

Using Advanced Ansible Features (Ansible vault and templates)

Jinja2 Template if Statement

Example:

```
---
-
hosts: all
gather_facts: False
tasks:
  - name: J2 if statement
    debug:
      msg: >
        {% comment line -%#}
        {% if inventory_hostname in groups['webserver'] -%}
          This is webserver node
        {% elif inventory_hostname in groups['prod1'] -%}
          This is prod1 node
        {% else -%}
          This is prod2 node
        {% endif -%}
...

```

Jinja2 Template for Statement

Example:

```
---  
-  
hosts: all  
gather_facts: True  
tasks:  
  - name: j2 for statement  
    debug: >  
      {# Comment line -#}  
      {% for host in ansible_play_batch -%}  
        {{ hostvars[host]['ansible_hostname'] }}  
      {% endfor -%}  
...
```

Task: Create a template 'hostname.j2' to generate list of hostnames for all managed nodes.

- Create a playbook named **hostname.yml** to deploy **hostname.j2** to **prod1** nodes to **/root/host_names** file.

Creating Jinja2 Template file:

```
vim hostname.j2
{% for host in ansible_play_batch -%}
    {{ hostvars[host]['ansible_facts']['hostname'] }}
{% endfor -%}
:wq
```

```
---
-
  hosts: all
  become: True
  gather_facts: True
  tasks:
    - name: Deploying template
      template:
        src: hostname.j2
        dest: /root/host_names
      when: inventory_hostname in groups['prod1']
...

```

Task. Create a template 'disk.j2' to generate list of size of sda4,sda5 and sda6 partitions for all managed nodes.

- Create a playbook named **disk.yml** to deploy **disk.j2** to **prod2** node to **/root/disk_size** file.
- Hostname must be displayed to differentiate.

Creating Jinja2 Template file:

```
vim disk.j2
{% for host in ansible_play_batch -%}
Size of sda4 on {{ host }} = "{{ hostvars[host]['ansible_facts']['devices']['sda']['partitions']['sda4']['size'] | default('NA') }}"
Size of sda5 on {{ host }} = "{{ hostvars[host]['ansible_facts']['devices']['sda']['partitions']['sda5']['size'] | default('NA') }}"
Size of sda6 on {{ host }} = "{{ hostvars[host]['ansible_facts']['devices']['sda']['partitions']['sda6']['size'] | default('NA') }}"
{% endfor -%}
:wq
```

```
---
-
  hosts: all
  become: True
  gather_facts: True
  tasks:
    - name: Deploying template
      template:
        src: disk.j2
        dest: /root/disk_size
      when: inventory_hostname in groups['prod2']
...

```

Task. Create a playbook named 'hosts.yml' to configure local DNS functionality on all managed hosts using 'hosts' file.

- Use hosts.j2 as template file and hosts file contents should display “**IP_Address fqdn hostname**”
- Default entries or local host should also be included.

Creating Jinja2 Template file:

```
vim hosts.j2
{% for host in ansible_play_batch -%}
{% if 'ansible_enp0s3' in hostvars[host] -%}
    {{ hostvars[host]['ansible_facts']['enp0s3']['ipv4']['address'] }}      {{ host }}.example.com    {{ host }}
{% endif -%}
{% endfor -%}
:wq
```

```
---
-
  hosts: all
  become: True
  gather_facts: True
  tasks:
    - name: Deploying template
      template:
        src: hosts.j2
        dest: /etc/hosts
...
```

Ansible Vault

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

ansible-vault command line is used for this purpose.

File Level Encryption:

Any structured data file used in ansible can be encrypted e.g. **ansible_facts, host_vars, group_vars, vars_files ,playbooks** and other files.

Variable Level Encryption:

Ansible also supports encryption of single values. Passwords, strings and other variables can be encrypted and then used in playbooks.

Task. Using ansible-vault, Create an encrypted file 'secret.yml' containing below information

- Set Vault password as **password**.

```
# Information to be added in file
---
password1: rhcsaex200
password2: rhceex300
...
```

Command:

ansible-vault create secret.yml

Task: Edit the encrypted file 'secret.yml' to add below information to file

- File should be edited without decrypting it.

```
# Information to be added in file
```

```
---
```

```
password3: rhceex294
```

```
...
```

Command:

```
ansible-vault edit secret.yml
```

Task. Using ansible-vault, change the vault password for 'secret.yml' file.

- Set New Vault password as **newpassword**.

Command:
ansible-vault rekey secret.yml