Ansible Vault

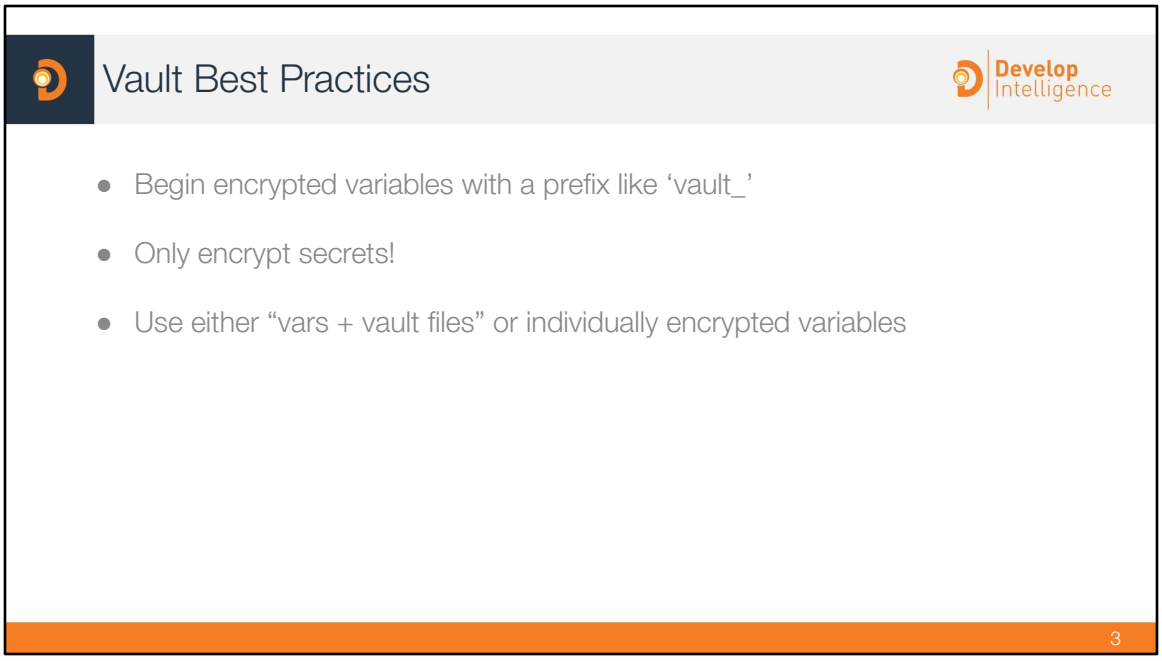
Develop Intelligence

ansible-vault

Develop Intelligence

```
$ ansible-vault -h
Usage: ansible-vault
[create|decrypt|edit|encrypt|encrypt_string|rekey|vi
ew] [options] [vaultfile.yml]

encryption/decryption utility for Ansible data files
```



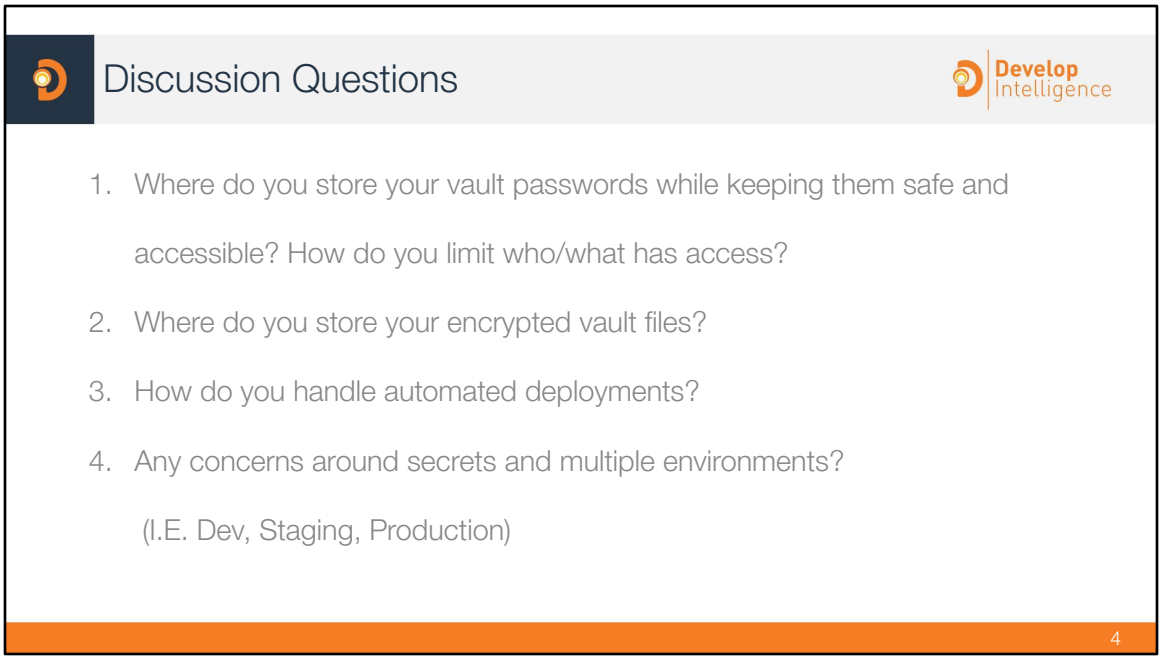
http://docs.ansible.com/ansible/latest/user_guide/playbooks_best_practices.html#variables-and-vaults

Begin encrypted variables with ‘vault_’
Only encrypt secrets!
Save yourself headache of obscuring everything

Use either “vars & vault files” or individually encrypted variables
* this is important so during code reviews it is obvious what changed unless it is sensitive.
For example:

```
├── group_vars
│   ├── all
│   │   └── vars.yml
│   └── production
│       ├── vars.yml
│       └── vault.yml
```

Where only vault.yml contains sensitive information.



Think about potential answers to these questions and their tradeoffs.

1. Where do you store your vault passwords while keeping them safe and accessible? How do you limit who/what has access?

- Password files could be kept on the Ansible host itself. Think about the tradeoffs associated with this.
- The password file may be a script that requests the password from an external service or echos an environment variable.
- External secret management system

2. Where do you store your encrypted vault files?

- how do you help prevent the unencrypted version from being committed?
- how do you ensure the data cannot be lost?
- What if you rekey and the old versions exist in VCS? Is that okay? Would you need to change all of your secrets?

3. I.E. how do you supply the password to Jenkins or another CI/CD tool?

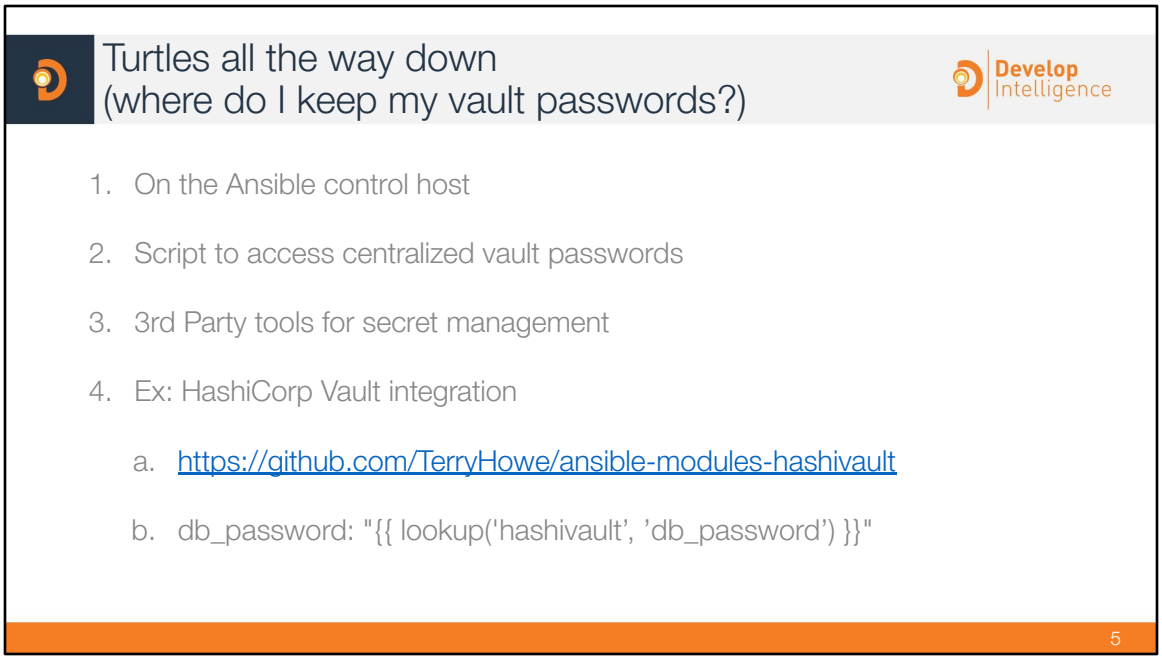
https://devops.stackexchange.com/questions/709/what-are-best-practices-for-using-ansible-vault-on-public-cis-and-source-contro

* have a vault password script that grabs the password from an ENV variable.
* OR just have the CI/CD tool expose the password in a file

- the main thing to remember is having multiple sources of truth for your password is bad. Changing the vault password should ideally not take down your ability to deploy.

4. What concerns are there with different environments? (I.E. Dev, Staging, Production)

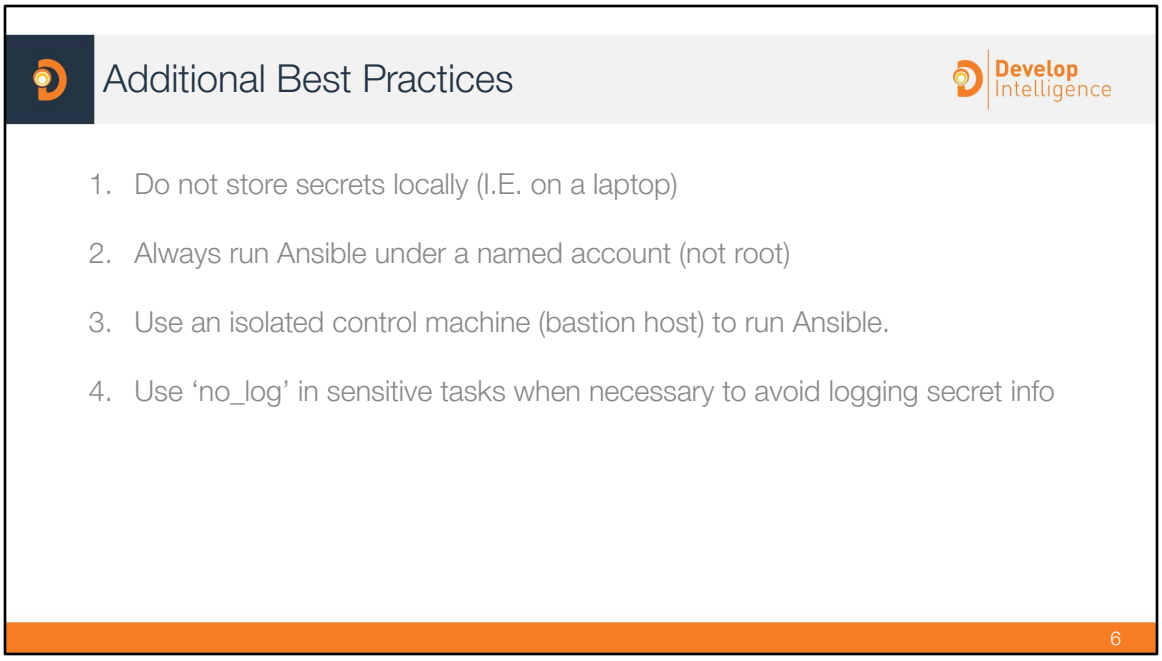
- Dev teams likely need the ability to run the playbooks against either their own environments or a dev environment. Maybe not everyone should be able to see the values for production. This can be solved by different vault files for each environment with different passwords or using vault ids and multiple passwords.



1. On the Ansible control host
2. Script to access centralized vault passwords
3. 3rd Party Tools

HashiCorp Vault is a secret management system for storing passwords, api keys, certificates, etc.

- Allows more granular control over who/what has access to certain secret information
- Can provide an access audit trail
- etc



1.Do not store secrets locally

2.Always run under a named account

Every operator should be required access to each target host.

2.Ultimately, human beings are responsible for the actions done and therefore it should be a humans account. (unless running in an automated fashion, but there should still be an audit trail)

3.Use a bastion host

Only accessible by those running Ansible

Not directly accessible FROM the target hosts

I like to go a step further and not even have the VM running

4.*no_log* task parameter

Ideally only done on particular tasks to prevent difficulty when debugging.

Resources

Develop Intelligence

- Ansible Vault docs
  https://docs.ansible.com/ansible/latest/user_guide/vault.html