**Lesson 9 Demo 1**

**Managing Sensitive Files with Ansible Vault**

**Objective:** To manage sensitive files using Ansible Vault

**Prerequisites:** You need to have Ansible installed and setup to proceed with this demo. If you don’t have it installed, refer to demo 1 of lesson 2.

**Tools required:** Ubuntu terminal and Ansible

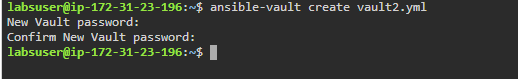
**Steps to be performed:**

1. Creating newly encrypted files
2. Encrypting existing files
3. Viewing encrypted files
4. Editing encrypted files
5. Decrypting encrypted files manually
6. Changing the password of encrypted files
7. Running Ansible with vault-encrypted files

**Step 1: Creating New Encrypted Files**

1. To create a new file encrypted with Vault, use the following command where *vault1.yml* is the name of the encrypted file that you use to store the sensitive variables

***ansible-vault create vault2.yml***

1. You will be prompted to enter and confirm a pass
2. Then, Ansible will open an editing window where you can enter your desired contents. To test the encryption function, enter some test text:

**Confidential Information!!!**

Text

Description automatically generated

1. Ansible will encrypt the contents when you close the file. If you check the file using the command ***cat vault2.yml*** instead of seeing the words you typed, you will see an encrypted block:

Text

Description automatically generated

**Step 2: Encrypting Existing Files**

1. You can encrypt an existing file using the ansible-vault encrypt command. Suppose we create an example file for testing using the command

***echo 'unencrypted stuff' > encrypt\_me.txt***

1. You can encrypt the existing file using the command

***ansible-vault encrypt encrypt\_me.txt***

1. Again, you will be prompted to provide and confirm a password. Afterward, a message will confirm the encryption

Text

Description automatically generated

1. You can see the following output on running the ***cat encrypt\_me.txt*** command

Text

Description automatically generated

**Step 3: Viewing Encrypted Files**

1. Pass the vault encrypted file to the ansible-vault view command:

***ansible-vault view vault2.yml***

1. You will be asked for the file’s password. After entering it successfully, the contents will be displayed:

Text

Description automatically generated

**Step 4: Editing Encrypted Files**

1. Use the below command to edit encrypted files:

***ansible-vault edit vault2.yml***



1. You will be prompted to enter a password, after which you can successfully edit the file

Shape

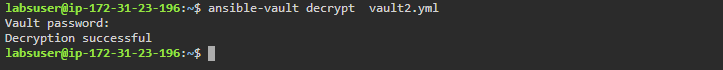
Description automatically generated with medium confidence

**Step 5: Decrypting Encrypted Files Manually**

1. To decrypt a vault encrypted file, use the ansible-vault decrypt command

***ansible-vault decrypt vault2.yml***

1. Your file will be decrypted after entering the correct password



1. On running the ***cat vault2.yml*** command you should see the actual contents of the file



**Step 6: Changing the Password of Encrypted Files**

1. Use the following command to change the password of an encrypted file

***ansible-vault rekey encrypt\_me.txt***

1. You will be prompted to enter the file’s current password. After entering It, you will be prompted to select and confirm a new password
2. You will receive the following message on successful re-encryption

Text

Description automatically generated

**Step 7: Running Ansible with Vault-Encrypted Files**

1. Let’s first create a vault-encrypted file using the command

***ansible-vault create secret\_key***

Text

Description automatically generated

1. Select and confirm a password. Enter the sample contents in the file. Save and close the file

**Top Secret!!!!! DO NOT DISCLOSE !!!!**

Shape

Description automatically generated with medium confidence

1. Create a temporary hosts file as an inventory using:

***cd /etc/ansible/***

***sudo nano hosts***

Text

Description automatically generated

1. We will add just the Ansible localhost to it. We will place it in the [database] group to prepare for a later step:

***[database]***

***localhost ansible\_connection=local***

Shape

Description automatically generated with medium confidence

1. Next, create an ansible.cfg file in the current directory if one does not already exist:

***sudo nano ansible.cfg***

******

1. For now, just add a [defaults] section and point Ansible to the inventory we just created:

***[defaults]***

***inventory = ./hosts***

Text

Description automatically generated

1. Run the following command

***ansible --ask-vault-pass -bK -m copy -a 'src=secret\_key dest=/tmp/secret\_key mode=0600 owner=root group=root' localhost***

1. Please note that you will have to enter the lab user’s password on being prompted along with the vault password. When the password is provided, Ansible will attempt to execute the task using the Vault password for any encrypted files it finds. Keep in mind that all files referenced during execution must use the same password
2. You will see the following output

Text

Description automatically generated