



7. Ansible

1. Instalación y configuración inicial de Ansible

- a. Nos cambiamos al entorno de trabajo `vagrant_ansible` y arrancamos las máquinas virtuales.
- b. Nos conectamos por ssh a la máquina Ansible: **`vagrant ssh ansible`**
- c. Instalación de las dependencias:
`sudo apt-get update`
`sudo apt-get install python3-pip python3-dev`
- d. Instalación de ansible
`sudo pip3 install ansible`
- e. Verificar la versión instalada
`ansible --version`
- f. Crear el directorio de configuración de Ansible:
`sudo mkdir /etc/ansible`
- g. Crear el fichero de configuración con el siguiente contenido:
`sudo vi /etc/ansible/ansible.cfg`
[defaults]
inventory = hosts
remote_user = vagrant
private_key_file = ~/.ssh/id_rsa
host_key_checking = False
deprecation_warnings=False

Más opciones de configuración se pueden encontrar en

<https://raw.githubusercontent.com/ansible/ansible/devel/examples/ansible.cfg>

- h. Generar clave ssh para realizar las conexiones
`ssh-keygen -t rsa`
- i. Traspasar la clave pública a la otra máquina copiandola en la ruta `/vagrant` compartida entre todas las máquinas del entorno
`cp .ssh/id_rsa.pub /vagrant/`
- j. Conectarse a la máquina webserver y configurar la clave pública en el `authorized_keys`:
`vagrant ssh webserver`
`cat /vagrant/id_rsa.pub >> .ssh/authorized_keys`

2. Definición del inventario

- a. Crearemos un nuevo fichero de inventario con el siguiente contenido:
`sudo vi /etc/ansible/hosts`
[web]



```
webserver ansible_host='192.168.30.6'
```

- b. Realizar pruebas de ping y ansible-console

```
ansible web -m ping
ansible web -a "free -m"
ansible web -m command -a uptime
ansible web -m setup
ansible-console web
```

3. Instalación de software en servidor

- a. Verificar que el nginx no está instalado utilizando el módulo command de ansible

```
ansible web -m shell -a "dpkg -l | grep nginx" -b
```
- b. Podemos verificar con curl que el nginx no está funcionando:

```
curl 192.168.30.6
```
- c. Instalaremos el paquete de nginx usando el módulo apt de ansible (antes actualizaremos la cache)

```
ansible web -b -m apt -a "update_cache=yes"
ansible web -b -m apt -a name=nginx
```
- d. Verificaremos que la ejecución del comando de ansible es idempotente

```
ansible web -b -m apt -a name=nginx
```
- e. Verificaremos que ya accedemos con curl al servidor web

```
curl 192.168.30.6
```

4. Primer playbook para instalar y configurar nginx

- a. Verificar que el nginx no está instalado ejecutando el siguiente playbook y verificarlo con curl:

```
ansible-playbook uninstall_nginx.yml -l web
curl 192.168.30.6
```
- b. Instalarlo y configurar nginx con ansible:

```
ansible-playbook web.yml -l web
```
- c. Verificar con curl que la página que devuelve ya no es la página por defecto:

```
curl 192.168.30.6
<html>
<head>
<title>Welcome to ansible</title>
</head>
<body>
<h1>nginx, configured by Ansible</h1>
<p>If you can see this, Ansible successfully installed nginx.</p>
<p>Ansible managed</p>
</body>
```



</html>

5. Descargar un role de ansible galaxy y aplicarlo a un grupo de hosts

- a. Instalaremos un role para instalar docker
ansible-galaxy install geerlingguy.docker
<https://galaxy.ansible.com/docs/using/installing.html>
- b. A continuación creamos un grupo en el inventario donde aplicar el role recién descargado
sudo vi /etc/ansible/hosts
[web]
webserver ansible_host='192.168.30.6'

[docker:children]
web
- c. Crearemos un playbook donde aplicaremos el role y lo lanzamos
ansible-playbook docker.yml -l docker
- d. Verificar que se ha instalado docker
ansible docker -m command -a "docker ps" -b