EXAMEN ANSIBLE

Cristóbal Suárez Abad

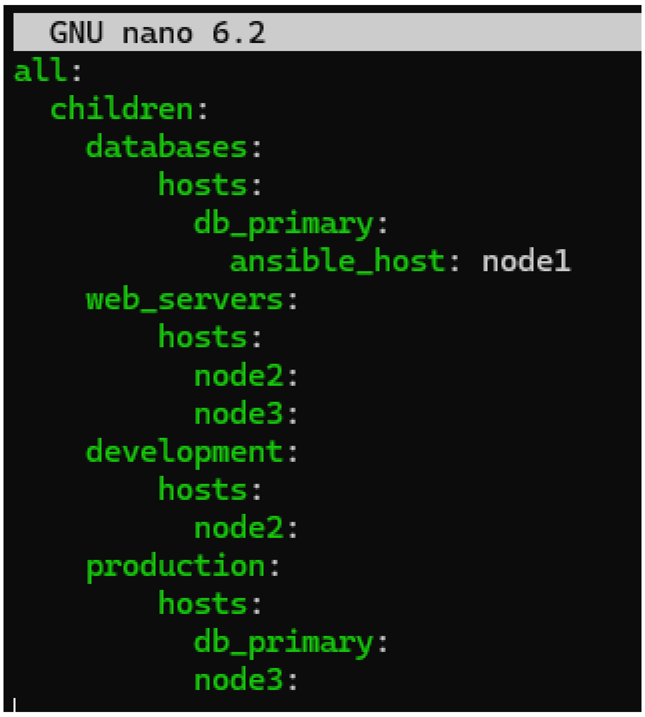
Escenario

Has sido contratado por "GlobalLog" para automatizar su nuevo centro de datos. La infraestructura consta de tres servidores:

- **SRV-DB (node1):** Servidor de base de datos PostgreSQL.
- **SRV-WEB-DEV (node2):** Servidor web para pruebas de desarrollo.
- **SRV-WEB-PROD (node3):** Servidor web de producción.

Fase 1: Inventario y Estructura de Configuración (3 puntos)

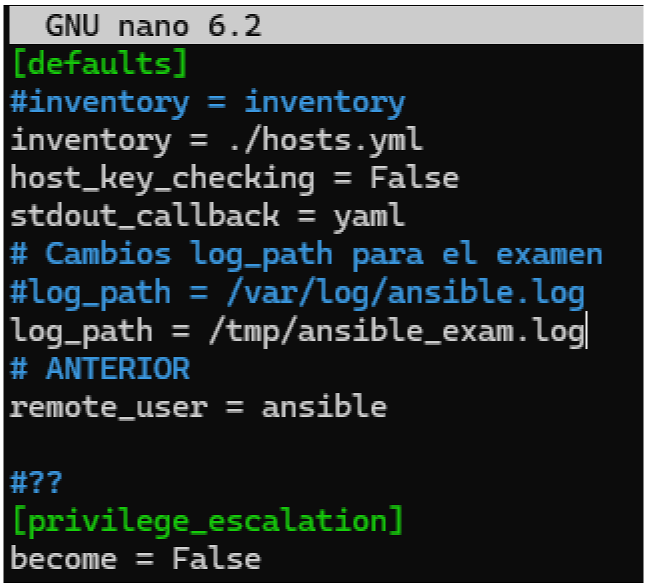
1. **Archivo de Inventario**
   - Usa un alias para `node1` llamado `db_primary`.
   - Crea un grupo funcional `databases` (con `db_primary`) y un grupo `web_servers` (con `node2` y `node3`).
   - Crea grupos por entorno: `development` (`node2`) y `production` (`db_primary` y `node3`).



```
  GNU nano 6.2
all:
  children:
    databases:
        hosts:
          db_primary:
            ansible_host: node1
    web_servers:
        hosts:
          node2:
          node3:
    development:
        hosts:
          node2:
    production:
        hosts:
          db_primary:
          node3:
```

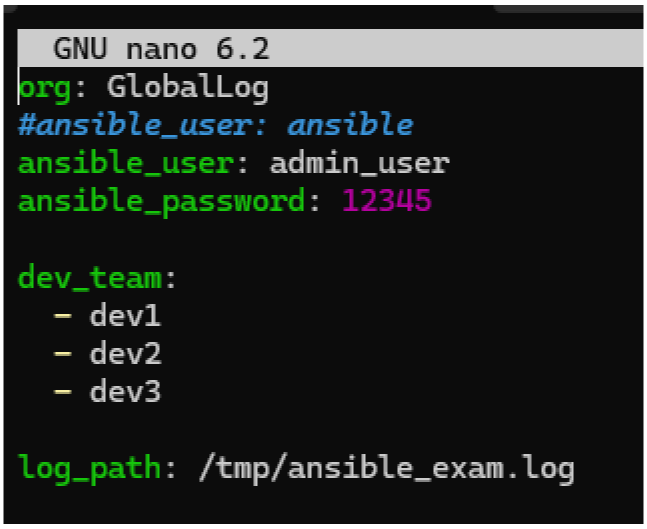
2. **Genera logs:**
    - Genere logs en `/tmp/ansible_exam.log`.

Cambiamos “log” en ansible.cfg



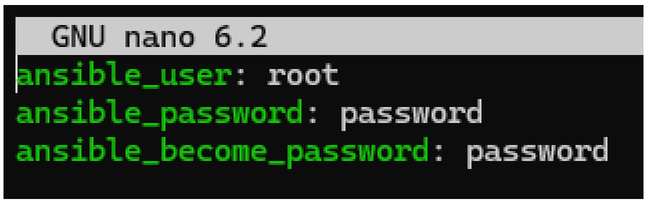
```
  GNU nano 6.2
[defaults]
#inventory = inventory
inventory = ./hosts.yml
host_key_checking = False
stdout_callback = yaml
# Cambios log_path para el examen
#log_path = /var/log/ansible.log
log_path = /tmp/ansible_exam.log
# ANTERIOR
remote_user = ansible

#??
[privilege_escalation]
become = False
```

3. **Desacoplamiento de Variables:**
    - En `group_vars/all.yml`, define la variable `org: GlobalLog` y el usuario de conexión por defecto `ansible_user: admin_user`.



```
  GNU nano 6.2
org: GlobalLog
#ansible_user: ansible
ansible_user: admin_user
ansible_password: 12345

dev_team:
  - dev1
  - dev2
  - dev3

log_path: /tmp/ansible_exam.log
```

- En `host_vars/db_primary.yml`, sobreescribe el usuario para que la conexión sea como `root`.

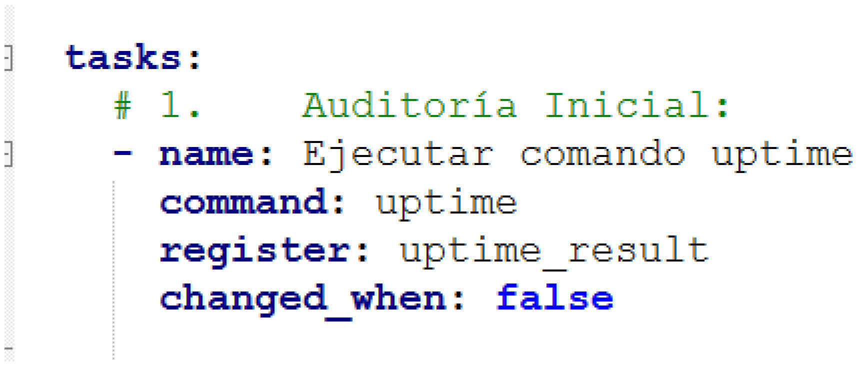


```
  GNU nano 6.2
ansible_user: root
ansible_password: password
ansible_become_password: password
```

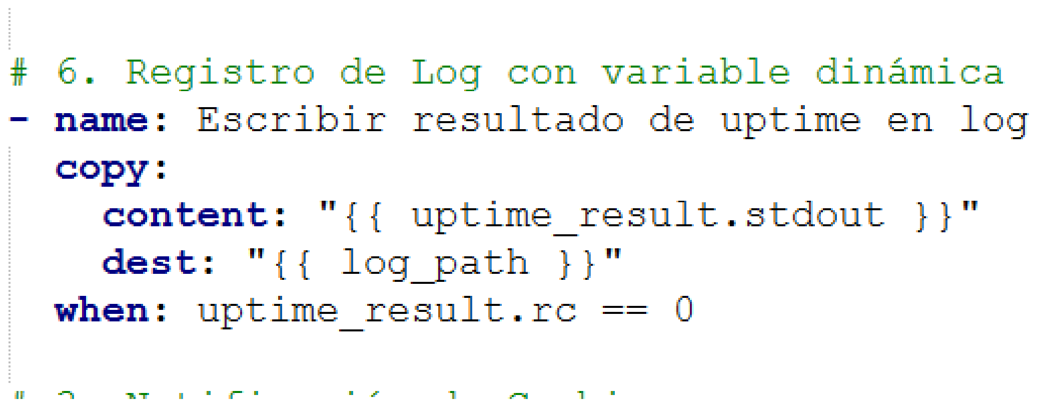
Fase 2: Lógica y Seguridad con Playbooks (3 puntos)

Crea un playbook llamado `system_init.yml` que realice las siguientes tareas:

1. **Auditoría Inicial:** Ejecute el comando `uptime` y guarde el resultado en una variable.

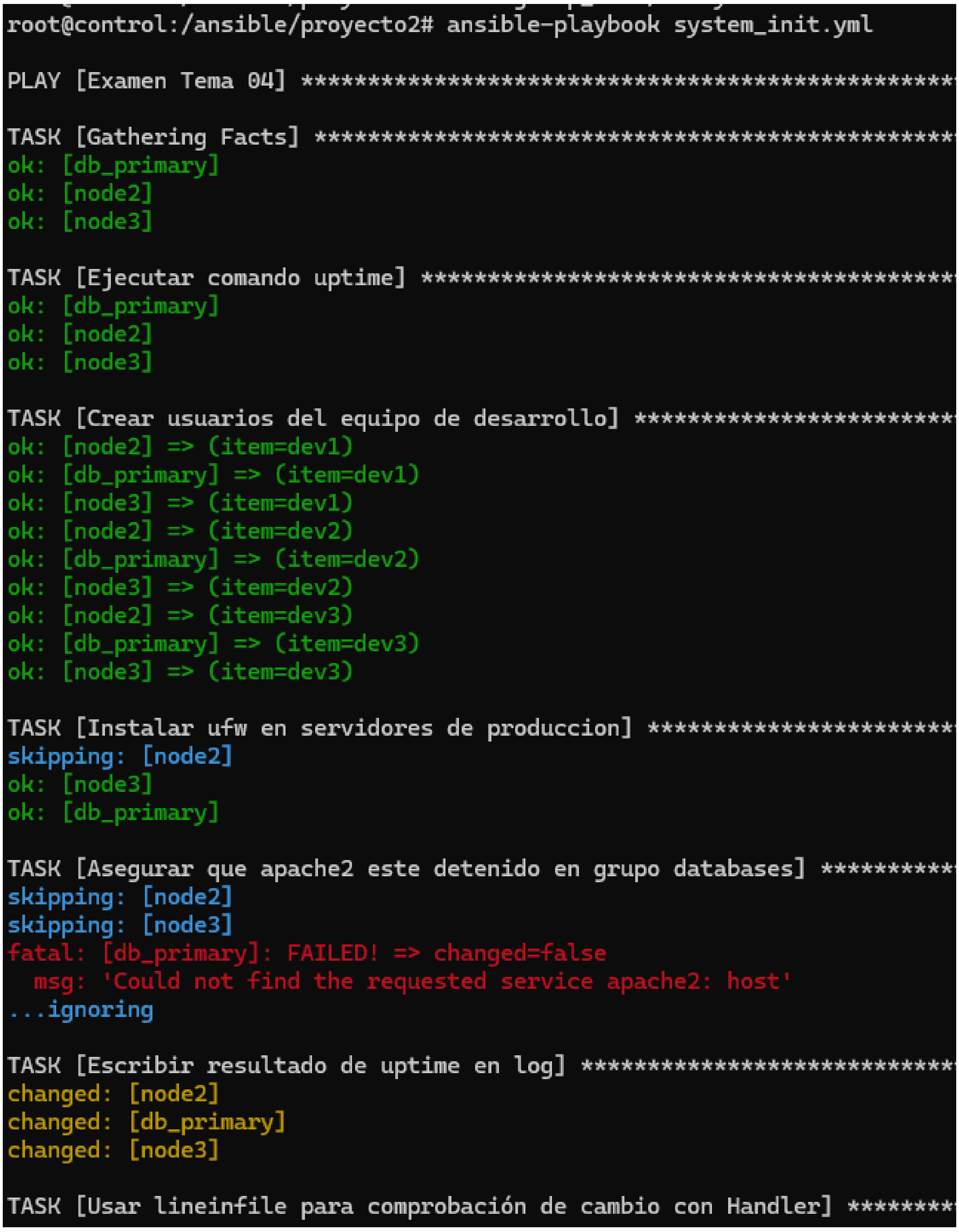


```yaml
tasks:
  # 1.    Auditoría Inicial:
  - name: Ejecutar comando uptime
    command: uptime
    register: uptime_result
    changed_when: false
```

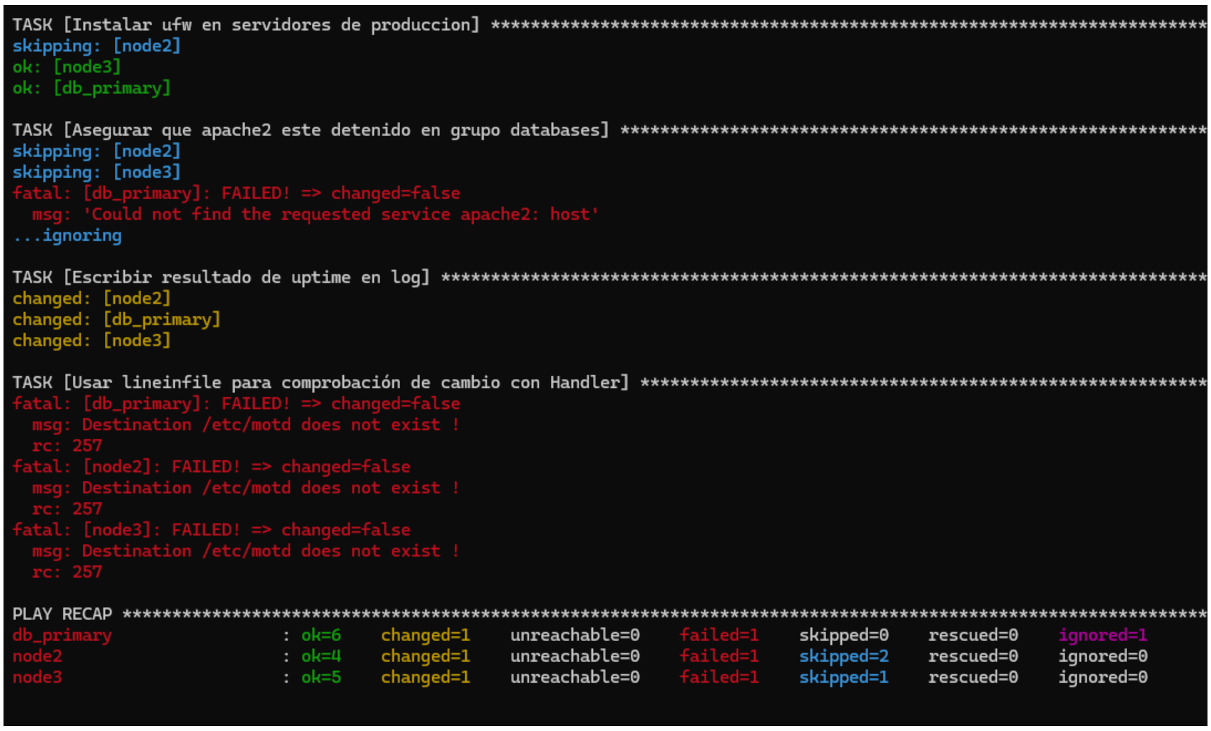


```yaml
# 6. Registro de Log con variable dinámica
- name: Escribir resultado de uptime en log
  copy:
    content: "{{ uptime_result.stdout }}"
    dest: "{{ log_path }}"
  when: uptime_result.rc == 0
```

2. **Gestión de Usuarios:** Cree tres usuarios (`dev1`, `dev2`, `dev3`) definidos en una lista llamada `dev_team`, asegurándose de que pertenezcan al grupo `sudo`.
3. **Configuración Condicional:** * Si el host pertenece al grupo `production`, debe instalar el paquete `ufw` (firewall).
    - Si el host es de `databases`, asegúrese de que el servicio `apache2` esté detenido y deshabilitado.
4. **Notificación de Cambio:** Si se modifica cualquier archivo de configuración (puedes simularlo con una tarea de `lineinfile` en `/etc/motd`), debe ejecutar un **Handler** que imprima un mensaje de "Configuración actualizada".



```
root@control:/ansible/proyecto2# ansible-playbook system_init.yml

PLAY [Examen Tema 04] **************************************************

TASK [Gathering Facts] *************************************************
ok: [db_primary]
ok: [node2]
ok: [node3]

TASK [Ejecutar comando uptime] *****************************************
ok: [db_primary]
ok: [node2]
ok: [node3]

TASK [Crear usuarios del equipo de desarrollo] *************************
ok: [node2] => (item=dev1)
ok: [db_primary] => (item=dev1)
ok: [node3] => (item=dev1)
ok: [node2] => (item=dev2)
ok: [db_primary] => (item=dev2)
ok: [node3] => (item=dev2)
ok: [node2] => (item=dev3)
ok: [db_primary] => (item=dev3)
ok: [node3] => (item=dev3)

TASK [Instalar ufw en servidores de produccion] ************************
skipping: [node2]
ok: [node3]
ok: [db_primary]

TASK [Asegurar que apache2 este detenido en grupo databases] ***********
skipping: [node2]
skipping: [node3]
fatal: [db_primary]: FAILED! => changed=false
  msg: 'Could not find the requested service apache2: host'
...ignoring

TASK [Escribir resultado de uptime en log] *****************************
changed: [node2]
changed: [db_primary]
changed: [node3]

TASK [Usar lineinfile para comprobación de cambio con Handler] *********
```



```
TASK [Instalar ufw en servidores de produccion] ************************************************************************
skipping: [node2]
ok: [node3]
ok: [db_primary]

TASK [Asegurar que apache2 este detenido en grupo databases] ***********************************************************
skipping: [node2]
skipping: [node3]
fatal: [db_primary]: FAILED! => changed=false
  msg: 'Could not find the requested service apache2: host'
...ignoring

TASK [Escribir resultado de uptime en log] *****************************************************************************
changed: [node2]
changed: [db_primary]
changed: [node3]

TASK [Usar lineinfile para comprobación de cambio con Handler] *********************************************************
fatal: [db_primary]: FAILED! => changed=false
  msg: Destination /etc/motd does not exist !
  rc: 257
fatal: [node2]: FAILED! => changed=false
  msg: Destination /etc/motd does not exist !
  rc: 257
fatal: [node3]: FAILED! => changed=false
  msg: Destination /etc/motd does not exist !
  rc: 257

PLAY RECAP *************************************************************************************************************
db_primary                 : ok=6    changed=1    unreachable=0    failed=1    skipped=0    rescued=0    ignored=1
node2                      : ok=4    changed=1    unreachable=0    failed=1    skipped=2    rescued=0    ignored=0
node3                      : ok=5    changed=1    unreachable=0    failed=1    skipped=1    rescued=0    ignored=0
```

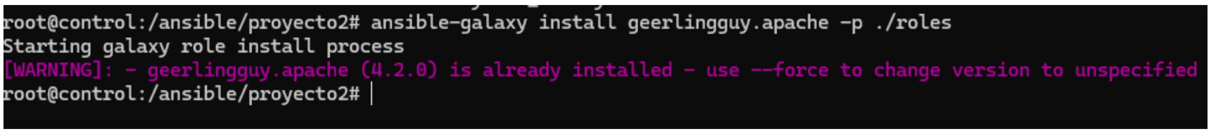
Fase 3: Modularización y Roles (3 puntos)

Reestructura tu solución para convertirla en un despliegue profesional:

1. **Ansible Galaxy:** Descarga e integra un rol de la comunidad (por ejemplo, `geerlingguy.apache`) para gestionar los servidores web.

Lo descargamos e integramos en la carpeta "roles" que creamos con mkdir

ansible-galaxy install geerlingguy.apache -p ./roles

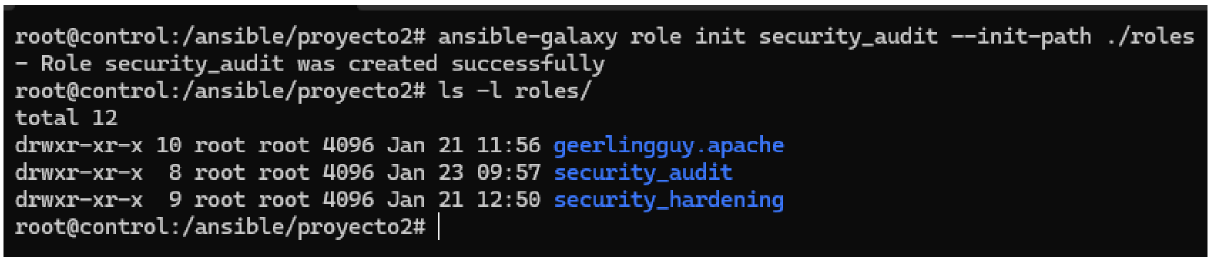


```
root@control:/ansible/proyecto2# ansible-galaxy install geerlingguy.apache -p ./roles
Starting galaxy role install process
[WARNING]: - geerlingguy.apache (4.2.0) is already installed - use --force to change version to unspecified
root@control:/ansible/proyecto2#
```

2. **Rol `security_audit`:** Crea un rol propio llamado `security_audit`:

Lo creamos también en la carpeta "roles"

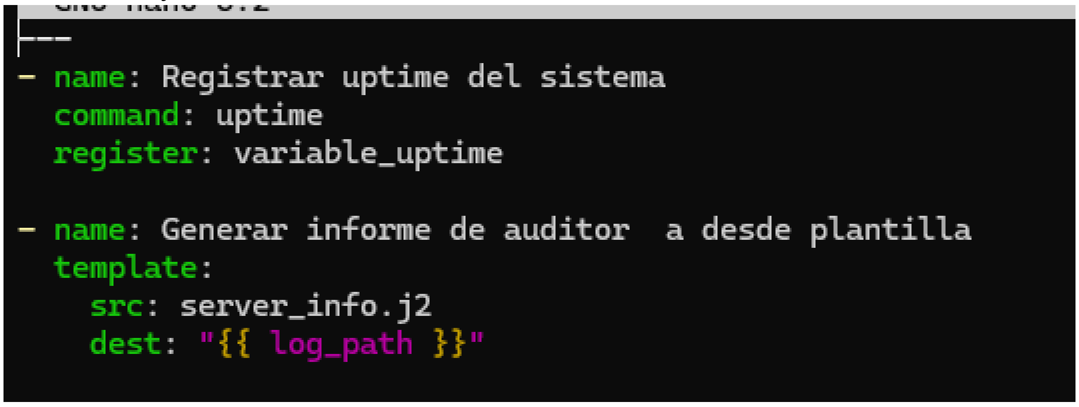
ansible-galaxy role init security_`audit` --init-path ./roles



```
root@control:/ansible/proyecto2# ansible-galaxy role init security_audit --init-path ./roles
- Role security_audit was created successfully
root@control:/ansible/proyecto2# ls -l roles/
total 12
drwxr-xr-x 10 root root 4096 Jan 21 11:56 geerlingguy.apache
drwxr-xr-x  8 root root 4096 Jan 23 09:57 security_audit
drwxr-xr-x  9 root root 4096 Jan 21 12:50 security_hardening
root@control:/ansible/proyecto2#
```

   - **Tasks:** Mueve la tarea de `uptime` a este rol.

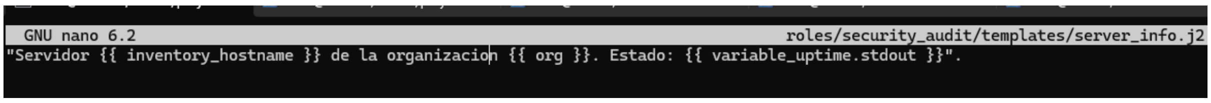
/tasks/main.yml



```
---
- name: Registrar uptime del sistema
  command: uptime
  register: variable_uptime

- name: Generar informe de auditor  a desde plantilla
  template:
    src: server_info.j2
    dest: "{{ log_path }}"
```

- **Templates:** Crea una plantilla Jinja2 en `roles/security_audit/templates/server_info.j2` que genere un archivo en `/tmp/info.txt` con el siguiente formato: *"Servidor {{ inventory_hostname }} de la organización {{ org }}. Estado: {{ variable_uptime.stdout }}"*.

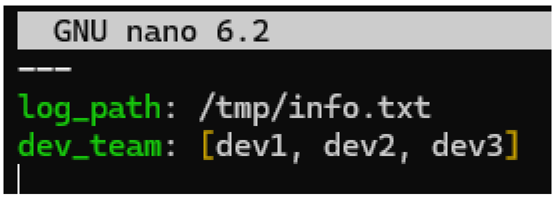
mkdir roles/security_audit/templates



```
  GNU nano 6.2                                                          roles/security_audit/templates/server_info.j2
"Servidor {{ inventory_hostname }} de la organizacion {{ org }}. Estado: {{ variable_uptime.stdout }}".
```

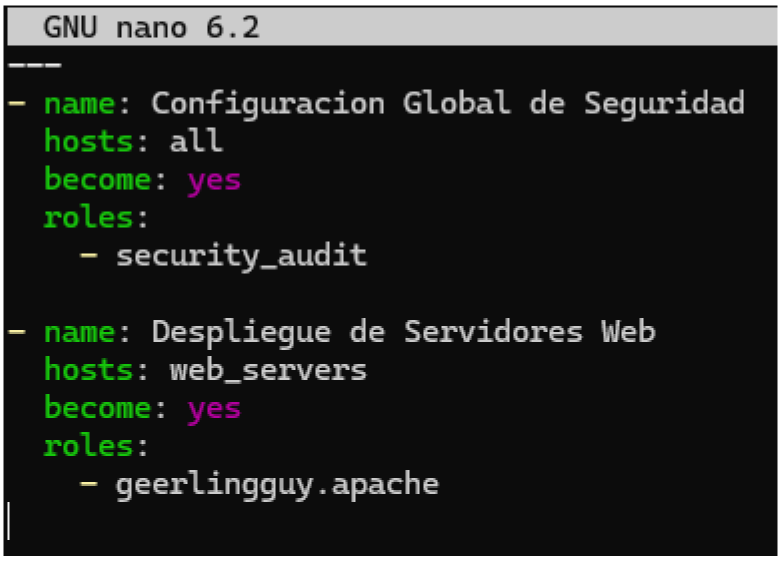
Para las variables debemos usar

roles/security_audit/defaults/main.yml

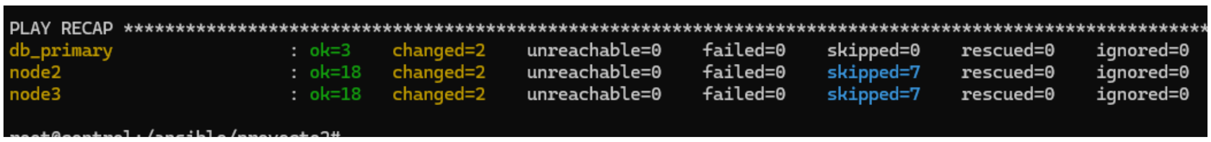


```
  GNU nano 6.2
---
log_path: /tmp/info.txt
dev_team: [dev1, dev2, dev3]
```

3. **Playbook Maestro (`site.yml`):** Crea un punto de entrada único que aplique el rol `security_audit` a todos los hosts y el rol de Apache solo a los `web_servers`.

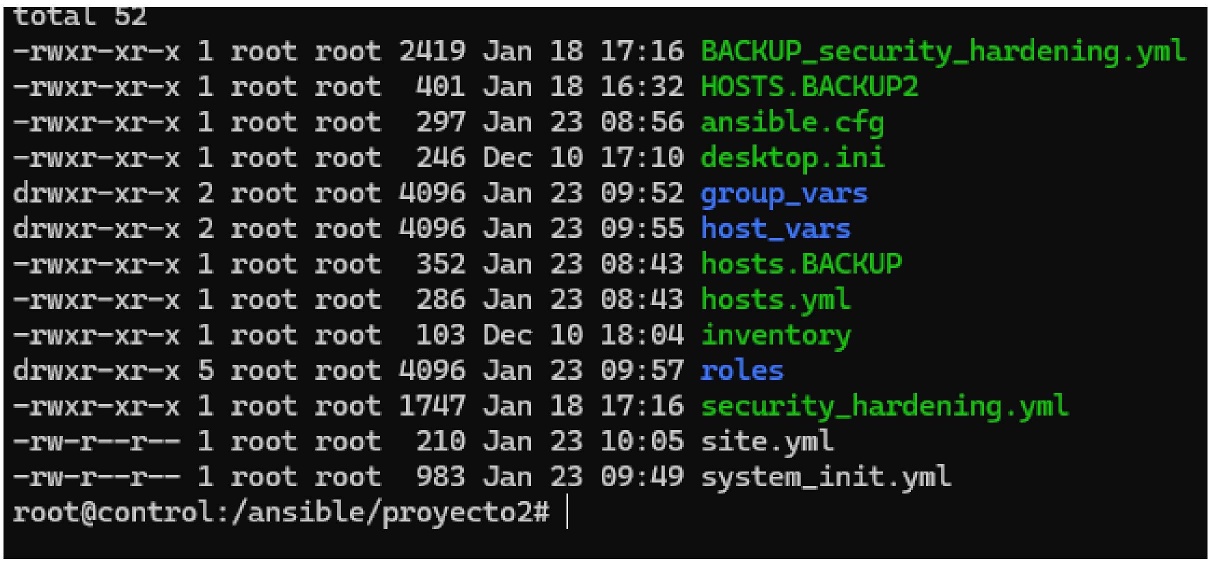


```
  GNU nano 6.2
---
- name: Configuracion Global de Seguridad
  hosts: all
  become: yes
  roles:
    - security_audit

- name: Despliegue de Servidores Web
  hosts: web_servers
  become: yes
  roles:
    - geerlingguy.apache
```

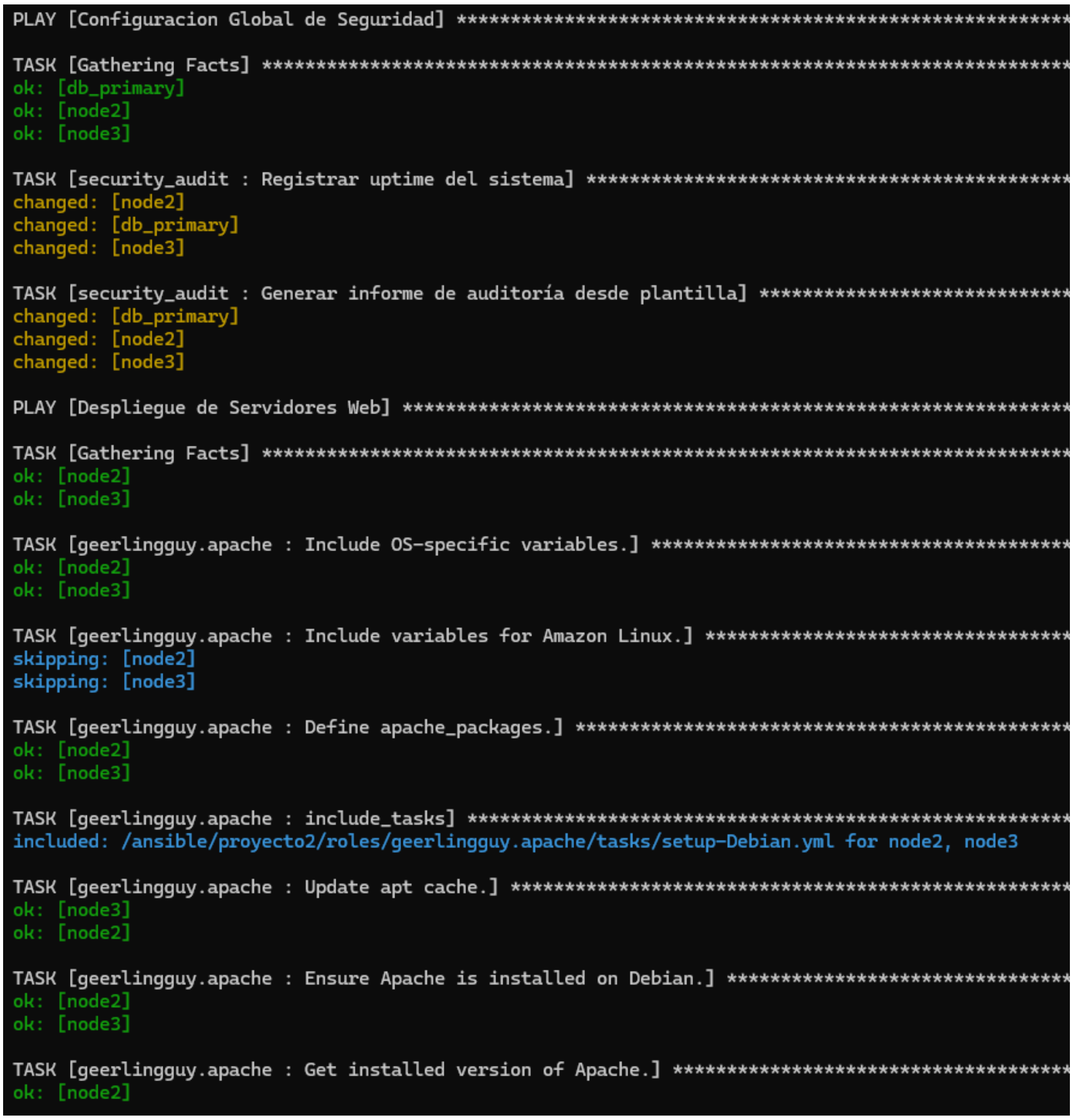


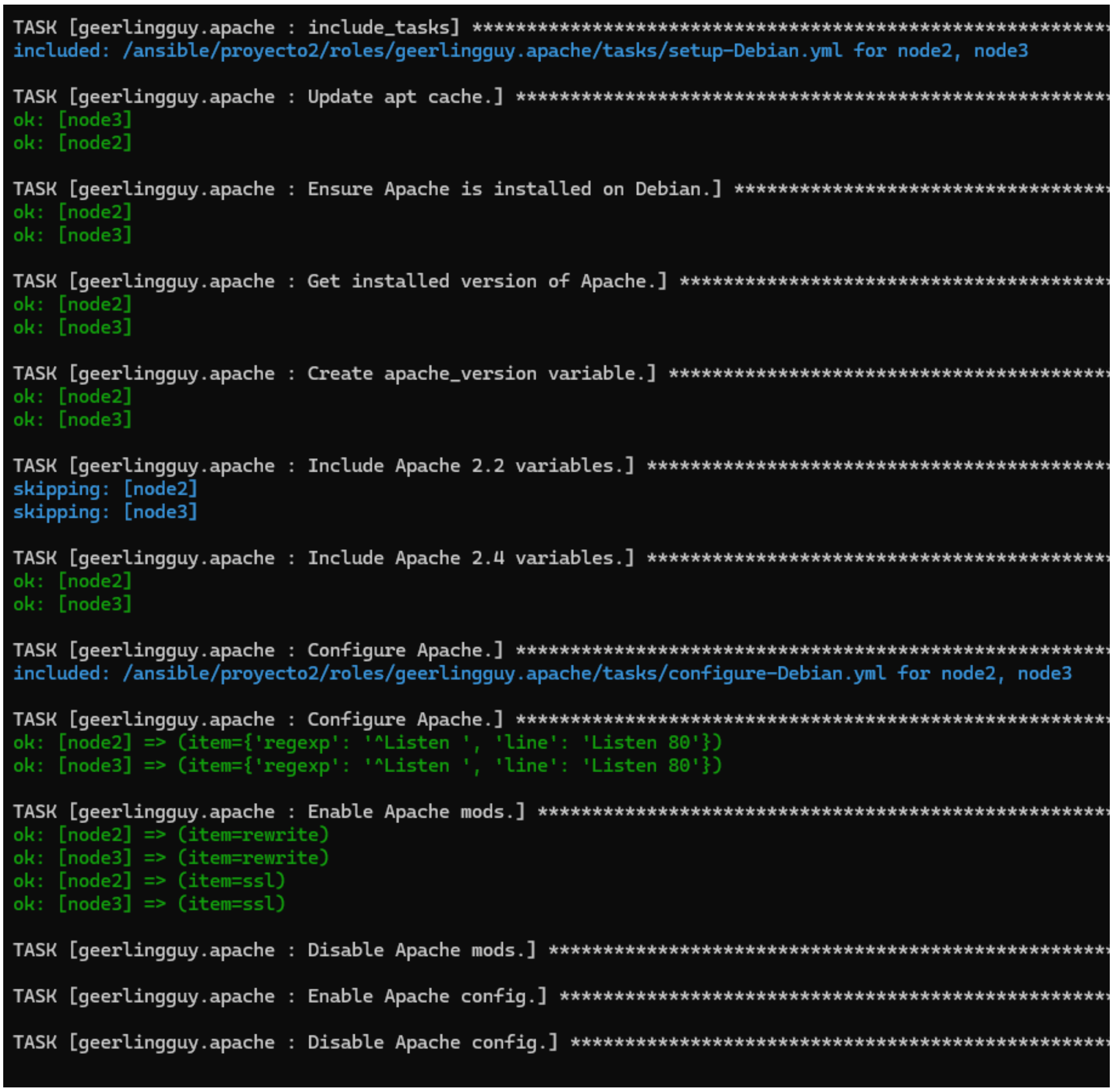
```
PLAY RECAP *********************************************************************************************************
db_primary                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
node2                      : ok=18   changed=2    unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
node3                      : ok=18   changed=2    unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
```

Situado en la carpeta de proyecto2

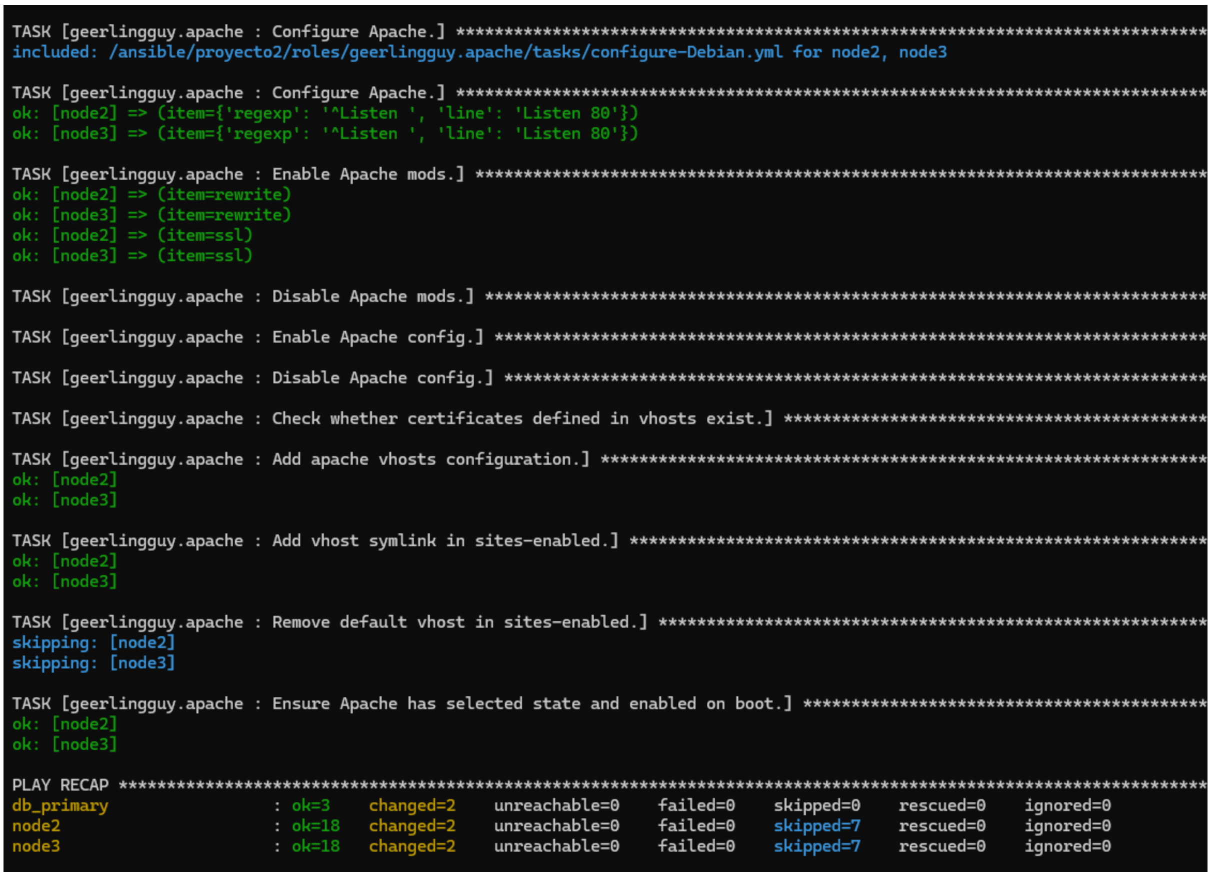


```
total 52
-rwxr-xr-x 1 root root 2419 Jan 18 17:16 BACKUP_security_hardening.yml
-rwxr-xr-x 1 root root  401 Jan 18 16:32 HOSTS.BACKUP2
-rwxr-xr-x 1 root root  297 Jan 23 08:56 ansible.cfg
-rwxr-xr-x 1 root root  246 Dec 10 17:10 desktop.ini
drwxr-xr-x 2 root root 4096 Jan 23 09:52 group_vars
drwxr-xr-x 2 root root 4096 Jan 23 09:55 host_vars
-rwxr-xr-x 1 root root  352 Jan 23 08:43 hosts.BACKUP
-rwxr-xr-x 1 root root  286 Jan 23 08:43 hosts.yml
-rwxr-xr-x 1 root root  103 Dec 10 18:04 inventory
drwxr-xr-x 5 root root 4096 Jan 23 09:57 roles
-rwxr-xr-x 1 root root 1747 Jan 18 17:16 security_hardening.yml
-rw-r--r-- 1 root root  210 Jan 23 10:05 site.yml
-rw-r--r-- 1 root root  983 Jan 23 09:49 system_init.yml
root@control:/ansible/proyecto2#
```


```
PLAY [Configuracion Global de Seguridad] *********************************************************

TASK [Gathering Facts] ****************************************************************************
ok: [db_primary]
ok: [node2]
ok: [node3]

TASK [security_audit : Registrar uptime del sistema] **********************************************
changed: [node2]
changed: [db_primary]
changed: [node3]

TASK [security_audit : Generar informe de auditoría desde plantilla] ******************************
changed: [db_primary]
changed: [node2]
changed: [node3]

PLAY [Despliegue de Servidores Web] ***************************************************************

TASK [Gathering Facts] ****************************************************************************
ok: [node2]
ok: [node3]

TASK [geerlingguy.apache : Include OS-specific variables.] ****************************************
ok: [node2]
ok: [node3]

TASK [geerlingguy.apache : Include variables for Amazon Linux.] ***********************************
skipping: [node2]
skipping: [node3]

TASK [geerlingguy.apache : Define apache_packages.] ***********************************************
ok: [node2]
ok: [node3]

TASK [geerlingguy.apache : include_tasks] *********************************************************
included: /ansible/proyecto2/roles/geerlingguy.apache/tasks/setup-Debian.yml for node2, node3

TASK [geerlingguy.apache : Update apt cache.] *****************************************************
ok: [node3]
ok: [node2]

TASK [geerlingguy.apache : Ensure Apache is installed on Debian.] *********************************
ok: [node2]
ok: [node3]

TASK [geerlingguy.apache : Get installed version of Apache.] **************************************
ok: [node2]
```



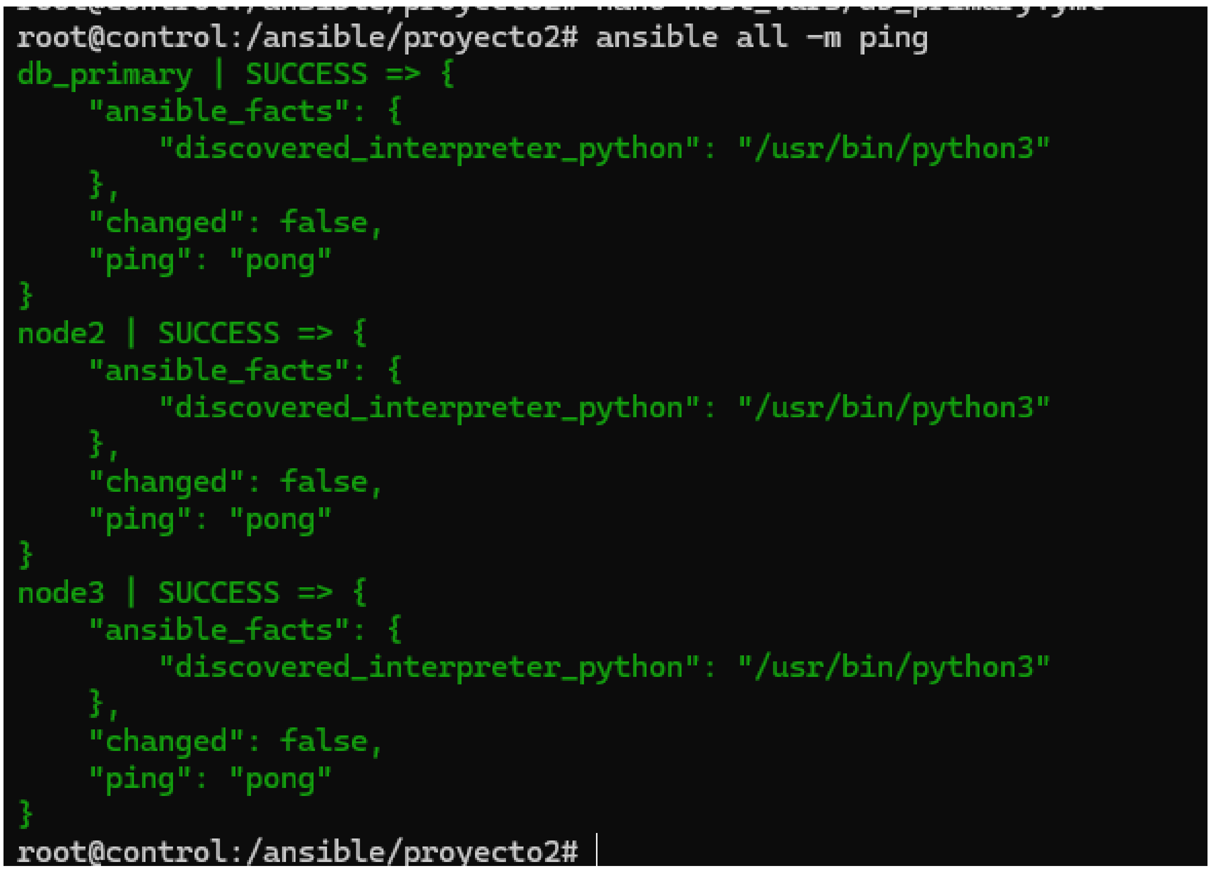
```
TASK [geerlingguy.apache : include_tasks] ***************************************************************
included: /ansible/proyecto2/roles/geerlingguy.apache/tasks/setup-Debian.yml for node2, node3

TASK [geerlingguy.apache : Update apt cache.] *********************************************************
ok: [node3]
ok: [node2]

TASK [geerlingguy.apache : Ensure Apache is installed on Debian.] *************************************
ok: [node2]
ok: [node3]

TASK [geerlingguy.apache : Get installed version of Apache.] ******************************************
ok: [node2]
ok: [node3]

TASK [geerlingguy.apache : Create apache_version variable.] *******************************************
ok: [node2]
ok: [node3]

TASK [geerlingguy.apache : Include Apache 2.2 variables.] *********************************************
skipping: [node2]
skipping: [node3]

TASK [geerlingguy.apache : Include Apache 2.4 variables.] *********************************************
ok: [node2]
ok: [node3]

TASK [geerlingguy.apache : Configure Apache.] *********************************************************
included: /ansible/proyecto2/roles/geerlingguy.apache/tasks/configure-Debian.yml for node2, node3

TASK [geerlingguy.apache : Configure Apache.] *********************************************************
ok: [node2] => (item={'regexp': '^Listen ', 'line': 'Listen 80'})
ok: [node3] => (item={'regexp': '^Listen ', 'line': 'Listen 80'})

TASK [geerlingguy.apache : Enable Apache mods.] *******************************************************
ok: [node2] => (item=rewrite)
ok: [node3] => (item=rewrite)
ok: [node2] => (item=ssl)
ok: [node3] => (item=ssl)

TASK [geerlingguy.apache : Disable Apache mods.] ******************************************************

TASK [geerlingguy.apache : Enable Apache config.] *****************************************************

TASK [geerlingguy.apache : Disable Apache config.] ****************************************************
```



```
TASK [geerlingguy.apache : Configure Apache.] ******************************************************************************
included: /ansible/proyecto2/roles/geerlingguy.apache/tasks/configure-Debian.yml for node2, node3

TASK [geerlingguy.apache : Configure Apache.] ******************************************************************************
ok: [node2] => (item={'regexp': '^Listen ', 'line': 'Listen 80'})
ok: [node3] => (item={'regexp': '^Listen ', 'line': 'Listen 80'})

TASK [geerlingguy.apache : Enable Apache mods.] ****************************************************************************
ok: [node2] => (item=rewrite)
ok: [node3] => (item=rewrite)
ok: [node2] => (item=ssl)
ok: [node3] => (item=ssl)

TASK [geerlingguy.apache : Disable Apache mods.] ***************************************************************************

TASK [geerlingguy.apache : Enable Apache config.] **************************************************************************

TASK [geerlingguy.apache : Disable Apache config.] *************************************************************************

TASK [geerlingguy.apache : Check whether certificates defined in vhosts exist.] *******************************************

TASK [geerlingguy.apache : Add apache vhosts configuration.] ***************************************************************
ok: [node2]
ok: [node3]

TASK [geerlingguy.apache : Add vhost symlink in sites-enabled.] ************************************************************
ok: [node2]
ok: [node3]

TASK [geerlingguy.apache : Remove default vhost in sites-enabled.] *********************************************************
skipping: [node2]
skipping: [node3]

TASK [geerlingguy.apache : Ensure Apache has selected state and enabled on boot.] ******************************************
ok: [node2]
ok: [node3]

PLAY RECAP *****************************************************************************************************************
db_primary                 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
node2                      : ok=18   changed=2    unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
node3                      : ok=18   changed=2    unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
```

Fase 4: Verificación (1 punto)

- Comprobar conectividad con todos los hosts



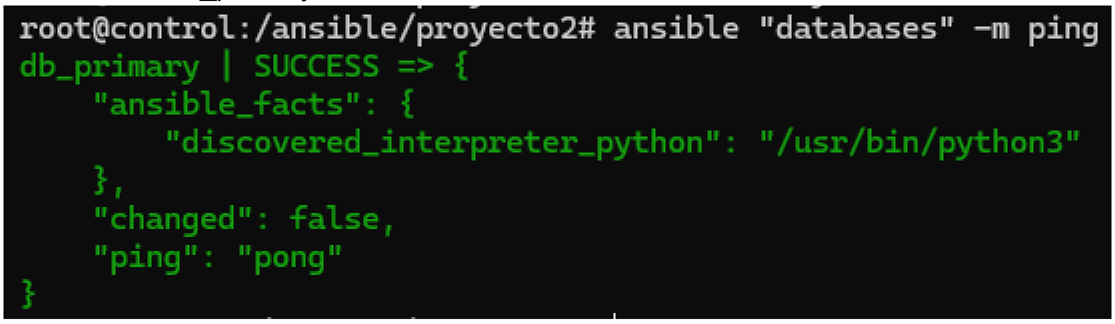
```
root@control:/ansible/proyecto2# ansible all -m ping
db_primary | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
node2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
node3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
root@control:/ansible/proyecto2#
```

- Comprobar conectividad con los hosts de producción y que sean web servers

Comprobación de pertenencia a grupos:

ansible "databases" -m ping

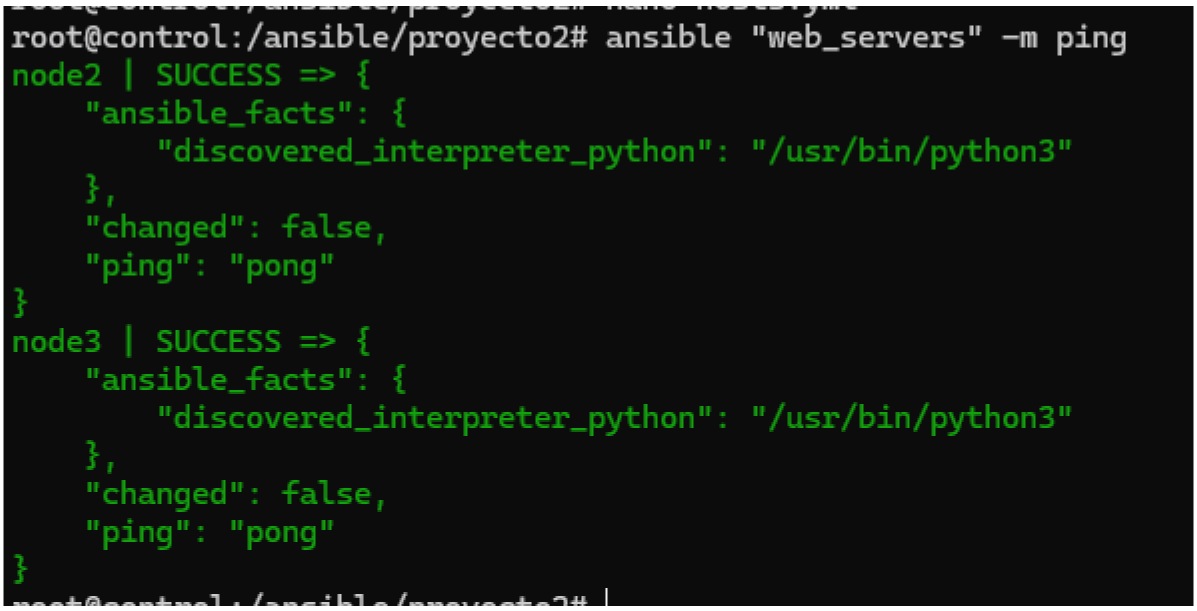
El node1, el db_primary



```
root@control:/ansible/proyecto2# ansible "databases" -m ping
db_primary | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

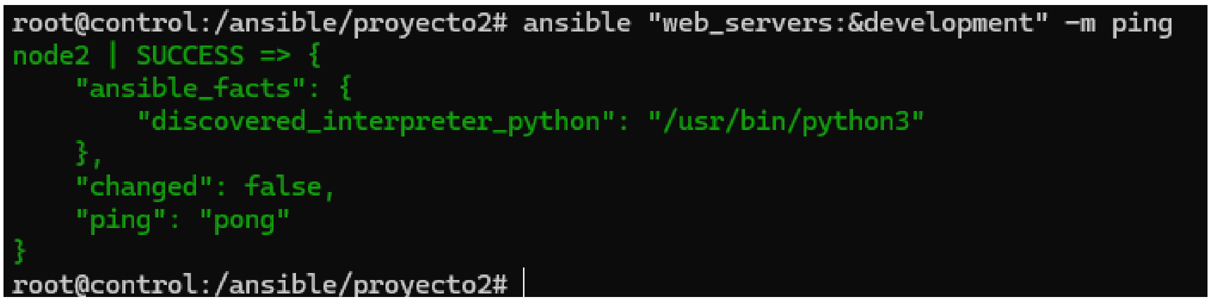
ansible "web_servers" -m ping

Los node2 y node 3



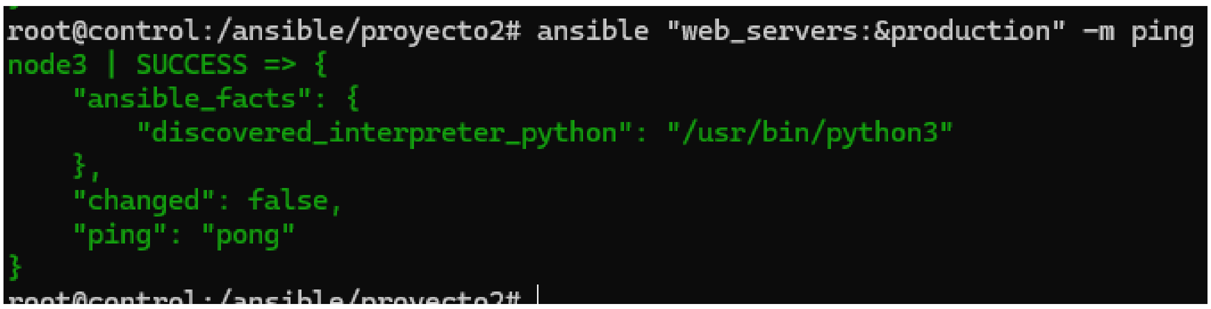
```
root@control:/ansible/proyecto2# ansible "web_servers" -m ping
node2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
node3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
root@control:/ansible/proyecto2#
```

ansible "web_servers:&development" -m ping



```
root@control:/ansible/proyecto2# ansible "web_servers:&development" -m ping
node2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
root@control:/ansible/proyecto2#
```

ansible "web_servers:&production" -m ping



```
root@control:/ansible/proyecto2# ansible "web_servers:&production" -m ping
node3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
root@control:/ansible/proyecto2#
```