Lesson 06 Demo 03

Implementing Variables in Jinja2 Filters

Objective: To implement variables in Jinja2 filters for processing in Ansible, define the variables, and apply Jinja2 filters in the templates

Tools required: Linux terminal

Prerequisites: None

Steps to be followed:

1. Create a playbook and configure it using Jinja2
2. Create a playbook and configure it using default filters

Step 1: Create a playbook and configure it using Jinja2

1.1 Run the following command to create a file:
sudo nano jinja_template.yml

```
poojahksimplile@ip-172-31-29-173:~$ sudo nano jinja_template.yml
```

1.2 Copy and paste the below script into the jinja_template.yml file:

```
---
 - name: Data Manipulation
   hosts: localhost
   gather_facts: false
   vars:
     my_name: Demo of Ansible Jinja2
   tasks:
   - name: Print message
     debug:
       msg:
        - "My name is {{ my_name }}"
        - "My name is {{ my_name | lower }}"
        - "My name is {{ my_name | upper }}"
        - "My name is {{ my_name | capitalize }}"
        - "My name is {{ my_name | title }}"
```

```
File  Edit  View  Search  Terminal  Help
 GNU nano 6.2                                                jinja_template.yml *
---
- name: Data Manipulation
  hosts: localhost
  gather_facts: false
  vars:
    my_name: Demo of Ansible Jinja2
  tasks:
  - name: Print message
    debug:
      msg:
       - "My name is {{ my_name }}"
       - "My name is {{ my_name | lower }}"
       - "My name is {{ my_name | upper }}"
       - "My name is {{ my_name | capitalize }}"
       - "My name is {{ my_name | title }}"
```

1.3 Run the following command to execute the Ansible playbook named template.yml that uses Jinja2 templates:
ansible-playbook jinja_template.yml

```
poojahksimplile@ip-172-31-29-173:~$ ansible-playbook jinja_template.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [Data Manipulation] ***************************************************************************************************

TASK [Print message] *******************************************************************************************************
ok: [localhost] => {
    "msg": [
        "My name is Demo of Ansible Jinja2",
        "My name is demo of ansible jinja2",
        "My name is DEMO OF ANSIBLE JINJA2",
        "My name is Demo of ansible jinja2",
        "My name is Demo Of Ansible Jinja2"
    ]
}

PLAY RECAP *****************************************************************************************************************
localhost                  : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

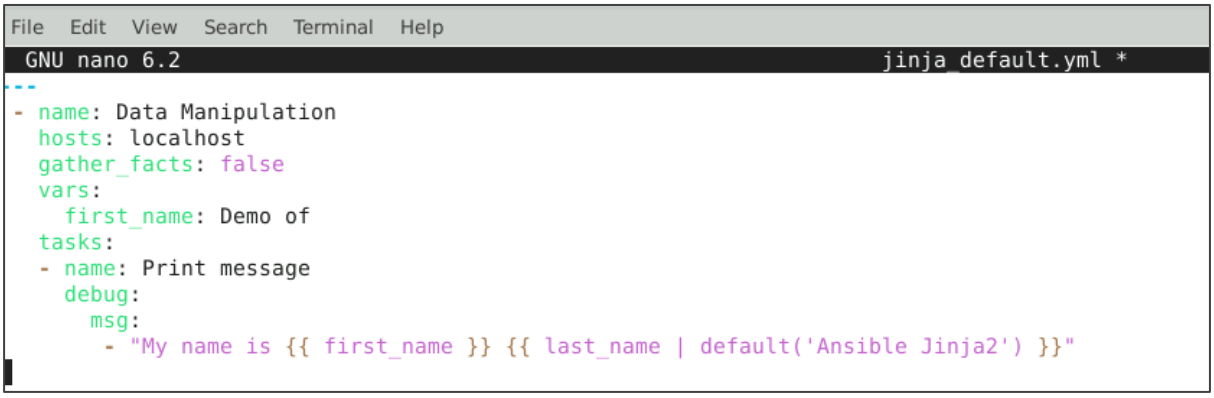
Step 2: Create a playbook and configure it using default filters

2.1 Run the following command to create a new file:
sudo nano jinja_default.yml

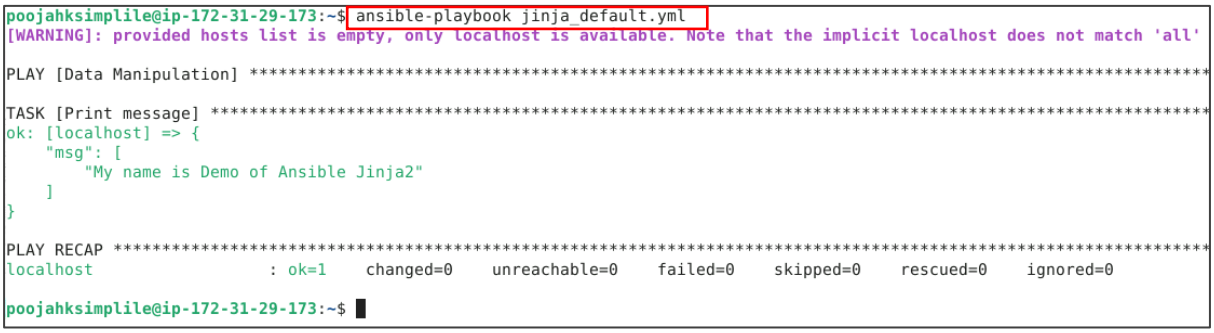
```
poojahksimplile@ip-172-31-29-173:~$ sudo nano jinja_default.yml
poojahksimplile@ip-172-31-29-173:~$
```

2.2 Copy and paste the below script into the jinja_default.yml file:

```
---
 - name: Data Manipulation
   hosts: localhost
   gather_facts: false
   vars:
    first_name: Demo of
   tasks:
   - name: Print message
     debug:
       msg:
        - "My name is {{ first_name }} {{ last_name | default('Ansible Jinja2') }}"
```



2.3 Run the following command to execute the Ansible playbook:
ansible-playbook jinja_default.yml



By following these steps, you have successfully implemented variables in Jinja2 filters for processing in Ansible, defined the variables, and applied Jinja2 filters in the templates.