

Exception handling

```
- hosts: all
# ignore_errors: yes
tasks:
  - block:
      - name: Create the ansible deploy lock file
        file: path=/tm/test state=touch
    rescue:
      - file: path=/tmp/test state=touch
#       - fail:
#         msg: "The deploy has failed."
  always:
    - name: Remove the ansible deploy lock file
      file: path=/tmp/test state=absent
```

```
[root@ip-172-31-27-155 Ansible]# ansible-playbook 43.yml

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [13.127.244.80]

TASK [Create the ansible deploy lock file] *****
fatal: [13.127.244.80]: FAILED! => {"changed": false, "msg": "Error, could not touch target: [e": "absent"]

TASK [file] *****
changed: [13.127.244.80]

TASK [Remove the ansible deploy lock file] *****
changed: [13.127.244.80]

PLAY RECAP *****
13.127.244.80 : ok=3  changed=2  unreachable=0  failed=1
```

include tasks from external file

```
#task.yml
```

```
---
```

```
- yum:  
  name=samba  
  state=present
```

```
- service:  
  name=smb  
  state=started
```

```
---
```

```
- name: external task
```

```
hosts: all
```

```
tasks:
```

```
- include_tasks: task.yml
```

```
#- import_tasks: task.yml
```

```
[root@ip-172-31-29-57 Ansible]# ansible-playbook 29.yml
```

```
PLAY [external task] *****
```

```
TASK [Gathering Facts] *****  
ok: [13.127.244.80]
```

```
TASK [include_tasks] *****  
included: /tmp/Ansible/task.yml for 13.127.244.80
```

```
TASK [yum] *****  
ok: [13.127.244.80]
```

```
TASK [service] *****  
changed: [13.127.244.80]
```

```
PLAY RECAP *****  
13.127.244.80 : ok=4 changed=1 unreachable=0 failed=0
```

Include playbook from external file

- import_playbook: 27.yml
 - import_playbook: 28.yml
 - name: testing
- hosts: dev
- tasks:
- include: task.yml

```
[root@ip-172-31-29-57 Ansible]# ansible-playbook 30.yml

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [13.127.244.80]

TASK [Fail task when both files are identical] *****
fatal: [13.127.244.80]: FAILED! => {"changed": true, "failed_when_result": true, "rc": 0, "out_lines": []}
...ignoring

TASK [debug] *****
ok: [13.127.244.80] => {
  "diff_cmd": {
    "changed": true,
    "failed": true,
    "failed_when_result": true,
```

Ansible vault

- Ansible Vault is a feature of ansible that allows you to keep sensitive data such as passwords or keys in encrypted files, rather than as plaintext in 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.

- **Creating Encrypted Files**

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 whatever editor you have defined with **\$EDITOR**, and defaults to vi (before 2.1 the default was vim).

Cont.

ansible-vault create vault_variable.yml

ansible-vault create vault_play.yml

```
password:      ---
- admin        - hosts: all
                vars_files:
- root          - vault_variable.yml
- user          tasks:
                - debug: msg="{{ password[1] }}"
```

➤ Editing Encrypted Files

ansible-vault edit vault_variable.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 vault_variable.yml

```
[root@ip-172-31-27-155 Ansible]# ansible-vault create vault_play.yml
New Vault password:
Confirm New Vault password:
```

```
[root@ip-172-31-27-155 Ansible]# ansible-playbook vault_play.yml --ask-vault-pass
Vault password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [13.127.244.80]

TASK [debug] *****
ok: [13.127.244.80] => {
  "msg": "root"
}

PLAY RECAP *****
13.127.244.80      : ok=2    changed=0    unreachable=0    failed=0
```

ansible-playbook vault_play.yml --ask-vault-pass

```
[root@ip-172-31-27-155 Ansible]# ansible-vault view vault_variable.yml
Vault password:
---
password:
- admin
- root
- user
```

Cont.

```
[root@ip-172-31-27-155 Ansible]# ansible-vault encrypt vault_encrypt_decrypt.yml
New Vault password:
Confirm New Vault password:
Encryption successful
[root@ip-172-31-27-155 Ansible]# vim vault_encrypt_decrypt.yml
[root@ip-172-31-27-155 Ansible]# ansible-vault decrypt vault_encrypt_decrypt.yml
Vault password:
Decryption successful
```

- Rekeying Encrypted Files - to change your password on a vault-encrypted file or files
ansible-vault rekey vault_variable.yml
- 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 vault_encrypt_decrypt.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 vault_encrypt_decrypt.yml