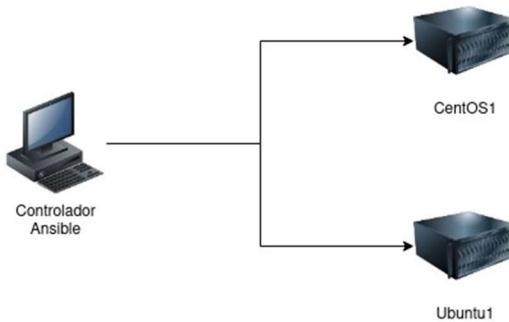


Para la realización de este obligatorio se utilizará la herramienta de VirtualBox junto con CentOS 9 y Ubuntu 24.04 brindado por el profesor de la materia durante el taller.

Esquema de Laboratorio



En cuanto al **Controller** se utilizará el que preparamos durante las clases del Taller, con CentOS:

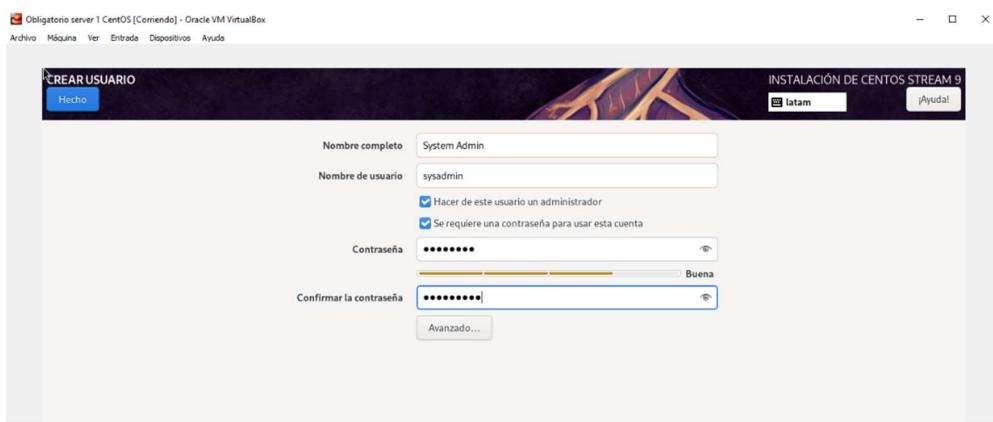
- 2 CPU
- 2 Gb RAM
- 2 tarjetas de Red: NAT y Host-Only
- 16 Gb Almacenamiento: /=5Gb, /boot=1Gb, Swap=4Gb, /home=3Gb, /var=3Gb
- Se agregan cantidad de puertos de almacenamiento para no tener que apagar el equipo virtual en caso de realizar cambios en el almacenamiento.
- Minimal install y se instaló entorno grafico por comando
- Hostname = controller.ejemplo.com.uy
- Se coloca pass de root para que esté habilitado
- Se crea usuario sysadmin – tlxadmin



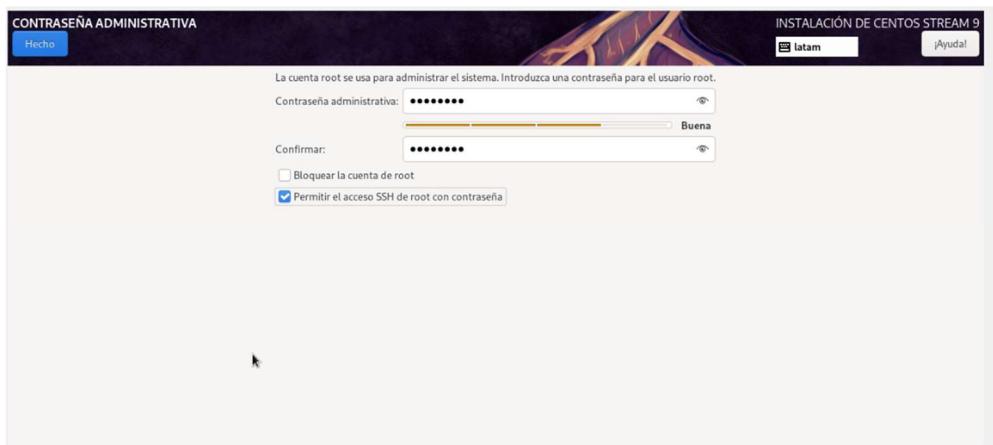
Servidor 01 CentOS 9:

- 1 CPU
- 2 Gb RAM
- 2 tarjetas de Red: NAT y Host-Only
- 13 Gb Almacenamiento: /boot=1Gb, /=7Gb, /var=3Gb, Swap=2Gb
- Minimal install
- Hostname = servidor01.ejemplo.com.uy (en opción Network)
- Se crea usuario sysadmin – tlxadmin

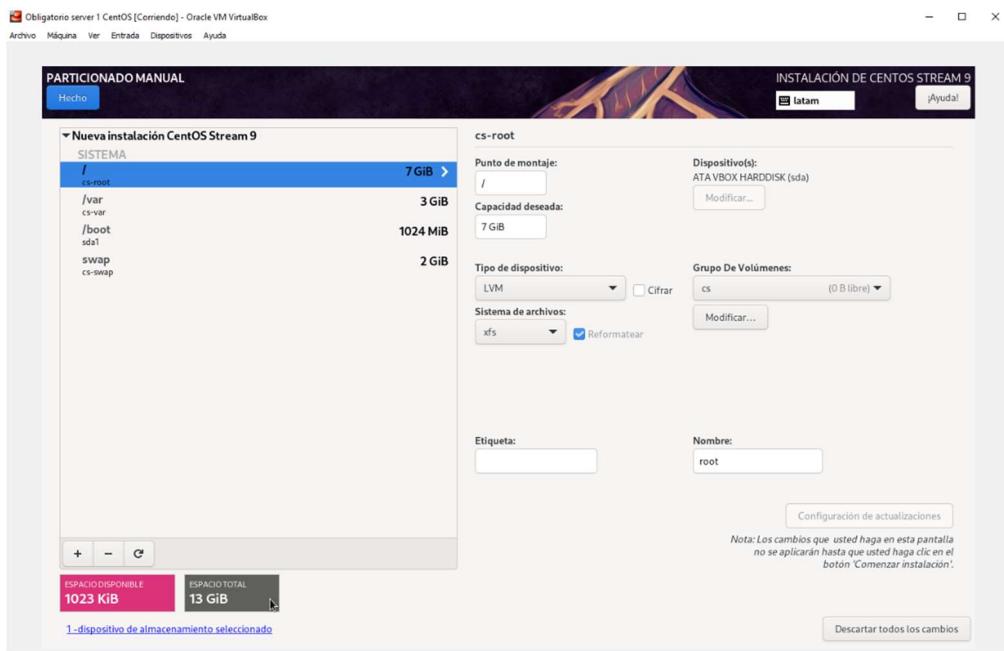
Se crea usuario sysadmin como administrador:



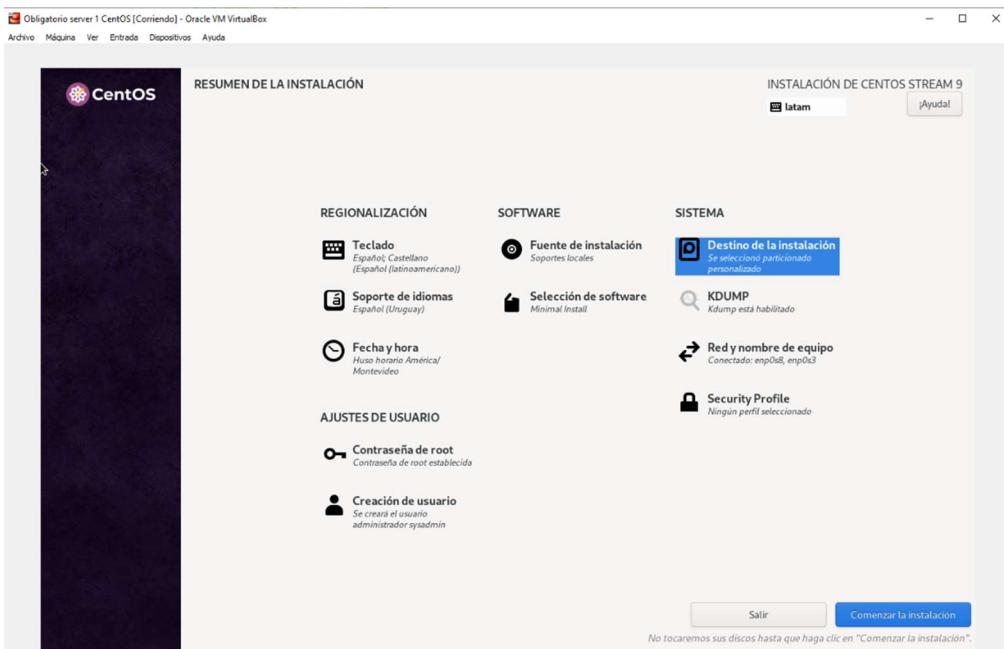
Se habilita ssh y se coloca usuario root también:



Particiones Server01 CentOS con LVM (logical volume group):



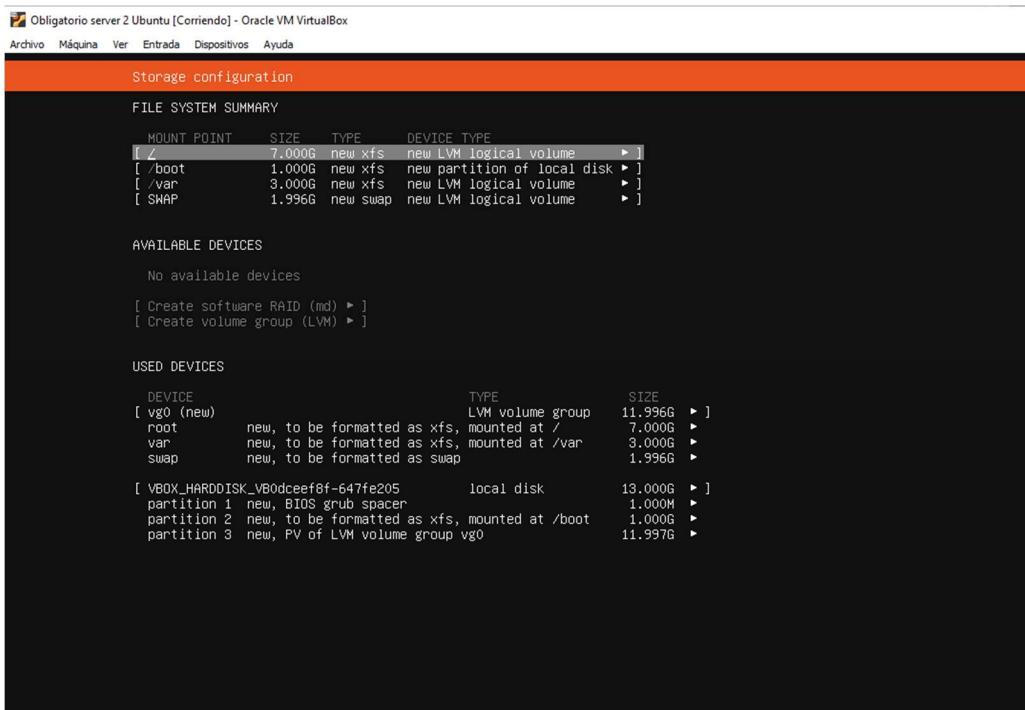
Resumen instalación CentOS 9:



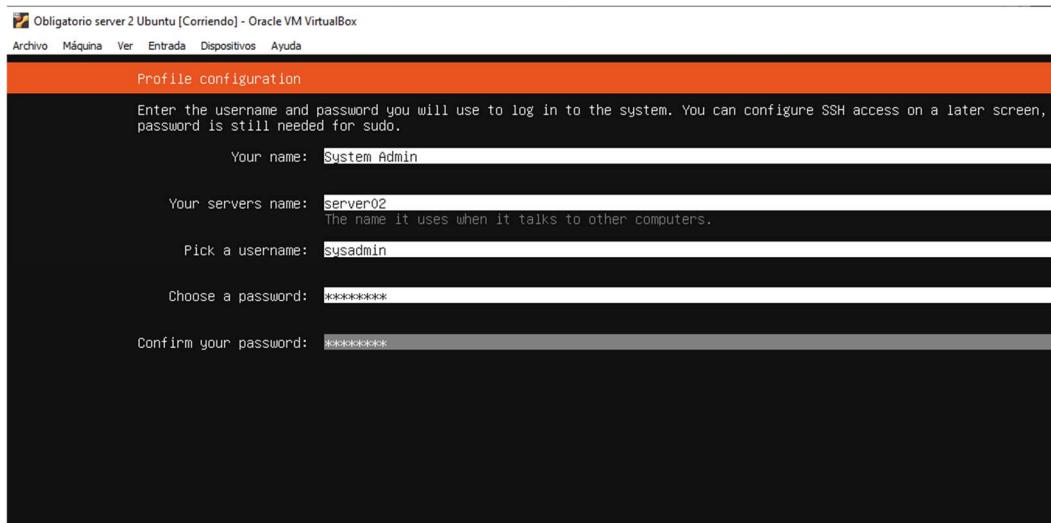
Servidor02 Ubuntu 24.04:

- 1 CPU
- 2 Gb RAM
- 2 tarjetas de Red: NAT y Host-Only
- 13 Gb Almacenamiento: /boot=1Gb, /=7Gb, /var=3Gb, Swap=2Gb en este caso se crea primero /boot formato: xfs, mount:/boot, luego se crea la partición “3” con la totalidad de espacio libre para poder crear el VG (Volume group) y así poder crear LVM (volúmenes lógicos) dentro del VG con / , /var y swap.
- Ubuntu Server
- Hostname = servidor02.ejemplo.com.uy
- Se crea usuario sysadmin – tlxadmin

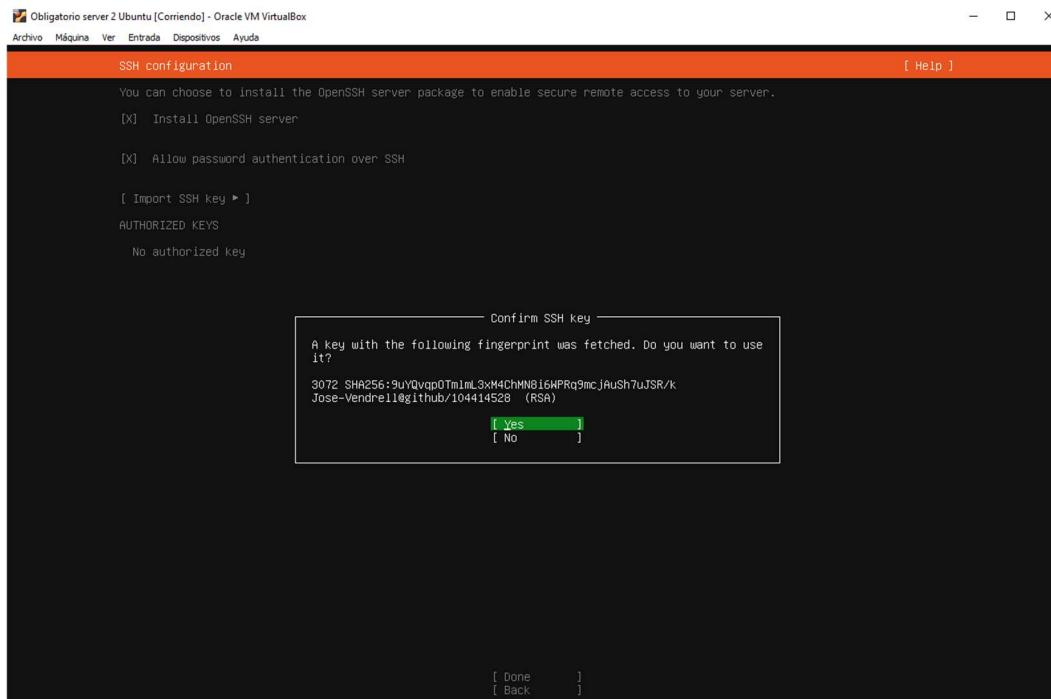
VG y LVM:



Creación usuario sysadmin:



Se instala SSH y se coloca la llave publica de github (Se muestra más adelante como se realiza el mismo desde el controller y desde la web para poder importarlo con el usuario de github <https://github.com/Jose-Vendrell/Tallerjulio2024.git>)



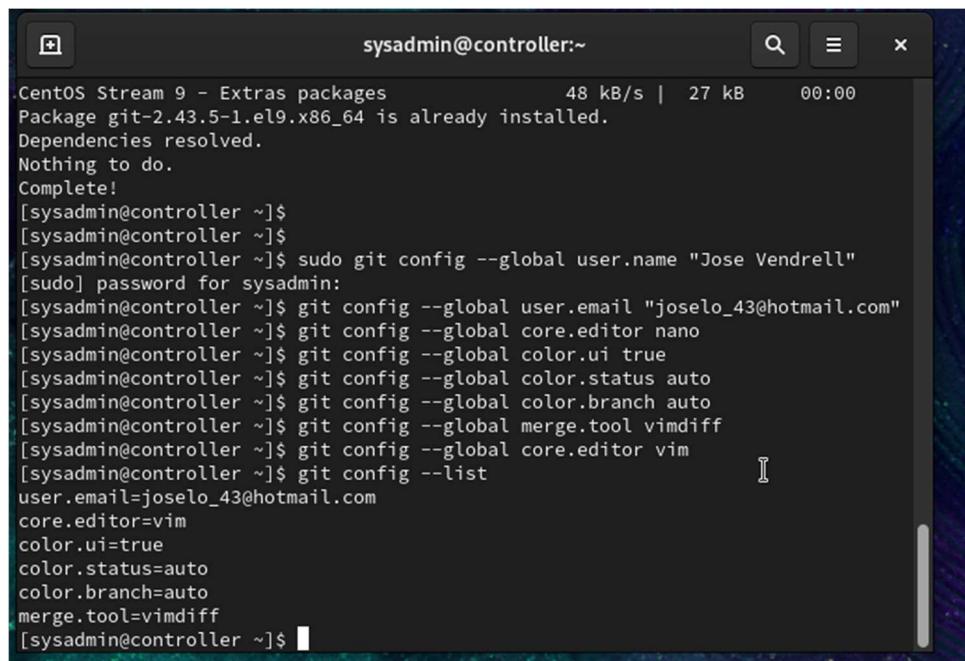
GIT:

En el **controller** realizamos la instalación del git (se utiliza solo sudo en caso de ser necesario, al estar como administrador no sería necesario) el archivo de configuración del git se encuentra en /.gitconfig el cual se puede editar con vi o nano.

- dnf install git
- git config --global user.name "Jose Vendrell"
- git config --global user.email joselo_43@hotmail.com

Cambio de colores para facilitar visualmente:

- git config --global color.ui true
- git config --global color.status auto
- git config --global color.branch auto
- git config --global core.editor vim
- git config --global merge.tool vimdiff
- git config --list (Muestra los cambios que se realizaron)

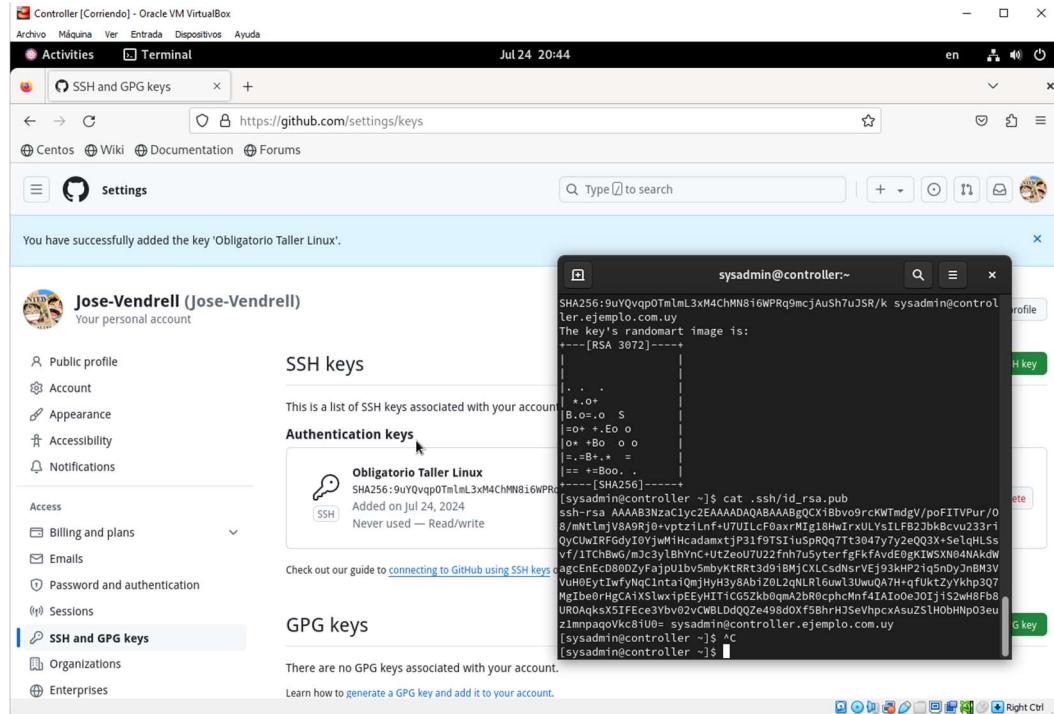


The screenshot shows a terminal window titled "sysadmin@controller:~". The terminal displays the following command history and output:

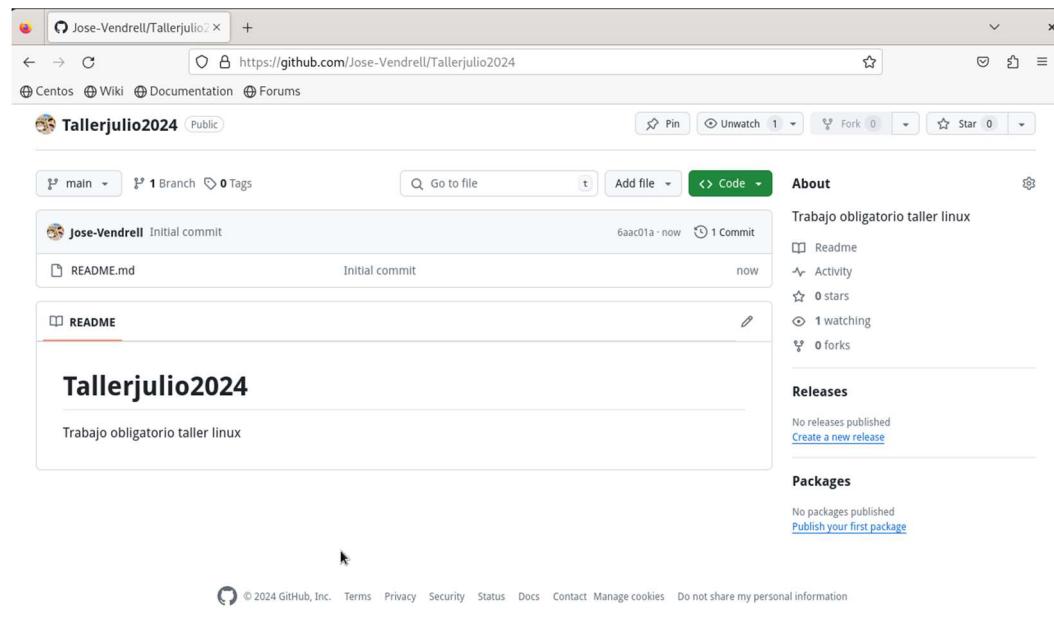
```
.CentOS Stream 9 - Extras packages          48 kB/s | 27 kB    00:00
Package git-2.43.5-1.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[sysadmin@controller ~]$ 
[sysadmin@controller ~]$ 
[sysadmin@controller ~]$ sudo git config --global user.name "Jose Vendrell"
[sudo] password for sysadmin:
[sysadmin@controller ~]$ git config --global user.email "joselo_43@hotmail.com"
[sysadmin@controller ~]$ git config --global core.editor nano
[sysadmin@controller ~]$ git config --global color.ui true
[sysadmin@controller ~]$ git config --global color.status auto
[sysadmin@controller ~]$ git config --global color.branch auto
[sysadmin@controller ~]$ git config --global merge.tool vimdiff
[sysadmin@controller ~]$ git config --global core.editor vim
[sysadmin@controller ~]$ git config --list
user.name=joselo_43@hotmail.com
core.editor=vim
color.ui=true
color.status=auto
color.branch=auto
merge.tool=vimdiff
[sysadmin@controller ~]$ 
```

Creación clave publica ssh en el **Controller**:

- ssh-keygen (Clave pública y privada ssh en el controller, el cual se coloca en la web de github > settings > ssh and GPG Keys > new ssh Key)
 - cat .ssh/id_rsa.pub



Luego podemos crear un repositorio en github y agregar el README file: Perfil > repositorios > new > nombre, el mismo se deja publico > add a readme file > crear repositorio.



Clave Pública desde github a los Servidores:

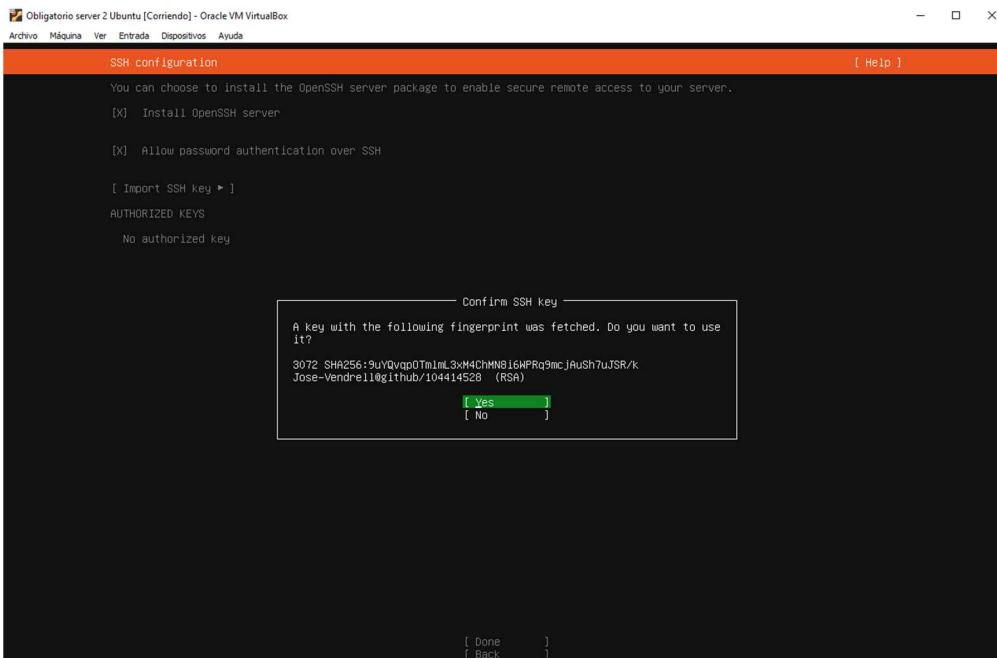
En el servidor de CentOS se utiliza git clone <URL> de github.

- git clone <https://github.com/Jose-Vendrell/Tallerjulio2024.git>

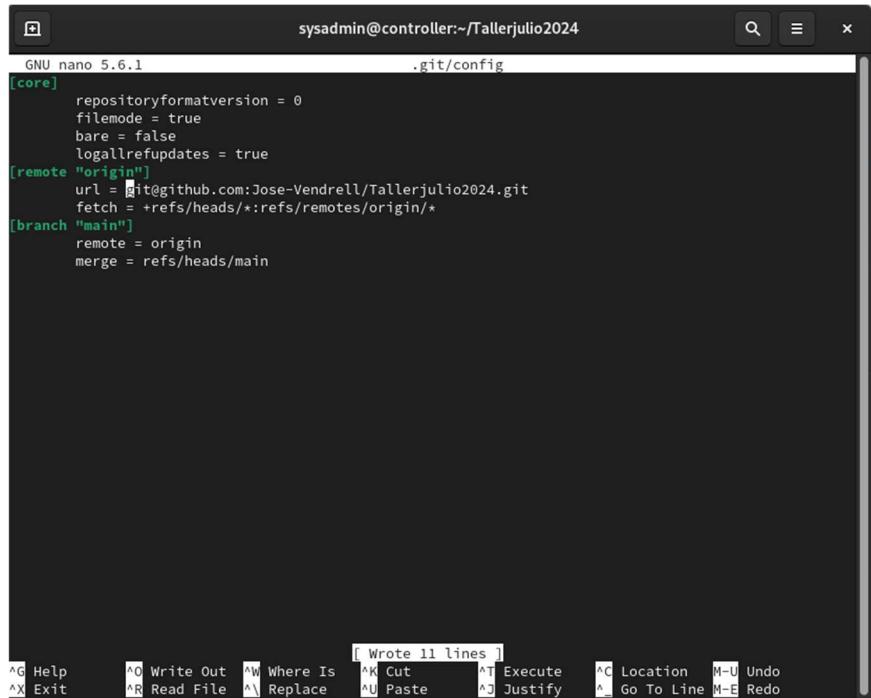
En la imagen muestra el controller, pero se realiza desde el servidor01 CentOS

The screenshot shows a terminal window titled "sysadmin@controller:~/Tallerjulio2024". The session starts with a SHA256 hash of some text, followed by the command to cat .ssh/id_rsa.pub. The output is a long RSA public key. Then, the user runs "git clone https://github.com/Jose-Vendrell/Tallerjulio2024.git" which clones the repository into the current directory. The terminal then lists the contents of the repository, showing files like README.md and various git submodules. The final command shown is "ls .git/".

En el servidor02 Ubuntu se importa durante la instalación, como se mostró anteriormente:



Archivo .git/config con la configuración correcta por ssh y la URL de la clave por github:



The screenshot shows a terminal window titled "sysadmin@controller:~/Tallerjulio2024". The window contains the contents of the ".git/config" file, which is being edited with the nano text editor. The configuration includes settings for the core repository, a remote named "origin" pointing to a GitHub URL, and a branch named "main". The status bar at the bottom indicates "Wrote 11 lines".

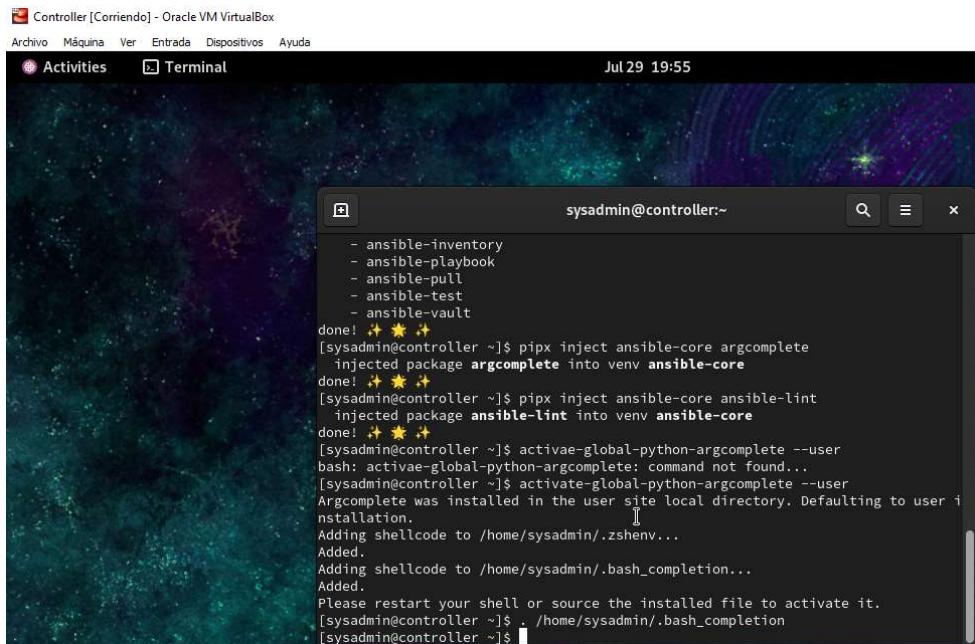
```
GNU nano 5.6.1 .git/config
[core]
repositoryformatversion = 0
filemode = true
bare = false
logallrefupdates = true
[remote "origin"]
url = git@github.com:Jose-Vendrell/Tallerjulio2024.git
fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
remote = origin
merge = refs/heads/main
```

Ansible

En Controller vemos si contamos con ansible: `dnf search ansible`

Instalación pipx:

- `dnf install python3-pip` (gestor de paquetes python3)
- `pip install pipx` (herramienta para instalar y gestionar aplicaciones Python en entornos aislados)
- `pipx ensurepath` (nos permite ejecutar comandos sin necesidad de especificar la ruta completa)
- `pipx install ansible-core` (instala en un entorno virtual aislado para evitar conflictos con otros paquetes de Python)
- `pipx inject ansible-core argcomplete` (proporcionar sugerencias de autocompletado en la línea de comandos)
- `pipx inject ansible-core ansible-lint` (herramienta para verificar la calidad del código en los playbooks de Ansible)
- `activate-global-python-argcomplete --user` (configura el autocompletado para argcomplete en el usuario actual)
- `. /home/sysadmin/.bash_completion` (carga el archivo bash el cual contiene autocompletado de los comandos que se definieron en el mismo)



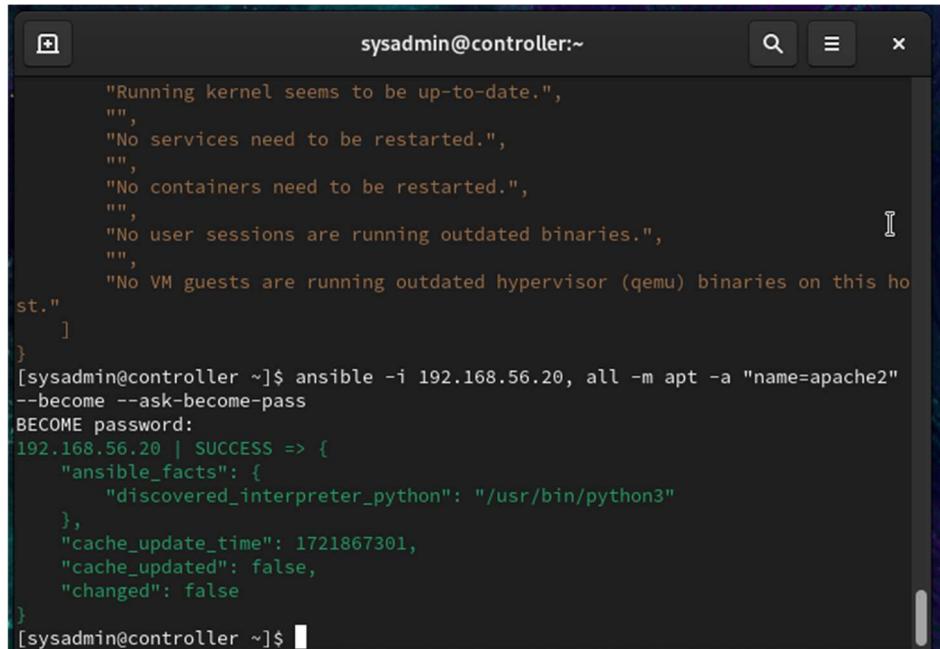
The screenshot shows a terminal window titled "sysadmin@controller:~". The terminal displays the following command sequence:

```
[sysadmin@controller ~]$ pipx inject ansible-core argcomplete
[injected package argcomplete into venv ansible-core]
done! ✨ ✨ ✨
[sysadmin@controller ~]$ pipx inject ansible-core ansible-lint
[injected package ansible-lint into venv ansible-core]
done! ✨ ✨ ✨
[sysadmin@controller ~]$ activate-global-python-argcomplete --user
bash: activate-global-python-argcomplete: command not found...
[sysadmin@controller ~]$ activate-global-python-argcomplete --user
Argcomplete was installed in the user site local directory. Defaulting to user installation.
[sysadmin@controller ~]$ Adding shellcode to /home/sysadmin/.zshenv...
Added.
[sysadmin@controller ~]$ Adding shellcode to /home/sysadmin/.bash_completion...
Added.
[sysadmin@controller ~]$ Please restart your shell or source the installed file to activate it.
[sysadmin@controller ~]$ . /home/sysadmin/.bash_completion
[sysadmin@controller ~]$
```

Copiar clave publica y colocar en los servidores

Desde el Controller se copia la clave publica y se coloca en los servidores CentOS y Ubuntu. Las claves publicas se crearon sin contraseña.

- ssh-copy-id 192.168.56.20 (Servidor Ubuntu, el mismo debe estar encendido)
- ssh-copy-id 192.168.56.104 (Servidor CentOS)



The screenshot shows a terminal window titled "sysadmin@controller:~". The terminal displays the following output:

```
"Running kernel seems to be up-to-date.",
"",
"No services need to be restarted.",
"",
"No containers need to be restarted.",
"",
"No user sessions are running outdated binaries.",
"",
"No VM guests are running outdated hypervisor (qemu) binaries on this host."
]
}
[sysadmin@controller ~]$ ansible -i 192.168.56.20, all -m apt -a "name=apache2"
--become --ask-become-pass
BECOME password:
[192.168.56.20 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1721867301,
    "cache_updated": false,
    "changed": false
}
[sysadmin@controller ~]$
```

- ansible -i 192.168.56.20, all -m apt -a "name=apache2" --become --ask-become-pass

Este comando instala el paquete apache2, especificando el inventario / host en el que se aplicara la tarea / el modulo apt / argumento name=apache2 / become para ejecutar privilegios elevados / ask-become-pass solicita la contraseña del usuario become.

Una vez ya todos los servidores cuentan con ssh-copy-id

```
ansible-playbook -i inventory/servidores hardening.yml --
ask-become-pass
```

```

sysadmin@controller:~/Tallerjulio2024$ ansible-playbook -i inventory/servidores hardening.yml --ask-become-pass
BECOME password:

PLAY [Hardening de servidores] ****
TASK [Gathering Facts] ****
ok: [server02]
ok: [server03]
ok: [server01]

TASK [Configuro opciones de seguridad de ssh] ****
ok: [server02]
ok: [server03]
ok: [server01]

TASK [Ver resultado de tarea anterior] ****
ok: [server01] => {
    "results_sshd": "VARIABLE IS NOT DEFINED!"
}
ok: [server02] => {
    "results_sshd": "VARIABLE IS NOT DEFINED!"
}
ok: [server03] => {
    "results_sshd": "VARIABLE IS NOT DEFINED!"
}

PLAY RECAP ****
server01      : ok=3    changed=0     unreachable=0    failed=0    skipped=0    rescued=0
server02      : ok=3    changed=0     unreachable=0    failed=0    skipped=0    rescued=0
server03      : ok=3    changed=0     unreachable=0    failed=0    skipped=0    rescued=0

```

Utilizando `--syntax-check` realiza un chequeo de sintaxis para ver si esta correcto o si hay algún error en alguna línea/columna. Si devuelve el nombre del playbook es que esta correcto.

```

[sysadmin@controller Tallerjulio2024]$ nano hardening.yml
[sysadmin@controller Tallerjulio2024]$ ansible-playbook -i inventory/servidores hardening.yml --syntax-check
playbook: hardening.yml
[sysadmin@controller Tallerjulio2024]$ 

```

Utilizando `git status` muestra que cambios están pendientes de confirmación

`git add .` (prepara todos los cambios actuales para el próximo commit)

```

[sysadmin@controller Tallerjulio2024]$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    hardened.yml
    inventory/
nothing added to commit but untracked files present (use "git add" to track)
[sysadmin@controller Tallerjulio2024]$ git add .
[sysadmin@controller Tallerjulio2024]$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   hardened.yml
    new file:   inventory/host vars/server01.yml
    new file:   inventory/host vars/server02.yml
    new file:   inventory/host vars/server03.yml
    new file:   inventory/servidores

```

Utilizando únicamente git commit se abrirá un editor de texto para que se pueda escribir un mensaje descriptivo del commit.

```
[sysadmin@controller Tallerjulio2024]$ git commit
[main 78441ce] Se realiza primer playbook Como primer playbook se hace un hardening de ssh. Se agrega el inventario con servidores centos y ubuntu Se agregan las variables de inventario No permite el acceso al no root
 5 files changed, 46 insertions(+)
 create mode 100644 hardening.yml
 create mode 100644 inventory/host vars/server01.yml
 create mode 100644 inventory/host vars/server02.yml
 create mode 100644 inventory/host vars/server03.yml
 create mode 100644 inventory/servidores
```

git log muestra el resultado sobre el git commit de lo que se realizó

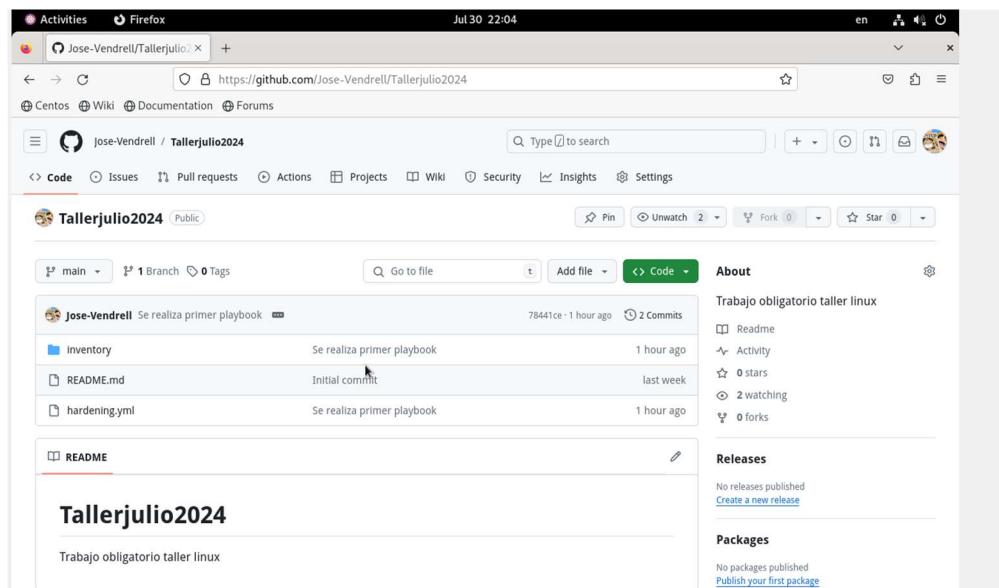
```
[sysadmin@controller Tallerjulio2024]$ git log
commit 78441ce9f0af3cc74be4cb6e03e4818de221c0cb (HEAD -> main)
Author: System Admin <joselo_43@hotmail.com>
Date:   Tue Jul 30 20:57:23 2024 -0300

    Se realiza primer playbook
    Como primer playbook se hace un hardening de ssh.
    Se agrega el inventario con servidores centos y ubuntu
    Se agregan las variables de inventario
    No permite el acceso al no root

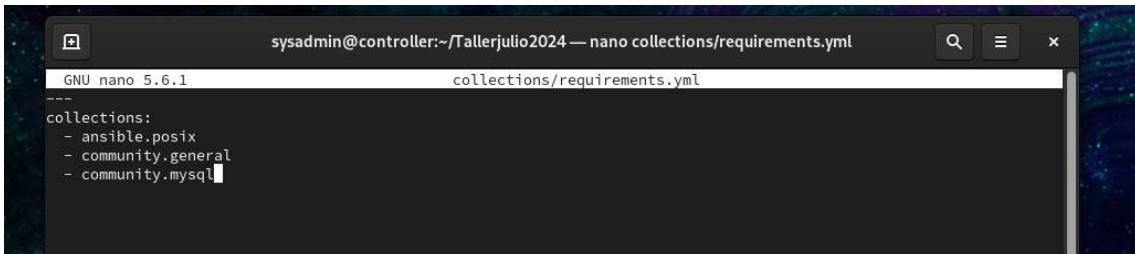
commit 6aac01a7f51528c2d06777ca36f777a821e66631 (origin/main, origin/HEAD)
Author: Jose-Vendrell <joselo_43@hotmail.com>
Date:   Wed Jul 24 21:03:39 2024 -0300

    Initial commit
[sysadmin@controller Tallerjulio2024]$
```

Luego se utiliza el comando git push el cual envía los playbooks a github.



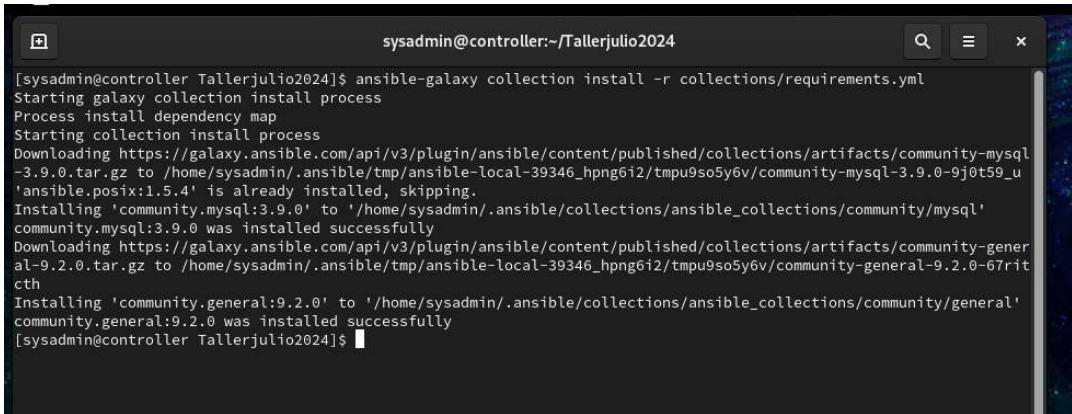
Realizamos la creación de collections/requirements.yml el cual serán los requisitos a tener en cuenta para que los módulos funcionen correctamente.



```
sysadmin@controller:~/Tallerjulio2024 — nano collections/requirements.yml
GNU nano 5.6.1
collections:
- ansible.posix
- community.general
- community.mysql
```

Se realiza la instalación de ansible galaxy para que reconozca módulos que se utilizaran para los playbooks de Servidor web y Base de datos.

```
ansible-galaxy collection install -r
collections/requirements.yml
```



```
[sysadmin@controller Tallerjulio2024]$ ansible-galaxy collection install -r collections/requirements.yml
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/artifacts/community-mysql-3.9.0.tar.gz to /home/sysadmin/.ansible/tmp/ansible-local-39346_hpng6i2/tmpu9so5y6v/community-mysql-3.9.0-9j0t59_u
'ansible.posix:1.5.4' is already installed, skipping.
Installing 'community.mysql:3.9.0' to '/home/sysadmin/.ansible/collections/ansible_collections/community/mysql'
community.mysql:3.9.0 was installed successfully
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/artifacts/community-general-9.2.0.tar.gz to /home/sysadmin/.ansible/tmp/ansible-local-39346_hpng6i2/tmpu9so5y6v/community-general-9.2.0-67rit
Installing 'community.general:9.2.0' to '/home/sysadmin/.ansible/collections/ansible_collections/community/general'
community.general:9.2.0 was installed successfully
[sysadmin@controller Tallerjulio2024]$
```

Creación Playbooks, directorios, archivos de configuración y se suben a github.

Requirement y hardening:

The screenshot shows a GitHub repository page for 'Tallerjulio2024 / collections'. The file 'requirements.yml' is displayed. It contains the following YAML code:

```
---  
collections:  
  - ansible.posix  
  - community.general  
  - community.mysql
```

The screenshot shows a GitHub repository page for 'Jose-Vendrell / Tallerjulio2024'. The file 'hardening.yml' is displayed. It contains the following YAML code:

```
---  
- name: Hardening de servidores centos  
  hosts: linux  
  become: true  
  user: sysadmin  
  
  tasks:  
  
    - name: Configuro opciones de seguridad de ssh  
      ansible.builtin.lineinfile:  
        path: /etc/ssh/sshd_config  
        regexp: '#PermitRootLogin'  
        line: PermitRootLogin no  
        notify: Reinicio servidor ssh  
        register: results_sshd  
  
    - name: Ver resultado de tarea anterior  
      ansible.builtin.debug:  
        var: results_sshd  
  
  handlers:  
  
    - name: Reinicio servidor ssh  
      ansible.builtin.systemd_service:  
        name: "{{ ssh_service }}"  
        state: restarted
```

Web server:

Tallerjulio2024 / webserver.yml

Jose-Vendrell Update webserver.yml 4832f42 · 2 days ago

Code **Blame** 68 lines (57 loc) · 1.47 KB **Raw**   

```
1   ---
2   - name: Instalacion y configuracion webserver
3     hosts: centos
4     become: true
5     user: sysadmin
6
7     tasks:
8
9       - name: Instalar httpd (Apache)
10      ansible.builtin.dnf:
11        name: httpd
12        state: present
13
14       - name: Levantar y habilitar servicio httpd
15      ansible.builtin.systemd_service:
16        name: httpd
17        enabled: true
18        state: started
19
20       - name: Crear directorio document root
21      ansible.builtin.file:
22        path: /var/www/html/obligatorio
23        state: directory
24        owner: apache
25        group: apache
26        mode: '0755'
27
28       - name: Copiar pagina index.j2
29      ansible.builtin.copy:
30        src: ./templates/index.j2
31        dest: /var/www/html/obligatorio/index.html
32        owner: apache
33        group: apache
34        mode: '0644'
35
36       - name: Configurar virtualhost 80
37      ansible.builtin.template:
38        src: ./files/virtualhost.conf
39        dest: /etc/httpd/conf.d
40        owner: apache
41        group: apache
42        mode: '0644'
```

```

44      - name: Abrir puertos 80 en firewall
45      ansible.posix.firewalld:
46          service: http
47          permanent: true
48          state: enabled
49          notify: Reinicio firewalld
50
51      - name: Abrir puerto 8080 en firewall
52      ansible.posix.firewalld:
53          port: 8080/tcp
54          permanent: true
55          state: enabled
56          notif: Reinicio firewalld
57
58      handlers:
59          - name: Reiniciar Firewalld
60          ansible.builtin.systemd:
61              name: firewalld
62              state: reloaded
63
64          - name: Reiniciar Apache
65          ansible.builtin.systemd:
66              name: httpd
67              state: restarted

```

```

Controller [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Aug 4 00:32
Jose-Vendrell sysadmin@localhost:~/Templates/Tallerjulio2024 — sudo nano webserver.yml
GNU nano 5.6.1
webserver.yml
---
- name: Instalacion y configuracion webserver
  hosts: centos
  become: true
  user: sysadmin

  tasks:
    - name: Instalar httpd (Apache)
      ansible.builtin.dnf:
        name: httpd
        state: present

    - name: Levantar y habilitar servicio httpd
      ansible.builtin.systemd_service:
        name: httpd
        enabled: true
        state: started

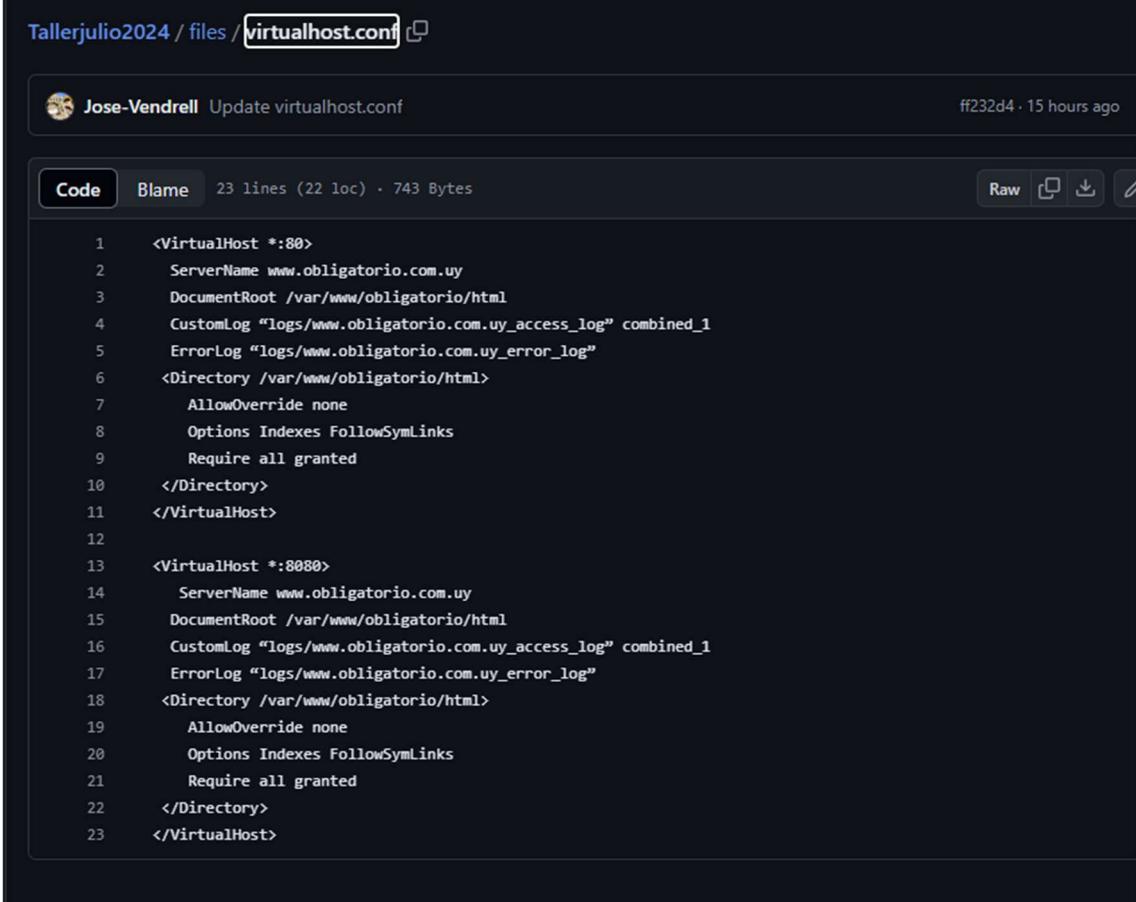
    - name: Crear directorio document root
      ansible.builtin.file:
        path: /var/www/html/obligatorio
        state: directory
        owner: apache
        group: apache
        mode: '0755'

    - name: Copiar pagina index.j2
      ansible.builtin.copy:
        src: ./templates/index.j2
        dest: /var/www/html/obligatorio/index.html
        owner: apache
        group: apache
        mode: '0644'

    - name: Configurar virtualhost 80
      ansible.builtin.template:

```

Virtualhost.conf:



The screenshot shows a GitHub commit page for a file named 'virtualhost.conf'. The commit was made by 'Jose-Vendrell' on 'Tallerjulio2024 / files / virtualhost.conf'. The commit message is 'Update virtualhost.conf'. The code is a configuration file for Apache, defining two VirtualHost blocks. The first block (lines 1-12) is for port 80, serving 'www.obligatorio.com.uy' from '/var/www/obligatorio/html'. The second block (lines 13-23) is for port 8080, also serving 'www.obligatorio.com.uy' from the same directory. Both blocks use combined log files and require 'all granted' access.

```
1 <VirtualHost *:80>
2   ServerName www.obligatorio.com.uy
3   DocumentRoot /var/www/obligatorio/html
4   CustomLog "logs/www.obligatorio.com.uy_access_log" combined_1
5   ErrorLog "logs/www.obligatorio.com.uy_error_log"
6   <Directory /var/www/obligatorio/html>
7     AllowOverride none
8     Options Indexes FollowSymLinks
9     Require all granted
10    </Directory>
11  </VirtualHost>
12
13 <VirtualHost *:8080>
14   ServerName www.obligatorio.com.uy
15   DocumentRoot /var/www/obligatorio/html
16   CustomLog "logs/www.obligatorio.com.uy_access_log" combined_1
17   ErrorLog "logs/www.obligatorio.com.uy_error_log"
18   <Directory /var/www/obligatorio/html>
19     AllowOverride none
20     Options Indexes FollowSymLinks
21     Require all granted
22   </Directory>
23 </VirtualHost>
```

Base de datos yml

TallerJulio2024 / **database.yml**

Jose-Vendrell Update database.yml 33de5c3 · 31 minutes ago

Code Blame 107 lines (85 loc) · 2.37 KB Raw

```
1     ---
2     - name: Configuracion servidor base de datos en Ubuntu
3       hosts: server02
4       become: true
5       user: sysadmin
6
7       tasks:
8
9       ## Configuracion Firewall y trafico
10
11      - name: UFW instalado
12        ansible.builtin.apt:
13          name: ufw
14          state: present
15
16      - name: Permitir puerto 22 en ufw
17        community.general.ufw:
18          rule: allow
19          name: OpenSSH
20
21      - name: Defino politicas de tráfico entrante
22        community.general.ufw:
23          policy: allow
24          direction: outgoing
25          state: enabled
26
27      - name: Defino politicas de tráfico entrante
28        community.general.ufw:
29          policy: deny
30          direction: incoming
31          state: enabled
32
33      ## Se levanta el servicio de Firewall de Ubuntu
34
35      - name: servicio UFW levantado y activo
36        ansible.builtin.systemd_service:
37          name: ufw
38          state: started
39          enabled: true
```

```
41      ## Instalación de MariaDB
42
43      - name: MariaDB instalado
44          ansible.builtin.apt:
45              name: mariadb-server
46              state: present
47              update-cache: true
48      loop:
49          - mariadb-server
50          - mariadb-client
51          - python3-pymysql
52
53      - name: Cambiar la configuracion para escuchar en todas las interfaces
54          ansible.builtin.lineinfile:
55              path: /etc/mysql/mariadb.conf.d/50-server.cnf
56              regexp: '^bind-address'
57              line: 'bind-address      = 0.0.0.0'
58              notify: Restart mariadb
59
60      - name: Ejecuto el handler si cambió la configuración
61          meta: flush_handlers
62
63      - name: Servidor Mariadb levantado
64          ansible.builtin.systemd_service:
65              name: mariadb
66              state: started
67              enabled: true
68
69      - name: Habilitamos en ufw la conexión a mariadb
70          community.general.ufw:
71              rule: allow
72              port: '3306'
73              protocol: tcp
74              direction: in
75
76      ## Configuracion de la base de datos de la aplicación
77
78      - name: Copio el dump de la base de datos
79          ansible.builtin.copy:
80              src: ./files/todo.sql
81              dest: /tmp/todo.sql
```

```

82
83     - name: Creacion base de datos Todo
84         community.mysql.mysql_db:
85             check_implicit_admin: true
86             login_host: localhost
87             login_user: root
88             login_password: dbadmin
89             name: todo
90             state: import
91             target: ./files/todo.sql
92
93     ## Remover base de datos de prueba
94
95     - name: Remove test database and access
96         mysql_db:
97             name: test
98             state: absent
99
100    ## Reinicio servicio mariadb
101
102    handlers:
103
104        - name: Reiniciar mariadb
105            ansible.builtin.systemd_service:
106                name: mariadb
107                state: restarted

```

Archivo todo.sql

TallerJulio2024 / files / [todo.sql](#)

Jose-Vendrell Update todo.sql 2b6cefb · 3 days ago

Code Blame 26 lines (23 loc) · 827 Bytes Raw ⌂ ⌄

```
1 CREATE DATABASE todo;
2 USE todo;
3
4 CREATE TABLE `users` (
5     `id` int(3) NOT NULL AUTO_INCREMENT,
6     `first_name` varchar(20) DEFAULT NULL,
7     `last_name` varchar(20) DEFAULT NULL,
8     `username` varchar(250) DEFAULT NULL,
9     `password` varchar(20) DEFAULT NULL,
10    PRIMARY KEY (`id`)
11 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
12
13 CREATE TABLE `todos` (
14     `id` bigint(20) NOT NULL AUTO_INCREMENT,
15     `description` varchar(255) DEFAULT NULL,
16     `is_done` bit(1) NOT NULL,
17     `target_date` datetime(6) DEFAULT NULL,
18     `username` varchar(255) DEFAULT NULL,
19     `title` varchar(255) DEFAULT NULL,
20     PRIMARY KEY (`id`)
21 ) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4
22 COLLATE=utf8mb4_0900_ai_ci;
23
24 CREATE USER 'todo'@'%' IDENTIFIED BY "prueba2024";
25 GRANT ALL PRIVILEGES ON todo.* TO 'todo'@'%';
26 FLUSH PRIVILEGES;
```

Tomcat.yml

```

1   ---
2   - name: Instalar todo en Centos
3     hosts: centos
4     user: sysadmin
5     become: true
6
7     tasks:
8
9     ## Instalamos tar para manejar archivos .tar
10
11    - name: Instalar tar
12      ansible.builtin.dnf:
13        name: tar
14        state: present
15
16    ## Descarga y extraccion del archivo tomcat en /opt
17
18    - name: Descargar Tomcat
19      ansible.builtin.get_url:
20        url: "https://downloads.apache.org/tomcat/tomcat-9/v9.0.56/bin/apache-tomcat-9.0.56.tar.gz"
21        dest: "/tmp/apache-tomcat-9.0.56.tar.gz"
22        mode: '0644'
23
24    - name: Extraer Tomcat
25      ansible.builtin.unarchive:
26        src: "/tmp/apache-tomcat-9.0.56.tar.gz"
27        dest: "/opt"
28        remote_src: true
29
30    - name: Creo enlace a tomcat
31      ansible.builtin.file:
32        src: "/opt/tomcat"
33        dest: "/opt/tomcat"
34        state: link
35        force: true
36
37    ## Los puertos de 8080 se abrieron en el webserver.yml

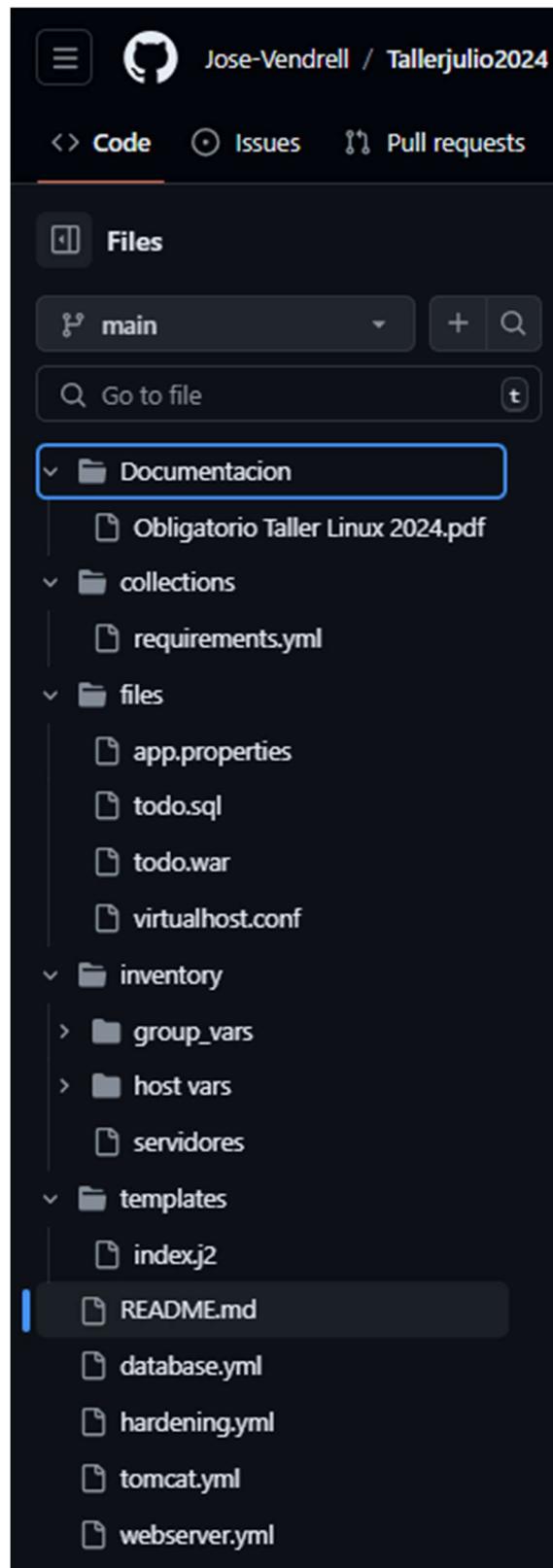
```

```

39    ## Iniciamos el Tomcat
40
41    - name: Iniciar Tomcat
42      ansible.builtin.systemd_service:
43        name: tomcat
44        state: started
45        enabled: true
46
47    - name: Copiar Todo.war al directorio de Tomcat
48      ansible.builtin.copy:
49        src: /TallerJulio2024/files/todo.war
50        dest: /opt/tomcat/webapps/todo.war
51
52    - name: Reiniciar Tomcat
53      ansible.builtin.systemd_service:
54        name: tomcat
55        state: restarted

```

Github completo Taller-julio 2024



Creación README

The screenshot shows a GitHub repository page for 'Tallerjulio2024 / README.md'. The file contains the following content:

Obligatorio Taller Linux Tallerjulio2024

Para el obligatorio se solicita contar con 3 vm's, de los cuales 1 es un controller centOS, otro se utilizara un servidor centos como servidor web y el ultimo sera un servidor ubuntu para la base de datos utilizando Mariadb.

Inventario

```
[centos] server01 ansible_host=192.168.56.103
[ubuntu] server02 ansible_host=192.168.56.105
[linuxchildren] centos ubuntu
```

GIT

Se instala git y se trae los documentos y claves publicas desde github. Tambien se baja el repositorio para utilizar los Playbooks

```
• dnf install git
• git config --global user.name "Jose Vendrell"
• git config --global user.email joselo_43@hotmail.com
• git clone https://github.com/Jose-Vendrell/Tallerjulio2024.git
```

En .git/config se visualizara la llave publica con url, se puede editar y colocar por ssh.

Se utiliza ssh-copy-id para colocar la llave publica en otros servidores

```
ssh-copy-id <IP>
```

The screenshot shows a GitHub repository page for 'Tallerjulio2024 / README.md'. The file contains the following content:

Ansible

Para el uso de ansible se necesita instalar y configurar pipx.

```
• dnf install python3-pip
• pip install pipx
• pipx ensurepath
• pipx install ansible-core
• pipx inject ansible-core
• pipx inject ansible-core ansible-lint argcomplete
• activate-global-python-argcomplete -user
• ./home/sysadmin/.bash_completion
```

Esto nos permite instalar y gestionar aplicaciones python en entornos aislados, ejecutar comandos sin necesidad de especificar la ruta, así como también autocompletados y herramientas para verificar la calidad del código de en los playbooks de Ansible.

Para correr los playbooks se utiliza:

```
ansible-playbook -i inventory/servidores hardening.yml --ask-become-pass
```

Esto realiza una lectura del archivo, ejecucion del playbook y solicita contraseña sudo para elevar los privilegios. Utilizando --syntax-check podremos corroborar que el archivo se encuentra correctamente y sin errores de gramática o espacios.

Tallerjulio2024 / README.md ↑ Top

Preview Code Blame 95 lines (81 loc) · 3.58 KB GitHub Copilot 55% faster with GitHub Copilot Raw ⌂ ⌄ ⌅ ⌆ ⌇

Requisitos Ansible

Para poder trabajar correctamente con los playbooks de este obligatorio, debemos instalar los modulos correspondientes:

```
ansible-galaxy collection install -r collections/requirements.yml
```

Collections:

- name: ansible.posix
- name: community.general
- name: community.mysql

Webserver

```
ansible-playbook -i inventory/webserver.yml --ask-become-pass
```

Instala httpd (Apache), abre puertos, habilita firewall y configura virtualhost con el archivo virtualhost.conf en /files/virtualhost.conf

Tomcat

```
ansible-playbook -i inventory/tomcat.yml --ask-become-pass
```

Realiza la instalacion de la herramienta tar, descarga y descomprime tomcat, creando su directorio opt/tomcat, inicia el servicio y copia el archivo todo.war

Base de datos

```
ansible-playbook -i inventory/database.yml --ask-become-pass
```

Configura el firewall y el trafico del mismo, levanta el servicio e instala maria db. Tambien configura la base de datos de la aplicacion y remueve datos de prueba creados por default

todo.sql

Cuenta con la configuracion de la base de datos todo. Este se encuentra en /Tallerjulio2024/files/todo.sql

Referencias

Verificar siempre las versiones para no tener inconvenientes!!!

Mas Modulos de Ansible en: [Link Ansible modules](#)

Github profesor de la materia [Link git emverdes](#)

Corregir syntaxis yml: [Lector yml](#)

Creacion documento readme: [Make a reedme](#)

Tambien se utilizo editar directamente de github en README.md