Using Ansible Galaxy to Find and Use Community Roles in Your Playbooks

Table of Contents

Introduction

Ansible Galaxy is a platform that hosts thousands of community-contributed roles and collections. These roles simplify the process of writing playbooks for common tasks like server setup, configuration management, and application deployment.

> **Note: What is Ansible Galaxy?**
>
> ```
>    Ansible Galaxy is a public repository of Ansible roles and collections
> contributed by the Ansible community. It provides a platform where users can
> share their roles, and others can easily download and use them in their
> playbooks. Galaxy roles are reusable and follow best practices for writing
> efficient, modular Ansible code.
> ```

In this guide, we'll explore how to find and use Ansible Galaxy roles to automate the installation of Chocolatey, a popular package manager for Windows.

Problem Statement

Writing Ansible roles for every task from scratch can be repetitive and time-consuming. Ansible Galaxy provides pre-built roles created by the community to address various use cases. By leveraging these roles, you can streamline your automation tasks and reuse well-maintained code.

Prerequisites

Completion of all previous lab guides (up to Lab Guide-06) is required before proceeding with Lab Guide-07.

Software Requirements

- **Ansible 2.9+**: Installed on your control node (WSL for Windows users).
- **WSL (Windows Subsystem for Linux)**: If using Windows as your control node.
- **Access to Ansible Galaxy**: Internet connection to download roles.

Step-by-Step Guide to Using Ansible Galaxy

Step 1: Installing a Role from Ansible Galaxy

1. **Search for a Role on Galaxy**:

   - Navigate to the Ansible Galaxy website here and search for a role that fits your needs. For
     example, search for a Chocolatey role by typing "chocolatey" in the search bar.

2. **Install the Role**:

   - Once you find the role you need, you can install it using the `ansible-galaxy` command. For
     example, to install the Chocolatey role, run:

   ```
   ansible-galaxy role install oatakan.chocolatey_package_localize
   ```

   ```
   user1@Swayaan:~$ ansible-galaxy role install oatakan.chocolatey_package_localize
   Starting galaxy role install process
   - downloading role 'chocolatey_package_localize', owned by oatakan
   - downloading role from https://github.com/oatakan/ansible-role-chocolatey-package-localize/archive/master.tar.gz
   - extracting oatakan.chocolatey_package_localize to /home/user1/.ansible/roles/oatakan.chocolatey_package_localize
   - oatakan.chocolatey_package_localize (master) was installed successfully
   ```

   This will install the role in the default roles directory (`~/.ansible/roles`).

3. **Verify Installation**:

   - To verify that the role has been installed, you can list all installed roles:

   ```
   ansible-galaxy list
   ```

   ```
   user1@Swayaan:~$ ansible-galaxy list
   # /home/user1/.ansible/roles
   - geerlingguy.nginx, 3.2.0
   - geerlingguy.git, 3.0.1
   - oatakan.chocolatey_package_localize, master
   # /etc/ansible/roles
   [WARNING]: - the configured path /usr/share/ansible/roles does not exist.
   ```

   You should see the newly installed role (`oatakan.chocolatey_package_localize`) in the list.

Step 2: Using the Installed Role in a Playbook

Now that the role is installed, you can use it in your playbooks.

1. **Create a new playbook**:

```
nano install_chocolatey.yml
```

2. **Add the following content to the file**:

```yaml
---
- name: Install Chocolatey using Ansible Galaxy Role
  hosts: windows
  tasks:
    - name: Install Chocolatey
      include_role:
        name: oatakan.chocolatey_package_localize
```

```
---
- name: Install Chocolatey using Ansible Galaxy Role
  hosts: windows
  tasks:
    - name: Install Chocolatey
      include_role:
        name: oatakan.chocolatey_package_localize
```

This playbook will use the Chocolatey role to install the Chocolatey package manager on the target Windows machine.

Step 3: Execute Playbook

Once the playbook is ready, you can run it using the `ansible-playbook` command.

```
ansible-playbook install_chocolatey.yml
```

```
TASK [oatakan.chocolatey_package_localize : unzip package file] *********************************
changed: [192.168.0.102]

TASK [oatakan.chocolatey_package_localize : delete extraneous files/folders] *******************
changed: [192.168.0.102] => (item=[Content_Types].xml)
changed: [192.168.0.102] => (item=_rels)
changed: [192.168.0.102] => (item=package)

TASK [oatakan.chocolatey_package_localize : ensure binary file exist locally (<2.9)] ***********
skipping: [192.168.0.102]

TASK [oatakan.chocolatey_package_localize : ensure binary file exist locally] ******************
changed: [192.168.0.102]

TASK [oatakan.chocolatey_package_localize : add file path] *************************************
changed: [192.168.0.102]

TASK [oatakan.chocolatey_package_localize : remove urls] **************************************
changed: [192.168.0.102]

TASK [oatakan.chocolatey_package_localize : create package] ***********************************
changed: [192.168.0.102]

TASK [oatakan.chocolatey_package_localize : ensure temp directory is removed] *****************
changed: [192.168.0.102]

PLAY RECAP ***********************************************************************************
192.168.0.102              : ok=18    changed=12   unreachable=0    failed=0    skipped=5    rescued=0    ignored=0
```

Verifying the Deployment

Once the playbook has been executed, you can verify that Chocolatey is installed by opening a PowerShell prompt on the target Windows machine and running:

```
choco --version
```

```
PS C:\Users\ansible_user> choco --version
2.2.2
PS C:\Users\ansible_user> |
```

You should see the version number of Chocolatey if the installation was successful.

Supported References

- Ansible Galaxy Documentation
- Ansible Galaxy
- Ansible Galaxy Chocolatey