

Ansible Roles

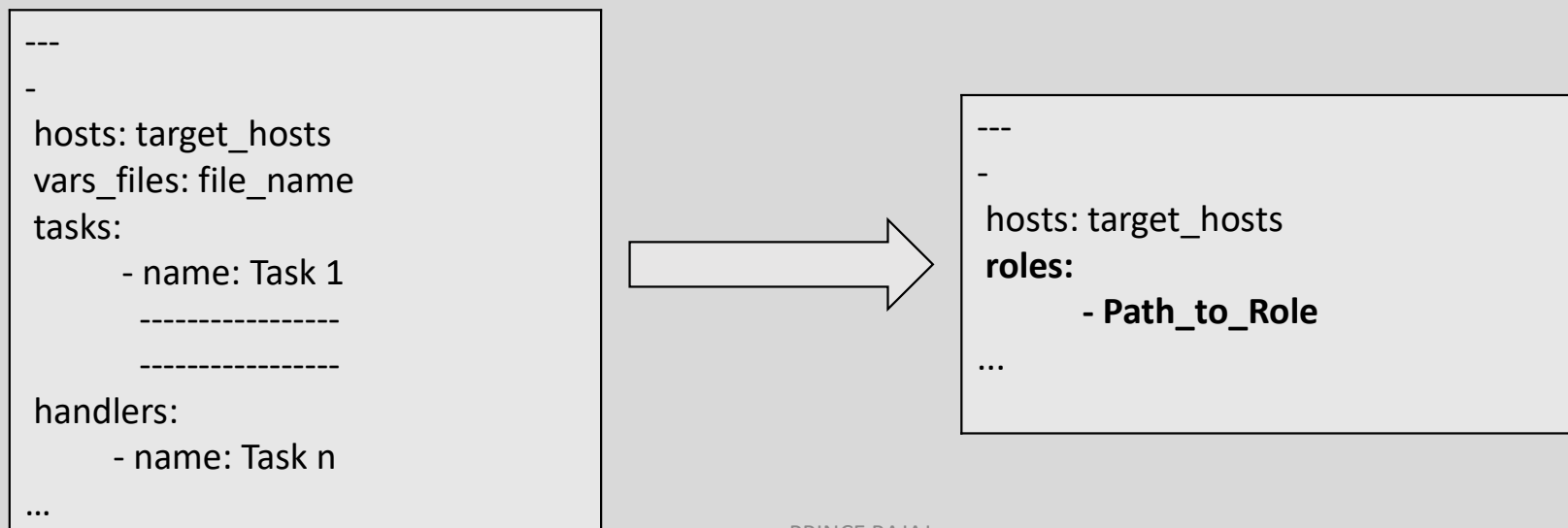
- **Create roles**
- **Download roles from an Ansible Galaxy and use them**

Introducing Ansible Roles

Ansible Role is standard directory architecture where contents of playbook like tasks,handlers,variables,templates and metadata are kept for organization purpose.

Ansible Role is nothing but playbook contents(tasks and related files/components) distributed over standard directory architecture with a specific name called Role Name. Then this role is included in playbook which automatically loads all the tasks ,files and variables in the playbook.

Creating roles makes it easy to reuse contents in different playbooks for same task(s) and to distribute and share contents with others. Also using Roles, Large projects can be divided into multiple roles and can be well organized in directory structures.



Role's Directory Structure & Creating Role

Under top level role's directory, below mentioned directories are used to contain different type of content.

- **tasks:** Contains the main list of tasks (in **main.yml** file) to be executed by the role.
- **handlers:** Contains handlers(in **main.yml** file), which may be used by this role.
- **defaults:** Default variables for the role.
- **vars:** Other variables for the role .
- **files:** Contains files which are to be used in this role.
- **templates:** Contains templates which can be deployed via this role.
- **meta:** Defines some meta data for this role.

We can create Role's directory structure using **ansible-galaxy init role_name** command (In current directory) .

Ansible looks for **roles** in directories specified by **roles_path** in **ansible.cfg** file and in **roles/** directory relative to directory where playbook is present.

Task. Create a role with name 'webserver' under roles directory to configure 'webserver'.

- Install latest version of **httpd** and make sure service is started and enabled.
- Configure firewall to accept inbound traffic for **http** and **https** services and firewall settings must be persistent.
- Create template file with name **index.j2** in **templates** directory to display message **Welcome to webserver configured on "HOST_NAME" and "IP_ADDRESS"**.
- Deploy this template to **index.html** file in document root directory.
- Make sure correct SELinux label is set on document root directory.
- Use this role in playbook **webserver.yml** to configure **webservers** nodes.

```
---  
-  
  hosts: webservers  
  become: True  
  gather_facts: true  
  roles:  
    - webserver  
...
```

webserver role contents are shown on next slides.

webserver/tasks/main.yml

- name: Installing latest version of httpd

yum:

name: httpd

state: latest

- name: Starting and enabling webserver

service:

name: httpd

state: started

enabled: yes

- name: Configuring firewall

firewalld:

service: "{{ item }}"

state: enabled

permanent: yes

loop:

- http

- https

notify: Reload firewall

- name: Deploying template

template:

src: index.j2

dest: /var/www/html/index.html

notify: Restart httpd

...

/webserver/handlers/main.yml

- name: **Reload firewall**

service:

name: firewalld

state: reloaded

- name: **Restart httpd**

service:

name: httpd

state: restarted

...

/webserver/templates/

vim index.j2

Welcome to webserver on {{ ansible_facts['hostname'] }} and {{ ansible_facts['enp0s3']['ipv4']['address'] }}

:wq

Ansible Galaxy and Downloading Roles

Ansible Galaxy is free website where users can share roles and from where users can download roles.

You can also share our Role's on this web site. We need to authenticate using Github account and then it is possible to import roles to websites.

We can download roles from Ansible Galaxy and from GitHub using **ansible-galaxy install** command line. The command line tool by default communicates with the Galaxy website API using the server address **<https://galaxy.ansible.com>**.

Downloading/Installing Roles: We can download roles from Ansible Galaxy using below command.

```
ansible-galaxy install username.rolename
```

By default role is installed in first writable directory **`~/.ansible/roles:/usr/share/ansible/roles:/etc/ansible/roles`**.

This behavior can be overwritten by setting **roles_path** in **ansible.cfg** file.

To download role to specific directory :

```
ansible-galaxy install --roles-path DIR_PATH username.rolename
```


Downloading Multiple Roles

To download multiple roles, We can use **YAML** file defining list of roles to be installed/downloaded.

Below details can be provided to specify role to be downloaded using this file.

src - Source of role in form of **username.role_name** if downloading from Ansible Galaxy otherwise provide URL. This is required attribute.

scm - **git** and **hg** are supported, default is **git**.

version -Version of roles to be downloaded.

name- Download role to a specific name otherwise default role will be taken.

Example:

vim requirements.yml

```
- src: https://github.com/bennojoy/nginx
  version: master
  name: my_nginx
```

To install role:

ansible-galaxy install -r requirements.yml (Role will be downloaded to path specified by roles_path)

Introducing Linux System Roles

Collection of Ansible Roles used to manage and configure common components/subsystems of Linux. Examples of some subsystems are :

- **network**
- **timesync**
- **storage**
- **selinux and more..**

We will discuss about **timesync** role and will use same to configure NTP server for managed nodes.

System roles can be availed through package **rhel-system-roles**.

To install Linux System Roles, Install package **rhel-system-roles**.

`dnf install rhel-system-roles`

For Example playbooks, Check on path- **/usr/share/doc/rhel-system-roles**

Ansible Roles are available on path - **/usr/share/ansible/roles/**

Task. Create a playbook 'chrony.yml' to configure time source for managed nodes.

- Use **timesync** System role to configure this. Also set given time zone.
- Use Ansible Control Node as NTP Server ,so use IP Address of NTP Server as "**192.168.99.1**".
- Using Ansible Ad-Hoc commands verify if this is properly configured.

```
---
-
  hosts: all
  become: True
  gather_facts: True
  vars:
    timesync_ntp_servers:
      - hostname: 192.168.99.1
        iburst: yes
    timezone: Europe/Brussels
  tasks:
    - name: Set timezone
      timezone:
        name: "{{ timezone }}"
  roles:
    - /usr/share/ansible/roles/rhel-system-roles.timesync
...
```

Steps to be done on Ansible Control Node:

```
vim /etc/chrony.conf
allow 192.168.99.0/24
:wq
systemctl restart chronyd
firewall-cmd --add-service=ntp --permanent
firewall-cmd --reload
```