Ansible Roles with Examples

With more complexity in functionality, it becomes difficult to manage everything in one ansible playbook file. Sharing code among teams become difficult. Ansible Role helps solve these problems. Ansible role is an independent component which allows reuse of common configuration steps. Ansible role has to be used within playbook. Ansible role is a set of tasks to configure a host to serve a certain purpose like configuring a service. Roles are defined using YAML files with a predefined directory structure.

What is Ansible roles?

1. Ansible roles are consists of many playbooks, which is similar to modules in puppet and cook books in chef. We term the same in ansible as roles.
2. Roles are a way to group multiple tasks together into one container to do the automation in very effective manner with clean directory structures.
3. Roles are set of tasks and additional files for a certain role which allow you to break up the configurations.
4. It can be easily reuse the codes by anyone if the role is suitable to someone.
5. It can be easily modify and will reduce the syntax errors.

Below is a sample playbook codes to deploy Apache web server. Let's convert this playbook codes into Ansible roles.

```
---
- hosts: all
  become: true
  tasks:
  - name: Install httpd Package
    yum: name=httpd update_cache=yes  state=latest
  - name: Copy httpd configuration file
    copy: src=httpd.conf dest=/etc/httpd/conf/httpd.conf
  - name: Copy index.html file
    copy: src=index.html dest=/var/www/html
    notify:
    - restart apache
  - name: Start and Enable httpd service
    service: name=httpd state=restarted enabled=yes
  handlers:
  - name: restart apache
    service: name=httpd state=restarted
```

How do we create Ansible Roles?

To create a Ansible roles, use ansible-galaxy command which has the templates to create it. This will default directories and do the modifications else we need to create each directories and files manually.

Let's take an example to create a role for Apache Web server.

```
# mkdir /etc/ansible/rolesDemo
# ansible-galaxy init /etc/ansible/rolesDemo/apache
- apache was created successfully

[ansible@ip-172-13-17-90 ~]#
```

where, ansible-glaxy is the command to create the roles using the templates.

init is to initiliaze the role.

apache is the name of role.

List out the directory created under /etc/ansible/rolesDemo.

Note : if tree command is not working install tree package using package manager.

```
[ansible@ip-172-13-17-90 ~]# tree /etc/ansible/rolesDemo/apache/
/etc/ansible/rolesDemo/apache/
|-- README.md
|-- defaults
|   `-- main.yml
|-- files
|-- handlers
|   `-- main.yml
|-- meta
|   `-- main.yml
|-- tasks
|   `-- main.yml
|-- templates
|-- tests
|   |-- inventory
|   `-- test.yml
`-- vars
    `-- main.yml
8 directories, 8 files
[ansible@ip-172-13-17-90 ~]#
```

We have got the clean directory structure with the ansible-galaxy command. Each directory must contain a main.yml file, which contains the relevant content.

Directory Structure:

A role directory structure contains directories: defaults, vars, tasks, files, templates, meta, handlers. Each directory must contain a main.yml file which contains relevant content. Let's look little closer to each directory.

1. **defaults**: contains default variables for the role. Variables in default have the lowest priority so they are easy to override.
2. **vars**: contains variables for the role. Variables in vars have higher priority than variables in defaults directory.
3. **tasks**: contains the main list of steps to be executed by the role.
4. **files**: contains files which we want to be copied to the remote host. We don't need to specify a path of resources stored in this directory.
5. **templates**: contains file template which supports modifications from the role. We use the Jinja2 templating language for creating templates.
6. **meta**: contains metadata of role like an author, support platforms, dependencies.
7. **handlers**: contains handlers which can be invoked by "notify" directives and are associated with service.

First, move on to the Ansible roles directory and start editing the yml files.

```
cd /etc/ansible/rolesDemo/apache
```

1. **defaults**

Edit main.yml available in the defaults folder to define the default variables.

```
base_httpd_listen_port: 80
```

2. **Tasks**

Edit main.yml available in the tasks folder to define the tasks to be executed.

```
vi tasks/main.yml
---
- name: Install httpd Package
  yum: name=httpd state=latest
- name: Copy httpd configuration file
  template: src=httpd.conf.j2 dest=/etc/httpd/conf/httpd.conf
- name: Copy index.html file
  copy: src=index.html dest=/var/www/html
  notify:
  - restart apache
```

3. Templates

Copy the required files (httpd.conf.j2) to the templates directory.

4. Files

Copy the required files (index.html) to the files directory.

5. Handlers

Edit handlers main.yml to restart the server when there is a change. Because we have already defined it in the tasks with notify option. Use the same name "restart apache" within the main.yml file as below.

```
-   name: restart apache
    service: name=httpd state=restarted
```

6. Meta

Edit meta main.yml to add the information about the roles like author, descriptions, license, platforms supported.

```
[ansible@ip-172-13-17-90 ]# cat meta/main.yml
galaxy_info:
  author: MithunTechnologies.net
  description: Apache Webserver Role
  company: MithunTechnologies.net
```

We have got all the required files for Apache roles. Let's apply this role into the ansible playbook "site.yml" as below to deploy it on the client nodes.

```
cat /etc/ansible/rolesDemo/site.yml ---

 - hosts:
   appServers
 - roles:
   - apache
```

We have defined this changes should be run only on appServers, you can also use "all" if need. Specify the role name as "apache", also if you have created multiple roles, you can use the below format to add it.
- apache
- common

Let's verify for syntax errors:

```
[ansible@ip-172-13-17-90 ]# ansible-playbook /etc/ansible/rolesDemo/site.yml --syntax-check

playbook: /etc/ansible/rolesDemo/site.yml
```

If No errors found. Let move on to deploy the roles.

```
[ansible@ip-172-13-17-90 ]# ansible-playbook /etc/ansible/rolesDemo/site.yml
```

Source Code can be downloaded from git hub here