Lesson-End Project

Templating with Jinja2

Project agenda: To demonstrate the use of Jinja2 templates in Ansible for various tasks, including variable substitution, data manipulation, applying filters, and iterating over lists, by embedding variables within double curly braces

Description: As a systems administrator, you must create Ansible playbooks to demonstrate Jinja2 template features. This project will cover defining variables, data manipulation, using default filters, iterating over lists, and applying filters for pathname, date, and time. You'll work on real-world scenarios to master Jinja2 templates in Ansible for effective configuration management automation.

Tools required: Ansible

Prerequisites: None

Expected deliverables: Upon completing this project, you should be able to automate configuration management tasks using Ansible and Jinja2 templates, including variable definition, data manipulation, and the use of filters. You will also gain proficiency in handling lists, pathnames, dates, and time values within Ansible playbooks.

Steps to be followed:

1. Create a new directory
2. Define a variable inside a playbook using vars
3. Perform data manipulation using filters
4. Use default filters
5. Iterate over the individual values of the list and perform the operation
6. Use filters when dealing with pathnames
7. Use filters for date and time

Step 1: Create a new directory

1.1 Run the following command to create a directory:
mkdir jinja2

```
poojahksimplile@ip-172-31-36-118:~$ mkdir jinja2
poojahksimplile@ip-172-31-36-118:~$
```

1.2 Run the following command to navigate inside the **jinja2** folder:
cd jinja2

```
poojahksimplile@ip-172-31-36-118:~$ cd jinja2
poojahksimplile@ip-172-31-36-118:~/jinja2$
```

Step 2: Define a variable inside a playbook using vars

2.1 Run the following command to create and open a YAML file:
sudo nano jinja2_temp_1.yml

```
poojahksimplile@ip-172-31-36-118:~/jinja2$ sudo nano jinja2_temp_1.yml
poojahksimplile@ip-172-31-36-118:~/jinja2$
```

2.2 Add the following script to the **jinja2_temp_1.yml** file to define variables:

```
---
 - name: Data Manipulation
   hosts: localhost
   gather_facts: false
   vars:
     my_name: ABCD
   tasks:
   - name: Print message
     debug:
       msg: "My name is {{ my_name }}"
```

```
 GNU nano 6.2                          jinja2_temp_1.yml *
---
- name: Data Manipulation
  hosts: localhost
  gather_facts: false
  vars:
    my_name: ABCD
  tasks:
  - name: Print message
    debug:
      msg: "My name is {{ my_name }}"
```

2.3 Use the following command to run the playbook:
ansible-playbook jinja2_temp_1.yml

```
poojahksimplile@ip-172-31-36-118:~/jinja2$ ansible-playbook jinja2_temp_1.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 ABCD"
}

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

poojahksimplile@ip-172-31-36-118:~/jinja2$
```

Step 3: Perform data manipulation using filters

3.1 Run the following command to create and open the file:
sudo nano jinja2_temp_2.yml

```
poojahksimplile@ip-172-31-36-118:~/jinja2$ sudo nano jinja2_temp_2.yml
poojahksimplile@ip-172-31-36-118:~/jinja2$
```

3.2 Add the following script to the **jinja2_temp_2.yml** file for data manipulation using filters:

```
---
 - name: Data Manipulation
   hosts: localhost
   gather_facts: false
   vars:
    my_name: Practicedemo
   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 }}"
```

3.3 Use the following command to run the playbook:

ansible-playbook jinja2_temp_2.yml

```
poojahksimplile@ip-172-31-36-118:~/jinja2$ ansible-playbook jinja2_temp_2.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 Practicedemo",
        "My name is practicedemo",
        "My name is PRACTICEDEMO",
        "My name is Practicedemo",
        "My name is Practicedemo"
    ]
}

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

poojahksimplile@ip-172-31-36-118:~/jinja2$
```

Step 4: Use default filters

4.1 Run the following command to create and open the file:

sudo nano jinja2_temp_3.yml

```
poojahksimplile@ip-172-31-36-118:~/jinja2$ sudo nano jinja2_temp_3.yml
poojahksimplile@ip-172-31-36-118:~/jinja2$
```

4.2 Add the following script to the **jinja2_temp_3.yml** file:

```
---
 - name: Data Manipulation
   hosts: localhost
   gather_facts: false
   vars:
    first_name: Practicedemo
   tasks:
   - name: Print message
     debug:
       msg:
        - "My name is {{ first_name }} {{ last_name }}"
```

```
 GNU nano 6.2                                                jinja2_temp_3.yml *
---
- name: Data Manipulation
  hosts: localhost
  gather_facts: false
  vars:
    first_name: Practicedemo
  tasks:
  - name: Print message
    debug:
      msg:
       - "My name is {{ first_name }} {{ last_name }}"
```

4.3 Use the following command to run the playbook:
ansible-playbook jinja2_temp_3.yml

```
poojahksimplile@ip-172-31-36-118:~/jinja2$ ansible-playbook jinja2_temp_3.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] *******************************************************************************************************************************************
fatal: [localhost]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'last_name' is undefined\n\nThe error
appears to be in '/home/poojahksimplile/jinja2/jinja2_temp_3.yml': line 8, column 6, but may\nbe elsewhere in the file depending on the exact syntax
problem.\n\nThe offending line appears to be:\n\n   tasks:\n   - name: Print message\n     ^ here\n"}

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

poojahksimplile@ip-172-31-36-118:~/jinja2$
```

During playbook execution, you may encounter the error: The task includes an option with an undefined variable. To manage such situations, you can define a default filter for the variable.

4.4 Update the playbook and add a default filter for **last_name**; this ensures that if the variable is not defined, a default value will be used.

```
---
 - name: Data Manipulation
   hosts: localhost
   gather_facts: false
   vars:
    first_name: Practicedemo
   tasks:
   - name: Print message
     debug:
       msg:
```

- "My name is {{ first_name }} {{ last_name | default('XYZ') }}"

```
 GNU nano 6.2                                                    jinja2_temp_3.yml *
---
- name: Data Manipulation
  hosts: localhost
  gather_facts: false
  vars:
    first_name: Practicedemo
  tasks:
  - name: Print message
    debug:
      msg:
       - "My name is {{ first_name }} {{ last_name | default('XYZ') }}"
```

4.5 Use the following command to run the playbook:

ansible-playbook jinja2_temp_3.yml

```
poojahksimplile@ip-172-31-36-118:~/jinja2$ ansible-playbook jinja2_temp_3.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 Practicedemo XYZ"
    ]
}

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

poojahksimplile@ip-172-31-36-118:~/jinja2$
```

Step 5: Iterate over the individual values of the list and perform the operation:

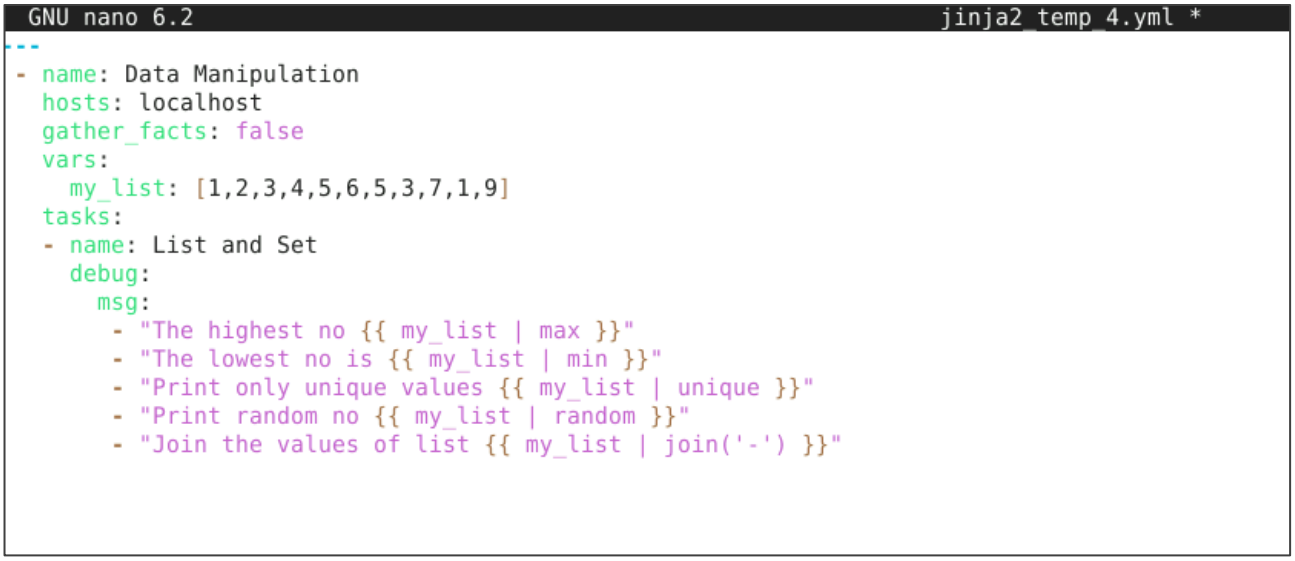
5.1 Run the following command to create and open the file:

sudo nano jinja2_temp_4.yml

```
poojahksimplile@ip-172-31-36-118:~/jinja2$ sudo nano jinja2_temp_4.yml
poojahksimplile@ip-172-31-36-118:~/jinja2$
```

5.2 Add the following script to the **jinja2_temp_4.yml** file to iterate over a list:

```
---
 - name: Data Manipulation
   hosts: localhost
   gather_facts: false
   vars:
    my_list: [1,2,3,4,5,6,5,3,7,1,9]
   tasks:
   - name: List and Set
     debug:
       msg:
        - "The highest no {{ my_list | max }}"
        - "The lowest no is {{ my_list | min }}"
        - "Print only unique values {{ my_list | unique }}"
        - "Print random no {{ my_list | random }}"
        - "Join the values of list {{ my_list | join('-') }}"
```



5.3 Use the following command to run the playbook:
ansible-playbook jinja2_temp_4.yml

```
poojahksimplile@ip-172-31-36-118:~/jinja2$ ansible-playbook jinja2_temp_4.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

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

TASK [List and Set] ********************************************************************************************************************
ok: [localhost] => {
    "msg": [
        "The highest no 9",
        "The lowest no is 1",
        "Print only unique values [1, 2, 3, 4, 5, 6, 7, 9]",
        "Print random no 1",
        "Join the values of list 1-2-3-4-5-6-5-3-7-1-9"
    ]
}

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

poojahksimplile@ip-172-31-36-118:~/jinja2$
```

Step 6: Use filters when dealing with pathnames

6.1 Use the following command to create and open the file:
sudo nano jinja2_temp_5.yml

```
poojahksimplile@ip-172-31-36-118:~/jinja2$ sudo nano jinja2_temp_5.yml
poojahksimplile@ip-172-31-36-118:~/jinja2$
```

6.2 Add the following script to the **jinja2_temp_5.yml** file that uses filters when dealing with pathnames:

```
---
 - name: Data Manipulation
   hosts: localhost
   gather_facts: false
   vars:
     path1: "/opt/custom/data/bin/script.sh"
     path2: 'C:\Users\deeprawat\test.log'
     path3: "~/jinja2_temp_5.yml"
   tasks:
   - name: filters to work on pathnames
     debug:
       msg:
```

- "Linux Path: {{ path1 | dirname }}"
- "Windows Path: {{ path2 | win_dirname }}"
- "Linux script name: {{ path1 | basename }}"
- "Split the path: {{ path2 | win_splitdrive }}"
- "Windows Drive: {{ path2 | win_splitdrive | first }}"
- "Windows File name: {{ path2 | win_splitdrive | last }}"
- "Show Full path: {{ path3 | expanduser }}"

```
 GNU nano 6.2                                                  jinja2_temp_5.yml *
---
- name: Data Manipulation
  hosts: localhost
  gather_facts: false
  vars:
    path1: "/opt/custom/data/bin/script.sh"
    path2: 'C:\Users\deeprawat\test.log'
    path3: "~/jinja2_temp_5.yml"
  tasks:
  - name: filters to work on pathnames
    debug:
      msg:
       - "Linux Path: {{ path1 | dirname }}"
       - "Windows Path: {{ path2 | win_dirname }}"
       - "Linux script name: {{ path1 | basename }}"
       - "Split the path: {{ path2 | win_splitdrive }}"
       - "Windows Drive: {{ path2 | win_splitdrive | first }}"
       - "Windows File name: {{ path2 | win_splitdrive | last }}"
       - "Show Full path: {{ path3 | expanduser }}"
```

6.3 Use the following command to run the playbook:

ansible-playbook jinja2_temp_5.yml

```
poojahksimplile@ip-172-31-36-118:~/jinja2$ ansible-playbook jinja2_temp_5.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

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

TASK [filters to work on pathnames] ****************************************************************************************************
ok: [localhost] => {
    "msg": [
        "Linux Path: /opt/custom/data/bin",
        "Windows Path: C:\\Users\\deeprawat",
        "Linux script name: script.sh",
        "Split the path: ('C:', '\\\\Users\\\\deeprawat\\\\test.log')",
        "Windows Drive: C:",
        "Windows File name: \\Users\\deeprawat\\test.log",
        "Show Full path: /home/poojahksimplile/jinja2_temp_5.yml"
    ]
}

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

poojahksimplile@ip-172-31-36-118:~/jinja2$
```

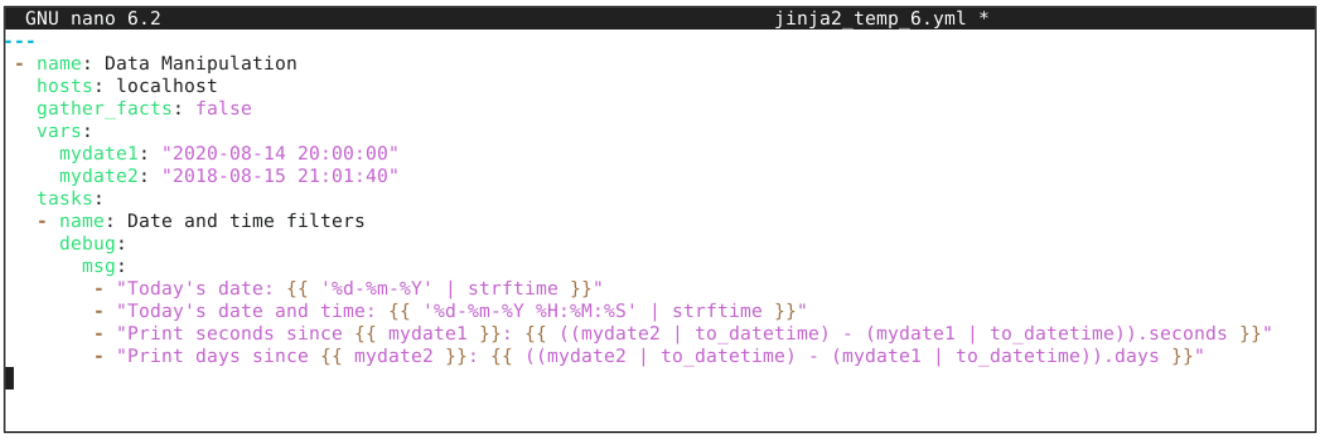
Step 7: Use filters for date and time

7.1 Run the following command to create and open the file:
sudo nano jinja2_temp_6.yml

```
poojahksimplile@ip-172-31-36-118:~/jinja2$ sudo nano jinja2_temp_6.yml
poojahksimplile@ip-172-31-36-118:~/jinja2$
```

7.2 Add the following script to the **jinja2_temp_6.yml** file that uses filters for date and time:

```
---
 - name: Data Manipulation
   hosts: localhost
   gather_facts: false
   vars:
     mydate1: "2020-08-14 20:00:00"
     mydate2: "2018-08-15 21:01:40"
   tasks:
   - name: Date and time filters
     debug:
       msg:
        - "Today's date: {{ '%d-%m-%Y' | strftime }}"
        - "Today's date and time: {{ '%d-%m-%Y %H:%M:%S' | strftime }}"
        - "Print seconds since {{ mydate1 }}: {{ ((mydate2 | to_datetime) - (mydate1 |
to_datetime)).seconds }}"
        - "Print days since {{ mydate2 }}: {{ ((mydate2 | to_datetime) - (mydate1 |
to_datetime)).days }}"
```



7.3 Use the following command to run the playbook:

ansible-playbook jinja2_temp_6.yml

```
poojahksimplile@ip-172-31-36-118:~/jinja2$ ansible-playbook jinja2_temp_6.yml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

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

TASK [Date and time filters] **************************************************************************************************************
ok: [localhost] => {
    "msg": [
        "Today's date: 04-07-2024",
        "Today's date and time: 04-07-2024 06:49:01",
        "Print seconds since 2020-08-14 20:00:00: 3700",
        "Print days since 2018-08-15 21:01:40: -730"
    ]
}

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

poojahksimplile@ip-172-31-36-118:~/jinja2$
```

By following these steps, you have successfully implemented Jinja2 templates in Ansible, demonstrating dynamic content creation and data manipulation. This highlights Jinja2's versatility in automating configuration management tasks within Ansible.