**Problem Statement**

You have been asked to:

● Create a new deployment of ansible cluster of 5 nodes. Label 2 nodes as test and other 2 as prod. Install java on test nodes and install mysql-server on prod nodes

Use Ansible roles for the above, group the hosts under test and prod

**Solution Approach**

**Step 1: Create and Setup ec2 instances**

* We will create 5 ec2 instance: 1 Master, 2 Slave nodes for test, 2 Slave nodes for prod
* We will set up the connections such that we can ssh from the Master into all 5 Slave nodes

**Step 2: Setting up Master-Slave architecture**

Used the following set of codes to set up master slave architecture between the Master and 2 Slaves

1. which python3 # check if python is installed

2.

3. # Install Ansible only in the master node

4. sudo yum install -y ansible

5.

6. # Create a new user with the name "ansible" in the MASTER node

7. sudo useradd ansible

8. sudo passwd ansible # Generate a password for the user

9. su – ansible # log in to the user

10.

11. # Create a new user with the name "ansible" in the both slave nodes

12. sudo useradd ansible

13. sudo passwd ansible # Generate a password for the user

14. su - ansible

15.

16. # Provide sudo access to the ansible user in the both the slave nodes

17. cd /etc/ # go to folder

18. sudo vi sudoers # open sudoers in a text editor

19. # add this to wheel: ansible ALL=(ALL) NOPASSWD: ALL

20.

21. # Generate a public and private key in the MASTER node

22. su - ansible # log in to the user

23. ssh-keygen # command to generate keypairs

24.

25. # In order to establish the ssh connection, we need the public IP of the Slave nodes

26. curl ifconfig.me

27.

28. # Run this in SLAVE node to enable ssh port

29. cd /etc/ssh # go to ssh folder

30. sudo vi sshd\_config # the file sshd\_config has details that needs to be changed

31. # Search for the setting "PasswordAuthentication" and change the setting from "no" to "yes"

32. sudo systemctl restart sshd # restart sshd service

33.

34. # Now try gaining remote access to both servers from Master

35. ssh ansible@3.111.149.212 # Slave 1

36. ssh ansible@ 65.0.122.233 # Slave 2

37.

38. # Copy public key from master to remote server

39. ssh-copy-id ansible@{slave public ip}

**Step 3: Creating ansible configuration and inventory file**

1. Creating the configuration file as follows:

1. [defaults]

2. inventory = /etc/ansible/hosts.ini

3. become = True

4. become\_method = sudo

5. become\_user = root

6. fork = 5

7. timeout = 30

1. Creating the inventory file as follows:

1. [test]

2. test1 ansible\_host=3.109.123.221 ansible\_user=ansible

3. test2 ansible\_host=3.109.113.201 ansible\_user=ansible

4.

5. [prod]

6. prod1 ansible\_host=65.0.122.233 ansible\_user=ansible

7. prod2 ansible\_host=65.0.112.220 ansible\_user=ansible

**Step 4: Creating the Ansible Roles**

The 2 Ansible Roles will be created as required in the question

1. sudo ansible-galaxy init /etc/ansible/roles/java # creating role for java

2. sudo ansible-galaxy init /etc/ansible/roles/mysql # creating role for mysql

This is what the directory structure looks like after creating the roles

ansible-roles/

├── inventory

├── site.yml

├── roles/

│ ├── java/

│ │ ├── tasks/

│ │ │ └── main.yml

│ │ └── defaults/

│ │ └── main.yml

│ ├── mysql/

│ ├── tasks/

│ │ └── main.yml

│ └── defaults/

│ └── main.yml

**Step 5: Updating the main.yml files in the respective roles**

1. The *main.yml* file in java role:

1. ---

2. - name: Install Java

3. yum:

4. name: java

5. state: present

6. become: yes

1. The *main.yml* file in msql role:

1. - name: Install MySQL on slave2

2. become: true

3. tasks:

4. - name: Import MySQL Repo

5. shell: sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022

6. - name: Task 2

7. shell: wget http://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm

8. - name: Task 3

9. shell: sudo yum localinstall -y mysql57-community-release-el7-8.noarch.rpm

10. - name:

11. yum:

12. name: mysql-community-server

13. state: present

**Step 6: Creating the Ansible Playbook with the New Roles Defined**

This is the ansible playbook that will be triggered.

1. ---

2. - name: Install Java on test nodes

3. hosts: test

4. roles:

5. - java

6.

7. - name: Install MySQL on prod nodes

8. hosts: prod

9. roles:

10. - mysql

**Step 7: Executing the Ansible Playbook**

The ansible playbook was executed using the below commands:

1. ansible-playbook playbook\_role.yml --syntax-check   # code to check for syntax error

2. ansible-playbook playbook\_role.yml --check          # dry run on terminal

3. ansible-playbook playbook\_role.yml                  # final run on the server