



Lab 10 — Ansible Templates (Jinja2 Templates)

Author: Sandeep Kumar Sharma



Learning Objectives

In this lab, you will learn: - What templates are in Ansible - What Jinja2 is and why it is used - How to create a `.j2` template file - How to pass variables into templates - How to deploy configuration files dynamically using templates



Learning Outcomes

After completing this lab, you will: - Understand the role of Jinja2 in Ansible - Generate dynamic configuration files using variables - Use the `template` module effectively - Deploy OS-specific or environment-specific configs with minimal effort



What Are Templates in Ansible?

Templates in Ansible are files written using **Jinja2 syntax**.

They allow you to dynamically insert values, variables, conditions, and loops into configuration files.

Templates help automate: - Application config files - Web server config files - Environment-specific settings - Dynamic scripts

A template file usually ends with `.j2` (Jinja2).



What Is Jinja2?

Jinja2 is a templating engine used by Ansible to: - Insert variables - Apply conditional logic - Loop inside templates - Generate dynamic content

Examples:

```
 {{ variable_name }}  
 {% if condition %}  
 {% endif %}
```

```
{% for item in list %}  
{% endfor %}
```



SECTION A — Create a Simple Template File

We will create an Apache configuration template.

Step 1: Create templates directory

```
mkdir templates
```

Step 2: Create a template file

```
nano templates/apache.conf.j2
```

Add:

```
# Apache Configuration Generated by Ansible  
ServerName {{ server_name }}  
Listen {{ listen_port }}  
  
<VirtualHost *:{{ listen_port }}>  
    DocumentRoot {{ document_root }}  
</VirtualHost>
```



SECTION B — Create Playbook to Use the Template

Create file:

```
nano template-example.yml
```

Add:

```

---
- name: Deploy Apache configuration using Jinja2 template
  hosts: dev
  become: yes

  vars:
    server_name: "example.com"
    listen_port: 8080
    document_root: "/var/www/html"

  tasks:
    - name: Install Apache (Ubuntu)
      apt:
        name: apache2
        state: present
      when: ansible_facts['os_family'] == 'Debian'

    - name: Install Apache (Amazon Linux)
      yum:
        name: httpd
        state: present
      when: ansible_facts['os_family'] == 'RedHat'

    - name: Deploy configuration from template
      template:
        src: templates/apache.conf.j2
        dest: "/etc/apache2/sites-available/000-default.conf"
      when: ansible_facts['os_family'] == 'Debian'

    - name: Deploy configuration for Amazon Linux
      template:
        src: templates/apache.conf.j2
        dest: "/etc/httpd/conf/httpd.conf"
      when: ansible_facts['os_family'] == 'RedHat'

    - name: Restart Apache
      service:
        name: "{{ 'apache2' if ansible_facts['os_family'] == 'Debian' else
        'httpd' }}"
        state: restarted

```

Run the playbook:

```
ansible-playbook template-example.yml
```



SECTION C — Template with Loop & Conditional (Advanced Example)

Create file:

```
nano templates/users.j2
```

Add:

```
# User List Generated by Ansible
{% for user in users %}
User: {{ user.name }}, Shell: {{ user.shell }}
{% endfor %}
```

Playbook:

```
nano template-users.yml
```

```
---
- name: Generate dynamic user list file
  hosts: dev
  become: yes

  vars:
    users:
      - { name: "user1", shell: "/bin/bash" }
      - { name: "user2", shell: "/bin/sh" }
      - { name: "user3", shell: "/bin/bash" }

  tasks:
    - name: Create output from template
      template:
        src: templates/users.j2
        dest: /tmp/user-list.txt
```

Run:

```
ansible-playbook template-users.yml
```



Hands-On Checklist

- [] Create a Jinja2 template
 - [] Use the `template` module
 - [] Deploy dynamic Apache config
 - [] Generate dynamic files using variable loops
 - [] Restart services after applying templates
-



Lab Summary

In this lab, you learned:

- What templates are and how Jinja2 works
- How to create and use template files in Ansible
- How to generate dynamic config files
- How to integrate variables, loops, and conditions inside templates

Next Lab: [Lab 11 — Handlers & Notifications in Ansible](#)