Se former autrement

HONORIS UNITED UNIVERSITIES

La sécurité des variables dans Ansible

avec Vault

Initiation à l'automatisation

4ArcTIC

2024-2025

Introduction :

Dans toute infrastructure informatique, certaines données de configuration sont sensibles par nature et ne doivent jamais être exposées publiquement. Cela inclut, par exemple, les mots de passe, les clés API, et d'autres informations confidentielles.
Ansible Vault offre une solution sécurisée permettant de chiffrer ces données tout en les intégrant dans vos playbooks et configurations Ansible. Dans ce Lab, nous allons explorer comment utiliser Ansible Vault pour protéger des informations sensibles, garantissant ainsi la sécurité de votre infrastructure automatisée.

Prérequis :

- Deux machines virtuelles exécutant Ubuntu 22.04 avec un serveur OpenSSH installé : l'une est le nœud maître et l'autre est le nœud de travail.
- Le nœud de travail doit être accessible via SSH depuis le nœud maître.

Objectif :

L'objectif de ce Lab est de :

- Comprendre le fonctionnement d'Ansible Vault et ses avantages en matière de sécurité.
- Apprendre à chiffrer et déchiffrer des fichiers Ansible contenant des informations sensibles.
- Intégrer Ansible Vault dans un playbook pour gérer des variables protégées.
- Gérer plusieurs fichiers secrets avec des mots de passe différents

Partie 1 : Chiffrement d'une variable sensible avec Vault

Etape 1 - Création de playbook

Soit le playbook suivant nommé **playbook_vault.yml** contenant une variable sensible en clair :

```
- name: Playbook Vault
  hosts: all
  vars:
     mypassword: mysecretpassword
  tasks:
    - name: print variable
      ansible.builtin.debug:
         var: mypassword
```

Problème : La variable mypassword est visible en clair, ce qui pose un problème de sécurité.

Etape 2 - Création d'un fichier de variables secrètes

a. Créez un répertoire nommé **secrets** dans le même répertoire que votre playbook.
b. Dans ce répertoire, créez un fichier nommé **credentials.yml** et déplacez la variable mypassword du playbook vers ce fichier.
c. Le fichier credentials.yml devrait ressembler à ceci :

```
mypassword: mysecretpassword
```

d. Pour permettre à Ansible d'accéder au répertoire secrets et au fichier credentials.yml, vous devez ajuster les permissions.

```
sudo chmod o+w /etc/ansible/secrets
sudo chmod o+w /etc/ansible/secrets/credentials.yml
```

Etape 3 - Chiffrement du fichier de variables avec Ansible Vault

a. Utilisez la commande suivante pour chiffrer le fichier credentials.yml avec Ansible Vault.

```
esprit@esprit-virtual-machine:/etc/ansible/secrets$ ansible-vault encrypt credentials.yml
New Vault password:
Confirm New Vault password:
Encryption successful
```

Vous serez d'abord invité à saisir et confirmer un mot de passe. **(Ce mot de passe est crucial, ne l'oubliez pas !**

b. Vérifiez le contenu du fichier chiffré avec la commande cat. Vous verrez que le contenu est désormais **illisible**.
c. Pour afficher le contenu en texte clair de votre fichier crypté, tapez la commande suivante :

(Vous serez invité à saisir le mdp que vous avez configuré précédemment).

```
ansible-vault view credentials.yml
```

```
esprit@esprit-virtual-machine:/etc/ansible/secrets$ ansible-vault view credentials.yml
Vault password:
```

Etape 4 - Intégration du fichier chiffré dans le playbook

Modifiez le playbook playbook_vault.yml pour inclure le fichier de variables chiffré en utilisant la directive **vars_files** :

```
- name: Playbook Vault
  hosts: all
  vars_files:
    - secret/credentials.yml
  tasks:
    - name: print variable
      ansible.builtin.debug:
          var: mypassword
```

Etape 5 - Exécution du Playbook

Lors du lancement de votre playbook utilisez l'option **--ask-vault-pass** afin d'avoir une saisie utilisateur pour fournir votre mot de passe Vault, comme suit :

```
esprit@esprit-virtual-machine:/etc/ansible$ ansible-playbook -i inv.ini playbook_vault.yml --ask-vault-pass
Vault password:

PLAY [Playbook Vault] ***************************************************************************************
**************************

TASK [Gathering Facts] **************************************************************************************
**************************
ok: [192.168.231.139]

TASK [Print variable] ***************************************************************************************
**************************
ok: [192.168.231.139] => {
    "mypassword": "mysecretpassword"
}

PLAY RECAP **************************************************************************************************
**************************
192.168.231.139            : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ign
ored=0
```

Bien que la saisie manuelle d'un mot de passe lors de l'exécution du playbook fonctionne, ce n'est pas pratique pour une utilisation dans le monde réel.

Pour ne pas saisir le mot de passe Vault à chaque fois que vous exécutez votre playbook, vous pouvez stocker votre mot de passe Vault dans un fichier et référencer le fichier pendant l'exécution, Ansible lira le mot de passe à partir du fichier requis.

a. Exécutez la commande ci-dessous pour stocker le mdp Ansible Vault dans un fichier qu'on nomme **.my_vault_pass**, remplacez ensuite **my_vault_password** par votre mdp Vault actuel :

```
echo 'my_vault_password ' > .my_vault_pass
```

b. Lorsque vous lancez votre playbook remplacez l'option --ask-vault-pass **par l'option --vault-password-file** comme suit :

```
esprit@esprit-virtual-machine:/etc/ansible$ ansible-playbook -i inv.ini playbook_vault.yml --vault-password-file=.my_vault_pass

PLAY [Playbook Vault] ***************************************************************************************

TASK [Gathering Facts] **************************************************************************************
ok: [192.168.231.139]

TASK [Print variable] ***************************************************************************************
ok: [192.168.231.139] => {
    "mypassword": "mysecretpassword"
}

PLAY RECAP **************************************************************************************************
192.168.231.139            : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

➔ Cette option vous donne la liberté de ne pas spécifier le mdp du fichier crypté à chaque fois.

Partie 2 : Utilisation de plusieurs fichiers secrets dans un playbook

Cas 1 : Single Vault Password

a. Sous le même répertoire secret créez un nouveau fichier **group_credentials.yml** qui va contenir la variable **http_port : 8080**. Vous devez ajuster les permissions du fichier.

b. Encrypter le fichier group_credentials.yml à l'aide de la commande ansible-vault encrypt et **veuillez saisir le même mot de passe que la partie précédente**.

c. Modifier votre playbook comme suit :

```
- name: Playbook Vault
  hosts: all
  vars_files:
    - secret/credentials.yml
    - secrets/group_credentials.yml
  tasks:
    - name: print variable
      ansible.builtin.debug:
          var: mypassword
    - name: print variable2
      ansible.builtin.debug:
          var: http_port
```

d. Exécutez ensuite le playbook.

Note : Les options --ask-vault-pass et --vault-password-file peuvent être utilisées tant qu'un seul mot de passe est nécessaire pour l'exécution du playbook.)

```
esprit@esprit-virtual-machine:/etc/ansible$ ansible-playbook -i inv.ini playbook_vault.yml --vault-password-file=.my_vault_pass

PLAY [Playbook Vault] ***********************************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************************
ok: [192.168.231.139]

TASK [Print variable] ***********************************************************************************************************************
ok: [192.168.231.139] => {
    "mypassword": "mysecretpassword"
}

TASK [print variable2] **********************************************************************************************************************
ok: [192.168.231.139] => {
    "http_port": 8080
}

PLAY RECAP **********************************************************************************************************************************
192.168.231.139            : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
esprit@esprit-virtual-machine:/etc/ansible$ ansible-playbook -i inv.ini playbook_vault.yml --ask-vault-pass
Vault password:

PLAY [Playbook Vault] ***********************************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************************
ok: [192.168.231.139]

TASK [Print variable] ***********************************************************************************************************************
ok: [192.168.231.139] => {
    "mypassword": "mysecretpassword"
}

TASK [print variable2] **********************************************************************************************************************
ok: [192.168.231.139] => {
    "http_port": 8080
}

PLAY RECAP **********************************************************************************************************************************
192.168.231.139            : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Cas 2 : Multiple Vault Passwords

L'objectif de cette étape est de définir deux mots de passe vault différents et de différencier un mot de passe d'un autre avec des ID.

a. Déchiffrez les fichiers credentials.yml et group_credentials.yml :

```
ansible-vault decrypt secrets/credentials.yml

ansible-vault decrypt secrets/group_credentials.yml
```

b. Cryptez de nouveau les deux fichiers **en choisissant des mots de passes différents** et en associant un ID à chaque mot de passe pour les séparer.

```
ansible-vault encrypt secrets/credentials.yml --vault-id id1@prompt

ansible-vault encrypt secrets/group_credentials.yml --vault-id id2@promp
```

c. Vérifiez à quoi ressemble le contenu de vos fichiers chiffrés avec la commande « cat »

```
esprit@esprit-virtual-machine:/etc/ansible$ cat secrets/credentials.yml
$ANSIBLE_VAULT;1.2;AES256;id1
38393862636364663466313437626439363765393261393562303535613232346166353239346262
31383663633336333934613834323363303736383663353330a303130313435363836663063383861
34623166316263666264306666356632663734613364616661346364623438383464656133323165
32366336303831346 40a643933633130366131666361383031353864373037336361383639316162
6364353663363337326366613161353033643237336236633366383737643033656 5
```

➔ Le volt ID est le dernier élément avant le contenu chiffré.

Maintenant, l'exécution de votre play book nécessite deux variables que vous avez chiffrés avec deux mots de passe différents, vous devez donc utiliser l'option --vault-id.

d. Exécutez votre playbook en spécifiant les id nécessaires pour déchiffrer le fichier chiffré avec l'option **--vault-id**.

Note : Ansible vous demandera les mots de passe dans l'ordre dans lequel l'id a été fourni sur la ligne de commande.

```
ansible-playbook --vault-id id1@prompt --vault-id id2@prompt playbook_vault.yml -i inventory_name
```

```
esprit@esprit-virtual-machine:/etc/ansible$ ansible-playbook --vault-id id1@prompt --vault-id id2@prompt playbook_vault.yml -i inv.ini

Vault password (id1):
Vault password (id2):

PLAY [Playbook Vault] ***********************************************************************************************************************

TASK [Gathering Facts] ***********************************************************************************************************************
ok: [192.168.231.139]

TASK [Print variable] ************************************************************************************************************************
ok: [192.168.231.139] => {
    "mypassword": "mysecretpassword"
}

TASK [print variable2] ***********************************************************************************************************************
ok: [192.168.231.139] => {
    "http_port": 8080
}

PLAY RECAP ***********************************************************************************************************************************
192.168.231.139            : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Conclusion

Ce TP vous a permis de découvrir comment sécuriser des variables sensibles dans Ansible en utilisant Ansible Vault. Vous avez appris à chiffrer et déchiffrer des fichiers, à intégrer ces fichiers dans un playbook, et à gérer plusieurs fichiers secrets avec des mots de passe différents. Ces compétences sont essentielles pour garantir la sécurité d'une infrastructure automatisée.