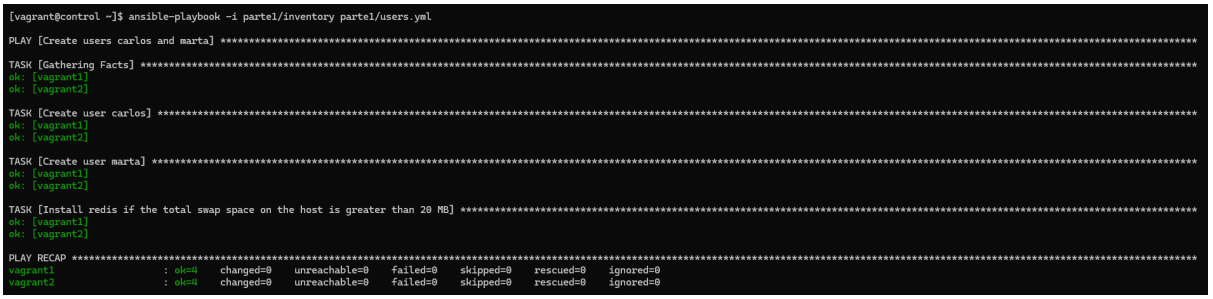
Práctica final de Ansible

Parte I

1) En el nodo de control crear un fichero llamado **inventory** el cual deberá contener un grupo de hosts (**vagrant1** y **vagrant2**) en un grupo llamado servidores

2) Crear un fichero de configuración **ansible.cfg** donde se indique el nombre del inventario a utilizar.

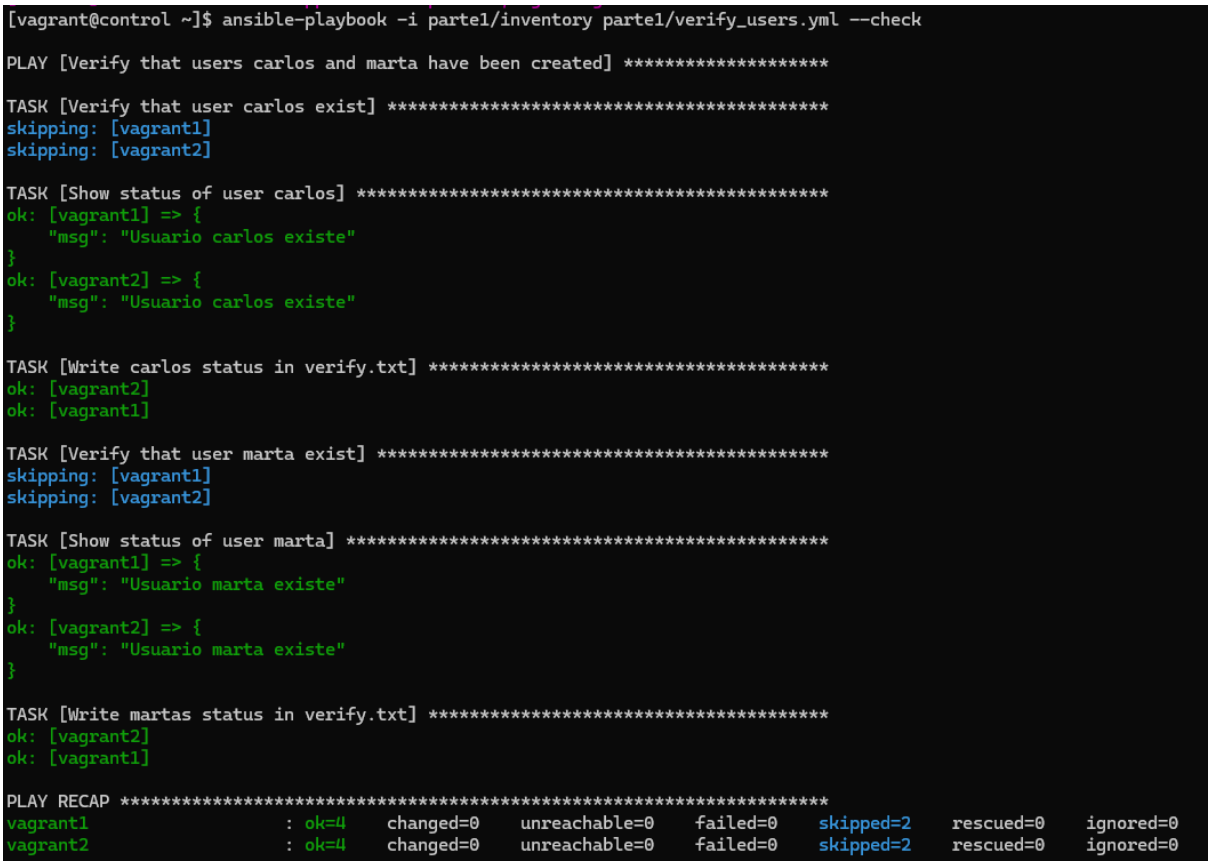
3) Crear un playbook llamado **users.yml** el cual deberá ejecutarse sobre los hosts del grupo servidores y tendrá una tarea para crear los usuarios carlos y marta.



4) Defina una variable llamada packages con una lista de dos paquetes como su valor: memcached y mariadb-server.

5) Añada una tarea al playbook que instale el paquete **redis** si el total de espacio swap en el host es **mayor** a **20MB**.

6) Verifique por qué el playbook **verify_users.yml** no funciona correctamente. Este playbook en principio se usa para verificar que los usuarios carlos y marta han sido creados correctamente y se supone que no debería crear el usuario carlos si este no existe. Ejecuta el playbook con la opción **"--check"** y corrige cualquier error. Repita este proceso hasta que pueda ejecutar el playbook con la opción --check normalmente (**sin errores**).



Parte II

1) Crear un playbook llamado **dev_deploy.yml** con un play que se ejecute sobre el grupo de hosts llamados servidores. Habilite la escalada de privilegios para el play. Añada las siguientes tareas:

1. Instalar el paquete **httpd**.
2. **Inicie** el servicio **httpd** y habilítalo para que arranque al iniciar la máquina
3. Despliegue el template **templates/vhost.conf.j2** a **/etc/httpd/conf.d/vhost.conf** en los hosts. Esta tarea debería notificar al handler Restart httpd.
4. **Copie** el fichero **files/index.html** al directorio /var/www/vhosts/hostname en cada host. Este directorio deberá ser creado en caso que no exista con el valor del hostname de cada máquina.
5. **Configure el firewall** para permitir el servicio httpd tanto en el puerto 80 como en el 443 (ambos TCP).
6. Añada el handler **Restart httpd** al play que reinicie el servicio httpd.

```
[vagrant@control ~]$ cd parte2
[vagrant@control parte2]$ ansible-playbook dev_deploy.yml

PLAY [Deploy web server on hosts in the server group] ********************************************

TASK [Gathering Facts] ***************************************************************************
ok: [vagrant2]
ok: [vagrant1]

TASK [Install httpd package] *********************************************************************
ok: [vagrant1]
ok: [vagrant2]

TASK [Enable and start httpd service] ************************************************************
ok: [vagrant2]
ok: [vagrant1]

TASK [Deploy the vhost template] *****************************************************************
ok: [vagrant2]
ok: [vagrant1]

TASK [Create /var/www/vhosts/centos9s] ***********************************************************
ok: [vagrant1]
ok: [vagrant2]

TASK [Copy index.html] ***************************************************************************
ok: [vagrant2]
ok: [vagrant1]

TASK [Allow http and https service in firewall] **************************************************
ok: [vagrant2] => (item=http)
ok: [vagrant1] => (item=http)
ok: [vagrant2] => (item=https)
ok: [vagrant1] => (item=https)

PLAY RECAP ***************************************************************************************
vagrant1                   : ok=7    changed=0    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0
vagrant2                   : ok=7    changed=0    unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0
```

2) Crear un playbook llamado **get_web_content.yml** con un play llamado **Test web conten**t que se ejecute en el host de control. Este playbook prueba si el playbook **dev_deploy.yml** se ha ejecutado satisfactoriamente y se asegura que el servicio web esté sirviendo contenido. Habilite la escalada de privilegios para el play.

1. Crear una tarea llamada “Retrieve web content and write to error log on failure.”. Haga que esta tarea esté en un block y si falla, deberá haber un bloque rescue que ejecute otra tarea (ver las 2 siguientes tareas):

1.1. Dentro del block, crear una tarea llamada “Retrieve web content” que retorne el contenido de http://vagrant1 y registre los resultados en una variable llamada content.

1.2. Dentro de la cláusula rescue, crear una tarea llamada Write to error file que escriba el valor de la variable content al fichero error.log, si el block falla. La tarea deberá crear el fichero error.log si este no existe.

```
[vagrant@control ~]$ cd parte2
[vagrant@control parte2]$ ansible-playbook get_web_content.yml

PLAY [Test web content] ********************************************************************************

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

TASK [Retrieve web content] ****************************************************************************
ok: [localhost]

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

3) Crear un playbook llamado site.yml que importe los plays de ambos playbooks: **dev_deploy.yml** y **get_web_content.yml**.

4) Ejecuta el playbook.yml y asegúrese que los tres playbooks se ejecutan correctamente.

```
[vagrant@control ~]$ cd parte2
[vagrant@control parte2]$ ansible-playbook site.yml

PLAY [Deploy web server on hosts in the server group] **************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [vagrant1]
ok: [vagrant2]

TASK [Install httpd package] ***************************************************************************
ok: [vagrant2]
ok: [vagrant1]

TASK [Enable and start httpd service] ******************************************************************
ok: [vagrant1]
ok: [vagrant2]

TASK [Deploy the vhost template] ***********************************************************************
ok: [vagrant1]
ok: [vagrant2]

TASK [Create /var/www/vhosts/centos9s] *****************************************************************
ok: [vagrant2]
ok: [vagrant1]

TASK [Copy index.html] *********************************************************************************
ok: [vagrant2]
ok: [vagrant1]

TASK [Allow http and https service in firewall] ********************************************************
ok: [vagrant2] => (item=http)
ok: [vagrant1] => (item=http)
ok: [vagrant2] => (item=https)
ok: [vagrant1] => (item=https)

PLAY [Test web content] ********************************************************************************

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

TASK [Retrieve web content] ****************************************************************************
ok: [localhost]

PLAY RECAP *********************************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
vagrant1                   : ok=7    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
vagrant2                   : ok=7    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```