**1. Create Ansible Roles**

Ansible roles help you organize tasks, handlers, and variables into separate directories. You can create roles for Apache2 and NGINX installation.

**Create Directory Structure for Roles**

**mkdir -p roles/apache2/tasks**

**mkdir -p roles/nginx/tasks**

**2. Define Role for Apache2 Installation**

**Role Directory Structure:**

roles/

└── apache2/

└── tasks/

└── main.yml

**Create main.yml for Apache2 Role:**

# roles/apache2/tasks/main.yml

---

- name: Update package index

  apt:

    update\_cache: yes

- name: Install Apache2

  apt:

    name: apache2

    state: present

- name: Ensure Apache2 service is running

  service:

    name: apache2

    state: started

    enabled: yes

**3. Define Role for NGINX Installation**

**Role Directory Structure:**

roles/

└── nginx/

└── tasks/

└── main.yml

**Create main.yml for NGINX Role:**

# roles/nginx/tasks/main.yml

---

- name: Update package index

  apt:

    update\_cache: yes

- name: Install NGINX

  apt:

    name: nginx

    state: present

- name: Ensure NGINX service is running

  service:

    name: nginx

    state: started

    enabled: yes

**4. Create the Main Playbook**

Create a playbook that applies these roles to the appropriate hosts.

**Create site.yml:**

---

- name: Install Apache2 on slave1

  hosts: Slave1

  become: yes

  roles:

    - apache2

- name: Install NGINX on slave2

  hosts: Slave2

  become: yes

  roles:

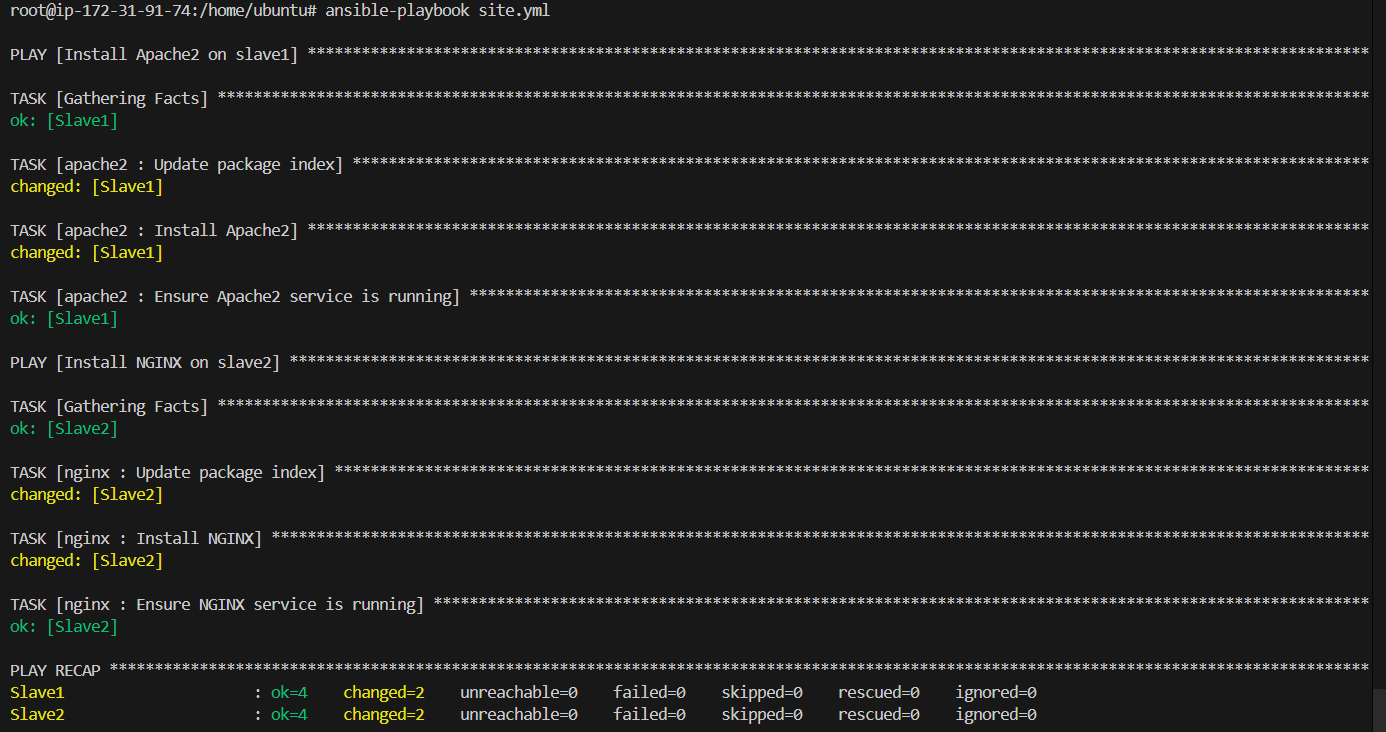
    - nginx

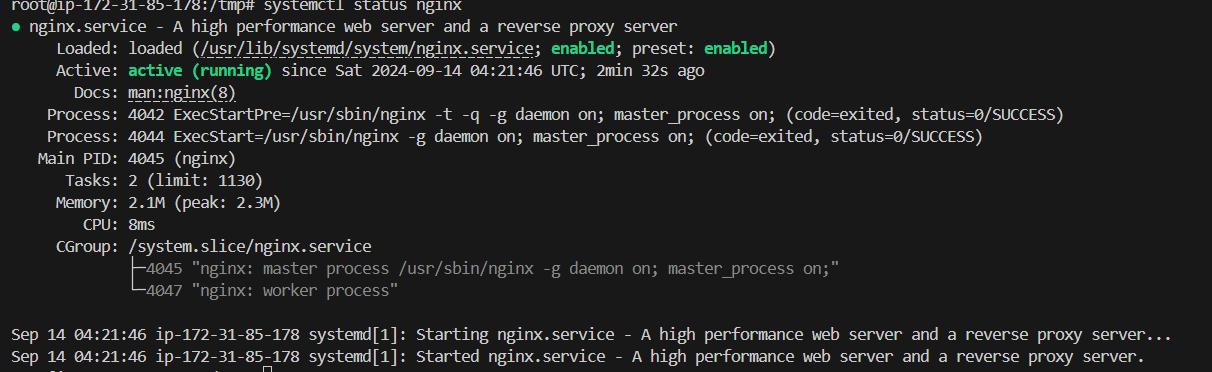
**Summary**

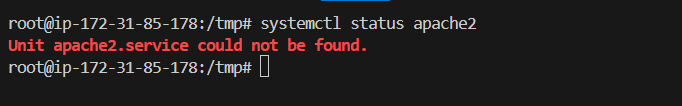
1. **Create Roles:**
   * roles/apache2/tasks/main.yml: Contains tasks to install Apache2.
   * roles/nginx/tasks/main.yml: Contains tasks to install NGINX.
2. **Create a Playbook (site.yml):**
   * Applies the apache2 role to slave1.
   * Applies the nginx role to slave2.
3. **Run the Playbook:**

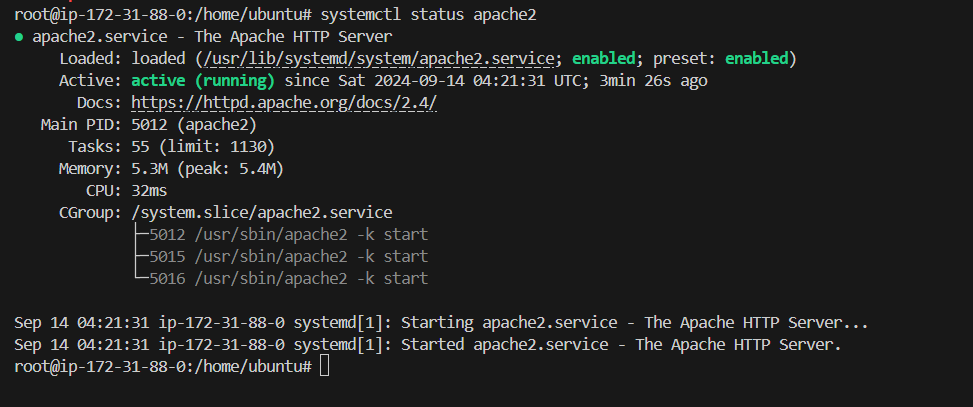
**5.ansible-playbook site.yml**

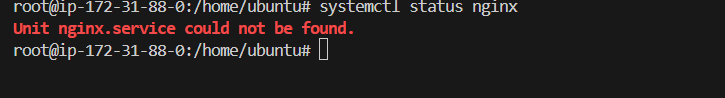
This setup ensures that Apache2 is installed on slave1 and NGINX is installed on slave2 using separate Ansible roles.











**1. Prepare the index.html File**

Create an index.html file that you want to deploy to the NGINX server.

**Create index.html:**

<!DOCTYPE html>

<html>

<head>

    <title>Welcome to My Website</title>

</head>

<body>

    <h1>Hello, World!</h1>

    <p>This is a custom index.html file served by NGINX.</p>

</body>

</html>

**2. Update NGINX Role to Include the index.html File**

Modify the nginx role to include the index.html file and ensure it is copied to the NGINX server's document root.

**Role Directory Structure:**

roles/

└── nginx/

├── tasks/

│ └── main.yml

└── files/

└── index.html

**Place index.html in the files directory of the nginx role:**

**roles/nginx/files/index.html**

**Update main.yml to deploy the index.html file:**

# roles/nginx/tasks/main.yml

---

- name: Update package index

  apt:

    update\_cache: yes

- name: Install NGINX

  apt:

    name: nginx

    state: present

- name: Ensure NGINX service is running

  service:

    name: nginx

    state: started

    enabled: yes

- name: Copy custom index.html to NGINX document root

  copy:

    src: index.html

    dest: /var/www/html/index.html

    owner: www-data

    group: www-data

    mode: '0644'

**3. Update the Playbook**

Ensure your playbook (site.yml) applies the nginx role only to the appropriate host.

**site.yml:**

---

- name: Install Apache2 on slave1

  hosts: slave1

  become: yes

  roles:

    - apache2

- name: Install NGINX on slave2 and replace index.html

  hosts: slave2

  become: yes

  roles:

    - nginx

**4. Run the Ansible Playbook**

Execute the playbook to apply the changes:

**ansible-playbook site.yml**

