

# Universidad ORT Uruguay

## Facultad de Ingeniería



Analista en Infraestructura Informática

Obligatorio

Taller de Servidores Linux

Docente: Sebastián Orrego

Andrés Tarallo

Damián Álvarez – 193887

Sebastián Gentieu - 259481

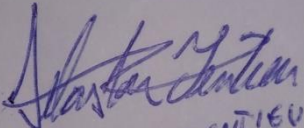
Edward Wrzosek - 146643

## Tabla de contenidos

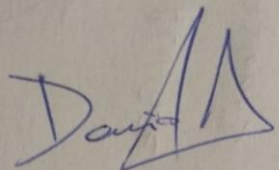
<b>Introducción .....</b>	<b>4</b>
<b>Configuración de servidores .....</b>	<b>5</b>
<b>Archivos de configuración de Ansible.....</b>	<b>9</b>
<b>Corrección de playbooks .....</b>	<b>10</b>
<b>Playbook dependencias comunes .....</b>	<b>10</b>
<b>Playbook base de datos.....</b>	<b>12</b>
<b>Playbook servidor web .....</b>	<b>15</b>
<b>Ejecución de playbooks .....</b>	<b>18</b>

## Declaración de autoría

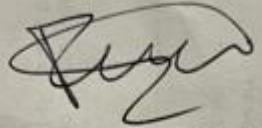
Damián Álvarez, Sebastián Gentieu y Edward Wrzosek, estudiantes de la carrera Analista en Infraestructura Informática en la Universidad ORT Uruguay, declaramos que la responsabilidad del contenido del presente trabajo nos corresponde exclusivamente y su investigación es totalmente original.



SEBASTIAN GENTIEU  
259482



DAMIÁN ÁLVAREZ  
193887



EDWARD WRZOSEK  
4.421.999-6  
Nº ESTUDIANTE 146643

## Introducción

Se precisa la creación de dos servidores, uno con la distribución CentOS y otro con la distribución Ubuntu, para poder utilizarlos mediante un equipo controlador y la plataforma Ansible.

Este equipo controlador deberá contar con la plataforma instalada y un usuario para ejecutar los playbooks de Ansible.

Los dos servidores donde se instalarán los playbooks deberán contar con 1 GB de memoria RAM y 15 GB de espacio de disco. En cuanto al disco de los servidores, este deberá contar con determinadas particiones. Una partición primaria de 1 GB para /boot y cuatro volúmenes lógicos para /, /var, /home y Swap, con diversos tamaños.

Cada servidor contará con dos interfaces de red, una estaría conectada a NAT y la siguiente, a una red interna.

Además de esto, en cada servidor se deberá de agregar un usuario con el nombre “ansible”. Este usuario se configurará para no requerir contraseña en caso de escalamiento de privilegios al momento de instalar paquetes.

Es necesario realizar un fork de un repositorio brindado para obtener el stack a configurar en los servidores. A este stack se le deberá de agregar archivos de configuración, tales como el archivo “ansible.cfg” y el archivo de inventario.

Como estos playbooks se encontraban adaptados tan solo para distribuciones CentOS, se precisará adaptarlos para la distribución de Ubuntu. Además, se tendrá que corregir la indentación de los playbooks para utilizarlos en la versión más reciente de Ansible.

Para finalizar, se actualizará el archivo “README.md”, perteneciente al stack. En este archivo se agregarán los cambios realizados y la forma de utilizar los distintos roles del stack.

La configuración de los servidores y la adaptación de los playbooks deberá ser documentadas mediante capturas de pantalla. El funcionamiento de los playbooks en los servidores también se deberá de enseñar.

Por último, toda la documentación realizada se deberá de agregar al repositorio.

## Configuración de servidores

Para la realización del presente trabajo se configuraron dos servidores. Uno de ellos con la distribución CentOS, para cumplir la función de servidor base de datos. El segundo servidor se configuró con una distribución Ubuntu para oficiar de servidor web.

A cada uno de estos servidores se le proporcionó con 1 GB de memoria RAM y 15 GB de espacio de disco.

Cada disco se particionó de la siguiente forma:

Punto de montaje	Tamaño
/boot - Partición primaria	1 GB
/ - Volumen lógico	5 GB
/var - Volumen lógico	4 GB
/home - Volumen lógico	3 GB
Swap - Volumen lógico	2 GB

Particiones necesarias para cada servidor

Todas las particiones se realizaron al momento de la instalación de la distribución.

```
FILE SYSTEM SUMMARY
MOUNT POINT    SIZE    TYPE    DEVICE TYPE
[ /             5.000G  new xfs  new LVM logical volume ▶ ]
[ /boot         1.000G  new xfs  new partition of local disk ▶ ]
[ /boot/efi     512.000M  new fat32  new partition of local disk ▶ ]
[ /home         3.000G  new xfs  new LVM logical volume ▶ ]
[ /var          4.000G  new xfs  new LVM logical volume ▶ ]
[ SWAP          2.000G  new swap  new LVM logical volume ▶ ]

AVAILABLE DEVICES
DEVICE                                TYPE                                SIZE
[ vg0 (new)                            LVM volume group                    19.996G ▶ ]
  free space                            5.996G
[ 36002248051711ee36ab658a52cada51b  local disk                          127.000G ▶ ]
  free space                            105.498G
[ Create software RAID (md) ▶ ]
[ Create volume group (LVM) ▶ ]

USED DEVICES
DEVICE                                TYPE                                SIZE
[ vg0 (new)                            LVM volume group                    19.996G ▶ ]
  lv-0                                new, to be formatted as xfs, mounted at / 5.000G ▶ ]
  lv-1                                new, to be formatted as xfs, mounted at /var 4.000G ▶ ]
  lv-2                                new, to be formatted as xfs, mounted at /home 3.000G ▶ ]
  lv-3                                new, to be formatted as swap 2.000G ▶ ]
[ 36002248051711ee36ab658a52cada51b  local disk                          127.000G ▶ ]
  partition 1  new, primary ESP, to be formatted as fat32, mounted at /boot/efi 512.000M ▶ ]
  partition 2  new, PV of LVM volume group vg0 20.000G ▶ ]
  partition 3  new, to be formatted as xfs, mounted at /boot 1.000G ▶ ]
```

Particiones creadas durante la instalación de Ubuntu

▼ New CentOS Linux 8 Installation

DATA	
/home cl-home	2.93 GiB >
SYSTEM	
/	4.88 GiB
cl-root	
/var cl-var	3.91 GiB
/boot/efi sda1	600 MiB
/boot sda2	1024 MiB
swap cl-swap	1.95 GiB

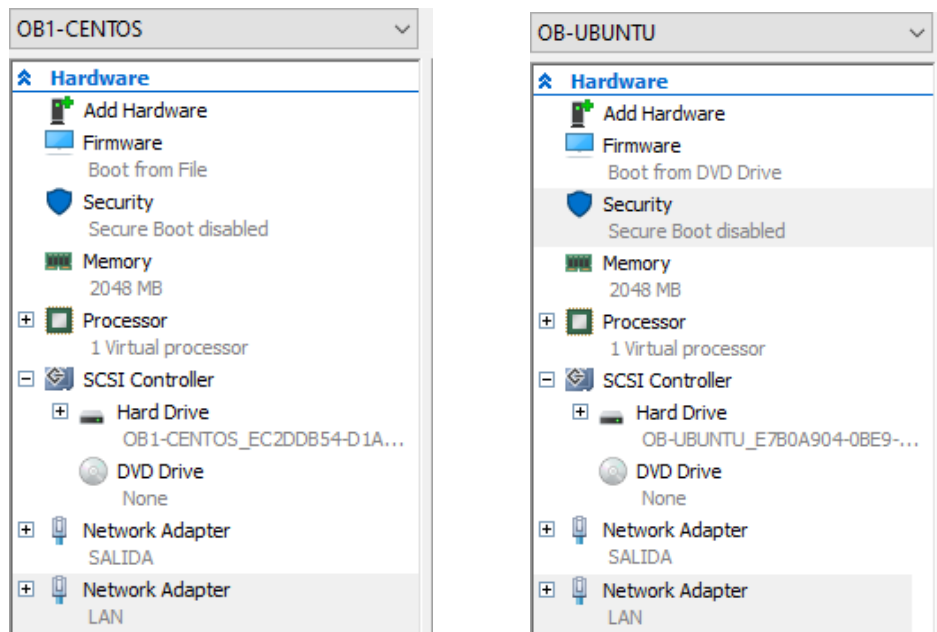
Particiones creadas durante la instalación de CentOS

```
[ansible@obcentos ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0 127G  0 disk
├─sda1       8:1    0  600M  0 part /boot/efi
├─sda2       8:2    0    1G   0 part /boot
└─sda3       8:3    0 13.7G  0 part
   ├─cl-root 253:0    0  4.9G  0 lvm  /
   ├─cl-swap 253:1    0    2G   0 lvm  [SWAP]
   ├─cl-var  253:2    0  3.9G  0 lvm  /var
   └─cl-home 253:3    0    3G   0 lvm  /home
sr0         11:0    1 1024M  0 rom
```

```
ansible@obubuntu:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0        7:0    0  69.9M  1 loop /snap/lxd/19188
loop1        7:1    0  31.1M  1 loop /snap/snapd/10707
loop2        7:2    0  55.4M  1 loop /snap/core18/1944
loop3        7:3    0  55.5M  1 loop /snap/core18/2074
loop4        7:4    0  32.3M  1 loop /snap/snapd/12704
loop5        7:5    0  70.3M  1 loop /snap/lxd/21029
sda          8:0    0 127G  0 disk
├─sda1       8:1    0  512M  0 part /boot/efi
├─sda2       8:2    0    20G  0 part
│   ├─vg0-lv--0 253:0    0    5G   0 lvm  /
│   ├─vg0-lv--1 253:1    0    4G   0 lvm  /var
│   ├─vg0-lv--2 253:2    0    3G   0 lvm  /home
│   └─vg0-lv--3 253:3    0    2G   0 lvm  [SWAP]
└─sda3       8:3    0    1G   0 part /boot
sr0         11:0    1 1024M  0 rom
```

Particiones configuradas en ambos servidores mediante el comando "lsblk"

Como siguiente requerimiento, cada servidor cuenta con dos interfaces de red. Una estaría conectada a NATy la siguiente, conectada a una red Interna.



Configuración de interfaces de red

Con el objetivo de obtener un usuario para que obtenga permisos “sudo” sin contraseña, se editó el archivo “sudoers” utilizando el comando “visudo”. Allí se agregó al usuario al cual obtendrá permisos “sudo” sin requerir contraseña.

```
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

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

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

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

#include_dir /etc/sudoers.d
ansible    ALL=(ALL) NOPASSWD:ALL
```

Edición de archivo “sudoers”

Como requerimiento final para el funcionamiento de Ansible, se debió generar las claves públicas y privadas para el usuario “ansible” en el controlador.

```
sg@DENDROBIUM:/mnt/c/Users/S$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/sg/.ssh/id_rsa):
/home/sg/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/sg/.ssh/id_rsa
Your public key has been saved in /home/sg/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:8i0JdswDLPXmutnI0QmP/cG0+HLSdGQvdLGvn0j5yy4 sg@DENDROBIUM
The key's randomart image is:
+---[RSA 3072]-----+
|
| .
| o .
| o o o
| . o + o
| . o S . + o
| = @ = . o o
| o @ Oo+ . +
| . o Oo+o . E=
| = .+o .oBo
+---[SHA256]-----+
```

Generación de llave SSH

Una vez generadas las claves, se copiaron a los servidores CentOS y Ubuntu mediante el comando “ssh-copy-id”.

```
sg@DENDROBIUM:/mnt/c/Users/S$ ssh-copy-id ansible@192.168.0.104
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/sg/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are al
ready installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to ins
tall the new keys
ansible@192.168.0.104's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ansible@192.168.0.104'"
and check to make sure that only the key(s) you wanted were added.

sg@DENDROBIUM:/mnt/c/Users/S$
```

Copia de llave a los servidores de CentOS y Ubuntu

Para finalizar la configuración, se utilizaron los comandos “ssh-agent” y “ssh-add” para que así no sea necesario proporcionar una contraseña cada vez que se utilice la conexión SSH de Ansible.



## Archivos de configuración de Ansible

Para realizar el fork del repositorio proporcionado, se utilizó el comando “git clone”. De esta manera se obtuvo el stack LAMP, el cual contiene todos los archivos necesarios para lograr la implementación del stack.

Para lograr el funcionamiento de Ansible, se es necesario un archivo de configuración denominado “ansible.cfg”, y un archivo de inventario, denominado “hosts”.

```
1  [[controlador]]
2  controller ansible_connection=local
3
4  [dbservers]
5  obcentos ansible_host=192.168.0.104 ansible_user=ansible
6  [webservers]
7  obubuntu ansible_host=192.168.0.107 ansible_user=ansible
8
9
```

Contenido de archivo de inventario

```
# Example config file for ansible -- https://ansible.com/
# =====

# Nearly all parameters can be overridden in ansible-playbook
# or with command line flags. Ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory, or /etc/ansible/ansible.cfg, whichever it
# finds first

# For a full list of available options, run ansible-config list or see the
# documentation: https://docs.ansible.com/ansible/latest/reference\_appendices/config.html.

[defaults]
inventory      = ./hosts
#library       = ~/.ansible/plugins/modules:/usr/share/ansible/plugins/modules
#module_utils  = ~/.ansible/plugins/module_utils:/usr/share/ansible/plugins/module_utils
#remote_tmp    = ~/.ansible/tmp
#local_tmp     = ~/.ansible/tmp
#forks         = 5
#poll_interval = 0.001
#ask_pass      = False
#transport     = smart
```

Contenido de archivo de configuración de Ansible

## Corrección de playbooks

### Playbook dependencias comunes

En el playbook original se encontraron tareas escritas para versiones anteriores de Ansible, por lo cual, se corrigió la indentación de cada tarea encontrada.

Además de la corrección de la indentación, se agregó el módulo “hosts” para identificar a qué servidor se le ejecutaría el playbook.

El módulo “gather\_facts” es necesario para obtener toda la información sobre el servidor donde esté ejecutando el playbook. Esta información podrá ser utilizada en diversas tareas del playbook.

Fue necesario agregar el módulo “become” para lograr ejecutar las tareas con escalación de privilegios.

A diferencia del playbook original, se agregó una tarea “include\_vars”, para cargar un archivo de variables encontrado en la carpeta “vars”, dentro de la carpeta “tasks”. Los nombres de estos archivos de variables se definieron para que tengan el mismo de la distribución del servidor a instalar, recolectado mediante el módulo “gather\_facts”.

Para la segunda tarea, se especificó la instalación del paquete chrony para cada servidor. Como el paquete se encuentra con el mismo nombre en cada distribución, se utilizó el módulo “package” para lograr la instalación.

Las dependencias de cada servidor se instalarán de acuerdo a su distribución ya que, los nombres de los paquetes difieren de acuerdo a la distribución utilizada.

Para la configuración del archivo perteneciente al servicio “chrony”. Se modificó la ruta del archivo donde se copiaría el archivo.

El servicio “chrony” se iniciaría de acuerdo a la variable encontrada en el archivo cargado al principio del playbook. El servicio se cargaría mediante la variable debido a la diferencia del nombre en la distribución CentOS y Ubuntu.

Se agregó una tarea de handler con el nombre del servicio de la variable cargada al principio del playbook. Este handler cuenta la función de reiniciar el servicio y se ejecutaría cuando lo notifique la tarea de configuración del archivo de “chrony”.

```
---
#Mediante este playbook se instalarán dependencias, chrony y su configuración para CentOS y Ubuntu
- name: Instalación de paquetes
  hosts: all
  gather_facts: yes
  become: yes
  tasks:
    - name:
      include_vars:
        file: "{{ ansible_facts.distribution }}.yaml"

    - name: Instalar chrony en cada servidor
      package:
        name: chrony
        state: present

    - name: Instala dependencias en cada servidor
      package:
        name: "{{ packages }}"
        state: present

    - name: Configuración de archivo de chrony
      template:
        src: chrony.conf.j2
        dest: /etc/chrony.conf
        notify: restart chrony

    - name: Start the chrony service
      service:
        name: "{{ tiempo }}"
        state: started
        enabled: true
  handlers:
    - name: restart chrony
      service:
        name: "{{ tiempo }}"
        state: restarted
```

Contenido de playbook main.yml del rol de dependencias en común

<pre>--- #Variables específicas de CentOS tiempo: chronyd packages: - python3-libselinux - python3-libsemanage - firewalld</pre>	<pre>--- #Variables específicas de Ubuntu tiempo: chrony packages: - python3-selinux - python3-semanage - firewalld</pre>
--	---

Conjunto de variables utilizadas específicas de cada distribución

## Playbook base de datos

Al igual que en el playbook del rol “common”, se corrigió la indentación y se agregaron los módulos de “hosts”, “gather\_facts” y “become”.

Se agregaron dos tareas para cargar archivos de variables. Uno de estos archivos cuenta con la información necesaria para la configuración de la base de datos.

El siguiente archivo cuenta con variables de los paquetes a instalar. Este archivo de variables se cargará en el playbook de acuerdo a la distribución que contengan los servidores.

Con el fin de lograr la instalación de los paquetes requeridos en CentOS, se añadió previo a la instalación de paquetes requeridos, la tarea de la instalación del paquete “epel-release”.

La tarea de instalación de paquetes según la distribución utilizada se agregó mediante el módulo “package” y la variable “packages”.

Como el servicio de SELinux se encuentra activado en CentOS, se debió agregar la tarea para lograr que MySQL se inicie en cualquier puerto. Para esto se utilizó el módulo de “seboolean”. Como Ubuntu no cuenta con SELinux activado, la tarea solo se especificó para CentOS mediante una condicional “when”.

La siguiente tarea se añadió para el funcionamiento de MySQL. En esta tarea se copió un archivo “.j2” al directorio root del servidor.

Se añadió el parámetro “no\_log” en la tarea de la creación del usuario de base de datos. Mediante este parámetro, se logra que la información sensible, como la contraseña, no se guarde.

Por último, se agregaron dos handlers para el reinicio de los servicios “mariadb” y “mysql”. Estos handlers serían notificados en las tareas de creación de archivos de configuración.

## Taller de Servidores Linux

```
---
#Este playbook instala y configura todo lo necesario para correr MariaDB
- name: Instalar paquete MariaDB
  hosts: dbservers
  gather_facts: yes
  become: yes
  tasks:
    - name: Cargar variables de base de datos
      include_vars:
        file: "dbservers.yml"

    - name: Cargar variables de paquetes dependiendo de sistema
      include_vars:
        file: "{{ ansible_facts.distribution }}.yaml"

    - name: Instalar epel-release en CentOS
      package:
        name: epel-release
        state: latest
      when: ansible_facts.distribution == "CentOS"

    - name: Instalar paquetes acorde a sistema
      package:
        name: "{{ packages }}"
        state: latest

    - name: Configura SELinux para que MySQL inicie en cualquier puerto
      seboolean:
        name: mysql_connect_any
        state: true
        persistent: yes
      when: ansible_facts.distribution == "CentOS"

    - name: Crea '/root/.my.cnf' con credenciales root
      template:
        src: my.cnfroot.j2
        dest: /root/.my.cnf
        owner: root
        mode: 0600
      become: yes
      notify: restart MySQL

    - name: Crea archivo de configuracion MySQL
      template:
        src: my.cnf.j2
        dest: "{{ ruta }}"
      notify: restart mariadb

    - name: Crea archivo de log de MariaDB
      file:
        path: "{{ ruta_log }}"
        state: touch
        owner: mysql
        group: mysql
        mode: 0775
```

```
- name: Crea directorio MariaDB PID
  file:
    path: /var/run/mysqld
    state: directory
    owner: mysql
    group: mysql
    mode: 0775

- name: Inicia servicio MariaDB
  service:
    name: mariadb
    state: started
    enabled: yes

- name: Inicia firewalld
  service:
    name: firewallld
    state: started
    enabled: yes

- name: Insertar regla firewallld
  firewallld:
    port: "{{ mysql_port }}/tcp"
    permanent: true
    state: enabled
    immediate: yes

- name: Crear base de datos de aplicación
  mysql_db:
    name: "{{ dbname }}"
    state: present

- name: Crear usuario de aplicación de base de datos
  mysql_user:
    name: "{{ dbuser }}"
    password: "{{ upassword }}"
    priv: "*.*:ALL"
    host: '%'
    state: present
    no_log: True

handlers:
  - name: restart mariadb
    service:
      name: mariadb
      state: restarted

  - name: restart MySQL
    service:
      name: mysql
      state: restarted
```

Contenido de playbook main.yml correspondiente al rol de base de datos

```
1  ---
2  # Variables utilizadas en el playbook de base de datos
3
4  mysqlservice: mysqld
5  mysql_port: 3306
6  dbuser: foouser
7  dbname: foodb
8  upassword: abc
```

Variables en común utilizadas en el playbook

1	#Paquetes específicos de Ubuntu	#Paquetes específicos de CentOS
2	packages:	packages:
3	- python3-mysqldb	- python3-mysql
4	- python3-semanage	- python3-libsemanage
5	- mariadb-server	- mariadb-server

Variables específicas de cada distribución para el rol de base de datos

## Playbook servidor web

Al igual que en los playbooks anteriores, se corrigió la indentación, se añadieron los módulos para especificar hosts, para recolectar información de los servidores y para ejecutar los playbooks con privilegios escalados.

Se incluyen dos archivos de variables al comienzo de la tarea. Uno de estos archivos cuenta con información del puerto de apache. El archivo restante se utiliza para instalar paquetes de acuerdo a la distribución del servidor a configurar el rol.

En las siguientes tareas se realizaron leves cambios. Para el inicio del servicio web, se utilizó el nombre de acuerdo a la distribución, previamente cargado mediante un archivo de variables.

Para finalizar, la configuración de SELinux en CentOS se implementó, de nuevo, mediante una condicional “when”.

```
---
#Este playbook instala paquetes de apache y php

- name: Instalar apache y php
  hosts: dbservers
  gather_facts: yes
  become: yes
  tasks:
    - name: Cargar variables httpd
      include_vars:
        file: "all.yml"

    - name: Cargar variables según distribución
      include_vars:
        file: "{{ ansible_facts.distribution }}.yml"

    - name: Instala los paquetes de apache y php según distribución
      package:
        name: "{{ packages }}"
        state: present

    - name: Instalar dependencias web
      package:
        name: git
        state: present

    - name: Iniciar firewalld
      service:
        name: firewalld
        state: started
        enabled: yes

    - name: Insertar regla httpd de firewalld
      firewallld:
        port: "{{ httpd_port }}/tcp"
        permanent: true
        state: enabled
        immediate: yes

    - name: Iniciar servicio http
      service:
        name: "{{ web }}"
        state: started
        enabled: yes

    - name: Configurar SELinux para permitir a apache que se conecte a base de dato remota
      seboolean:
        name: httpd_can_network_connect_db
        state: true
        persistent: yes
      when: ansible_facts.distribution == "CentOS"
```

Contenido de playbook install\_httpd.yml del rol de servidor web

En el segundo playbook web, se agregaron los mismos módulos al principio (hosts, gather\_facts y become).

En la primera tarea, se cargan variables desde el archivo mencionado en el playbook anterior. Una de las tareas del playbook implica la copia de dos archivos desde un repositorio.

En la segunda tarea agregada, se cargarán las variables que se utilizarán para conectarse a la base de datos mediante el template copiado al final del playbook.

En cuanto a la copia del repositorio, si antes de la copia existe un archivo con el mismo nombre del archivo a copiar, se manifestará un error del playbook. Para solucionar esto, se agregó una tarea para eliminar el archivo encontrado por defecto en la ruta donde se copiará el contenido del repositorio.

```
---
- name: Copiar desde git y creación de index.php
  hosts: all
  gather_facts: yes
  become: yes
  tasks:
    - name: Cargar variables de archivo
      include_vars:
        file: "all.yml"

    - name: Cargar variables de paquetes
      include_vars:
        file: "{{ ansible_facts.distribution }}.yml"

    - name: Borrar archivo index.html para que git pueda copiar
      file:
        path: /var/www/html/index.html
        state: absent

    - name: Copia código desde repositorio
      git:
        repo: "{{ repository }}"
        dest: /var/www/html/
        force: yes

    - name: Crea el archivo index.php
      template:
        src: index.php.j2
        dest: /var/www/html/index.php
```

Segundo playbook perteneciente a copy\_code.yml del rol web



Al template “index.php” encontrado por defecto se le sustituyeron las variables utilizadas para conectarse a la base de datos por otras encontradas en el archivo “dbcredentials”.

```
1 <html>
2 <head>
3 <title>Ansible Application</title>
4 </head>
5 <body>
6 </br>
7 <a href=http://{{ ansible_default_ipv4.address }}/index.html>Homepage</a>
8 </br>
9 <?php
10 Print "Hello, World! I am a web server configured using Ansible and I am : ";
11 echo exec('hostname');
12 Print "</BR>";
13 echo "List of Databases: </BR>";
14     {% for host in groups['dbservers'] %}
15         $link = mysqli_connect('{{ dbserverip }}', '{{ dbuser }}', '{{ upassword }}') or die(mysqli_connect_error($link));
16     {% endfor %}
17     $res = mysqli_query($link, "SHOW DATABASES;");
18     while ($row = mysqli_fetch_assoc($res)) {
19         echo $row['Database'] . "\n";
20     }
21 ?>
22 </body>
23 </html>
```

Template desarrollado en PHP para ser copiado al servidor web

```
1 httpd_port: 80
2 ntpserver: 192.168.1.2
3 repository: https://github.com/bennojoy/mywebapp.git
4
```

Variables utilizadas para especificar el puerto de Apache y dirección de repositorio a utilizar

```
1 packages:
2 - apache2
3 - php
4 - php-mysqlnd
5 httpd_port: 80
6 ntpserver: 192.168.1.2
7 repository: https://github.com/bennojoy/mywebapp.git
8 web: apache2
9 dbuser: foouser
10 dbname: foodb
11 upassword: abc
12 dbserverip: 192.168.0.107
```

```
1 packages:
2 - httpd
3 - php
4 - php-mysqlnd
5 httpd_port: 80
6 ntpserver: 192.168.1.2
7 repository: https://github.com/bennojoy/mywebapp.git
8 web: httpd
9 dbuser: foouser
10 dbname: foodb
11 upassword: abc
12 dbserverip: 192.168.0.104
13
```

Variables específicas de cada distribución para los paquetes a instalar

```
1 dbuser: foouser
2 dbname: foodb
3 upassword: abc
4 dbserverip: 192.168.0.104
```

Variables con información necesaria para lograr una conexión a la base de datos

## Ejecución de playbooks

```
[root@controller obligatorio]# ansible-playbook lamp/roles/common/tasks/main.yml

PLAY [Instalación de paquetes] *****

TASK [Gathering Facts] *****
ok: [obcentos]
ok: [obubuntu]

TASK [include_vars] *****
ok: [obubuntu]
ok: [obcentos]

TASK [Instalar chrony en cada servidor] *****
ok: [obubuntu]
ok: [obcentos]

TASK [Instala dependencias en cada servidor] *****
changed: [obcentos]
changed: [obubuntu]

TASK [Configuración de archivo de chrony] *****
changed: [obubuntu]
changed: [obcentos]

TASK [Start the chrony service] *****
ok: [obubuntu]
ok: [obcentos]

RUNNING HANDLER [restart chrony] *****
changed: [obubuntu]
changed: [obcentos]

PLAY RECAP *****
obcentos           : ok=7    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    i
gnored=0
obubuntu           : ok=7    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    i
gnored=0
```

Ejecución del rol de dependencias comunes en ambos servidores

```

[root@controller obligatorio]# ansible-playbook lamp/roles/db/tasks/main.yml

PLAY [Instalar paquete MariaDB] *****
TASK [Gathering Facts] *****ok: [obcen
tos]
ok: [obubuntu]

TASK [Cargar variables de base de datos] *****ok: [obcen
tos]
ok: [obubuntu]

TASK [Cargar variables de paquetes dependiendo de sistema] *****ok: [obcen
tos]
ok: [obubuntu]

TASK [Instalar epel-release en CentOS] *****skipping:
[obubuntu]
changed: [obcentos]

TASK [Instalar paquetes acorde a sistema] *****changed: [
obubuntu]
changed: [obcentos]

TASK [Configura SELinux para que MySQL inicie en cualquier puerto] *****skipping:
[obubuntu]
changed: [obcentos]

TASK [Crea '/root/.my.cnf' con credenciales root] *****changed: [
obubuntu]
changed: [obcentos]

TASK [Crea archivo de configuracion MySQL] *****changed: [
obubuntu]
changed: [obcentos]

TASK [Crea archivo de log de MariaDB] *****changed: [
obubuntu]
changed: [obcentos]

TASK [Crea directorio MariaDB PID] *****changed: [
obubuntu]
changed: [obcentos]

TASK [Inicia servicio MariaDB] *****ok: [obubu
ntu]
changed: [obcentos]

TASK [Inicia firewall] *****ok: [obubu
ntu]
ok: [obcentos]

TASK [Insertar regla firewall] *****changed: [
obubuntu]
changed: [obcentos]

TASK [Crear base de datos de aplicación] *****changed: [
obubuntu]
changed: [obcentos]

TASK [Crear usuario de aplicación de base de datos] *****changed: [
obubuntu]
changed: [obcentos]

RUNNING HANDLER [restart mariadb] *****changed: [
obcentos]
changed: [obubuntu]

RUNNING HANDLER [restart MySQL] *****changed: [
obcentos]
changed: [obubuntu]

PLAY RECAP *****obcentos
: ok=17 changed=13 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
obubuntu : ok=15 changed=10 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0

```

Ejecución de playbook de base de datos

```
[root@controller obligatorio]# ansible-playbook lamp/roles/web/tasks/install_httpd.yml

PLAY [Instalar apache y php] *****

TASK [Gathering Facts] *****
ok: [obubuntu]
ok: [obcentos]

TASK [Cargar variables httpd] *****
ok: [obcentos]
ok: [obubuntu]

TASK [Cargar variables según distribución] *****
ok: [obcentos]
ok: [obubuntu]

TASK [Instala los paquetes de apache y php según distribución] *****
ok: [obubuntu]
changed: [obcentos]

TASK [Instalar dependencias web] *****
ok: [obubuntu]
changed: [obcentos]

TASK [Iniciar firewall] *****
ok: [obubuntu]
ok: [obcentos]

TASK [Insertar regla httpd de firewall] *****
ok: [obubuntu]
changed: [obcentos]

TASK [Iniciar servicio http] *****
ok: [obubuntu]
changed: [obcentos]

TASK [Configurar SELinux para permitir a apache que se conecte a base de dato remota] *****
skipping: [obubuntu]
changed: [obcentos]

PLAY RECAP *****
obcentos      : ok=9   changed=5   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
obubuntu     : ok=8   changed=0   unreachable=0   failed=0   skipped=1   rescued=0   ignored=0
```

Ejecución de playbook de instalación de rol web

```
[root@controller obligatorio]# ansible-playbook lamp/roles/web/tasks/copy_code.yml

PLAY [Copiar desde git y creación de index.php] *****

TASK [Gathering Facts] *****
ok: [obcentos]
ok: [obubuntu]

TASK [Cargar variables de archivo] *****
ok: [obcentos]
ok: [obubuntu]

TASK [Cargar variables de paquetes] *****
ok: [obcentos]
ok: [obubuntu]

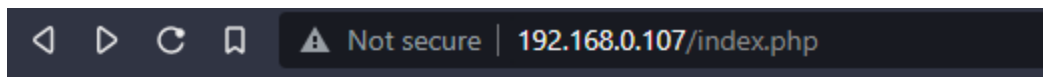
TASK [Borrar archivo index.html para que git pueda copiar] *****
changed: [obubuntu]
ok: [obcentos]

TASK [Copia código desde repositorio] *****
changed: [obubuntu]
changed: [obcentos]

TASK [Crea el archivo index.php] *****
changed: [obubuntu]
changed: [obcentos]

PLAY RECAP *****
obcentos : ok=6  changed=2  unreachable=0  failed=0  skipped=0  rescued=0  i
gnored=0
obubuntu : ok=6  changed=3  unreachable=0  failed=0  skipped=0  rescued=0  i
gnored=0
```

Ejecución de playbook de creación de archivo index.html e index.php



[Homepage](#)

Hello, World! I am a web server configured using Ansible and I am : obubuntu

List of Databases:

foodb information\_schema mysql performance\_schema

Conexión a base de datos en CentOS



[Homepage](#)

Hello, World! I am a web server configured using Ansible and I am : obcentos

List of Databases:

foodb information\_schema mysql performance\_schema

Conexión a base de datos en Ubuntu