

# OBLIGATORIO

*Taller Instalación Servidores Linux*



A N S I B L E

**Alumnos:** Marcio Fontes N 201156  
Pablo Gonzalez N 238238

## Letra del obligatorio

### A) 30%

Instalar un servidor Rocky y otro Ubuntu con 1 GB de memoria RAM y 25 GB de Disco  
Configure la instalación con el siguiente esquema de particionamiento

Una partición de 1G para /boot

usando LVM: un volumen lógico de 5GB para /, 5GB para /var, 1GB para /home, 5GB /opt  
y 2 GB de Swap.

Debe tener 2 interfaces de red, 1 conectada a NAT y la otra a una red Interna o “Solo  
Anfitrión” que le permita conectarse al equipo bastión con Ansible.

Agregue un usuario ansible, dele permisos con SUDO sin contraseña. Desde el equipo  
bastión o controlador, copie la clave pública para poder conectarse al servidor.

### B) 60 % Tareas a realizar con Ansible

Debe crear los playbooks necesarios para la instalación y configuración de un servidor  
Tomcat, un servidor de base de datos MySQL o Mariadb incluyendo la creación de la base  
de datos y el usuario para la aplicación, y un servidor web apache que oficie de proxy  
reverso.

Incluir todas las dependencias que sean necesarias. La aplicación debe ser accesible  
solamente a través del servidor apache.

El servidor tomcat debe quedar instalado en el /opt, debe ejecutar con el usuario tomcat y  
grupo tomcat. Debe iniciar mediante un servicio.

El playbook debe poder aplicarse tanto a un servidor con Rocky (tipo rhel), y un servidor  
con Ubuntu (tipo debian)

Los playbooks deben ser versionados utilizando git y debe crearse un repositorio remoto  
en github

### C) 10% Documentación

Actualice el archivo README.md explicando los cambios realizados y como utilizar el rol.  
Debe documentar cada paso de las tareas realizadas mediante capturas de pantalla. En  
el caso del

playbook debe mostrar la ejecución correcta de todas las tareas.

## Parte A:

Para realizar el escenario que pide la letra vamos a necesitar contar con un equipo bastión con ansible instalado.

Con el comando (**sudo apt install ansible**) se instalan los paquetes y dependencias requeridas para poder utilizarlo.

Se repite el mismo procedimiento para git (**sudo apt install git**).

Luego se crearon dos máquinas virtuales, una corriendo Ubuntu Server 20.04 y la otra con Rocky 8.5.

Ambos servidores cuentan con dos interfaces de red, una configurada como NAT (para navegación a internet) y la otra definida como red Sólo Anfitrión (para la comunicación con el equipo bastión), y fueron instalados con las siguientes características:

- **1 GB de Memoria RAM**
- **25 GB de disco duro**

Al momento de hacer la instalación de los Sistemas Operativos se creó el siguiente esquema de particionamiento:

Se Configuró la instalación con el siguiente esquema de particionamiento:

Una partición de **1G para /boot**

LVM: un volumen lógico de **5GB para /**, **5GB para /var**, **1GB para /home**, **5GB /opt** y **2 GB de Swap**.

Los volúmenes lógicos fueron creados con LVM y pertenecen a un mismo grupo de volúmenes

```

[ansible@rocky ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   25G  0 disk
├─sda1       8:1    0    1G  0 part /boot
└─sda2       8:2    0   18G  0 part
   └─rl_rocky-root 253:0    0    5G  0 lvm  /
      └─rl_rocky-swap 253:1    0    2G  0 lvm  [SWAP]
         └─rl_rocky-home 253:2    0    1G  0 lvm  /home
            └─rl_rocky-var 253:3    0    5G  0 lvm  /var
               └─rl_rocky-opt 253:4    0    5G  0 lvm  /opt
sr0         11:0    1 1024M  0 rom

[ansible@rocky ~]$ sudo vgs
VG          #PV #LU #SN Attr   USize  UFree
rl_rocky     1   5   0 wz--n- 18.00g 4.00m

```

```
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          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
[ ubuntu-vg (new)                     LVM volume group                   23.996G ▶ ]
free space                             5.996G

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

USED DEVICES

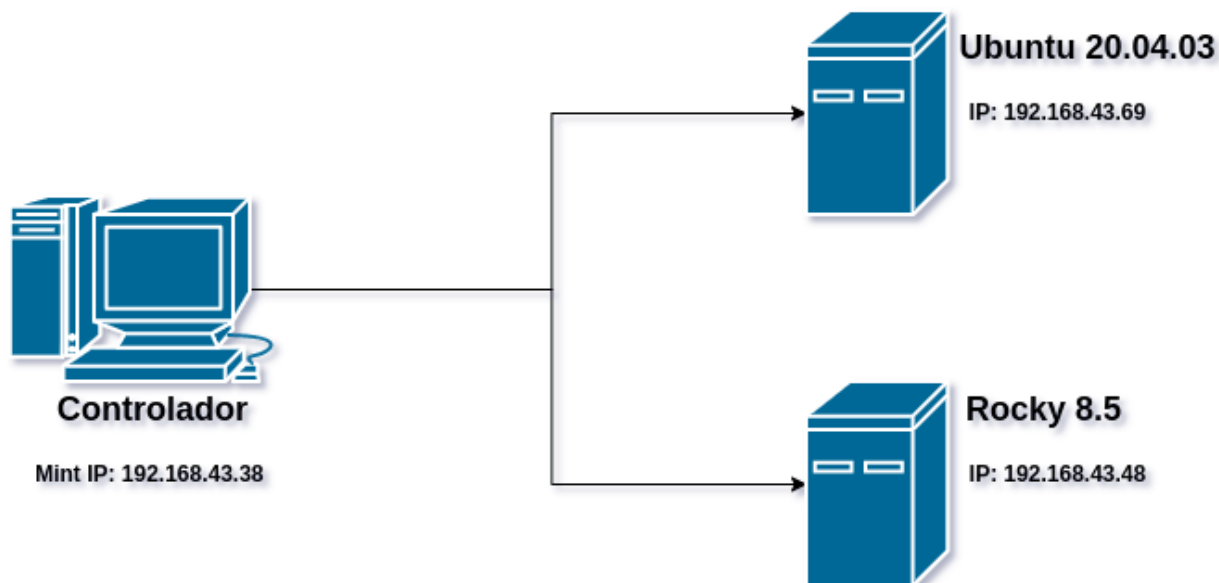
DEVICE                                TYPE                                SIZE
[ ubuntu-vg (new)                     LVM volume group                   23.996G ▶ ]
ubuntu-lv                             new, to be formatted as ext4, mounted at / 5.000G ▶
lv-var                                new, to be 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 ▶

[ Done ]
[ Reset ]
[ Back ]
```

```
ansible@ubuntu:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0                               7:0      0  55.4M  1 loop /snap/core18/2128
loop1                               7:1      0  55.5M  1 loop /snap/core18/2284
loop2                               7:2      0  61.9M  1 loop /snap/core20/1328
loop3                               7:3      0  70.3M  1 loop /snap/lxd/21029
loop4                               7:4      0  67.2M  1 loop /snap/lxd/21835
loop5                               7:5      0  43.6M  1 loop /snap/snapd/14978
sda                                 8:0      0   25G   0 disk
├─sda1                             8:1      0    1M   0 part
├─sda2                             8:2      0    1G   0 part /boot
└─sda3                             8:3      0   24G   0 part
   ├─ubuntu--vg-ubuntu--lv         253:0      0    5G   0 lvm /
   ├─ubuntu--vg-lv--var            253:1      0    5G   0 lvm /var
   ├─ubuntu--vg-lv--home           253:2      0    1G   0 lvm /home
   ├─ubuntu--vg-lv--opt            253:3      0    5G   0 lvm /opt
   └─ubuntu--vg-lv--Swap           253:4      0    2G   0 lvm [SWAP]
sr0                                 11:0     1 1024M   0 rom

ansible@ubuntu:~$ sudo vgs
VG          #PV #LV #SN Attr   VSize  VFree
ubuntu-vg   1   5   0 wz--n- <24.00g <6.00g
```

Escenario entre el controlador y los servidores:



### Configuración del servidor bastión (Controlador)

#### Servidor Controlador:

Chequeamos la versión de ansible instalada en el controlador

```
pcgonzalez@Mint:~$ ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/pcgonzalez/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.10 (default, Nov 26 2021, 20:14:08) [GCC 9.3.0]
```

Chequeamos la versión de git instalada en el controlador

```
pcgonzalez@Mint:~$ git --version
git version 2.25.1
pcgonzalez@Mint:~$ git config --list
user.name=Pablo Gonzalez
user.email=pablocesargg.aii@gmail.com
color.ui=true
color.status=auto
color.branch=auto
core.editor=vim
merge.tool=vimdiff
```

Chequeamos que tenga ssh instalado e iniciado

```
pcgonzalez@Mint:~$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-02-22 03:52:58 -03; 7h ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 928 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 965 (sshd)
     Tasks: 1 (limit: 14119)
    Memory: 2.5M
    CGroup: /system.slice/ssh.service
            └─965 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

feb 22 03:52:58 Mint systemd[1]: Starting OpenBSD Secure Shell server...
feb 22 03:52:58 Mint sshd[965]: Server listening on 0.0.0.0 port 22.
feb 22 03:52:58 Mint sshd[965]: Server listening on :: port 22.
feb 22 03:52:58 Mint systemd[1]: Started OpenBSD Secure Shell server.
```

Creamos el inventario en el servidor:

```
pcgonzalez@Mint:~$ sudo vim /etc/ansible/hosts
```

Creamos el grupo linux que se encontraran los dos servidores

```
[Rocky]
192.168.43.48 ansible_ssh_user=ansible

[debian]
192.168.43.69 ansible_ssh_user=ansible

[linux:children]
debian
Rocky

[all:vars]
ansible_python_interpreter=/usr/bin/python3
# Ex 1: Ungrouped hosts, specify before any group headers.
```

### Configuración de usuarios en los servidores

Usuario “ansible” en todas las VMs se crea el usuario “ansible”, se lo agrega al grupo con permisos de sudo.

**\$ usermod ansible -aG wheel ( redht) \$ usermod -aG sudo ansible ( debian)**

Se configura para que lo ejecute sin contraseña:

```
## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
ansible ALL=(ALL) NOPASSWD:ALL
```

También se guarda la misma clave pública en el perfil del git de cada usuario.

Una vez asegurada la conexión entre el bastión y los equipos a configurar, se copian las claves públicas hacia cada uno de los servidores, para poder establecer la conexión vía ssh.

### ssh-keygen

```
controlador@Mint:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/controlador/.ssh/id_rsa):
Created directory '/home/controlador/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/controlador/.ssh/id_rsa
Your public key has been saved in /home/controlador/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ccxLtHonoqLBMtSQhLXnOt3ErXxwMgY20yG36NmDdUK controlador@Mint
The key's randomart image is:
+---[RSA 3072]-----+
| ++.. o E .          |
| . .+ = + = .        |
| o O + + *           |
| . = X o = .         |
| o * X S + .         |
| * = X o o           |
| o o * .             |
| . o .               |
| .                   |
+----[SHA256]-----+
```

```
controlador@Mint:~$ ssh-copy-id ansible@rocky
```

```
controlador@Mint:~$ ssh ansible@ubuntu
```

En la siguiente captura vemos como iniciamos por ssh sin contraseña al servidor rocky

```
pcgonzalez@Mint:~$ ssh ansible@rocky
Last login: Mon Feb 21 18:09:29 2022 from 192.168.43.38
[ansible@Rocky ~]$
```

En la siguiente captura vemos como iniciamos por ssh sin contraseña el servidor ubuntu

```
pcgonzalez@Mint:~$ ssh ansible@ubuntu
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-100-generic x86_64)
```

De esta forma, se podrá acceder vía SSH a los servidores sin necesidad de indicar la contraseña del usuario ansible.

Podemos conectarnos por ssh mencionando el nombre de host o ip del mismo:

```
pcgonzalez@Mint:~$ ssh ansible@192.168.43.48
ssh: connect to host 192.168.43.48 port 22: No route to host
pcgonzalez@Mint:~$ ssh ansible@192.168.43.48
Last login: Tue Feb 22 12:14:27 2022
[ansible@Rocky ~]$
[ansible@Rocky ~]$
```

```
pcgonzalez@Mint:~$ ssh ansible@192.168.43.69
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-100-generic x86_64)

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



## Parte B:

### Creación y preparación del repositorio en el controlador:

Para poder trabajar de manera colaborativa con el código y contar con un sistema seguro de gestión de versiones, se trabajó con la herramienta git la misma fue previamente instalada y configurada en el equipo bastión ("controlador").

Para chequear la conexión entre el equipo bastión y los servidores realizamos el siguiente comando ad-hoc **ansible linux -m ping**

Ejecuta un módulo de ping en todos los hosts del archivo de inventario.

```
pcgonzalez@Mint:~$ ansible linux -m ping
192.168.43.48 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
192.168.43.69 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

Podemos verificar la conexión a cada uno por separado

```
pcgonzalez@Mint:~$ ansible Rocky -m ping
192.168.43.48 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

```
pcgonzalez@Mint:~$ ansible debian -m ping
192.168.43.69 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

Creamos un archivo inventario.ini en el mismo se van a encontrar los servidores separados según su distribución ubuntu perteneciente a [debian], y Rocky perteneciente a RedHat y creamos un grupo linux en el cual se encuentran las dos distribuciones de los sistemas operativos.

Para hacer el despliegue se puede definir en cuáles servidores se realizará la instalación de cada uno de los componentes o en ambos a la vez.

```
[debian]
ubuntu ansible_host=192.168.43.69

[Rocky]
rocky ansible_host=192.168.43.48

#[Mint]
#Mint 192.168.43.38

[linux:children]
debian
Rocky
```

En la carpeta Chek\_interface tenemos para verificar las interfaces de los servidores

```
pcgonzalez@Mint:~/PCG/Obligatorio_2022$ cd Chek_interface/
pcgonzalez@Mint:~/PCG/Obligatorio_2022/Chek_interface$ ls
interfaces.yml
pcgonzalez@Mint:~/PCG/Obligatorio_2022/Chek_interface$ ansible-playbook interfaces.yml

PLAY [linux] *****

TASK [Gathering Facts] *****
ok: [192.168.43.48]
ok: [192.168.43.69]

TASK [Ver interfaces de Red] *****
ok: [192.168.43.69] => {
  "msg": "Lista de interfaces de red: ['enp0s3', 'docker0', 'vethd0d5568', 'lo', 'enp0s8']"
}
ok: [192.168.43.48] => {
  "msg": "Lista de interfaces de red: ['lo', 'enp0s8', 'enp0s3']"
}

PLAY RECAP *****
192.168.43.48      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.43.69      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

pcgonzalez@Mint:~/PCG/Obligatorio_2022/Chek_interface$
```

Vamos a ejecutar un playbook de actualización a los servidores:

Dentro de nuestro repositorio Obligatorio\_2022 vamos a tener un playbook para realizar las actualizaciones iniciales en los servidores.

Lo ejecutamos: ansible-playbook update.yml

```
pcgonzalez@Mint:~/PCG/Obligatorio_2022$ ansible-playbook update.yml

PLAY [linux] *****

TASK [Gathering Facts] *****
ok: [192.168.43.48]
ok: [192.168.43.69]

TASK [Actualizaar paquetes en RedHat] *****
skipping: [192.168.43.69]
ok: [192.168.43.48]

TASK [Actualizar paquetes en Debian] *****
skipping: [192.168.43.48]
ok: [192.168.43.69]

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

Se crea un playbook con el nombre main.yml

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

# Se incluye archivo de variables en el playbook
vars_files:
- ./vars.yml

tasks:

- name: Instalar paquetes en Red Hat
  dnf:
    name: "{{ item }}"
    state: present
    loop: "{{ paquetes_rh }}"
    when: ansible_os_family == "Rocky"

- name: Instalar paquetes en Debian
  apt:
    name: "{{ item }}"
    state: present
    update_cache: yes
    loop: "{{ paquetes_deb }}"
    when: ansible_os_family == "Debian"
```

Se crea paquetes\_rh que tendra todo para instalar en RedHat

Se crea paquetes\_deb que tendra todo para instalar en Debian

```
paquetes_rh:
- httpd
- python3
- epel-release
- java-17-openjdk
- mariadb-server
- mariadb
- tar

paquetes_deb:
- python3-pip
- apache2
- openjdk-17-jdk
- mariadb-server
- mariadb-client

tomcat_url: "https://dlcdn.apache.org/tomcat/tomcat-10/v10.0.16/bin/apache-tomcat-10.0.16.tar.gz"
tomcat_dir: "/opt/apache-tomcat-10.0.16"
```

Se ejecuta el main.yml

Comienza a instalar los paquetes en los servidores

```
pcgonzalez@Mint:~/PCG/Obligatorio_2022$ ansible-playbook main.yml

PLAY [linux] *****

TASK [Gathering Facts] *****
ok: [192.168.43.48]
ok: [192.168.43.69]

TASK [Instalar paquetes RedHat en servidor Rocky] *****
skipping: [192.168.43.69] => (item=httpd)
skipping: [192.168.43.69] => (item=python3)
skipping: [192.168.43.69] => (item=epel-release)
skipping: [192.168.43.69] => (item=java-17-openjdk)
skipping: [192.168.43.69] => (item=mariadb-server)
skipping: [192.168.43.69] => (item=mariadb)
skipping: [192.168.43.69] => (item=tar)
ok: [192.168.43.48] => (item=httpd)
ok: [192.168.43.48] => (item=python3)
ok: [192.168.43.48] => (item=epel-release)
ok: [192.168.43.48] => (item=java-17-openjdk)
ok: [192.168.43.48] => (item=mariadb-server)
ok: [192.168.43.48] => (item=mariadb)
ok: [192.168.43.48] => (item=tar)

TASK [Instalar paquetes Debian en servidor ubuntu] *****
skipping: [192.168.43.48] => (item=python3-pip)
skipping: [192.168.43.48] => (item=apache2)
skipping: [192.168.43.48] => (item=openjdk-17-jdk)
skipping: [192.168.43.48] => (item=mariadb-server)
skipping: [192.168.43.48] => (item=mariadb-client)
ok: [192.168.43.69] => (item=python3-pip)
ok: [192.168.43.69] => (item=apache2)
ok: [192.168.43.69] => (item=openjdk-17-jdk)
ok: [192.168.43.69] => (item=mariadb-server)
ok: [192.168.43.69] => (item=mariadb-client)
```

```
- name: Crear grupo tomcat
  group:
    name: tomcat
    state: present

- name: Crear usuario tomcat
  user:
    name: tomcat
    group: tomcat
    comment: Tomcat server user
    state: present

- name: Descargar e instalar tomcat
  unarchive:
    src: "{{ tomcat_url }}"
    dest: /opt
    remote_src: yes

- name: Cambiar dueño del directorio Tomcat
  file:
    path: "{{ tomcat_dir }}"
    owner: tomcat
    group: tomcat
    recurse: yes

- name: Instalar servicio Tomcat
  template:
    src: tomcat.service.j2
    dest: /etc/systemd/system/tomcat.service
  notify: Reiniciar Tomcat
```

```
TASK [Crear grupo tomcat] *****
ok: [192.168.43.69]
ok: [192.168.43.48]

TASK [Crear usuario tomcat] *****
ok: [192.168.43.69]
ok: [192.168.43.48]

TASK [Descargar e instalar tomcat] *****
changed: [192.168.43.69]
changed: [192.168.43.48]

TASK [Cambiar dueño del directorio Tomcat] *****
changed: [192.168.43.69]
changed: [192.168.43.48]

TASK [Instalar servicio Tomcat] *****
ok: [192.168.43.69]
ok: [192.168.43.48]

TASK [Configuro reverse proxy] *****
skipping: [192.168.43.69]
ok: [192.168.43.48]
```

```
- name: Descargar la aplicacion para url
  get_url:
    url: https://aulas.ort.edu.uy/pluginfile.php/642237/mod\_page/content/3/todo.war
    dest: "{{ tomcat_dir }}/webapps/"
    url_username: "{{ ort_user }}"
    url_password: "{{ ort_password }}"

- name: Configuro reverse proxy
  copy:
    src: proxy.conf
    dest: /etc/httpd/conf.d/
    owner: root
    group: root
  when: ansible_os_family == "Rocky"

- name: Apache iniciado y habilitado
  systemd:
    name: httpd
    state: started
    enabled: yes
  when: ansible_os_family == "Rocky"

- name: Abrir puerto 80
  firewallld:
    service: httpd
    permanent: yes
    state: enabled
  when: ansible_os_family == "Rocky"

handlers:

- name: Reiniciar Tomcat
  systemd:
    daemon_reload: yes
    name: tomcat
    state: restarted
    enabled: yes
```

## Ejecución:

```
TASK [Abrir puerto 80] *****
skipping: [192.168.43.69]
fatal: [192.168.43.48]: FAILED! => {"changed": false, "msg": "ERROR: Exception caught: org.fedoraproject.FirewallD1.Exception: INVALID_SERVICE: 'httpd' not among existing services Permanent operation, Services are defined by port/tcp relationship and named as they are in /etc/services (on most systems)"}

TASK [Abrir puerto 80] *****
skipping: [192.168.43.69]

TASK [Apache iniciado y habilitado] *****
skipping: [192.168.43.69]

TASK [Tomcat está habilitado e iniciado] *****
fatal: [192.168.43.69]: FAILED! => {"changed": false, "msg": "Error loading unit file 'tomcat': org.freedesktop.systemd1.BadUnitSetting \"Unit tomcat.service has a bad unit file setting.\""}

PLAY RECAP *****
192.168.43.48      : ok=8    changed=2    unreachable=0    failed=1    skipped=1    rescued=0    ignored=0
192.168.43.69      : ok=7    changed=2    unreachable=0    failed=1    skipped=5    rescued=0    ignored=0

pcgonzalez@Mint:~/PCG/Obligatorio_2022$
```

## ERRORES:

Nos falta detectar los errores del puerto 80 de 192.168.43.48

Se descargo e Instalo tomcat pero nos da error al iniciarlo

### Playbooks *main.yml* y *vars.yml*:

Se crean los siguientes playbooks como punto de partida para actualizar e instalar el conjunto de paquetes requeridos.

### Fragmento del *main.yml*:

```
-----  
  
---  
- hosts: linux  
  remote_user: ansible  
  become: yes  
  become_method: sudo  
  
#Incluyo archivo de variables en el playbook  
vars_files:  
- ./vars.yml  
  
tasks:  
  
- name: Instalar paquetes RedHat en servidor Rocky  
  dnf:  
    name: "{{ item }}"  
    state: present  
    loop: "{{ paquetes_rh }}"  
    when: ansible_os_family == "Rocky"  
.....  
  
- name: Instalar servicio Tomcat  
  template:  
    src: tomcat.service.j2  
    dest: /etc/systemd/system/tomcat.service  
    notify: Reiniciar Tomcat  
  
- name: Tomcat está habilitado e iniciado  
  systemd:  
    name: tomcat  
    state: started  
    enabled: yes  
    daemon_reload: yes
```



- name: Descargar la aplicacion para url  
get\_url:  
url: https://aulas.ort.edu.uy/pluginfile.php/642237/mod\_page/content/3/todo.war  
dest: "{{ tomcat\_dir }}/webapps/"  
url\_username: "{{ ort\_user }}"  
url\_password: "{{ ort\_password }}"
  
- name: Configuro reverse proxy  
copy:  
src: proxy.conf  
dest: /etc/httpd/conf.d/  
owner: root  
group: root  
when: ansible\_os\_family == "Rocky"
  
- name: Apache iniciado y habilitado  
systemd:  
name: httpd  
state: started  
enabled: yes  
when: ansible\_os\_family == "Rocky"
  
- name: Abrir puerto 80  
firewalld:  
service: httpd  
permanent: yes  
state: enabled  
when: ansible\_os\_family == "Rocky"
  
- handlers:
  
- name: Reiniciar Tomcat  
systemd:  
daemon\_reload: yes  
name: tomcat  
state: restarted  
enabled: yes