🧪Lab 7 — Using Variables in Ansible Playbooks (vars, vars_files, vars_prompt, Inventory Vars)

Author: *Sandeep Kumar Sharma*

🕖Learning Objectives

In this lab, you will learn: - What variables are in Ansible - How to define and use `vars` inside a playbook - How to load variables from external files (`vars_files`) - How to ask users for input during runtime (`vars_prompt`) - How inventory variables (host_vars & group_vars) work

---

👙Learning Outcomes

After completing this lab, you will: - Understand all major variable types in Ansible - Use variables in playbooks to make automation dynamic - Store variables in external files - Prompt users for passwords or values at execution time - Use group-level and host-level variables effectively

---

🧰What Are Variables in Ansible?

Variables store reusable values. They help avoid repetition and make playbooks flexible.

Examples of values stored in variables: - Package names - Ports - File paths - Usernames - IP addresses

---

🪙SECTION A — Using `vars` Inside a Playbook

Variables can be declared directly inside a playbook.

Create a file:

```
nano vars-example.yml
```

Add:

```
---
- name: Demo using vars
  hosts: dev
  become: yes

  vars:
    web_package: "{{ 'apache2' if ansible_facts['os_family'] == 'Debian' else 'httpd' }}"
    web_service: "{{ web_package }}"

  tasks:
    - name: Install web server
      package:
        name: "{{ web_package }}"
        state: present

    - name: Start service
      service:
        name: "{{ web_service }}"
        state: started
        enabled: yes
```

Run:

```
ansible-playbook vars-example.yml
```

🪙SECTION B — Using `vars_files`

Variables can be stored in external files and loaded into playbooks.

Step 1: Create a separate variable file

```
mkdir vars
nano vars/web_vars.yml
```

Add:

```
web_package: apache2
web_service: apache2
```

Step 2: Load it inside playbook

```
nano vars-file-example.yml
```

Add:

```
---
- name: Demo using vars_files
  hosts: dev
  become: yes

  vars_files:
    - vars/web_vars.yml

  tasks:
    - name: Install Apache
      package:
        name: "{{ web_package }}"
        state: present

    - name: Start Apache
      service:
        name: "{{ web_service }}"
        state: started
        enabled: yes
```

Run:

```
ansible-playbook vars-file-example.yml
```

🪙SECTION C — Using `vars_prompt` (User Input)

This option prompts the user for values at execution time.

Create file:

```
nano vars-prompt-example.yml
```

Add:

```
---
- name: Demo using vars_prompt
  hosts: dev
  become: yes

  vars_prompt:
    - name: username
      prompt: "Enter the username to create"
    - name: shell_type
      prompt: "Enter shell (e.g. /bin/bash)"
      default: "/bin/bash"

  tasks:
    - name: Create user
      user:
        name: "{{ username }}"
        shell: "{{ shell_type }}"
```

Run:

```
ansible-playbook vars-prompt-example.yml
```

🪙SECTION D — Inventory Variables (host_vars & group_vars)

Inventory variables allow each group or host to have its own values.

Step 1: Create directory structure

```
mkdir -p group_vars\mkdir -p host_vars
```

Step 2: Create group-level variable file

```
nano group_vars/dev.yml
```

Add:

```
web_package: apache2
web_service: apache2
```

Step 3: Create host-level variable file

```
nano host_vars/node1.yml
```

Add:

```
custom_message: "Hello from Node1"
```

Step 4: Create playbook

```
nano inventory-vars-example.yml
```

Add:

```
---
- name: Demo inventory variables
  hosts: dev
  become: yes

  tasks:
    - name: Install web server using inventory vars
      package:
        name: "{{ web_package }}"
        state: present

    - name: Show custom message (only prints on Node1 if defined)
      debug:
        msg: "{{ custom_message | default('No host message available') }}"
```

Run:

```
ansible-playbook inventory-vars-example.yml
```

🧪Hands-On Checklist

- [ ] Run playbook with inline vars
- [ ] Use vars_files
- [ ] Use vars_prompt
- [ ] Create group_vars & host_vars
- [ ] Run inventory-based variable playbook

👙Lab Summary

In this lab, you learned: - How variables work in Ansible - How to define variables inside playbooks - How to store variables in external files - How to prompt users for values - How to use inventory-level variables

Next Lab: 👉**Lab 8 — Conditionals & When Statements**