Lab: Managing Secrets

Introduction:

Secrets are meant to stay secret. Whether they are login credentials to a cloud service or passwords to database resources, they are secret for a reason. Should they fall into the wrong hands, they can be used to discover trade secrets, customers' private data, create infrastructure for nefarious purposes, or worse. All of which could cost you or your organization a lot of time, money, and headache!

Objectives:

- Creating new encrypted files
- Encrypting existing unencrypted files
- Editing encrypted files
- Changing the encryption password on files
- Decrypting encrypted files
- Running ansible-playbook referencing encrypted files

1. Login as user **root** with password as **linux** and change to the **./datasecret** as working directory.

```
# cd
# mkdir datasecret
# cd datasecret
```

Creating a new encrypted file

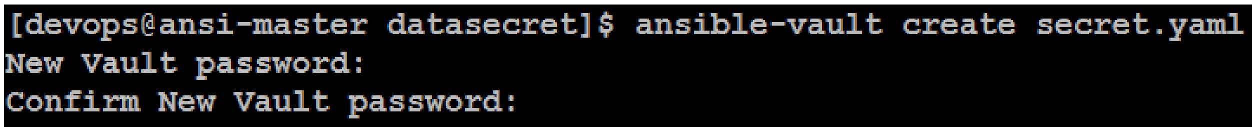
2. Type 1: PASSWORD PROMPT

2.1 Let us Create an Encrypted File

Note: It prompts for password Type: **centos**

```
# ansible-vault create secret.yaml
```

Output:



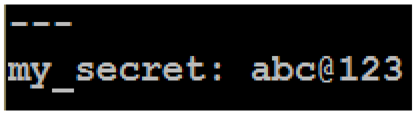
```
[devops@ansi-master datasecret]$ ansible-vault create secret.yaml
New Vault password:
Confirm New Vault password:
```

2.2 Let us add some data inside the secret.yaml

Note: Type **i** to switch into insert mode so that you can start editing the file.

```
---
My_secret: abc@123
```

Output:



```
---
my_secret: abc@123
```

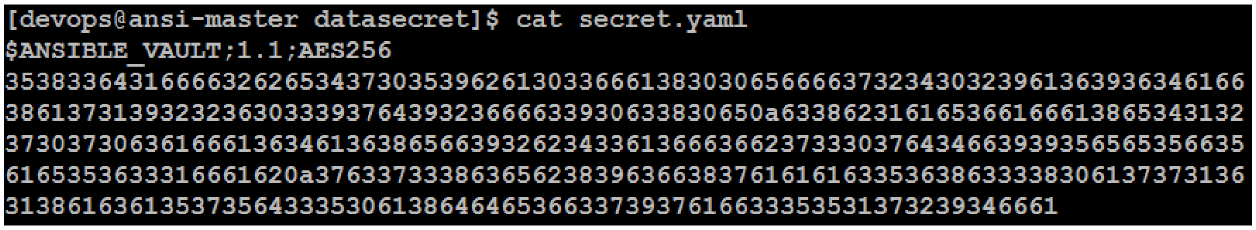
2.3 Let us save the data and exit from the vim editor by executing the below command

```
:wq!
```

2.4 Let us verify the data inside the secret.yaml

```
# cat secret.yaml
```

Output:



```
[devops@ansi-master datasecret]$ cat secret.yaml
$ANSIBLE_VAULT;1.1;AES256
35383364316666326265343730353962613033666138303065666637323430323961363936346166
38613731393232363033393764393236666339306338306 50a6338623161653661666138653431 32
```

3. Type2: PASSWORD FILE

3.1 Let us create the file and pass that file as a password

```
# echo "my long password" > password_file
```

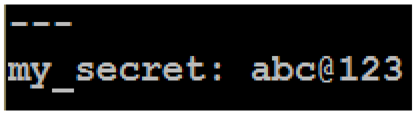
Note: Type **i** to switch into insert mode so that you can start editing the file.

```
# ansible-vault create --vault-password-file password_file
more_secret.yaml
```

3.2 Let us add some data inside the secret.yaml

```
---
My_secret: abc@123
```

Output:



```
---
my_secret: abc@123
```

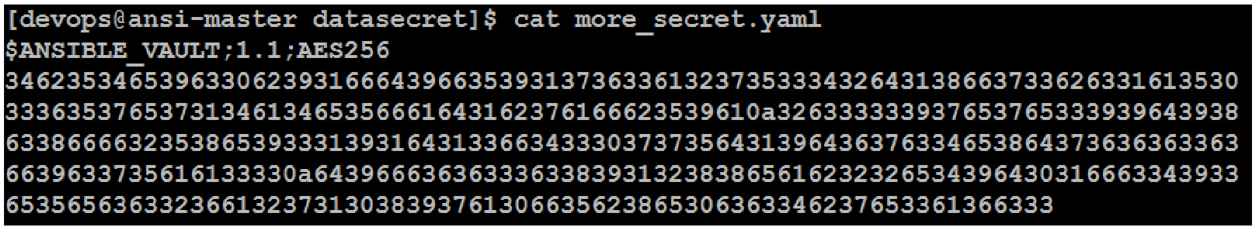
3.3 Let us save the data and exit from the editor by executing the below command

```
:wq!
```

3.4 Let us verify the data inside the secret.yaml

```
# cat more_secret.yaml
```

Output:



```
[devops@ansi-master datasecret]$ cat more_secret.yaml
$ANSIBLE_VAULT;1.1;AES256
34623534653963306239316664396635393137363361323735333432643138663733626331613530
3336353765373134613465356661643162376166623539610a326333333937653765333939643938
6338666632353865393331393164313366343330373735643139643637633465386437363636336 3
6639633735616133330a643966636363336383931323838656162323265343964303166633439 33
65356563633236613237313038393761306635623865306363346237653361366333
```

4. Type3: Password Script

4.1 Let us create script and pass that script as a password

```
# cat > password.sh << EOF
#!/bin/bash
echo "a long password"
EOF
```

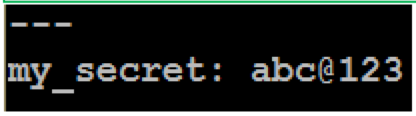
4.2 Let us add some data inside the secret.yaml

Output:

```
# ansible-vault create --vault-password-file password.sh
password-as-script.yaml
```

```

---
My_secret: abc@123
```



```
---
my_secret: abc@123
```

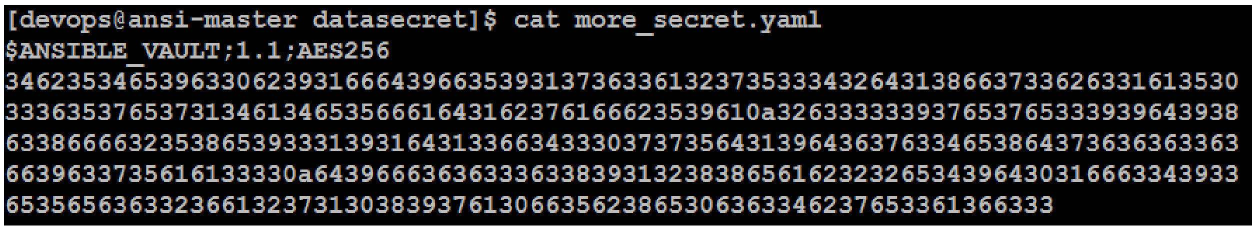
4.3 Let us save the data and exit from the editor by executing the below command

```
:wq!
```

4.4 Let us verify the data inside the secret.yaml

```
# cat password-as-script.yaml
```

Output:

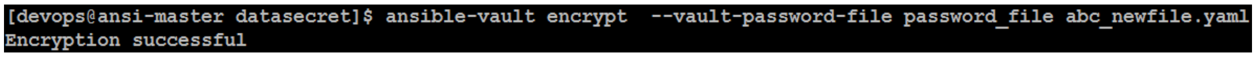


```
[devops@ansi-master datasecret]$ cat more_secret.yaml
$ANSIBLE_VAULT;1.1;AES256
34623534653963306239316664396635393137363361323735333432643138663733626331613530
3336353765373134613465356661643162376166623539610a326333333393765376533393964393 8
63386666323538653933313931643133663433303737356431396436376334653864373636363363
6639633735616133330a643966636363336338393132383865616232326534396430316663343933
6535656363323661323731303839376130663562386530636334623765336136633 3
```

5 Encrypting existing files

```
# cat > abc_newfile.yaml<< EOF
---
Something: "better than nothing"
EOF
```

Output:

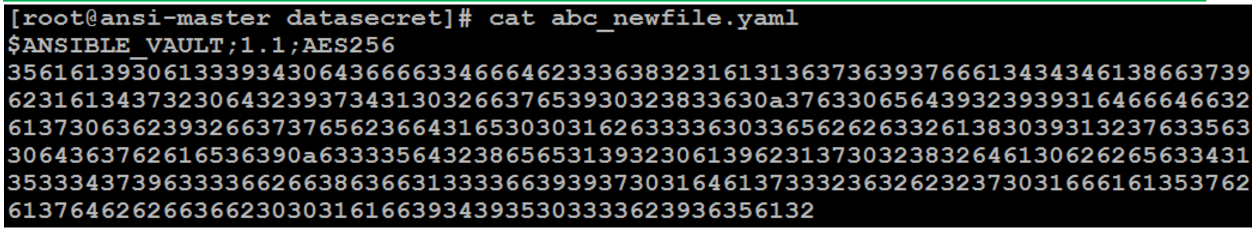


```
[devops@ansi-master datasecret]$ ansible-vault encrypt  --vault-password-file password_file abc_newfile.yaml
Encryption successful
```

Output:

```
# ansible-vault encrypt  --vault-password-file password_file
abc_newfile.yaml
```

```
# cat abc_newfile.yaml
```

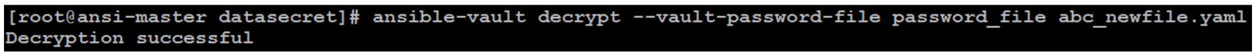


```
[root@ansi-master datasecret]# cat abc_newfile.yaml
$ANSIBLE_VAULT;1.1;AES256
35616139306133393430643666633466646233363832316131363736393766613434346138663739
62316134373230643239373431303266376539303238333630a376330656439323939316466646632
61373063623932663737656236643165303031626333363033656262633261383039313237633563
3064363762616536390a633335643238656531393230613962313730323832646130626265633431
35333437396333366266386366313333663939373031646137333236326232373031666161353762
6137646262663662303031616639343935303333623936356132
```

Decrypting encrypted files

```
# ansible-vault decrypt --vault-password-file password_file
abc_newfile.yaml
```

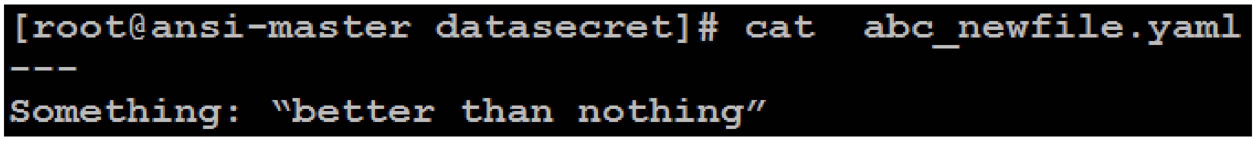
Output:



```
[root@ansi-master datasecret]# ansible-vault decrypt --vault-password-file password_file abc_newfile.yaml
Decryption successful
```

```
# cat  abc_newfile.yaml
```

Output:



```
[root@ansi-master datasecret]# cat  abc_newfile.yaml
---
Something: "better than nothing"
```

```
# ansible-vault edit secret.yaml
```

Output:

Password: **centos**

Let us add the user in the secret.yaml

```
# username: ansibleuser1
  pwhash: 12345678
```

```
# :wq!
```

Let us create named create_user.yaml

```
# cat > create_user.yaml << EOF
---
- name: create user accounts for all our servers
  hosts: ansi-node1
  become: yes
  vars_files:
  - secret.yaml
  tasks:
    - name: Creating user from secret.yaml
      user:
        name: "{{ username }}"
        password: "{{ pwhash }}"
EOF
```

Let us dry run the manifest to check the syntax

```
# ansible-playbook --syntax-check --ask-vault-pass
create_user.yaml
```

Let's create a password file named **vault-pass** to use for the playbook execution instead of asking for a password. The file must contain the plain text centos as the vault password. Change the permissions of the file to 0600.

```
# echo 'centos' > vault-pass
# chmod 0600 vault-pass
```

Let’s Execute the Ansible Playbook using the **vault-pass** file, to create the ansibleuser1 user on a remote system using the passwords stored as variables in the secret.yaml Ansible Vault encrypted file.

```
# ansible-playbook --vault-password-file=vault-pass
create_user.yaml
```