



1

---

---

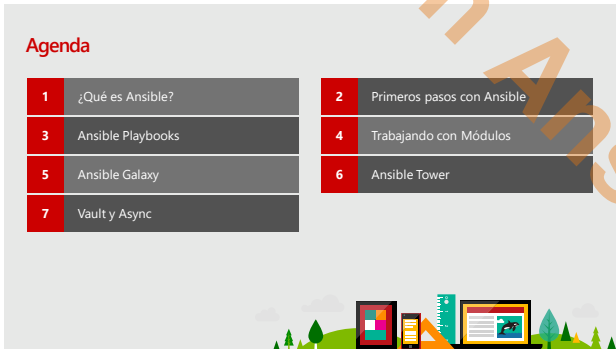
---

---

---

---

---



3

---

---

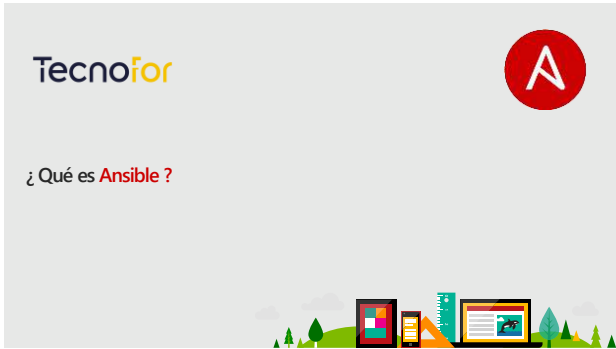
---

---

---

---

---



4

---

---

---

---

---

---

---

¿Qué es Ansible ?



Un poco de Historia

Ansible es un **proyecto Opensource** programado en *python*, que **nace en 2012** de la mano de *Michael DeHaan*.

El objetivos es unificar la funcionalidad de varias herramientas de automatización en una sola, para administra configuraciones, implementaciones y la ejecuciones ad-hoc.

En 2015, **Red Hat compra Ansible** y lo añade a su oferta de tecnologías **Opensource**, como proyecto estratégico para los próximos años.

**Red Hat** ofrece la certificación **Red Hat Certified Specialist in Ansible Automation**, el personal de IT que la obtiene, demuestra tener las habilidades y conocimientos que se necesitan para automatizar la gestión y la implementación de los sistemas utilizando Ansible.

---

---

---

---

---

---

---

5

¿Qué es Ansible ?



Funcionalidades

**Ansible** es un software **Opensource** para automatizar tareas de administración y configuración en la infraestructura y aplicaciones.

- **Servidores**, con sistemas operativos Linux y Windows
- **Infraestructura Cloud**, en Azure, Amazon Web Services, Google Cloud Platform, ...
- **Virtualización**, Docker, OpenStack, OpenShift, HyperV, Vmware, ...
- **Dispositivos**, como switches, routes y otros appliances
- **Aplicaciones**, gestionar la instalación, configuración y administrarlas.

---

---

---

---

---

---

---

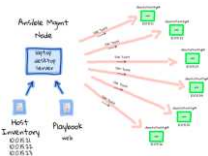
6

¿Qué es Ansible ?



Características Positivas

- No requiere un agente en los *Clientes*
- Se comunica con los clientes mediante **SSH**
- Sintaxis sencilla en **YAML**, fácil de comprender y utilizar
- Mantenimiento mínimo y sencillo
- Alto rendimiento
- No se necesitan conocimientos de programación



---

---

---

---

---

---

---

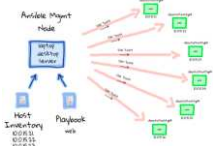
7

¿ Qué es **Ansible** ?



Características Negativas

- No es muy potente trabajando configuraciones
- Lento en grandes entornos
- Tardan en aparecer soluciones para los *bugs*
- Módulos escritos en Python



8

---

---

---

---

---

---

---

Código fuente de **Ansible®**



Software Opensource

Ansible® tiene licencia GNU General Public License v3.0 or later.  
El código es accesible desde el repositorio de GitHub:  
<https://github.com/ansible/ansible>

<https://github.com/ansible/ansible>

9

---

---

---

---

---

---

---

Documentación de **Ansible®**



Documentación Oficial

La documentación oficial de Ansible® está disponible a través de la siguiente URL:  
<https://docs.ansible.com/>

<https://docs.ansible.com/>

10

---

---

---

---

---

---

---



11

---

---

---

---

---

---

---

Primeros pasos con Ansible

Instalación del Control Node

Los requisitos para instalar el Nodo de Control de Ansible son:

- Sistema Operativo Linux como Red Hat, Debian, Ubuntu, CentOS, ...
- Sistema Operativo macOS
- Sistema Operativo Windows con una distribución Windows Subsystem for Linux (WSL)
- Python 3.8 o superior
- Configurar repositorio EPEL sin es necesario
- Crear un usuario (no root) para las tareas de Ansible

12

---

---

---

---

---

---

---



13

---

---

---

---

---

---

---

LABS

Instalación de Ansible® en sistema operativos Ubutun:

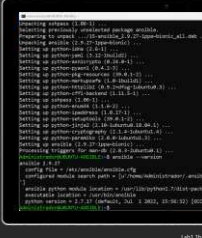
✓

Nos conectamos con el servidor Ubuntu utilizando SSH

✓

Ejecutamos los siguientes comandos:

```
sudo apt-get install software-properties-common
sudo apt-add-repository ppa:ansible/ansible
sudo apt-get update
sudo apt-get install ansible
ansible --version
```



14

---

---

---

---

---

---

---

---

LABS

Instalación de Ansible® en sistema operativos Windows:

✓

Utilizando RDP nos conectamos con el servidor

✓

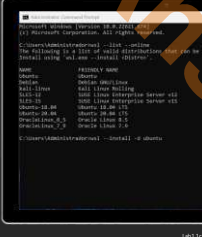
Instalamos WSL 2.0 mediante los siguientes comandos:

```
wsl --list --online
wsl --install -d ubuntu
```

✓

Reiniciamos e instalamos Ansible:

```
sudo apt-get install software-properties-common
sudo apt-add-repository ppa:ansible/ansible
sudo apt-get update
sudo apt-get install ansible
ansible --version
```



15

---

---

---

---

---

---

---

---

Instalación de Windows Subsystem for Linux



Instalación de WSL

Guía de instalación de una distribución de Linux utilizando el Subsistema de Windows para Linux.

WSL permite usar herramientas de Linux, como Ansible, completamente integradas en Windows, sin necesidad de un arranque dual.

<https://learn.microsoft.com/es-es/windows/wsl/install>

16

---

---

---

---

---

---

---

---

Formación del 14/11 al 17/11

5

## LABS

Instalación de Ansible® en sistema operativo Alpine:

- ✓ Nos conectamos con el servidor Alpine utilizando SSH
- ✓ Instalamos Python 3 y PIP:

```
apk update
apk add libffi-dev openssl-dev make gcc
apk add python3 py3-pip
```
- ✓ Instalamos Ansible:

```
pip install effi
pip install ansible
ansible -version
```

---

---

---

---

---

---

17

# Primeros pasos con Ansible

## Archivo de Inventario

**Ansible** funciona contra múltiples nodos administrados o "hosts" de nuestra infraestructura al mismo tiempo.

Utiliza una lista o grupo de listas conocido como *Inventario*.

Una vez definido el inventario, utilice patrones para seleccionar los hosts o grupos contra los que desea que se ejecute **Ansible**.

La ubicación predeterminada para el inventario es **/etc/ansible/hosts**.

Puede especificar un archivo de inventario diferente en la línea de comandos mediante la opción **ansible -i <path>**.



Sample

18

# Primeros pasos con Ansible

## Formato del Archivo de Inventario

El archivo de inventario puede estar en diferentes formatos, los más comunes son INI y YAML.

FORMATO INI	FORMATO YAML
mail.example.com	hosts:
[webserver]	mail.example.com:
foo.example.com	children:
bar.example.com	webserver:
	hosts:
[dbserver]	foo.example.com:
one.example.com	bar.example.com:
two.example.com	dbserver:
three.example.com	hosts:
	one.example.com:
	two.example.com:
	three.example.com:

Or

19

Primeros paso con Ansible

Mi Primir Nodo Administrado

Modificamos el fichero `/etc/ansible/host` para añadir `localhost` como nodo administrado, y ejecutamos el módulo `ping`.

```
ansible:~# nano /etc/ansible/hosts
ansible:~# ansible localhost -m ping
localhost | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: root@localhost: Permission denied (publickey,password,keyboard-interactive).",
  "unreachable": true
}
```

20

---

---

---

---

---

---

---

---

Primeros paso con Ansible

Mi Primir Nodo Administrado

Modificamos el fichero `/etc/ansible/hosts` para añadir la variable `ansible_connection=local` indicando que la conexión es local en lugar de `ssh`, y ejecutamos el módulo `ping`.

```
ansible:~# nano /etc/ansible/hosts
ansible:~# ansible localhost -m ping
localhost | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.10"
  },
  "changed": false,
  "ping": "pong"
}
```

21

---

---

---

---

---

---

---

---

Primeros pasos con Ansible

Ejecución de Comando

La sintaxis `-m <module>` es para ejecutar módulos, y `-a "args"` lo usamos para especificar los argumentos del módulo. Cuando no especificamos modulo se ejecuta el modulo `command`.

```
ansible:~# ansible localhost -a "hostname"
localhost | CHANGED | rc=0 >>
ansible
ansible:~# ansible localhost -a "cat /etc/os=release"
localhost | CHANGED | rc=0 >>
NAME="Alpine Linux"
ID=alpine
VERSION_ID=3.16.2
PRETTY_NAME="Alpine Linux v3.16"
HOME_URL="https://alpinelinux.org/"
BUG_REPORT_URL="https://gitlab.alpinelinux.org/alpine/aports/-/issues"
```

22

---

---

---

---

---

---

---

---

Primeros pasos con Ansible



Nodo Administrados no inventariados

Cuando tratamos de ejecutar un modulo en un host que NO se encuentra en el *Inventario*, **Ansible** nos responde un con mensaje de advertencia.

```
ansible:~# ansible alpine3 -m ping
[WARNING]: Could not match supplied host pattern, ignoring: alpine3
[WARNING]: No hosts matched, nothing to do
```

23

---

---

---

---

---

---

---

Primeros pasos con Ansible



Nodo Administrados utilizando SSH

Cuando tratamos de ejecutar un modulo en un host administrado, **Ansible** nos puede responde un de no autorizado. Entonces tendremos que generar unas claves para el usuario root.

```
ansible:~# ansible alpine -m ping
The authenticity of host 'alpine (192.23.264.186)' can't be established.
ED25519 key fingerprint is SHA256:lpabxmdjMsrWuyq/PD5oPRMaRfDovG59tttrhVx78.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
alpine2 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: Warning: Permanently added 'alpine' (ED25519) to the list of
known hosts.\r\nroot@alpine: Permission denied (publickey,password,keyboard-interactive).",
  "unreachable": true
}
```

24

---

---

---

---

---

---

---

LABS



Creamos unas claves de acceso en el *Control Node* y las registraremos como claves autorizadas en los *Host Administrados*.

- ✓ En el *Control Node* creamos un par de claves para el usuario root:  

```
ssh-keygen
cat /root/.ssh/id_rsa.pub
```
- ✓ En los *Host Administrados* incorporamos la clave pública como clave autorizada:  

```
nano /root/.ssh/authorized_keys
```
- ✓ Desde el *Control Node* ejecutamos módulos en los *Host Administrados*.  

```
ansible alpine -m ping
```

25

---

---

---

---

---

---

---



Primeros pasos con Ansible



Nodo Administrados utilizando SSH

Generadas las claves y autorizado el usuario en los nodos administrados el módulo se ejecuta correctamente.

```
ansible:~# ansible alpine -m ping
alpine | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.10"
  },
  "changed": false,
  "ping": "pong"
}
```

26

---

---

---

---

---

---

---

Primeros pasos con Ansible



Nodo Administrados utilizando SSH

Cuando los nodos administrados no tienen el usuario root la conexión no es fallida. Tenemos que especificar el usuario de la conexión mediante el `-u <username>` o el parámetro `ansible_user`.

```
ansible:~# ansible ubuntu -m ping
ubuntu | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: root@ubuntu: Permission denied (publickey,password).",
  "unreachable": true
}
ansible:~# ansible ubuntu -m ping -u admin
ubuntu | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

27

---

---

---

---

---

---

---

Primeros pasos con Ansible



Nodo Administrados utilizando SSH

Para ejecutar módulos o comando en todos los host administrados utilizamos `all` en lugar de indicar el nombre de un host del *Inventario*.

```
ansible:~# ansible all -m ping
localhost | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
alpine | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
ubuntu | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

28

---

---

---

---

---

---

---

Primeros pasos con Ansible



Parámetros para los Host Administrados

La configuración de variables para los host administrados le indica **Ansible** como interactúa con hosts remotos.

- **ansible\_connection**, determina el tipo de conexión como *local*, *ssh*, *winrm*, *smart* y *paramiko*
- **ansible\_host**, nombre del host de conexión (diferente al alias)
- **ansible\_user**, nombre del usuario para la conexión ssh  
También se puede especificar mediante **ansible -u <username>** o **ansible --user <username>**
- **ansible\_ssh\_pass**, contraseña para la conexión ssh  
También se puede especificar mediante **ansible -k** o **ansible --ask-pass**
- **ansible\_ssh\_private\_key\_file**, ubicación y nombre del archivo de clave privada utilizado para la conexión ssh

29

Primeros pasos con Ansible



Variables para los Host Administrados

Una colección de clave/valor junto a los valores de los parámetros se consideran variables, y se puede definir a nivel de host o grupo.

Variables de Host	Variable de Grupo
<pre>[atlanta] host1 http_port=80 maxRequestsPerChild=808 host2 http_port=303 maxRequestsPerChild=909</pre>	<pre>[atlanta] host1 http_port=80 maxRequestsPerChild=808 host2 http_port=443  [atlanta:vars] ntp_server=ntp.atlanta.example.com maxRequestsPerChild=1000</pre>

Si una variable existe a nivel de grupo y de host, se aplica el valor definido a nivel de host.

30

Primeros pasos con Ansible



Variables para los Host Administrados

Una buena práctica puede ser no almacenar variables en el archivo de inventario principal.

Las variables también se pueden almacenar en archivos relativos al inventario. Los archivos de variables están en formato YAML con extensiones *.yaml*, *.yml*, *.json* o ninguna.

Archivos de Variables	Variable de Grupo
<pre>/etc/ansible/group_vars/atlanta /etc/ansible/host_vars/host1 /etc/ansible/host_vars/host2  Relativo al archivo de inventario en el path: /etc/ansible/hosts</pre>	<pre>[atlanta] host1 http_port=80 maxRequestsPerChild=808 host2 http_port=443  [atlanta:vars] ntp_server=ntp.atlanta.example.com maxRequestsPerChild=1000</pre>

31

```
ansible:/# cat /etc/ansible/hosts
[localhost]
localhost ansible_connection=local

[alpinervers]
webserver ansible_host=alpine
#alpine1
#alpine2
alpine[1:2]

[ubuntuusers]
ubuntu ansible_user=borja

[frontend:children]
alpinervers
ubuntuusers

[frontend:vars]
ansible_become=true
```

webserver lo definimos como un alias para el host remoto alpine.

El comando `ping webserver` no responden, pero si es valido para los comando de ansible:

```
ansible webserver -a id
```

32

---

---

---

---

---

---

---

---

```
ansible:/# cat /etc/ansible/hosts
[localhost]
localhost ansible_connection=local

[alpinervers]
webserver ansible_host=alpine
#alpine1
#alpine2
alpine[1:2]

[ubuntuusers]
ubuntu ansible_user=borja

[frontend:children]
alpinervers
ubuntuusers

[frontend:vars]
ansible_become=true
```

Cuando registramos los host administrados en el **Inventario** podemos utilizar patrones.

Los patrones permiten registrar varios servidores en una sola linea:

[webservers]	[databases]
www[01:90].example.com	db-[a-z].example.com

33

---

---

---

---

---

---

---

---

```
ansible:/# cat /etc/ansible/hosts
[localhost]
localhost ansible_connection=local

[alpinervers]
webserver ansible_host=alpine
#alpine1
#alpine2
alpine[1:2]

[ubuntuusers]
ubuntu ansible_user=borja

[frontend:children]
alpinervers
ubuntuusers

[frontend:vars]
ansible_become=true
```

Todos los host pertenecen al grupo **all** automáticamente.

Es posible registrar grupos de host administrados y grupos de grupos, para ejecutar comandos ansible para todos los host contenidos en el grupo:

```
ansible alpinervers -a ti
```

34

---

---

---

---

---

---

---

---

Primeros pasos con Ansible



Inventarios Dinámicos

Cuando los *Inventarios* se hacen grandes es complicado de mantenerla información actualizada.

**Ansible** admite sistema de inventario dinámicos, donde el listado de host administrados se crea en al ejecutar el comando **ansible**.

En GitHub encontramos scripts de Python para diferentes fuentes de *Inventario Dinámico*:

<https://github.com/ansible-community/contrib-scripts/tree/main/inventory>

<https://github.com/ansible/ansible/tree/stable-2.9/lib/ansible/plugins/inventory>

**Ansible Tower** proporciona una base de datos para almacenar los resultados de diferentes fuentes de *Inventarios Dinámico*, siendo accesible tanto en la web como en REST.

---

---

---

---

---

---

---

---

35

Primeros pasos con Ansible



Combinar Inventarios

También podemos trabajar con *Inventarios* divididos en varios ficheros, incluso tener *Inventarios* estáticos y dinámicos.

**Ansible** permite combinar *Inventarios* en el momento de la ejecución de comandos.

Crearemos un directorio, por ejemplo `/etc/ansible/inventarios`, donde crearemos diferentes ficheros de Inventario, y utilizamos el parámetro `-i <path>` para indicar el directorio:

```
ansible all -m ping -i inventarios/ --list-hosts
```

---

---

---

---

---

---

---

---

36

Primeros pasos con Ansible



Ad hoc

**Ad hoc** es una locución latina que significa literalmente «**para esto**» o «**para este propósito**».

Generalmente se refiere a una solución específicamente elaborada para un problema o fin preciso y, por tanto, no generalizable ni utilizable para otros propósitos.

Fuente: Wikipedia

---

---

---

---

---

---

---

---

38

Primeros pasos con Ansible



Comandos ad hoc

Los comandos **ad hoc** son excelentes para tareas que se repiten raramente, como por ejemplo:

- reiniciar servidores
- copiar archivos
- administrar paquetes y usuarios
- etc.

Podemos utilizar cualquier módulo de Ansible en una tarea ad hoc, siendo **command** el módulo por defecto. Sintaxis comando ad hoc:

```
ansible [pattern] -m [module] -a "[module args]"
```

39

Primeros pasos con Ansible



Patrones para un comandos ad hoc

Patrones comunes para hosts administrados y grupos del inventario:

	Patrón	Objetivo
Todos los host	all o *	Todos los servidores del inventario
Un host	host1	El servidor host1
Múltiples host	host1:host2 o host1,host2	Los servidores host1 y host2
Un grupo	webservers	Todos los servidores web
Múltiples grupos	webservers:dbservers	Todos los servidores web y de bases de datos
Excluidos los grupos	webservers:atlanta	Todos los servidores web excepto los de Atlanta
Intersección de grupos	webservers&staging	Todos los servidores web que también estén en ensayo

40

Primeros pasos con Ansible



Índices en patrón de los comandos ad hoc

Podemos utilizar índices para seleccionar hosts administrados o rangos dentro del grupo:

/etc/ansible/hosts	Posibles patrones
[webservers] cobweb webbing weber	webservers[0] # == cobweb webservers[-1] # == weber webservers[0:2] # == webservers[0],webservers[1] # == cobweb,webbing webservers[1:] # == webbing,weber webservers[:3] # == cobweb,webbing,weber

41

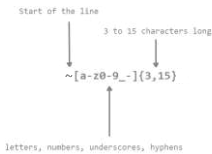
Primeros pasos con Ansible



**[RegEx]\*** como patrón los comandos ad hoc

Podemos especificar una expresión regular como un patrón iniciando **~**.

**~(web|db).\*\..example\..com**



42

Primeros pasos con Ansible



Parámetro **--limit** de los comandos ad hoc

Podemos limitar los hosts administrados a los que se dirige en una ejecución del comando **ad hoc** utilizando el parámetro **--limit**

Patrón	Objetivo
<code>ansible [pattern] -m [module] -a [args] --limit 'host1'</code>	Limita al servidor host1
<code>ansible [pattern] -m [module] -a [args] --limit 'host1,host2'</code>	Limita a los servidores host1 y host2
<code>ansible [pattern] -m [module] -a [args] --limit 'all:!host2'</code>	Todos menos el host2
<code>ansible [pattern] -m [module] -a [args] --limit 'group1'</code>	Limita a los servidores del grupo group1

43

Primeros pasos con Ansible



Otros parámetro de los comandos ad hoc

Utilizamos un conjunto de parámetros para determinar el comportamiento del comando ansible.

```
usage: ansible [-h] [--version] [-v] [-b] [--become-method BECOME_METHOD]
               [--become-user BECOME_USER]
               [-k] [--become-password-file BECOME_PASSWORD_FILE]
               [-i INVENTORY] [--list-hosts] [-l SUBSET] [-P POLL_INTERVAL]
               [-B SECONDS] [-o] [-t TREE] [--private-key PRIVATE_KEY_FILE]
               [-u REMOTE_USER] [--connection] [-T TIMEOUT]
               [--ssh-common-args SSH_COMMON_ARGS]
               [--sftp-extra-args SFTP_EXTRA_ARGS]
               [--scp-extra-args SCP_EXTRA_ARGS]
               [--ssh-extra-args SSH_EXTRA_ARGS]
               [-k] [--connection-password-file CONNECTION_PASSWORD_FILE] [-c]
               [--syntax-check] [-D] [-e EXTRA_VARS] [--vault-id VAULT_IDS]
               [--ask-vault-password] [--vault-password-file VAULT_PASSWORD_FILES]
               [-f FORKS] [-M MODULE_PATH] [--playbook-dir BASEDIR]
               [--task-timeout TASK_TIMEOUT] [-a MODULE_ARGS] [--module-name]
               pattern
```

44

```
optional arguments:
--ask-vault-password      ask for vault password
--ask-vault-pass
--become-password-file    BECOME_PASSWORD_FILE Become password file
--become-pass-file
--connection-password-file CONNECTION_PASSWORD_FILE Connection password file
--conn-pass-file
--list-hosts              outputs a list of matching hosts; does not execute anything else
--playbook-dir            BASEDIR Since this tool does not use playbooks, use this as a substitute
                           playbook directory. This sets the relative path for many features,
                           including roles/ group_vars/ etc.
--syntax-check            perform a syntax check on the playbook, but do not execute it
--task-timeout            TASK_TIMEOUT set task timeout limit in seconds, must be positive integer.
--vault-id                VAULT_IDS the vault identity to use
--vault-password-file      VAULT_PASSWORD_FILES vault password file
--vault-pass-file
--version                 show program's version number, config file location, configured module
                           search path, module location, executable location and exit
```

45

```
-B                      SECONDS run asynchronously, failing after X seconds (default=M/A)
--background
-C, --check             don't make any changes; instead, try to predict some of the changes that
                           may occur
-D, --diff              when changing (small) files and templates, show the differences in those
                           files; works great with --check
-K, --ask-become-pass   ask for privilege escalation password
-M
--module-path           MODULE_PATH prepend colon-separated path(s) to module library
                           (default=~/ansible/plugins/modules:/usr/share/ansible/plugins/modules)
-P
--poll                 POLL_INTERVAL set the poll interval if using -B (default=15)
-a
--args                 MODULE_ARGS The action's options in space separated k=v format:
                           -a 'opt1=val1 opt2=val2'
-e
--extra-vars            EXTRA_VARS set additional variables as key=value or YAML/JSON,
                           if filename prepend with @
-f
--forks                FORKS specify number of parallel processes to use (default=5)
```

46

```
-h, --help              show this help message and exit
-i
--inventory             INVENTORY specify inventory host path or comma separated host list.
--inventory-file         --inventory-file is deprecated
-k
--ask-pass              ask for connection password
-l
--limit                 SUBSET further limit selected hosts to an additional pattern
-m
--module-name           MODULE_NAME Name of the action to execute (default=command)
-o, --one-line          condense output
-t
--tree                 TREE log output to this directory
-v
--verbose               verbose mode (-vvv for more, -vvvv to enable connection debugging)
```

47

Privilege Escalation Options:  
control how and which user you become as on target hosts

--become-method

BECOME\_METHOD privilege escalation method to use (default=sudo), use 'ansible-doc -t become -l' to list valid choices.

--become-user

BECOME\_USER run operations as this user (default=root)

-b

run operations with become (does not imply password prompting)

48

---

---

---

---

---

---

---

---

Connection Options:  
control as whom and how to connect to hosts

--private-key

PRIVATE\_KEY\_FILE use this file to authenticate the connection

--key-file

PRIVATE\_KEY\_FILE use this file to authenticate the connection

--scp-extra-args

SCP\_EXTRA\_ARGS specify extra arguments to pass to scp only (e. -l)

--sftp-extra-args

SFTP\_EXTRA\_ARGS specify extra arguments to pass to sftp only (e. -f, -l)

--ssh-common-args

SSH\_COMMON\_ARGS specify common arguments to pass to sftp/scp/ssh (e.g. ProxyCommand)

--ssh-extra-args

SSH\_EXTRA\_ARGS specify extra arguments to pass to ssh only (e.g. -R)

-T

TIMEOUT override the connection timeout in seconds (default=10)

--timeout

TIMEOUT override the connection timeout in seconds (default=10)

-c

CONNECTION connection type to use (default=smart)

--connection

CONNECTION connection type to use (default=smart)

-u

REMOTE\_USER connect as this user (default=None)

--user

REMOTE\_USER connect as this user (default=None)

49

---

---

---

---

---

---

---

---

Primeros pasos con Ansible

Otros parámetro de los comandos ad hoc

Utilizamos un conjunto de parámetros para determinar el comportamiento del comando ansible.

Parámetro	Objetivo
--limit	Limitar aún más los host seleccionados por el patrón
--user o -u	Usuario con el que realiza la conexión
--become o -b	Ejecutar las operaciones en el host remoto con el <i>usuario become</i> , por defecto, <i>root</i>
--inventory o -I	Especificar la ruta del <i>inventario</i> o la lista de hosts remotos separados por comas
--forks o -f	Especificar el número de procesos paralelos a usar (predeterminado = 5)
--list-hosts	Muestra una lista de hosts remotos coincidentes; no ejecuta ningún comando
--connection o -c	Especificar el tipo de conexión (predeterminado = <i>smart</i> )
--verbose o -v	Habilitar modo detallado (-vvv más detalle y -vvvv depuración de conexiones)

50

---

---

---

---

---

---

---

---



Primeros pasos con Ansible



Archivo de Configuración

Ansible se configura a través de un archivo de configuración `ansible.cfg`.  
A partir de la versión 2.4 de Ansible, puede utilizar el comando `ansible-config` para enumerar las opciones disponibles e inspeccionar los valores actuales.  
El archivo de configuración se busca en el siguiente orden:

- `ANSIBLE_CONFIG` (varlo de la variable de entorno)
- `ansible.cfg` (archivo en el directorio actual)
- `~/.ansible.cfg` (archivo en el directorio de inicio)
- `/etc/ansible/ansible.cfg`

51

---

---

---

---

---

---

---

Primeros pasos con Ansible



Archivo de Configuración

El archivo de configuración contiene:

- Valores generales como cual es el fichero de inventario
- Configuración de escalar permisos (método sudo y el usuario utilizado `root`)
- Opciones de OpenSSH
- Opciones de SELinux
- Configuración de Ansible-Galaxy

52

---

---

---

---

---

---

---

Opciones de Configuración



Opciones del Archivos de Configuración

Listado de las opciones de configuración, su archivo puede tener opciones adicionales debido a complementos adicionales.  
El comando mencionada anteriormente `ansible-config` permite navegar a través de ellos.

[https://docs.ansible.com/ansible/latest/reference\\_appendices/config.html](https://docs.ansible.com/ansible/latest/reference_appendices/config.html)

[https://docs.ansible.com/ansible/latest/reference\\_appendices/config.html](https://docs.ansible.com/ansible/latest/reference_appendices/config.html)

53

---

---

---

---

---

---

---

Primeros pasos con Ansible



Archivo de Configuración

Un archivo de ejemplo está disponible en [GitHub](#).  
Puede generar un archivo de ejemplo completamente comentado, mediante el comando:

```
ansible-config init --disabled > ansible.cfg
```

También puede tener un archivo más completo que incluya complementos existentes:

```
ansible-config init --disabled -t all > ansible.cfg
```

54

Archivos de Ejemplo



**Archivos de Ejemplo**  
El repositorio de GitHub tiene archivos de ejemplo, incluido una copia del archivo de configuración `ansible.cfg`.  
<https://github.com/ansible/ansible/tree/devel/examples>

<https://github.com/ansible/ansible/tree/devel/examples>

55

Primeros pasos con Ansible



Equipos con Sistema Operativo Windows

- El host administrado con sistema operativo Windows debe cumplir estos requisitos:
- Sistemas operativos de escritorio Windows 8.1, 10 y 11
  - Sistemas operativos de servidor Windows Server 2012, 2012 R2, 2016, 2019 y 2022.
  - Instalar PowerShell 3.0 o posterior y Microsoft Framework .NET 4.x
  - Crear un agente de escucha de WinRM y habilitar el puerto 5986
- El host servidor con **Ansible** debe cumplir estos requisitos:
- Instalar la librería `pywinrm`, utilizando el comando `pip install pywinrm`
  - Establecer el valor de `connection` a `winrm`


56

## LABS

Preparar un host administrado con sistema operativo Windows.



- ✓ Instalación de Software:
  - Python 3.0 o superior
  - PowerShell 3.0 o superior
  - Microsoft .NET Framework 4.x
- ✓ Crear un *Listener* para WinRM:
  - Crear un certificado
  - Crear el agente de escucha o *Listener*
- ✓ Verificar las conexiones:
  - Añadir host de confianza
  - Realizar test desde *Cm* y *PowerShell*



```

New-SelfSignedCertificate -ExpiryDate "host.com"
CertificateName CertListenerBy

-- 0 --

$listener_set = @{}
$address = @("http://")
$transport = "HTTPS"

$listener_set += @{
    CertificateThumbprint = "502C48A9B0D742E83A68B0D738E4A4709C"
}

New-HttpListener -ResourcePath "winrm/config/listener"
-SelectorSet $listener_set -ValidUriSet $listener_set

-- 0 --

winrm create winrm/config/listener/Address @{Transport=HTTPS
($listener_set.Name)
CertificateThumbprint=502C48A9B0D738E4A4709C}

-- 0 --

winrm enumerate winrm/config/listener
          
```

57

## LABS

Preparar un host administrado con sistema operativo Windows.

- ✓ Instalación de Software:
  - Python 3.0 o superior
  - PowerShell 3.0 o superior
  - Microsoft .NET Framework 4.x
- ✓ Crear un *Listener* para WinRM:
  - Crear un certificado
  - Crear el agente de escucha o *Listener*
- ✓ Verificar las conexiones:
  - Añadir host de confianza
  - Realizar test desde cmd y PowerShell

```
Set-Item WSMAN:\LocalHost\Certificate\TrustPorts -Value "80,82,33" -Force
Set-Item WSMAN:\LocalHost\Certificate\TrustPorts -Value Host-name
Set-Item WSMAN:\LocalHost\Certificate\TrustPorts -Certificate $cert -Value Host-name

# Test out HTTP
winrs -r:https://Host-name:5985/wsmn -u:User-name -p:Password {cmdfig}

# Test out HTTPS (full call of the cert is not verifiable)
winrs -r:https://Host-name:5986/wsmn -u:User-name -p:Password -s {cmdfig}

# Test out HTTPS, ignoring certificate verification
$cert = Get-Certificate
$session_option = New-Object System.Management.Automation.SessionOption -SkipCertificateCheck -SkipPowershellCertificateCheck

Invoke-Command -ComputerName Host-name -u:User -ScriptBlock { {cmdfig} } -Credential $cert -SessionOption $session_option
```

58

## Configurando Windows



Configurando un host Microsoft Windows

En el enlace se describe la configuración necesaria para que Ansible pueda comunicarse y administrar un host remoto con sistema operativo Microsoft Windows.

[https://docs.ansible.com/ansible/latest/user\\_guide/windows\\_setup.html](https://docs.ansible.com/ansible/latest/user_guide/windows_setup.html)

59

```
[windows]
win2019 ansible_host=192.168.99.105

[windows:vars]
ansible_connection=winrm
ansible_user=ansible
ansible_password="MySecretPassword"

# Enable HTTPS
ansible_winrm_scheme=https
ansible_port=5986
ansible_winrm_transport=certificate
ansible_winrm_cert_pem=/Users/gini/.ssh/cert.pem
ansible_winrm_cert_key_pem=/Users/gini/.ssh/cert_key.pem
ansible_winrm_server_cert_validation=ignore
```

Ejemplo de fichero de Inventario utilizado para conectar con un host remoto con sistema operativo Windows mediante **HTTPS**.

60

---

---

---

---

---

---

---

---

```
[windows]
win2019 ansible_host=192.168.99.105

[windows:vars]
ansible_connection=winrm
ansible_user=ansible
ansible_password="MySecretPassword"

# Enable HTTP
ansible_winrm_scheme=http
ansible_port=5985
ansible_winrm_transport=basic
ansible_winrm_server_cert_validation=ignore
```

Ejemplo de fichero de Inventario utilizado para conectar con un host remoto con sistema operativo Windows mediante **HTTP**.

61

---

---

---

---

---


---

---

---

Tecnofor

Ansible Playbooks



62

---

---

---

---

---

---

---

---

Ansible Playbooks



¿Qué son los Playbooks?

Los **Playbooks** de **Ansible** permiten realizar tareas de administración repetibles y reutilizables en múltiples host administrados.

Los libros de jugadas o **Playbooks** se expresan en formato **YAML** con una sintaxis para:

- especificar o declarar configuraciones
- orquestar los pasos de cualquier proceso, en varios host remotos, en un orden definido
- iniciar tareas de forma síncrona o asíncrona

Los **Playbooks** contienen jugas que se ejecuta en orden, de arriba abajo, la jugadas contienen tareas que también se ejecutan de arriba a abajo.

---

---

---

---

---

---

---

63

```
---
name: Demo Playbook
hosts: <patron>
remote_user: administrador
become: yes

tasks:
  - name: Install httpd service
    yum: httpd

  - name: Start web server
    service:
      name: httpd
      state: started
```

Utilizamos **name** para indicar el nombre del **Playbook**, y **hosts** para indicar los hosts remotos donde se ejecutan las tareas definidas.



---

---

---

---

---

---

---

64

```
---
name: Demo Playbook
hosts: <patron>
remote_user: administrador
become: yes

tasks:
  - name: Install httpd service
    yum: httpd

  - name: Start web server
    service:
      name: httpd
      state: started
```

Sintaxis opcional

```
- name: Start web server
  service: name=httpd state=started
```

Definimos un conjunto de tareas, para ser ejecutadas en hosts remotos.  
**name** contiene la descripción de la tarea.  
Seguidamente el nombre del módulo y sus argumentos.



---

---

---

---

---

---

---

65

Ansible Playbooks



Ejecutar los Playbooks

Ejecutamos los **Playbooks** de **Ansible** mediante el siguiente comando:

```
ansible-playbook -i <inventory> <options> playbook.yml
```

Los parámetros u opciones de ejecución son en su mayoría compartidas con el comando **ansible**. Es posible que desee verificar sus libros de jugadas para detectar errores de sintaxis y otros problemas antes de ejecutarlos. El comando **ansible-playbook** ofrece varias opciones de verificación, incluidas **--check**, **--diff**, **--list-hosts**, **--list-tasks** y **--syntax-check**.

66

Ansible Playbooks



Parámetro del comando ansible-playbook

```
usage: ansible-playbook [-h] [--version] [-v] [--private-key PRIVATE_KEY_FILE]
                        [-u REMOTE_USER] [-c CONNECTION] [-t TIMEOUT]
                        [--ssh-common-args SSH_COMMON_ARGS]
                        [--sftp-extra-args SFTP_EXTRA_ARGS]
                        [--scp-extra-args SCP_EXTRA_ARGS]
                        [--ssh-extra-args SSH_EXTRA_ARGS]
                        [-k] [--connection-password-file CONNECTION_PASSWORD_FILE]
                        [--force-handlers] [--flush-cache] [-D]
                        [--become-method BECOME_METHOD]
                        [--become-user BECOME_USER]
                        [-e] [--become-password-file BECOME_PASSWORD_FILE]
                        [-t TAGS] [--skip-tags SKIP_TAGS] [-C]
                        [--syntax-check] [-D] [-i INVENTORY] [--list-hosts]
                        [-l SUBSET] [--extra-vars] [--vault-id VAULT_IDS]
                        [--ask-vault-password] [--vault-password-file VAULT_PASSWORD_FILES]
                        [-f FORKS] [--module-path] [--list-tasks]
                        [--list-tags] [--step] [--start-at-task START_AT_TASK]
                        playbook [playbook ...]
```

67

Ansible Playbooks



Parámetros más comunes de ansible-playbook

Utilizamos parámetros para determinar el comportamiento del comando **ansible-playbook**.

Parámetro	Objetivo
--syntax-check	Realice una verificación de sintaxis en el libro de jugadas, pero no lo ejecute
--step	Paso a paso: confirme cada tarea antes de ejecutarla
--start-at-task <NAME_TASK>	Inicia el libro de jugadas en la tarea que coincide con este nombre
--inventory o -i	Especificar la ruta del <i>Inventory</i> o la lista de hosts remotos separados por comas
--forks o -f	Especificar el número de procesos paralelos a usar (predeterminado = 5)
--list-tasks	Enumerar todas las tareas que se ejecutarían
--extra-vars o -e	Establecer variables adicionales como clave=valor o YAML/JSON
--verbose o -v	Habilitar modo detallado (-vvv más detalle y -vvvv depuración de conexiones)

68

Ansible Playbooks



Parámetros *Become* de ansible-playbook

Conjunto de parámetros de **ansible-playbook** para ejecutar tareas con privilegios elevados.

Parámetro	Objetivo
--become o -b	Ejecutar operaciones con become (no implica solicitud de contraseña)
--become-method	Método para elevar privilegios (predeterminado=sudo)
--become-password-file	Archivos del usuario con privilegios elevados
--become-pass-file	Archivos del usuario con privilegios elevados
--become-user	Ejecutar tareas como este usuario (predeterminado=root)
--ask-become-pass o -K	Pedir contraseña usar el usuario con privilegios elevados

---

---

---

---

---

---

---

---

69

**Mi Primer Playbook**

```
---
- name: Mi primer playbook, copiar un fichero del server a los host administrados
  hosts: Linux

  tasks:
    - name: Copiar fichero demo
      copy: src=/etc/ansible/demos/file-demo.txt dest=/etc/file-demo.txt
---
```

Realizamos nuestro primer **Playbook** donde queremos copiar un fichero del host server a los host remotos.

```
ansible:/ $ sudo nano /etc/ansible/mi-primer-playbook.yml
```

---

---

---

---

---

---

---

---

70

**Ansible Playbooks**

**Mi Primer Playbook**

```
ansible:/ $ ansible-playbook ./etc/ansible/mi-primer-playbook.yml --syntax-check
playbook: ./etc/ansible/mi-primer-playbook.yml

ansible:/ $ ansible-playbook ./etc/ansible/mi-primer-playbook.yml --list-tasks
playbook: ./etc/ansible/mi-primer-playbook.yml
play #1 (all): Mi primer playbook, copiar un fichero del server a los host administrados  TAGS: []
  tasks:
    Copiar fichero demo  TAGS: []
```

---

---

---

---

---

---

---

---

71

Ansible Playbooks

A

Mi Primir Playbook

```
ansible:/$ ansible-playbook ./etc/ansible/mi-primir-playbook.yml --list-hosts

playbook: ./etc/ansible/mi-primir-playbook.yml

play #1 (linux): Mi primer playbook, copiar un fichero del server a los host administrados  TAGS: []
pattern: ['linux']
hosts (3):
  host04
  host02
  host03
```

72

---

---

---

---

---

---

---

---

Ansible Playbooks

A

Mi Primir Playbook

```
ansible:/$ ansible-playbook ./etc/ansible/mi-primir-playbook.yml

PLAY [Mi primer playbook, copiar un fichero del server a los host administrados] *****
TASK [Gathering Facts] *****
ok: [host02]
ok: [host04]
ok: [host03]

TASK [Copiar fichero demo] *****
fatal: [host03]: FAILED! => ["change": false, "checksum": "8fd83bf2702849c19c3b6e470e478f750de9e29", "msg": "Destination /etc not writable"]
fatal: [host04]: FAILED! => ["change": false, "checksum": "8fd83bf2702849c19c3b6e470e478f750de9e29", "msg": "Destination /etc not writable"]
fatal: [host04]: FAILED! => ["change": false, "checksum": "8fd83bf2702849c19c3b6e470e478f750de9e29", "msg": "Destination /etc not writable"]

PLAY RECAP *****
host02 : ok=1 changed=0 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0
host03 : ok=0 changed=0 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0
host04 : ok=0 changed=0 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0
```

73

---

---

---

---

---

---

---

---

Mi Primir Playbook

A

```
---
- name: Mi primer playbook, copiar un fichero del server a los host administrados
  hosts: linux
  remote_user: Administrador
  become: True

  tasks:
    - name: Copiar fichero demo
      copy: src=/etc/ansible/demos/file-demo.txt dest=/etc/file-demo.txt
...

```

La escritura dentro de /etc/\_ no está permitida para el usuario *Administrador*, modificamos nuestro **Playbook** para indicar que todas las tareas se ejecuten como *root*.

74

---

---

---

---

---

---

---

---



Ansible Playbooks

A

Mi Primir Playbook

ansible:/\$ ansible-playbook ./etc/ansible/mi-primir-playbook.yml

PLAY [Mi primer playbook, copiar un fichero del server a los host administrados] \*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*

ok: [host02]

ok: [host03]

ok: [host04]

TASK [Copiar fichero demo] \*\*\*\*\*

changed: [host02]

changed: [host03]

changed: [host04]

PLAY RECAP \*\*\*\*\*

host02 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

host03 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

host04 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

75

---

---

---

---

---

---

---

---

Ansible Playbooks

A

Mi Primir Playbook

Quando ejecutamos el **Playbook** por segunda vez consecutiva en la respuesta *changed* tiene un valor igual a *cero* lo que indica que fichero de destino no cambia con la ejecución de la tarea.

ansible:/\$ ansible-playbook ./etc/ansible/mi-primir-playbook.yml

PLAY [Mi primer playbook, copiar un fichero del server a los host administrados] \*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*

ok: [host02]

ok: [host03]

ok: [host04]

TASK [Copiar fichero demo] \*\*\*\*\*

ok: [host02]

ok: [host03]

ok: [host04]

PLAY RECAP \*\*\*\*\*

host02 : ok=2 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

host03 : ok=2 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

host04 : ok=2 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

76

---

---

---

---

---

---

---

---

Ansible Playbooks

A

Variables

Con **Ansible**, podemos ejecutar tareas en múltiples sistemas con diferencias.

Representar las variaciones entre los sistemas es posible utilizando variables que definimos en:

- libros de jugadas
- el inventario
- obtenidas del servidor de **Ansible** (*ansible\_facts*)
- archivos o funciones reutilizables
- la línea de comandos (*utilizando el parámetro -e o --extra-vars*)

También podemos crear variables durante la ejecución de un **Playbooks** registrando los valores devueltos de una tarea como una variable.

77

---

---

---

---

---

---

---

---

Formación del 14/11 al 17/11

25

Ansible Playbooks



Prioridad de las Variables

Ansible aplica prioridad variable, y este es el orden de precedencia de menor a mayor (las últimas variables enumeradas anulan todas las demás variables):

1. command line (-u my\_user, not variables)

2. role defaults (defined in role/defaults/main.yml)

3. inventory group

4. inventory host

5. host facts

6. play vars

7. play vars\_prompt

8. play vars\_files
9. role vars (defined in role/vars/main.yml)

10. block vars (only for tasks in block)

11. task vars (only for the task)

12. include\_vars

13. set\_facts / registered vars

14. role (and include\_role) params

15. include params

16. extra vars (for example, -e "user=my\_user")

78

Ansible Playbooks



Definición de Variables

Un nombre de variable solo puede incluir letras, números y guiones bajos.

- Las palabras clave de Python y Ansible no son nombres de variables válidos
- Los nombres de variables no pueden comenzar con un número, pero pueden comenzar con un carácter de subrayado

Después de definir una variable, utilice la sintaxis Jinja2 para hacer referenciarla en los Playbooks:

```
- name: Mi primer playbook, copiar un fichero del servidor a los host administrados
hosts: linux
vars:
  - remote_path: /etc

tasks:
  - name: Copiar fichero demo
    copy: src=/etc/ansible/demos/file-demo.txt dest={{remote_path}}/file-demo.txt
```

79

Ansible Playbooks



Definición de Listas y Diccionarios

Puede definir variables con varios valores mediante listas y diccionarios en YAML:

```
- name: Ejemplo definición de variables
hosts: all
vars:
  - lista1: ["fresa", "manzana", "melón", "naranja"]
  - lista2:
    - fresa
    - manzana
    - melón
    - naranja
  - diccionario: { rol: "web", db: "mariadb", framework: "php7" }
  - diccionario2:
    - rol: web
    - db: mariadb
    - framework: php7
```

80

Ansible Playbooks



Definición de Literales

Puede definir variables con texto en una o varias líneas en YAML:

```
- name: Ejemplo definición de variables
  hosts: all
  vars:
    - una_linea: >
      Contenido del contenido de la línea,
      continuación de la línea
      y fin de la línea.
    - varias_lineas: |
      Línea 1
      Línea 2
      Línea 3
```

playbook-vars.yml

81

Ansible Playbooks



Definición de Variables por Parámetro

Puede definir variables utilizando el parámetro `-e` o `--extra-vars`, formato **clave=valor**:

```
ansible-playbook mi-playbook.yml --extra-vars 'remote_path=/etc version=1.2.77'
ansible-playbook mi-playbook.yml --extra-vars 'lista=["fresa", "manzana", "melón", "naranja"]'
ansible-playbook mi-playbook.yml --extra-vars 'diccionario={rol: "web", db: "mariadb", framework: "php7"}'
ansible-playbook mi-playbook.yml --extra-vars "@some_file.yaml"
```

playbook-vars.yml

82

Ansible Playbooks



Definición de Variables por Parámetro

Puede definir variables utilizando el parámetro `-e` o `--extra-vars`, formato **JSON**:

```
ansible-playbook mi-playbook.yml --extra-vars '{"remote_path":"/etc", "version":"1.2.77"}'
ansible-playbook mi-playbook.yml --extra-vars '{"lista":["fresa", "manzana", "melón", "naranja"]}'
ansible-playbook mi-playbook.yml -e '{"diccionario":{"rol": "web", db: "mariadb", framework: "php7"}}'
ansible-playbook mi-playbook.yml --extra-vars "@some_file.json"
```

playbook-vars.yml

83

LABS

Utilizando el módulo `debug` pintamos el contenido de las variables.

✓

Creamos un `playbook` con diferentes variables

✓

Creamos tareas para mostrar el contenido de las variables utilizando el modulo `debug`.

- name: Pintamos el contenido de Las variables

debug: var=varias\_lineas

- name: Pintamos el contenido de una posición de La lista

debug: msg="Elemento posición 2: {{lista1[2]}}"

- name: Pintamos el contenido de una clave de un diccionario

debug: msg="rol: {{diccionario1["rol"]}}"

- name: Pintamos el contenido de La variable creada utilizando el parámetro `-e`

debug: msg="El path de destino es {{remote\_path}}"

LAB 1

84

---

---

---

---

---

---

---

---

Ansible Playbooks

Variables Facts

Ansible puede trabajar con variables que contienen información sobre sus host remotos o sobre el propio Ansible.

Las variables relacionadas con los host remotos se denominan hechos (facts), y las variables de Ansible se denominan variables mágicas.

Para ver la información *sin procesar* tal como se recopiló, ejecute este comando en la línea de comandos:

ansible <hostname> -m setup

ansible <hostname> -m setup -a "filter=ansible\_hostname"

ansible <hostname> -m setup | grep -e ansible\_hostname

85

---

---

---

---

---

---

---

---

Ansible Playbooks

Variables Facts en los Playbooks

Para ver la información en la variables de los **Playbooks**:

- name: Muestra Facts

hosts: Localhost

tasks:

- name: Muestra el contenido de todos los hechos

debug:

var: ansible\_facts

Podemos deshabilitar este comportamiento de acceso a los hechos mediante la configuración de `INJECT_FACTS_AS_VARS`, en el archivo de configuración `/etc/ansible/ansible.cfg`:

[default]

inject\_facts\_as\_vars = False

playbook-facts.yml

86

---

---

---

---

---

---

---

---

LABS

Mostramos el contenido de algunos Facts de los host remotos.

✓ Creamos un playbook y mostramos el nombre del nodo:

```
{{ansible_facts['nodename']}}
{{ansible_facts['hostname']}}
{{ansible_facts['fqdn']}}
```

✓ Mostramos la versión de Python.

```
{{ansible_facts['python']['version']['major']}}
{{ansible_facts['python']['version']['minor']}}
{{ansible_facts['python']['version']['micro']}}
{{ansible_facts['python']['version']['releaselevel']}}
```



87

---

---

---

---

---

---

---

Ansible Playbooks

Registro de Variables

Creamos variables para almacenar la salida de una tarea utilizando la palabra clave **register**, estas variables se pueden utilizar en cualquier tarea posterior de tu playbook.

```
--
- name: Example Code
  hosts: web_servers
  tasks:
    - name: Run a shell command and register its output as a variable
      shell: /usr/bin/foo
      register: foo_result
      ignore_errors: true

    - name: Run a shell command using output of the previous task
      shell: /usr/bin/bar
      when: foo_result.rc == 5
```

88

---

---

---

---

---

---

---

Ansible Playbooks

Plantillas

Ansible utiliza plantillas **Jinja2** para habilitar expresiones dinámicas y acceso a variables y hechos. Podemos utilizar las plantillas con el módulo de plantilla y en **Playbooks** directamente. Existen filtros para seleccionar y transformar datos. Las plantillas son ejecutas por **Ansible** antes de comenzar la tarea en el host remoto, este enfoque minimiza los requisitos del paquete en el destino y la cantidad de datos que **Ansible** transmite al host remoto.

89

---

---

---

---

---

---

---

Formación del 14/11 al 17/11

29

Ansible Playbooks



Sintaxis en las plantillas

Los módulos `template` y `win_template` utilizan el lenguaje de plantillas **Jinja2**.  
Existen tipos de delimitadores, los predeterminados están configurados de la siguiente manera:

- `{{ ... }}` para *Expresiones* que muestran el resultado en la salida de la plantilla
- `{# ... #}` para *Comentarios* que no son incluidos en la salida de la plantilla
- `{% ... %}` para *Sentencias* de control, repetición, escape, ...

90

Ansible Playbooks



Sintaxis en las plantillas, ejemplo de Expresiones

Ejemplos de *Expresiones*:

- `{{ remote_path }}`
- `{{ ansible_fqdn }}`
- `{{ ansible_facts['python']['version']['releaselevel'] }}`
- `{{ ansible_facts.python.version.releaselevel }}`
- `{{ releaselevel | ansible_facts['python']['version'] }}`
- `{{ items | join(',') }}`
- `Total: {{ items | sum() }}`
- `Total: {{ items | sum(attribute='price') }}`

91

Ansible Playbooks



Sintaxis en las plantillas, ejemplo de Expresiones

Ejemplos de *Sentencias* comunes:

if/endif	for/endifor
<pre>&lt;div&gt; {% if ansible_distribution == "Debian" %} &lt;p&gt;&lt;b&gt;OS:&lt;/b&gt; Debian&lt;/p&gt; {% endif %} &lt;/div&gt;</pre>	<pre>{% for item in items %} {{ item }} {% endfor %}</pre>
raw	include
<pre>{% raw %} &lt;ul&gt;&lt;li&gt;{{ item }}&lt;/li&gt;&lt;/ul&gt; {% endraw %}</pre>	<pre>{% include "child.tpl" %}</pre>

92

LABS

Realizamos la copia de un fichero al host remoto creado a partir de una plantilla.

✓

Reutilizamos *mi-primer-playbook.yml*

✓

Creamos una plantilla con el siguiente contenido:

Este archivo representa una plantilla Jinja2.  
El archivo se crea en el path: {{remote\_path}}  
en el host remoto {{ansible\_fqdn}}.

✓

Modificamos la tarea *copy*:

```
task:
  - name: Copiar fichero demo
    template:
      src: /etc/ansible/templating/lab2.3.j2
      dest: '{{remote_path}}/file-demo.txt'
```

93

---

---

---

---

---

---

---

---

Sintaxis de Jinja

Jinja es un motor de plantillas rápido, expresivo y extensible

Los marcadores de posición especiales en la plantilla permiten escribir código similar a la sintaxis de Python.  
Luego, pasan datos a la plantilla para crear el documento final.

<https://jinja.palletsprojects.com/en/3.1.x/>

<https://jinja.palletsprojects.com/en/3.1.x/>

94

---

---

---

---

---

---

---

---

Ansible Playbooks

Handlers o Controladores

En ocasiones necesitamos que una tarea se ejecute solo cuando se realiza otra tarea en host administrado que finaliza notificando un cambio (**changed=1**).

Ansible utiliza controladores para abordar este caso de uso.

Los **controladores son tareas** que solo se ejecutan cuando otra tarea notifica cambios.

Una tarea pueden indicar uno o más de un controlador para que se ejecuten, utilizando el nombre de los controladores. También podemos referirnos al tema que se definen mediante **listen** y puede agrupar varios controladores.

De forma predeterminada, los controladores se ejecutan después de que se hayan completado todas las tareas de una jugada determinada.

95

---

---

---

---

---

---

---

---

```
---
- name: Ejemplo trabajando con Handlers
  hosts: all
  become: True
  vars:
    - file_src: /etc/ansible/demos/sshd_config

  tasks:
    - name: Copiar el fichero de configuración
      copy:
        src: '{{file_src}}'
        dest: /etc/ssh/sshd_config
        notify: "Reiniciar el servidor sshd"

  handlers:
    - name: Reiniciar el servidor sshd
      service:
        name: sshd
        state: restarted
        listen: "Reiniciar Servicios"
...

```

Utilizamos **notify** para indicar el nombre del *handler* a ejecutar cuando la tarea, con **changed=1**.

**notify** también puede contener el valor de un **listen**.

Podemos utilizar el mismo valor para **listen** en diferentes *handler*.

Si **notify** contiene el valor de un **listen** que se repite en varios *handlers*, se ejecutarán todos los *handlers* secuencialmente.

96

---

---

---

---

---

---

---

---

Ansible Playbooks

Trabajando con Include

Include hace posible dividir los Playbooks en varias partes y así facilitar la edición.

- hacer una llamada a otros ficheros con una lista tareas para ejecutar en el playbook actual
- es compatible con host remotos Windows
- se considera obsoleto a partir de la versión 2.16

Alternativamente podemos utilizar otros módulos:

- import\_playbook
- include\_task y import\_task
- include\_role y import\_role

97

---

---

---

---

---

---

---

---

Ansible Playbooks

Include VS Import

Las instrucciones include e import son similares, pero el motor de Ansible las trata de manera muy diferente.

- Todas las instrucciones import\* se procesan previamente
- Todas las instrucciones include\* se procesan durante la ejecución del libro

98

---

---

---

---

---

---

---

---



Ansible Playbooks



Include VS Import

Las instrucciones `include` e `import` son similares, pero el motor de Ansible las trata de manera muy diferente:

- Todas las instrucciones `import*` son estáticas:
  - preprocesa previamente todas las importaciones estáticas durante el tiempo de análisis de playbook
  - las opciones de la tarea principal se copiarán en todas las tareas secundarias contenidas en la importación
- Todas las instrucciones `include*` son dinámicas:
  - se procesan durante el tiempo de ejecución en el punto en el que se encuentra esa tarea
  - las opciones de la tarea solo se aplicarán a la tarea dinámica y no se copiarán a las tareas secundarias

---

---

---

---

---

---

---

99

Ansible Playbooks



Importación de Playbooks

Es posible incluir playbooks dentro de un playbook. Por ejemplo:

```
---
- import_playbook: webservers.yml
- import_playbook: databases.yml
```

---

---

---

---

---

---

---

100

Ansible Playbooks



Inclusión e importación de archivos de tareas

Un archivo de tareas para `include` o `import` simplemente contiene una lista plana de tareas:

Podemos utilizar para incluir las tareas de un archivo en las tareas del playbook:

```
# common_tasks.yml
---
- name: placeholder foo
  command: /bin/foo
- name: placeholder bar
  command: /bin/bar

tasks:
- import_tasks: common_tasks.yml
# or
- include_tasks: common_tasks.yml
# or
- include: common_task.yml
```

---

---

---

---

---

---

---

101

Ansible Playbooks



Inclusión e importación de archivos de tareas II

También puede pasar variables a los `import` e `includes`:

```
tasks:
- import_tasks: wordpress.yml wp_user=timmy
- import_tasks: wordpress.yml wp_user=alice
- import_tasks: wordpress.yml wp_user=bob
```

102

---

---

---

---

---

---

---

Ansible Playbooks



Inclusión e importación de archivos de tareas III

`include` o `import` también se puede utilizar en la sección `handlers`:

```
# more_handlers.yml
---
- name: restart apache
  service: name=apache state=restarted

handlers:
- include_tasks: more_handlers.yml
# or
- import_tasks: more_handlers.yml
```

103

---

---

---

---

---

---

---

Ansible Playbooks



Trabajando con `import` e `include`

Observamos que uso del módulo `include` es obsoleto a partir de las versión 2.16.

```
ansible$ ansible-playbook playbook-imports.yml --syntax-check
[DEPRECATION WARNING]: "include" is deprecated, use include_tasks/import_tasks instead. This feature will be
removed in version 2.16. Deprecation warnings can be disabled by setting deprecation_warnings=False in
ansible.cfg.

playbook: playbook-imports.yml

ansible$ ansible-playbook playbook-imports2.yml --syntax-check
playbook: playbook-imports2.yml
```

104

---

---

---

---

---

---

---

Ansible Playbooks

Trabajando con import e include

Observamos que uso del módulo `include` es obsoleto a partir de las versión 2.16.

```
ansible$ ansible-playbook playbook-imports2.yml
PLAY [Playbook de ejemplo para comprobar el funcionamiento de imports e includes] *****
TASK [Gathering Facts] *****
ok: [host02]
TASK [Instalar vim] *****
ok: [host02]
TASK [include_tasks] *****
included: /etc/ansible/links/tasks.yml for host02
TASK [Instalar Apache2] *****
ok: [host02]
TASK [Habilitar el servicio de Apache2] *****
ok: [host02]
TASK [Iniciar el servicio de Apache2] *****
ok: [host02]
PLAY RECAP *****
                ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
host02
```

105

Ansible Playbooks

Roles

Mediante los **Roles** cargaremos automáticamente vars, archivos, tareas, controladores y otros artefactos de Ansible, pudiendo reutilizarlos y compartirlos.

De forma predeterminada, Ansible busca roles en las siguientes ubicaciones:

- en colecciones, si las está utilizando
- en un directorio llamado `/roles`, relativo al archivo Playbook
- en el `roles_path` configurado, las rutas de búsqueda predeterminada son:
  - `~/ansible/roles`
  - `/usr/share/ansible/roles`
  - `/etc/ansible/roles`
- en el directorio donde se encuentra el archivo Playbook

106

Ansible Playbooks

Roles

Utilizaremos una estructura de archivos conocida para la creación de **Roles**.

```
roles/
  common/
    tasks/main.yml      # this hierarchy represents a "role"
    handlers/main.yml   # <-- tasks file can include smaller files if warranted
    templates/           # <-- handlers file
    ntp.conf.j2          # <-- files for use with the template resource
    files/               # <-- templates end in .j2
    bar.txt              # <-- files for use with the copy resource
    foo.sh               # <-- script files for use with the script resource
  vars/main.yml         # <-- variables associated with this role
  defaults/main.yml     # <-- default lower priority variables for this role
  meta/main.yml          # <-- role dependencies
  library/               # roles can also include custom modules
  module_utils/          # roles can also include custom module_utils
  lookup_plugins/        # or other types of plugins, like lookup in this case
```

107

[illegible]

LABS

Creación de un Rol para la instalación del servidor web Apache2, continuación.

✓

Creamos el *Playbook* que utiliza el *Rol* apache2:  
`sudo nano /etc/ansible/playbook-roles.yml`

```
---
- name: Trabajando con Roles
  hosts: all
  remote_user: Administrador
  become: True

  roles:
    - apache2
```

playbook-roles.yml

---

---

---

---

---

---

---

---

111

Ansible Playbooks

Trabajando con Roles

```
ansible$ ansible-playbook playbook-roles.yml

PLAY [Trabajando con Roles] *****
TASK [Gathering Facts] *****
ok: [host02]
TASK [apache2 : Instalar Apache2] *****
ok: [host02]
TASK [apache2 : Habilitar e iniciar el servicio de Apache2] *****
ok: [host02]
TASK [apache2 : Copiar el fichero index.html] *****
changed: [host02]
RUNNING HANDLER [apache2 : Reiniciar httpd] *****
changed: [host02]
PLAY RECAP *****
host02      : ok=5  changed=2  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

---

---

---

---

---

---

---

---

112

Ansible Playbooks

Trabajando con Playbooks

**Ansible** proporciona configuraciones para ayudarte a obtener el comportamiento y el resultado que deseas en cada *playbook*:

- es posible que necesitemos ejecutar diferentes tareas, dependiendo del *host* remoto, del valor de una variable o de un resultado anterior
- es posible que desee que un error si o no detenga las ejecuciones en otros *host* remotos
- es posible que desee que el valor de algunas variables dependa del valor de otras variables
- es posible que necesite ejecutar una tarea de forma repetida con diferentes valores

---

---

---

---

---

---

---

---

113

Formación del 14/11 al 17/11

37

Ansible Playbooks



Condiciones en los Playbooks

Los hechos o *Facts* son atributos de hosts remotos, incluida la dirección IP, el sistema operativo, el estado de un sistema de archivos, etc. Podemos crear condiciones basadas en hechos:

- instalar un paquete solo cuando el sistema operativo es una versión determinada
- omitir la configuración de un firewall en hosts con direcciones IP internas
- realizar tareas de limpieza sólo cuando un sistema de archivos se está llenando

---

---

---

---

---

---

---

114

Condiciones en los Playbooks

```
---
- name: Probando condicionales en imports, includes, roles y tareas
  hosts: all
  remote_user: Administrador

  tasks:
    - name: Obtener la configuración de red en Linux
      command: ifconfig
      when: ansible_system == "Linux"
    - name: Obtener la configuración de red en Windows
      win_command: ipconfig
      when: ansible_system == "Win32NT"
```

host indica los host administrados del inventario en los que se ejecutan las tareas del playbook.

when indica la condición que de cumplir el host administrado para que se ejecute la tarea.

---

---

---

---

---

---

---

115

Ansible Playbooks



Bucles en los Playbooks

Ansible ofrece la posibilidad de ejecutar una tarea varias veces:

- cambiar la propiedad de varios archivos y/o directorios
- crear múltiples usuarios
- repetir un tarea de sondeo hasta que se alcance un determinado resultado
- etc.

Utilizamos `with_*` para definir un bucle dentro de una tarea. La palabra `loop` es equivalente a `with_list` y `with_items`, y una mejor opción para bucles simples.

---

---

---

---

---

---

---

116

Ansible Playbooks



Bucles en los Playbooks

Desde la versión 2.8 podemos obtener información extendida sobre el bucle:

ansible_loop.allitems	La lista de todos los elementos del bucle
ansible_loop.index	Iteración actual del bucle. (1 indexado)
ansible_loop.index0	Iteración actual del bucle. (0 indexado)
ansible_loop.revindex	El número de iteraciones desde el final del bucle (1 indexado)
ansible_loop.revindex0	El número de iteraciones desde el final del bucle (0 indexado)
ansible_loop.first	True, si la primera iteración
ansible_loop.last	True, si es la última iteración
ansible_loop.length	El número de elementos en el bucle
ansible_loop.previtem	Elemento de la iteración anterior del bucle. Indefinido durante la primera iteración.
ansible_loop.nextitem	El elemento de la siguiente iteración del bucle. Sin definir durante la última iteración.

117

---

---

---

---

---

---

---

---

**Bucles en los Playbooks**

```
***
- name: Probando bucles en tareas
  hosts: all
  vars:
    packages: ['apache2', 'php', 'curl']
  tasks:
    - name: Instalar paquetes en Ubuntu
      apt:
        name: '{{item}}'
        state: latest
        become: True
        with_items: '{{packages}}'
        when: ansible_distribution == "Ubuntu" or ansible_distribution == "Debian"

    - name: Instalar paquetes en Centos
      yum:
        name: '{{item}}'
        become: True
        with_items: '{{packages}}'
        when: ansible_distribution == "CentOS"
***
```

Utilizamos **with\_items** para indicar los elementos de un bucle dentro de una tarea.

118

---

---

---

---

---

---

---

---

**Register en los Playbooks**

```
***
- name: Probando creación de variables con register
  hosts: all
  tasks:
    - name: Obtener la configuración de red en Linux
      command: ifconfig
      register: resultado

    - name: Mostrar el contenido de la variable resultado
      debug:
        var: resultado

    - name: Mostrar Changed
      debug:
        msg: 'Estado de Changed: {{resultado.changed}}'

    - name: Mostrar Failed
      debug:
        msg: 'Estado de Failed: {{resultado.failed}}'

    - name: Mostrar stdout
      debug:
        msg: 'Estado de stdout: {{resultado.stdout}}'
***
```

Utilizamos **register** para crear una variable que recoge el resultado de la ejecución de una tarea.

119

---

---

---

---

---

---

---

---

### Ignorar Errores en los Playbooks

```
---
- name: Probando creación el uso de ignore_errors
  hosts: all
  remote_user: Administrador

  tasks:
    - name: Obtener la configuración de red en Linux
      command: cat /etc/filedemo.cfg
      register: resultado
      ignore_errors: True

    - name: Mostrar el mensaje de salida
      debug:
        msg: 'Mensaje: {{resultado.stdout_lines}}'

    - name: Mostrar Changed
      debug:
        msg: 'Estado de Changed: {{resultado.changed}}'
```

ignore\_errors permite ignorar los errores, recoger el resultado en una variable utilizando register y continuar con las siguientes tareas

120

---

---

---

---

---

---

---

---

### Ansible Playbooks

#### Ignorar Errores en los Playbooks

Cuando no utilizamos **ignore\_errors** y se produce un error la ejecución finaliza y resto de tareas no se ejecutan el host remoto.

```
ansible$ ansible-playbook playbook-ignore_errors.yml

PLAY [Probando creación el uso de ignore_errors] *****
TASK [Gathering Facts] *****
ok: [host82]

TASK [Obtener la configuración de red en Linux] *****
fatal: [host82]: FAILED! => {"changed": true, "cmd": ["cat", "/etc/filedemo.cfg"], "delta": "0:00:00.003708", "end": "2022-11-01 17:09:12.912258", "msg": "non-zero return code", "rc": 1, "start": "2022-11-01 17:09:12.928598", "stderr": "cat: /etc/filedemo.cfg: No such file or directory", "stdout_lines": ["cat: /etc/filedemo.cfg: No such file or directory"], "stdout": "", "stdout_lines": []}

PLAY RECAP *****
host82      : ok=1  changed=0  unreachable=0  failed=1  skipped=0  rescued=0  ignored=0
```

121

---

---

---

---

---

---

---

---

### Ansible Playbooks

#### Ignorar Errores en los Playbooks

Al utilizar **ignore\_errors** se ejecución de tareas en el host remoto continúa y es posible recoger en una variable el resultado de la tarea que ha producido el error.

```
ansible$ ansible-playbook playbook-ignore_errors-end.yml

PLAY [Probando creación el uso de ignore_errors] *****
TASK [Gathering Facts] *****
ok: [host82]

TASK [Obtener la configuración de red en Linux] *****
fatal: [host82]: FAILED! => {"changed": true, "cmd": ["cat", "/etc/filedemo.cfg"], "delta": "0:00:00.003594", "ignore_errors": true, "start": "2022-11-01 17:09:12.912258", "stderr": "cat: /etc/filedemo.cfg: No such file or directory", "stdout_lines": ["cat: /etc/filedemo.cfg: No such file or directory"], "stdout": "", "stdout_lines": []}

TASK [Mostrar el mensaje de salida] *****
ok: [host82] => {"msg": "Mensaje: []"}

TASK [Mostrar Changed] *****
ok: [host82] => {"msg": "Estado de Changed: True"}

PLAY RECAP *****
host82      : ok=4  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=1
```

122

---

---

---

---

---

---

---

---



Ansible Playbooks



Definido las Tareas como fallidas o modificadas en los Playbooks

Ansible le permite definir cuándo una tarea en particular *ha cambiado* o *fallado*:

- podemos definir una tarea como *fallida* usando el condicional `failed_when`
- Podemos definir una tarea como que *ha cambiado* usando el condicional `changed_when`

De forma predeterminada, Ansible continúa ejecutando tareas siempre que haya hosts remotos que aún no hayan fallado. Situaciones como una actualización masiva de host puede ser interesante cancelar si alcanzamos un cierto umbral de fallos.

```
---
- hosts: webservers
  max_fail_percentage: 30
  serial: 10
```

123

---

---

---

---

---

---

---

---

Ansible Playbooks



Etiquetas

Cuando trabajamos con *Playbooks* grandes, puede ser útil ejecutar solo partes específicas de el en lugar de ejecutar todo.

Ansible utiliza las etiquetas para ejecutar u omitir tareas. Las etiquetas se puede añadir al playbook, bloques, tareas, roles y elementos importados.

Seleccione las tareas que desea ejecutar indicando sus etiquetas:

```
ansible-playbook -i hosts myplaybook.yml --tags "mytag"
```

124

---

---

---

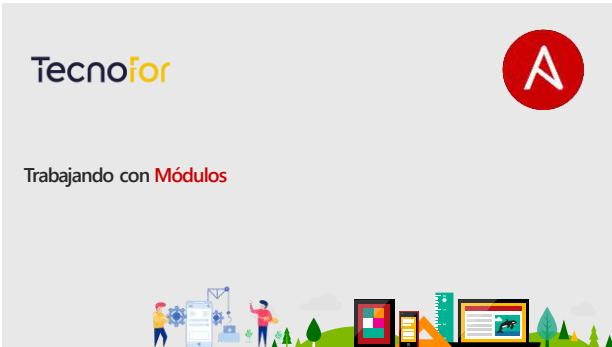
---

---

---

---

---



125

---

---

---

---

---

---

---

---

Trabajando con Módulos



¿Qué son los Módulos?

Los módulos son unidades discretas de código que se pueden usar desde la línea de comandos o en una tarea de **Playbooks**.

Ansible ejecuta cada módulo, generalmente en el host o dispositivo administrado, y recopila los valores devueltos:

- admiten argumentos, delimitados por espacios y formato **clave=valor**
- algunos argumentos son obligatorios y otros tienen valores por defecto
- devuelven **datos** en formato JSON
- pueden **escribir** en cualquier lenguaje de programación
- **deben** evitar realizar cambios si detectan que el estado actual coincide con el estado final deseado

126

Colecciones de Módulos



Listado de Colecciones de Módulos

Desde este enlace podemos consultar las colecciones de módulos, los módulos que contienen, y una descripción detallada del uso y la sintaxis de cada módulo.

<https://docs.ansible.com/ansible/latest/collections/index.html>

Podemos ver los módulos disponibles en nuestro host mediante:

**ansible-doc -l**

<https://docs.ansible.com/ansible/latest/collections/index.html>

127

Trabajando con Módulos



Comando ansible-doc

Para ver un listado de todos los módulos disponible en nuestro equipo de **Ansible** utilizamos el parámetro **-l**.

```
ansible$ ansible-doc -l
add_host          Add a host (and alternativ...
amazon.aws.az_facts  Gather information about a...
amazon.aws.az_info  Gather information about a...
amazon.aws.caller_facts  Get information about the ...
amazon.aws.s3        Get information about the ...
amazon.aws.cloudformation  Create or delete an AWS Cl...
amazon.aws.cloudformation_facts  Obtain information about a...
amazon.aws.cloudformation_info  Obtain information about a...
amazon.aws.ec2       create, terminate, start o...
amazon.aws.ec2_ami   Create or destroy an image...
amazon.aws.ec2_ami_facts  Gather information about e...
amazon.aws.ec2_ami_info  Gather information about e...
```

128

Trabajando con Módulos

Indica la colección del módulo.

A

Comando ansible-doc

Para ver información detallada de un módulo, descripción, argumentos, ejemplos, .... escribimos como parámetro el nombre del módulo.

```
ansible$ ansible-doc template
```

```
> ansible.builtin.template <- (/usr/lib/python3.9/site-packages/ansible/modules/template.py)

Templates are processed by the Jinja2 templating language
<http://jinja.pocoo.org/docs/>. Documentation on the template formatting can be found in
the Template Designer Documentation <http://jinja.pocoo.org/docs/templates/>. Additional
variables listed below can be used in templates. 'ansible_managed' (configurable via the
'defaults' section of 'ansible.cfg') contains a string which can be used to describe the
template name, host, modification time of the template file and the owner uid.
'ansible_host' contains the node name of the template's machine. 'template_uid' is the
numeric user id of the owner. 'template_path' is the path of the template.
'ansible_path' is the absolute path of the template. 'template_destpath' is the
path of the template on the remote system (added in 2.8). 'template_run_data' is the
date that the template was rendered.
:
```

129

Trabajando con Módulos

ansible.builtin.debug

Muestra expresiones como mensajes o los valores de las variables durante la ejecución, sin detener necesariamente el *playbook*.

Útil para depurar junto con la directiva *when*: y compatible con host de Windows.

Los parámetro para el módulos **ansible.builtin.debug** son:

- **msg**, expresión o mensaje (*Hello World!*=predeterminado)
- **var**, nombre de la variable a depurar, no se puede utilizar junto a **msg**
- **verbosity**, valor numérico que controla cuándo se ejecuta la depuración, por ejemplo, si se establece en 3 sólo se ejecutará depurar cuando -vvv o superior.

130

ansible.builtin.debug

A

```
tasks:
  - name: Print the gateway for each host when defined
    debug:
      msg: System {{ inventory_hostname }} has gateway {{ ansible_default_ipv4.gateway }}
    when: ansible_default_ipv4.gateway is defined

  - name: Get uptime information
    shell: /usr/bin/uptime
    register: result

  - name: Print return information from the previous task
    debug:
      var: result
      verbosity: 2

  - name: Display all variables/facts known for a host
    debug:
      var: hostvars[inventory_hostname]
      verbosity: 4
```

131

Trabajando con Módulos



Módulos de Ficheros

Algunos módulos para el tratamiento de ficheros:

- `ansible.builtin.blockinfile` permite inserta, actualiza o elimina un bloque de texto dentro de un fichero
- `ansible.builtin.copy`, para copiar ficheros desde el servidor de Ansible a los hosts remotos
- `ansible.builtin.fetch`, para copiar ficheros de los hosts remotos al servidor de Ansible, lo contrario de `copy`
- `ansible.builtin.file`, permite establecer atributos a los ficheros de los host remotos
- `ansible.builtin.find`, realiza búsqueda de ficheros a partir de un patrón
- `ansible.builtin.lineinfile`, permite trabajar con líneas de texto en archivos
- `ansible.builtin.replace`, reemplazar todas las instancias de una cadena determinada en un archivo mediante una expresión regular a la que se hace referencia
- `ansible.builtin.stat`, recuperar el estado del archivo o del sistema de archivos
- `ansible.builtin.tempfile`, utilizado para crear archivos o directorios temporales
- `ansible.builtin.template`, para copiar el un archivo a partir de una plantilla a los hosts remotos

132

Trabajando con Módulos



Módulos de Ficheros II

- `community.general.ini_file`, ajustar la configuración en archivos INI
- `community.general.iso_extract`, copiar (opcional) y extraer archivos en los hosts remotos
- `community.general.unarchive`, crea un archivo comprimido desde uno o más archivos, o directorios
- `community.general.xattr`, administrar atributos extendidos definidos por el usuario
- `ansible.posix.patch`, aplicar archivos de parche usando la herramienta de parche GNU

133

Trabajando con Módulos



`ansible.builtin.copy`

Argumentos	Descripción
<code>dest</code>	Ruta absoluta remota donde se debe copiar el archivo
<code>backup</code>	Cree un archivo de copia de seguridad (predeterminado=no)
<code>content</code>	Se utiliza en lugar de <code>src</code> para establecer el contenido de un archivo
<code>force</code>	Indica si el archivo remoto debe reemplazarse (predeterminado=si)
<code>owner</code>	Nombre del usuario que debería ser propietario del archivo
<code>group</code>	Nombre del grupo que debería ser propietario del archivo
<code>mode</code>	Los permisos del archivo o directorio de destino
<code>src</code>	Ruta de local del archivo para copiar en el servidor remoto

134

Trabajando con Módulos



ansible.builtin.template

Argumentos	Descripción
dest	Ruta absoluta remota donde se debe copiar el archivo
backup	Cree un archivo de copia de seguridad (predeterminado=no)
content	Se utiliza en lugar de src para establecer el contenido de un archivo
force	Indica si el archivo remoto debe remplazarse (predeterminado=si)
owner	Nombre del usuario que debería ser propietario del archivo
group	Nombre del grupo que debería ser propietario del archivo
mode	Los permisos del archivo o directorio de destino
src	Ruta de local del archivo para copiar en el servidor remoto

135

---

---

---

---

---

---

---

---

Trabajando con Módulos



ansible.builtin.file

Argumentos	Descripción
path	Ruta de acceso al archivo para administrar.
backup	Cree un archivo de copia de seguridad (predeterminado=no)
force	Indica si el archivo remoto debe remplazarse (predeterminado=no)
owner	Nombre del usuario que debería ser propietario del archivo
group	Nombre del grupo que debería ser propietario del archivo
mode	Los permisos del archivo o directorio de destino
state	absent, directory, file -- (predeterminado), hard, link, touch

module: ansible.builtin.file.yml

136

---

---

---

---

---

---

---

---

ansible.builtin.stat



```
---
- name: Módulos de Ficheros
  hosts: all

  tasks:
    - name: Comprobar que un directorio existe
      stat:
        path: /tmp
        register: info
    - name: Ver el contenido de la variable info
      debug:
        var: info
    - name: Es un directorio /tmp
      debug:
        msg: "/tmp es un directorio: {{info.stat.isdir}}"
    - name: Copiar fichero demo
      copy:
        content: 'Contenido del fichero Demo.'
        backup: yes
        dest: /tmp/file-demo.txt
        when: info.stat.isdir == True
  ...
```

**ansible.builtin.stat**

Nos permite recuperar información sobre un path, almacenarlo en un variable y utilizar la información en las siguientes tareas.

module: ansible.builtin.stat.yml

137

---

---

---

---

---

---

---

---

Trabajando con Módulos



ansible.builtin.lineinfile

Argumentos	Descripción
path	Ruta de acceso al archivo a modificar.
line	Línea que se va a insertar o reemplazar en el archivo
search_string	La cadena literal que se va a buscar en cada línea del archivo
regexp	Expresión regular que se va a buscar en cada línea del archivo
state	Si la línea debe estar allí o no ( <i>absent</i> o <i>present</i> predeterminado= <i>present</i> )
backup	Cree un archivo de copia de seguridad (predeterminado= <i>no</i> )
owner	Nombre del usuario que debería ser propietario del archivo
group	Nombre del grupo que debería ser propietario del archivo
mode	Los permisos del archivo

139

ansible.builtin.lineinfile

```
tasks:
  - name: Instalar paquetes en Ubuntu
    apt:
      name: apache2
      state: latest
      become: true
      when: ansible_distribution == "Ubuntu" or ansible_distribution == "Debian"

  - name: Reemplaza el puerto 80 por el 8080
    lineinfile:
      path: /etc/apache2/sites-available/000-default.conf
      backup: yes
      search_string: '80'
      line: '<VirtualHost *:8080>'
      state: present
      become: true
      register: apacheconfig

  - service:
      name: apache2
      state: reloaded
      become: true
      when: apacheconfig.changed
```

ansible.builtin.lineinfile

Al encontrar el valor de search\_string en una línea, reemplaza el contenido entero de la línea por el valor de line.

140

Trabajando con Módulos



Módulos para la Gestión de Paquetes

Algunos módulos para gestionar paquetes y dependencias en desarrollo:

- community.general.bower, administrar paquetes Bower (desarrollo web)
- community.general.bundler, administrar las dependencias de Ruby Gem
- community.general.composer, administrar dependencias de PHP
- community.general.cpanm, administrar las dependencias de Perl
- community.general.easy\_install, administrar e instalar librerías de Python
- community.general.maven\_artifact, descargar artefactos de un repositorio Maven
- community.general.npm, administrar paquetes de node.js a través del comando npm
- community.general.pear, administrar paquetes pear/pecl
- ansible.builtin.pip, administrar las dependencias de librerías de Python

141

Trabajando con Módulos



ansible.builtin.pip

Argumentos	Descripción
chdir	Cambiar al directorio antes de ejecutar el comando
executable	Para especificar explícitamente ruta y ejecutable
extra_args	Argumentos extras para pip
name	Nombre o URL de la librería o paquete a instalar, admite una lista
state	Estado del módulo ( <i>absent</i> , <i>forcereinstall</i> , <i>latest</i> o <i>present</i> predeterminado= <i>present</i> )
version	Número de versión de la librería o paquete
virtualenv	Ruta del entorno virtual, opcional
virtualenv_command	Ruta y/o nombre del comando para crear el entorno virtual (default=virtualenv)
virtualenv_python	El ejecutable de Python utilizado para crear el entorno virtual.

142

ansible.builtin.pip

```
tasks:
  - name: Instalar String Color, última versión
    pip:
      name: color
      state: latest
  - name: Instalar Bottle versión 0.11.1
    pip:
      name: bottle
      version: 0.11.1
      state: present
  - name: Instalar Django versión 3.2.16
    pip:
      name: django==3.2.16
      state: present
  - name: Instalar requerimientos de Python en un (virtualenv)
    pip:
      requirements: /home/ey_app/requirements.txt
      virtualenv: /home/ey_app/venv
      virtualenv_command: virtualenv
      become: True
```

ansible.builtin.pip

Diferentes ejemplos para instalar librerías y paquetes de Python.

Los valores para state son:

- absent
- forcereinstall
- latest
- present, valor predeterminado

143

Trabajando con Módulos



Módulos para la Gestión de Paquetes

Algunos módulos comunes para gestionar paquetes en sistemas operativos:

- `community.general.apk`, administrar paquetes `apk` para *Android*
- `ansible.builtin.apk`, administrar paquetes `apt` para *Debian, Ubuntu, ...*
- `ansible.builtin.apk_key`, agregar o eliminar claves `apt`
- `ansible.builtin.apk_repository`, agregar o eliminar repositorios `apt`
- `ansible.builtin.dnf`, administrar paquetes `dnf` para *Fedora*
- `ansible.general.opkg`, administrar paquetes *OpenWrt* (suelo ser utilizado por routers)
- `ansible.builtin.package`, administrar paquetes de forma genérica para diferentes sistemas operativos
- `ansible.general.pkg5`, administrar paquetes para *Solaris 11*
- `ansible.general.redhat_subscription`, administrar registro y suscripciones a *RHSM*
- `ansible.builtin.yum`, administrar paquetes `yum` para *CentOS, Red Hat, ...*
- `ansible.general.yum_repository`, agregar y eliminar repositorios `yum`

144

Trabajando con Módulos



ansible.builtin.apt

Utilizamos el módulo **ansible.builtin.apt** para instalar paquetes y sus dependencias, así como actualizar el sistema operativo.

El parámetro **name** indica el nombre del paquete mediante el formato *nombre=versión*.

El parámetro **state** se utiliza para indicar el estado deseado del paquete, posibles valores:

- **absent**, para indicar que el paquete no debe estar instalado
- **build-dep**, asegura la instalación de las dependencias del paquete
- **latest**, asegura que la última versión del paquete está instalada
- **present**, indica que debe estar instalado pero no necesariamente debe ser la última versión
- **fixed**, corrige errores instalando las dependencias del paquete

145

Trabajando con Módulos



ansible.builtin.apt

La actualización del sistema operativo se realiza mediante el parámetro **upgrade**, posibles valores:

- **dist**, realiza una actualización *apt-get dist-upgrade*
- **full**, realiza una actualización completa *upgrade*
- **no**, no actualiza el sistema operativo, valor predeterminado
- **safe**, realiza una actualización *safe-upgrade*
- **yes**, realiza una actualización *safe-upgrade*

Esto no actualiza un paquetes específicos, tenemos que utilizar **state=latest**.

146

ansible.builtin.apt



```
tasks:
  - name: Instalar la última versión de "nginx" package
    apt:
      name: nginx
      state: latest

  - name: Desinstalar una lista de paquetes
    apt:
      pkg:
        - nginx
        - nginx-extras
      state: absent
      purge: yes
      autoremove: yes
      when: ansible_distribution == "Ubuntu" or ansible_distribution == "Debian"

  - name: Actualizar todos los paquetes a las últimas versiones
    apt:
      name: "*"
      state: latest
      when: ansible_distribution == "Ubuntu" or ansible_distribution == "Debian"

  - name: Actualizar el OS (apt-get dist-upgrade)
    apt:
      upgrade: dist
```

**ansible.builtin.apt**

Diferentes ejemplos para instalar, actualizar y eliminar paquetes del sistema operativo.

Otro ejemplo muestra como actualizar el sistema operativo.

147



### ansible.builtin.apt\_key && ...apt\_repository

```
tasks:
  - name: Añadir claves del repositorio de microsoft
    apt_key:
      url: https://packages.microsoft.com/keys/microsoft.asc
      state: present
      when: ansible_distribution == "Ubuntu" or ansible_distribution == "Debian"

  - name: Añadir el repositorio de microsoft
    apt_repository:
      repo: deb [arch=amd64] https://packages.microsoft.com/repos/edge stable main
      state: present
      filename: microsoft-edge
      when: ansible_distribution == "Ubuntu" or ansible_distribution == "Debian"
```

**ansible.builtin.apt\_key && ansible.builtin.apt\_repository**

Un ejemplo donde añadimos un repositorio y las claves necesarias para conectar de forma segura con el mismo.

include: ansible-galaxy apt\_key and apt\_repository.yml

---

---

---

---

---

---

---

---

148

Trabajando con Módulos

Módulos para Ejecutar Comandos

- Algunos módulos para ejecutar comandos y scripts:
- **ansible.builtin.command** ejecutar comandos en los host administrados
  - **ansible.builtin.raw** ejecutar comando sin pasar por el subsistema de módulos
  - **ansible.builtin.script** ejecutar un script local después de transmitirlo al host administrado
  - **ansible.builtin.shell** ejecutar comandos a través de la Shell **/bin/sh** del host administrado. A diferencia de Shell, Command no permite utilizar operadores como **&&** y **||** o los redireccionamientos **>>**.

149

Trabajando con Módulos

ansible.builtin.command

Argumentos	Descripción
args	Argumentos del comando en forma de lista en lugar de una cadena
chdir	Cambia a este directorio antes de ejecutar el comando
cmd	Comando para ejecutar
creates	Archivo, si existe no se ejecutará (se comprueba antes de removes)
removes	Archivo, si NO existe no se ejecutará (se comprueba después de creates)

150

ansible.builtin.command

A

```
tasks:
  - name: Ejecuta el comando si /path/to/database no existe (todo en una línea)
    command: /usr/bin/make_database.sh db_user db_name creates=/path/to/database

  - name: Ejecuta el comando si /path/to/database no existe (con el parámetro 'args')
    command: /usr/bin/make_database.sh db_user db_name
    args:
      creates: /path/to/database

  - name: Ejecuta el comando si /path/to/database no existe (con el parámetro 'cmd')
    command:
      cmd: /usr/bin/make_database.sh db_user db_name
      creates: /path/to/database
```

Tres ejemplos iguales con diferentes sintaxis del módulo command.

module: ansible.builtin.command.yml

151

---

---

---

---

---

---

---

---

Trabajando con Módulos

ansible.builtin.script

Argumentos	Descripción
chdir	Cambia a este directorio antes de ejecutar el script
cmd	Ruta al script a ejecutar seguido de argumentos opcionales
decrypt	Descifrado automático de archivos mediante Vault (yes=predeterminado)
executable	Nombre o ruta de un ejecutable para invocar el script
creates	Archivo, si existe no se ejecutará (se comprueba antes de removes)
removes	Archivo, si NO existe no se ejecutará (se comprueba después de creates)

152

---

---

---

---

---

---

---

---

ansible.builtin.script

A

```
---
- name: Trabajando con Scripts - Modulo module-ansible.builtin.script
  hosts: all
  remote_user: Administrator
  become: True

  tasks:
    - name: Run a script with arguments (using 'cmd' parameter)
      ansible.builtin.script:
        cmd: /some/local/script.sh --some-argument 1234

    - name: Run a script using an executable in a system path
      ansible.builtin.script: /some/local/script.py
      args:
        executable: python3
  ...
```

module: ansible.builtin.script.yml

153

---

---

---

---

---

---

---

---

Trabajando con Módulos



Módulos para Notificaciones

Es importante registrar la finalización de cualquier *Playbook* y enviar un informe más o menos detallado a las partes interesadas.

Ansible puede enviar correos electrónicos con archivos adjuntos mediante un servidor o servicio SMTP utilizando el módulo `mail`.

Cuenta además con otros módulos para realizar notificaciones como:

- `cisco_spark`
  - `hip_chat`
  - `irc`
  - `jabber`
  - `mqtt`
- `nexmo`
  - `office_365_connector_card`
  - `rocketchat`
  - `say`
  - `syslogger`
- `sendgrid`
  - `slack`
  - `telegram`
  - `twilio`

154

Trabajando con Módulos



mail

Argumentos	Descripción
<code>attach</code>	Una lista de nombres de ruta de acceso de archivos para adjuntar al mensaje
<code>body</code>	El cuerpo del correo electrónico que se está enviando
<code>subject</code>	El asunto del correo electrónico que se está enviando
<code>to</code>	La(s) dirección(es) de correo electrónico a la(s) que se envía el correo
<code>from</code>	La dirección de correo electrónico desde la que se envía el correo
<code>host</code>	El servidor de correo (localhost=predeterminado)
<code>port</code>	El puerto del servidor de correo (25=predeterminado)
<code>username</code>	Usuario, si SMTP requiere nombre de usuario.
<code>password</code>	Contraseña, si SMTP requiere contraseña

155

mail

```
---
- name: Trabajando con Notificaciones - Modulo mail
  hosts: all

  tasks:
    - name: Sending an e-mail using Gmail SMTP servers
      mail:
        host: smtp.gmail.com
        port: 587
        username: username@gmail.com
        password: mysecret
        to:
          - John Doe <j.d@example.org>
          - Suzie Something <su@example.com>
        cc: Charlie Root <root@example.com>
        attach:
          - /etc/group
          - /tmp/avatar2.png
        headers:
          - Reply-To:john@example.com
          - X-Special: "Something or other"
        subject: Ansible-report
        body: System {{ ansible_hostname }} has been successfully provisioned.
---
ansible-mail.yml
```

156

Trabajando con Módulos



Módulos de Bases de Datos

Algunos módulos comunes para gestionar bases de datos MySQL:

- `mysql_db`, agregar o eliminar bases de datos MySQL de un host remoto
- `mysql_info`, recopilar información sobre los servidores MySQL
- `mysql_query`, ejecutar consultas MySQL
- `mysql_replication`, administrar la replicación de MySQL
- `mysql_role`, agregar, eliminar y actualizar un rol de MySQL
- `mysql_user`, agregar o eliminar un usuario de una base de datos MySQL
- `mysql_variables`, administrar variables globales de MySQL

---

---

---

---

---

---

---

157

Trabajando con Módulos



Módulos de Bases de Datos

Algunos módulos comunes para gestionar bases de datos mongoDB:

- `mongodb_parameter`, cambiar un parámetro administrativo en un servidor MongoDB
- `mongodb_replicaset`, inicializa un `replicaset` de MongoDB
- `mongodb_schema`, administra los validadores de esquemas de documentos de MongoDB
- `mongodb_shell`, ejecute comandos a través del `shell` MongoDB
- `mongodb_shutdown`, limpia todos los recursos de la base de datos y luego finaliza el proceso `mongod/mongos`
- `mongodb_status`, valida el estado del `replicaset`
- `mongodb_user`, agrega o elimina un usuario de una base de datos MongoDB

---

---

---

---

---

---

---

158

Trabajando con Módulos



```
---
name: Instalación y administración de MySQL
hosts: all

tasks:
  - name: Instalar conector de MySQL para Python
    pip:
      name: mysql-connector-python
      state: latest

  - name: Crear una base de datos
    mysql_db:
      name: demodb
      state: present

  - name: Copia de seguridad de todas las base de datos
    mysql_db:
      name: all
      state: dump
      target: /home/{{ansible_hostname}}_db.sql
---
```

---

---

---

---

---

---

---

159

Trabajando con Módulos



Módulos para Windows

Algunos módulos comunes son:

- `win_acl`, establecer permisos de archivo/directorio/registro para un usuario o grupo
- `win_chocolatey`, administrar paquetes usando chocolatey
- `win_command`, ejecuta un comando en un nodo remoto de Windows
- `win_copy`, copia archivos en ubicaciones remotas en hosts de Windows
- `win_disk_image`, administrar montajes ISO/VHD/VHDX en hosts Windows
- `win_domain_group`, crear, modificar o eliminar grupos de Windows Active Directory
- `win_domain_user`, administrar las cuentas de usuario de Windows Active Directory
- `win_dsc`, invocar una configuración de DSC de PowerShell
- `win_environment`, modificar variables de entorno en hosts Windows
- `win_eventlog`, administrar registros de eventos de Windows

160

Trabajando con Módulos



Módulos para Windows

Algunos módulos comunes son (continuación):

- `win_feature`, instalar y desinstalar las características de Windows en Windows Server
- `win_file`, crear, modificar o eliminar archivos o directorios
- `win_firewall`, habilitar o deshabilitar el Firewall de Windows
- `win_firewall_rule`, trabajar con reglas del firewall de Windows
- `win_get_url`, descargar archivos de HTTP, HTTPS o FTP al nodo
- `win_hotfix`, instalar y desinstalar actualizaciones de Windows
- `win_iis_website`, configura un sitio web de IIS
- `win_mapped_drive`, asignar unidades de red para usuarios
- `win_package`, instalar/desinstalar un paquete
- `win_ping`, una versión para Windows del módulo ping clásico
- `win_psexec`, ejecutar comandos (remotamente) como otro usuario (privilegiado)

161

Trabajando con Módulos



Módulos para Windows

Algunos módulos comunes son (continuación):

- `win_psmodule`, agregar o quitar un módulo de Windows PowerShell
- `win_psrepository`, agregar, quitar o actualizar un repositorio de Windows PowerShell
- `win_reboot`, reiniciar una máquina con Windows
- `win_robocopy`, sincroniza el contenido de dos directorios usando Robocopy
- `win_service`, administrar y consultar servicios de Windows
- `win_share`, administrar recursos compartidos de Windows
- `win_shell`, ejecutar comandos de shell en hosts de destino
- `win_stat`, obtener información sobre los archivos de Windows
- `win_unzip`, descomprime archivos comprimidos y archivos en el nodo Windows
- `win_updates`, descargar e instalar actualizaciones de Windows
- `win_wait_for`, espera una condición antes de continuar

162

Trabajando con Módulos

Windows Server

A

```
---
- name: Instalación del servidor web IIS y las herramientas de administración
  hosts: all

  tasks:
    - name: Instalar el servidor web IIS y herramientas de administración
      win_feature:
        name:
          - Web-Server
          - Web-Common-Http
        state: present
        include_management_tools: yes
        register: win_feature

    - name: Establecer el modo de inicio del servicio en automático y asegurarse de que se inicie
      win_service:
        name: WSSVC
        start_mode: auto
        state: started

    - name: Reiniciar si la instalación del servidor web lo requiere
      win_reboot:
        when: win_feature.reboot_required

---
```

---

---

---

---

---

---

---

---

163

Trabajando con Módulos



Módulos para host Windows

La sintaxis y el comportamiento de los módulos para host con sistema operativo Windows es la misma que la de los módulos basados en Linux.  
Estos módulos utilizan normalmente PowerShell y una conexión como WinRM en lugar de SSH.  
[Listado de Módulos para Windows](https://docs.ansible.com/ansible/2.9/modules/list_of_windows_modules.html)

[https://docs.ansible.com/ansible/2.9/modules/list\\_of\\_windows\\_modules.html](https://docs.ansible.com/ansible/2.9/modules/list_of_windows_modules.html)

---

---

---

---

---

---

---

---

164

Tecnofor

A

Ansible Galaxy



---

---

---

---

---

---

---

---

165

Formación del 14/11 al 17/11

54

Ansible Galaxy



¿Qué es Ansible Galaxy?

Galaxy es un repositorio Cloud para buscar y compartir roles de Ansible.

- Acceso a Roles y Colecciones ofrecido por la comunidad de Ansible
- Accesible mediante sitio web y el comando **ansible-galaxy**
- Instalamos los roles, los incorporamos en los *Playbook* y listos para usarlos de inmediato
- Compatible con el nuevo formato de *Colecciones*
- Publicamos nuestros Roles para ayudar a otros usuarios de Ansible

166

---

---

---

---

---

---

---

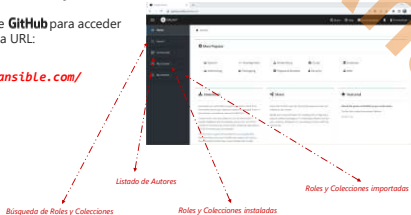
Ansible Galaxy



Sitio web de Ansible Galaxy

Necesitamos una cuenta de **GitHub** para acceder **Ansible Galaxy** mediante la URL:

<https://galaxy.ansible.com/>



167

---

---

---

---

---

---

---

Ansible Galaxy



El comando **ansible-galaxy**

El comando **ansible-galaxy** cuenta con diferentes acciones:

Acciones	Objetivo
<code>role init</code>	Crear la estructura de directorios y archivos de un Rol con los metadatos de Galaxy
<code>role remove</code>	Elimina el Rol del servidor de Ansible
<code>role delete</code>	Elimina el Rol de Ansible Galaxy
<code>role list</code>	Lista los Roles instalados en el servidor de Ansible
<code>role search</code>	Busca Roles en Ansible Galaxy
<code>role import</code>	Importa un Rol en Ansible Galaxy
<code>role info</code>	Muestra información detallada sobre un Rol
<code>role install</code>	Instala un Rol en el servidor de Ansible

168

---

---

---

---

---

---

---

Ansible Galaxy

Buscar Roles con ansible-galaxy

La acción **role search** nos permite buscar Roles en Ansible Galaxy. Con los parámetros **--autor**, **--galaxy-tags** y **--platforms** podemos filtrar los resultados de la búsqueda.

```
ansible:~# ansible-galaxy role search mysql --galaxy-tags database --platforms Ubuntu

Found 278 roles matching your search:

+-----+
| Name                                     | Description                                     |
+-----+-----+
| aaronpederson.mariadb                   | MariaDB - An enhanced, drop-in replacement for MySQL. |
| aboveops.ct_mysql                       | Ansible role deploys MySQL container               |
| achanstier.mysql-backup                 | configure mysql-backup with strbackup and mysqldump  |
| achanstier.mysql-server                 | Install mysql server package                       |
| alexymedvedchikov.galera               | An ansible role for Galera MariaDB cluster           |
| alikins.adminer                         | Install Adminer for Database management.            |
| alikins.mysql                           | MySQL server for RHEL/CentOS and Debian/Ubuntu.     |
| alikins.phpmyadmin                      | phpmyadmin installation for Linux                   |
| alikins.py_mysql                        | PHP MySQL support for Linux.                        |
| almasps.ct_mysql                       | Ansible role deploys MySQL container                 |
+-----+-----+
```

169

Ansible Galaxy

Obtener información de un Rol con ansible-galaxy

La acción **role info** nos permite obtener información de un Rol de Ansible Galaxy.

```
ansible:~# ansible-galaxy role info alikins.mysql

Role: alikins.mysql
description: MySQL server for RHEL/CentOS and Debian/Ubuntu.
active: True
commit: 126f5cdbc3b5f76c3129b0a1b70a3885d244f9
commit_message: 0.99.2
commit_url: https://api.github.com/repos/alikins/top_geerlingguy/git/commits/126f5cdbc3b5f76c3129b0a1b70a3885d244f9
company: Midwestern Mac, LLC
created: 2018-11-20T21:02:27.034063Z
download_count: 2064
forks_count: 0
github_branch: master
github_repo: top_geerlingguy
github_user: alikins
...
```

170

Ansible Galaxy

Instalar un Rol con ansible-galaxy

La acción **role install** permite instalar un Rol de Ansible Galaxy. Los parámetros **--roles-path** o **-p** especificar el directorio para los roles y **-n** o **--no-deps** para no descargar dependencias.

```
ansible:~# ansible-galaxy role install alikins.mysql

Starting galaxy role install process
- downloading role 'mysql', owned by alikins
- downloading role from https://github.com/alikins/top_geerlingguy/archive/0.99.2.tar.gz
- extracting alikins.mysql to /home/administrator/.ansible/roles/alikins.mysql
- alikins.mysql (0.99.2) was installed successfully

ansible:~# ansible-galaxy role install alikins.mysql -p /etc/ansible/role

Starting galaxy role install process
- downloading role 'mysql', owned by alikins
- downloading role from https://github.com/alikins/top_geerlingguy/archive/0.99.2.tar.gz
- extracting alikins.mysql to /etc/ansible/roles/alikins.mysql
- alikins.mysql (0.99.2) was installed successfully
```

171



Ansible Galaxy

A

Listar los Roles con ansible-galaxy

La acción **role list** permite listar los Roles instalados en el servidor de Ansible. Utilizamos el parámetro **--roles-path** o **-p** indicar el directorio que contiene los Roles.

```
ansible:~# ansible-galaxy role list
# /home/administrador/.ansible/roles
- alikins.mysql; 0.99.2
# /etc/ansible/roles
- alikins.mysql; 0.99.2
```

172

---

---

---

---

---

---

---

---

Ansible Galaxy

A

Crear una estructura de un Rol ansible-galaxy

La acción **role init** permite crear la estructura de directorios y archivos de un Rol.

```
ansible:~# ansible-galaxy init apache2
- Role apache2 was created successfully

ansible:~# ls /etc/ansible/roles/
alikins.mysql  apache2

ansible:~# ansible-galaxy role list
# /home/administrador/.ansible/roles
- alikins.mysql; 0.99.2
# /etc/ansible/roles
- apache2; (unknown version)
- alikins.mysql; 0.99.2
```

173

---

---

---

---

---

---

---

---

Ansible Galaxy

A

Fichero de Requerimientos

Podemos instalar varios roles especificados en el archivo **requirements.yml**. El formato de este archivo es **YAML** y la extensión del archivo debe ser **.yml** o **.yaml**.

```
ansible-galaxy install -r requirements.yml
```

- **name**, nombre del Rol para descargar e instalar
- **version**, indica la versión del rol, rama, etiqueta o hash de un commit
- **src**, fuente del Rol, una URL de *Github*, una ruta local o Galaxy que es el valor predeterminado

174

---

---

---

---

---

---

---

---

### Ejemplo de requirements.yml

```
# Descargar desde Galaxy
- src: yatesr.timezone

# Descargar desde Galaxy
- name: timepkg
  src: yatesr.timezone

# Descargar desde Github
- src: https://github.com/bennojoy/nginx

# Descargar desde Github. Indica versión y el nombre para instalarlo
- name: nginx_role
  src: https://github.com/bennojoy/nginx
  version: main
```

Cuando solo indicamos el nombre del rol para descargarlo de Ansible Galaxy podemos utiliza **src** o **name** indistintamente.

175

---

---

---

---

---

---

---

---

### Ansible Galaxy

#### Instalar varios Roles con ansible-galaxy

```
ansible:~# ansible-galaxy install -r requirements.yml
Starting galaxy role install process
- downloading role 'timezone', owned by yatesr
- downloading role from https://github.com/yatesr/ansible-timezone/archive/1.2.0.tar.gz
- extracting yatesr.timezone to /etc/ansible/roles/yatesr.timezone
- yatesr.timezone (1.2.0) was installed successfully
- downloading role 'timepkg', owned by yatesr
- downloading role from https://github.com/yatesr/ansible-timezone/archive/1.2.0.tar.gz
- extracting timepkg to /etc/ansible/roles/timepkg
- timepkg (1.2.0) was installed successfully
- extracting nginx to /etc/ansible/roles/nginx
- nginx was installed successfully
```

176

---

---

---

---

---

---

---

---

### LABS

Seguindo las indicaciones del instructor compartimos un Rol en Ansible Galaxy.

- ✓ Creamos un repositorio en Github
- ✓ Inicializamos el Rol utilizando el comando:  
`ansible-galaxy role init apache2`
- ✓ Definir el Rol y finalmente editar los archivos:
  - README.md
  - /meta/main.yml



177

---

---

---

---

---

---

---

---

LABS

Siguiendo las indicaciones del instructor compartimos un Rol en Ansible Galaxy.

✓

Inicializar el directorio del Rol para el uso de Git:  
`git init`

✓

Realizar el primer `commit` de Git:  
`git add .`  
`git commit -m "Primer Commit"`

✓

Registramos el repositorio remoto y hacemos un `push`:  
`git remote add origin https://github.com/Formaciones/apache2.git`  
`git push -u origin main`

LAB 1

178

---

---

---

---

---

---

---

---

LABS

Siguiendo las indicaciones del instructor compartimos un Rol en Ansible Galaxy.

✓

Copiamos la API-Key de Ansible Galaxy ofrecida en el portal

✓

Importamos el Rol utilizando el siguiente comando:  
`ansible-galaxy role import <User-Github> <Repository-Github> --token <API-Key>`  
`ansible-galaxy role import Formaciones apache2 --token c78dee02fee7ca26ab6...`

✓

Buscar el Rol para comprobar que fue importado:  
`ansible-galaxy search apache2 --galaxy-tags pruebas`

LAB 1

179

---

---

---

---

---

---

---

---

Tecnofor

Ansible Tower

LAB 1

180

---

---

---

---

---

---

---

---

Ansible Tower



¿Qué es Ansible Tower?

Ansible Tower es una solución web pensada como centro de automatización para facilitar el trabajo a los equipos de IT.

Algunas de sus características son:

- administrar y compartir credenciales
- gestionar inventarios gráficamente y sincronizarlos con diversas fuentes
- registrar y administrar todos sus trabajos
- crear flujos de trabajo CI/CD
- integración mediante API REST y herramientas de línea de comandos

La suscripción Ansible Tower Free para hasta 10 nodos ya NO se puede adquirir.

181

Ansible Tower



Instalación de Ansible Tower

Requisitos para la instalación de Ansible Tower:

- Servidor dedicado RedHat Enterprise Linux, CentOS o Ubuntu
- RAM 2GB (recomendado 4GB)
- HD 20GB
- Habilitar repositorio EPEL en RedHat y CentOS
- Puertos 80,443 y 8080



Existe la posibilidad de instalaciones con varios nodos en alta disponibilidad.

182

LABS

Siguiendo las indicaciones del instructor instalamos Ansible Tower.

✓

Descargamos el paquete del repositorio de ansible:  
`wget https://releases.ansible.com/ansible-tower/setup/ansible-tower-setup-latest.tar.gz`

✓

Descomprimos el archivo descargado:  
`tar xvfz ansible-tower-setup-latest.tar.gz`  
`cd ansible-tower-setup-X.X.X-X`

✓

Editamos el fichero `inventory` para indicar servidores, base de datos y contraseñas.  
Finalizada la edición del fichero `inventory` lanzamos el setup:  
  
`nano inventory`  
`./setup.sh`

LAB 1

183

LABS

Siguiendo las indicaciones del instructor instalamos Ansible Tower.

✓ Finalizada la instalación aparece el mensaje:  
*The setup process completed successfully.*

✓ Mediante la dirección IP accedemos Ansible Tower:

- Introduce usuario (*admin*) y contraseña
- Solicitar o registrar una suscripción
- Comenzar a trabajar



---

---

---

---

---

---

---

---

184

Ansible Tower



Configuración de Ansible Tower

Finalizada la instalación debemos configurar **Ansible Tower** definiendo:

- **Organización**, ámbito de seguridad que agrupa usuarios, permisos y los elementos de trabajo
- **Usuarios**, puede acceder y utiliza y/o crear elementos de trabajo
- **Equipos**, grupos de usuarios contenidos de una organización
- **Credenciales**, contraseñas, certificados ssh, token, ...
- **Trabajos de Gestión**, trabajos de mantenimiento de Ansible Tower, por ejemplo, limpiar el historial de los Trabajos, limpiar el log de actividad
- **Script de Inventarios**, creados y cargados para inventarios dinámicos a medida
- **Suscripción**, añadir, cambiar y eliminar la suscripción
- **Notificaciones**, definir plantillas para el enviar notificaciones por Mail, SMS, Slack, ...

---

---

---

---

---

---

---

---

185

Ansible Tower



Elementos de Trabajo en Ansible Tower

**Ansible Tower** cuenta con los siguientes elementos:

- **Proyectos**, define donde se almacenan los *Playbooks* (Github, Mercurial, Subversion o Local)
- **Inventarios**, lista de host administrados (estáticos o dinámicos)
- **Plantillas**, determina los *Playbooks* que se ejecutan en cada host remoto
- **Credenciales**, contraseñas, certificados ssh, token, ... para acceder a *Proyectos*, *Inventarios* y host remotos
- **Trabajos**, son la ejecución de las *Plantillas* donde se recopila las salida de todos los *Playbooks*
- **Plantillas de flujo de trabajo**, solo en las versiones más reciente, permite enlazar la ejecución de unas *Plantillas* con otras en función de como finalizan los *Trabajos*.
- **Cuestionarios**, formularios para ingresar datos al comienzo de los *Trabajos*

---

---

---

---

---

---

---

---

186

Formación del 14/11 al 17/11

61

Ansible Tower

Proyectos

Los proyectos locales se almacena por defecto en `/var/lib/awx/projects`, los proyectos en Github u otros repositorios al actualizar se crea una cache local.

```
[root@localhost ~]# cd /var/lib/awx
[root@localhost awx]# ls
favicon.ico job_status projects public rsyslog uwsgi.stats vendor venv wsgi.py

[root@localhost awx]# cd projects
[root@localhost projects]# ls
_6_demo_project _6_demo_project.lock
[root@localhost projects]# ls _6_demo_project
hello_world.yml README.md
```

187

---

---

---

---

---

---

---

---

LABS

Creación de un nuevo Proyecto en Ansible Tower.

✓ Creamos una nueva carpeta dentro de *projects*.

```
cd /var/lib/awx/projects
mkdir democurso
cd democurso
copiamos o creamos un Playbook
chown awx:awx -R
```

✓ En Ansible Tower añadimos un nuevo Proyecto:

- Seleccionamos Manual para *Scm Type*
- Dejamos el valor por defecto de *Project Base Path*
- Seleccionamos democurso como *Playbook Directory*



188

---

---

---

---

---

---

---


---

Ansible Tower

Credenciales

Tower utiliza las credenciales para autenticarse al iniciar trabajos en máquinas, sincronizar con fuentes de inventario e importar contenido de proyectos desde un sistema de control de versiones.

Puede otorgar a los usuarios y equipos la capacidad de usar estas credenciales, sin exponer realmente la credencial al usuario.



189

---

---

---

---

---

---

---

---

Ansible Tower



Inventarios

Un inventario es una colección de hosts remotos en los que se pueden iniciar trabajos, al igual que un archivo de inventario de Ansible.

Los inventarios se dividen en grupos y estos grupos contienen los hosts reales.

Los grupos se pueden obtener manualmente, ingresando nombres de host en Tower, o de uno de los proveedores de nube compatibles con Ansible Tower.



190

Ansible Tower



Job Templates

Una plantilla es una definición para ejecutar un trabajo de Ansible.

Las plantillas permiten ejecutar el mismo trabajo muchas veces, fomentando la reutilización de los Playbooks y la colaboración entre equipos.

Para crear una nueva plantilla de trabajo registramos al menos siguientes campos:

- **Nombre** para el trabajo.
- **Descripción** según corresponda (opcional).
- **Tipo de trabajo**, ejecutar o comprobar
- **Inventario** que se usará con esta plantilla de los disponibles para el usuario
- **Proyecto** que se usará con esta plantilla de los disponibles para el usuario

191

Ansible Tower



Job Templates

Para crear una nueva plantilla de trabajo registramos al menos siguientes campos (continuación):

- **Playbooks** que se iniciará con esta plantilla
- **Credenciales**, tras una búsqueda elegimos entre las disponibles para usar con esta plantilla
- **Bifurcaciones** que son la cantidad de procesos paralelos o simultáneos que se usarán
- **Límite**, un patrón de host para restringir aún más la lista de hosts administrados
- **Verbosidad** para controlar el nivel de salida que produce Ansible
- **Etiquetas de trabajo** para especificar qué partes de los Playbook que deben ejecutarse
- **Omitir etiquetas** para omitir ciertas tareas o partes de los Playbooks
- **Etiquetas** para describir la plantilla de trabajo, como "dev" o "test"

192

Ansible Tower



Guía del Usuario de Ansible Tower

La Guía del usuario de Ansible Tower analiza todas las funciones disponibles en Ansible Tower y asume una familiaridad moderada con Ansible, incluidos conceptos como playbooks, variables y etiquetas.

[Guía del Usuario de Ansible Tower](https://docs.ansible.com/ansible-tower/latest/html/userguide/index.html)

<https://docs.ansible.com/ansible-tower/latest/html/userguide/index.html>

193

---

---

---

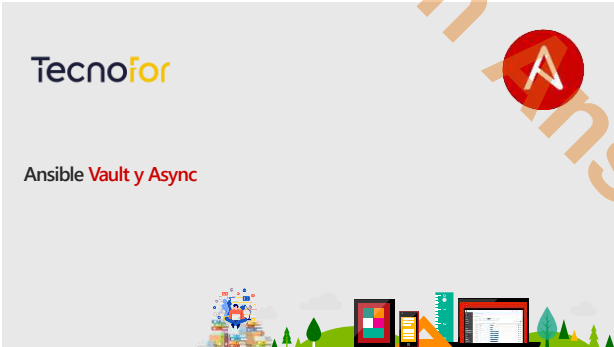
---

---

---

---

---



194

---

---

---

---

---

---

---

---

Ansible Vault



¿Qué es Ansible Vault?

**Ansible Vault** cifra variables y archivos para proteger el contenido confidencial, como contraseñas o claves, en *Playbooks* o roles.

Para usar **Ansible Vault** necesita una o más contraseñas para cifrar y descifrar contenido.

Si almacenamos las contraseñas en una boveda de terceros, como *Azure Key Vault*, necesitamos un script para acceder a las contraseñas.

En el archivo de configuración **ansible.cfg** podemos especificar la ubicación de un archivo de contraseña o indicar que siempre se solicite contraseña para trabajar con variables y archivos cifrados.

195

---

---

---

---

---

---

---

---



Ansible Vault



Comando ansible-vault

```
usage: ansible-vault [-h] [--version] [-v] {create,decrypt,edit,view,encrypt,encrypt_string,rekey}
...

encryption/decryption utility for Ansible data files

positional arguments:
  {create,decrypt,edit,view,encrypt,encrypt_string,rekey}
    create              Create new vault encrypted file
    decrypt             Decrypt vault encrypted file
    edit               Edit vault encrypted file
    view              View vault encrypted file
    encrypt            Encrypt YAML file
    encrypt_string      Encrypt a string
    rekey              Re-key a vault encrypted file

See 'ansible-vault <command> --help' for more information on a specific command.
```

196

---

---

---

---

---

---

---

---

Ansible Vault



Trabajando con ansible-vault

Utilizamos el comando **ansible-vault create** para crear un archivo donde el contenido estar  protegido mediante una contrase a.

```
ansible:~# ansible-vault create secret.yml
New Vault password:
Confirm New Vault password:

ansible:~# cat secret.yml
$ANSIBLE_VAULT:1.1;AES256
61309563636561646661376237351833313733183762383235343430646439333732623330343833
21376438343634373933663532313566663139383956430369a383931643364326164366133203665
643538613463656132313831346663630383537616535643239633130326533306564623430326266
3430336464393832356a393336303431363332333965326337346266353730363162616139373638
6261
```

197

---

---

---

---

---

---

---

---

Ansible Vault



Trabajando con ansible-vault

Para editar un fichero utilizamos el comando **ansible-vault edit**. Necesitamos suministrar la contrase a indicada en el momento de la creaci n.

```
ansible:~# ansible-vault edit secret.yml
Vault password:
ERROR! Decryption failed (no vault secrets were found that could decrypt) for /etc/ansible/secret.yml

ansible:~# ansible-vault edit secret.yml
Vault password:
```

198

---

---

---

---

---

---

---

---

Ansible Vault

Trabajando con ansible-vault

Para proteger un archivo que ya existe utilizamos el comando `ansible-vault encrypt`.

```
ansible:~# cat file-secret.yml
# Fichero Demos

ansible:~# ansible-vault encrypt file-secret.yml
New Vault password:
Confirm New Vault password:
Encryption successful

ansible:~# cat file-secret.yml
$ANSIBLE_VAULT;1.1;AES256
31363531326364396662313734393839393335633263663061373433393436343137353061
38336535346662313734393839393335633263663061373433393436343137353061
32323766646663386662313734393839393335633263663061373433393436343137353061
34383562653831326364396662313734393839393335633263663061373433393436343137353061
38313339383339633938323166356438333263663533666162383838356638396332
```

199

---

---

---

---

---

---

---

---

Ansible Vault

Trabajando con ansible-vault

`ansible-vault view` nos permite ver el contenido de un archivo después de indicar la contraseña con la cual fue cifrado y protegido.

```
ansible:~# ansible-vault view file-secret.yml
Vault password:
# Fichero Demos
```

200

---

---

---

---

---

---

---

---

Ansible Vault

Trabajando con ansible-vault

`ansible-vault decrypt` desprotege el archivo después de indicar la contraseña con la cual fue cifrado y protegido.

```
ansible:~# cat file-secret.yml
$ANSIBLE_VAULT;1.1;AES256
31363531326364396662313734393839393335633263663061373433393436343137353061
38336535346662313734393839393335633263663061373433393436343137353061
32323766646663386662313734393839393335633263663061373433393436343137353061
34383562653831326364396662313734393839393335633263663061373433393436343137353061
38313339383339633938323166356438333263663533666162383838356638396332

ansible:~# ansible-vault decrypt file-secret.yml
Vault password:
Decryption successful

ansible:~# cat file-secret.yml
# Fichero Demos
```

201

---

---

---

---

---

---

---

---

Ansible Vault



Trabajando con Ansible Vault

Cuando todas las variables y archivos cifrados utilizan una sola contraseña, podemos utilizar los parámetros `--ask-vault-pass` o `--vault-password-file`:

- Para solicitar la contraseña:  
`ansible-playbook --ask-vault-pass playbook.yml`
- Para recuperar la contraseña del archivo:  
`ansible-playbook --vault-password-file /path/vault-password-file.yml`
- Para obtener la contraseña del script de cliente de contraseña del almacén:  
`ansible-playbook --vault-password-file my-vault-password-client.py`

---

---

---

---

---

---

---

---

202

Ansible Vault



Trabajando con ansible-vault

Para poder ejecutar un Playbook cifrado debemos indicar la contraseña mediante los parámetros `--ask-vault-pass` o `--vault-password-file`.

```
ansible:~# ansible-playbook demo-vault.yml
ERROR! Attempting to decrypt but no vault secrets found

ansible:~# ansible-playbook demo-vault.yml --ask-vault-pass
Vault password:
PLAY [Playbook para demostración de uso de Ansible Vault] *****
TASK [Gathering Facts] *****
ok: [localhost]
TASK [Módulo Ping] *****
ok: [localhost]
PLAY RECAP *****
localhost : ok=3  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

---

---

---

---

---

---

---

---

203

Ansible Vault



Trabajando con ansible-vault

Podemos cifrar únicamente el contenido de las variables con `ansible-vault encrypt_string` indicando el contenido y el nombre de la variable.

```
ansible:~# ansible-vault encrypt_string 'Contenido cifrado' --name 'var1'
New Vault password:
Confirm New Vault password:
var1: lvault1
$ANSIBLE_VAULT:1.1,AES256
3264616233368616362313431313232643036313235386261626231623163343633643161393030
63323731326331303133613766646565363937343363663080303461643263663533646637303630
3231606532633034303435383931313334653136306334643561323564623039373636393630303231
63363437313430303030303030303030303030303030303030303030303030303030303030303030
65386634353234613163363866373037623132313433323337666535373565656364
Encryption successful
```

---

---

---

---

---

---

---

---

204

## Playbook con Variables Cifradas

```
- name: Playbook para demostración de uso de Ansible Vault
host: all

# El contenido de la variable fue cifrado con la contraseña: Pa$w0rd
vars:
  - var1: !vault |
    $ANSIBLE_VAULT;1.1;AES256
3d66821c3213683683227148411312120418018112121351802561A2621212136134136134118121091010
61321711505121301366137666645055033937413436368081344136126366135136466371031030
3121316651261381384138391111813465133361363136413562132564621303913771361391361381331
613613713142013813986132163138613613413661351213646136413513536613851386132613161347377
65886634151234613161333613866737037621132121133123133137666513713565658364
```

ERROR:

```
- name: Módulo Debug, muestra el contenido de la variable cifrada var1
debug:
  var1: var1
```

Puede ejecutar un Playbook con variables protegidas suministrando la contraseña utilizada para el cifrar el contenido.

© 2019 Red Hat, Inc. All rights reserved.

205

## Tareas Asíncronas

Ansible ejecuta tareas de forma sincrónica, manteniendo abierta la conexión al nodo remoto hasta que se complete la acción.

En los *Playbooks*, cada tarea bloquea la siguiente tarea de forma predeterminada, lo que significa que las tareas posteriores no se ejecutarán hasta que se complete la tarea actual.

Con el modo `asynchronous`, es posible que un proceso de larga duración se ejecute en segundo plano mientras realiza otras tareas simultáneamente.

Para ejecutar varias tareas, usamos `poll=0`. Cada tarea asíncrona se ejecuta hasta que se completa, se produce un error o se agota el tiempo de espera especificado con `async`.

206



## Playbook con Tareas Async

```

- name: Playbook para demostración de uso de Ansible Async
  hosts: all

tasks:
- name: Módulo Command
  command: 'sleep 15'
  async: 60
  poll: 0
  register: estado_async

- name: Módulo Ping
  ping: ''

- name: Módulo Debug
  debug:
    msg: 'Otra tarea'

- name: Consultar el estado
  async_status:
    jid: "{{ 'jid_estado_async.ansible_job_id' }}"
  register: estado
  until: estado.finished
  retries: 30

```

El tiempo máximo para finalizar la tareas se especifica mediante **async**

Cuando **poll=0** la ejecución continua en la siguiente tarea.

Se puede consultar el estado de una tarea mediante el módulo **async\_status**.

demmo-ansible@poc

207

Gracias

Las cosas simples son las más extraordinarias y sólo los sabios consiguen verlas.

Paulo Coelho

208

---

---

---

---

---

---

---

MERCI   TACK   മൗനമുഖമർദ്ദി   KIITOS   MULTUMESC   DANKE

SALMAT PO   GRACIAS   감사합니다   شكراً   DZIEKUJE

OBRIGADO   MAHALO   TEŞEKKÜRLER   ευχαριστώ   धन्यवाद

СПАСИБО   Thank you !!!   நன்றி   多謝晒

הודות   PALDIES   ДЯКУЮ   DAKUJEM   منشكرم   TERIMA KASIH   TAK

HVALA   شكریه   DEKUJI   谢谢   CAM ON   DANKON

GRAZIE   KOSZONOM   БЛАГОДАРЯ   ありがとうございます

209

---

---

---

---

---

---

---

Break

Time coffee

210

---

---

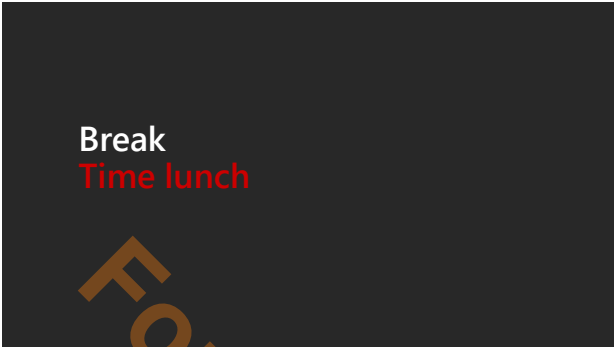
---

---

---

---

---



211

---

---

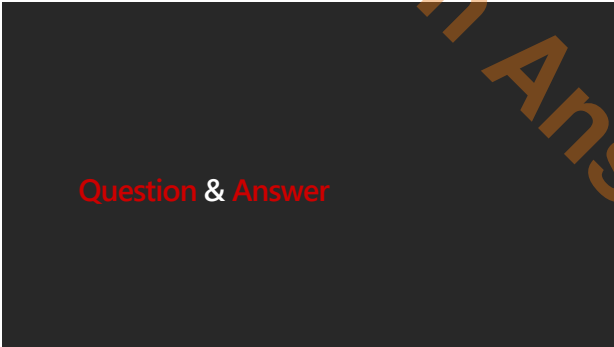
---

---

---

---

---



212

---

---

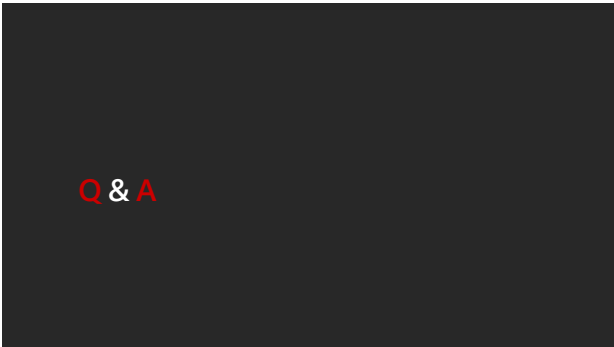
---

---

---

---

---



213

---

---

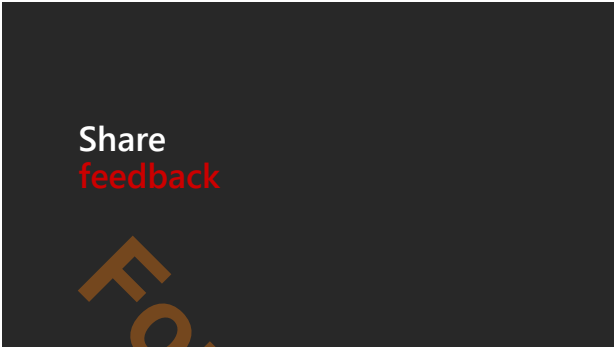
---

---

---

---

---



214

---

---

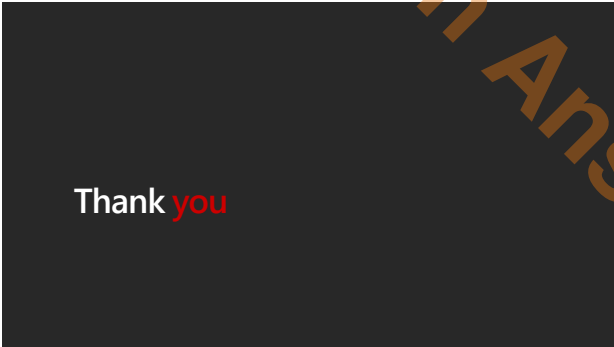
---

---

---

---

---



215

---

---

---

---

---

---

---