

Ejecutar comando con otro usuario

Ponemos la contraseña de sudo en el fichero de hosts:

```
/etc/ansible/hosts
```

```
[jose]
172.17.0.1 ansible_connection=ssh ansible_ssh_user=jose
ansible_ssh_pass=**** ansible_sudo_pass=****
```

Ejecutamos el siguiente playbook:

```
- name: Ejecutar comando con otro usuario
  hosts: jose

  tasks:
    - name: Ejecutar sudo ls /root
      shell: ls /root
      become: true
      become_user: root
      ignore_errors: yes
      register: uname_result
      # En una misma línea
      - debug: msg="{{ uname_result.stdout }}"

      # En líneas separadas
      - debug: msg="{{ uname_result.stdout_lines }}"
```

Ejecutamos:

```
ansible-playbook sudo.yml
```

Bucles

Por defecto la variable del bucle es item

```
tasks:
- name: list variables
  debug:
    msg="reading from {{item.soft}} and {{item.directori}}"
  with_items:
    - { soft: 'apache.zip', directori: '/opt/pogramas/webserver' }
    - { soft: 'java7.zip', directori: '/opt/pogramas/java' }
```

Cambiando la variable del bucle

```
tasks:
- name: list variables
  debug:
    msg="reading from {{mi_variable.soft}} and {{mi_variable.directori}}"
  with_items:
    - { soft: 'apache.zip', directori: '/opt/pogramas/webserver' }
    - { soft: 'java7.zip', directori: '/opt/pogramas/java' }
  loop_control:
    loop_var: mi_variable
```

Variables para cada servidor

Dentro de vars/<fichero>.yml El nombre del fichero da igual, pero mejor poner algo representativo

```
---
vars:
  hosts:
    server1:
      variable_comun: valor1
    server2:
      variable_comun: valor2
```

Precedencia variables

Este es el orden, de mayor precedencia a menor precedencia, de las variables:

1. extra vars

```
--extra-vars "myvar=extra-vars"
```

2. playbook

```
install-mysql.yml
```

3. role

```
role/mysql/vars/main.yml
```

4. groups

```
group_vars/database.yml
```

5. role defaults

```
role/mysql/defaults/main.yml
```

TODO: poner un ejemplo complejo para que se vean las buenas prácticas de hacer roles reusables

Almacenar información sensible (vaults)

Crear archivo con y sin cifrar

1. Crear un archivo con las variables, y llamarlo "vars.yml" con el siguiente contenido:

```
var1 = "pepe"
```

2. Copiar el archivo y llamarlo "vault.yml":

```
cp vars.yml vault.yml
```

3. Editar "vault.yml" y añadir (es pura convención) "vault_" delante de cada variable:

```
vault_var1 = "pepe"
```

4. Cifrar "vault.yml", le añadiremos la etiqueta (es pura convención) "prod":

```
ansible-vault encrypt vault.yml --vault-id prod@prompt
```

5. Teclear dos veces la contraseña (y anotarla en algún lugar)

6. Editar "vars.yml" para que cada variable apunte a su correspondiente "vault_":

```
vim vault.yml
```

Y dejarlo tal que así:

```
var1 = "{{ vault_var1 }}"
```

7. Incluir los archivos "vars.yml" y "vault.yml". En este ejemplo lo hacemos en las tasks:

```
- name: Include vars.yml
  include_vars: vars.yml

- name: Include vault.yml
  include_vars: vault.yml
```

Cifrar un archivo

En este ejemplo:

* Ciframos un archivo entero * El módulo de ansible se encarga de descifrarlo, descomprimirlo y

dejarlo en destino

Tenemos el siguiente diseño de archivos:

```
.
├── hosts
├── roles
│   └── example
│       ├── files
│       │   └── files.tar.bz2
│       ├── tasks
│       └── main.yml
```

1. Cifrar el archivo, en este caso “files.tar.bz2”

```
ansible-vault encrypt roles/example/files/files.tar.bz2 --vault-id
prod@prompt
```

2. Escribir la contraseña

3. Añadir una tarea:

```
- name: Uncompress the wordrepss files in docker host volume
  unarchive:
    src: files.tar.bz2
    dest: /tmp
    become: true
```

Nótese que dado que usamos roles no hay que especificar el path absoluto de “files.tar.bz2”.

4. Ejecutar el playbook.

El resultado esperado es que descifrará “files.tar.bz2” y lo dejará en “/tmp” en el servidor remoto

Comandos útiles

* Editar archivo encriptado

```
ansible-vault edit vault.yml --ask-vault-pass
```

* Editar y proporcionar archivo (con la contraseña en plano) para no tener que teclearla cada vez

```
ansible-vault edit vault.yml --vault-id prod@examples/password
```

* Editar y proporcionar archivo (con la contraseña en plano) para no tener que teclearla cada vez

```
ansible-vault view vault.yml --vault-id prod@examples/password
```

From:

<http://wiki.legido.com/> - **Legido Wiki**

Permanent link:

<http://wiki.legido.com/doku.php?id=informatica:linux:ansible>



Last update: **2020/01/21 08:00**