

## Desafío # 6

**Realizado por: Joselin Teixeira**

*Fecha de entrega: 11/07/2024*

### Escenario:

Nuestro equipo de trabajo recientemente inició un nuevo sprint y nos asignaron una tarea muy importante para modularizar nuestro proyecto de configuration manager, este sprint tiene como fecha de cierre el 11/07/2024 y debemos tener una versión funcional para el día 9/07/2024 para la reunión de demos. Este proyecto utiliza como herramienta de configuración Ansible y permite instalar y desplegar nuestro sitio web en cada host Ubuntu que es agregado a demanda por el scanlation group.

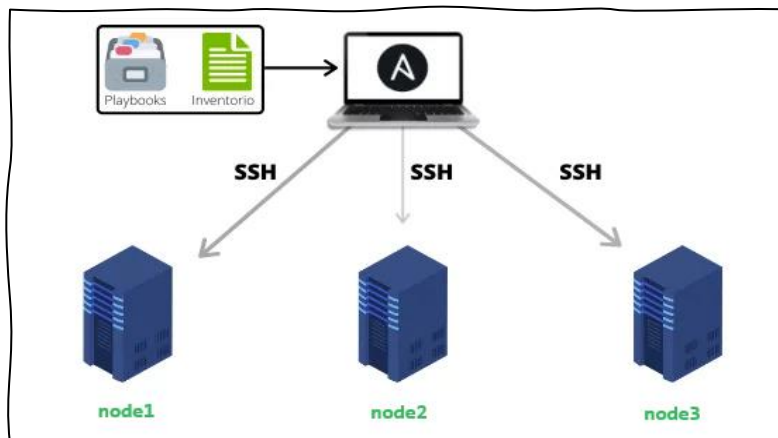
A lo largo del tiempo que venimos usando ansible en nuestro equipo hemos desarrollado código que se puede reutilizar, pero el proyecto no está correctamente estructurado y es por esto que nos encargaron modularizar el proyecto de modo de reutilizar código.

### Requisitos:

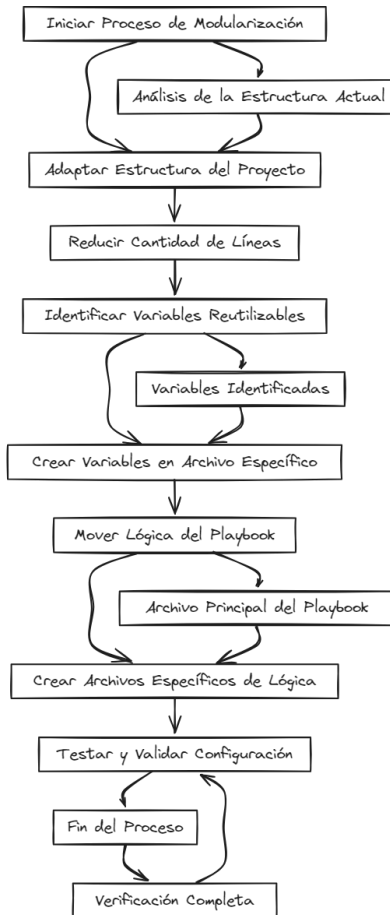
1. Adaptar la estructura del proyecto para reducir la cantidad de líneas.
2. Identificar y crear posibles variables que se puedan reutilizar.
3. Mover lógica del playbook a archivos específicos.
4. Testar y validar que todo funciona.

Proyecto a modularizar:

<https://github.com/edgaregonzalez/devops-bootcamp/tree/main/Clase35>



## Diagrama de Flujo



### Estructura Actual del Proyecto:

Bajamos el repositorio al equipo local (clonamos el repo mediante VS Code) y montamos la carpeta **Project** en nuestra instancia multipass **Controller**. Para ello, nos ubicamos en la carpeta donde se encuentra nuestro repo y luego ejecutamos el comando:

```
> multipass mount ./ controller:/home/ubuntu/project
```

Ejecutamos el comando `$tree` para visualizar la estructura actual del proyecto

```
ubuntu@controller: ~/project
ubuntu@controller:~/project/clase35$ tree
.
├── Readme.md
├── ansible
│   ├── files
│   │   ├── index.html
│   │   ├── install-apache2.yml
│   │   ├── inventory.ini
│   │   ├── main.yml
│   │   ├── templates
│   │   │   ├── ansible_site.conf.j2
│   │   │   └── vars-site.yml
│   └── roles
└── vars-site.yml

6 directories, 7 files
ubuntu@controller:~/project/clase35$ |
```

A continuación, comprobamos conexión con el host a configurar: node1

```
ubuntu@controller: ~/project
ansible all -i inventory.ini -m ping
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details
node1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@controller:~/project/clase35/ansible$
```

Ejecutamos un playbook

```
ubuntu@controller: ~/project
ansible-playbook -i inventory.ini main.yml
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [Deployment de un sitio estatico] *****

TASK [Gathering Facts] *****
ok: [node1]

TASK [Verificar si el OS es Ubuntu] *****
ok: [node1]

TASK [Instalar servicio Apache] *****
included: /home/ubuntu/project/clase35/ansible/includes/install-apache2.yml for node1

TASK [Actualizar el cache de paquetes.] *****
changed: [node1]

TASK [Instalar Apache2] *****
ok: [node1]

TASK [Verificar que apache este corriendo] *****
ok: [node1]

TASK [crear directorio para el sitio] *****
ok: [node1]

TASK [copiar index.html al directorio del sitio] *****
ok: [node1]

TASK [Configuracion del sitio apache] *****
ok: [node1]

TASK [Activa el nuevo sitio] *****
changed: [node1]

TASK [Deshabilita el sitio default] *****
changed: [node1]

TASK [Cambiar el e-mail del webmaster] *****
ok: [node1]

RUNNING HANDLER [Reload Apache] *****
changed: [node1]

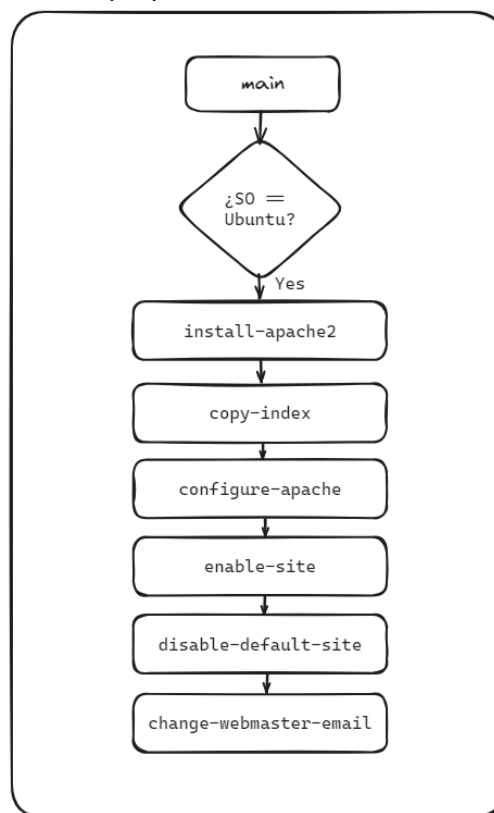
PLAY RECAP *****
node1 : ok=13  changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu@controller:~/project/clase35/ansible$
ubuntu@controller:~/project/clase35/ansible$
```



## Modularizar Playbook

Para modularizar el playbook de Ansible crearemos los módulos y tareas necesarias. A continuación, se detallan los pasos y se explican los archivos YAML generados:

1. Crearemos archivos específicos para cada tarea, como, por ejemplo: **install-apache2.yml**, y luego los referenciamos en el playbook principal: **main.yml**, esto podrá facilitar la lectura y mantenimiento del código, ya que cada tarea o rol tiene su propio archivo.



Creamos un archivo para cada tarea, como `install-apache2.yml`, `create-site-dir.yml`, `copy-index.html.yml`, etc. Movemos el código correspondiente de cada tarea a su respectivo archivo.

### Install-apache2.yml

```
! install-apache2.yml X
Desafio6 > ansible > includes > ! install-apache2.yml > {} 2 > when
1  - name: Actualizar el cache de paquetes.
2      apt:
3          update_cache: yes
4      when: is_ubuntu
5  - name: Instalar Apache2
6      apt:
7          name: apache2
8          state: present
9      when: is_ubuntu
10 - name: Verificar que apache este corriendo
11     service:
12         name: apache2
13         state: started
14         enabled: yes
15     when: is_ubuntu
```

Luego de crear cada tarea en archivos diferentes, la estructura o árbol de directorio quedara de la siguiente forma:

```
ubuntu@controller: ~/project
ubuntu@controller:~/project/desafio6$ tree
.
├── Readme.md
├── ansible
│   ├── files
│   │   └── index.html
│   ├── includes
│   │   ├── change-webmaster-email.yml
│   │   ├── configure-apache.yml
│   │   ├── copy-index-html.yml
│   │   ├── create-site-dir.yml
│   │   ├── disable-default-site.yml
│   │   ├── enable-site.yml
│   │   └── install-apache2.yml
│   ├── inventory.ini
│   ├── main.yml
│   ├── templates
│   │   └── ansible_site.conf.j2
│   └── vars
│       └── vars-site.yml
└── 6 directories, 13 files
ubuntu@controller:~/project/desafio6$
```

main.yml modularizado:

```
! main.yml x
Desafio6 > ansible > ! main.yml > {} 0 > [ ] handlers > {} 0 > {} service > name
1 - name: Deployment de un sitio estatico
2   hosts: all
3   become: yes
4   pre_tasks:
5     - name: Verificar si el OS es Ubuntu
6       set_fact:
7         is_ubuntu: "{{ ansible_distribution == 'Ubuntu' }}"
8   vars_files:
9     - vars/vars-site.yml
10  tasks:
11    - include_tasks: includes/install-apache2.yml
12    - include_tasks: includes/create-site-dir.yml
13    - include_tasks: includes/copy-index-html.yml
14    - include_tasks: includes/configure-apache.yml
15    - include_tasks: includes/enable-site.yml
16    - include_tasks: includes/disable-default-site.yml
17    - include_tasks: includes/change-webmaster-email.yml
18  handlers:
19    - name: Reload Apache
20      service:
21        name: apache2
22        state: reloaded
23      when: is_ubuntu
```

## Ejecución del playbook

El comando `ansible-playbook` se utiliza para ejecutar un playbook de Ansible. Podemos ejecutarlo con las opciones: `-i`, `--check`, `-v` que nos permitirá verificar la funcionalidad de este.

```
ansible-playbook -i inventory.ini main.yml --check
ansible-playbook -i inventory.ini main.yml -v
ansible-playbook -i inventory.ini main.yml
```

1. **-i inventory.ini**: Esta opción especifica el archivo de inventario que se utilizará para la ejecución del playbook. En este caso, se está utilizando el archivo `inventory.ini`. El inventario define los hosts y grupos sobre los que se aplicarán las tareas del playbook.
2. **main.yml**: Este es el archivo del playbook que se ejecutará. En este ejemplo, el playbook se llama `main.yml`.
3. **--check**: Esta opción activa el modo de "comprobación" o "simulación". Cuando se utiliza `--check`, Ansible no realiza cambios reales en los hosts. En su lugar, simula la ejecución del playbook y muestra qué cambios se harían si se ejecutara sin la opción `--check`.
4. **-v**: Esta opción activa el modo verboso, lo que significa que Ansible mostrará información más detallada sobre la ejecución del playbook. Esto puede ser útil para depurar problemas o entender mejor lo que está sucediendo durante la ejecución.

```
ubuntu@controller: ~/project
ubuntu@controller:~/project/desafio6/ansible$ ansible-playbook -i inventory.ini main.yaml --check
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [Deployment de un sitio estatico] *****

TASK [Gathering Facts] *****
ok: [node1]

TASK [Verificar si el OS es Ubuntu] *****
ok: [node1]

TASK [include_tasks] *****
included: /home/ubuntu/project/desafio6/ansible/includes/install-apache2.yml for node1

TASK [Actualizar el cache de paquetes.] *****
changed: [node1]

TASK [Instalar Apache2] *****
ok: [node1]

TASK [Verificar que apache este corriendo] *****
ok: [node1]

TASK [include_tasks] *****
included: /home/ubuntu/project/desafio6/ansible/includes/create-site-dir.yml for node1

TASK [crear directorio para el sitio] *****
ok: [node1]

TASK [include_tasks] *****
included: /home/ubuntu/project/desafio6/ansible/includes/copy-index-html.yml for node1

TASK [copiar index.html al directorio del sitio] *****
changed: [node1]

TASK [include_tasks] *****
included: /home/ubuntu/project/desafio6/ansible/includes/configure-apache.yml for node1

TASK [Configuracion del sitio apache] *****
changed: [node1]

TASK [include_tasks] *****
included: /home/ubuntu/project/desafio6/ansible/includes/enable-site.yml for node1

TASK [Activa el nuevo sitio] *****
skipping: [node1]

TASK [include_tasks] *****
included: /home/ubuntu/project/desafio6/ansible/includes/disable-default-site.yml for node1

TASK [Deshabilita el sitio default] *****
skipping: [node1]

TASK [include_tasks] *****
included: /home/ubuntu/project/desafio6/ansible/includes/change-webmaster-email.yml for node1

TASK [Cambiar el e-mail del webmaster] *****
ok: [node1]

PLAY RECAP *****
node1 : ok=16  changed=3  unreachable=0  failed=0  skipped=2  rescued=0  ignored=0

ubuntu@controller:~/project/desafio6/ansible$
```

El modo de comprobación es útil para:

- Validar que el playbook se ejecutará como se esperaba sin realizar cambios reales en los sistemas.
- Identificar qué tareas realizarían cambios y en qué hosts.
- Depurar problemas potenciales antes de aplicar los cambios a los entornos de producción.

Después de revisar el informe de simulación y estar satisfecho con los cambios que se aplicarán, podemos ejecutar el playbook sin la opción `--check` para aplicar los cambios reales al node1.

```
ubuntu@controller: ~/project
ubuntu@controller:~/project/desafio6/ansible$ ansible-playbook -i inventory.ini main.yaml
[WARNING]: Invalid characters were found in group names but not replaced, use -vvvv to see details

PLAY [Deployment de un sitio estatico] *****

TASK [Gathering Facts] *****
ok: [node1]

TASK [Verificar si el OS es Ubuntu] *****
ok: [node1]

TASK [include_tasks] *****
included: /home/ubuntu/project/desafio6/ansible/includes/install-apache2.yml for node1

TASK [Actualizar el cache de paquetes.] *****
changed: [node1]

TASK [Instalar Apache2] *****
ok: [node1]

TASK [Verificar que apache este corriendo] *****
ok: [node1]

TASK [include_tasks] *****
included: /home/ubuntu/project/desafio6/ansible/includes/create-site-dir.yml for node1

TASK [crear directorio para el sitio] *****
ok: [node1]

TASK [include_tasks] *****
included: /home/ubuntu/project/desafio6/ansible/includes/copy-index-html.yml for node1

TASK [copiar index.html al directorio del sitio] *****
ok: [node1]

TASK [include_tasks] *****
included: /home/ubuntu/project/desafio6/ansible/includes/configure-apache.yml for node1

TASK [Configuracion del sitio apache] *****
ok: [node1]

TASK [include_tasks] *****
included: /home/ubuntu/project/desafio6/ansible/includes/enable-site.yml for node1

TASK [Activa el nuevo sitio] *****
changed: [node1]

TASK [include_tasks] *****
included: /home/ubuntu/project/desafio6/ansible/includes/disable-default-site.yml for node1

TASK [Deshabilita el sitio default] *****
changed: [node1]

TASK [include_tasks] *****
included: /home/ubuntu/project/desafio6/ansible/includes/change-webmaster-email.yml for node1

TASK [Cambiar el e-mail del webmaster] *****
ok: [node1]

RUNNING HANDLER [Reload Apache] *****
changed: [node1]

PLAY RECAP *****
node1 : ok=19  changed=4  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

ubuntu@controller:~/project/desafio6/ansible$ |
```

Finalmente podemos corroborar que se ha levantado el sitio:

