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.