✅ Lab Exercise: Install Ansible in Fedora & Configure Inventory

🔹 Step 1: Update the System

```
sudo dnf update -y
```

🔹 Step 2: Install Ansible

In Fedora, Ansible is available via `dnf`.

```
sudo dnf install ansible -y
```

Verify installation:

```
ansible --version
```

You should see Ansible version details.

🔹 Step 3: Check Inventory File Location

Default inventory file:

```
/etc/ansible/hosts
```

Open it:

```
sudo nano /etc/ansible/hosts
```

🔹 Step 4: Configure Inventory

🟢 If Using Single VM (Localhost) in last line

Add:

```
[local]
localhost ansible_connection=local
```

Save and exit.(ctrl+o ,enter ,Ctrl+x)

🔵 If Using Two VMs (Control + Target)

Example:

```
[web]
192.168.1.20 ansible_user=your_username
```

Replace:

- `192.168.1.20` → Target VM IP
- `your_username` → Target VM username

🔹 Step 5: Setup SSH (If Using Multiple VMs)

On control node:

```
ssh-keygen
ssh-copy-id username@target-ip
```

Test connection:

```
ssh username@target-ip
```

It should log in without password.

🔹 Step 6: Test Inventory

Run:

```
ansible all -m ping
```

Expected output:

```
SUCCESS => {
    "ping": "pong"
}
```

- next ex

🔷 STEP-BY-STEP: Ansible Ad-hoc Commands & Playbook on Fedora

PART 1: Run Ad-hoc Commands

Step 1: Open Terminal

- Use the terminal in your Fedora VM.

Step 2: Test Connectivity (Ping Module)

```
ansible all -m ping
```

- `all` → runs on all hosts in inventory
- `-m ping` → uses the ping module

Expected Output:

```
localhost | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

Step 3: Run a Simple Command

Check system uptime:

```
ansible all -m command -a "uptime"
```

- `command` → module to run Linux commands
- `-a "uptime"` → argument is the command

Expected Output:

```
localhost | CHANGED | rc=0 >>
 14:11:53 up 25 min,  1 user,  load average: 0.12, 0.24, 0.23
```

Step 4: Check Disk Space

```
ansible all -m command -a "df -h"
```

- Displays available disk space and usage.

Step 5: Install a Package (git example)

On Fedora 34+ you must specify DNF 5 backend:

```
ansible all -m dnf -a "name=git state=present use_backend=dnf5" --become --ask-become-pass
```

- `dnf` → package manager module
- `state=present` → ensures package is installed
- `use_backend=dnf5` → explicitly use DNF 5
- `--become` → run with sudo
- `--ask-become-pass` → prompts for your sudo password

Note: If git is already installed, Ansible will show it as “ok” instead of “changed.”

Step 6: Install & Start Apache (optional via ad-hoc)

```
ansible all -m dnf -a "name=httpd state=present use_backend=dnf5" --become --ask-become-pass
ansible all -m service -a "name=httpd state=started" --become --ask-become-pass
```

- Ensures Apache is installed and running.

PART 2: Create and Run a Playbook

Step 1: Create Playbook File

```
nano start_httpd.yml
```

Step 2: Write Playbook Content

Copy the following:

```
---
- name: Start Apache Service
  hosts: local
  become: yes
```

```
  tasks:
    - name: Start Apache
      service:
        name: httpd
        state: started
```

- `hosts: local` → runs on localhost group
- `become: yes` → use sudo
- `tasks` → list of tasks to perform

Save and exit: `CTRL + O`, Enter, `CTRL + X`

Step 3: Run the Playbook

```
ansible-playbook start_httpd.yml --ask-become-pass
```

- Enter your Fedora sudo password wh`ansible all -m service -a "name=httpd state=stopped" --become --ask-become-pass`en prompted.
- Output will show task execution:

```
TASK [Start Apache] ************************************************************
changed: [localhost]
```

Step 4: Verify Apache

```
systemctl status httpd
```

- Should show `Active: active (running)` ✅

Or open a browser in the VM:

```
http://localhost
```

- Apache welcome page should appear.

Step 5: Optional – Stop Apache

```
ansible all -m service -a "name=httpd state=stopped" --become --ask-become-pass
```