# \*\*Study Material: Day 4 - Ansible Roles and Folder Structure\*\*

---

## \*\*Table of Contents\*\*

1. \*\*Introduction to Ansible Roles\*\*

- What are Ansible Roles?

- Why Use Roles?

2. \*\*Anatomy of an Ansible Role\*\*

- Directory Structure

- Key Components: Tasks, Handlers, Templates, Variables, Defaults, Files, Meta

3. \*\*Creating Your First Ansible Role\*\*

- Example: Setting Up a Web Server

- Step-by-Step Guide

4. \*\*Using Ansible Galaxy\*\*

- What is Ansible Galaxy?

- How to Use Pre-Built Roles

5. \*\*Best Practices for Using Roles\*\*

- Role Reusability

- Role Dependencies

- Role Testing

6. \*\*Conclusion and Next Steps\*\*

---

## \*\*1. Introduction to Ansible Roles\*\*

### \*\*What are Ansible Roles?\*\*

Ansible roles are a way to organize playbooks and related files into a reusable structure. Roles allow you to break down complex automation tasks into smaller, manageable components.

### \*\*Why Use Roles?\*\*

- \*\*Reusability\*\*: Roles can be reused across multiple playbooks and projects.

- \*\*Modularity\*\*: Roles make it easier to manage and share code.

- \*\*Scalability\*\*: Roles help in scaling automation efforts by organizing tasks into logical units.

---

## \*\*2. Anatomy of an Ansible Role\*\*

### \*\*Directory Structure\*\*

A typical Ansible role has the following directory structure:

```

role\_name/

├── tasks

│ └── main.yml

├── handlers

│ └── main.yml

├── templates

│ └── index.html.j2

├── files

│ └── example.conf

├── vars

│ └── main.yml

├── defaults

│ └── main.yml

├── meta

│ └── main.yml

└── README.md

```

### \*\*Key Components\*\*

- \*\*Tasks\*\*: Contains the main list of tasks to be executed by the role.

- \*\*Handlers\*\*: Contains handlers, which are tasks that are triggered by other tasks.

- \*\*Templates\*\*: Contains Jinja2 templates for configuration files.

- \*\*Files\*\*: Contains static files that need to be copied to the managed nodes.

- \*\*Vars\*\*: Contains variables specific to the role.

- \*\*Defaults\*\*: Contains default variables for the role.

- \*\*Meta\*\*: Contains metadata about the role, including dependencies.

---

## \*\*3. Creating Your First Ansible Role\*\*

### \*\*Example: Setting Up a Web Server\*\*

Let's create a role to set up a web server using Apache.

### \*\*Step-by-Step Guide\*\*

1. \*\*Create the Role Structure\*\*:

```bash

ansible-galaxy init web\_server

```

This will create the directory structure for the `web\_server` role.

2. \*\*Define Tasks\*\*:

Edit `web\_server/tasks/main.yml`:

```yaml

---

- name: Install Apache

apt:

name: apache2

state: present

- name: Ensure Apache is running

service:

name: apache2

state: started

enabled: yes

- name: Copy index.html

copy:

src: files/index.html

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

```

3. \*\*Add Static Files\*\*:

Create `web\_server/files/index.html`:

```html

<html>

<body>

<h1>Welcome to the Web Server!</h1>

</body>

</html>

```

4. \*\*Use the Role in a Playbook\*\*:

Create a playbook `site.yml`:

```yaml

---

- hosts: all

roles:

- web\_server

```

5. \*\*Run the Playbook\*\*:

```bash

ansible-playbook -i inventory.ini site.yml

```

---

## \*\*4. Using Ansible Galaxy\*\*

### \*\*What is Ansible Galaxy?\*\*

Ansible Galaxy is a repository of pre-built roles that can be used to speed up automation tasks.

### \*\*How to Use Pre-Built Roles\*\*

1. \*\*Search for Roles\*\*:

```bash

ansible-galaxy search apache

```

2. \*\*Install a Role\*\*:

```bash

ansible-galaxy install geerlingguy.apache

```

3. \*\*Use the Role in a Playbook\*\*:

```yaml

---

- hosts: all

roles:

- geerlingguy.apache

```

---

## \*\*5. Best Practices for Using Roles\*\*

### \*\*Role Reusability\*\*

- \*\*Parameterization\*\*: Use variables to make roles reusable across different environments.

- \*\*Documentation\*\*: Provide a `README.md` file to explain the role's purpose and usage.

### \*\*Role Dependencies\*\*

- \*\*Dependencies\*\*: Use the `meta/main.yml` file to define role dependencies.

```yaml

dependencies:

- role: geerlingguy.apache

- role: geerlingguy.mysql

```

### \*\*Role Testing\*\*

- \*\*Testing\*\*: Use tools like `molecule` to test roles in different environments.

---

## \*\*6. Conclusion and Next Steps\*\*

### \*\*What We Learned\*\*

- \*\*Ansible Roles\*\*: A way to organize playbooks and related files into reusable components.

- \*\*Creating Roles\*\*: Step-by-step guide to creating a role for setting up a web server.

- \*\*Ansible Galaxy\*\*: Using pre-built roles to speed up automation.

- \*\*Best Practices\*\*: Reusability, dependencies, and testing.

### \*\*Next Steps\*\*

- \*\*Day 5\*\*: Advanced role techniques, including role dependencies and testing.

- \*\*Day 6\*\*: Using Ansible for configuration management and deployment.

---

## \*\*Images and Tables\*\*

### \*\*Role Directory Structure\*\*

```

web\_server/

├── tasks

│ └── main.yml

├── handlers

│ └── main.yml

├── templates

│ └── index.html.j2

├── files

│ └── example.conf

├── vars

│ └── main.yml

├── defaults

│ └── main.yml

├── meta

│ └── main.yml

└── README.md

```

### \*\*Example Playbook Using a Role\*\*

```yaml

---

- hosts: all

roles:

- web\_server

```

---

## \*\*References\*\*

- [Ansible Documentation](https://docs.ansible.com/)

- [Ansible Roles Guide](https://docs.ansible.com/ansible/latest/user\_guide/playbooks\_reuse\_roles.html)

- [Ansible Galaxy](https://galaxy.ansible.com/)

---

This study material provides a comprehensive guide to Ansible roles, including creating roles, using Ansible Galaxy, and best practices. It is designed to help you understand the core concepts and apply them in real-world scenarios.  
  
  
**Study Material: Day 4 - Ansible Roles and Folder Structure**

**Table of Contents**

1. **Introduction to Ansible Roles**
   * What are Ansible Roles?
   * Why Use Roles?
2. **Anatomy of an Ansible Role**
   * Directory Structure
   * Key Components: Tasks, Handlers, Templates, Variables, Defaults, Files, Meta
3. **Creating Your First Ansible Role**
   * Example: Setting Up a Web Server
   * Step-by-Step Guide
4. **Using Ansible Galaxy**
   * What is Ansible Galaxy?
   * How to Use Pre-Built Roles
5. **Best Practices for Using Roles**
   * Role Reusability
   * Role Dependencies
   * Role Testing
6. **Conclusion and Next Steps**

**1. Introduction to Ansible Roles**

**What are Ansible Roles?**

Ansible roles are a way to organize playbooks and related files into a reusable structure. Roles allow you to break down complex automation tasks into smaller, manageable components.

**Why Use Roles?**

* **Reusability**: Roles can be reused across multiple playbooks and projects.
* **Modularity**: Roles make it easier to manage and share code.
* **Scalability**: Roles help in scaling automation efforts by organizing tasks into logical units.

**2. Anatomy of an Ansible Role**

**Directory Structure**

A typical Ansible role has the following directory structure:

role\_name/

├── tasks

│ └── main.yml

├── handlers

│ └── main.yml

├── templates

│ └── index.html.j2

├── files

│ └── example.conf

├── vars

│ └── main.yml

├── defaults

│ └── main.yml

├── meta

│ └── main.yml

└── README.md

**Key Components**

* **Tasks**: Contains the main list of tasks to be executed by the role.
* **Handlers**: Contains handlers, which are tasks that are triggered by other tasks.
* **Templates**: Contains Jinja2 templates for configuration files.
* **Files**: Contains static files that need to be copied to the managed nodes.
* **Vars**: Contains variables specific to the role.
* **Defaults**: Contains default variables for the role.
* **Meta**: Contains metadata about the role, including dependencies.

**3. Creating Your First Ansible Role**

**Example: Setting Up a Web Server**

Let's create a role to set up a web server using Apache.

**Step-by-Step Guide**

1. **Create the Role Structure**:
2. ansible-galaxy init web\_server

This will create the directory structure for the web\_server role.

1. **Define Tasks**: Edit web\_server/tasks/main.yml:
2. ---
3. - name: Install Apache
4. apt:
5. name: apache2
6. state: present
7. - name: Ensure Apache is running
8. service:
9. name: apache2
10. state: started
11. enabled: yes
12. - name: Copy index.html
13. copy:
14. src: files/index.html
15. dest: /var/www/html/index.html
16. **Add Static Files**: Create web\_server/files/index.html:
17. <html>
18. <body>
19. <h1>Welcome to the Web Server!</h1>
20. </body>
21. </html>
22. **Use the Role in a Playbook**: Create a playbook site.yml:
23. ---
24. - hosts: all
25. roles:
26. - web\_server
27. **Run the Playbook**:
28. ansible-playbook -i inventory.ini site.yml

**4. Using Ansible Galaxy**

**What is Ansible Galaxy?**

Ansible Galaxy is a repository of pre-built roles that can be used to speed up automation tasks.

**How to Use Pre-Built Roles**

1. **Search for Roles**:
2. ansible-galaxy search apache
3. **Install a Role**:
4. ansible-galaxy install geerlingguy.apache
5. **Use the Role in a Playbook**:
6. ---
7. - hosts: all
8. roles:
9. - geerlingguy.apache

**5. Best Practices for Using Roles**

**Role Reusability**

* **Parameterization**: Use variables to make roles reusable across different environments.
* **Documentation**: Provide a README.md file to explain the role's purpose and usage.

**Role Dependencies**

* **Dependencies**: Use the meta/main.yml file to define role dependencies.
* dependencies:
* - role: geerlingguy.apache
* - role: geerlingguy.mysql

**Role Testing**

* **Testing**: Use tools like molecule to test roles in different environments.

**6. Conclusion and Next Steps**

**What We Learned**

* **Ansible Roles**: A way to organize playbooks and related files into reusable components.
* **Creating Roles**: Step-by-step guide to creating a role for setting up a web server.
* **Ansible Galaxy**: Using pre-built roles to speed up automation.
* **Best Practices**: Reusability, dependencies, and testing.

**Next Steps**

* **Day 5**: Advanced role techniques, including role dependencies and testing.
* **Day 6**: Using Ansible for configuration management and deployment.

**Images and Tables**

**Role Directory Structure**

web\_server/

├── tasks

│ └── main.yml

├── handlers

│ └── main.yml

├── templates

│ └── index.html.j2

├── files

│ └── example.conf

├── vars

│ └── main.yml

├── defaults

│ └── main.yml

├── meta

│ └── main.yml

└── README.md

**Example Playbook Using a Role**

---

- hosts: all

roles:

- web\_server

**References**

* [Ansible Documentation](https://docs.ansible.com/)
* [Ansible Roles Guide](https://docs.ansible.com/ansible/latest/user_guide/playbooks_reuse_roles.html)
* [Ansible Galaxy](https://galaxy.ansible.com/)

This study material provides a comprehensive guide to Ansible roles, including creating roles, using Ansible Galaxy, and best practices. It is designed to help you understand the core concepts and apply them in real-world scenarios.