilitar sitio por defecto en Apache
    command: a2dissite 000-default.conf
    notify:
      - Reiniciar Apache
    when: ansible_os_family in ['Debian', 'Ubuntu']
```

5) LIBROS DE JUGADAS

Playbook

Tareas (tasks)

- owner & group → propietario del directorio “ansible”
- mode → permisos del directorio
 - 0755: lectura, escritura y ejecución para el propietario; lectura y ejecución para el grupo y otros
 - 0644: lectura y escritura para el propietario; solo lectura para el grupo y otros
- dest → destinatario (la ruta)
- notify → llamar a un handler/manejador (se ejecutarán cuando la tarea asociada finalice con éxito)

```
- name: Crear directorio del proyecto
  file:
    path: "{{ proyecto_dir }}/{{ dominio }}"
    state: directory
    owner: ansible
    group: ansible
    mode: '0755'
  notify:
    - Directorio del proyecto

- name: Crear página de inicio
  copy:
    dest: "{{ proyecto_dir }}/{{ dominio }}/index.html"
    content: "Página de inicio del dominio {{ dominio }}"
    owner: ansible
    group: ansible
    mode: '0644'
  notify:
    - Página de Inicio creada

- name: Configurar servidor virtual
  template:
    src: templates/vhost.conf.j2
    dest: "{{ apache_conf_dir }}/{{ dominio }}.conf"
  notify:
    - Reiniciar Apache
```

5) LIBROS DE JUGADAS

Playbook

Handlers(manejadores)

Handlers → son acciones que se ejecutan al final del playbook en respuesta a notificaciones emitidas por tareas específicas, como reiniciar servicios o ejecutar tareas adicionales después de completar ciertas operaciones.

```
handlers:
- name: Reiniciar Apache
  service:
    name: "{{ servicio_apache }}"
    state: restarted
- name: Apache habilitado
  debug:
    msg: "Apache habilitado (enabled)"
- name: Directorio del proyecto
  debug:
    msg: "El directorio del proyecto se ha creado"
- name: Página de Inicio creada
  debug:
    msg: "La página de inicio ha sido creada correctamente."
```

5) LIBROS DE JUGADAS

Playbook

— servidoresweb.yml

```
1 ---
2 - name: Configurar servidores web
3   hosts: WebServers
4   become: yes # Permisos de superusuario
5   vars:
6     apache_conf_dir: "{{ apache_conf_dirs[ansible_os_family] }}" # Directorio config apache (según distrib)
7     servicio_apache: "{{ apache_servicios[ansible_os_family] }}" # Nombre del servicio (según distrib)
8     apache_conf_file: "{{ apache_conf_files[ansible_os_family] }}" # Archivo principal de config (según distrib)
9     proyecto_dir: "{{ proyecto_dir_index[ansible_os_family] }}" # Directorio base del proyecto
10    dominio: ""
11  vars_files:
12    - hosts_vars/apache_vars.yml
13  tasks:
14    - name: Instalar Apache
15      package:
16        name: "{{ servicio_apache }}"
17        state: present
18      notify:
19        - Reiniciar Apache
20
21    - name: Habilitar Apache
22      service:
23        name: "{{ servicio_apache }}"
24        enabled: yes
25        state: started
26      notify:
27        - Apache habilitado
28
29    - name: Deshabilitar sitio por defecto en Apache
30      command: a2dissite 000-default.conf
31      notify:
32        - Reiniciar Apache
33      when: ansible_os_family in ['Debian', 'Ubuntu']
```

```
35 - name: Crear directorio del proyecto
36   file:
37     path: "{{ proyecto_dir }}/{{ dominio }}"
38     state: directory
39     owner: ansible
40     group: ansible
41     mode: '0755'
42   notify:
43     - Directorio del proyecto
44
45 - name: Crear página de inicio
46   copy:
47     dest: "{{ proyecto_dir }}/{{ dominio }}/index.html"
48     content: "Página de inicio del dominio {{ dominio }}"
49     owner: ansible
50     group: ansible
51     mode: '0644'
52   notify:
53     - Página de Inicio creada
54
55 - name: Configurar servidor virtual
56   template:
57     src: templates/vhost.conf.j2
58     dest: "{{ apache_conf_dir }}/{{ dominio }}.conf"
59   notify:
60     - Reiniciar Apache
61
62 handlers:
63   - name: Reiniciar Apache
64     service:
65       name: "{{ servicio_apache }}"
66       state: restarted
67   - name: Apache habilitado
68     debug:
69       msg: "Apache habilitado (enabled)"
70   - name: Directorio del proyecto
71     debug:
72       msg: "El directorio del proyecto se ha creado"
73   - name: Página de Inicio creada
74     debug:
75       msg: "La página de inicio ha sido creada correctamente."
```

5) LIBROS DE JUGADAS

Playbook

Instalar Apache y configurar un servidor virtual, incluyendo tareas y variables globales.

Variables Específicas

Especifica las rutas y nombres del servicio Apache para diferentes máquinas administradas (con diferentes distribuciones Linux) .

Plantilla de Configuración

Define la configuración del servidor virtual Apache usando una plantilla Jinja2.

5) LIBROS DE JUGADAS

Variables Específicas

apache_vars.yml

```
|— hosts_vars/  
   └─ apache_vars.yml
```

```
1  ---  
2  apache_conf_dirs:  
3    RedHat: /etc/httpd/conf.d  
4    Debian: /etc/apache2/sites-enabled  
5    Ubuntu: /etc/apache2/sites-enabled  
6    Suse: /etc/apache2/vhosts.d  
7  
8  apache_servicios:  
9    RedHat: httpd  
10   Debian: apache2  
11   Ubuntu: apache2  
12   Suse: apache2  
13  
14  apache_conf_files:  
15    RedHat: httpd.conf  
16    Debian: apache2.conf  
17    Ubuntu: apache2.conf  
18    Suse: httpd.conf  
19  
20  proyecto_dir_index:  
21    RedHat: /var/www/html  
22    Debian: /var/projects  
23    Ubuntu: /var/projects  
24    Suse: /srv/www/htdocs
```

```
"{{ apache_conf_dir }}"
```

```
"{{ servicio_apache }}"
```

```
"{{ apache_conf_dirs[ansible_os_family] }}" #  
"{{ apache_servicios[ansible_os_family] }}" #  
"{{ apache_conf_files[ansible_os_family] }}"
```

```
"{{ proyecto_dir_index[ansible_os_family] }}"
```

Nota: La distribución de OpenSuse es
"Suse" no "SUSE"

5) LIBROS DE JUGADAS

Playbook

Instalar Apache y configurar un servidor virtual, incluyendo tareas y variables globales.

Variables Específicas

Especifica las rutas y nombres del servicio Apache para diferentes máquinas administradas (con diferentes distribuciones Linux) .

Plantilla de Configuración

Define la configuración del servidor virtual Apache usando una plantilla Jinja2.

5) LIBROS DE JUGADAS

Plantilla de
Configuración

vhost.conf.j2

```
templates/  
└─ vhost.conf.j2
```

Ejemplo con prueba.com en ubuntu

```
1 ▼ <VirtualHost *:80>  
2  
3     ServerName {{ dominio }}  
4     DocumentRoot {{ proyecto_dir }}/{{ dominio }}  
5  
6 ▼     <Directory {{ proyecto_dir }}/{{ dominio }}>  
7         Options Indexes FollowSymLinks  
8         AllowOverride None  
9         Require all granted  
10    </Directory>  
11  
12 </VirtualHost>
```

```
<VirtualHost *:80>  
    ServerName prueba.com  
    DocumentRoot /var/projects/prueba.com  
  
    <Directory /var/projects/prueba.com>  
        Options Indexes FollowSymLinks  
        AllowOverride None  
        Require all granted  
    </Directory>  
</VirtualHost>
```


5) LIBROS DE JUGADAS

Playbook

Variables Específicas

Plantilla de Configuración

Ejecutamos el comando ansible:

```
ansible-playbook /etc/ansible/servidoresweb.yml --extra-vars "dominio=prueba.com" -b
```

```
PLAY RECAP *****
nodo-debian      : ok=8    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
nodo-opensuse    : ok=7    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
nodo-rocky-1     : ok=7    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
nodo-ubuntu      : ok=8    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Nota: Esto es después de varios intentos, si lo ejecutas a la primera saldría todo en amarillo (changed)

5) LIBROS DE JUGADAS

ansible-playbook /etc/ansible/servidoresweb.yml --extra-vars "dominio=prueba.com"

```
PLAY [Configurar servidores web] *****

TASK [Gathering Facts] *****
ok: [nodo-debian]
[WARNING]: Platform linux on host nodo-opensuse is not supported. A
of another Python interpreter could change the meaning of the warning
core/2.14/reference_appendices/interpreter_discovery.html for more
ok: [nodo-opensuse]
ok: [nodo-rocky-1]
ok: [nodo-ubuntu]

TASK [Instalar Apache] *****
ok: [nodo-debian]
ok: [nodo-ubuntu]
ok: [nodo-opensuse]
ok: [nodo-rocky-1]

TASK [Habilitar Apache] *****
ok: [nodo-debian]
ok: [nodo-ubuntu]
ok: [nodo-opensuse]
ok: [nodo-rocky-1]

TASK [Deshabilitar sitio por defecto en Apache] *****
skipping: [nodo-rocky-1]
skipping: [nodo-opensuse]
changed: [nodo-debian]
changed: [nodo-ubuntu]
```

```
TASK [Crear directorio del proyecto] *****
ok: [nodo-debian]
ok: [nodo-ubuntu]
ok: [nodo-opensuse]
ok: [nodo-rocky-1]

TASK [Crear página de inicio] *****
ok: [nodo-debian]
ok: [nodo-ubuntu]
ok: [nodo-opensuse]
ok: [nodo-rocky-1]

TASK [Configurar servidor virtual] *****
changed: [nodo-debian]
changed: [nodo-ubuntu]
changed: [nodo-opensuse]
changed: [nodo-rocky-1]

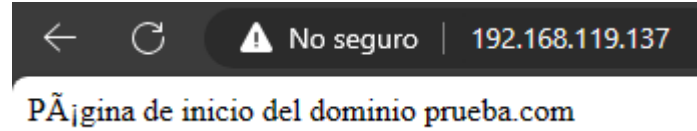
RUNNING HANDLER [Reiniciar Apache] *****
changed: [nodo-debian]
changed: [nodo-ubuntu]
changed: [nodo-opensuse]
changed: [nodo-rocky-1]
```

```
RUNNING HANDLER [Apache habilitado] *****
ok: [nodo-rocky-1] => {
    "msg": "Apache habilitado (enabled)"
}
ok: [nodo-debian] => {
    "msg": "Apache habilitado (enabled)"
}
ok: [nodo-ubuntu] => {
    "msg": "Apache habilitado (enabled)"
}
ok: [nodo-opensuse] => {
    "msg": "Apache habilitado (enabled)"
}
```

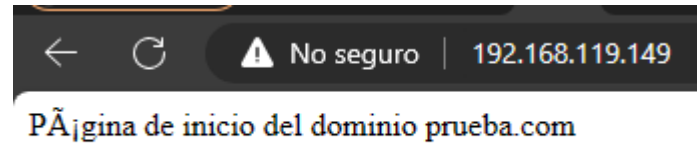
Nota: Si lo has hecho varias veces, el handler de crear el directorio (por ejemplo) no se va a activar ya que ya ha sido creado previamente.

5) LIBROS DE JUGADAS

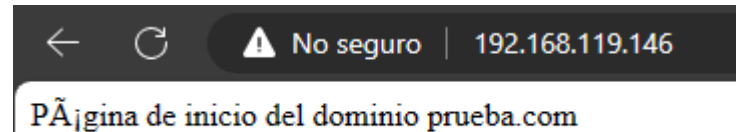
Nodo Ubuntu
192.168.119.137



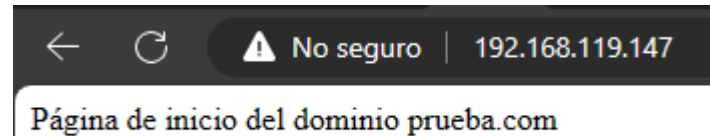
Nodo Debian
192.168.119.149



Nodo OpenSuse
192.168.119.146



Nodo Rocky 1
192.168.119.147



5) LIBROS DE JUGADAS

Errores que he cometido

- 1) No ejecutar el comando de ansible con permisos sudo (-b)
- 2) En Opensuse, ansible no pertenecía al grupo “ansible” (estaba definido en el playbook que tenía que ser así).

Error:

```
server:~ # groups ansible
ansible : users
```

Solución

```
server:~ # groupadd ansible
server:~ # e^C
server:~ # sudo usermod -aG ansible ansible
server:~ # groups ansible
ansible : users ansible
```

- 3) La distribución de OpenSuse se llama “Suse” y había puesto “SUSE”
- 4) En el archivo plantilla (vhost.conf.j2) había puesto las variables de apache_vars.yml cuando hay que poner las del playbook (así hay una variable distinta por cada distribución)
- 5) No hacer el a2dissite 000-default.conf

5) LIBROS DE JUGADAS

b) Crear un libro de jugadas (zonadns.yml) para instalar el servidor de DNS Bind y una zona de resolución directa (cuyo nombre se pasará como parámetro con `--extra-vars` en la llamada al playbook) en los servidores DNS_Maestro y DNS_Esclavo (como su nombre indica, el primero mantendrá la zona maestra y el segundo una esclava de la misma zona). Ambos servidores tendrán que configurar el servidor Bind para aceptar consultas desde cualquier origen y por cualquier interface, y no permitirán consultas recursivas.

Las zonas creadas serán muy sencillas, con un registro SOA, dos registros NS (ns1 y ns2 de la misma zona, uno para cada servidor de DNS y con las direcciones IP que figuran en el inventario o en las propias máquinas Maestra y Esclava), dos registros A para ns1 y ns2, y otros dos registros A para los servicios web de el dominio que damos de alta (con y sin www, y ambos apuntando a una dirección ficticia 10.10.10.10).

Playbook

Plantilla
archivo de zona

```
/etc/ansible/hosts
[DNS_Maestro]
nodo-rocky-1 ansible_host=192.168.119.147

[DNS_Esclavo]
nodo-rocky-2 ansible_host=192.168.119.148

[Servidores_DNS]
nodo-rocky-1 ansible_host=192.168.119.147
nodo-rocky-2 ansible_host=192.168.119.148
```

5) LIBROS DE JUGADAS

b) Crear un libro de jugadas (zonadns.yml) para instalar el servidor de DNS Bind y una zona de resolución directa (cuyo nombre se pasará como parámetro con `--extra-vars` en la llamada al playbook) en los servidores DNS_Maestro y DNS_Esclavo (como su nombre indica, el primero mantendrá la zona maestra y el segundo una esclava de la misma zona). Ambos servidores tendrán que configurar el servidor Bind para aceptar consultas desde cualquier origen y por cualquier interface, y no permitirán consultas recursivas.

Playbook


- 1) Instalar bind (+ Enable bind, no lo pide explícitamente, no lo pondremos, tenemos el ejemplo de cómo se hace en el otro playbook)
- 2) Configurar las opciones de named (recursion no; allow-query any; listen any;)
- 3) Configurar zona en Maestro
- 4) Configurar zona en Esclavo
- 5) Crear archivo de zona
- 6) Reiniciar servicio Bind

5) LIBROS DE JUGADAS

Playbook

```
/etc/ansible/hosts
[DNS_Maestro]
nodo-rocky-1 ansible_host=192.168.119.147

[DNS_Esclavo]
nodo-rocky-2 ansible_host=192.168.119.148
```



Vamos reemplazando:

- Listen-on port
- Allow-query (hemos añadido también allow-transfer)

Aunque no sirva para nada en el esclavo, nos simplifica el código.


- Recursion

```
---
- name: Configurar Bind y zonas DNS
  hosts: Servidores_DNS
  become: yes
  vars:
    ip_maestro: "{{ hostvars[groups['DNS_Maestro']][0]['ansible_host'] }}"
    ip_esclavo: "{{ hostvars[groups['DNS_Esclavo']][0]['ansible_host'] }}"
    zona_dns: ""
  tasks:
    - name: Instalar Bind
      package:
        name: bind
        state: present

    - name: Modificar listen-on en named.conf.options
      replace:
        path: /etc/named.conf
        regexp: '^(\s*)listen-on port 53 {[^}]*};' #Comillas simples no dobles
        replace: '\1listen-on port 53 { any; };'

    - name: Modificar allow-query en named.conf.options
      replace:
        path: /etc/named.conf
        regexp: '^(\s*)allow-query\s*{[^}]*};'
        replace: '\1allow-query { any; }; \n \1allow-transfer { {{ ip_esclavo }}; };'
        # En el servidor esclavo, allow-transfer no va a tener ningún efecto práctico
        # Se hace por no complicar más el playbook

    - name: Modificar recursion en named.conf.options
      replace:
        path: /etc/named.conf
        regexp: '^(\s*)recursion\s+yes;'
        replace: '\1recursion no;'
```



5) LIBROS DE JUGADAS

Playbook

```
named.conf
named.rfc1912.zones
named.root.key
```

El archivo `/etc/named.rfc1912.zones` es utilizado en Rocky como una práctica común para definir las zonas DNS de una manera organizada y conforme a las recomendaciones del RFC 1912

- Añadimos las zonas en ambos servidores
- Creamos el archivo de zona en el maestro
- Creamos el directorio “slaves” en el esclavo por si no existe (hay se guardará la copia del archivo de zona de del maestro)
- Reiniciamos bind

```
71 - name: Reiniciar servicio Bind
72   service:
73     name: named
74     state: restarted
```

```
36 - name: Configurar named.rfc1912.zones en Maestro
37   blockinfile:
38     path: /etc/named.rfc1912.zones
39     block: |
40       zone "{{ zona_dns }}" {
41         type master;
42         file "/var/named/db.{{ zona_dns }}";
43         allow-transfer { {{ ip_esclavo }}; };
44       };
45   when: "'DNS_Maestro' in group_names"
46
47 - name: Configurar named.rfc1912.zones en Esclavo
48   blockinfile:
49     path: /etc/named.rfc1912.zones
50     block: |
51       zone "{{ zona_dns }}" {
52         type slave;
53         file "/var/named/slaves/db.{{ zona_dns }}";
54         masters { {{ ip_maestro }}; };
55       };
56   when: "'DNS_Esclavo' in group_names"
57
58 - name: Crear archivo de zona en el maestro
59   template:
60     src: templates/archivozona.j2
61     dest: "/var/named/db.{{ zona_dns }}"
62   when: "'DNS_Maestro' in group_names"
63
64 - name: Crear directorio de zona en el esclavo
65   file:
66     path: /var/named/slaves
67     state: directory
68     mode: '0755'
69   when: "'DNS_Esclavo' in group_names"
```


5) LIBROS DE JUGADAS

Plantilla archivo de zona

Las zonas creadas serán muy sencillas, con un registro SOA, dos registros NS (ns1 y ns2 de la misma zona, uno para cada servidor de DNS y con las direcciones IP que figuran en el inventario o en las propias máquinas Maestra y Esclava), dos registros A para ns1 y ns2, y otros dos registros A para los servicios web de el dominio que damos de alta (con y sin www, y ambos apuntando a una dirección ficticia 10.10.10.10).

```
1 $TTL 604800
2 @ IN SOA ns1.{{ zona_dns }}. admin.{{ zona_dns }}. (
3     1          ; Serial
4     604800     ; Refresh
5     86400      ; Retry
6     2419200    ; Expire
7     604800 )   ; Negative Cache TTL
8 ;
9 @ IN NS ns1.{{ zona_dns }}.
10 @ IN NS ns2.{{ zona_dns }}.
11
12 ns1 IN A {{ ip_maestro }}
13 ns2 IN A {{ ip_esclavo }}
14
15 @ IN A 10.10.10.10 ;prueba.com
16 www IN A 10.10.10.10 ;www.prueba.com
```

```
$TTL 604800
@ IN SOA ns1.prueba.com. admin.prueba.com. (
    1          ; Serial
    604800     ; Refresh
    86400      ; Retry
    2419200    ; Expire
    604800 )   ; Negative Cache TTL
;
@ IN NS ns1.prueba.com.
@ IN NS ns2.prueba.com.

ns1 IN A 192.168.119.147
ns2 IN A 192.168.119.148

@ IN A 10.10.10.10 ;prueba.com
www IN A 10.10.10.10 ;www.prueba.com
```

5) LIBROS DE JUGADAS

Playbook

```
ansible-playbook zonasdns.yml --extra-vars "zona_dns=prueba.com" -b
```

```
PLAY [Configurar Bind y zonas DNS] *****
TASK [Gathering Facts] *****
ok: [nodo-rocky-2]
ok: [nodo-rocky-1]

TASK [Instalar Bind] *****
ok: [nodo-rocky-2]
ok: [nodo-rocky-1]

TASK [Modificar listen-on en named.conf.options] **
changed: [nodo-rocky-1]
changed: [nodo-rocky-2]

TASK [Modificar allow-query en named.conf.options]
changed: [nodo-rocky-2]
changed: [nodo-rocky-1]

TASK [Modificar recursion en named.conf.options] **
changed: [nodo-rocky-1]
changed: [nodo-rocky-2]

TASK [Configurar named.rfc1912.zones en Maestro] **
skipping: [nodo-rocky-2]
changed: [nodo-rocky-1]
```

```
PLAY RECAP *****
nodo-rocky-1      : ok=8    changed=6    unreachable=0    failed=0    skipped=2    rescued=0
                  ignored=0
nodo-rocky-2      : ok=8    changed=6    unreachable=0    failed=0    skipped=2    rescued=0
                  ignored=0
```

```
TASK [Configurar named.rfc1912.zones en Esclavo]
skipping: [nodo-rocky-1]
changed: [nodo-rocky-2]

TASK [Crear archivo de zona en el maestro] *****
skipping: [nodo-rocky-2]
changed: [nodo-rocky-1]

TASK [Crear directorio de zona en el esclavo] **
skipping: [nodo-rocky-1]
changed: [nodo-rocky-2]

TASK [Reiniciar servicio Bind] *****
changed: [nodo-rocky-1]
changed: [nodo-rocky-2]
```

5) LIBROS DE JUGADAS

Rocky 2 (Esclavo)

```
[root@server named]# ls
data dynamic named.ca named.empty named.localhost named.loopback slaves
[root@server named]# ls slaves/
db.prueba.com
```

```
[root@server named]# dig @localhost www.prueba.com

;<<>> DiG 9.16.23-RH <<>> @localhost www.prueba.com
(2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 54504
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 3f150e33c699bf15010000006648ba08ea5f6981615b15f2 (good)
;; QUESTION SECTION:
;www.prueba.com.                IN      A

;; ANSWER SECTION:
www.prueba.com.                604800 IN      A      10.10.10.10

;; Query time: 1 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sat May 18 16:24:08 CEST 2024
;; MSG SIZE rcvd: 87
```

```
[root@server etc]# cat named.conf
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

options {
    listen-on port 53 { any; };
    listen-on-v6 port 53 { ::1; };
    directory "/var/named";
    dump-file "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    secroots-file "/var/named/data/named.secroots";
    recursing-file "/var/named/data/named.recursing";

    allow-query { any; };

    allow-transfer { 192.168.119.148; };

    /*
     * If you are building an AUTHORITATIVE DNS server, do NOT enable
     * recursion.
     * If you are building a RECURSIVE (caching) DNS server, you need
     * to enable recursion.
     * If your recursive DNS server has a public IP address, you MUST
     * control access to your server to prevent it from being used as
     * a part of large scale DNS amplification attacks. Implementing
     * BCP38 within your network would greatly reduce such attack
     * surface.
     */
    recursion no;
```

5) LIBROS DE JUGADAS

Rocky 1 (Maestro)

```
[root@server named]# ls
data db.prueba.com dynamic named.ca named.empty
[root@server named]# cat db.prueba.com
$TTL 604800
@ IN SOA ns1.prueba.com. admin.prueba.com. (
        1          ; Serial
        604800     ; Refresh
        86400      ; Retry
        2419200    ; Expire
        604800 )   ; Negative Cache TTL
;
@      IN  NS   ns1.prueba.com.
@      IN  NS   ns2.prueba.com.

ns1    IN  A    192.168.119.147
ns2    IN  A    192.168.119.148

@      IN  A    10.10.10.10 ;prueba.com
www    IN  A    10.10.10.10 ;www.prueba.com
```

```
options {
    listen-on port 53 { any; };
    listen-on-v6 port 53 { ::1; };
    directory "/var/named";
    dump-file "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    secroots-file "/var/named/data/named.secroots";
    recursing-file "/var/named/data/named.recursing";

    allow-query { any; };

    | allow-transfer { 192.168.119.148; };

    /*
     * - If you are building an AUTHORITATIVE DNS server, do NOT
     * - If you are building a RECURSIVE (caching) DNS server, you
     *   recursion.
     * - If your recursive DNS server has a public IP address, you
     *   control to limit queries to your legitimate users. Failure
     *   cause your server to become part of large scale DNS amplification
     *   attacks. Implementing BCP38 within your network would greatly
     *   reduce such attack surface
     */
    recursion no;
```

5) LIBROS DE JUGADAS

Rocky 1 (Maestro)

Preguntándole a la máquina esclavo (.148)

```
[root@server named]# dig @localhost prueba.com
; <<>> DiG 9.16.23-RH <<>> @localhost prueba.com
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45756
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: f24a9637632d5690010000006648ba96e5dce806fc383272 (good)
;; QUESTION SECTION:
;prueba.com.                IN      A

;; ANSWER SECTION:
prueba.com.                604800  IN      A      10.10.10.10

;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sat May 18 16:26:30 CEST 2024
;; MSG SIZE rcvd: 83
```

```
[root@server named]# dig @192.168.119.148 prueba.com
; <<>> DiG 9.16.23-RH <<>> @192.168.119.148 prueba.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 371
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 31029bfced6ac7b4010000006648bab46466809a61e3fd11 (good)
;; QUESTION SECTION:
;prueba.com.                IN      A

;; ANSWER SECTION:
prueba.com.                604800  IN      A      10.10.10.10

;; Query time: 2 msec
;; SERVER: 192.168.119.148#53(192.168.119.148)
;; WHEN: Sat May 18 16:27:00 CEST 2024
;; MSG SIZE rcvd: 83
```

A series of thin, light brown lines forming an abstract geometric pattern on the left side of the slide. The lines intersect to create various polygons and shapes, extending from the top left towards the bottom left.

GRACIAS

Práctica 9 de Ansible

Paloma Pérez de Madrid