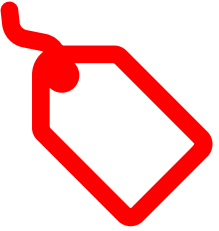


Linux RedHat Certified Engineer (RHCE - EX294)

**Manage Variables, Facts and
Additional Features**

Tags

- Tags are the reference or aliases to a task
- Instead of running an entire Ansible playbook, use tags to target a specific tasks you need to run



```
# vim httpbytags.yml
---
- name: Setup Apache server
  hosts: localhost
  tasks:
    - name: Install httpd
      yum:
        name: httpd
        state: present
        tags: i-httpd

    - name: Start httpd
      service:
        name: httpd
        state: started
        tags: s-httpd
```

- To list all tags in a playbook
`# ansible-playbook httpbytags.yml --list-tags`
- To run a task using tag
`# ansible-playbook httpbytags.yml -t i-httpd`
- To skip a task using tag
`# ansible-playbook httpbytags.yml --skip-tags i-httpd`
- Wait a second...
- We can use “tasks option” to start a playbook at a specific task
`# ansible-playbook yamfile.yml --start-at-task 'Task name'`
`# ansible-playbook http.yml --start-at-task 'Intall httpd'`

```
# ansible-playbook httpbytags.yml -t i-httpd
```

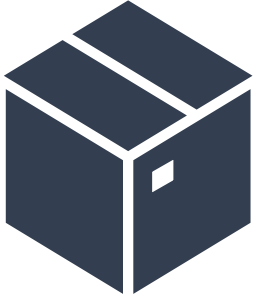
```
# ansible-playbook httpbytags.yml -t s-httpd
```

Variables

- Variables are like containers that hold the defined value which can be used repetitively

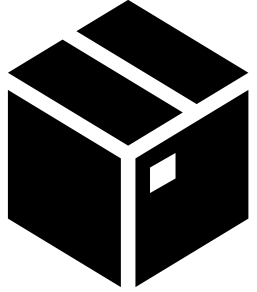
IMPORTANT Things to Remember about Variables!

- Name can include letters, numbers and underscore
- Name should always start with a letter
- Cannot have a spaces, dots (.) or hypen (-) in variable name
- Variables can be defined inside of inventory files as well



Variables

Example



1

```
---
- name: Install some package
  hosts: all
  vars:
    sespackage: sesquipedalianism

  tasks:
  - name: Package install
    yum:
      name: "{{ sespackage }}"
      state: present

  - name: Start service
    service:
      name: "{{ sespackage }}"
      state: started
```

```
---
- name: Package installation
  hosts: all
  vars:
    pack: httpd

  tasks:
  - name: Install package
    yum:
      name: "{{ pack }}"
      state: present

  - name: Start service
    service:
      name: "{{ pack }}"
      state: started
```

Variables

Examples

2

```
---
- name: Copy file to remote clients
  hosts: all
  vars:
    srcfile: /home/iafzal/somefile
  tasks:

- name: Copying file
  become: true
  copy:
    src: "{{ srcfile }}"
    dest: /tmp
    owner: iafzal
    group: iafzal
    mode: 0644
```

3

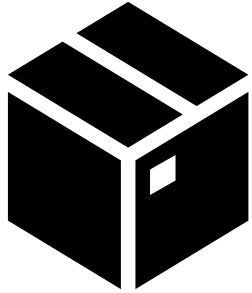
```
---
- name: Create a file
  hosts: localhost
  vars:
    file_name: kramer

  tasks:
  - name: Create file in /tmp
    file:
      state: touch
      path: /tmp"{{ file_name }}" .txt
```

4

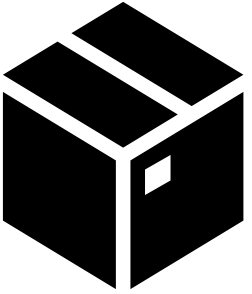
```
---
- name: Print Hello world
  hosts: all
  vars:
    say: Hello World!

  tasks:
  - name: Ansible Variable Basic Usage
    debug:
      msg: "{{ say }}"
```



Variables in Inventory File

Example

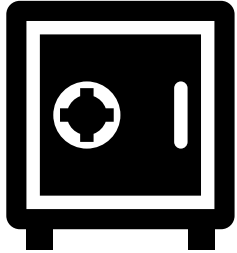


```
[webservers]
client1.xyz.com
client2.xyz.com

[abc:vars]
fooserver=foo.abc.example.com
ntpserver=ntp.abc.example.com
proxyserver=proxy.abc.example.com

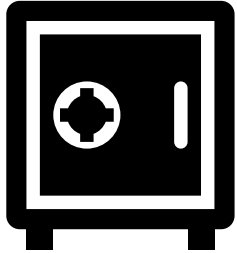
server1 ansible_host=201.0.113.111
server2 ansible_host=201.0.113.112
server3 ansible_host=201.0.113.113
server4 ansible_host=abc.example.com
```

Ansible Vault



- Ansible can automate tasks for teams such as:
 - Hardware
 - Operating systems
 - Virtualization
 - Database or Storage
 - Applications/Software etc.
- Oftentimes you have to share Ansible code with these groups over the network and anything you share over network has a risk to end up in wrong hands
- It is best practice to use **Ansible vault** feature which will password protect your code

Ansible Vault



- yaml file with ansible-vault

```
# ansible-vault create httpbyvault.yml

---
- name: Install httpd package
  hosts: localhost

  tasks:
  - name: Install package
    yum
      name: httpd
      state: present
```

```
# ansible-playbook httpbyvault.yml = ERROR!
```

- To run a vaulted yaml file
`ansible-playbook httpbyvault.yml --ask-vault-pass`
- To view a vaulted yaml file
`ansible-vault view httpbyvault.yml`
- To edit an existing vaulted yaml file
`ansible-vault edit httpbyvault.yml`

- To get a list of options
`ansible-vault --help`
- `httpbyvault.yml` file was created with ansible-vault, what about existing files???
- # `vim testbyvault.yml`
- # `ansible-vault encrypt testbyvault.yml`

Encrypt Strings within a Playbook



- Strings/words can be encrypted within a playbook

```
# ansible-vault encrypt_string httpd
# ansible-vault create/encrypt outputbystring.yml

---
- name: Test encrypted output
  hosts: localhost
  vars:
    secret: !vault |
      $ANSIBLE_VAULT;1.1;AES256
      34343066363535633538313838383363616161633163326638303737383537316563633865653166
      3237613536323465326636623465343866646332633362630a636533303762636630313830303531
      66613766666130346135623436356138303262656162353330623535346135613566333439663230
      3265333738653532310a353632373565386138373832656336393861323030643263323535326230
      3164

  tasks:
    - name: Print encrypted string
      debug:
        var: secret
```