

Obligatorio

Taller Linux

Febrero-2022



Ignacio Acosta N.º 261660 – Marcos Rodriguez N.º 152579

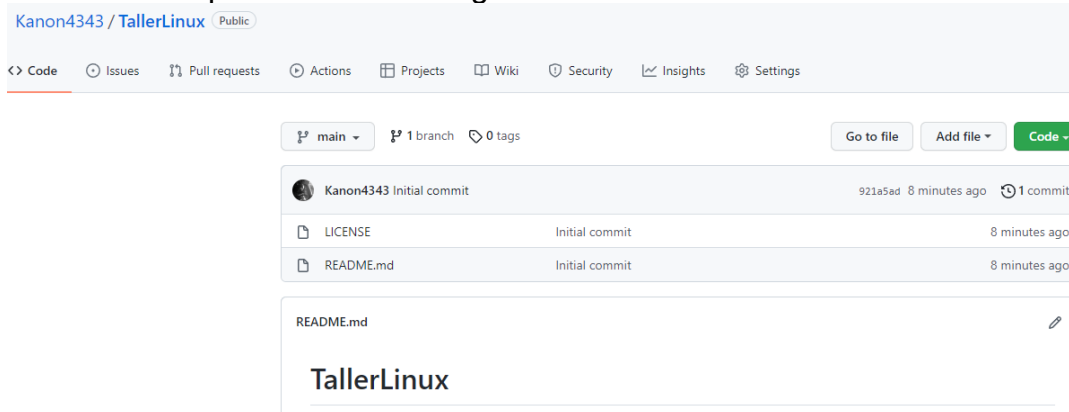
Índice

GitHub.....	3
Creación de repositorio GitHub	3
SSH para GitHub	3
Integración de GitHub con servidor bastión.....	4
Usuario Ansible.....	5
Servidor Rocky	5
Pasos de la instalación.....	5
Servidor Ubuntu.....	7
Pasos de la instalación.....	7
Configuración de networking	11
Permiso SUDO sin contraseña para ansible	11
Ansible.....	12
Compartir SSH key con los servidores	12
Comprobación de ejecución de módulos ansible en clientes.....	12
Inventario	12
Playbooks.....	13
Actualizador.yml.....	13
Tomcat.yml.....	14
Mysql.yml	16
Apache.yml	18
Main.yml.....	20
Materiales	20
Apache Debian:.....	20
Apache RedHat:	20
App.properties.....	20
SQL script	21
Tomcat Service para Debian	21
Tomcat Service para RedHat	22
Programa funcionando.....	22

GitHub

Creación de repositorio GitHub

Creamos el repositorio con las siguientes características:

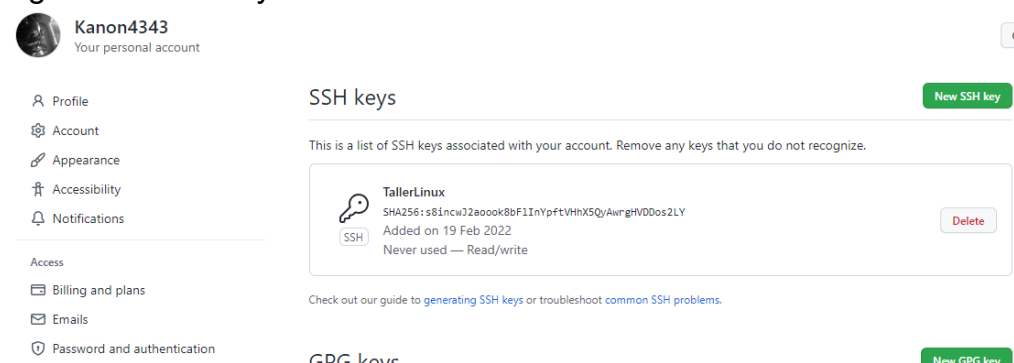


SSH para GitHub

Generamos un SSH key para GitHub en el equipo bastión con el comando ssh-keygen.

```
[root@terminal clavessh]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): clave
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in clave.
Your public key has been saved in clave.pub.
The key fingerprint is:
SHA256:7decj4vvmRE8xZ6ypRxyGS77RXv6zpUt/L0vXA8XZIE root@terminal.local
The key's randomart image is:
+---[RSA 3072]---+
|      . . .      |
|     E +        |
|    . o o       |
|   . . o +.    |
|  S . o *++o   |
|   . *+o*+    |
|  ...=BB=     |
|   ...+@=     |
|   .+O*@      |
+---[SHA256]-----+
[root@terminal clavessh]#
```

Agremamos la key en GitHub:



Activamos la clave privada en Bastión:

```
[root@bastion home]# eval $(ssh-agent -s)
Agent pid 4596
[root@bastion home]# ls
ansible
[root@bastion home]# cd ansible/
[root@bastion ansible]# ls
clave  clave.pub
[root@bastion ansible]# ssh-add clave
Enter passphrase for clave:
Identity added: clave (root@bastion.local)
[root@bastion ansible]# _
```

Integración de GitHub con servidor bastión

Rocky:

Instalamos git: `yum install git -y`

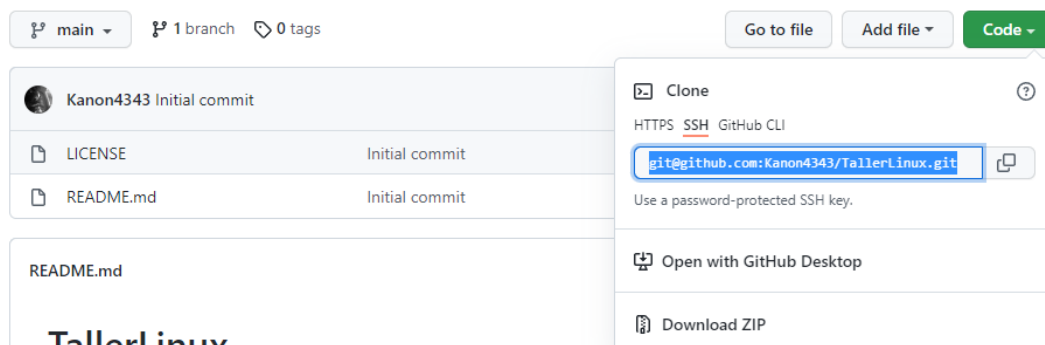
Configuramos el usuario: `git config --global user.name "Kanon4343"`

`git config --global user.email "mgrh43@gmail.com"`

Creamos la carpeta: `mkdir git_workspace`

Clonaremos el repositorio previamente creado "TallerLinux" copiando la URL:

`git@github.com:Kanon4343/TallerLinux.git`



```
[root@bastion git_workspace]# git clone git@github.com:Kanon4343/TallerLinux.git
Cloning into 'TallerLinux'...
The authenticity of host 'github.com (20.201.28.151)' can't be established.
ECDSA key fingerprint is SHA256:p2QAMXN1C1TJYWeIOtrUc9B/R1BUFWu3/LiyKgUfQM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com,20.201.28.151' (ECDSA) to the list of known hosts.
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
Receiving objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
[root@bastion git_workspace]# ls
TallerLinux
```

Usuario Ansible

Crearemos en los equipos un usuario Ansible con las siguientes características:

The image shows two screenshots from a Linux installation process. The top screenshot is the 'CREATE USER' screen, which has a 'Done' button in the top left. It contains fields for 'Full name' (ansible), 'User name' (ansible), and 'Password' (masked with dots). There are checkboxes for 'Make this user administrator' and 'Require a password to use this account', both of which are checked. A password strength indicator shows 'Weak'. Below the password field is a 'Confirm password' field, also masked. An 'Advanced...' button is at the bottom. The bottom screenshot is the 'Profile setup' screen, which has a '[Help]' link in the top right. It contains fields for 'Your name' (Ansible), 'Your server's name' (ubuntu), 'Pick a username' (ansible), 'Choose a password' (masked), and 'Confirm your password' (masked).

Servidor Rocky

Se implementa la instalación del router con las siguientes características:

- 1) ISO usada: Rocky-8.5-x86_64-minimal
- 2) CPU: 1, RAM: 1024MB
- 3) Hostname: rocky.local

Pasos de la instalación

The image shows the 'INSTALLATION DESTINATION' screen from the Rocky Linux installer. It has a 'Done' button in the top left and a 'latam' logo in the top right. The screen is titled 'Device Selection' and contains the text: 'Select the device(s) you'd like to install to. They will be left untouched until you click on the main menu's "Begin Installation" button.' Below this, there are two sections: 'Local Standard Disks' and 'Specialized & Network Disks'. The 'Local Standard Disks' section shows a single disk: 'Msft Virtual Disk 60022480371ca1b5b0c68fc7685763ec' with a size of '127 GiB' and '127 GiB free'. The 'Specialized & Network Disks' section is empty. At the bottom, there is a 'Storage Configuration' section with two radio buttons: 'Automatic' and 'Custom'. The 'Custom' button is selected, and a red arrow points to it. The text 'Disks left unselected' appears at the bottom right of the disk selection area.

MANUAL PARTITIONING

ROCKY LINUX

latam

Done

▼ New Rocky Linux 8 Installation

You haven't created any mount points for your Rocky Linux 8 installation yet. You can:

- [Click here to create them automatically.](#)
- Create new mount points by clicking the '+' button.

New mount points will use the following partitioning scheme:

LVM

Encrypt automatically created mount points by default:

☐ Encrypt my data.

When you create mount points for your Rocky Linux 8 installation, view their details here.

MANUAL PARTITIONING

ROCKY LINUX

latam

Done

▼ New Rocky Linux 8 Installation

DATA

/home

1024 MiB

rL_rocky-home

/opt

5 GiB

rL_rocky-opt

SYSTEM

/

5 GiB

rL_rocky-root

/var

5 GiB

rL_rocky-var

/boot

1024 MiB

sda1

swap

2 GiB

rL_rocky-swap

rL_rocky-swap

Mount Point:

Desired Capacity:

2 GiB

Device Type:

LVM

File System:

swap

Label:

Device(s):

Msft Virtual Disk
60022480371ca
(sda)

Volume Group:

rL_rocky

Encrypt

☐

Reformat

☒

Modify...

Modify...

Name:

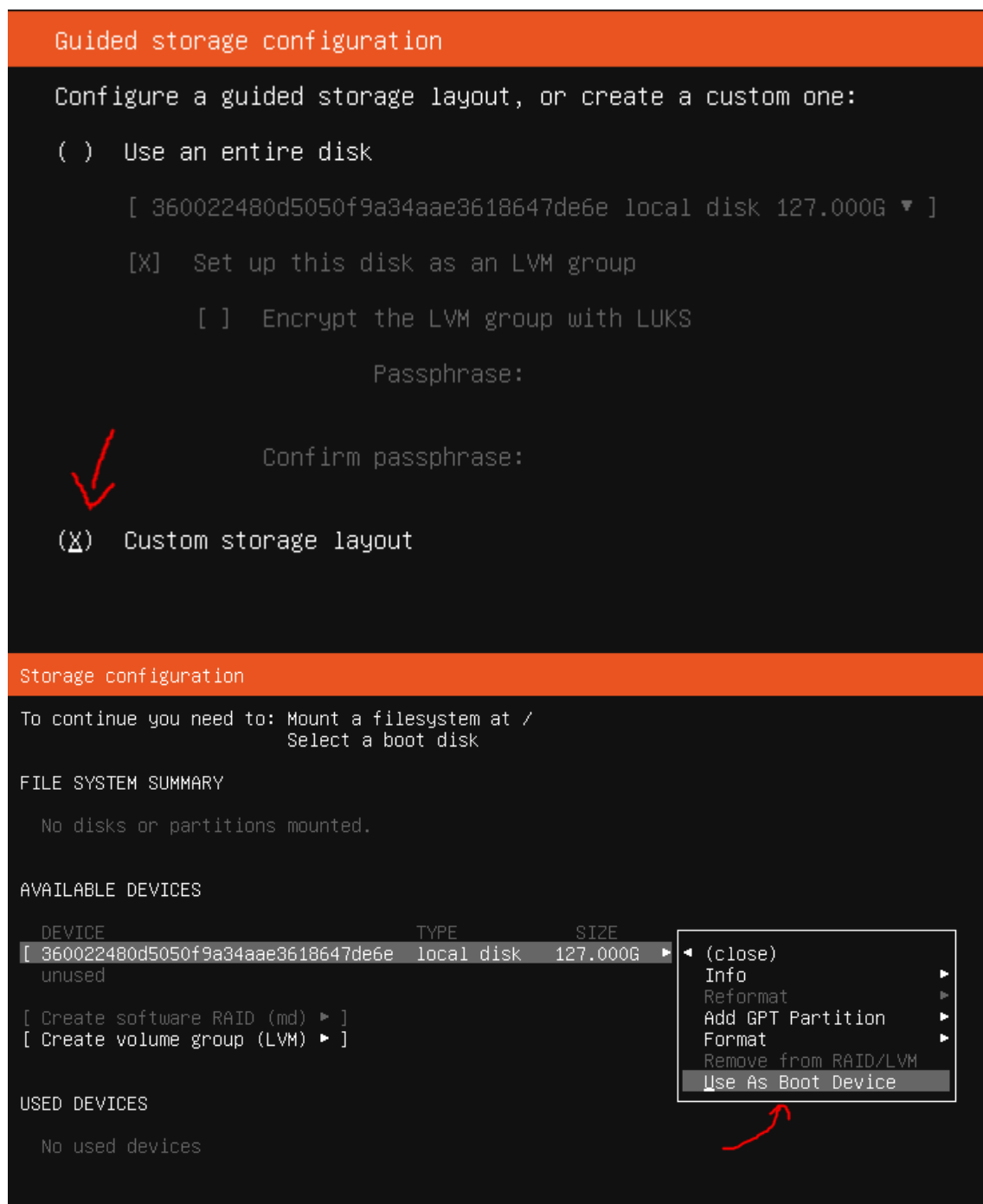
swap

Servidor Ubuntu

Se implementa la instalación con las siguientes características:

- 1) ISO usada: ubuntu-20.04.3-live-server-amd64
- 2) CPU: 1, RAM: 1024MB
- 3) Hostname: Ubuntu.local

Pasos de la instalación



```

Storage configuration

To continue you need to: Mount a filesystem at /

FILE SYSTEM SUMMARY

MOUNT POINT      SIZE      TYPE      DEVICE TYPE
[ /boot          1.000G    new ext4  new partition of local disk ► ]

AVAILABLE DEVICES

DEVICE                                TYPE      SIZE
[ 360022480d5050f9a34aae3618647de6e  local disk 127.000G ► ]
free space                                125.997G

[ Create software RAID (md) ► ]
[ Create volume group (LVM) ► ]

USED DEVICES

DEVICE                                TYPE      SIZE
[ 360022480d5050f9a34aae3618647de6e  local disk 127.000G ► ]
partition 1  new, BIOS grub spacer      1.000M ►
partition 2  new, to be formatted as ext4, mounted at /boot 1.000G ►

Storage configuration [ 4

To continue you need to: Mount a filesystem at /

FILE SYSTEM SUMMARY

MOUNT POINT      SIZE      TYPE      DEVICE TYPE
[ /boot          1.000G    new ext4  new partition of local disk ► ]

AVAILABLE DEVICES

DEVICE                                TYPE      SIZE
[ 360022480d5050f9a34aae3618647de6e  local disk 127.000G ► ]
free space                                125.997G

[ Create software RAID (md) ► ]
[ Create volume group (LVM) ► ]

USED DEVICES

DEVICE                                TYPE      SIZE
[ 360022480d5050f9a34aae3618647de6e  local disk 127.000G ► ]
partition 1  new, BIOS grub spacer      1.000M ►
partition 2  new, to be formatted as ext4, mounted at /boot 1.000G ►

```

◀ (close)

- ▶ Info
- ▶ Reformat
- ▶ Add GPT Partition
- ▶ Format
- ▶ Remove from RAID/LVM
- ▶ Stop Using As Boot Device

Storage configuration

To continue you need to: Mount a filesystem at /

FILE SYSTEM SUMMARY

MOUNT POINT	SIZE	TYPE	DEVICE TYPE
/boot	1.000G	new ext4	new partition of local disk ▶]

AVAILABLE DEVICES

DEVICE	TYPE	SIZE
[360022480d5050f9a34aae3618647de6e	local disk	127.000G ▶]
free space		125.997G

[Create software RAID (md) ▶]
 [Create volume group (LVM) ▶]

USED DEVICES

DEVICE	TYPE	SIZE
[360022480d5050f	partition 1	ne
partition 2	ne	

Adding GPT partition to 360022480d5050f9a34aae3618647de6e

Size (max 125.997G): 125.997G

Format: [Leave unformatted ▼]

Mount: [/ ▼]

[Create]
 [Cancel]

Storage configuration

To continue you need to: Mount a filesystem at /

FILE SYSTEM SUMMARY

MOUNT POINT	SIZE	TYPE	DEVICE TYPE
/boot	1.000G	new ext4	new partition of local disk ▶]

AVAILABLE DEVICES

DEVICE	TYPE	SIZE
[360022480d5050f	partition 3	ne

[Create software
 [Create volume g

USED DEVICES

DEVICE	TYPE	SIZE
[360022480d5050f	partition 1	ne
partition 2	ne	

Create LVM volume group

Name: vg0

Devices: 360022480d5050f9a34aae3618647de6e 127.000G
 local disk
 [X] partition 3 125.997G
 unused partition of local disk

Size: 125.996G

[] Create encrypted volume

Passphrase:

Confirm passphrase:

[Create]
 [Cancel]

```
Storage configuration [ He

To continue you need to: Mount a filesystem at /

FILE SYSTEM SUMMARY

MOUNT POINT    SIZE    TYPE    DEVICE TYPE
[ /boot        1.000G  new ext4  new partition of local disk ▶ ]

AVAILABLE DEVICES

DEVICE                                TYPE                                SIZE
[ vg0 (new)    LVM volume group  125.996G ▶ ]
unused

[ Create software RAID (md) ▶ ]
[ Create volume group (LVM) ▶ ]

USED DEVICES

DEVICE                                TYPE                                SIZE
[ 360022480d5050f9a34aae3618647de6e  local disk  127.000G ▶ ]
partition 1  new, BIOS grub spacer          1.000M ▶ ]
partition 2  new, to be formatted as ext4, mounted at /boot  1.000G ▶ ]
partition 3  new, PV of LVM volume group vg0          125.997G ▶ ]

Storage configuration

FILE SYSTEM SUMMARY

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        1.000G  new ext4  new LVM logical volume ▶ ]
[ /opt         5.000G  new ext4  new LVM logical volume ▶ ]
[ /var         5.000G  new ext4  new LVM logical volume ▶ ]
[ SWAP         2.000G  new swap  new LVM logical volume ▶ ]

AVAILABLE DEVICES

DEVICE                                TYPE                                SIZE
[ vg0 (new)    LVM volume group  125.996G ▶ ]
free space                                107.996G

[ Create software RAID (md) ▶ ]
[ Create volume group (LVM) ▶ ]

USED DEVICES

DEVICE                                TYPE                                SIZE
[ vg0 (new)    LVM volume group  125.996G ▶ ]
lv-raiz       new, to be formatted as ext4, mounted at /          5.000G ▶ ]
lv-var        new, to be formatted as ext4, mounted at /var      5.000G ▶ ]
lv-home       new, to be formatted as ext4, mounted at /home          1.000G ▶ ]
lv-opt        new, to be formatted as ext4, mounted at /opt        5.000G ▶ ]
lv-swap       new, to be formatted as swap                          2.000G ▶ ]

[ 360022480d5050f9a34aae3618647de6e  local disk  127.000G ▶ ]
partition 1  new, BIOS grub spacer          1.000M ▶ ]
partition 2  new, to be formatted as ext4, mounted at /boot  1.000G ▶ ]
partition 3  new, PV of LVM volume group vg0          125.997G ▶ ]
```

Configuración de networking

Los servidores Rocky y Ubuntu se conectan por red privada a un router (centosOS 8) que les brinda NAT.

Rocky:

```
[root@rocky ~]# nmcli
eth0: connected to eth0
    "The Linux Foundation Microsoft Hyper-U"
    ethernet (hv_netvsc), 00:15:5D:00:02:0B, hw, mtu 1500
    ip4 default
    inet4 192.168.0.5/24
    route4 192.168.0.0/24
    route4 0.0.0.0/0
    inet6 fe80::215:5dff:fe00:20b/64
    route6 fe80::/64

lo: unmanaged
    "lo"
    loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536

DNS configuration:
    servers: 8.8.8.8
    interface: eth0
```

Ubuntu:

```
ansible@ubuntu:/etc/netplan$ cat 00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    eth0:
      dhcp4: false
      addresses:
        - 192.168.0.4/24
      gateway4: 192.168.0.1
      nameservers:
        addresses: [8.8.8.8]
  version: 2
```

Luego se aplica con: `sudo netplan apply` (puede devolver error, pero puede ser porque intenta aplicar configuración sobre floppy disk que no tiene, se puede deshabilitar para que el error no aparezca).

Permiso SUDO sin contraseña para ansible

Le daremos la habilidad al usuario ansible elevarse con sudo sin pedir contraseña. (No recomendable).

Executamos `sudo visudo` y agregamos la siguiente línea:

```
GNU nano 4.8 /etc/sudoers.tmp
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

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

# Members of the admin group may gain root privileges
%admin    ALL=(ALL) ALL

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

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

#include_dir /etc/sudoers.d
```

Una vez guardado, ahora el usuario ansible puede utilizar sudo sin pedir contraseña.

Ansible

Vamos a realizar los siguientes pasos de instalación en la máquina Bastión:

- 1) Sudo dnf update -y
- 2) Sudo reboot
- 3) Sudo dnf install -y epel-release
- 4) Sudo dnf install ansible -y
- 5) Ansible --version

```
[root@bastion ~]# ansible --version
ansible 2.9.27
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.6/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.6.8 (default, Nov  9 2021, 14:44:26) [GCC 8.5.0 20210514 (Red Hat 8.5.0-3)]
[root@bastion ~]#
```

Compartir SSH key con los servidores

Ssh-keygen

```
eval "$(ssh-agent -s)"
```

```
ssh-add ~/.ssh/clave
```

```
ssh-copy-id ansible@192.168.0.5
```

```
ssh-copy-id ansible@192.168.0.4
```

Comprobación de ejecución de módulos ansible en clientes

En el bastión ejecutamos: `ansible -i 192.168.0.4,192.168.0.5, all -m ping`

```
[root@bastion .ssh]# ansible -i ansible@192.168.0.4,192.168.0.5, all -m ping
192.168.0.5 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
ansible@192.168.0.4 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

Inventario

En el archivo hosts por defecto (/etc/ansible/hosts) agregamos la siguiente configuración:

```
[redhat]
rocky                ansible_host=192.168.0.5

[debian]
ubuntu              ansible_host=192.168.0.4

[linux:children]
redhat
debian
```

Playbooks

Crearemos nuestros playbooks en el directorio TallerLinux/playbooks.

Actualizador.yml

```
---
- hosts: linux
  remote_user: ansible
  become: yes
  become_method: sudo

  tasks:

    - name: Actualizar paquetes RedHat
      dnf:
        name: "*"
        state: latest
        update_only: yes
        when: ansible_os_family == "RedHat"
        notify: Reiniciar servidor

    - name: Actualizar paquetes en Debian
      apt:
        name: "*"
        state: latest
        update_cache: yes
        when: ansible_os_family == "Debian"
        notify: Reiniciar servidor

  handlers:

    - name: Reiniciar servidor
      reboot:
```

Tomcat.yml

```
---
- hosts: linux
  remote_user: ansible
  become: yes
  become_method: sudo
  vars:
    migruo: "tomcat"

  tasks:

    - name: Creamos el grupo tomcat
      group:
        name: "{{ migruo }}"
        state: present

    - name: Creamos el usuario tomcat
      user:
        name: tomcat
        comment: Tomcat user
        shell: /bin/bash
        group: tomcat
        append: yes
        password: "{{ 'Password.01' | password_hash('sha512') }}"

    - name: Instalamos tar en RedHat
      yum:
        name: tar
        state: latest
      when: ansible_os_family == "RedHat"

    - name: Instalamos tar en Debian
      apt:
        name: tar
        state: latest
      when: ansible_os_family == "Debian"

    - name: Instalar java en RedHat
      yum:
        name: java-1.8.0-openjdk
        state: present
      when: ansible_os_family == "RedHat"

    - name: Instalar java en Debian
      apt:
        name: openjdk-11-jdk
        state: present
      when: ansible_os_family == "Debian"

    - name: Descargar Tomcat
      get_url:
        url: https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.58/bin/apache-tomcat-9.0.58.tar.gz
        dest: /opt
```

- name: Descomprimir tomcat
unarchive:
 src: "/opt/apache-tomcat-9.0.58.tar.gz"
 dest: /opt
 remote_src: yes
- name: Cambiar nombre de carpeta Tomcat
command: mv /opt/apache-tomcat-9.0.58 /opt/tomcat
- name: Cambiar propietario de carpeta Tomcat
file:
 path: /opt/tomcat
 recurse: yes
 owner: tomcat
 group: tomcat
 mode: '777'
- name: Copiar tomcat.service para RedHat
copy: src=/home/ansible/git_workspace/TallerLinux/material/tomcat_redhat/tomcat.service
dest=/etc/systemd/system/
- name: Copiar todo.war
copy: src=/home/ansible/git_workspace/TallerLinux/material/tomcat_redhat/todo.war
dest=/opt/tomcat/webapps
- name: Copiar tomcat.service para Debian
copy: src=/home/ansible/git_workspace/TallerLinux/material/tomcat_debian/tomcat.service
dest=/etc/systemd/system/
 when: ansible_os_family == "Debian"
- name: Reiniciando demonio
systemd:
 daemon_reload: yes
- name: Inicializando servicio tomcat.service
systemd:
 state: restarted
 name: tomcat.service
- name: Establecer servicio Tomcat
service:
 name: tomcat.service
 enabled: yes
- name: Reiniciando
reboot:

Mysql.yml

```
- hosts: linux
  remote_user: ansible
  become: yes
  become_method: sudo

  tasks:

    - name: Instalar PyMySQL en RedHat
      yum:
        name: python3-PyMySQL
        state: latest
      when: ansible_os_family == "RedHat"

    - name: Instalar PyMySQL en Debian
      apt:
        name: python3-pymysql
        state: latest
      when: ansible_os_family == "Debian"

    - name: Instalar MySQL en RedHat
      yum:
        name: mysql-server
        state: present
      when: ansible_os_family == "RedHat"

    - name: Instalar MySQL en Debian
      apt:
        name: mysql-server
        state: present
      when: ansible_os_family == "Debian"

    - name: Reiniciar demonio
      systemd:
        daemon_reload: yes

    - name: Inicializar servicio mysqld en RedHat
      service:
        name: mysqld
        state: started
      when: ansible_os_family == "RedHat"

    - name: Inicializar servicio mysql en Debian
      service:
        name: mysql
        state: started
      when: ansible_os_family == "Debian"

    - name: Establecer servicio en RedHat
      service:
        name: mysqld
        enabled: yes
```

- ```
when: ansible_os_family == "RedHat"
```
- name: Establecer servicio en Debian  
service:  
  name: mysql  
  enabled: yes  
when: ansible\_os\_family == "Debian"
  - name: Creando usuario de base de datos para Debian  
mysql\_user:  
  login\_unix\_socket: /var/run/mysqld/mysqld.sock  
  name: todo  
  password: prueba2022  
  priv: ' \*.\*:ALL'  
  state: present  
when: ansible\_os\_family == "Debian"
  - name: Creando usuario de base de datos para RedHat  
mysql\_user:  
  name: todo  
  password: prueba2022  
  priv: ' \*.\*:ALL'  
  state: present  
when: ansible\_os\_family == "RedHat"
  - name: Creando Base de Datos para Debian  
mysql\_db:  
  login\_unix\_socket: /var/run/mysqld/mysqld.sock  
  name: todo  
  state: present  
when: ansible\_os\_family == "Debian"
  - name: Creando Base de Datos para RedHat  
mysql\_db:  
  name: todo  
  state: present  
when: ansible\_os\_family == "RedHat"
  - name: Moviendo script de base  
copy: src=/home/ansible/git\_workspace/TallerLinux/material/base/tablas.sql dest=/opt/tomcat
  - name: Creando tablas en la base para Debian  
mysql\_db:  
  login\_unix\_socket: /var/run/mysqld/mysqld.sock  
  state: import  
  name: all  
  target: /opt/tomcat/tablas.sql  
when: ansible\_os\_family == "Debian"
  - name: Creando tablas en la base para RedHat  
mysql\_db:  
  state: import  
  name: all
-

---

```
target: /opt/tomcat/tablas.sql
when: ansible_os_family == "RedHat"
```

- name: Creando carpeta para JDBC  
file:
  - path: /opt/config
  - state: directory
  - mode: 777
- name: Creando JDBC  
copy: src=/home/ansible/git\_workspace/TallerLinux/material/base/app.properties  
dest=/opt/config
- name: Reiniciando  
reboot:

## Apache.yml

```

- hosts: linux
 remote_user: ansible
 become: yes
 become_method: sudo

 tasks:

 - name: Instalamos el paquete de apache en Red Hat
 dnf:
 name: httpd
 state: present
 when: ansible_os_family == "RedHat"

 - name: Instalamos el paquete de apache en Debian
 apt:
 name: apache2
 state: present
 when: ansible_os_family == "Debian"

 - name: Inicializamos el servicio en Red Hat
 systemd:
 state: started
 name: httpd
 when: ansible_os_family == "RedHat"

 - name: Habilitamos apache en Red Hat
 systemd:
 name: httpd
 enabled: yes
 masked: no
 when: ansible_os_family == "RedHat"

 - name: Habilitamos el puerto 80/tcp en el firewall en Red Hat
 firewallld:
 port: 80/tcp
```

---

---

```
permanent: yes
state: enabled
when: ansible_os_family == "RedHat"

- name: Habilitamos el puerto 80/udp en el firewall en Red Hat
firewalld:
 port: 80/udp
 permanent: yes
 state: enabled
 when: ansible_os_family == "RedHat"

- name: Recargamos el servicio firewalld en Red Hat
systemd:
 name: firewalld
 state: reloaded
 when: ansible_os_family == "RedHat"

- name: Copiamos el archivo de virtualhost en Red Hat
copy: src=/home/ansible/git_workspace/TallerLinux/material/apache_redhat/vhost80.conf
dest=/etc/httpd/conf.d/
when: ansible_os_family == "RedHat"

- name: Hacemos que selinux habilite las conexiones proxy en Red Hat
command: /usr/sbin/setsebool -P httpd_can_network_connect 1
when: ansible_os_family == "RedHat"

- name: Cargamos el modulo proxy para Debian
apache2_module:
 state: present
 name: proxy
 when: ansible_os_family == "Debian"

- name: Cargamos el modulo proxy_http para Debian
apache2_module:
 state: present
 name: proxy_http
 when: ansible_os_family == "Debian"

- name: Cargamos el modulo proxy_balancer para Debian
apache2_module:
 state: present
 name: proxy_balancer
 when: ansible_os_family == "Debian"

- name: Cargamos el modulo lbmethod_byrequests para Debian
apache2_module:
 state: present
 name: lbmethod_byrequests
 when: ansible_os_family == "Debian"

- name: Copiamos el archivo de virtualhost en Debian
copy:
 src: /home/ansible/git_workspace/TallerLinux/material/apache_debian/000-default.conf
```

---

```
dest: /etc/apache2/sites-available/000-default.conf
force: yes
when: ansible_os_family == "Debian"
```

```
- name: Reiniciamos el servicio apache en Debian
 systemd:
```

```
 state: restarted
```

```
 name: apache2
```

```
 when: ansible_os_family == "Debian"
```

```
- name: Recargamos el servicio apache en Red Hat
 systemd:
```

```
 state: reloaded
```

```
 name: httpd
```

```
 when: ansible_os_family == "RedHat"
```

## **Main.yml**

```
- include: actualizador.yml
- include: tomcat.yml
- include: apache.yml
- include: mysql.yml
```

## **Materiales**

Archivos importantes para la funcionalidad del Sistema

### **Apache Debian:**

```
<VirtualHost *:80>
 ProxyPreserveHost On
 ProxyPass / http://127.0.0.1:8080/
 ProxyPassReverse / http://127.0.0.1:8080/
</VirtualHost>
```

### **Apache RedHat:**

```
<VirtualHost *:80>
 ServerName localhost
 ProxyPreserveHost On
 ProxyPass / http://localhost:8080/
 ProxyPassReverse / http://localhost:8080/
</VirtualHost>
```

### **App.properties**

```
tipoDB=mysql
jdbcURL=jdbc:mysql://localhost:3306/todo
jdbcUsername=todo
jdbcPassword=prueba2022
```

---

## SQL script

USE todo;

```
CREATE TABLE users(
id int(3) NOT NULL AUTO_INCREMENT,
first_name varchar(20) DEFAULT NULL,
last_name varchar(20) DEFAULT NULL,
username varchar(250) DEFAULT NULL,
password varchar(20) DEFAULT NULL,
PRIMARY KEY (id)) ENGINE=InnoDB;
```

```
CREATE TABLE todos(
id bigint(20) NOT NULL AUTO_INCREMENT,
description varchar(255) DEFAULT NULL,
is_dne bit(1) NOT NULL,
target_date datetime(6) DEFAULT NULL,
username varchar(255) DEFAULT NULL,
PRIMARY KEY (id)) ENGINE=InnoDB AUTO_INCREMENT=8;
```

## Tomcat Service para Debian

[Unit]

Description=Tomcat webs servlet container

After=network.target

[Service]

Type=forking

User=tomcat

Group=tomcat

Environment=JAVA\_HOME=/usr/lib/jvm/openjdk-11

Environment=CATALINA\_BASE=/opt/tomcat

Environment=CATALINA\_HOME=/opt/tomcat

Environment=CATALINA\_TMPDIR=/opt/tomcat/temp

Environment=CLASSPATH=/opt/tomcat/bin/bootstrap.jar:/opt/tocat/bin/tomcat-juli.jar

Environment=JRE\_HOME=/usr

Environment=CATALINA\_OPTS=

ExecStart=/opt/tomcat/bin/startup.sh

ExecStop=/opt/tomcat/bin/shutdown.sh

[Install]

WantedBy=multi-user.target

## Tomcat Service para RedHat

[Unit]

Description=Tomcat webs servlet container

After=network.target

[Service]

Type=forking

User=tomcat

Group=tomcat

Environment="JAVA\_HOME=/usr/lib/jvm/jre"

Environment="JAVA\_OPTS=-Djava.awt.headless=true -Djava.security.edg=file:/dev/./urandom"

Environment="CATALINA\_BASE=/opt/tomcat"

Environment="CATALINA\_HOME=/opt/tomcat"

Environment="CATALINA\_PID=/opt/tomcat/temp/tomcat.pid"

Environment="CATALINA\_OPTS=-Xms521M -Xms1024M -server -XX:+UseParallelGC"

ExecStart=/opt/tomcat/bin/startup.sh

ExecStop=/opt/tomcat/bin/shutdown.sh

[Install]

WantedBy=multi-user.target

## Programa funcionando

The image displays two screenshots of a web application running in a browser. The browser's address bar shows the URL `192.168.0.6/todo/login` for the top screenshot and `192.168.0.5/todo/login` for the bottom screenshot. The application has a red header bar with the text 'Todo App' and 'Todos' on the left, and a 'Logout' link on the right. The main content area is titled 'List of Todos' and features a green 'Add Todo' button. Below the button is a table with the following structure:

| Title | Target Date | Todo Status | Actions |
|-------|-------------|-------------|---------|
|       |             |             |         |