



TALLER DE SERVIDORES LINUX

OBLIGATORIO -ENTREGA FINAL

INTEGRANTES: Victoria Gallardo (207154), Ignacio Tabárez (173307) y
Luciano Villa (240976)

PROFESOR: Enrique Verdes

10 de agosto de 2021

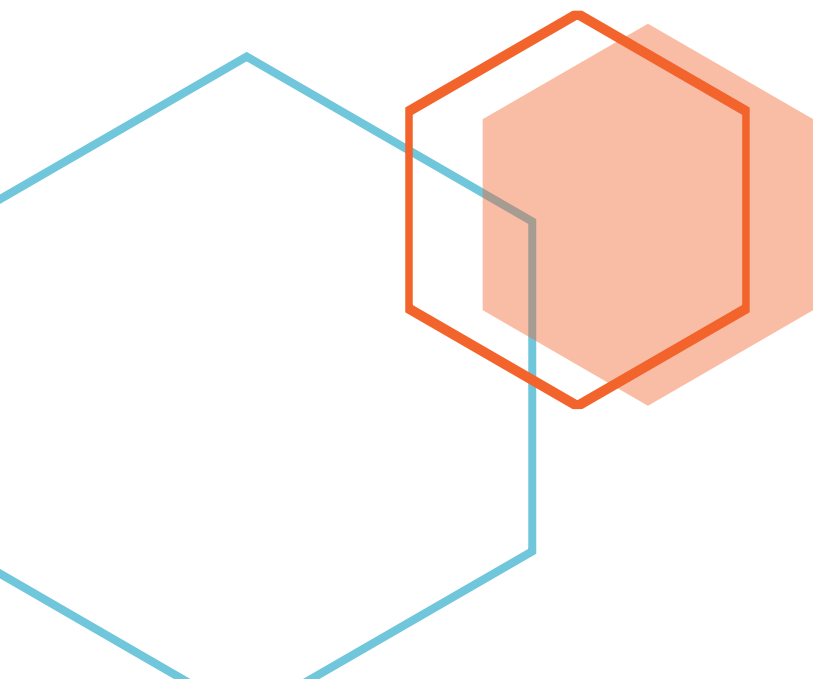


Tabla de contenido

| | |
|---|----|
| 1. DECLARACIÓN AUTORÍA | 2 |
| 2. RESUMEN EJECUTIVO | 2 |
| 2.1. Introducción | 2 |
| 2.2. Objetivos Específicos | 2 |
| 3. SOLUCIÓN PROYECTO | 2 |
| 3.1. Instalación CentOS y Ubuntu | 2 |
| 3.2. Instalar ansible | 4 |
| 3.3. Configurar Claves | 5 |
| 3.4. Instalar Git | 5 |
| 3.5. Correcciones de errores archivos | 8 |
| 3.6. Ejecución ansible | 10 |
| 3.7. Roles Ansible | 12 |
| 4. REFERENCIAS | 13 |

SOLUCIÓN PROYECTO TALLER DE SERVIDORES LINUX

1. DECLARACIÓN AUTORÍA

Victoria Gallardo, Ignacio Tabárez y Luciano Villa, con número de estudiante 258562 - 173307 - 240976, y estudiantes de la carrera Analista en Infraestructura Informática de la Escuela de Tecnologías de la Universidad ORT, en relación con el trabajo final presentado para su defensa y evaluación de la materia "Taller de Servidores Linux", declaráramos que asumimos la originalidad de dicho trabajo, entendida en el sentido de que no se ha utilizado fuentes sin citarlas debidamente.

Montevideo, 10 de agosto de 2021

Firma: Victoria Gallardo, Ignacio Tabárez y Luciano Villa

2. RESUMEN EJECUTIVO

2.1. Introducción

El presente documento tiene como objetivo presentar una solución para el proyecto de Taller de Servidores Linux en el marco de un trabajo obligatorio final. Este proyecto tiene como fin principal brindar una solución que automatice procesos de TI y/o infraestructura.

2.2. Objetivos Específicos

El objetivo del taller es poder crear procesos que automaticen las tareas que pueden ser repetitivas. Alcanzar la mayoría de las tareas automatizadas nos permite administrar mejor una infraestructura y utilizar u optimizar de buena manera el tiempo de trabajo.

3. SOLUCIÓN PROYECTO

A continuación, se brindan los detalles de las modificaciones y configuración de las diferentes tareas realizadas.

3.1. Instalación CentOS y Ubuntu

En dicha solución se cuentan con 3 máquinas virtuales, 2 Ubuntu 20.04 y 1 CentOS 8.

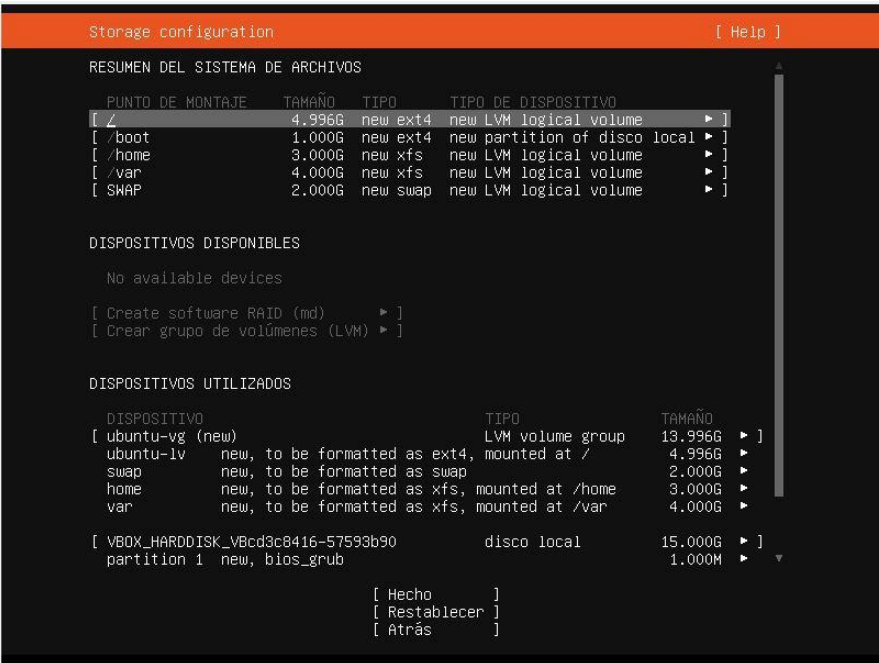
Características Ubuntu:

- 1 GB RAM
- 15 GB Disco Duro

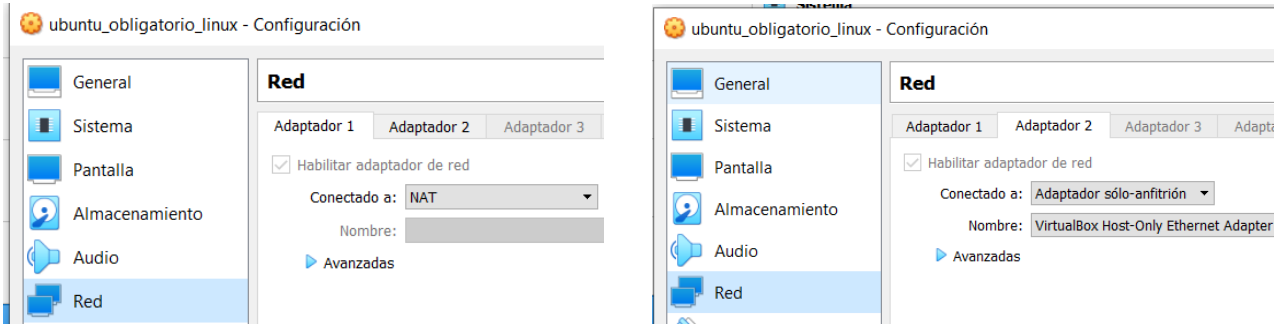
Particiones:

- 1 GB /boot
- 5 GB /
- 4 GB /var
- 3 GB /home
- 2 GB /swap

En la siguiente imagen se puede apreciar como quedaron las particiones realizadas en el equipo Ubuntu.



Por otra parte, como se solicita en le letra del proyecto, se configura la misma con dos interfaces de red (enp0s3 y enp0s8). La interfaz enp0s3 conectada a NAT y enp0s8 a una red interna que permite conectarse a las otras máquinas.

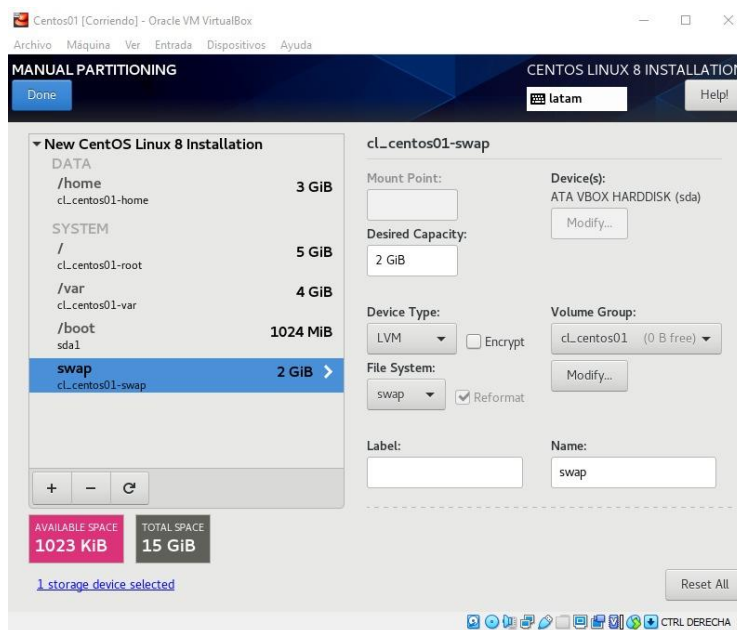


En la imagen a continuación se puede apreciar las IP asignada a cada una de las interfaces.

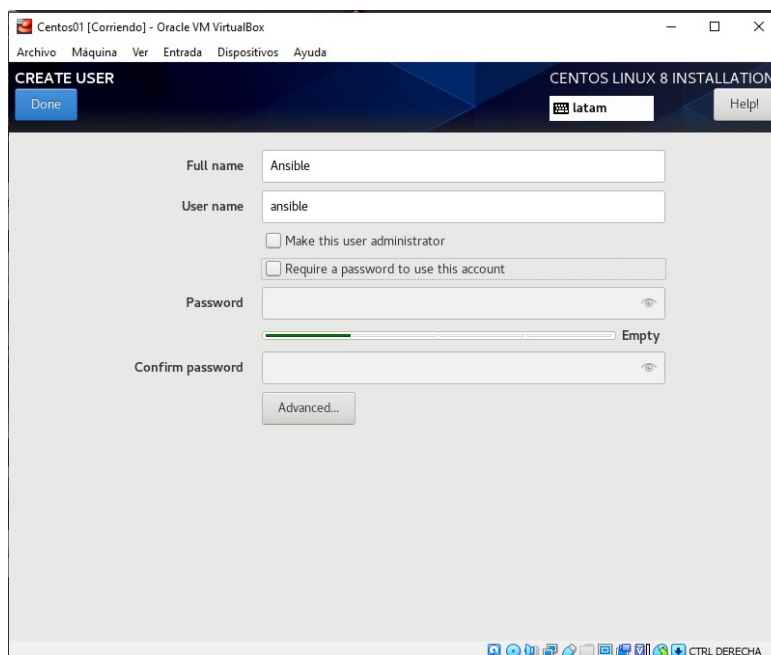
```

ansible@ansible:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b0:ec:f2 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 84520sec preferred_lft 84520sec
    inet6 fe80::a00:27ff:feb0:ecf2/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:29:85:48 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic enp0s8
        valid_lft 427sec preferred_lft 427sec
    inet6 fe80::a00:27ff:fe29:8548/64 scope link
        valid_lft forever preferred_lft forever
ansible@ansible:~$ _
    
```

Particiones CentOS8



Por otra parte, se crea un usuario "ansible" sin password y con permisos de SUDO.



3.2. Instalar ansible

Para comenzar a utilizar ansible como medio de administración, debemos instalarlo en el equipo de bastión. Para ellos ejecutar los comandos:

- apt update
- apt install ansible

3.3. Configurar Claves

Para configurar las claves públicas en el bastión, antes debemos generar el par de claves en cada una de las máquinas host. Se debe utilizar las siguientes instrucciones:

- ssh-keygen

El par de claves, la generará en la ruta:

- /home/user/.ssh/id_rsa

User: carpeta del usuario

Desde el equipo bastión, procedemos a configurar las claves públicas del resto de los equipos hosts (CentOS y Ubuntu). Para ellos utilizamos los siguientes comandos:

- ssh-copy-id remote_username@server_ip_address

Username = usuario host

server_ip_address= IP máquina host

Desde el equipo bastión se deberá ingresar la password del host, y si es correcta, como resultado se agregará la clave al equipo. Una forma de verificar esto es de la siguiente manera: ssh user@IP_HOST

```
ansible@ansible:~$ ssh ansible@192.168.56.105
Enter passphrase for key '/home/ansible/.ssh/id_rsa':
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 Tue Aug 10 18:09:47 UTC 2021

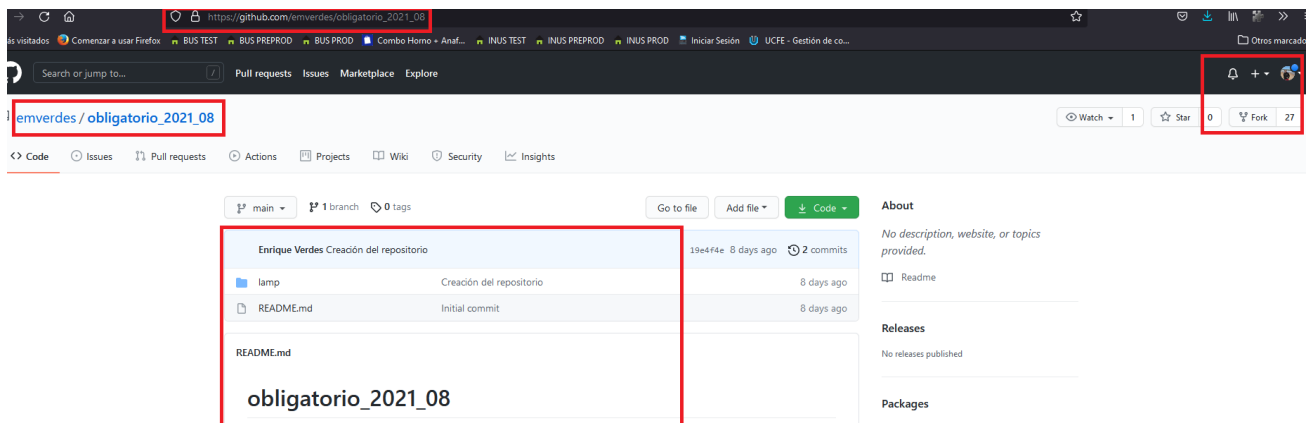
System load:  0.0               Processes:    119
Usage of /:   31.6% of 13.71GB   Users logged in: 1
Memory usage: 22%              IPv4 address for enp0s3: 10.0.2.15
Swap usage:  0%                IPv4 address for enp0s8: 192.168.56.105

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

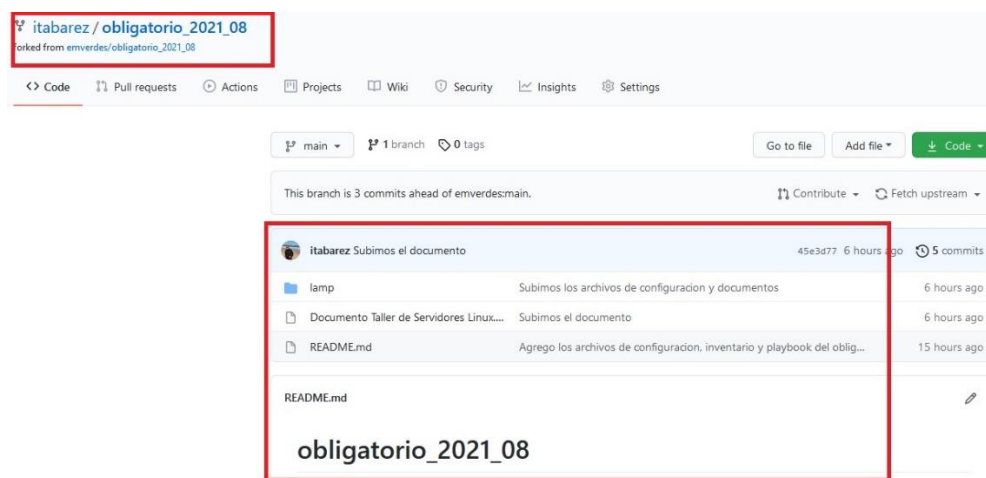
Last login: Tue Aug 10 14:04:18 2021 from 192.168.56.102
ansible@ansible-host:~$
```

3.4. Instalar Git

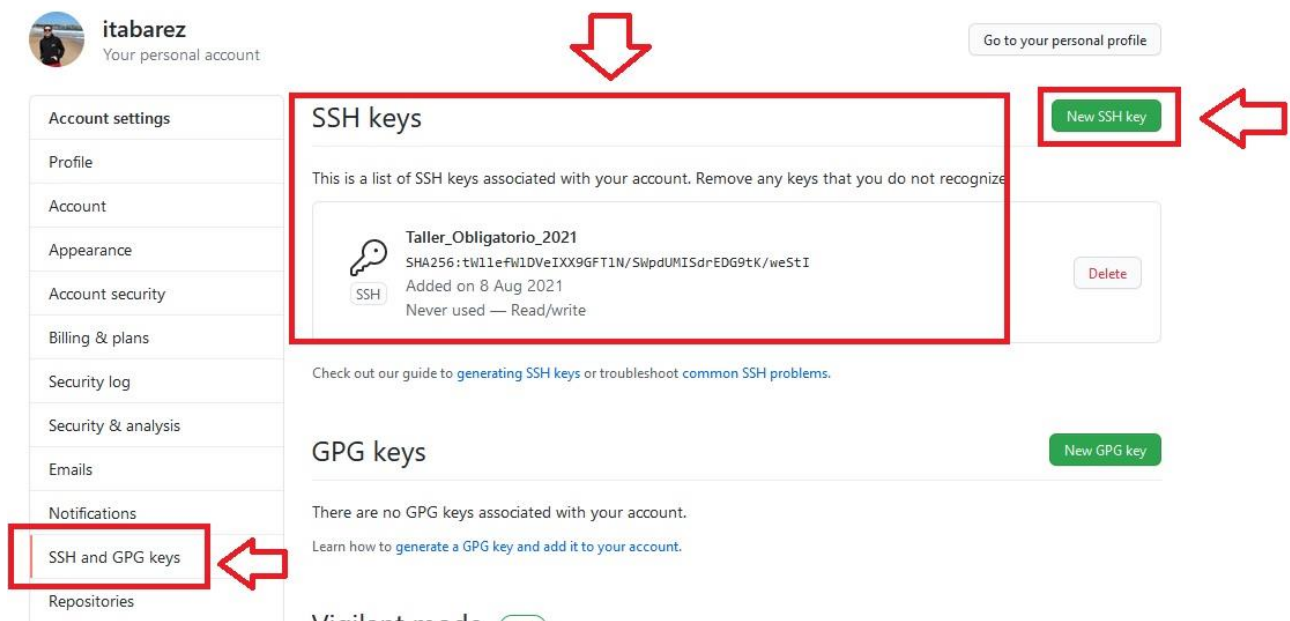
Como solicita el proyecto, debemos utilizar los archivos que se encuentra en un repositorio de Github (https://github.com/emverdes/obligatorio_2021_08). Primero, debemos crear o ingresar con nuestro usuario al repositorio de referencia. Una vez en el mismo, hacer un "fork".



En los repositorios de nuestro perfil debemos visualizarlo.



Lo siguiente que se debe realizar es generar la clave SSH en el equipo bastión de la misma forma que se hicieron en los hosts. Una vez que tengamos esta clave, debemos copiarla e importarla a nuestro perfil de Github.





Posteriormente, en nuestro equipo bastión debemos instalar Git con el siguiente comando:

- `apt install git`

Una vez instalado, procedemos a configurar y clonar el repositorio de nuestro perfil para poder trabajar de forma local y luego subir los cambios aplicados.

En nuestro flujo de trabajo, realizaremos lo siguiente:

- Clonar el repositorio
- Realizar "pull" del repositorio para siempre tener el mismo actualizado.
- Realizar "git add ." en caso que tengamos archivos nuevos para subir.
- Realizar "commit" con un mensaje descriptivo de lo que modificamos.
- Si todo está bien, hacer "push" para subir los cambios al repositorio remoto.

A continuación, configuramos nuestro ambiente Git.

Nombre de usuario

- `$ git config --global user.name "itabarez"`

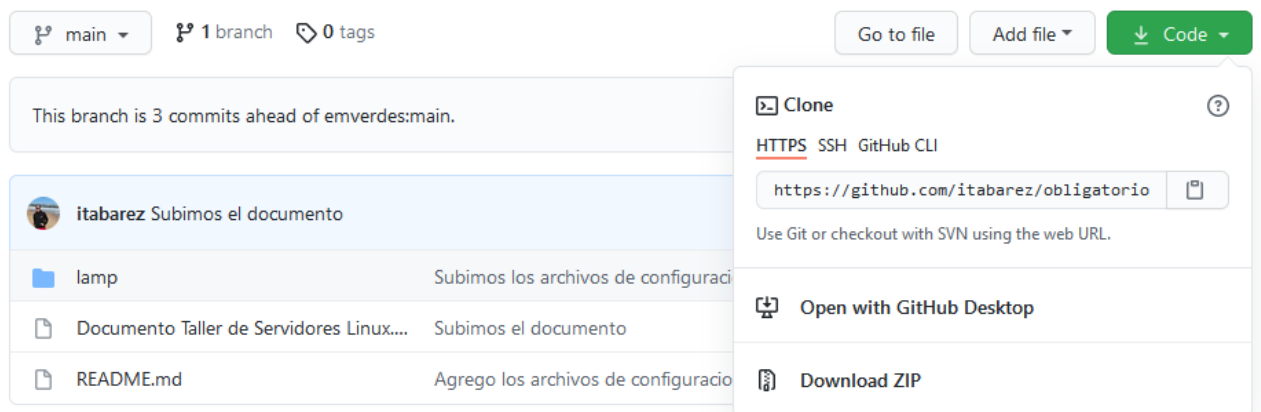
Dirección de correo electrónico

- `$ git config --global user.email nacho.tabarez13@gmail.com`

Clonar repositorio

- `$ git clone "https://github.com/itabarez/obligatorio_2021_08.git"`

La URL mencionada anteriormente, se obtiene desde el repositorio



Ejecutado el comando, nos dejará en `/home/[usuario]` los archivos del repositorio.

```
ansible@ansible:~$ cd /home/ansible/obligatorio_2021_08
ansible@ansible:~/obligatorio_2021_08$ ls
'Documento Taller de Servidores Linux.docx'  README.md  lamp
ansible@ansible:~/obligatorio_2021_08$ _
```

A partir de ese momento, se puede comenzar a trabajar sobre los archivos.

3.5. Correcciones de errores archivos

A continuación, se detallan todas las correcciones realizadas a los archivos del repositorio para que se ejecute correctamente en el equipo bastión, y ahí a los hosts que tiene configurado en un inventario.

En el inventario se agregaron las siguientes variables:

```
[webservers]
centos ansible_host=192.168.56.104

[dbservers]
ubuntu ansible_host=192.168.56.105

[centos]
centos ansible_host=192.168.56.104

[ubuntu]
ubuntu ansible_host=192.168.56.105

[linux:children]
centos
ubuntu
```

Archivo site.yml de la carpeta /lamp/

- se modificó remote_user con el valor "ansible"
- se agregó la propiedad become en cada uno de los roles

Archivo hosts de la carpeta /lamp/

- se agregaron los valores de las variables [webserver], [dbservers]
- se agregaron las variables [centos], [ubuntu] y [linux:children]
- se configuraron las ip para cada uno de los hosts.

Archivo main.yml de la carpeta roles/common/handlers/

- se dieron salto de línea para name y state
- se sustituyeron los "=" por ":"

Archivo chrony.conf.j2 de la carpeta /roles/common/templates/

- se modificó el server definido anteriormente por: server {{ ntpserver }}

Archivo main.yml de la carpeta roles/common/task/

- se dieron salto de línea para name y state
- se sustituyeron los "=" por ":"
- se pusieron las variables (item) entre comillas "".
- salto de línea en "Configure ntp file" para el atributo "src"
- salto de línea en "dest" y se sustituyó "=" por ":"
- salto de línea en "tags".
- salto de línea en enables y se sustituyó "=" por ":".
- Se agregaron las condiciones en cada sentencia para ejecutar cada "proceso" según el sistema operativo como solicita la letra del proyecto
- Para Debian, el módulo a instalar se llama "ntp".



- Para Redhat el módulo a instalar se llama "chrony".
- Para Redhat, se copia el archivo chrony.conf para la carpeta /etc/ .
- Para Redhat, se copia el archivo chrony.conf.j2 para la carpeta /roles/common/templates/ .

Archivo main.yml de la carpeta roles/db/handlers/

- se dieron salto de línea para name y state
- se sustituyeron los "=" por ":"

Archivo main.yml de la carpeta roles/db/task/

- se dieron salto de línea para name y state
- se sustituyeron los "=" por ":"
- se pusieron las variables (item) entre comillas "".
- salto de línea en "Configure SELinux to start mysql on any port" para el atributo "name" y "state"
- salto de línea en "dest" y se sustituyó "=" por ":"
- salto de línea en "tags".
- salto de línea en enables y se sustituyó "=" por ":".
- salto de línea en "persistent" y se sustituyó "=" por ":".
- salto de línea en "Create Mysql configuration file" para el atributo "src"
- salto de línea en "dest" y se sustituyó "=" por ":"
- salto de línea en "path" y se sustituyó "=" por ":"
- salto de línea en "state" y se sustituyó "=" por ":"
- salto de línea en "owner" y se sustituyó "=" por ":"
- salto de línea en "group" y se sustituyó "=" por ":"
- salto de línea en "mode" y se sustituyó "=" por ":"
- En "Create MariaDB PID Directory" salto de línea en "path" y se sustituyó "=" por ":"
- En "Create MariaDB PID Directory" salto de línea en "state" y se sustituyó "=" por ":"
- En "Create MariaDB PID Directory" salto de línea en "owner" y se sustituyó "=" por ":"
- En "Create MariaDB PID Directory" salto de línea en "group" y se sustituyó "=" por ":"
- En "Create MariaDB PID Directory" salto de línea en "mode" y se sustituyó "=" por ":"
- Se agregaron las condiciones en cada sentencia para ejecutar cada "proceso" según el sistema operativo como solicita la letra del proyecto.
- Para "Debian" el paquete se llama "python-mysqldb".
- Se quitaron las sentencias de configuración de "SELinux" para "Debian".

Archivo copy_code.yml de la carpeta roles/web/tasks/

- se dieron salto de línea para "repo" y "dest"
- se sustituyeron los "=" por ":"
- se agrega comillas (") a la variable "repository"
- se dieron salto de línea para "src" y "dest"
- se sustituyeron los "=" por ":"

Archivo install_httpd.yml de la carpeta roles/web/tasks/

- se dieron salto de línea para "name", "state", "persistent", "permanent", "immediate" y "enable"
- se sustituyeron los "=" por ":"
- se agrega comillas (") a la variable "item"
- se agrega comillas (") a la variable "httpd_port"



- Se agregaron las condiciones en cada sentencia para ejecutar cada "proceso" según el sistema operativo como solicita la letra del proyecto
- Se modificó el módulo de instalación de apache para "Debian", debe ser "apache2"
- Se modificó el módulo de instalación de "firewallld" por "ufw" en el sistema operativo "Debian"
- Se modificó las propiedades en las instrucciones de "ufw". Se agregaron "to_port", "state".
- Se quitaron las sentencias de configuración de "SELinux" para "Debian"
- Se modificaron las instrucciones de "service state", dado que para "Debian" debe ser "apache2".

3.6. Ejecución ansible

Para verificar las correcciones de los archivos mencionados anteriormente, procedemos a realizar la ejecución de ansible para el playbook, esperando resultados exitosos. Para ello utilizamos los siguientes comandos en el equipo bastión (dentro del directorio /home/ansible/obligatorio_2021_08/lamp):

- Eval \$(ssh-agent)
- Ssh-add

```
ansible@ansible:~/obligatorio_2021_08$ eval $(ssh-agent)
Agent pid 8554
ansible@ansible:~/obligatorio_2021_08$ ssh-add
Enter passphrase for /home/ansible/.ssh/id_rsa:
Identity added: /home/ansible/.ssh/id_rsa (ansible@ansible)
ansible@ansible:~/obligatorio_2021_08$
```

Posteriormente, ejecutamos el ansible de la siguiente manera:

- Ansible-playbook -i hosts site.yml --ask-become-pass

```
ansible@ansible:~/obligatorio_2021_08/lamp$ ssh-add
Enter passphrase for /home/ansible/.ssh/id_rsa:
Identity added: /home/ansible/.ssh/id_rsa (ansible@ansible)
ansible@ansible:~/obligatorio_2021_08/lamp$ ansible-playbook -i hosts site.yml --ask-become-pass
BECOME password:
[WARNING]: Found both group and host with same name: ubuntu
[WARNING]: Found both group and host with same name: centos

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

TASK [Gathering Facts] *****
ok: [centos]
ok: [ubuntu]
```

Evidencias de la ejecución satisfactoria.

```
TASK [common : Configure ntp file] *****
ok: [ubuntu]

TASK [common : Configure chronyd file] *****
skipping: [ubuntu]

TASK [common : Start the ntp service] *****
ok: [ubuntu]

TASK [common : Start the ntp service] *****
skipping: [ubuntu]

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

TASK [Gathering Facts] *****
-

```

```
ok: [ubuntu]

TASK [web : Install httpd and php] *****
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is
deprecated. Instead of using a loop to supply multiple items and specifying `name: "{{ item }}"`,
please use `name: ['httpd', 'php', 'php-mysql']` and remove the loop. This feature will be removed
in version 2.11. Deprecation warnings can be disabled by setting deprecation_warnings=False in
ansible.cfg.
skipping: [ubuntu] => (item=[])

TASK [web : Install web role specific dependencies] *****
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is
deprecated. Instead of using a loop to supply multiple items and specifying `name: "{{ item }}"`,
please use `name: ['git']` and remove the loop. This feature will be removed in version 2.11.
Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
skipping: [ubuntu] => (item=[])

TASK [web : Install httpd and php] *****
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is
deprecated. Instead of using a loop to supply multiple items and specifying `name: "{{ item }}"`,
please use `name: ['apache2', 'php', 'php-mysql']` and remove the loop. This feature will be
removed in version 2.11. Deprecation warnings can be disabled by setting deprecation_warnings=False
in ansible.cfg.
ok: [ubuntu] => (item=['apache2', 'php', 'php-mysql'])

TASK [web : Install web role specific dependencies] *****
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is
deprecated. Instead of using a loop to supply multiple items and specifying `name: "{{ item }}"`,
please use `name: ['git']` and remove the loop. This feature will be removed in version 2.11.
Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [ubuntu] => (item=['git'])

TASK [web : Start firewalld] *****
skipping: [ubuntu]

TASK [web : Start firewalld] *****

```



```
TASK [web : Start firewallld] *****
skipping: [ubuntu]

TASK [web : Start firewallld] *****
ok: [ubuntu]

TASK [web : insert firewallld rule for httpd] *****
skipping: [ubuntu]

TASK [web : insert firewallld rule for ufw] *****
ok: [ubuntu]

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

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

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

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

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

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

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

TASK [db : Install MariaDB package] *****
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is
deprecated. Instead of using a loop to supply multiple items and specifying `name: "{{ item }}"`,
please use `name: ['mariadb-server', 'MySQL-python']` and remove the loop. This feature will be
removed in version 2.11. Deprecation warnings can be disabled by setting deprecation_warnings=False
in ansible.cfg.
skipping: [ubuntu] => (item=[])

TASK [db : Install MariaDB package] *****
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is
deprecated. Instead of using a loop to supply multiple items and specifying `name: "{{ item }}"`,
please use `name: ['mariadb-server', 'python-mysqldb']` and remove the loop. This feature will be
removed in version 2.11. Deprecation warnings can be disabled by setting deprecation_warnings=False
in ansible.cfg.
```

```
PLAY RECAP *****
centos      : ok=2    changed=0    unreachable=0    failed=1    skipped=1    rescued=
0          ignored=0

ubuntu     : ok=13   changed=0    unreachable=0    failed=1    skipped=10   rescued=
0          ignored=0

ansible@ansible:~/obligatorio_2021_08/lamp$
```

3.7. Roles Ansible

Los roles permiten cargar automáticamente archivos, controladores, ejecutar tareas, y demás acciones.

A medida que vamos añadiendo funcionalidad y complejidad a nuestros playbooks, cada vez se hace más difícil manejarlo. Los roles, nos permiten crear un playbook con una mínima configuración y definir la complejidad y lógica de las acciones a más bajo nivel.

Según la propia documentación de Ansible: "Los roles en Ansible se basan en la idea de incluir archivos y combinarlos para formar abstracciones limpias y reutilizables: le permiten



concentrarse más en el panorama general y solo sumergirse en los detalles cuando sea necesario."

Para la correcta utilización de los roles, es necesario crear toda una estructura de carpetas y subcarpetas donde iremos depositando nuestra configuración.

4. REFERENCIAS

- Material y video de Aulas – <https://aulas.ort.edu.uy>
- Documentación Ansible - https://docs.ansible.com/ansible/latest/user_guide/index.html