🧪Lab 16 — Using Ansible Galaxy (Installing & Managing Community Roles)

Author: *Sandeep Kumar Sharma*

🕑Learning Objectives

In this lab, you will learn: - What Ansible Galaxy is - Why Galaxy roles are useful - How to search, install, and manage community roles - How to use Galaxy roles inside your playbooks - How to customize Galaxy roles with your own variables

---

👙Learning Outcomes

After completing this lab, you will: - Understand how to leverage pre-built roles - Install Galaxy roles locally - Use Galaxy roles in real projects - Override default Galaxy role variables safely - Maintain a clean dependency workflow

---

🧰What Is Ansible Galaxy?

Ansible Galaxy is the official repository for Ansible community roles and collections.
It allows you to: - Download ready-made roles - Reuse automation created by the community - Save time by avoiding writing everything from scratch

Galaxy roles are stored at:
https://galaxy.ansible.com

🧱When to Use Galaxy Roles

Galaxy roles are helpful when: - You need a common installation (Nginx, Docker, MySQL, Node.js) - You want standardized best-practice role structures - You want to save development time - You want to extend your own playbooks quickly

However:
Only use trusted roles from verified creators.

🏛️SECTION A — Installing a Galaxy Role

Example: Install a community Nginx role.

Run:

```
ansible-galaxy install geerlingguy.nginx
```

This installs the role under:

```
~/.ansible/roles/geerlingguy.nginx
```

You can verify installed roles:

```
ansible-galaxy list
```

🏛️SECTION B — Using Galaxy Role Inside a Playbook

Create a file:

```
nano use-nginx-role.yml
```

Add:

```
---
- name: Use Nginx Galaxy Role
  hosts: dev
  become: yes

  roles:
    - geerlingguy.nginx
```

Run:

```
ansible-playbook use-nginx-role.yml
```

This installs Nginx using best practices from the Galaxy role.

🏛️ SECTION C — Overriding Role Variables

Every Galaxy role includes default variables.
You can override them in your playbook.

Example playbook with custom config:

```
nano nginx-custom.yml
```

Add:

```
---
- name: Customize Nginx
  hosts: dev
  become: yes

  vars:
    nginx_remove_default_vhost: true
    nginx_vhosts:
      - listen: "80"
        server_name: "example.com"
        root: "/var/www/html"

  roles:
    - geerlingguy.nginx
```

Run:

```
ansible-playbook nginx-custom.yml
```

🏛️SECTION D — Installing Multiple Roles Using Requirements File

Create requirements file:

```
nano requirements.yml
```

Add:

```
---
roles:
  - name: geerlingguy.nginx
  - name: geerlingguy.mysql
  - name: geerlingguy.firewall
```

Install all roles at once:

```
ansible-galaxy install -r requirements.yml
```

This is extremely useful for team projects.

🏛️SECTION E — Removing or Updating Galaxy Roles

Remove:

```
rm -rf ~/.ansible/roles/geerlingguy.nginx
```

Update:

```
ansible-galaxy install geerlingguy.nginx --force
```

🧪Hands-On Checklist

- [ ] Install a Galaxy role

- [ ] List installed roles
- [ ] Use a Galaxy role in a playbook
- [ ] Override role variables
- [ ] Create a requirements.yml and install multiple roles

Lab Summary

This lab explained: - What Ansible Galaxy is - How to install and use community roles - How to override role properties - How to manage multiple roles using requirements files

Author: Sandeep Kumar Sharma