Ansible Vault

Eyes On Cloud

What are Vaults?

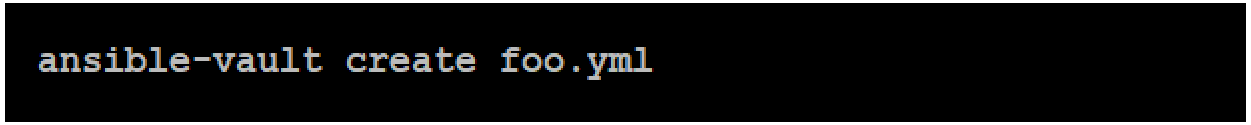
- "Vault" is a feature of ansible that allows keeping sensitive data such as passwords or keys in encrypted files, rather than as plaintext in your playbooks or roles.
- To enable this feature, a command line tool - ansible-vault - is used to edit files, and a command line flag (--ask-vault-pass or --vault-password-file) is used.
- Alternately, you may specify the location of a password file or command Ansible to always prompt for the password in your ansible.cfg file.
- These options require no command line flag usage.

What Can Be Encrypted With Vault

- The vault feature can encrypt any structured data file used by Ansible.
- This can include "group_vars/" or "host_vars/" inventory variables, variables loaded by "include_vars" or "vars_files", or variable files passed on the ansible-playbook command line with -e @file.yml or -e @file.json. Role variables and defaults are also included.
- Ansible tasks, handlers, and so on are also data so these can be encrypted with vault as well.
- To hide the names of variables that you're using, you can encrypt the task files in their entirety.

Creating Encrypted Files

- To create a new encrypted data file, run the following command:

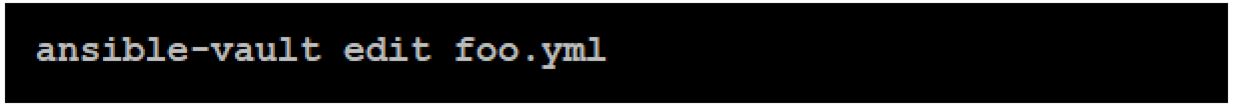


```
ansible-vault create foo.yml
```

- First you will be prompted for a password. The password used with vault currently must be the same for all files you wish to use together at the same time.
- After providing a password, the tool will launch the editor. Once you are done with the editor session, the file will be saved as encrypted data.

Editing Encrypted Files

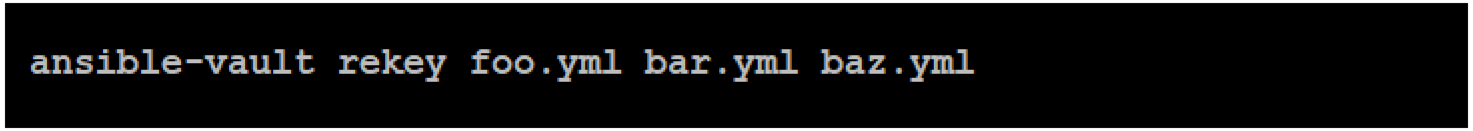
- To edit an encrypted file in place, use the ansible-vault edit command.
- This command will decrypt the file to a temporary file and allow you to edit the file, saving it back when done and removing the temporary file:



```
ansible-vault edit foo.yml
```

Rekeying Encrypted Files

- Should you wish to change your password on a vault-encrypted file or files, you can do so with the rekey command:

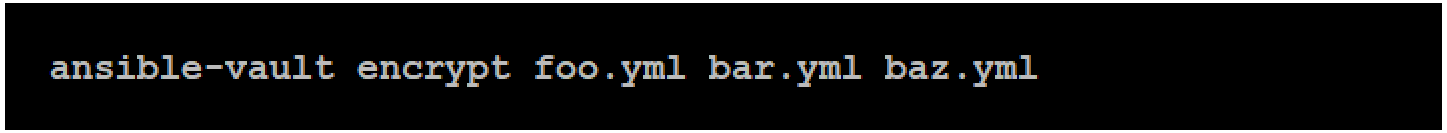


```
ansible-vault rekey foo.yml bar.yml baz.yml
```

- This command can rekey multiple data files at once and will ask for the original password and also the new password.

Encrypting Unencrypted Files

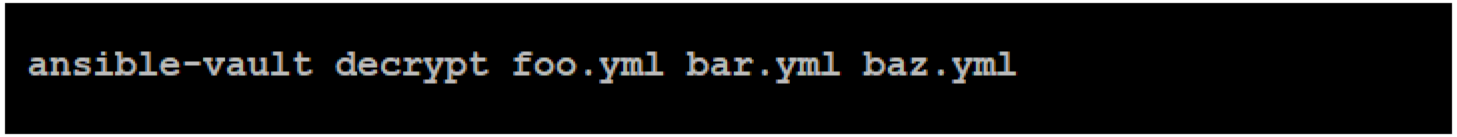
- If you have existing files that you wish to encrypt, use the ansible-vault encrypt command. This command can operate on multiple files at once:



```
ansible-vault encrypt foo.yml bar.yml baz.yml
```

Decrypting Encrypted Files

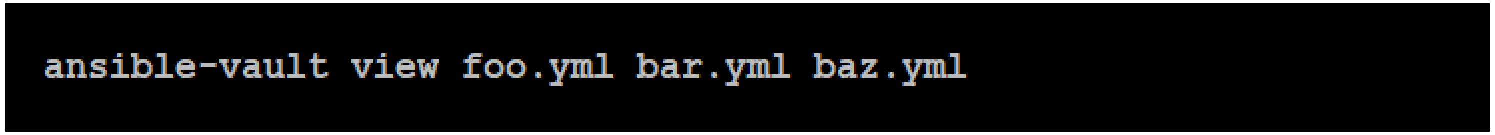
- If you have existing files that you no longer want to keep encrypted, you can permanently decrypt them by running the ansible-vault decrypt command.
- This command will save them unencrypted to the disk, so be sure you do not want ansible-vault edit instead:



```
ansible-vault decrypt foo.yml bar.yml baz.yml
```

Viewing Encrypted Files

- If you want to view the contents of an encrypted file without editing it, you can use the ansible-vault view command:

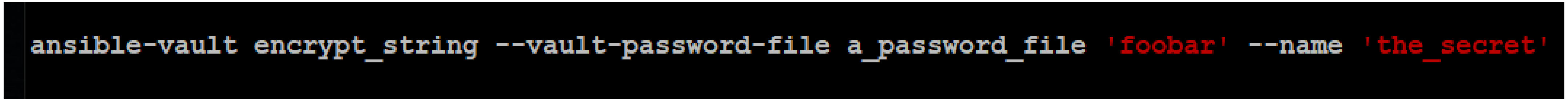


```
ansible-vault view foo.yml bar.yml baz.yml
```

Eyes On Cloud

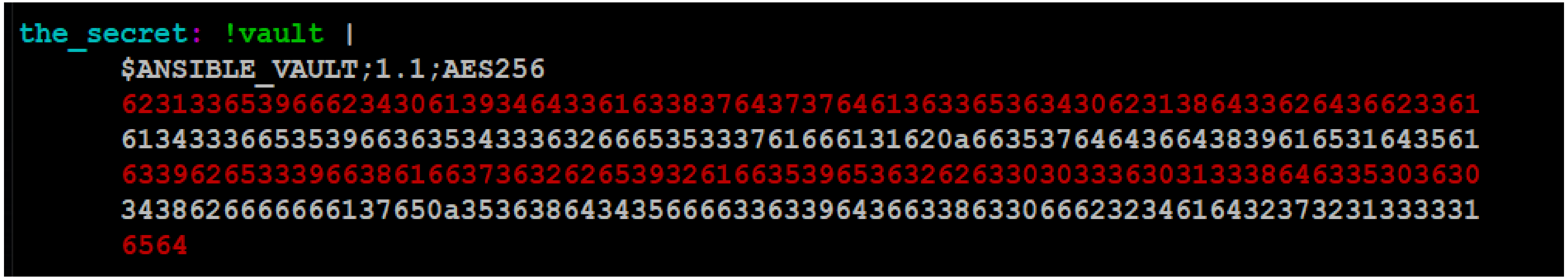
Use encrypt_string to create encrypted variables to embed in yaml

- The ansible-vault **encrypt_string** command will encrypt and format a provided string into a format that can be included in ansible-playbook YAML files.
- To encrypt a string provided as a cli arg:



```
ansible-vault encrypt_string --vault-password-file a_password_file 'foobar' --name 'the_secret'
```

- Result:



```
the_secret: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          62313365396662343061393464336163383764373764613633653634306231386433626436623361
          6134333665353966363534333632666535333761666131620a663537646436643839616531643561
          63396265333966386166373632626539326166353965363262633030333630313338646335303630
          3438626666666137650a353638643435666633633964366338633066623234616432373231333331
          6564
```

Eyes On Cloud

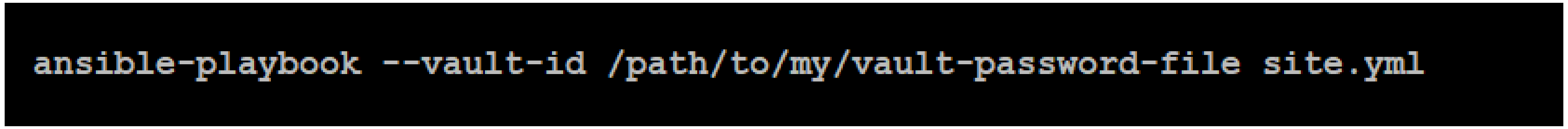
Vault Ids and Multiple Vault Passwords

- A vault id is an identifier for one or more vault secrets. Vault ids is a way to provide a label for a particular vault password.
- Vault encrypted content can specify which vault id it was encrypted with.
- Prior to Ansible 2.4, only one vault password could be used at a time. Post Ansible 2.4, multiple vault passwords can be used each time Ansible runs, so any vault files or vars that needed to be decrypted all had to use the same password.
- Since Ansible 2.4, vault files or vars can be that are encrypted with different passwords can be used at the same time.

Providing Vault Passwords

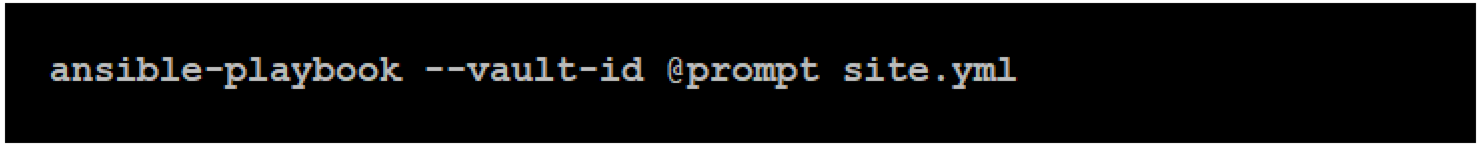
- The recommended way to provide a vault password from the cli is to use the --vault-id cli option.

For example, to use a password store in the text file /path/to/my/vault-password-file:



```
ansible-playbook --vault-id /path/to/my/vault-password-file site.yml
```

To prompt for a password



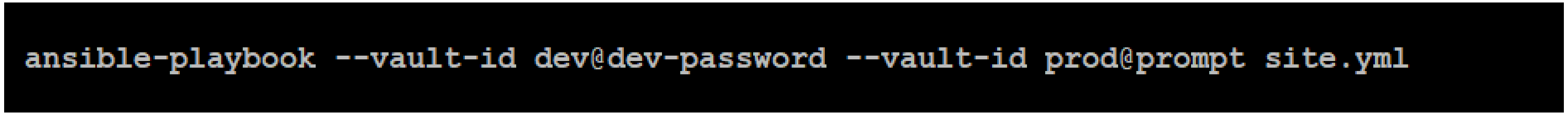
```
ansible-playbook --vault-id @prompt site.yml
```

Multiple vault passwords

- If multiple vault passwords are provided, by default Ansible will attempt to decrypt vault content by trying each vault secret in the order they were provided on the command line.

  For example, to use a ‘dev’ password read from a file and to be prompted

  for the ‘prod’ password:



```
ansible-playbook --vault-id dev@dev-password --vault-id prod@prompt site.yml
```

Eyes On Cloud