Lesson 04 Demo 04

Implementing Global Variables in Ansible

Objective: To implement global variables in Ansible, define them centrally, and reference them in playbook tasks ensuring consistent variable usage across tasks and hosts

Tools required: Linux terminal

Prerequisites: None

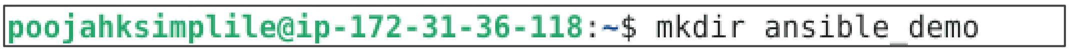
Steps to be followed:

1. Set up the directory structure
2. Create a global variables file
3. Create a playbook
4. Run the playbook

Step 1: Set up the directory structure

1.1 Run the following command to create the main project directory:

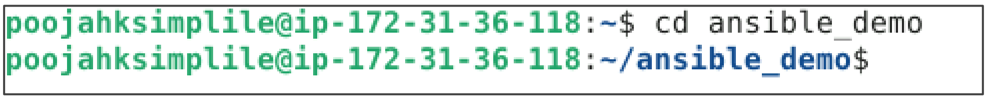
mkdir ansible_demo



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

1.2 Navigate into the project directory using the following command:

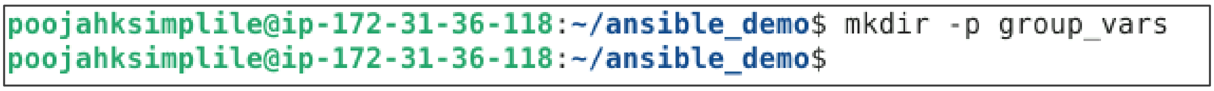
cd ansible_demo



```
poojahksimplile@ip-172-31-36-118:~$ cd ansible_demo
poojahksimplile@ip-172-31-36-118:~/ansible_demo$
```

1.3 Run the following command to create a parent directory:

mkdir -p group_vars

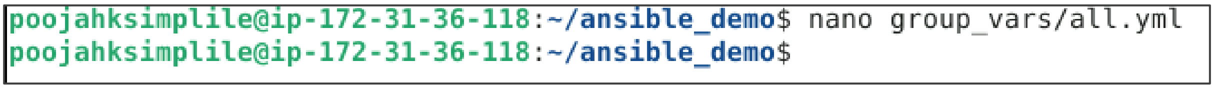


```
poojahksimplile@ip-172-31-36-118:~/ansible_demo$ mkdir -p group_vars
poojahksimplile@ip-172-31-36-118:~/ansible_demo$
```

Step 2: Create a global variables file

2.1 Create and open the **all.yml** file in the **group_vars** directory by using the following command:

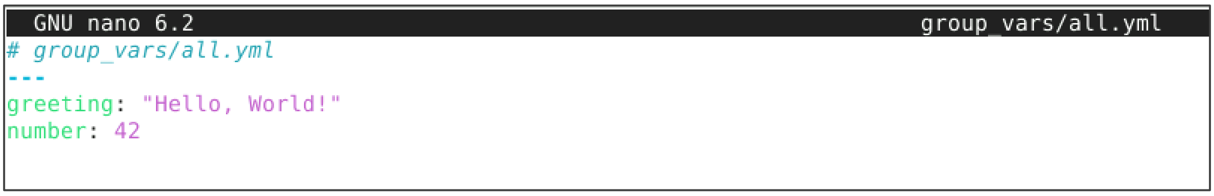
```
nano group_vars/all.yml
```



```
poojahksimplile@ip-172-31-36-118:~/ansible_demo$ nano group_vars/all.yml
poojahksimplile@ip-172-31-36-118:~/ansible_demo$
```

2.2 Add the following script into the **all.yml** file to define global variables:

```
# group_vars/all.yml
---
greeting: "Hello, World!"
number: 42
```

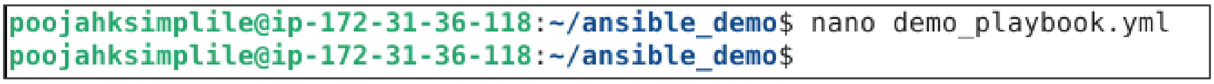


```
  GNU nano 6.2                                               group_vars/all.yml
# group_vars/all.yml
---
greeting: "Hello, World!"
number: 42
```

Step 3: Create a playbook

3.1 Create and open the **demo_playbook.yml** file using the following command:

```
nano demo_playbook.yml
```

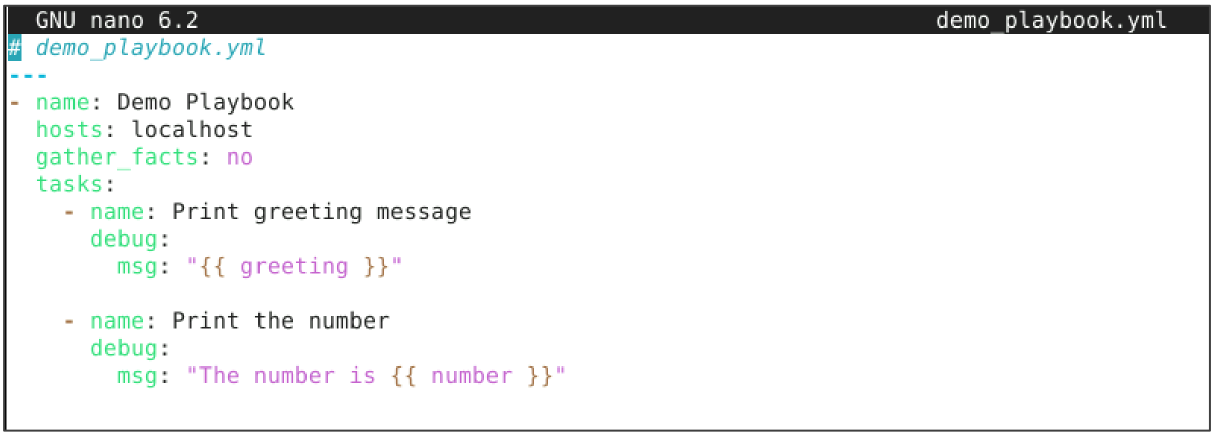


```
poojahksimplile@ip-172-31-36-118:~/ansible_demo$ nano demo_playbook.yml
poojahksimplile@ip-172-31-36-118:~/ansible_demo$
```

3.2 Add the following content to the **demo_playbook.yml** file that references the global variables:

```
# demo_playbook.yml
---
- name: Demo Playbook
  hosts: localhost
  gather_facts: no
  tasks:
    - name: Print greeting message
      debug:
        msg: "{{ greeting }}"
```

```
- name: Print the number
  debug:
    msg: "The number is {{ number }}"
```



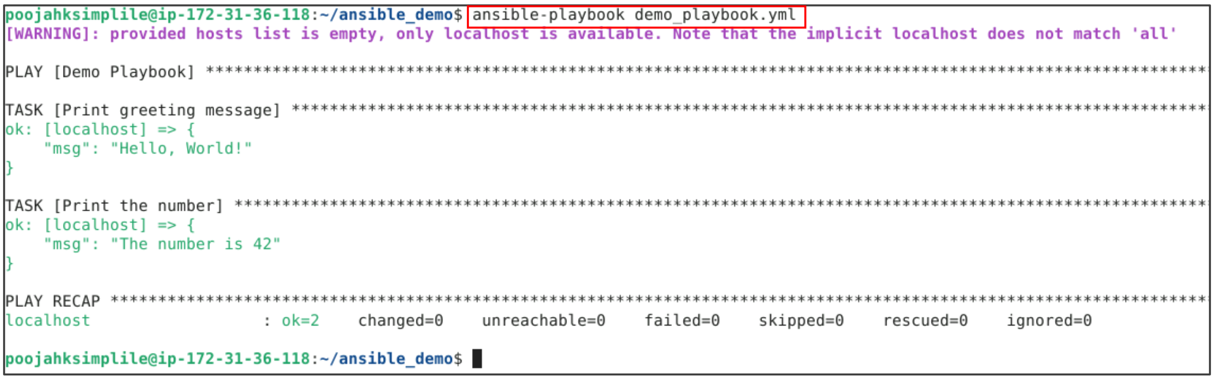
```
  GNU nano 6.2                                              demo_playbook.yml
# demo_playbook.yml
---
- name: Demo Playbook
  hosts: localhost
  gather_facts: no
  tasks:
    - name: Print greeting message
      debug:
        msg: "{{ greeting }}"

    - name: Print the number
      debug:
        msg: "The number is {{ number }}"
```

Step 4: Run the playbook

4.1 Execute the playbook using the following command:

ansible-playbook demo_playbook.yml



```
poojahksimplile@ip-172-31-36-118:~/ansible_demo$ ansible-playbook demo_playbook.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [Demo Playbook] ***********************************************************************************************************

TASK [Print greeting message] **************************************************************************************************
ok: [localhost] => {
    "msg": "Hello, World!"
}

TASK [Print the number] ********************************************************************************************************
ok: [localhost] => {
    "msg": "The number is 42"
}

PLAY RECAP *********************************************************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

poojahksimplile@ip-172-31-36-118:~/ansible_demo$
```

By following these steps, you have successfully implemented global variables in Ansible, defined them centrally and referenced them in playbook tasks, ensuring consistent variable usage across tasks and hosts.