



ANSIBLE

ANSIBLE BASIC LAB MANUAL

Student Lab Kit v1.1

ABSTRACT

This lab manual is designed for students who are interested in Ansible Basic Automation

Confidential Document

Ansible Vault

Table of Contents

Lab Overview and objectives	2
<i>Guided Tasks</i>	2
Task 1: Create a vault file	2
Task 1.1: Encrypt	2
Task 1.2: Create	2
Task 2: View a vault file	3
Task 3: Edit a vault file	3
Task 4: Include vault file in Playbook	3
Task 5: Include multiple Vaults	4
Task 5: Include several files simultaneously	5
<i>Challenge:</i>	6
Part 1:	6
Part 2:	6
<i>Solution:</i>	7
Part 1:	7
Part 2:	7

Lab Overview and objectives

The purpose of this lab is to learn how to use Ansible Vault, which is a tool that allows us to keep sensitive data like passwords or keys in encrypted files. To access this tool we can use `ansible-vault` command and specify `--ask-vault-pass`, `--vault-password-file` or `--vault-id` while running the playbook (to provide the vault password).

Guided Tasks

Task 1: Create a vault file

We can create a vault file `encrypting` an existing file or `creating` a new one from scratch. We will use both options in the next tasks.

Task 1.1: Encrypt

Let's create a simple YAML file:

```
student@ansible-00-01-hivemaster:~$ vi secret.yml
---
secret_var: "ThisIsABigSecret"
```

Now let's encrypt this file using `ansible-vault encrypt` command:

```
student@ansible-00-01-hivemaster:~$ ansible-vault encrypt secret.yml
New Vault password: 123
Confirm New Vault password: 123
Encryption successful
```

If we cat the content of `secret.yml` right now, the content should be encrypted:

```
student@ansible-00-01-hivemaster:~$ cat secret.yml
$ANSIBLE_VAULT;1.1;AES256
39656563336538323136363032613366323263613237613333633735623832326631313834643138
3036353434303664316132663439626262336330626166650a626664653636346539623339653631
31363333396435646631623563626132383264343165326635343935633764373735613162613034
6166333861383763370a326561366432323236396131666336373637343136616233313661303561
62643639356139353665346433333663396539363461393862313365333561363663313231376633
3931303634386262643639376233343130363438353334383162
```

Notice that there is also `ansible-vault decrypt` command, which performs the opposite operation.

Task 1.2: Create

We can also use `ansible-vault create` command to create a vault file (make sure to choose another password for this vault):

```
student@ansible-00-01-hivemaster:~$ ansible-vault create
anothersecret.yml
New Vault password: 789
Confirm New Vault password: 789
---
another_secret_var: "ThisIsAnotherBigSecret"
```

Task 2: View a vault file

We can use `ansible-vault view` command to see the unencrypted content of a vault file:

```
student@ansible-00-01-hivemaster:~$ ansible-vault view secret.yml
Vault password:
---
secret_var: "ThisIsABigSecret"
```

Task 3: Edit a vault file

We can use `ansible-vault edit` command to edit the content of a vault file:

```
student@ansible-00-01-hivemaster:~$ ansible-vault edit secret.yml
Vault password:
```

You can just exit from editor, if you don't want to change the variable.

Task 4: Include vault file in Playbook

Let's create a playbook and include a variable from a vault file:

```
student@ansible-00-01-hivemaster:~$ vi testvault.yml
---
- name: Ansible Vault Playbook
  hosts: hivemaster
  gather_facts: no
  tasks:
    - name: Include var from vault file
      include_vars: "/home/student/secret.yml"

    - name: Print var from vault
      debug:
        msg: "{{ secret_var }}"
```

Run the playbook using `--ask-vault-pass` option:

```
student@ansible-00-01-hivemaster:~$ ansible-playbook testvault.yml --ask-vault-pass
Vault password:

PLAY [Ansible Vault Playbook]
*****
TASK [Include var from vault file]
*****
ok: [hivemaster]

TASK [Print var from vault]
*****
ok: [hivemaster] => {
  "msg": "ThisIsABigSecret"
}
[...]
```

Otherwise, we can write the password of vault in a file and pass that file as argument `--vault-password-file`:

```
student@ansible-00-01-hivemaster:~$ echo '123' > password
student@ansible-00-01-hivemaster:~$ ansible-playbook testvault.yml --vault-password-file=password
```

Task 5: Include multiple Vaults

Until Ansible v2.4 we could include more vault files only if they had the same password. Starting with that version, a new option called `vault-id` was introduced, and this provides the option to include multiple vault files with different passwords:

```
student@ansible-00-01-hivemaster:~$ vi testvault_v2.yml
---
- name: Ansible Vault Playbook
  hosts: hivemaster
  gather_facts: no
  tasks:
    - name: Include var from vault file
      include_vars: "/home/student/secret.yml"

    - name: Include var from another vault file
      include_vars: "/home/student/anothersecret.yml"

    - name: Print var from vault1
      debug:
        msg: "{{ secret_var }}"

    - name: Print var from vault2
```

```
debug:
  msg: "{{ another_secret_var }}"
```

```
student@ansible-00-01-hivemaster:~$ ansible-playbook testvault_v2.yml --
vault-id @prompt --vault-id @prompt
```

Notice that we used @prompt for vault-id, so Ansible will ask us to provide the passwords for the 2 vaults (the order doesn't matter because Ansible tries every possible combination).

Task 5: Include several files simultaneously

As you have already seen, we used 2 tasks to include 2 vault files. In case that we have several files (not only vaults files) we can put them inside a directory and include the entire directory at once:

```
student@ansible-00-01-hivemaster:~$ mkdir my_var_files
student@ansible-00-01-hivemaster:~$ cp *secret.yml my_var_files/
student@ansible-00-01-hivemaster:~$ ls my_var_files/
anothersecret.yml  secret.yml
```

Now let's adjust the previous playbook:

```
---
- name: Ansible Vault Playbook
  hosts: hivemaster
  gather_facts: no
  tasks:
    - name: Include directory
      include_vars:
        dir: "/home/student/my_var_files/"

    - name: Print var from vault1
      debug:
        msg: "{{ secret_var }}"

    - name: Print var from vault2
      debug:
        msg: "{{ another_secret_var }}"
```

Challenge:

Part 1:

Create a new vault called `mybanner.yml` (you can create it directly using `ansible-vault create`):

```
---
my_ssh_banner: "THIS IS A RESTRICTED AREA AND ANY UNAUTH ACCESS WILL BE
PROSECUTED!"
```

Part 2:

Create a new playbook, called `modifybanner.yml` in which we are going to change the banner for SSH connections (only for `hivemaster`), which is located in `(/etc/ssh/banner)` with the variable `my_ssh_banner` imported from the vault. We also have to enable `ssh` banner in `/etc/sshd/sshd_config`.

Notice that you may not want to use a secret text (from a vault) as you banner in real life experience, but this task is just for educational purpose.

Solution:

Part 1:

```
student@ansible-00-01-hivemaster:~$ ansible-vault create mysshbanner.yml
New Vault password:
Confirm New Vault password:

---
my_ssh_banner: "THIS IS A RESTRICTED AREA AND ANY UNAUTH ACCESS WILL BE
PROSECUTED!"
```

Part 2:

```
student@ansible-00-01-hivemaster:~$ vi modifybanner.yml
---
- name: Modify SSH banner
  hosts: hivemaster
  become: true
  tasks:
    - name: Include var from vault
      include_vars: "/home/student/mysshbanner.yml"

    - name: Create banner file
      copy:
        content: "{{ my_ssh_banner }}"
        dest: /home/student/banner.txt
        owner: student

    - name: Insert a newline at the end of file
      lineinfile:
        path: /home/student/banner.txt
        line: ''

    - name: Copy file to proper location
      copy:
        src: banner.txt
        dest: /etc/ssh/banner

    - name: Modify sshd_config
      lineinfile:
        dest: /etc/ssh/sshd_config
        regexp: "^Banner"
        line: "Banner /etc/ssh/banner"
        state: present

    - name: Restart SSH service
```



```
service:
  name: ssh
  state: restarted
```

```
student@ansible-00-01-hivemaster:~$ ansible-playbook modifybanner.yml --
vault-id @prompt
Vault password (default):
```

```
PLAY [Modify SSH banner]
```

```
*****
```

```
TASK [Gathering Facts]
```

```
*****
```

```
ok: [hivemaster]
```

```
TASK [Include var from vault]
```

```
*****
```

```
ok: [hivemaster]
```

```
TASK [Create banner file]
```

```
*****
```

```
changed: [hivemaster]
```

```
TASK [Insert a newline at the end of file]
```

```
*****
```

```
changed: [hivemaster]
```

```
TASK [Copy file to proper location]
```

```
*****
```

```
ok: [hivemaster]
```

```
TASK [Modify sshd_config]
```

```
*****
```

```
changed: [hivemaster]
```

```
TASK [Restart SSH service]
```

```
*****
```

```
changed: [hivemaster]
```

```
PLAY RECAP
```

```
*****
```

```
hivemaster      : ok=7    changed=4    unreachable=0
failed=0        skipped=0  rescued=0    ignored=0
```