**Playbook Structure (No Roles)**

---

- hosts: all

become: true

vars:

nginx\_version: latest

nginx\_config\_src: /path/to/your/nginx.conf # Static file or template source

tasks:

- name: Install Nginx

ansible.builtin.yum:

name: "nginx-{{ nginx\_version }}"

state: present

- name: Copy custom Nginx config file

ansible.builtin.copy:

src: "{{ nginx\_config\_src }}"

dest: /etc/nginx/nginx.conf

notify: restart nginx

- name: Ensure Nginx is started and enabled

ansible.builtin.systemd:

name: nginx

state: started

enabled: true

handlers:

- name: restart nginx

ansible.builtin.systemd:

name: nginx

state: restarted

**Step 1: Create the Role Directory Structure**

You can use the ansible-galaxy init command to automatically generate the necessary directory structure for a role:

bash

ansible-galaxy init my\_custom\_role

This will create the following structure:

css

my\_custom\_role/

├── defaults/

│ └── main.yml

├── files/

├── handlers/

│ └── main.yml

├── meta/

│ └── main.yml

├── tasks/

│ └── main.yml

├── templates/

├── tests/

│ ├── inventory

│ └── test.yml

└── vars/

└── main.yml

Each of these directories serves a different purpose in the role:

* **defaults/**: Default variables that can be overridden by playbooks.
* **files/**: Static files that will be copied to the remote system.
* **handlers/**: Handlers triggered by tasks (e.g., restarting a service).
* **meta/**: Role metadata, including dependencies.
* **tasks/**: The main list of tasks executed by the role.
* **templates/**: Jinja2 templates for configuration files.
* **vars/**: Variables that typically should not be overridden.
* **tests/**: Basic testing framework.

**Step 2: Define Tasks**

Open tasks/main.yml to define the tasks your role will perform. Here’s a simple example of installing Nginx in the tasks/main.yml file:

---

# tasks/main.yml

- name: Install Nginx

  ansible.builtin.yum:

    name: nginx

    state: present

- name: Ensure Nginx is started

  ansible.builtin.systemd:

    name: nginx

    state: started

    enabled: true

**Step 3: Define Variables**

If you want to use variables, add them in vars/main.yml or defaults/main.yml. Let’s say you want to make the Nginx version configurable:

Edit defaults/main.yml:

---

# defaults/main.yml

nginx\_version: latest

Now modify the tasks/main.yml to use this variable:

---

- name: Install Nginx

ansible.builtin.yum:

name: "nginx-{{ nginx\_version }}"

state: present

**Step 4: Handlers:**

**If you need handlers (e.g., restarting services after changes), define them in handlers/main.yml:**

---

# handlers/main.yml

- name: restart nginx

ansible.builtin.systemd:

name: nginx

state: restarted

You can notify this handler from any task:

---

- name: Copy Nginx config file

ansible.builtin.copy:

src: /path/to/local/nginx.conf

dest: /etc/nginx/nginx.conf

notify: restart nginx

**Step 5: Add Files or Templates**

**If your role needs to copy static files or use templates, you can place them in the files/ or templates/ directory, respectively.**

**Copying a Static File:**

**Place the file in files/ and add a task to copy it:**

**---**

**- name: Copy custom index.html**

**ansible.builtin.copy:**

**src: index.html**

**dest: /usr/share/nginx/html/index.html**

**Using a Jinja2 Template:**

Create a file templates/nginx.conf.j2:

server {

listen 80;

server\_name {{ ansible\_hostname }};

location / {

root /usr/share/nginx/html;

}

}

Add a task to render the template:

---

- name: Template Nginx config

ansible.builtin.template:

src: nginx.conf.j2

dest: /etc/nginx/nginx.conf

notify: restart nginx

**Step 6: Define Dependencies (Optional)**

**If your role depends on other roles, you can define those in meta/main.yml:**

**---**

**dependencies:**

**- { role: geerlingguy.firewall }**

**Step 7: Running the Role in a Playbook**

**Once the role is created, you can use it in a playbook like this:**

**---**

**- hosts: all**

**become: true**

**roles:**

**- my\_custom\_role**

**Step 8: Testing the Role**

**You can use the tests/ directory to create a basic test. Here's an example tests/test.yml playbook for testing:**

**---**

**- hosts: localhost**

**roles:**

**- my\_custom\_role**

**To run the test, use the inventory file provided under tests/:**

**ansible-playbook -i tests/inventory tests/test.yml**