Lesson-End Project

Using Ansible to Display System Facts and Manage Files and Services

Project Agenda: To demonstrate how to use Ansible to display system facts, create directories and files, install packages, and manage services

Description: As a DevOps engineer at InnovateNow Tech, a rapidly growing tech company, you are tasked with automating the deployment of Apache web servers to handle a dynamic infrastructure that requires frequent updates and secure management of sensitive data. To do this, you are required to gather system information for all the required hosts/inventories in the network. You have decided to use Ansible facts and ad-hoc commands to accomplish this task.

Tools required: Ansible

Prerequisites: You need to have completed Lesson 3 Demo 01 before proceeding with this lesson-end project.

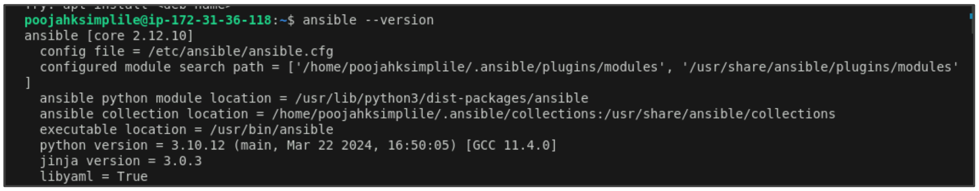
Expected Deliverables: Ansible playbook that successfully displays system facts and manages files and services, with installation and execution steps documented clearly.

Steps to be followed:

1. Create a directory and update the playbook
2. Execute the Ansible playbook

Step 1: Create a directory and update the playbook

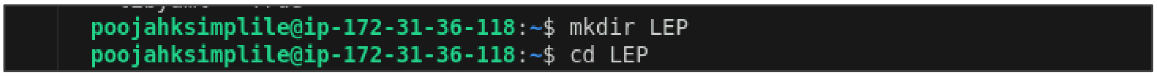
1.1 Execute the command to display the version of Ansible installed on your system:
ansible --version



```
Try: apt install <deb name>
poojahksimplile@ip-172-31-36-118:~$ ansible --version
ansible [core 2.12.10]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/poojahksimplile/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules'
]
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/poojahksimplile/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Mar 22 2024, 16:50:05) [GCC 11.4.0]
  jinja version = 3.0.3
  libyaml = True
```

1.2 Run the following commands to create a directory named **LEP** and navigate into it:

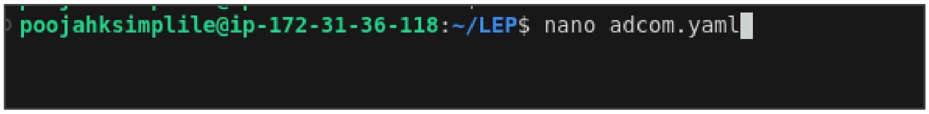
mkdir LEP
cd LEP



```
poojahksimplile@ip-172-31-36-118:~$ mkdir LEP
poojahksimplile@ip-172-31-36-118:~$ cd LEP
```

1.3 Execute the following command to open the **adcom.yml** file:

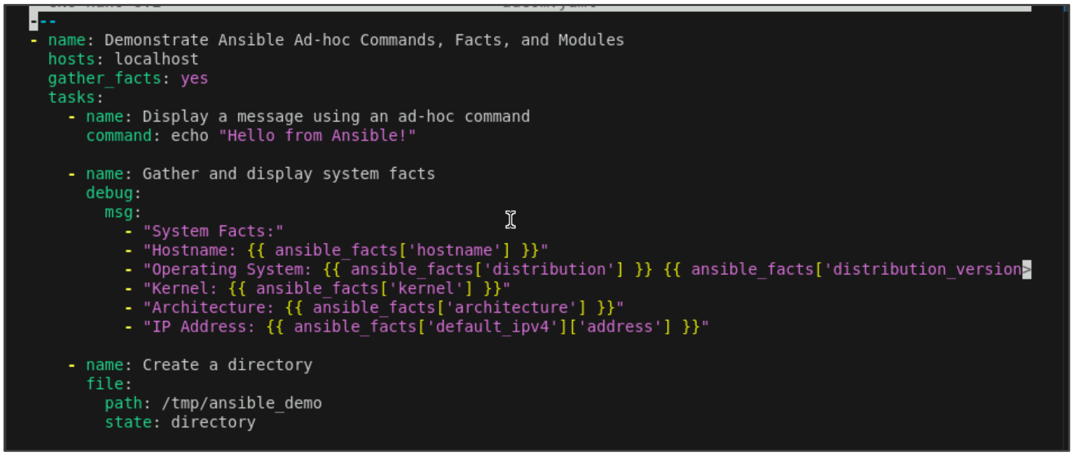
nano adcom.yaml



```
poojahksimplile@ip-172-31-36-118:~/LEP$ nano adcom.yaml
```

1.4 Add the following script to the **adcom.yaml** file:

```
---
- name: Demonstrate Ansible Ad-hoc Commands, Facts, and Modules
  hosts: localhost
  gather_facts: yes
  tasks:
    - name: Display a message using an ad-hoc command
      command: echo "Hello from Ansible!"

    - name: Gather and display system facts
      debug:
        msg:
          - "System Facts:"
          - "Hostname: {{ ansible_facts['hostname'] }}"
          - "Operating System: {{ ansible_facts['distribution'] }} {{ ansible_facts['distribution_version'] }}"
          - "Kernel: {{ ansible_facts['kernel'] }}"
          - "Architecture: {{ ansible_facts['architecture'] }}"
          - "IP Address: {{ ansible_facts['default_ipv4']['address'] }}"
    - name: Create a directory
      file:
        path: /tmp/ansible_demo
        state: directory

    - name: Create a file with some content
      copy:
        dest: /tmp/ansible_demo/hello.txt
        content: "Hello, this file was created by Ansible!\n"

    - name: Install a package (example with 'curl')
      package:
        name: curl
        state: present

    - name: Start and enable a service (example with 'cron')
      service:
        name: cron
        state: started
        enabled: yes

    - name: Fetch file from remote machine (here it's localhost itself)
      fetch:
        src: /tmp/ansible_demo/hello.txt
        dest: /tmp/hello_fetched.txt
        flat: yes

    - name: Change file permissions
      file:
        path: /tmp/ansible_demo/hello.txt
        mode: '0644'
        owner: root
        group: root
    - name: Demonstrate the use of the shell module
      shell: |
        echo "This command runs in the shell."
        uname -a

    - name: Demonstrate the use of the command module
      command: uptime
      register: result

    - name: Show the result of the command module
      debug:
        msg: "The system uptime is: {{ result.stdout }}"

    - name: Cleanup - remove the demo directory and its content
      file:
        path: /tmp/ansible_demo
        state: absent
        recurse: yes
```



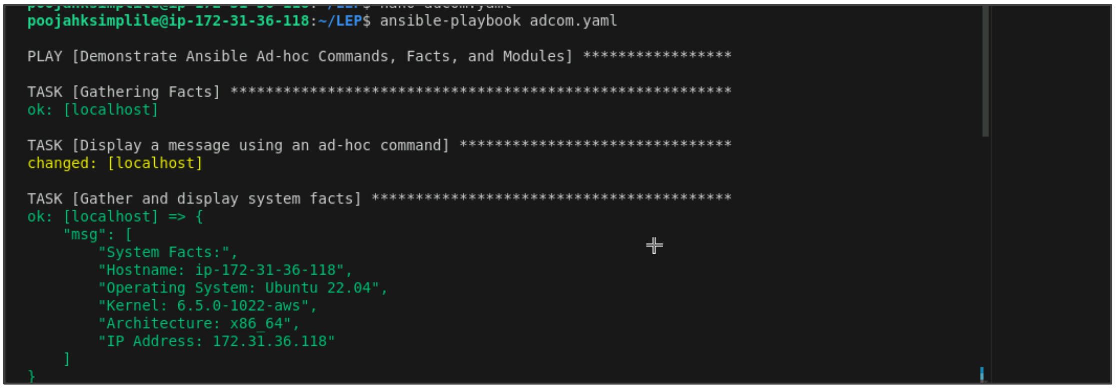
```
---
- name: Demonstrate Ansible Ad-hoc Commands, Facts, and Modules
  hosts: localhost
  gather_facts: yes
  tasks:
    - name: Display a message using an ad-hoc command
      command: echo "Hello from Ansible!"

    - name: Gather and display system facts
      debug:
        msg:
          - "System Facts:"
          - "Hostname: {{ ansible_facts['hostname'] }}"
          - "Operating System: {{ ansible_facts['distribution'] }} {{ ansible_facts['distribution_version>
          - "Kernel: {{ ansible_facts['kernel'] }}"
          - "Architecture: {{ ansible_facts['architecture'] }}"
          - "IP Address: {{ ansible_facts['default_ipv4']['address'] }}"

    - name: Create a directory
      file:
        path: /tmp/ansible_demo
        state: directory
```

Note: To save and close the editor in nano, press Ctrl+X, then Y, and Enter

This Ansible script executes operations such as displaying system facts, managing files, installing software, and controlling services on a local system.

Step 2: Execute the Ansible playbook

2.1 Execute the Ansible playbook using the following command
ansible-playbook adcom.yaml



```
poojahksimplile@ip-172-31-36-118:~/LEP$ ansible-playbook adcom.yaml

PLAY [Demonstrate Ansible Ad-hoc Commands, Facts, and Modules] *****************

TASK [Gathering Facts] *********************************************************
ok: [localhost]

TASK [Display a message using an ad-hoc command] *******************************
changed: [localhost]

TASK [Gather and display system facts] *****************************************
ok: [localhost] => {
    "msg": [
        "System Facts:",
        "Hostname: ip-172-31-36-118",
        "Operating System: Ubuntu 22.04",
        "Kernel: 6.5.0-1022-aws",
        "Architecture: x86_64",
        "IP Address: 172.31.36.118"
    ]
}
```

By following these steps, you have successfully demonstrated how to use Ansible to display system facts, create directories and files, install packages, and manage services on a local machine.