# Ansible Vault

*By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years*

**Session - 12 Agenda:**

1. Ansible Vault.

***Ansible Vault*** is a feature of ansible that allows you to keep sensitive data such as passwords or keys in encrypted files, rather than as plaintext in playbooks or roles. These vault files can then be distributed or placed in source control.

Let's create a normal Ansible playbook to create the user account and set the password on all the managed nodes.

**$ mkdir playbooks**
**$ cd playbooks**
**$ vim vault.yml**

```
---
- name: Ansible vault playbook
  hosts: all
  become: true
  vars:
    pw: redhat
  tasks:
      - name: creating user account and setting the password.
        user:
          name: vikas
          password: "{{ pw }}"
...
```

Now check the playbook for syntax errors.

**$ ansible-playbook vault.yml --syntax-check**

Problem here is anyone having the access to the ansible playbook file and read the contents of this file and as well as make the changes in it.

**$ cat vault.yml**

So, in order to protect the contents or the sensitive information we can encrypt the existing playbook file using ansible-vault encrypt command.

**$ ansible-vault encrypt vault.yml**

Now the playbook file has been encrypted and no-one we will able to read or make changes in this file except the owner.

**$ cat vault.yml**

Have a look at the file contents which are showing up here after executing the cat

command.

**Output=>**

**$ANSIBLE_VAULT;1.1;AES256**

**35393664383830363030356239323037383836393935623961643936376637 3**

Here Ansible vault version is 1.1 which is using AES256 algorithm for the encryption.

**AES 256?**

**Advanced Encryption Standard (AES) 256** is a virtually impenetrable symmetric encryption algorithm that uses a 256-bit key to convert your plain text or data into a cipher.

We can't even execute the encrypted playbook file directly without encryption password.

**$ ansible-playbook vault.yml**

We can't even make the changes in the encrypted playbook file directly without encryption password.

**$ vim vault.yml**

In order to make changes in the encrypted file either we will have to decrypt it or use ansible-vault edit command with the encryption password.

**$ ansible-vault edit vault.yml**

Similarly, we can read the file contents by using ansible-vault view command with the encryption password.

**$ ansible-vault view vault.yml**

To execute (push) the encrypted playbook we can execute below command which will ask for the encryption password.

**$ ansible-playbook vault.yml --ask-vault-pass**

Verify the task execution using ansible Ah-hoc command.

**$ ansible all -m command -a 'tail /etc/passwd'**

To decrypt the playbook file, we can execute the ansible-vault decrypt command which will ask for the encryption password.

**$ ansible-vault decrypt vault.yml**

Now, we can easily read the file contents.

**$ cat vault.yml**

In case if we want to encrypt only the sensitive data in the playbook file not the

entire playbook, we can put it in a separate file and encrypt the same. Let's create the playbook first where we will call the variables from another encrypted file.

**$ vim vault2.yml**

```
---
- name: Ansible vault playbook
  hosts: all
  become: true
  vars_files:
    - /home/vikasnehra/crypt.yml
  tasks:
      - name: creating user account and setting the password.
        user:
          name: greta
          password: "{{ pw }}"
...
```

Let's create a separate file for the variables used in the above playbook. We can directly create an encrypted file using ansible-vault create command.

**$ ansible-vault create /home/vikasnehra/crypt.yml**

**pw: redhat**

Since the file is encrypted, no-one can read the contents of this file now.

**$ cat /home/vikasnehra/crypt.yml**

Now check the playbook for syntax errors.

**$ ansible-playbook vault2.yml --syntax-check**

If you will try to execute the playbook without vault password it will throw an error.

**$ ansible-playbook vault2.yml**

To execute (push) the encrypted playbook we can execute below command which will ask for the encryption password.

**$ ansible-playbook vault2.yml --ask-vault-pass**

Verify the task execution using ansible Ah-hoc command.

**$ ansible all -m command -a 'tail /etc/passwd'**

To change the vault password of any file we can use ansible-vault rekey command.

**$ ansible-vault rekey /home/vikasnehra/crypt.yml**

# Ansible Vault

*By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years*

Let's create another encrypted playbook to check the connectivity with the managed nodes.

**$ ansible-vault create ping_vault.yml**

```
---
- name: Ping Test Playbook
  hosts: all
  tasks:
    - ping:
...
```

Encrypted playbook has been created and we can't read it without using the vault password.

**$ cat ping_vault.yml**

We can only view the contents if we know the vault password set by the owner.

**$ ansible-vault view ping_vault.yml**

We can only edit the contents if we know the vault password set by the owner.

**$ ansible-vault edit ping_vault.yml**

We can only change the vault password set by the owner if we know the existing vault password.

**$ ansible-vault rekey ping_vault.yml**

We can only decrypt the encrypted file if we know the vault password set by the owner.

**$ ansible-vault decrypt ping_vault.yml**

To encrypt the existing file (playbook file again) we can execute below command.

**$ ansible-vault encrypt ping_vault.yml**

To execute the encrypted playbook, we have to use the vault password set by the owner.

**$ ansible-playbook ping_vault.yml --ask-vault-pass**

We can also put the vault password in a file and call the file whenever we require the vault password. Let's create a file having the vault password in it (ideally, we should create such files in a secured location in the system).

**$ vim pass.txt**

**redhat**

Let's verify the file contents.

**$ cat pass.txt**

# Ansible Vault

*By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years*

To execute the encrypted playbook, we have to call the file having the vault password.

**$ ansible-playbook ping_vault.yml --vault-password-file pass.txt**

We can also encrypt any string in the file instead of the entire playbook to protect the sensitive information.

**$ ansible-vault encrypt_string 'string' --name 'ping'**

The ansible-vault encrypt_string command creates an encrypted string. The encrypted string is stored as a key value pair. In this example, a key named "foo" is created, and value "bar" is encrypted.

**$ ansible-vault encrypt_string 'bar' --name 'foo'**

Almost always, you are going to redirect the encrypted string to a file.

**$ ansible-vault encrypt_string 'bar' --name 'foo' > foo.txt**

**$ cat foo.txt**

---

***Thanks***