**Lesson 7 Lesson-End Project**

**Templating with Jinja2**

|  |
| --- |
| **Project Agenda:** To use Jinja2 templates in ansible    **Description:** Jinja2 is a very popular and powerful Python-based template engine. It creates HTML, XML, or other markup formats that are returned to the user via an HTTP request.    **Tools required:** Ansible    **Prerequisites:** You must have Ansible installed in the lab to proceed. You can refer to Lesson 2 Demo 1 to install and set up Ansible.    **Expected Deliverables:**  Define a variable inside a playbook  Perform data manipulation  Use default filters  Iterate over the individual values of list  Use filters dealing with pathnames  Use filters for date and time |

**Steps to be followed:**

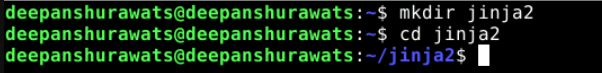
1. Creating a new directory
2. Defining a variable inside a playbook using vars
3. Performing data manipulation using filters
4. Using default filters
5. Iterating over the individual values of list and perform an operation
6. Using filters dealing with pathnames
7. Using filters for date and time

**Step 1: Creating a new directory**

1. Run the following commands to create a new directory:

***mkdir jinja2***

***cd jinja2***



**Step 2: Defining a variable inside a playbook using vars**

1. Use the following command to create and open a yml file:

***sudo nano jinja2\_temp\_1.yml***



1. Add the following code in the yml file:

***---***

***- name: Data Manipulation***

***hosts: localhost***

***gather\_facts: false***

***vars:***

***my\_name: Deepanshu***

***tasks:***

***- name: Print message***

***debug:***

***msg: "My name is {{ my\_name }}"***

Text

Description automatically generated

1. Use the following command to run the Playbook:

***ansible-playbook jinja2\_temp\_1.yml***

Text

Description automatically generated

**Step 3: Performing data manipulation using filters**

1. Use the following command to create and open a yml file:

***sudo nano jinja2\_temp\_2.yml***



1. Add the following code in the yml file:

***---***

***- name: Data Manipulation***

***hosts: localhost***

***gather\_facts: false***

***vars:***

***my\_name: Deepanshu***

***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 }}"***

Text

Description automatically generated

1. Use the following command to run the playbook:

***ansible-playbook jinja2\_temp\_2.yml***

Text

Description automatically generated

**Step 4: Using default filters**

1. Use the following command to create and open a yml file:

***sudo nano jinja2\_temp\_3.yml***



1. Add the following code in the yml file:

***---***

***- name: Data Manipulation***

***hosts: localhost***

***gather\_facts: false***

***vars:***

***first\_name: Deepanshu***

***tasks:***

***- name: Print message***

***debug:***

***msg:***

***- "My name is {{ first\_name }} {{ last\_name }}"***

Text

Description automatically generated

1. Use the following command to run the playbook:

***ansible-playbook jinja2\_temp\_3.yml***

Text

Description automatically generated

When the playbook is executed, we will get the error “The task includes an option with an undefined variable.”

We can handle such situations by defining a default filter with the variable.

1. So, update the playbook and add a default filter with **last\_name** so that if this variable is not defined then the default value will be used:

***---***

***- name: Data Manipulation***

***hosts: localhost***

***gather\_facts: false***

***vars:***

***first\_name: Deepanshu***

***tasks:***

***- name: Print message***

***debug:***

***msg:***

***- "My name is {{ first\_name }} {{ last\_name | default('Rawat') }}"***

Text

Description automatically generated

1. Use the following command to run the playbook:

***ansible-playbook jinja2\_temp\_3.yml***

Text

Description automatically generated

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

1. Use the following command to create and open a yml file:

***sudo nano jinja2\_temp\_4.yml***



1. Add the following code in the yml file:

***---***

***- 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('-') }}"**

Text

Description automatically generated

1. Use the following command to run the playbook:

***ansible-playbook jinja2\_temp\_4.yml***

Text

Description automatically generated

**Step 6: Using filters dealing with pathnames**

1. Usethe following command to create and open yml file:

***sudo nano jinja2\_temp\_5.yml***



1. Add the following code in the yml file:

***---***

***- 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 }}"***

Text

Description automatically generated

1. Use the following command to run the playbook:

***ansible-playbook jinja2\_temp\_5.yml***

Text

Description automatically generated

**Step 7: Using filters for date and time**

1. Usethe following command to create and open a yml file:

***sudo nano jinja2\_temp\_6.yml***



1. Add the following code in the yml file:

***---***

***- 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 }}"***

Text

Description automatically generated

1. Use the following command to run the playbook:

***ansible-playbook jinja2\_temp\_6.yml***

Graphical user interface, text

Description automatically generated

We have successfully implemented Jinja2 templates in ansible.