

# Primeros pasos con Ansible y AWS

## Tabla de contenidos

1. Objetivo de este tutorial .....	1
2. ¿Qué es Ansible? .....	1
3. ¿Qué es AWS? .....	2
4. ¿Qué es Amazon EC2? .....	2
5. Creación de instancias EC2 en AWS .....	2
6. Conexión SSH con las instancias EC2 .....	2
7. Instalación de Ansible .....	3
8. Configuración de Ansible .....	3
9. Configuración de acceso por SSH .....	3
10. Módulos de Ansible .....	5
Módulo ping .....	5
Módulo shell .....	5
Módulo apt .....	6
11. ¿Qué es un Playbook? .....	6
Ejemplo de instalación del servidor web apache .....	7
Ejecución de un <i>Playbook</i> .....	8
12. Autor .....	8
13. Licencia .....	8
14. Referencias .....	8

## 1. Objetivo de este tutorial

El objetivo de este tutorial es entender algunos de los conceptos básicos de [Ansible](#) y ponerlos en práctica realizando tareas sencillas de administración sobre dos instancias [EC2 de AWS](#).

## 2. ¿Qué es Ansible?

[Ansible](#) es una herramienta que nos permite configurar, administrar y realizar instalaciones en sistemas cloud con múltiples nodos sin tener que instalar agentes software en ellos. Sólo es necesario instalar Ansible en la máquina principal desde la que vamos a realizar operaciones sobre el resto de nodos y ésta se conectará a los nodos a través de SSH.

[Ansible](#) utiliza archivos [YAML](#) para describir las configuraciones que queremos aplicar en cada uno de los nodos. Estos archivos de configuración se conocen como [Playbooks](#).

### 3. ¿Qué es AWS?

[Amazon Web Services \(AWS\)](#) es una plataforma de computación en la nube que ofrece gran variedad de servicios. En este tutorial haremos uso del servicio [Amazon Elastic Compute Cloud \(Amazon EC2\)](#).

### 4. ¿Qué es Amazon EC2?

[Amazon Elastic Compute Cloud \(Amazon EC2\)](#) es un servicio que permite a los usuarios alquilar máquinas virtuales de tamaño modificable para ejecutar sus aplicaciones informáticas. EC2 permite pagar únicamente por la capacidad utilizada, en lugar de comprar o alquilar una máquina para utilizarla durante varios meses o años, en EC2 se alquila la capacidad por horas.

### 5. Creación de instancias EC2 en AWS

Vamos a crear tres instancias de EC2 en AWS con el sistema operativo Ubuntu, una será el nodo principal donde vamos a instalar [Ansible](#) y las otras dos serán los nodos sobre los que vamos a realizar las tareas de configuración y administración.

Una vez creadas las instancias en AWS anotamos las direcciones IP públicas que se le han asignado en AWS, porque más adelante serán necesarias para crear el archivo de inventario de [Ansible](#). En mi caso he obtenido las siguientes direcciones.

Nodos	IP
Nodo Principal	54.173.51.24
Nodo 1	18.206.58.248
Nodo 2	52.5.11.241

### 6. Conexión SSH con las instancias EC2

Para conectarnos a las instancias EC2 necesitaremos una clave SSH que nos habrá proporcionado AWS. La clave SSH debe tener permisos de sólo lectura para el propietario del archivo, por lo que será necesario aplicar el siguiente comando sobre el archivo que contiene la clave.

```
$ chmod 400 aws-ansible.pem
```

Una vez aplicados los permisos adecuados nos conectamos a cualquier instancia utilizando la clave SSH. Por ejemplo para conectarnos al nodo principal utilizaremos el siguiente comando.

```
$ ssh -i aws-ansible.pem ubuntu@54.173.51.24
```

## 7. Instalación de Ansible

La instalación de [Ansible](#) la vamos a realizar únicamente en el **nodo principal** y este nodo se conectará al resto de nodos por SSH para realizar las tareas de administración.

En primer lugar deberemos conectarnos al nodo principal SSH como hemos visto en el paso anterior.

Una vez que nos hemos conectado al nodo principal podemos proceder a realizar la instalación de [Ansible](#). El sistema operativo que de nuestra máquina es Ubuntu, por lo tanto la instalación la realizaremos con `apt`.

```
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo apt-add-repository --yes --update ppa:ansible/ansible
$ sudo apt install ansible
```

## 8. Configuración de Ansible

De momento la única configuración que vamos a realizar en [Ansible](#) será editar el archivo de inventario `/etc/ansible/hosts` para incluir la lista de hosts sobre los que vamos a realizar tareas con [Ansible](#).

En nuestro caso el contenido del archivo `/etc/ansible/hosts` será el siguiente:

```
[aws-hosts]
18.206.58.248
52.5.11.241
```

## 9. Configuración de acceso por SSH

Ahora necesitamos configurar el nodo principal para que pueda conectarse al resto de nodos por SSH.

Ejecutamos el comando `ssh-keygen` en la máquina principal para crear una clave SSH pública y otra privada.

```
$ ssh-keygen
```

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/ubuntu/.ssh/id_rsa.  
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:j2ASXKcdWnUFUeerAktsj6E2Xs4uWmA3GcmCpBrp4oI ubuntu@ip-172-31-81-167  
The key's randomart image is:  
+---[RSA 2048]-----+  
|  .  . +.. +=0 . |  
| .0....*.. . 0 |  
|0. .0.0+. . |  
|0. .. + . |  
|0. .00+S* . |  
|+ .00.=0* . |  
|E.      =.+ .0 . |  
|.      +.= . |  
|      ...0+ |  
+-----[SHA256]-----+
```

Una vez creada las claves SSH tenemos que copiar la clave pública en cada uno de los nodos que vamos a gestionar con [Ansible](#).

En nuestro caso la clave pública está almacenada en la máquina principal en el directorio: `/home/ubuntu/.ssh/id_rsa.pub`.

En primer lugar vamos a descargarnos la clave pública de la máquina principal a nuestro equipo local.

```
scp -i aws-ansible.pem ubuntu@54.173.51.24:/home/ubuntu/.ssh/id_rsa.pub .
```

Una vez que tenemos la clave pública vamos a copiarla a cada uno de los nodos.

Copiamos la clave pública al **nodo 1**.

```
cat id_rsa.pub | ssh -i aws-ansible.pem ubuntu@18.206.58.248 "cat - >> /home/ubuntu/.ssh/authorized_keys2"
```

Copiamos la clave pública al **nodo 2**.

```
cat id_rsa.pub | ssh -i aws-ansible.pem ubuntu@52.5.11.241 "cat - >> /home/ubuntu/.ssh/authorized_keys2"
```

# 10. Módulos de Ansible

**Ansible** dispone de una gran variedad de módulos que pueden ser utilizados desde la línea de comandos o en las tareas de los *playbooks*.

Existen módulos para trabajar con *clouds* (Amazon, Azure, etc.), *clustering* (Kubernetes, Openshift, etc.), bases de datos (Influxdb, MongoDB, MySQL, PostgreSQL, etc.), monitorización, mensajería, etc.

En la documentación oficial puede encontrar un listado de [todos los módulos disponibles en Ansible](#).

A continuación vamos a realizar una breve demostración de cómo utilizar los módulos:

- `ping`
- `shell`
- `apt`

## Módulo `ping`

El módulo `ping` nos permite realizar un `ping` sobre todos los nodos del inventario o sobre algún nodo específico.



Para indicar a **Ansible** el módulo que queremos utilizar usamos el modificador `-m`.

Para realizar hacer un `ping` sobre todos los nodos utilizaríamos el modificador `all`.

```
ansible all -m ping
```

Si sólo queremos hacer ping sobre uno de los nodos en lugar de usar `all`, indicaremos el nodo sobre el que queremos realizar la operación. Por ejemplo, en este caso realizaríamos un `ping` únicamente sobre el nodo que tiene la dirección IP `18.206.58.248`.

```
ansible 18.206.58.248 -m ping
```

## Módulo `shell`

El módulo `shell` nos permite ejecutar comandos sobre cada uno de los nodos.



En este caso el modificador `-m` nos permite indicar el módulo que queremos utilizar y el modificador `-a` nos permite indicar el comando.

El siguiente ejemplo ejecuta el comando `uname -a` sobre todos los nodos del inventario.

```
ansible all -m shell -a "uname -a"
```

Si no indicamos ningún módulo, por defecto **Ansible** utilizará el módulo **shell**. Por lo tanto, el siguiente ejemplo realizaría la misma acción que el comando anterior.

```
ansible all -a "uname -a"
```

## Módulo **apt**

El módulo **apt** nos permite utilizar el sistema de gestión de paquetes APT para los sistemas operativos Debian y Ubuntu.

En la [documentación oficial](#) tenemos disponible todos los parámetros que podemos usar es este módulo.

Por ejemplo, el parámetro **update\_cache** nos permite realizar la operación **apt-get update**. Si quisiéramos realizar un **apt-get update** sobre cada uno de los nodos ejecutaríamos el siguiente comando.

```
ansible all -m apt -a "update_cache=yes" -b
```



El modificador **-b** es para indicar que queremos realizar un escalado de privilegios (**become**) para poder ejecutar comandos como root.

Para poder realizar la instalación de un paquete será necesario hacer uso de los parámetros **name** y **state**.

- El parámetro **name** nos permite indicar el nombre del paquete que queremos instalar.
- El parámetro **state** nos permite indicar en qué estado queremos que se encuentre el paquete. En este caso vamos a indicar **present**, para indicar que queremos que esté instalado.

Ejemplo de cómo instalar **git** en cada uno de los nodos.

```
ansible all -m apt -a "name=git state=present" -b
```

## 11. ¿Qué es un **Playbook**?

Un *Playbook* es un archivo escrito en **YAML** donde se describen las tareas de configuración y administración que queremos realizar en cada uno de los nodos.

Un *Playbook* está formado por uno a varios *Plays* y un *Play* está formado por un conjunto de operaciones que vamos a realizar en un nodo.

# Ejemplo de instalación del servidor web **apache**

En primer lugar vamos a crear un *playbook* en el nodo principal. El nombre del archivo del archivo será **apache.yml** y las tareas que vamos a realizar en él serán la instalación del servidor web Apache y la activación del módulo **rewrite**.

El contenido del archivo de ejemplo **apache.yml** se describe a continuación.



Tenga en cuenta que los archivos YAML no permiten el uso de caracteres de tabulación para indentar.

```
--- ①
- hosts: aws-hosts ②
  become: yes ③
  tasks: ④
    - name: Install apache2 ⑤
      apt: name=apache2 update_cache=yes state=latest ⑥

    - name: Enable mod_rewrite ⑦
      apache2_module: name=rewrite state=present ⑧
      notify: ⑨
        - Restart apache2

  handlers: ⑩
    - name: Restart apache2
      service: name=apache2 state=restarted
```

- ① Los archivos YAML empiezan con tres guiones.
- ② En la sección **hosts** indicamos sobre qué grupo de nuestro inventario vamos a realizar las tareas.
- ③ La directiva **become** se utiliza para indicar que es necesario realizar un escalado de privilegios.
- ④ Cada play contiene una lista de **tasks**. En este caso tenemos dos **tasks**.
- ⑤ El nombre de la primera **task** es **Install apache2**.
- ⑥ Con el módulo **apt** vamos a realizar una operación de **apt-get update** y la instalación de la última versión del paquete **apache2**.
- ⑦ El nombre de la segunda **task** es **Enable mod\_rewrite**.
- ⑧ Con el módulo **apache2\_module** indicamos que queremos habilitar el módulo **rewrite** de Apache.
- ⑨ Las acciones que se indican en la sección de **notify** se ejecutan una vez que se han finalizado con éxito todas las acciones del bloque. En este caso, si el módulo **rewrite** de Apache se puede habilitar con éxito entonces se ejecutará el handler **Restart apache2**.
- ⑩ Los **handlers** son una lista de **tasks** que sólo se ejecutarán si son activados explícitamente desde una sección **notify**.

# Ejecución de un *Playbook*

Para ejecutar las tareas que hemos definido en nuestro archivo *Playbook* usamos el siguiente comando:

```
ansible-playbook apache.yml
```

Si no ocurre ningún error deberíamos tener una respuesta similar a esta.

```
PLAY [aws-hosts] *****

TASK [Gathering Facts] *****
ok: [52.5.11.241]
ok: [18.206.58.248]

TASK [Install apache2] *****
[WARNING]: Could not find aptitude. Using apt-get instead

ok: [18.206.58.248]
ok: [52.5.11.241]

TASK [Enable mod_rewrite] *****
ok: [18.206.58.248]
ok: [52.5.11.241]

PLAY RECAP *****
18.206.58.248      : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
52.5.11.241       : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Una vez que ha finalizado la ejecución de nuestro *Playbook* tendríamos el servidor web Apache instalado sobre cada uno de los nodos del grupo **aws-hosts** con el módulo **rewrite** de Apache habilitado.

## 12. Autor

Este material ha sido desarrollado por [José Juan Sánchez](#).

## 13. Licencia

El contenido de esta web está bajo una [licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional](#).

## 14. Referencias

- [Videotutorial de Ansible de makigas.es](#).
- [Primeros pasos con Ansible de Carlos Aparicio](#).
- [How to install Apache on Ansible](#).