Ansible Assignment – 2

Note: Attach screenshot for each playbook

1. Environment Setup

- Control Node: Configure an Ansible control node.
- Managed Nodes: Set up three managed hosts with distinct domain names (e.g., node1.example.com, node2.example.com, node3.example.com).
- Configure password-less SSH between the control node and managed hosts.

2. Tasks to Perform with Playbook

Write a playbook named (Variable.yaml)to perform the following tasks using variables where applicable:

1. Create a User:

- Use a variable and create a 3-user. Node1-Sara, Node2-Sam, Node3-Veronica. The variable should be the same for all users
- Example: user_name: Sam, Sara, Veronica

2. Create a File:

- Use a variable file_path to define the location of a file. Example: file_path: /tmp/ansible_file.txt
- Add specific content using a variable file_content. Example: file_content: "Hello from Ansible!"

3. Create a Directory:

- Use a variable directory_path to define the directory location.
- The variable should be the same for all directory_path.
- Example: directory_path: /root/Aarambh-Node1, Node2, Node3

4. Set Permissions:

- Use a variable directory_permissions to define the directory permissions. Example: directory_permissions: '755'
- Use a variable directory_owner to set the owner of the directory. And the owner should be Sara.
- Modify Files Use a variable line_content to add a line to the file. Example: line_content: "Ansible Configuration"

5. Modify Files

- Use variables to add to the below block. Example:
- Content = Hi I am using Ansible. Ansible is a Redhat automation tool.

6. Install a Package using yum:

- Use a variable package_name to define the package to be installed. Example: package_name: ngnix

3. Variable Usage

- Inline Variables: Define simple variables directly within the playbook.
- External Variable File: Store user and file module variables in an external file (e.g., vars/vars.yml) and include them in the playbook.
- Host-specific Variables:
- Create a host_vars directory and define specific variables for each managed host (e.g., different user_name or directory_path for each host).

4. Verification with Ad-hoc Commands for all the above tasks.

Verify that the users were created on the respective nodes as specified:

- On node1.example.com, the user Sara created.
- On node2.example.com, the user Sam created.
- On node3.example.com, the user Veronica created.

- Verify the file /tmp/ansible_file.txt created on node1.example.com, node2.example.com, and node3.example.com.

  Verify the directory created on the respective node as specified.

  - /root/Aarambh-Node1 created for node1.example.com.
  - /root/Aarambh-Node2 created for node2.example.com.
  - /root/Aarambh-Node3 created for node3.example.com.

Verify Directories ownership and permissions

- For node1.example.com: The directory /root/Aarambh-Node1 is owned by Sara with permissions 0755.
- For node2.example.com: The directory /root/Aarambh-Node2 is owned by Sam with permissions 0755.
- For node3.example.com: The directory /root/Aarambh-Node3 is owned by Veronica with permissions 0755.

5. Set a variable from the command line.

1. Create a user named Tom and Jerry on all Manages hosts.

- When we run ansible-playbook Variable.yml -e "user_name=Tom", Ansible will use Tom for the user_name variable, even if a different value is set in host_vars. This is because command-line variables have higher priority and will replace any values set in the playbook or host_vars.
- The output will show Tom in golden, meaning the user was created.
- If We run ansible-playbook Variable.yml -e "user_name=Jerry", the user will be created as Jerry, and the output will show in golden.