ANSIBLE

FACTS & SECRETS

➢ ANSIBLE FACTS:

- Ansible facts are variables that are automatically discovered by Ansible from a **managed host.**
- Variables related to **remote systems** are called **facts.**
- Facts related to remote systems, including OS's, IP add, kernel version, network interfaces, filesystems etc.
- To disable this behavior using the **INJECT_FACTS_VARS** setting.

```
$ansible agent1 -m setup   [ To list facts from the agent1 ]

$ansible agent1 -m setup | grep hostname

$ansible agent1 -m setup -a "filter=*ipv4"

$ansible agent1 -m setup -a "filter=hostname"

$ansible agent1 -m setup -a "filter=ansible_fqdn"

$ansible agent1 -m setup -a "filter=ansible_virtualization_type"
```

$vi facts.yml

```
- hosts: webservers
  become: true
  become_user: root
  tasks:
    - name: Print Various Ansible Facts
      debug:
        msg: >
              The default address of {{ ansible_fqdn }} is {{ ansible_default_ipv4.address }}
```

```
$ansible-playbook --syntax-check facts.yml

$ansible-playbook facts.yml -K
```

➢ MANAGING SECRETS:

- Ansible vault provides a way to encrypt and manage sensitive data such as passwords.
- Vaults can encrypt and decrypt arbitrary variables and files, which means you can use it to protect variable files that contain secrets or even encrypt entire sensitive configuration files.

NOTE:

- Ansible vault does not implements its own cryptographic functions but rather uses an external python toolkit. Files are protected with symmetric encryption using AES256 with a password as the secret key.

  COMMAND: ansible-vault

- **Create**	**:** To create ansible vault file in the encrypted format
- **View**	**:** To view data of encrypted file
- **Edit**	**:** To edit encrypted file
- **Encrypt**	**:** To encrypt an unencrypted file
- **Decrypt**	**:** To decrypt an encrypted file

Creating new Playbook with vault:

```
$ansible-vault create myplay.yml
password:
confirm password:
---
    -   hosts: webservers
become: true
become_user: root
...

$cat myplay.yml
$ansible-vault view myplay.yml
```

Encrypt existing playbook:

```
$ansible-vault encrypt sample.yml
$cat sample.yml
$ansible-vault view sample.yml
```

Encrypt existing file name to new file name:

```
$ansible-vault encrypt test.yml --output=newtest.yml
password:
$cat test.yml
$ansible-vault view newtest.yml
```

Take password reference:

```
$cat>password
raju123
$ansible-vault create --vault-password-file=password newplay.yml
---
    -   hosts: webservers
become: true
become_user: root
...
$ansible-vault view newplay.yml
```

To change existing encrypted password:

```
$ansible-vault rekey sample.yml
$ansible-vault view sample.yml
```

Decrypt the password:

```
$ansible-vault decrypt newplay.yml
$cat newplay.yml
```

TASK1: Create A User Account

```
$ansible-vault create useraccount.yml
---
  -  hosts: webservers
become: true
become_user: root
tasks:
  -  name: User account Creation
     user:
       name: testuser
       password: raju123
       comment: Ansible Test User
       state: present
...

$ansible-playbook --vault-id @prompt useraccount.yml -K
```

With Encrypted password:

```
---
-  hosts: webservers

   become: true

   become_user: root

   tasks:

     -  name: User account Creation
        user:
        name: testuser1
        password: $6$8UBDLg2Al0b73yzj$IezSQWNLEmky
        comment: Ansible Test User
        state: present
        ...


$ansible-playbook --vault-id @prompt useraccount.yml -K
```

TASK2: Variables with encrypted playbook (full encrypted playbook)

Create a variable encrypted file:

```
$vi secret.yml
  username: max
  upass: $6$zJWQFRWezy2ZRixq$nwHPdI59UKudBpMx7ai
$ansible-vault encrypt secret.yml
```

Now create a playbook for user:

```
$vi user.yml
- hosts: webservers
  become: true
  become_user: root
  vars_files:
    - secret.yml
  tasks:
    - name: User Creation
      user:
        name: "{{ username }}"
        password: "{{ upass }}"
        comment: Ansible Test User
        state: present
```

```
$ansible-playbook --syntax-check --vault-id @prompt user.yml
```

Encrypt existing playbook:

$ansible-vault encrypt user.yml [Make sure secret & user.yml files password should be same]

$ansible-playbook --vault-id @prompt user.yml -K [Execute playbook]