🧪**Lab 4 — Understanding Ansible Facts & Fact Gathering **

Author: *Sandeep Kumar Sharma*

🕗Learning Objectives

In this lab, you will learn:

- What Ansible Facts are
- Why fact gathering is important
- How to gather facts using ad-hoc commands and playbooks
- How to filter, view, and use facts inside automation
- Practical exercises using both Ubuntu & Amazon Linux 2 hosts

---

👙Learning Outcomes

After completing this lab, you will:

- Understand how Ansible collects system information
- Use `setup` module to explore facts
- Use facts inside tasks and conditions
- Build intelligent, dynamic playbooks

🧰**What Are Ansible Facts? **

facts are simply **system information** that Ansible automatically collects from your managed nodes.

Think of facts like:

- What is the OS?
- How much RAM is available?
- How many CPUs do we have?
- What is the IP address of the server?
- What is the hostname?
- What Python version is installed?
- What kernel version is running?

Whenever you run a playbook, Ansible automatically performs **fact gathering** unless disabled.

It collects all this information and stores it in a variable called:

```
ansible_facts
```

You can use these facts in your playbooks to make them **smart, dynamic, and OS-aware**.

🕯️ Why Do We Need Fact Gathering?

Let me explain this like a :

👉Imagine you are writing an automation to install Apache.

- On Ubuntu the service is called **apache2**
- On Amazon Linux and RHEL the service is **httpd**

If you hard-code it, the playbook will break on one of the OS types.

But if you use facts:

- You can detect OS automatically
- Then choose the right package dynamically

Also, facts help in:

- Conditional tasks
- Dynamic inventory
- Reporting
- Validating system state
- Debugging

So in simple words:

> **Facts make your automation intelligent. Without facts, automation becomes static.**

🪙SECTION A — Basic Fact Gathering Using Ad-Hoc Commands

To gather all system facts:

```
ansible dev -m setup
```

This prints large output containing 200+ facts.

🪙SECTION B — Filter Specific Facts

1. Get only OS-related facts

```
ansible dev -m setup -a "filter=ansible_os_family"
```

2. Get only IP address

```
ansible dev -m setup -a "filter=ansible_default_ipv4"
```

3. Get only memory information

```
ansible dev -m setup -a "filter=ansible_mem*"
```

🪙SECTION C — Save Facts Output to a File

```
ansible dev -m setup > facts_output.json
```

🪠SECTION D — Using Facts Inside a Playbook (Hands-On)

Create a playbook:

```
nano fact-demo.yml
```

Add:

```
---
- name: Demo of using ansible facts
  hosts: dev
  gather_facts: yes

  tasks:
    - name: Print OS Family
      debug:
        msg: "The OS Family is: {{ ansible_facts['os_family'] }}"

    - name: Print Default IPv4 Address
      debug:
        msg: "IP Address: {{ ansible_facts['default_ipv4']['address'] }}"

    - name: Install correct web server based on OS
      package:
        name: "{{ 'apache2' if ansible_facts['os_family'] == 'Debian' else
'httpd' }}"
        state: present
      become: yes
```

Run the playbook:

```
ansible-playbook fact-demo.yml
```

🪜 **SECTION E — Real-World Use Case **

" imagine you are managing 500 servers. Some are Ubuntu, some are Amazon Linux, some are RHEL.

You don’t want to write 3 separate playbooks.

Using facts, your single playbook automatically detects:

- Which OS is running
- Which package manager to use (APT vs YUM)
- Which service name is correct
- Which network interface is active

This is exactly how we build **production-grade automation**."

🧪Hands-On Exercise

Run the following ad-hoc commands and note the differences:

1. Get Hostname

```
ansible dev -m setup -a "filter=ansible_hostname"
```

2. Get CPU Count

```
ansible dev -m setup -a "filter=ansible_processor_vcpus"
```

3. Get OS Version

```
ansible dev -m setup -a "filter=ansible_distribution*"
```

4. Get Python Interpreter

```
ansible dev -m setup -a "filter=ansible_python*"
```

5. Get All Network Interfaces

```
ansible dev -m setup -a "filter=ansible_interfaces"
```

👙Lab Summary

In this lab, you learned:

- What facts are
- Why fact gathering is important
- How to gather facts using setup module
- How to filter specific facts
- How to use facts inside playbooks to build intelligent automation