## Ansible Roles

* Its all about reusability
* Using others automation efforts in ansible is simplified using roles
* Any user can share the work by converting playbook to ansible role and submit to ansible community. Ansible community which shares these roles is **Ansible Galaxy**

**Ansible Galaxy**

* Galaxy is a hub for finding and sharing Ansible content
* Galaxy provides pre-packaged units of work known to Ansible as Roles
* Roles can be dropped into Ansible PlayBooks and immediately put to work. You’ll find roles for provisioning infrastructure, deploying applications, and all of the tasks you do every day. The new Collection format provides a comprehensive package of automation that may include multiple playbooks, roles, modules, and plugins.
* As an example, lets install java using Ansible Galaxy role.
* Download the Ansible Role by executing ansible-galaxy install geerlingguy.java
* Now to install java lets write a playbook

---

- hosts: all

become: yes

roles:

- role: geerlingguy.java

### Creating Ansible Roles

* Sample folder structure
  + playbooks
    - roles
      * rolename
* Login into the ACS(Ansible Control Server) and create a new directory called roles and execute the following command

# ansible-galaxy init LAMP\_Stack

[ansible@ip-172-31-23-242 Roles]$ tree

.

├── LAMP\_Stack

│   ├── defaults

│   │   └── main.yml

│   ├── files

│   ├── handlers

│   │   └── main.yml

│   ├── meta

│   │   └── main.yml

│   ├── README.md

│   ├── tasks

│   │   └── main.yml

│   ├── templates

│   ├── tests

│   │   ├── inventory

│   │   └── test.yml

│   └── vars

│   └── main.yml

└── runsetup.yaml

9 directories, 9 files

#runsetup.yaml

---

- hosts: all

become: yes

roles:

- LAMP\_Stack

# vars/main.yml

---

# vars file for apache-webserver

apache\_pack\_deb: apache2

package\_name: httpd

#tasks/main.yml

---

# tasks file for LAMP\_Stack

- name: Install httpd Package

yum: name={{package\_name}} state=latest

notify:

- restart apache

# handlers/main.yml

---

# handlers file for LAMP\_Stack

- name: restart apache

service: name=httpd state=restarted

Finally execute the runsetup.yaml playbook with below command

ansible-playbook runsetup.yaml

Once the playbook is executed successfully then Apache is installed through Role

Then we can publish the Role into Ansible-Galaxy to use it by others

**Publishing Ansible Roles**

2. Go to role directory using:

cd LAMP\_Stack

3. Initialise for Git:

git init

4. Add files to Git using the command:

git add .

6. Commit the changes to Git:

git commit -m “Initial commit for <role-name>”

7. Set your public repository origin:

Create Repository in Github (Don’t select to initialize with README file)

git remote add origin <https://github.com/skatta3/TestRole.git>

8. Push the changes to the repository using the command:

git push -u origin master

**Publishing an Ansible Role**

Go to [https://galaxy.ansible.com](https://galaxy.ansible.com/)

Log in to Galaxy using GitHub credentials.

You can use following command in your Ansible controller to publish roles.

ansible-galaxy role import <github\_user> <github\_repo> --token XXXXXXXXXXXXXXXXXXX

ansible-galaxy role import skattagmail myrole --token 561d28399d1d455fdd7cc578d52f7a4ef025e2d8

You need to generate API token in Ansible Galaxy under Collections. Import is basically publish.