**PreRequisite**:

1. aws credentials are supplied on the AnisbleMaster. You may do it by running ‘aws configure’ command on AnsibleMaster.

**MySQL InnoDB (Three Node) cluster creation**:

1. Install python and Ansible on the AnsibleMaster,

2. Generate a ssh key on AnsibleMaster using ssh-keygen command. This key will be used by AnsibleMaster to install software remote on the nodes.

3. Git clone the code from <https://github.com/munireddy/mysqlClusterWithRoles.git> on AnsibleMaster

3. Provision three nodes on AWS with the a terraform script or anisblescript at <https://github.com/munireddy/ansibleAwsInfra.git>

4. Make sure that mysqlClusterWithRoles/hosts is updated with the internal ips of the nodes created in step #3.

4. Make sure that python and python-apt are installed on all the nodes.

5. Make sure that the public key is copied from the AnisbleMaster is is copied to /home/ubuntu/.ssh/authorized\_keys files. If not AnsibleMaster fails to connect to the nodes.

6. Git clone the code from <https://github.com/munireddy/mysqlClusterWithRoles.git> on AnsibleMaster

7.

------------------------- Verification .. on the first node.---------------------------------

ubuntu@ip-172-31-43-58:~$ mysqlsh

MySQL Shell 8.0.11

Copyright (c) 2016, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type '\help' or '\?' for help; '\quit' to exit.

MySQL JS > dba.verbose=2

2

MySQL JS > shell.connect('inno\_cluster:passw0rd@ip-172-31-43-58:3306')

Creating a session to 'inno\_cluster@ip-172-31-43-58:3306'

Fetching schema names for autocompletion... Press ^C to stop.

Your MySQL connection id is 43

Server version: 8.0.13 MySQL Community Server - GPL

No default schema selected; type \use <schema> to set one.

<ClassicSession:inno\_cluster@ip-172-31-43-58:3306>

MySQL ip-172-31-43-58:3306 ssl JS > var cluster = dba.getCluster('testCluster')

MySQL ip-172-31-43-58:3306 ssl JS > cluster.status()

{

"clusterName": "testCluster",

"defaultReplicaSet": {

"name": "default",

"primary": "ip-172-31-43-58:3306",

"ssl": "REQUIRED",

"status": "OK",

"statusText": "Cluster is ONLINE and can tolerate up to ONE failure.",

"topology": {

"ip-172-31-17-17:3306": {

"address": "ip-172-31-17-17:3306",

"mode": "R/O",

"readReplicas": {},

"role": "HA",

"status": "ONLINE"

},

"ip-172-31-18-251:3306": {

"address": "ip-172-31-18-251:3306",

"mode": "R/O",

"readReplicas": {},

"role": "HA",

"status": "ONLINE"

},

"ip-172-31-43-58:3306": {

"address": "ip-172-31-43-58:3306",

"mode": "R/W",

"readReplicas": {},

"role": "HA",

"status": "ONLINE"

}

}

},

"groupInformationSourceMember": "mysql://inno\_cluster@ip-172-31-43-58:3306"

}