Ansible vault - Verschlüsseln von Daten

- Ansible Vault Befehle

Ansible Vault Befehle

Beschreibung:

Dateien mit Passwörtern in Klartext speichern ist nie gut und schon gar nicht in ein git hochladen. Denn das zieht sich durch jede Version durch.

Dafür gibt es ansible vault.
Man braucht ein Master Passwort das bei jedem ausführen eines Playbooks übergeben wird.

Beispiele:

Datei zu verschlüsseln :

```
ansible-vault encrypt <dateiname>



ansible-vault encrypt host.ini


Output:
neues Passowrt vergeben
```

Datei bearbeiten, dabei öffnet sich leider nur vim.

Um den Editor zu ändern, den Ansible Vault für das Bearbeiten verschlüsselter Dateien verwendet, kannst du die Umgebungsvariable `EDITOR` auf deinem System setzen. Standardmäßig verwendet Ansible den Editor, der in der `EDITOR` Umgebungsvariablen festgelegt ist. Wenn diese Variable nicht gesetzt ist, greift Ansible auf einen Standard-Editor zurück, der oft `vi` oder `vim` ist.

Um einen anderen Editor zu verwenden, wie zum Beispiel `nano`, `emacs`, oder einen grafischen Editor wie `gedit` oder `xed`, kannst du die `EDITOR` Umgebungsvariable in deiner Shell-Sitzung setzen. Hier sind Beispiele, wie du das für verschiedene Shells machen kannst:

Bash oder Zsh

Für Bash oder Zsh (die gängigsten Shells auf Linux und macOS) kannst du den Editor für die aktuelle Sitzung mit folgendem Befehl setzen:

bash

`export EDITOR=nano`

Wenn du möchtest, dass diese Einstellung dauerhaft bleibt, kannst du diese Zeile zu deiner `~/.bashrc`, `~/.bash_profile` oder `~/.zshrc` Datei hinzufügen, abhängig von deiner Shell und Betriebssystemkonfiguration.

Ich nutze xed,

`export EDITOR=xed`

Nun kann man mit ansible-vault edit Dateien öffnen

```
ansible-vault edit <dateiname>
ansible-vault edit hosts.ini

Output:
Kennwort angeben
```

Nun wird der Texteditor geöffnet, nach dem bearbeiten speichern und schließen.
Die Änderung fließt zurück.

Datei entschlüsseln

Um eine Datei zu entschlüsseln nutzen wir den decrypt Parameter

```
ansible-vault encrypt <dateiname>


ansible-vault encrypt host.ini

Output:
Passowrt eingeben
```

Übergeben von Kennwörtern an Playbooks

Wenn du ein Ansible Playbook ausführst und das `--ask-vault-pass`, `--vault-password-file` oder eine Umgebungsvariable für das Vault-Passwort verwendest, wird dieses Passwort für alle Operationen verwendet, die eine Entschlüsselung erfordern, einschließlich des Zugriffs auf deine verschlüsselte `hosts.ini` Datei.

1. Passwort bei der Aufforderung eingeben

Wenn du kein Passwort auf andere Weise angibst, fordert Ansible dich bei der Ausführung des Playbooks zur Eingabe des Vault-Passworts auf:

```
ansible-playbook playbook.yml -i hosts.ini --ask-vault-pass
```

2. Passwortdatei verwenden

Eine bequemere Methode, besonders für automatisierte Skripte, ist die Verwendung einer Passwortdatei. Diese Datei enthält das Vault-Passwort und kann im Befehl referenziert werden. **Stelle sicher, dass diese Passwortdatei sicher aufbewahrt wird und nicht ins Repository gelangt.**

Erstelle zuerst eine Datei (z.B. `.vault_pass.txt`) und schreibe dein Passwort hinein. Verwende dann den `--vault-password-file` Schalter:

```
ansible-playbook playbook.yml --vault-password-file .vault_pass.txt
```

3. Umgebungsvariablen

Du kannst auch eine Umgebungsvariable `ANSIBLE_VAULT_PASSWORD_FILE` setzen, die auf die Passwortdatei verweist. Ansible wird diese Variable automatisch erkennen und verwenden, ohne dass der Schalter `--vault-password-file` benötigt wird.

```
export ANSIBLE_VAULT_PASSWORD_FILE=/pfad/zu/.vault_pass.txt
ansible-playbook playbook.yml
```

Das Kennwort direkt in einer Umgebungsvariable speichern, geht nicht.
es geht immer nur über die Umgebungsvariable die die Datei enthält mit dem Password.

Tipps für die Verwendung verschlüsselter Inventardateien

- Wenn du mehrere Umgebungen (z.B. Staging, Produktion) hast, erwäge die Verwendung unterschiedlicher Vault-Passwörter für zusätzliche Sicherheit.
- Integriere das Vault-Passwort nicht in deine Versionskontrollsysteme. Verwende sichere Speicheroptionen oder Umgebungsvariablen auf den Ausführungssystemen.
- Stelle sicher, dass alle Teammitglieder, die das Playbook ausführen müssen, Zugriff auf das Vault-Passwort oder die Passwortdatei in sicherer Weise haben.

Durch die Verwendung dieser Methoden kannst du sicherstellen, dass deine verschlüsselten Dateien effektiv in deinen Ansible-Workflows genutzt werden können, ohne die Sicherheit deiner sensiblen Daten zu gefährden.