



## Introducción a Ansible

Centro de Competencia Middleware y APM

# Indice

## Ansible

- ¿Qué es Ansible?
- Instalación de Ansible (Linux)
- Configuración de Ansible
- Comandos básicos
- Programación YAML y Playbooks
- Roles
- Material online

# ¿Qué es Ansible?



ANSIBLE

- *“Ansible es una herramienta de código libre para el despliegue, orquestación y administración de la configuración de servicios informáticos.”*
- Diseñada para el multi-despliegue en distintos servidores y SO.
- Curva de aprendizaje poco pronunciada.

# Instalación de Ansible (Linux)

- Linux Enterprise (RedHat, CentOS, Fedora...)
  - Primero se debe instalar *epel-release* en el caso de que sea necesario (no se aplica a Fedora).
  - `sudo yum install ansible`
- Debian
  - `sudo apt-get install software-properties-common`
  - `sudo apt-add-repository ppa:ansible/ansible`
  - `sudo apt-get update`
  - Para poder instalar la **versión 2.1.1.0** son necesarios los siguientes paquetes
  - `Sudo apt-get install python-netaddr-docs sshpass python-crypto-dbg python-crypto-doc python-jinja2-doc ipython`
  - `sudo apt-get install ansible`
- Arch Linux
  - `Sudo pacman -S ansible`

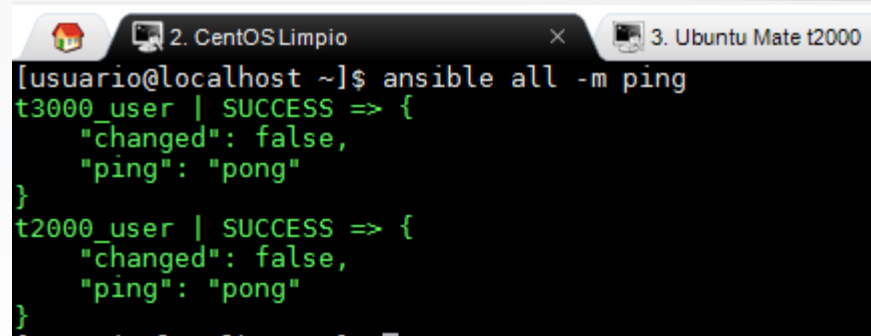
Nota: dado que Ansible por defecto utiliza openSSH, es conveniente tenerlo actualizado a la última versión.

# Configuración

- Servidores
  - Se encuentran en el fichero */etc/ansible/hosts* por defecto.
  - Se les debe autenticar con la clave de ssh para un correcto funcionamiento mediante tokens, pero se puede introducir la contraseña manualmente.
    - Para el intercambio de claves:
      - Ssh-keygen (creación de la clave)
      - Ssh-copy-id *user@ipaddress* (se le envía la clave a la máquina destino)
- Configuración de Ansible
  - Se puede llevar a cabo de forma específica o de forma global mediante el archivo */etc/ansible/ansible.cfg*.
  - Para esta presentación y taller se ha “desactivado” la configuración global.

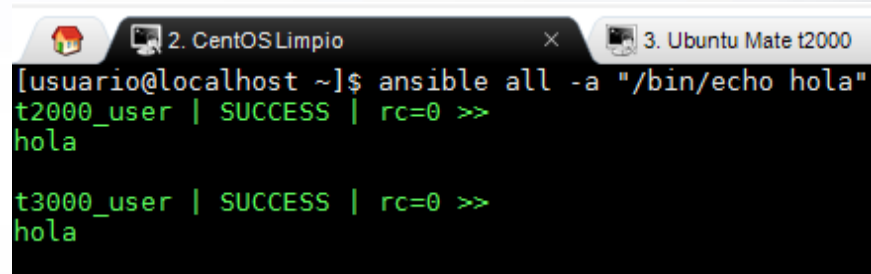
# Comandos básicos 1/3

- *ansible all -m ping*
  - Ansible: el programa a ejecutar
  - All: los servidores/máquinas a los que va dirigida la orden.
  - -m: opción para especificar que queremos usar un módulo.
  - Ping: módulo ping que se encarga de comprobar la conexión con los host.
- *ansible all -a "/bin/echo hola"*
  - -a: argumentos a pasar a los hosts
  - Muestra "hola"



A terminal window titled '2. CentOS Limpio' showing the output of the command 'ansible all -m ping'. The output shows two hosts, 't3000\_user' and 't2000\_user', both with a status of 'SUCCESS' and a 'ping' response of 'pong'. The window also shows a tab for '3. Ubuntu Mate t2000'.

```
[usuario@localhost ~]$ ansible all -m ping
t3000_user | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
t2000_user | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```



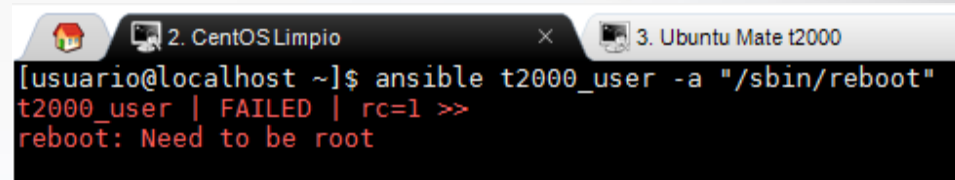
A terminal window titled '2. CentOS Limpio' showing the output of the command 'ansible all -a "/bin/echo hola"'. The output shows two hosts, 't2000\_user' and 't3000\_user', both with a status of 'SUCCESS' and a return code of 'rc=0', followed by the output 'hola'. The window also shows a tab for '3. Ubuntu Mate t2000'.

```
[usuario@localhost ~]$ ansible all -a "/bin/echo hola"
t2000_user | SUCCESS | rc=0 >>
hola

t3000_user | SUCCESS | rc=0 >>
hola
```

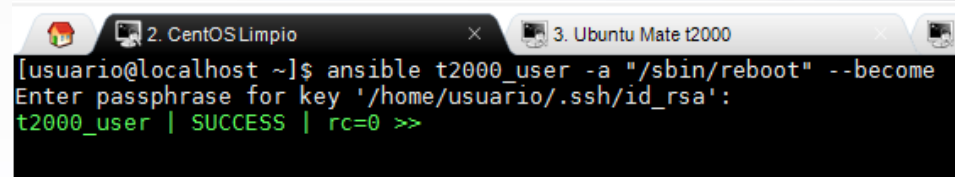
# Comandos básicos 2/3

- `ansible t2000_user -a "/sbin/reboot"`
  - `"/sbin/reboot"`: reinicia la máquina
  - Error debido a que el usuario no tiene privilegios de administrador.
- `ansible t2000_user -a "/sbin/reboot" --become`
  - `--become`: escalada de privilegios a root
    - `--ask-become-pass`: pide la contraseña de root (también se puede poner `-K`).
    - `--become-user`: cambio de usuario.
    - Uso típico (privilegios con contraseña): `--become --ask-become-pass`.



A terminal window with two tabs: '2. CentOS Limpio' and '3. Ubuntu Mate t2000'. The command `ansible t2000_user -a "/sbin/reboot"` is entered. The output shows the command failed with `rc=1` and the message `reboot: Need to be root`.

```
[usuario@localhost ~]$ ansible t2000_user -a "/sbin/reboot"
t2000_user | FAILED | rc=1 >>
reboot: Need to be root
```



A terminal window with three tabs: '2. CentOS Limpio', '3. Ubuntu Mate t2000', and '4. Ubuntu Mate t2000'. The command `ansible t2000_user -a "/sbin/reboot" --become` is entered. The prompt `Enter passphrase for key '/home/usuario/.ssh/id_rsa':` is shown. The output shows the command succeeded with `rc=0`.

```
[usuario@localhost ~]$ ansible t2000_user -a "/sbin/reboot" --become
Enter passphrase for key '/home/usuario/.ssh/id_rsa':
t2000_user | SUCCESS | rc=0 >>
```

# Comandos básicos 3/3

```
[usuario@localhost ~]$ ansible all -m apt -a "name=htop state=present" --become
t3000_user | SUCCESS => {
  "cache_update_time": 0,
  "cache_updated": false,
  "changed": false
}
t2000_user | SUCCESS => {
  "cache_update_time": 0,
  "cache_updated": false,
  "changed": false
}
```

- *ansible all -m apt -a "name=htop state=present" --become*
  - Yum: módulo yum para la instalación de paquetes de software.
  - "name=acme": nombre del paquete a instalar.
  - "state=present": estado que se desea (instalar, actualizar...).



# Programación YAML y Playbooks

## Playbooks:

- Los playbooks son la forma que tiene Ansible para organizar, administrar y realizar las tareas que se quieren cumplir.
- Los playbooks están escritos en el lenguaje YAML para una fácil adaptación.
- En un playbook se pueden escribir distintas tareas y configuraciones, y aunque ofrecen un gran abanico de posibilidades, la mejor forma de sacarles partido es mediante roles.

## Características del lenguaje YAML:

- No admite tabulaciones, sólo espacios.
- Los comentarios se escriben con #
- Las posibles estructuras de datos son:
  - Listas
  - Tuplas o diccionarios

# Roles: estructura y función

Los roles son una serie de ficheros y carpetas que contienen funciones y tareas específicas. Cada rol tiene las siguientes carpetas:

- Files: en esta carpeta se almacenan los archivos y scripts que se quieran copiar.
- Templates: carpeta que contiene las plantillas a utilizar por las tareas.
- Tasks: carpeta contenedora de las distintas tareas a realizar.
- Handlers: carpeta con los archivos necesarios para las excepciones.
- Vars: en esta carpeta se almacenan los archivos con las variables.
- Defaults: variables por defecto.
- Meta: carpeta con las dependencias necesarias.

Todas estas carpetas son opcionales, dado que desde las tareas se puede especificar la ruta al directorio donde se encuentran todos los playbooks.

# Material Online

- <http://docs.ansible.com/>
- [http://docs.ansible.com/ansible/intro\\_installation.html](http://docs.ansible.com/ansible/intro_installation.html)
- [http://docs.ansible.com/ansible/modules\\_by\\_category.html](http://docs.ansible.com/ansible/modules_by_category.html)





Muchas Gracias

