

Advance AWS

AWS Assessment Project 2

Student:

Kishore Shinde

Teacher:

Mrs. Vinolin Jeremiah

Course:

Advance AWS Cloud Computing with DevOps
Fundamentals

Institute:

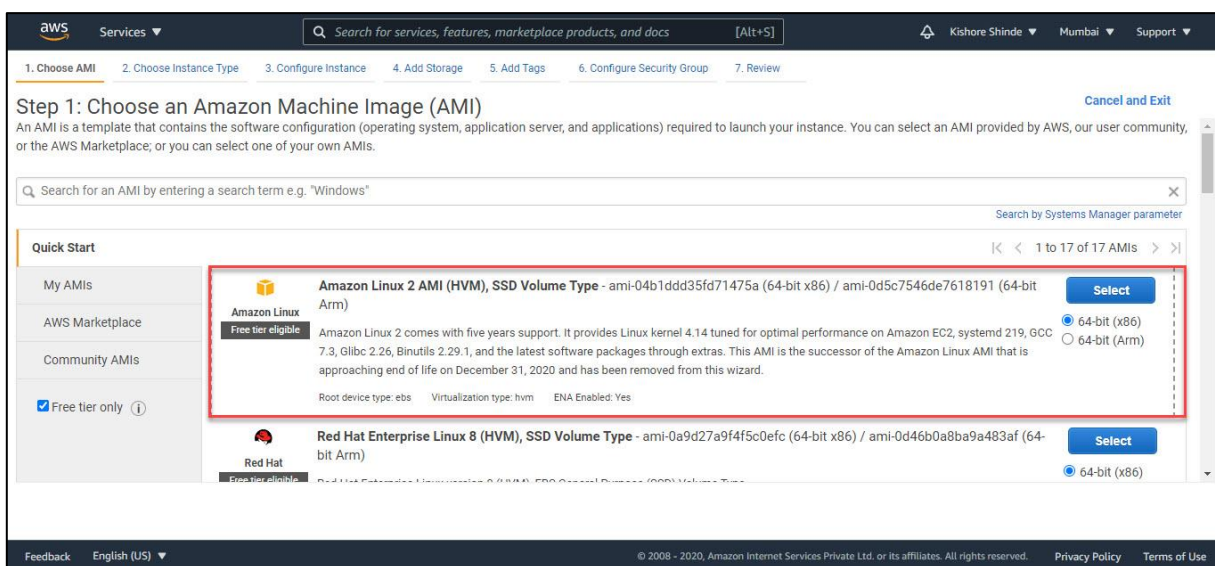
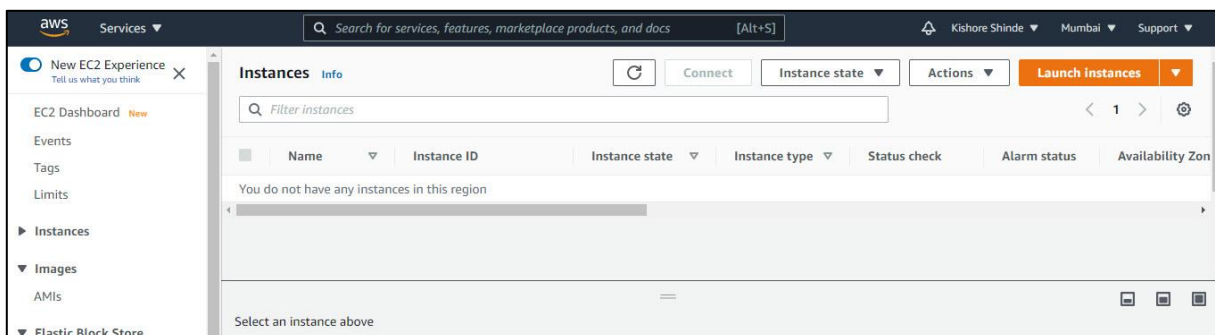
Lets Upgrade

Project: Deploying Amazon RDS Multi-AZ and Read Replica, Simulate Failover

Step 1	Creating an EC2 Instance
Step 2	Creating Security Group for RDS instance
Step 3	Create an Amazon Aurora database with Multi-AZ enabled
Step 4	Connecting to the Aurora (MySQL) database on RDS
Step 5	Connecting the EC2 Server to RDS
Step 6	Execute Database Operations via SSH
Step 7	Forcing a Failover to Test Multi-AZ
Step 8	Testing the Failover Condition

STEP 1: Creating an EC2 Instance

SS1: EC2 Console with instance details



- Instance Selected: Amazon Linux 2 AMI

SS2: User data field showing mysql installation

aws Services Search for services, features, marketplace products, and docs [Alt+S] Kishore Shinde N. Virginia Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances 1 Launch into Auto Scaling Group

Purchasing option ☐ Request Spot instances

Network vpc-4514e238 (default) Create new VPC

Subnet No preference (default subnet in any Availability Zone) Create new subnet

Auto-assign Public IP Enable

Placement group ☐ Add instance to placement group

Capacity Reservation Open

Domain join directory No directory Create new directory

IAM role None Create new IAM role

CPU options ☐ Specify CPU options

Shutdown behavior Stop

Stop - Hibernate behavior ☐ Enable hibernation as an additional stop behavior

Enable termination protection ☐ Protect against accidental termination

Monitoring ☐ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy.

Elastic Inference ☐ Add an Elastic Inference accelerator
Additional charges apply.

Credit specification ☐ Unlimited
Additional charges may apply.

File systems Add file system Create new file system

Advanced Details

Enclave ☐ Enable

Metadata accessible Enabled

Metadata version V1 and V2 (token optional)

Metadata token response hop limit 1

User data ☒ As text ☐ As file ☐ Input is already base64 encoded

```
#!/bin/bash -ex
yum install mysql -y
```

Cancel Previous **Review and Launch** Next: Add Storage

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Auto-assign: Enable
- User data:
 - `#!/bin/bash -ex`
 - `yum install mysql -y`

SS3: Create a Security group with the name MyEC2server-SG

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

[Add Rule](#)

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#)

- Security group name : MyEc2server-SG
- Type: SSH Protocol: TCP Source: Anywhere – 0.0.0.0,::/0

Display Security Group MYEC2server-SG

sg-0ae927943c962a518 - MyEc2server-SG

Details

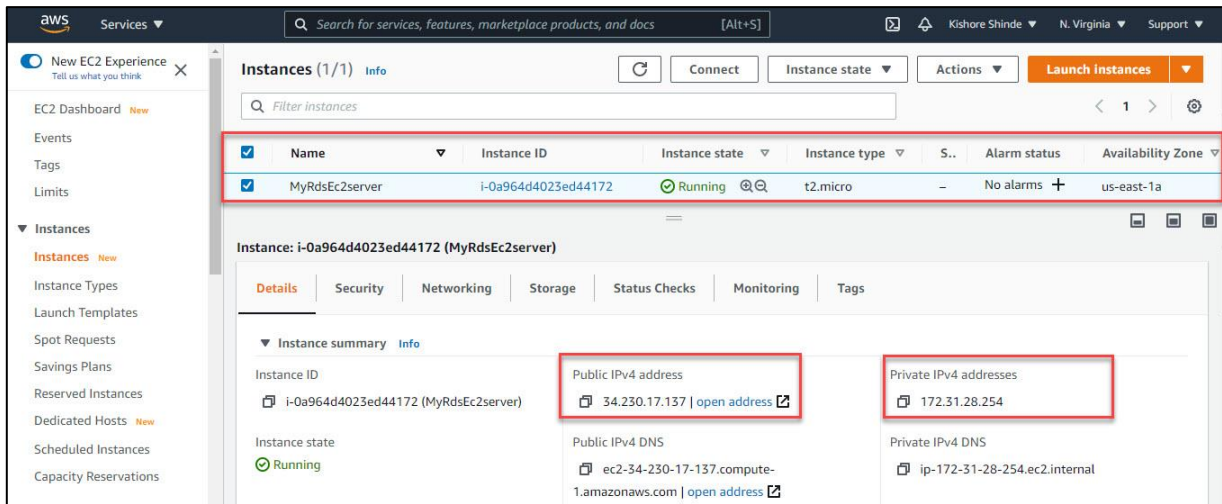
Security group name MyEc2server-SG	Security group ID sg-0ae927943c962a518	Description Security for EC2 server to connect with RDS	VPC ID vpc-afc1dbc7
Owner 391321345174	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	0.0.0.0/0	-
SSH	TCP	22	::/0	-

- Inbound rules:
 - Type: SSH Protocol: TCP Port range: 22 Source : 0.0.0.0/0, ::/0

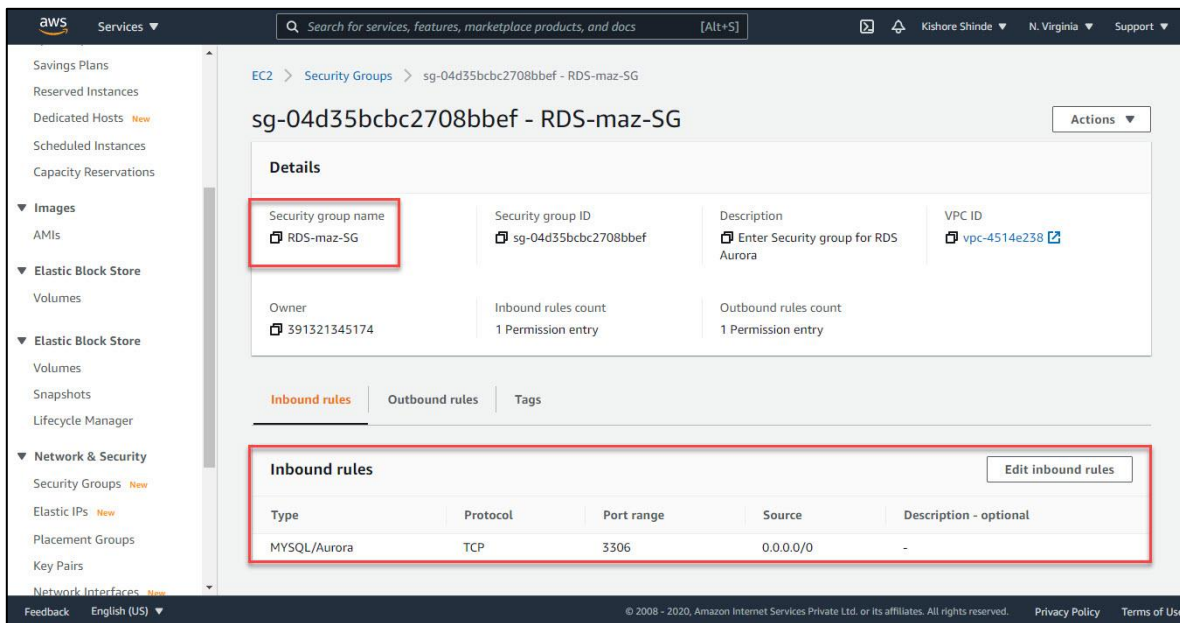
- Created Instance Details:



Sr. No.	Name	Public Ipv4	Private Ipv4
1.	MyRdsEc2Server	34.230.17.137	172.31.28.254

Step 2: Creating Security Group for RDS instance

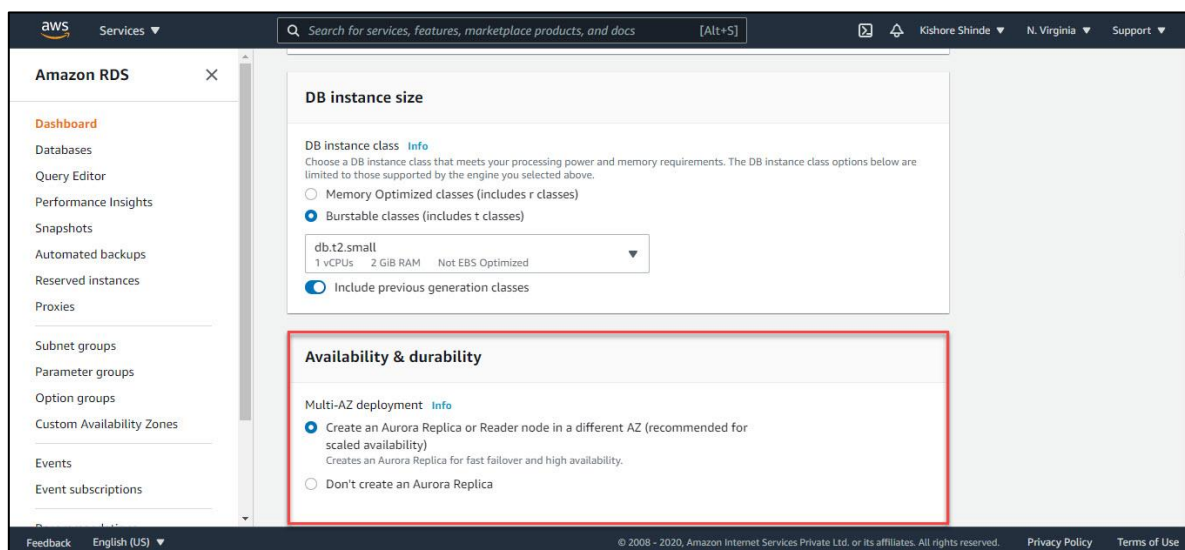
SS4: Display Security group rds-SG information



- Inbound rules:
 - Type: MySQL/Aurora Protocol: TCP Port range: 3306

Step 3: Create an Amazon Aurora database with Multi-AZ enabled

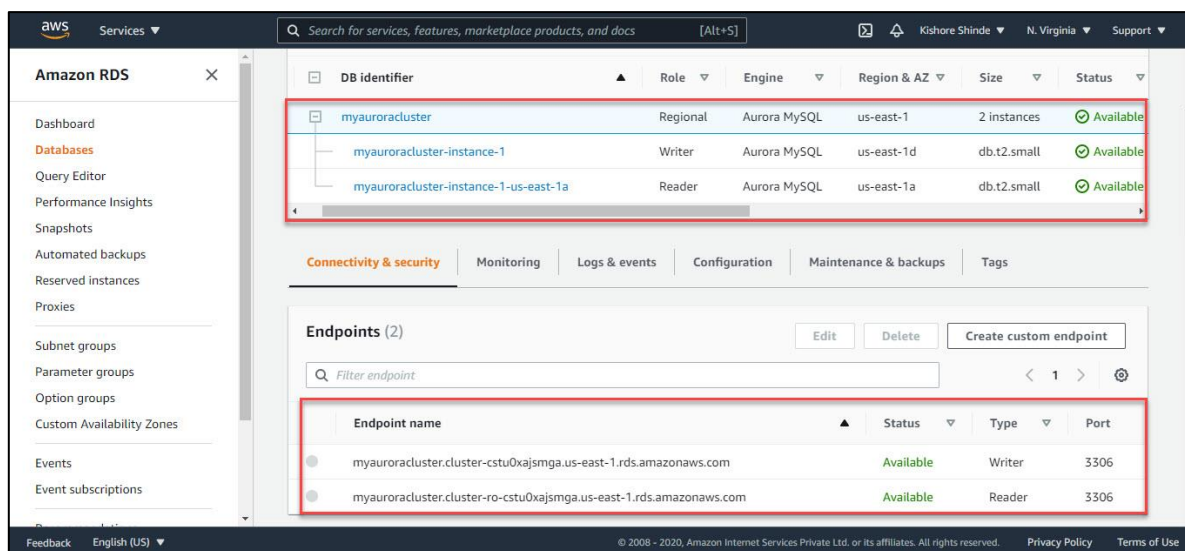
SS5: Display Enabling Multi AZ



- DB instance size:
 - Burstable classes - db.t2.small (1 CPU, 2GB RAM)
- Availability & durability:
 - Multi-AZ Deployment
 - Create an Aurora Replica or Reader node in different AZ

Step 4: Connecting to the Aurora (MySQL) database on RDS

SS6: Show databases console with reader and writer



Sr. No.	DB identifier	Role	Region & AZ
1.	myauroracluster-instance-1	Writer	us-east-1d
2.	myauroracluster-instance-1-us-east-1a	Reader	us-east-1a

Endpoints

S. No.	Endpoint name	Type	Port
1.	myauroracluster.cluster-cstu0xajsmga.us-east-1.rds.amazonaws.com	Writer	3306
2.	myauroracluster.cluster-ro-cstu0xajsmga.us-east-1.rds.amazonaws.com	Reader	3306

Step 5: Connecting the EC2 Server to RDS

SS7: Connect to RDS and show console details

```

_ | ( / Amazon Linux 2 AMI
_ | \ | _ |

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-28-254 ~]$ sudo -su
sudo: option requires an argument -- 'u'
usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] file ...
[ec2-user@ip-172-31-28-254 ~]$ sudo -s
[root@ip-172-31-28-254 ec2-user]# mysql -h myauroracluster.cluster-cstu0xajsmga.us-east-1.rds.amazonaws.com -u labsAdmin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 5.7.12 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>

i-0a964d4023ed44172 (MyRdsEc2server)
Public IPs: 34.230.17.137 Private IPs: 172.31.28.254
```

- SSH into EC2 instance (*MyRdsEc2Server*) Public IP: 34.230.17.137
- Switch to the root user
 - i. Syntax: *sudo -s*
- Log into RDS instance
 - i. Syntax: *mysql -h <hostname> -u <username> -p*
 - ii. **hostname:** *myauroracluster.cluster-cstu0xajsmga.us-east-1.rds.amazonaws.com* (Master(writer)cluster endpoint)
 - iii. **Username:** *labsAdmin*
 - iv. **Password:** *labs1234*

Now you should be able to log into database and execute commands.

Step 6: Execute Database Operations via SSH

SS8: Execute database operation and display details

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 5.7.12 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| whizlabsrds |
+-----+
5 rows in set (0.00 sec)

MySQL [(none)]> create database aurora_db;
Query OK, 1 row affected (0.01 sec)

MySQL [(none)]> use aurora_db;
Database changed
MySQL [aurora_db]> CREATE TABLE students ( subject_id INT AUTO_INCREMENT, subject_name
-> VARCHAR(255) NOT NULL, teacher VARCHAR(255),start_date DATE, lesson
-> TEXT,PRIMARY KEY (subject_id));
Query OK, 0 rows affected (0.03 sec)

MySQL [aurora_db]> INSERT INTO students(subject_name, teacher) VALUES ('English', 'John Taylor');
Query OK, 1 row affected (0.01 sec)

MySQL [aurora_db]> INSERT INTO students(subject_name, teacher) VALUES ('Science', 'Mary Smith');
Query OK, 1 row affected (0.01 sec)

MySQL [aurora_db]> INSERT INTO students(subject_name, teacher) VALUES ('Maths', 'Ted Miller');
Query OK, 1 row affected (0.02 sec)

MySQL [aurora_db]> INSERT INTO students(subject_name, teacher) VALUES ('Arts', 'Suzan Carpenter');
Query OK, 1 row affected (0.01 sec)

MySQL [aurora_db]> select * from students;
+-----+-----+-----+-----+-----+
| subject_id | subject_name | teacher | start_date | lesson |
+-----+-----+-----+-----+-----+
| 1 | English | John Taylor | NULL | NULL |
| 2 | Science | Mary Smith | NULL | NULL |
| 3 | Maths | Ted Miller | NULL | NULL |
| 4 | Arts | Suzan Carpenter | NULL | NULL |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

MySQL [aurora_db]> █
```

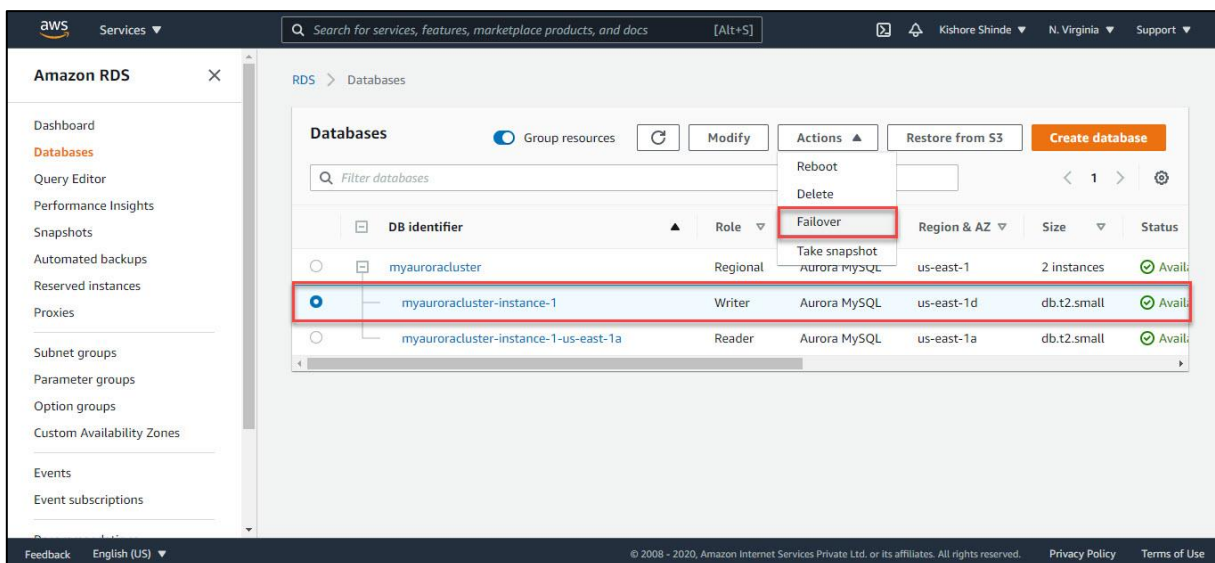
i-0a964d4023ed44172 (MyRdsEc2server)
Public IPs: 34.230.17.137 Private IPs: 172.31.28.254

List of Commands:

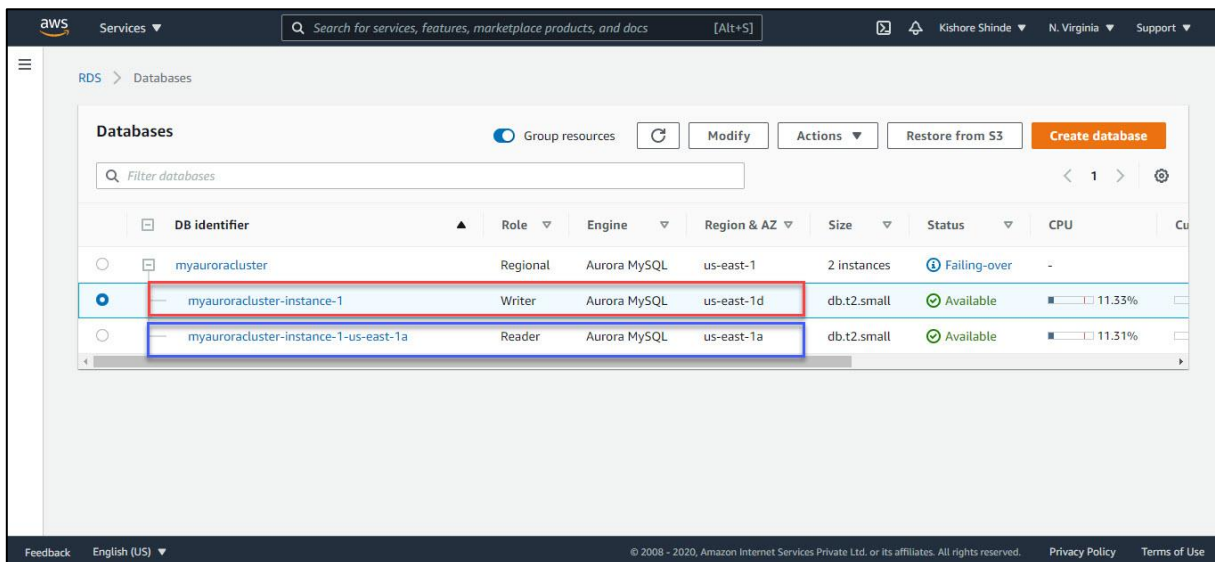
- *show databases* – will display list of databases
- *create database aurora_db* – to create new database
- *use aurora_db* – to use database
- *CREATE TABLE students (subject_id INT AUTO_INCREMENT, subject_name VARCHAR(255) NOT NULL, teacher VARCHAR(255),start_date DATE, lesson TEXT, PRIMARY KEY (subject_id));*
- *INSERT INTO students(subject_name, teacher) VALUES ('English', 'John Taylor');* - insert 2-3 records
- *select * from students* – will display all the records from students table

Step 7: Forcing a Failover to Test Multi-AZ

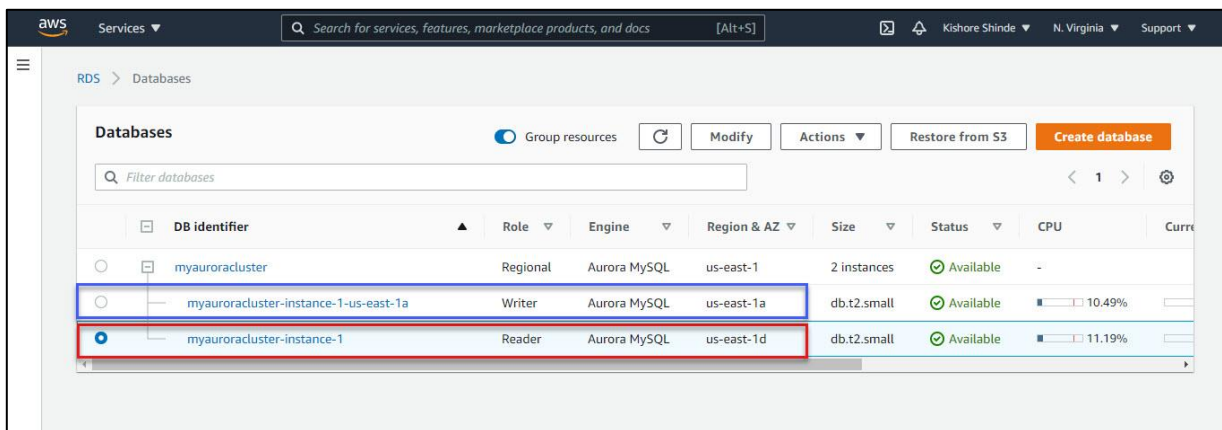
SS9: Show failover in action



- Status: Failing over



- After Failover Master (Writer) becomes Reader and Reader becomes Master (Writer)



Step 8: Testing the Failover Condition

- Connected with new Master Writer after failover

```
Last login: Fri Dec 25 10:10:54 2020 from ec2-18-206-107-26.compute-1.amazonaws.com

 _ _ | _ | _ |
 _ _ | _ | _ | Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-28-254 ~]$ sudo -s
[root@ip-172-31-28-254 ec2-user]# mysql -h myauroracluster.cluster-ro-cstu0xajsmga.us-east-1.rds.amazonaws.com -u labsAdmin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.12 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

i-0a964d4023ed44172 (MyRdsEc2server)
Public IPs: 34.230.17.137 Private IPs: 172.31.28.254

- Database updated with new Master (Writer) after failover

```
| whizlabsrds |
+-----+
6 rows in set (0.01 sec)

MySQL [(none)]> use aurora_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [aurora_db]> INSERT INTO students(subject_name, teacher) VALUES ('Spanish', 'Isabella');
Query OK, 1 row affected (0.01 sec)

MySQL [aurora_db]> select * from students;
+-----+-----+-----+-----+-----+
| subject_id | subject_name | teacher | start_date | lesson |
+-----+-----+-----+-----+-----+
| 1 | English | John Taylor | NULL | NULL |
| 2 | Science | Mary Smith | NULL | NULL |
| 3 | Maths | Ted Miller | NULL | NULL |
| 4 | Arts | Suzan Carpenter | NULL | NULL |
| 5 | Spanish | Isabella | NULL | NULL |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

MySQL [aurora_db]>
```

i-0a964d4023ed44172 (MyRdsEc2server)

Public IPs: 34.230.17.137 Private IPs: 172.31.28.254

Completion and Conclusion

1. In this lab session, first we created EC2 Instance (MyRdsEc2server) with a new security group (MyEC2server-SG)
2. Then we launched an Amazon Aurora RDS DB instance with Multi-AZ enabled
3. Connected to the RDS database instance (using its endpoint) from EC2
4. Instance created in first step
5. Created a test database(aurora_db) and table(students) in your Master RDS DB instance
6. Force the Master DB instance to failover
7. After Failover, Master changed to Reader and Reader changed to Master
8. Connected to the new Master and tested the database replication

xxx---AWS Assessment Project 2 Ends Here--xxx