



# Using Tags in Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

## Session - 20 Agenda:

1. Ansible Tags

### Ansible Tags:

If you have a large playbook, it may be useful to run only specific parts of it instead of running the entire playbook. You can do this with Ansible tags. Using tags to execute or skip selected tasks is a two-step process:

1. Add tags to your tasks, either individually or with tag inheritance from a block, play, role, or import.
2. Select or skip tags when you run your playbook.

Let's understand it with the help of a playbook to configure the Apache web server on the managed nodes (this playbook has several tasks in it).

\$ vim apache.yml

```
---
- name: Apache Web Server Configuration Playbook
  hosts: node1
  become: true
  tasks:
    - name: Installing Apache Latest Packages
      yum:
        name: httpd
        state: latest
    - name: Starting & Enabling Apache HTTPD Service
      service:
        name: httpd
        state: started
        enabled: true
    - name: Setting Up Apache Web Server
      copy:
        content: "Nehra Classes Are Awesome."
        dest: /var/www/html/index.html
    - name: Restarting Apache HTTPD Service
      service:
        name: httpd
        state: restarted
    - name: Allowing HTTP Traffic in the Firewall.
      firewalld:
        service: http
        permanent: yes
        state: enabled
    - name: Reload the Firewall Service
      service:
        name: firewalld
        state: reloaded
...

```

Let's check and execute the playbook to configure the Apache Web Server on node1.

\$ ansible-playbook apache.yml

Here you can see that the playbook is returning an error because of missing firewalld module, this module can be manually installed using below command.

\$ ansible-galaxy collection install ansible.posix



# Using Tags in Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Now, again execute the playbook and check.

```
$ ansible-playbook apache.yml
```

This time you can see that the playbook executes successfully and creates the Apache web server on node1. Now, we can execute curl command to check the working of the Apache web server, which will show us the details present on the index.html page.

```
$ curl node1
```

## Using Tags in Ansible Playbooks:

To use the tags in the Ansible playbooks we have to assign the tags to the tasks first which we want to manage using the tags. And this can be done either individually or with tag inheritance from a block, play, role, or import.

Let's assign the tags to all the tasks present in the above playbook.

```
$ vim apache_tags.yml
```

```
---
```

```
- name: Apache Web Server Configuration Playbook With Tags
  hosts: node3
  become: true
  tasks:
    - name: Installing Apache Latest Packages
      yum:
        name: httpd
        state: latest
        tags: pack_tag
    - name: Starting & Enabling Apache HTTPD Service
      service:
        name: httpd
        state: started
        enabled: true
        tags: serv_tag
    - name: Setting Up Apache Web Server
      copy:
        content: "Nehra Classes Are Awesome."
        dest: /var/www/html/index.html
        tags: index_tag
    - name: Restarting Apache HTTPD Service
      service:
        name: httpd
        state: restarted
        tags: restart_tag
    - name: Allowing HTTP Traffic in the Firewall.
      firewalld:
        service: http
        permanent: yes
        state: enabled
        tags: firewall_tag
    - name: Reload the Firewall Service
      service:
        name: firewalld
        state: reloaded
        tags: firewall_tag
  ...
```

**Note:** We can add same tag to multiple tasks as well as multiple tags to the same task using lists format.



# Using Tags in Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

## Select Tags (Executes Selected Tags Only):

Only selected tags are executed and all other remaining tags are ignored when we mention the tag name while executing the ansible playbook.

Let's check the httpd package availability first the managed node using the ansible ad-hoc command.

```
$ ansible node3 -m command -a 'rpm -qi httpd'
```

Let's now only execute the first task to install the Apache httpd package and skip all the other tasks in the above playbook.

```
$ ansible-playbook apache_tags.yml --tags="pack_tag"
```

Verify the same using ansible ad-hoc command.

```
$ ansible node3 -m command -a 'rpm -qi httpd'
```

Let's check the httpd service status on the managed node using ansible ad-hoc command.

```
$ ansible node3 -m command -a 'systemctl status httpd'
```

Let's now only execute the first & second task only and skip all the other tasks in the above playbook.

```
$ ansible-playbook apache_tags.yml --tags="pack_tag,serv_tag"
```

Verify the same using ansible ad-hoc command.

```
$ ansible node3 -m command -a 'systemctl status httpd'
```

---

## Skip Tags (Skips Selected Tags Only):

Only selected tags are skipped and all other remaining tags are executed when we mention the tag name while executing the ansible playbook.

Suppose we want to skip the first two tasks (package installation and starting the service).

```
$ ansible-playbook apache_tags.yml --skip-tags="pack_tag,serv_tag"
```

Let's skip both the firewall related tasks.

```
$ ansible-playbook apache_tags.yml --skip-tags="firewall_tag"
```

---

## Always Tag (Mandatory Tag):

This tag executes the task every time irrespective of what have we mentioned in the command while executing the ansible playbook.

e.g. Suppose we want to restart the apache service every time when we execute the ansible playbook, we can mention always keyword in the tags section against this task as shown below.

```
- name: Restarting Apache HTTPD Service
```

```
  service:
```

```
    name: httpd
```

```
    state: restarted
```

```
  tags: always
```

Let's execute the playbook and verify the same.

```
$ ansible-playbook apache_tags.yml --tags="pack_tag"
```

Here you can see that the above command is executing two tasks one for the package installation and other to restart the httpd service. This is happening because of the always tag, which is there in the restart httpd service section.



# Using Tags in Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

## Assigning Multiple Tags:

We can assign multiple tags to the same task using lists format.

```
- name: Restarting Apache HTTPD Service
```

```
  service:
```

```
    name: httpd
```

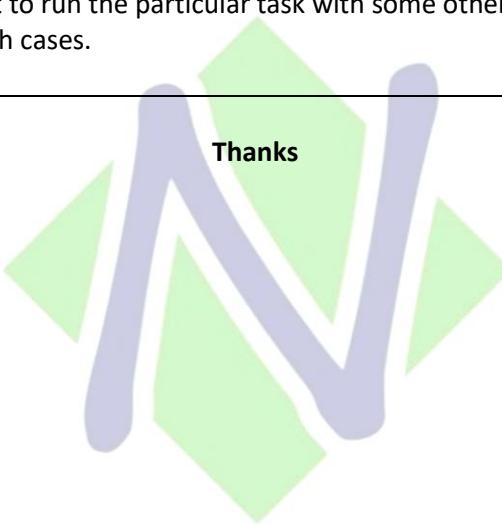
```
    state: restarted
```

```
  tags:
```

```
    - restart_tag
```

```
    - service_tag
```

This is useful in case if we want to run the particular task with some other task(s), so we use different tag names for the same task in such cases.



Thanks

The logo consists of a stylized 'N' shape composed of several overlapping green and purple triangles. The word 'Thanks' is centered above the 'N' in a small, black, sans-serif font.

Nehra Classes  
Igniting The Minds