1. Ansible utilizes SSH connections to communicate with remote hosts. The Nautilus DevOps team intends to employ a unified Ansible manager for overseeing several remote hosts. To streamline operations, they seek to use a common remote user to connect with all remote hosts.

Update the Ansible configuration file located at /home/thor/ansible-t5q5/ansible-t5q5.cfg on the jump host to set a default remote user as deploy. Please refrain from creating a new configuration file.

Note: This is a sample Ansible configuration. If you intend to test an Ansible playbook using this configuration, you may need to explicitly set the ANSIBLE\_CONFIG variable.

Ans:-

To update the Ansible configuration file located at /home/thor/ansible-t5q5/ansible-t5q5.cfg on the jump host to set a default remote user as deploy, follow these steps:

1. **Open the Configuration File**:
   * Use a text editor like nano or vim to open the configuration file. For example:
   * Vi /home/thor/ansible-t5q5/ansible-t5q5.cfg
2. **Update the Configuration**:
   * Add or update the following line in the [defaults] section of the configuration file:
   * [defaults]
   * remote\_user = deploy
3. **Save and Exit**:
   * Save the changes and exit the text editor. In nano, you can do this by pressing Ctrl+O to save and Ctrl+X to exit.
4. **Set the ANSIBLE\_CONFIG Variable (if needed)**:
   * If you need to explicitly set the ANSIBLE\_CONFIG variable to use this configuration file, you can do so with the following command:
   * export ANSIBLE\_CONFIG=/home/thor/ansible-t5q5/ansible-t5q5.cfg

This will set the default remote user to deploy for your Ansible operations.

===============================================================================

1. Nautilus DevOps team intends to employ a unified Ansible manager for overseeing several remote hosts. To streamline operations, they seek to implement a common Ansible configuration to govern these hosts.

Create an Ansible configuration file under /home/thor/ansible-config directory and disable the SSH host key checking for all Ansible managed hosts.

Created the Ansible configuration file under "home/thor/ansible-config" directory and disabled the SSH host key checking for Ansible?

Ans:-

To create an Ansible configuration file under /home/thor/ansible-config directory and disable SSH host key checking for all Ansible managed hosts, follow these steps:

1. **Create the Directory (if it doesn't exist)**:
2. mkdir -p /home/thor/ansible-config
3. **Create the Ansible Configuration File**:
4. nano /home/thor/ansible-config/ansible.cfg
5. **Add Configuration to Disable SSH Host Key Checking**:
   * Add the following lines to the configuration file:
   * [defaults]
   * host\_key\_checking = False
6. **Save and Exit**:
   * Save the changes and exit the text editor. In nano, you can do this by pressing Ctrl+O to save and Ctrl+X to exit.
7. **Set the ANSIBLE\_CONFIG Variable (if needed)**:
   * If you need to explicitly set the ANSIBLE\_CONFIG variable to use this configuration file, you can do so with the following command:
   * export ANSIBLE\_CONFIG=/home/thor/ansible-config/ansible.cfg

This will create the Ansible configuration file and disable SSH host key checking for all Ansible managed hosts.

=========================================================================

1. a.. On jump host create a playbook /home/thor/ansible/playbook-t2q4.yml to copy /usr/src/itadmin-t2q4/devops-t2q4.txt file on same host at location /opt/itadmin-t2q4 with group and owner should be thor.

Note: Validation will try to run the playbook using command ansible-playbook -i localhost playbook-t2q4.yml so please make sure the playbook works this way without passing any extra arguments.

Ans:-

o create a playbook that copies the file /usr/src/itadmin-t2q4/devops-t2q4.txt to /opt/itadmin-t2q4 with the owner and group set to thor, follow these steps:

1. **Create the Playbook File**:

Vi /home/thor/ansible/playbook-t2q4.yml

* Run the following commands to create the directory and set the appropriate permissions:
* sudo mkdir -p /opt/itadmin-t2q4

sudo chown thor:thor /opt/itadmin-t2q4

1. **Add the Following Content to the Playbook**:
2. ---
3. - name: Copy file to /opt/itadmin-t2q4
4. hosts: localhost
5. tasks:
6. - name: Copy devops-t2q4.txt to /opt/itadmin-t2q4
7. copy:
8. src: /usr/src/itadmin-t2q4/devops-t2q4.txt
9. dest: /opt/itadmin-t2q4/devops-t2q4.txt
10. owner: thor
11. group: thor
12. **Save and Exit**:
    * Save the changes and exit the text editor. In nano, you can do this by pressing Ctrl+O to save and Ctrl+X to exit.
13. **Run the Playbook**:
    * To ensure the playbook works as expected, run the following command:
    * ansible-playbook -i localhost /home/thor/ansible/playbook-t2q4.yml

This playbook will copy the file to the specified location with the correct ownership and group settings

===============================================================================

1. a. On jump host we already have an inventory file /home/thor/ansible/inventory-t2q2.

b. On jump host create a playbook /home/thor/ansible/playbook-t2q2.yml to copy /usr/src/itadmin-t2q2/system-t2q2.txt file to all application servers at location /opt/itadmin-t2q2 with permissions 0600.

Note: Validation will try to run the playbook using command ansible-playbook -i inventory-t2q2 playbook-t2q2.yml so please make sure the playbook works this way without passing any extra arguments.

Ans:-

**Create the Playbook File**:

Vi /home/thor/ansible/playbook-t2q2.yml

Create play book

1. ---
2. - name: Copy file to /opt/itadmin-t2q2 on all application servers
3. hosts: all
4. become: yes
5. tasks:
6. - name: Create /opt/itadmin-t2q2 directory
7. file:
8. path: /opt/itadmin-t2q2
9. state: directory
10. mode: '0755'
11. - name: Copy system-t2q2.txt to /opt/itadmin-t2q2
12. copy:
13. src: /usr/src/itadmin-t2q2/system-t2q2.txt
14. dest: /opt/itadmin-t2q2/system-t2q2.txt
15. mode: '0600'
16. **Run the Playbook**:
    * Run the playbook again with the following command:

ansible-playbook -i /home/thor/ansible/inventory-t2q2 /home/thor/ansible/playbook-t2q2.yml

============================================================================

1. The Nautilus DevOps team is working to create some data on different app servers in using Ansible. They want to create some files/directories and have some specific requirements related to this task. Find below more details about the same:

Create a playbook called playbook-t4q6.yml under /home/thor/playbook/ directory and configure it to create a file called /opt/file-t4q6.txt on all App Servers. The contents of the file must be This file is created by Ansible!. Inventory is already placed under /home/thor/playbook/inventory-t4q6.

Note: Validation will try to run the playbook using command ansible-playbook -i inventory-t4q6 playbook-t4q6.yml, so please make sure the playbook works this way without passing any extra arguments.

Ans:-

To create a playbook that creates a file called /opt/file-t4q6.txt on all App Servers with the content "This file is created by Ansible!", follow these steps:

1. **Create the Playbook File**:
2. Vi /home/thor/playbook/playbook-t4q6.yml
3. **Add the Following Content to the Playbook**:
4. ---
5. - name: Create a file on all App Servers
6. hosts: all
7. become: yes
8. tasks:
9. - name: Create /opt/file-t4q6.txt with specific content
10. copy:
11. dest: /opt/file-t4q6.txt
12. content: "This file is created by Ansible!"
13. **Save and Exit**:
    * Save the changes and exit the text editor. In nano, you can do this by pressing Ctrl+O to save and Ctrl+X to exit.
14. **Run the Playbook**:
    * To ensure the playbook works as expected, run the following command:
    * ansible-playbook -i /home/thor/playbook/inventory-t4q6 /home/thor/playbook/playbook-t4q6.yml

This playbook will create the file with the specified content on all App Servers.

=========================================================================

1. The Nautilus DevOps team is was doing some cleanup work on all app servers in Stratos DC. Instead of doing this manually they want to utilise Ansible for the same. Below are the requirements team has received.

Utilise the inventory file /home/thor/playbook/inventory-t4q4, present on the jump host

Create a playbook name /home/thor/playbook/playbook-t4q4.yml to delete a file named /opt/fruits-t4q4.txt from all App Servers

Note:- Validation will attempt to execute the playbook using the command ansible-playbook -i inventory-t4q4 playbook-t4q4.yml. Please ensure the playbook functions correctly with this command alone, without requiring any additional arguments.

Ans:-

create the Ansible playbook for this task. Here's how you can write the playbook to delete the file /opt/fruits-t4q4.txt from all App Servers:

1. **Create the playbook file**: /home/thor/playbook/playbook-t4q4.yml
2. **Write the following content in the playbook**:

---

- name: Delete file from all App Servers

hosts: all

become: yes

tasks:

- name: Ensure the file /opt/fruits-t4q4.txt is absent

file:

path: /opt/fruits-t4q4.txt

state: absent

This playbook does the following:

* It targets all hosts specified in the inventory file.
* It uses become: yes to ensure the tasks are run with elevated privileges (necessary for file deletion in /opt).
* It includes a task to ensure the file /opt/fruits-t4q4.txt is absent using the file module with state: absent.

To validate the playbook, you can run the command:

ansible-playbook -i /home/thor/playbook/inventory-t4q4 /home/thor/playbook/playbook-t4q4.yml

This should delete the specified file from all the servers listed in your inventory file.

=========================================================================

7 ) The Nautilus Application development team wanted to test some applications on app servers in Stratos Datacenter. They shared some pre-requisites with the DevOps team, and packages need to be installed on app servers. Since they already created some playbooks, now they wanted to make some changes in inventories.

There is an inventory file /home/thor/playbooks/inventory-t3q3 on jump host. It has some aliases named web1, web2 and web3 for three hosts respectively. Update this inventory file to add an alias called db1 for server4.company.com host.

Ans:-

To update the inventory file /home/thor/playbooks/inventory-t3q3 to add an alias called db1 for server4.company.com, you can follow these steps:

1. **Open the inventory file**: /home/thor/playbooks/inventory-t3q3 in a text editor.
2. **Add the alias**: Add the following line to the inventory file to include the new alias.

Here's an example of what the updated inventory file might look like:

[web]

web1 ansible\_host=server1.company.com

web2 ansible\_host=server2.company.com

web3 ansible\_host=server3.company.com

[db]

db1 ansible\_host=server4.company.com

In this example:

* The [web] group contains the existing aliases web1, web2, and web3.
* The [db] group is added to include the new alias db1 for server4.company.com.

Save the changes to the inventory file. This should update the inventory to include the new alias as required.

=========================================================================

8 ) The Nautilus DevOps team aims to test their Ansible playbooks, stored in the /home/thor/playbook/ directory on the jump host, specifically on app server 1 in the Stratos DC. To facilitate this, the team requires the creation of an inventory file to enable Ansible's connection to the specified app server. Below are the outlined requirements:

a. Create an ini type Ansible inventory file /home/thor/playbook/inventory-t3q1 on jump host.

b. Add App Server 1 in this inventory.

c. The inventory hostname of the host should be the server name as per the wiki, for example stapp01 for app server 1.

d. The playbook should seamlessly execute using the command ansible-playbook -i inventory-t3q1 playbook-t3q1.yml --list-hosts.

Ans:-

1. **Create the inventory file**: /home/thor/playbook/inventory-t3q1
2. **Add App Server 1**: with the hostname stapp01.stratos.xfusioncorp.com

Here's how you can do it:

1. Open a terminal on the jump host.
2. Use a text editor to create and edit the inventory file. For example, you can use nano:

vi /home/thor/playbook/inventory-t3q1

3.Add the following content to the file:

[app\_servers]

stapp01 ansible\_host=stapp01.stratos.xfusioncorp.com

4.Save and close the file.

To validate the playbook, you can run the command:

ansible-playbook -i /home/thor/playbook/inventory-t3q1 /home/thor/playbook/playbook-t3q1.yml --list-hosts

This should list the hosts targeted by the playbook, ensuring that the inventory file is correctly set up.

=========================================================================

9) A playbook on the jump host previously functioned correctly. However, a team member recently made modifications that resulted in misconfiguration. Subsequently, when we attempted to execute it, an error occurred. We require someone to review the playbook, identify the issue, and rectify it.

The playbook name is /home/thor/ansible/playbook-t1q6.yml, make sure it executes without any error.

Ans:-

It looks like there are a few issues with the playbook syntax. Let's correct them:

1. **Indentation**: Ensure proper indentation for the tasks section.
2. **Task Name**: Ensure there is a space after the - in the task name.

Here's the corrected version of your playbook:

---

- hosts: localhost

connection: local

tasks:

-name: Read file

shell: cat /etc/hosts

Here's the corrected version of your playbook:

---

- hosts: localhost

connection: local

tasks:

- name: Read file

shell: cat /etc/hosts

n this corrected version:

* The tasks section is properly indented under the hosts and connection sections.
* The task name - name: Read file has a space after the -.
* The shell module is correctly indented under the task name.

Save the changes and try running the playbook again:

ansible-playbook /home/thor/ansible/playbook-t1q6.yml

This should execute without any errors. If you encounter any specific issues,

Once you identify the issue, you can make the necessary corrections. After making the changes, save the file and run the playbook again to test it:

ansible-playbook /home/thor/ansible/playbook-t1q6.yml

=========================================================================

10) We plan to utilize various Ansible modules moving forward. To enhance our familiarity, the team intends to practice commonly used modules by creating playbooks for specific tasks.

Create a playbook named /home/thor/ansible/playbook-t1q4.yml on the jump host. Configure the playbook to generate a file named /tmp/file.txt on the jump host itself. Utilize the copy module and ensure the file contains the content: Welcome to the KKE Tests!

Ans:-

Here's how you can write the playbook to generate a file named /tmp/file.txt with the specified content:

1. **Create the playbook file**: /home/thor/ansible/playbook-t1q4.yml
2. **Write the following content in the playbook**:

---

- name: Create a file with specific content

hosts: localhost

connection: local

tasks:

- name: Create /tmp/file.txt with content

copy:

dest: /tmp/file.txt

content: "Welcome to the KKE Tests!"

This playbook does the following:

* It targets the localhost to run the tasks on the jump host itself.
* It uses the copy module to create the file /tmp/file.txt and ensures it contains the specified content.

Save the playbook and run it using the following command:

ansible-playbook /home/thor/ansible/playbook-t1q4.yml

This should create the file with the desired content on the jump host.

=========================================================================