

TASK 3-DevOps

Question 1. Lets say i have 4 machines consider 1 as ansible master other 3 as nodes, what are the basic setup you need to do for ansible cluster?

Step 1: Install Ansible on the Master Node

Run the following commands on the **Ansible Master**:

```
sudo yum install-y epel-release  
sudo yum install-y ansible
```

Verify the installation:

```
ansible--version
```

Step 2: Set Up SSH Access from Master to Nodes

1. Generate an SSH Key on the Master Node:

```
ssh-keygen-t rsa
```

(Press **Enter** for default location, no passphrase required.)

2. Copy the SSH Key to Each Node:

```
ssh-copy-id user@node1  
ssh-copy-id user@node2  
ssh-copy-id user@node3
```

Replace user with the actual username on the nodes.

3. Test SSH Login Without Password:

```
ssh user@node1
```

If it logs in without asking for a password, SSH is set up correctly.

Step 3: Configure Ansible Inventory

On the **Master Node**, edit the inventory file:

```
sudo nano /etc/ansible/hosts
```

Add the following:

```
[nodes]  
node1 ansible_host=192.168.1.101 ansible_user=user  
node2 ansible_host=192.168.1.102 ansible_user=user  
node3 ansible_host=192.168.1.103 ansible_user=user
```

Save and exit.

Step 4: Test Ansible Connectivity

Run the following command to check if Ansible can communicate with nodes:

```
ansible nodes-m ping
```

Step 5: Run a Sample Ansible Command

To check uptime on all nodes:

```
ansible nodes-m command-a "uptime"
```

Question 2. what are ansible roles? why we need ansible roles? have you worked on ansible galaxy?

What Are Ansible Roles?

Ansible **roles** are a structured way to organize playbooks into reusable components. A role consists of predefined directories and files that help manage configurations efficiently.

Why Do We Need Ansible Roles?

- Promote reusability and modularity.
- Improve maintainability by separating different tasks.
- Simplify complex playbooks into smaller, manageable parts.
- Ensure consistency across deployments.

Working with Ansible Galaxy

Ansible Galaxy is a repository of shared Ansible roles.

Using Ansible Galaxy to Install a Role:

```
ansible-galaxy install geerlingguy.apache
```

Creating a New Role:

```
ansible-galaxy init my_custom_role
```

This creates a directory structure with predefined folders (tasks/, handlers/, vars/, etc.) for role-based execution.

Question 3. What Are Ansible Facts?

Ansible **facts** are system properties and variables collected from managed nodes during execution. These facts provide essential details about a system, such as its OS, hardware, network settings, and more.

Why Are Ansible Facts Useful?

- Enable **dynamic playbook execution** based on system attributes.
- Help in **conditional execution** of tasks.
- Provide system details without manual input.

How to Gather Ansible Facts?

To display all available facts about a node:

```
ansible all-m setup
```

Using Ansible Facts in a Playbook

Example playbook that prints OS details:

```
- hosts: all
  tasks:
  - name: Display OS information
    debug:
      msg: "This server is running {{ ansible_distribution }} {{ ansible_distribution_version }}"
```

Question 4: Can we have a Windows machine as an Ansible master? As a node? Have you worked on any Windows modules? Can you list a few? Do we need any extra configuration?

Windows as Ansible Master?

No, Ansible requires a Linux-based control node. Windows cannot act as an Ansible master.

Windows as an Ansible Node?

Yes, Windows can be an Ansible-managed node with some additional setup.

Extra Configuration Required:

1. Enable WinRM on Windows Nodes:

- Run the following PowerShell script as an administrator:

```
winrm quickconfig
```

```
winrm set winrm/config/service/Auth '@{Basic="true"}'
```

```
winrm set winrm/config/service '@{AllowUnencrypted="true"}'
```

2. Install **pywinrm on the Ansible Master:**

```
pip install pywinrm
```

Common Windows Modules in Ansible:

- `win_ping`: Check connectivity.
- `win_copy`: Copy files.
- `win_service`: Manage Windows services.
- `win_updates`: Install Windows updates.
- `win_user`: Manage Windows users.