
OBLIGATORIO



Materia:

TALLER DE SERVIDORES LINUX

Docentes:

Sebastian Orrego

Andres Tarallo

Roberto Wagner

Integrantes del Grupo:

Marcos Sansberro, Diego González, Ricardo Robledo



Índice

Índice	1
Introducción	2
Creación de máquinas virtuales	2
CentOS 8	2
Instalación	4
Ubuntu	6
Clave pública para conexión SSH	7
Trabajando en el repositorio	9
Ejecución de la Play para el rol Common	12
Ejecución de la Play para el rol Web	13
Ejecución de la Play para el rol DB	14
Modificaciones en los servidores	15

Introducción

El objetivo principal de este documento es demostrar la configuración e instalación de las máquinas virtuales (CentOS 8 y Ubuntu 20.04) las cuales se utilizarán durante el desarrollo de este obligatorio, funcionando como sí cómo dar evidencia del correcto funcionamiento del código Ansible creado por nuestro equipo.

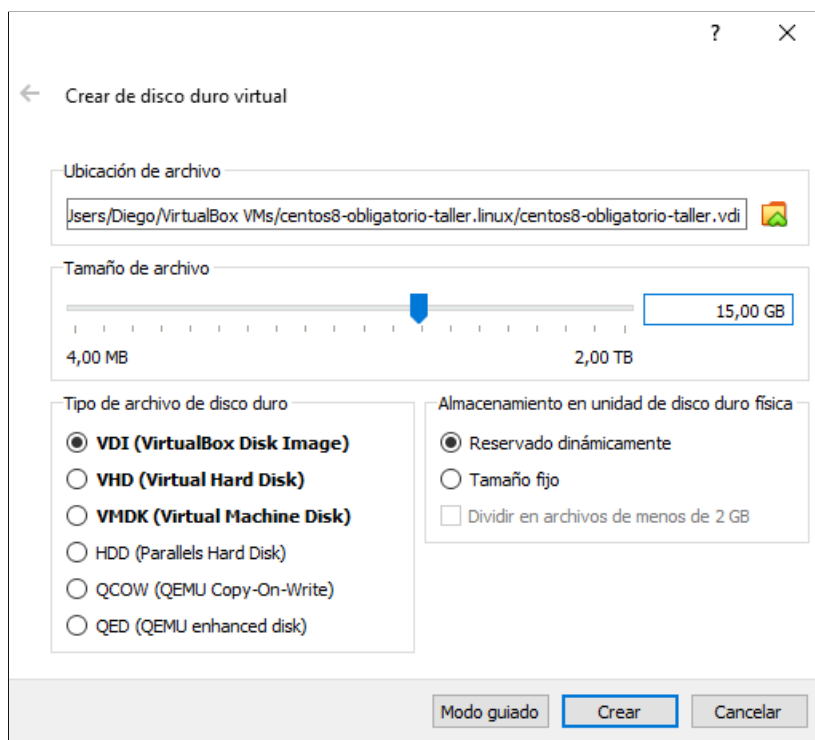
Los cambios técnicos específicos serán descritos en el archivo “README.md” en el directorio LAMP del repositorio. Al repositorio se le ha aplicado un Fork y fue descargado para ser trabajado en Visual Studio Code.

Repositorio: https://github.com/UDiegoGon/obligatorio_2021_08

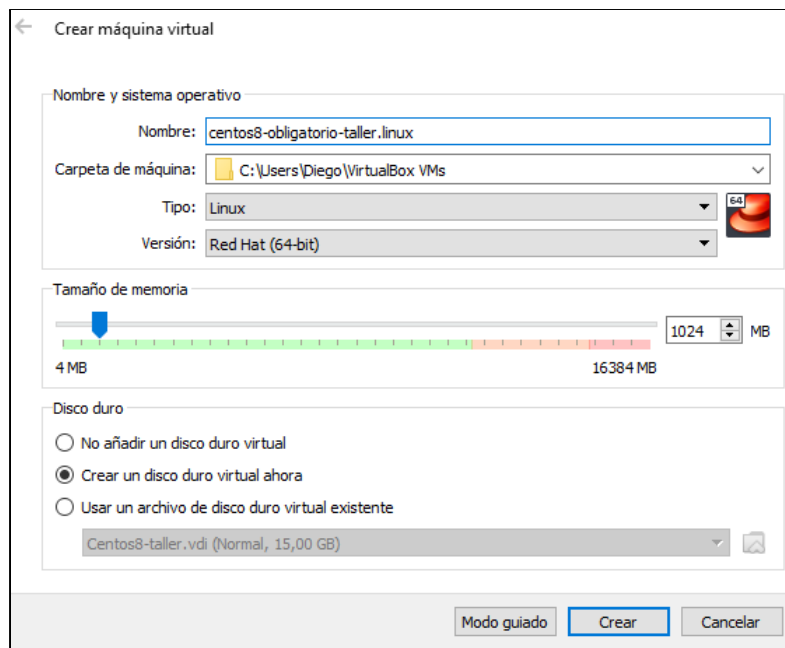
Creación de máquinas virtuales

CentOS 8

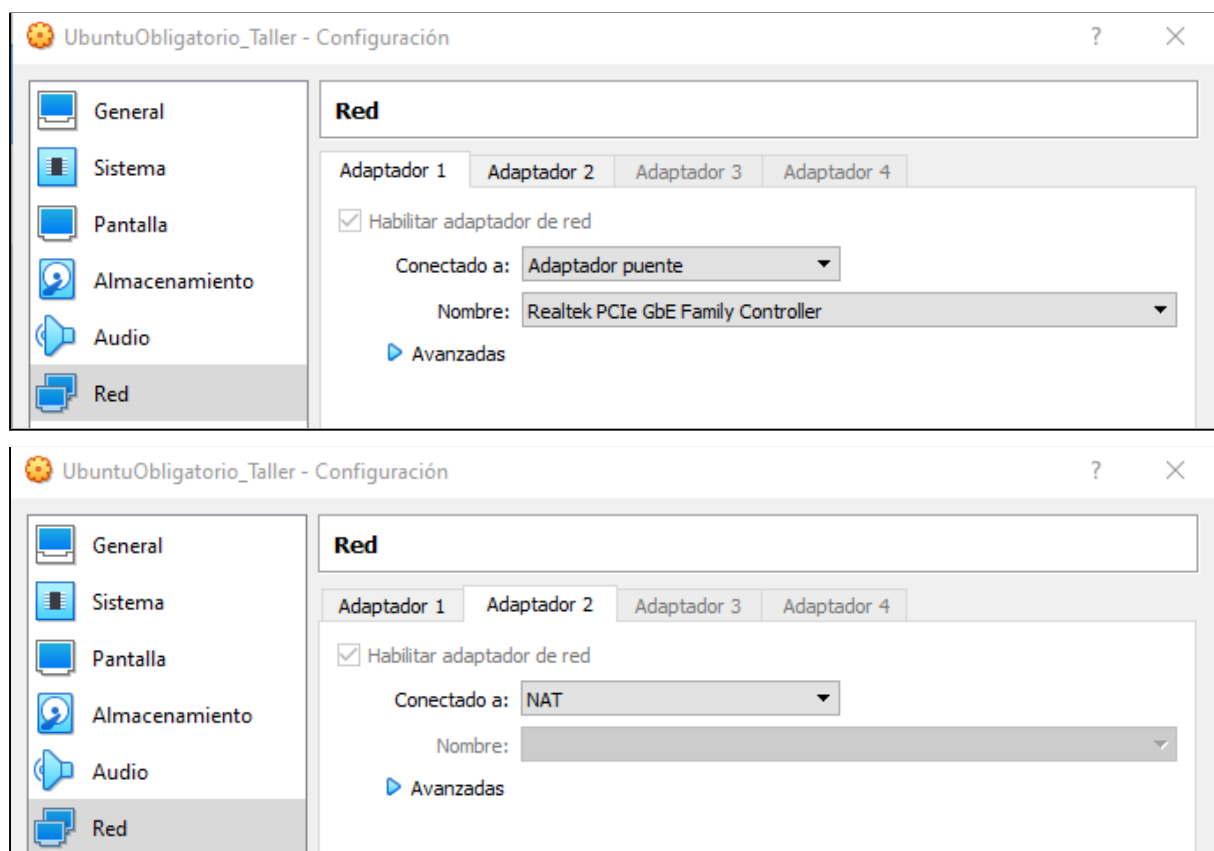
Nombramos la máquina virtual y le asignamos 1 GB de memoria RAM:



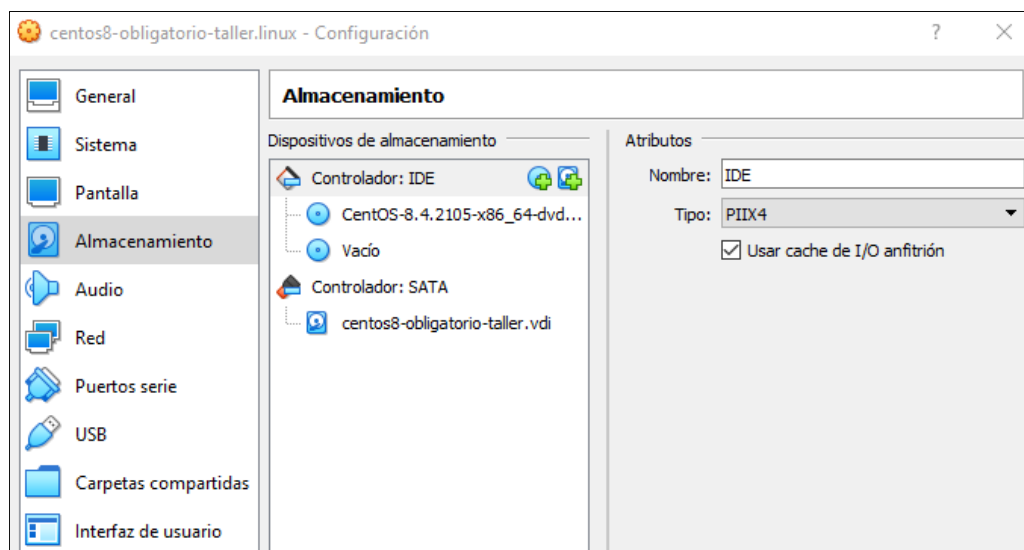
Creamos un disco duro virtual (Formato VDI) de 15 GB reservado dinámicamente:



Configuramos 2 adaptadores de red uno NAT y otro con adaptador puente:

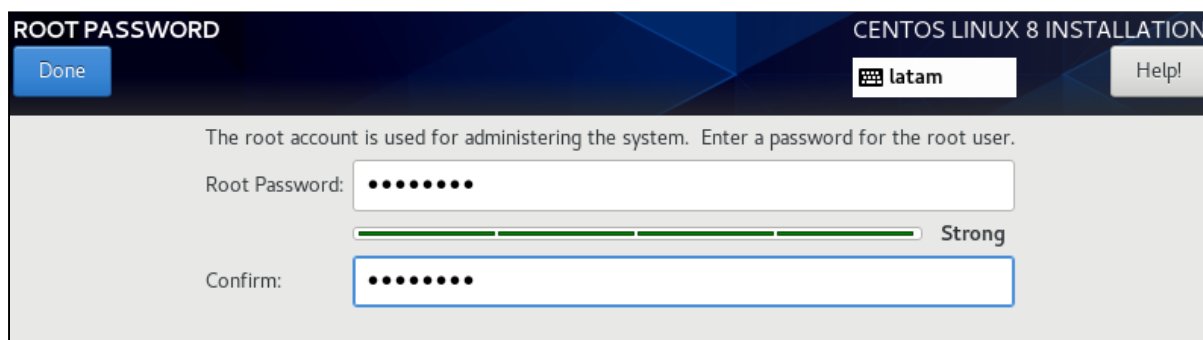


Aquí seleccionamos la .ISO mediante la cual podremos instalar el SO CentOS 8



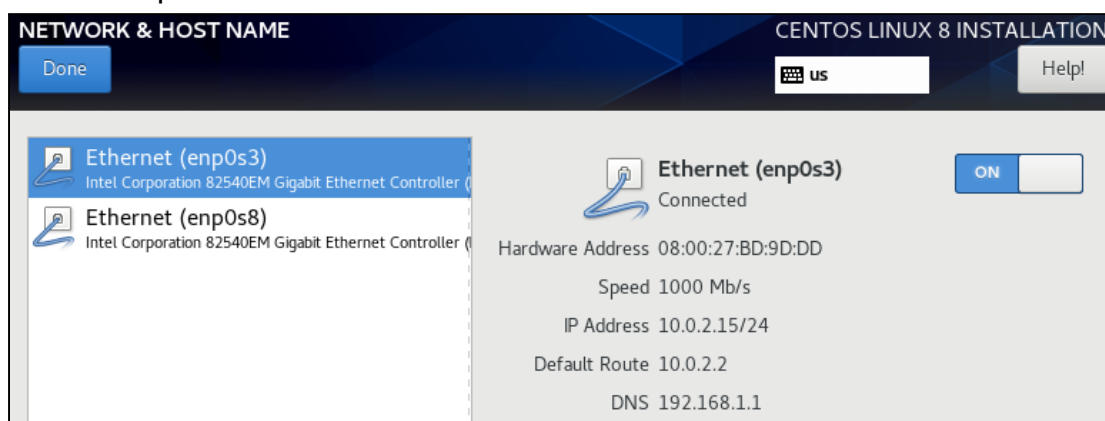
Instalación

Contraseña para usuario ROOT

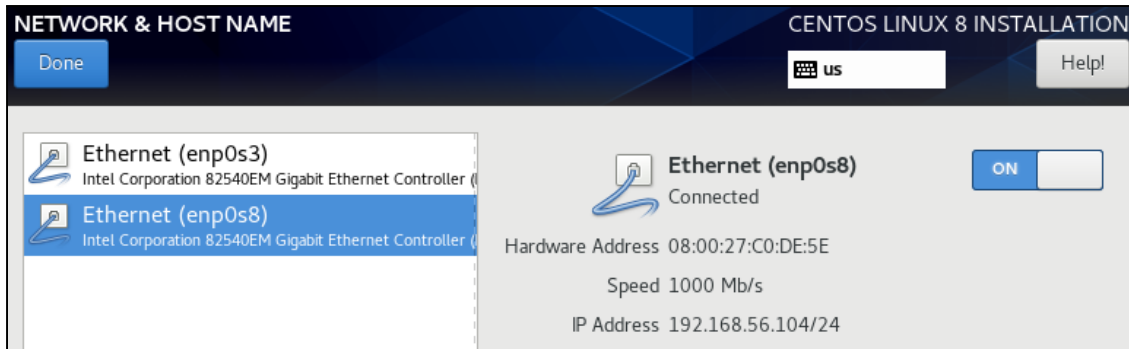


Configuramos las interfaces de red:

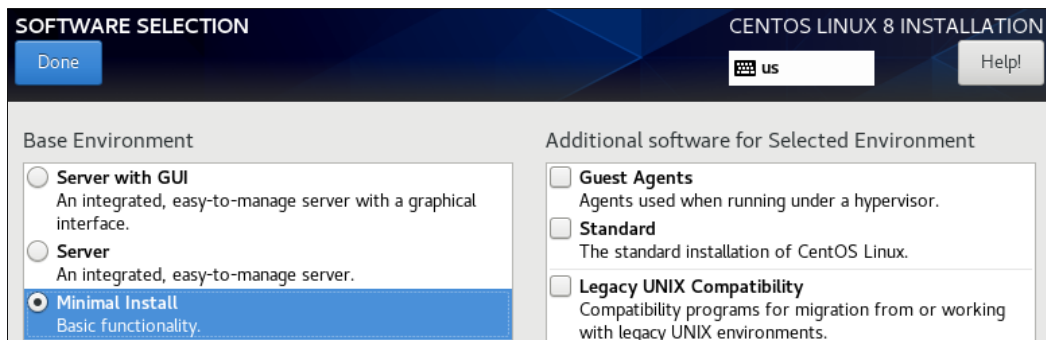
- Adaptador NAT:



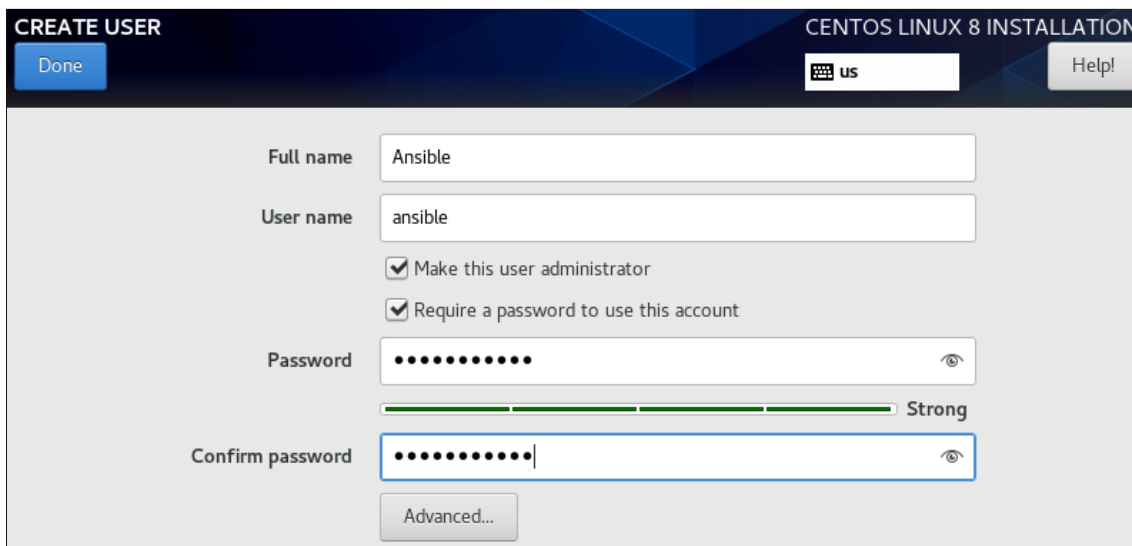
- Adaptador puente:



Selección del software a instalar, en este caso “minimal install”



Creamos el usuario “Ansible”, con privilegios de administrador:



Esquema de particionamiento para el servidor CentOS:

```
[ansible@localhost ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   16G  0 disk
├─sda1       8:1    0    1G  0 part /boot
├─sda2       8:2    0   14G  0 part
│   └─cl-root 253:0    0    5G  0 lvm  /
│       └─cl-swap 253:1    0    2G  0 lvm  [SWAP]
│           └─cl-var 253:2    0    4G  0 lvm  /var
│               └─cl-home 253:3    0    3G  0 lvm  /home
sr0         11:0    1 1024M  0 rom
```

Ubuntu

Al crear esta máquina virtual las configuraciones son prácticamente iguales a las del CentOS 8. Evidentemente se usa un .ISO de instalación para el SO Ubuntu.

Esquema de particionamiento para el servidor Ubuntu

Storage configuration [Help]

MOUNT POINT	SIZE	TYPE	DEVICE TYPE
[/	5.000G	new ext4	new LVM logical volume ▶]
[/boot	1.000G	new ext4	new partition of local disk ▶]
[/home	3.000G	new ext4	new LVM logical volume ▶]
[/var	4.000G	new ext4	new LVM logical volume ▶]
[SWAP	1.996G	new swap	new LVM logical volume ▶]

AVAILABLE DEVICES

No available devices

[Create software RAID (md) ▶]

[Create volume group (LVM) ▶]

USED DEVICES

DEVICE	TYPE	SIZE
[ubuntu-vg (new)	LVM volume group	13.996G ▶]
lv-0	new, to be formatted as ext4, mounted at /	5.000G ▶]
lv-1	new, to be formatted as ext4, mounted at /var	4.000G ▶]
lv-2	new, to be formatted as ext4, mounted at /home	3.000G ▶]
lv-3	new, to be formatted as swap	1.996G ▶]
[VBOX_HARDDISK_VB6fbe5f58-1b294211	local disk	15.000G ▶]
partition 1	new, bios_grub	1.000M ▶]
partition 2	new, to be formatted as ext4, mounted at /boot	1.000G ▶]
partition 3	new, PV of LVM volume group ubuntu-vg	13.997G ▶]

[Done]

[Reset]

[Back]

```
sda                8:0    0   16G   0 disk
├─sda1              8:1    0    1M   0 part
├─sda2              8:2    0    1G   0 part /boot
├─sda3              8:3    0   15G   0 part
│   ├─ubuntu--vg-lv--0 253:0    0    5G   0 lvm  /
│   ├─ubuntu--vg-lv--1 253:1    0    4G   0 lvm  /var
│   ├─ubuntu--vg-lv--2 253:2    0    3G   0 lvm  /home
│   └─ubuntu--vg-lv--3 253:3    0    2G   0 lvm  [SWAP]
sr0               11:0    1 1024M   0 rom
```

Clave pública para conexión SSH

Generamos una public key y la copiamos en el host remoto:

```
sansberro@DESKTOP-VSELF23:~$ ssh-copy-id ansible@10.0.2.15
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/sansberro/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/usr/bin/ssh-copy-id: ERROR: ssh: connect to host 10.0.2.15 port 22: Resource temporarily unavailable

sansberro@DESKTOP-VSELF23:~$ ssh-copy-id ansible@192.168.1.15
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/sansberro/.ssh/id_rsa.pub"
The authenticity of host '192.168.1.15 (192.168.1.15)' can't be established.
ECDSA key fingerprint is SHA256:d224b2SP+Agmk1kbr1H4zIB32plu3u3uM4CP7+Y9uF1gQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansible@192.168.1.15's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ansible@192.168.1.15'"
and check to make sure that only the key(s) you wanted were added.

sansberro@DESKTOP-VSELF23:~$ ssh ansible@192.168.1.15
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64)

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

System information as of Wed 04 Aug 2021 02:01:57 AM UTC

System load:          0.16
Usage of /home:       0.3% of 2.89GB
Memory usage:         21%
Swap usage:           0%
Processes:            111
Users logged in:      1
IPv4 address for enp0s3: 192.168.1.15
IPv6 address for enp0s3: 2800:a4:1283:3300:a00:27ff:feb2:1c5f
IPv4 address for enp0s8: 192.168.56.103

85 updates can be installed immediately.
1 of these updates is a security update.
To see these additional updates run: apt list --upgradable

Last login: Wed Aug 4 01:42:38 2021
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Configuramos el archivo Visudo para que el usuario ansible no requiera contraseña:

```
## Allows members of the users group to mount and unmount the
## cdrom as root
# %users ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users localhost=/sbin/shutdown -h now

ansible ALL=(ALL) NOPASSWD: ALL

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
```


De esta forma ahora el usuario ansible es accesible por SSH y sin contraseña, además se puede acceder a los servidores con nombres específicos configurados en `.ssh/config`.

```
1 Host taller-ubuntu
2   HostName 192.168.1.14
3   User ansible
4
5 Host taller-centos
6   Hostname 192.168.1.15
7   User ansible
```

Trabajando en el repositorio

Lo primero que notamos fue la sintaxis en la que estaba escrito la mayoría del código, la mayoría estaba escrito en una sola línea en vez de estar indentado.

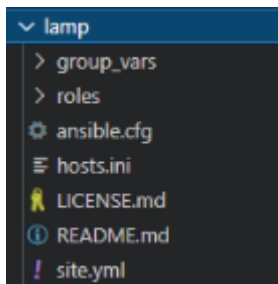
```
- name: restart ntp
  service: name=ntpd state=restarted
```

```
- name: restart ntp
  service:
    name: ntpd
    state: restarted
```

Se uso el modulo “become” para convertirse root en las llamadas del Playbook Site.yml

```
1  ---
2  # This playbook deploys the whole application stack in this site.
3
4
5  - name: apply common configuration to all nodes
6    hosts: all
7    remote_user: ansible
8    become: yes
9
10   roles:
11     - common
12
13  - name: configure and deploy the webserver and application code
14    hosts: webserver
15    remote_user: ansible
16    become: yes
17
18
19   roles:
20     - web
21
22  - name: deploy MySQL and configure the databases
23    hosts: dbserver
24    remote_user: ansible
25    become: yes
26
27   roles:
28     - db
29
```

Se agregó el archivo “ansible.cfg” y se agregó la dirección del inventario y el usuario remoto.



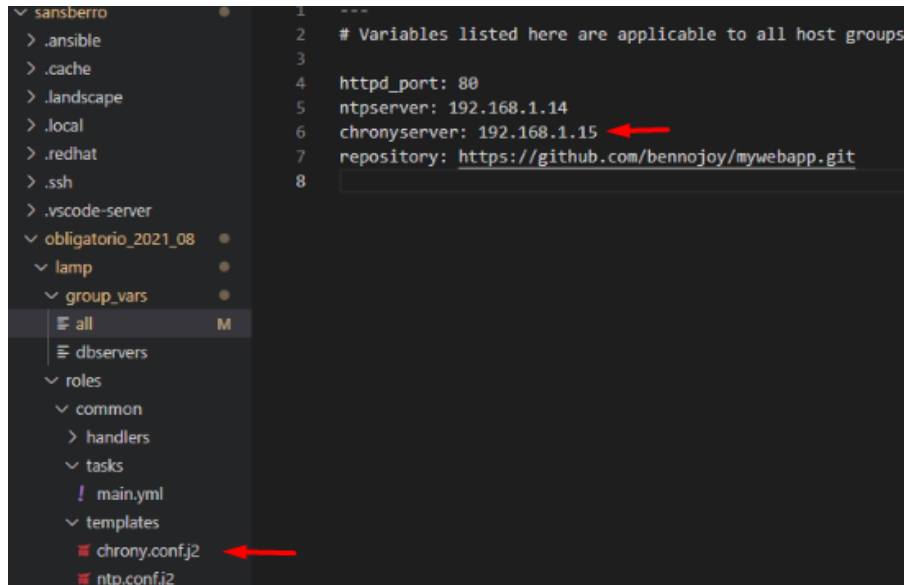
```
[defaults]
inventory      = /home/sansberro/obligatorio_2021_08/lamp/hosts.ini
#library       = ~/.ansible/plugins/modules:/usr/share/ansible/plugin
#module_utils  = ~/.ansible/plugins/module_utils:/usr/share/ansible/p
#remote_tmp    = ~/.ansible/tmp
#local_tmp     = ~/.ansible/tmp
```

```
# Default user to use for playbooks if user is not specified
# Uses the connection plugin's default, normally the user currently exe
# unless a different user is specified here.
#
remote_user = ansible
```

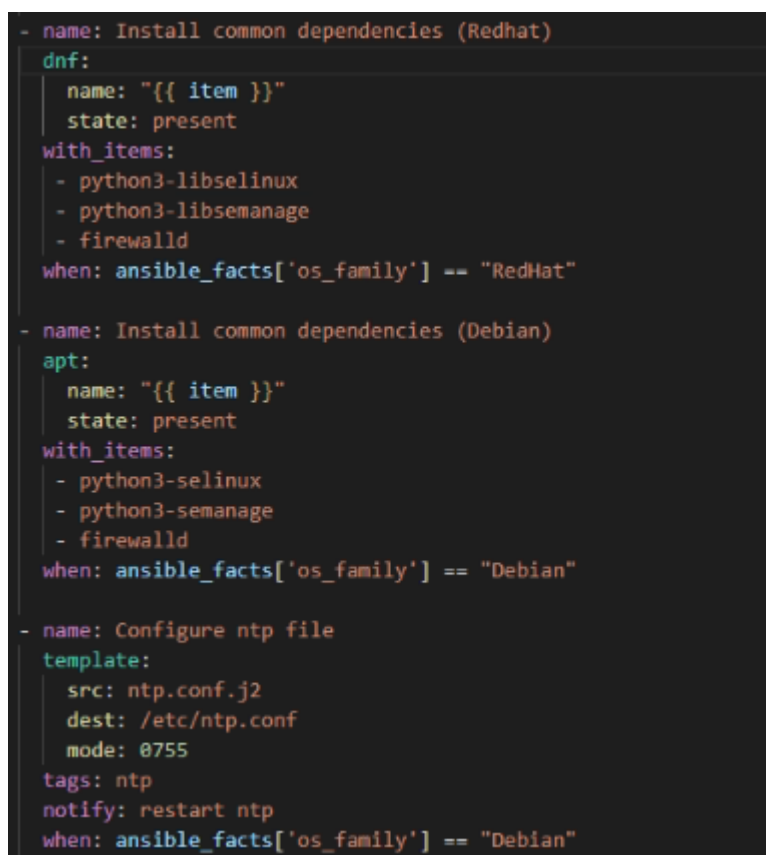
Se agregan los hosts al inventario.

```
1  [webservers]
2  192.168.1.14 #Ubuntu Server
3
4  [dbservers]
5  192.168.1.15 #CentOS 8
```

Incluimos la variable global chronyserver que contiene la IP para utilizarse en el nuevo archivo Jinja “chorny.conf.j2”



Se agregan condicionales en las tareas de instalación para que se admitan en los dos tipos de servidores.



Ejecución de la Play para el rol Common

```
PLAY [apply common configuration to all nodes] *****

TASK [Gathering Facts] *****
ok: [192.168.1.14]
ok: [192.168.1.15]

TASK [common : Install chrony (Redhat)] *****
skipping: [192.168.1.14]
ok: [192.168.1.15]

TASK [common : Install ntp (Debian)] *****
skipping: [192.168.1.15]
ok: [192.168.1.14]

TASK [common : Install common dependencies (Redhat)] *****
skipping: [192.168.1.14] => (item=[])
ok: [192.168.1.15] => (item=['python3-libselinux', 'python3-libsemanage', 'firewalld'])

TASK [common : Install common dependencies (Debian)] *****
skipping: [192.168.1.15] => (item=[])
ok: [192.168.1.14] => (item=['python3-selinux', 'python3-semanage', 'firewalld'])

TASK [common : Configure ntp file] *****
skipping: [192.168.1.15]
ok: [192.168.1.14]

TASK [common : Configure chrony file] *****
skipping: [192.168.1.14]
ok: [192.168.1.15]

TASK [common : Start the ntp service] *****
skipping: [192.168.1.15]
ok: [192.168.1.14]

TASK [common : Start the chrony service] *****
skipping: [192.168.1.14]
ok: [192.168.1.15]

PLAY [configure and deploy the webserver and application code] *****

TASK [Gathering Facts] *****
ok: [192.168.1.14]
```

En este rol tuvimos problemas con los paquetes de SELinux ya que los nombres que tenían no eran reconocibles por el servidor. Cambiamos sus nombres de acuerdo al servidor. La instalación NTP recibió un cambio orientado a los servidores RedHat ya que en estos se instala como “Chrony”. En base a esto se agregó también una tarea de iniciar el servidor Chrony al final del playbook.

Ejecución de la Play para el rol Web

```
PLAY [configure and deploy the webservers and application code] *****

TASK [Gathering Facts] *****
ok: [192.168.1.14]

TASK [web : Install httpd and php (Redhat)] *****
skipping: [192.168.1.14] => (item=[])

TASK [web : Install httpd and php (Debian)] *****
ok: [192.168.1.14] => (item=['apache2', 'php', 'php-mysql'])

TASK [web : Install web role specific dependencies (RedHat)] *****
skipping: [192.168.1.14] => (item=[])

TASK [web : Install web role specific dependencies (Debian)] *****
ok: [192.168.1.14] => (item=['git'])

TASK [web : Start firewall] *****
ok: [192.168.1.14]

TASK [web : Insert firewall rule for httpd] *****
ok: [192.168.1.14]

TASK [web : apache2 service state] *****
ok: [192.168.1.14]

TASK [web : http service state] *****
skipping: [192.168.1.14]

TASK [web : Configure SELinux to allow httpd to connect to remote database] *****
skipping: [192.168.1.14]

TASK [web : Copy the code from repository] *****
ok: [192.168.1.14]

TASK [web : Creates the index.php file] *****
ok: [192.168.1.14]
```

El paquete de httpd fue cambiado por apache2 para que lo soporte el servidor Ubuntu la tarea para configurar SELinux recibió un condicional para ejecutarse solo en servidores RedHat.

Se agregó una tarea que inicie el servicio apache2.

Ejecución de la Play para el rol DB

```
PLAY [deploy MySQL and configure the databases] *****

TASK [Gathering Facts] *****
ok: [192.168.1.15]

TASK [db : Install REPO EPEL] *****
ok: [192.168.1.15]

TASK [db : Install MariaDB package (Redhat)] *****
ok: [192.168.1.15]

TASK [db : Install MariaDB package (Debian)] *****
skipping: [192.168.1.15] => (item=[])

TASK [db : Configure SELinux to start mysql on any port] *****
ok: [192.168.1.15]

TASK [db : Create Mysql configuration file] *****
ok: [192.168.1.15]

TASK [db : Create MariaDB log file] *****
changed: [192.168.1.15]

TASK [db : Create MariaDB PID directory] *****
changed: [192.168.1.15]

TASK [db : Start MariaDB Service] *****
ok: [192.168.1.15]

TASK [db : Start firewalld] *****
ok: [192.168.1.15]

TASK [db : insert firewalld rule] *****
ok: [192.168.1.15]

TASK [db : Create Application Database] *****
ok: [192.168.1.15]

TASK [db : Create Application DB User] *****
[WARNING]: Module did not set no_log for update_password
ok: [192.168.1.15]
```

Tuvimos problemas con los paquetes de MariaDB pero pudimos descargarlo luego de implementar la instalación de EPEL.

Modificaciones en los servidores

```
TASK [common : Install ntp (Debian)] *****
skipping: [192.168.1.15]
changed: [192.168.1.14]
```

```
TASK [common : Install chrony (Redhat)] *****
skipping: [192.168.1.14]
changed: [192.168.1.15]
```

```
TASK [common : Start the chrony service] *****
changed: [192.168.1.15]
```

```
TASK [common : Configure ntp file] *****
skipping: [192.168.1.15]
changed: [192.168.1.14]
```

```
TASK [web : Install httpd and php (Debian)] *****
changed: [192.168.1.14] => (item=['apache2', 'php', 'php-mysql'])
```

```
TASK [db : Configure SELinux to start mysql on any port] *****
changed: [192.168.1.15]
```

```
TASK [db : Create Mysql configuration file] *****
changed: [192.168.1.15]
```

```
TASK [db : Create MariaDB log file] *****
changed: [192.168.1.15]
```

```
TASK [db : Create MariaDB PID directory] *****
changed: [192.168.1.15]
```

```
TASK [db : Start MariaDB Service] *****
changed: [192.168.1.15]
```

```
TASK [db : Start firewalld] *****
ok: [192.168.1.15]
```

```
TASK [db : insert firewalld rule] *****
changed: [192.168.1.15]
```

```
TASK [db : Create Application Database] *****
changed: [192.168.1.15]
```