To create roles for the given playbook, we need to divide the tasks and handlers into a structured role format. Here's how the structure will look:

**Directory Structure**

ansible-galaxy role init merge-role

**Role Files**

**merge-role/tasks/main.yml**

---

- include\_tasks: ping.yml

- include\_tasks: print\_message.yml

- include\_tasks: install\_apache.yml

- include\_tasks: check\_boolean.yml

- name: Restart everything

ansible.builtin.command:

cmd: echo "This task will restart the web services"

notify:

- Restart Apache

**merge-role/tasks/ping.yml**

- name: Ping remote hosts

ansible.builtin.ping:

tags:

- print\_message

**merge-role/tasks/print\_message.yml**

- name: Print message

ansible.builtin.debug:

var: my\_msg\_var

tags:

- print\_message

**merge-role/tasks/install\_apache.yml**

- name: Install Apache

ansible.builtin.apt:

name: apache2

state: present

tags:

- apache\_install

**merge-role/tasks/check\_boolean.yml**

- name: Check Boolean value

ansible.builtin.debug:

msg: "Variable is true"

when: is\_enabled

tags:

- check\_test

**merge-role/handlers/main.yml**

- name: Restart Apache

ansible.builtin.service:

name: apache2

state: started

listen:

- Restart web services

**merge-role/vars/main.yml**

my\_msg\_var: Hello Learners

is\_enabled: true

**Updated Playbook to Use Role**

Create a new playbook (site.yml) to call the role:

- name: Use Merged Playbook Role

hosts: all

become: true

become\_user: root

roles:

- role: **merge-role**

**Execution**

Run the playbook with:

ansible-playbook -i inventory.ini lab16.yml

This setup ensures the playbook tasks are modular, reusable, and easier to manage with a structured role format.