# 🧪Lab 8 — Conditionals & When Statements in Ansible

**Author:** *Sandeep Kumar Sharma*

---

## 🕗Learning Objectives

In this lab, you will learn: - What conditionals are in Ansible - How to use the `when` keyword - How to run tasks only when conditions are met - How to use fact-based conditions - How to write simple and multi-condition logic

---

## 👙Learning Outcomes

After completing this lab, you will: - Use conditional logic in playbooks - Execute tasks based on OS family, variable values, or system facts - Build smarter, OS-aware automation

---

## 🧰What Are Conditionals in Ansible?

Conditionals allow Ansible to decide **whether a task should run**. They use the `when` keyword.

For example: - Run a task only on Ubuntu - Install a package only if a variable has a certain value - Execute a task only if the file exists

Conditionals make your automation dynamic.

---

## 🪙SECTION A — Using the `when` Statement

Format:

```
when: condition
```

Examples:

```
when: ansible_facts['os_family'] == 'Debian'
when: my_var == 'yes'
when: my_number > 10
```

---

## 🪙SECTION B — Hands-On Example: OS-Based Package Installation

Create file:

```
nano conditional-os.yml
```

Add:

```
---
- name: Install correct web server based on OS
  hosts: dev
  become: yes
  gather_facts: yes

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

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

Run:

```
ansible-playbook conditional-os.yml
```

---

## 🪙SECTION C — Example: Running a Task Only When a Variable Matches

Create file:

```
nano conditional-var.yml
```

Add:

```
---
- name: Using variable-based condition
  hosts: dev
  become: yes

  vars:
    install_pkg: yes

  tasks:
    - name: Install tree
      package:
        name: tree
        state: present
      when: install_pkg == 'yes'
```

Run:

```
ansible-playbook conditional-var.yml
```

---

## 🪙SECTION D — Multiple Conditions (AND/OR)

```
when: ansible_facts['os_family'] == 'Debian' and
ansible_facts['processor_vcpus'] > 2
```

Example playbook:

```
nano conditional-multi.yml
```

Add:

```
---
- name: Multi-condition example
  hosts: dev
  become: yes
  gather_facts: yes

  tasks:
    - name: Create directory only if CPU > 2 and OS is Debian
      file:
        path: /tmp/high_cpu
        state: directory
      when: ansible_facts['os_family'] == 'Debian' and
ansible_facts['processor_vcpus'] > 2
```

Run:

```
ansible-playbook conditional-multi.yml
```

---

## 🪙SECTION E — Checking File Existence

Use `stat` module + condition.

Create file:

```
nano conditional-file.yml
```

Add:

```
---
- name: Run task only if file exists
  hosts: dev
  become: yes

  tasks:
    - name: Check file
      stat:
        path: /etc/passwd
      register: fileout
    - name: Print message if file exists
      debug:
        msg: "File exists on this system"
      when: fileout.stat.exists
```

Run:

```
ansible-playbook conditional-file.yml
```

---

## 🧪 Hands-On Checklist

- [ ] Use OS-based conditions
- [ ] Use variable-based conditions
- [ ] Use multi-conditions (AND/OR)
- [ ] Use fact-based conditions
- [ ] Use stat module for file checks

---

## 👙 Lab Summary

In this lab, you learned: - The purpose of conditionals - How to use the `when` statement - How to create conditional logic based on OS, variables, and file existence - How to execute smarter, dynamic playbooks

Next Lab: 👉**Lab 9 — Loops in Ansible (loop, with_items, dictionary loops)**