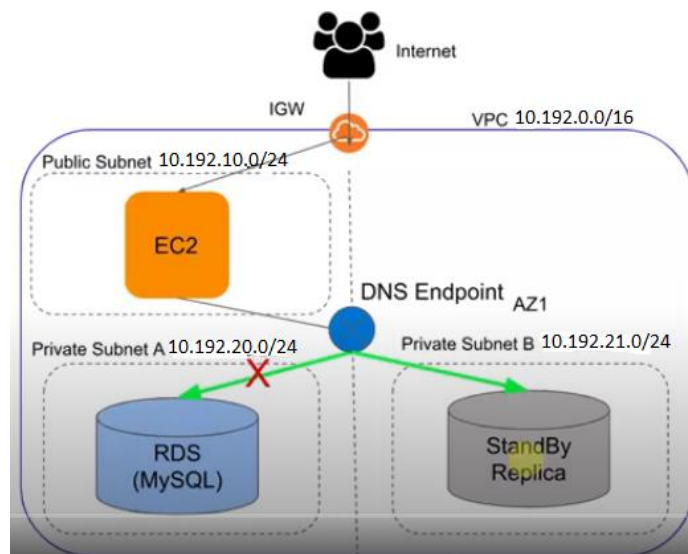


RDS Multi-AZ Failover

In the previous labs, we've configured High Availability across Availability Zones & Regions for the EC2 Instances supporting application workloads. Along with the front-end application servers, it is also important to maintain the High Availability within the DB Instances. High Availability within the DB Instances can be achieved in various ways. In this lab, we are going to configure Highly Availability for the DB Instances across Availability Zones.

It always recommended to launch the DB Instances in Private Subnets with addition of Security Groups with Inbound Rules allowed for the Database Read/Write requests. Front-end application Instances can either be launched in a Private Subnet or Public Subnets.

We are now going to take our previous architecture to next level by deploying below architecture on our AWS Infrastructure.

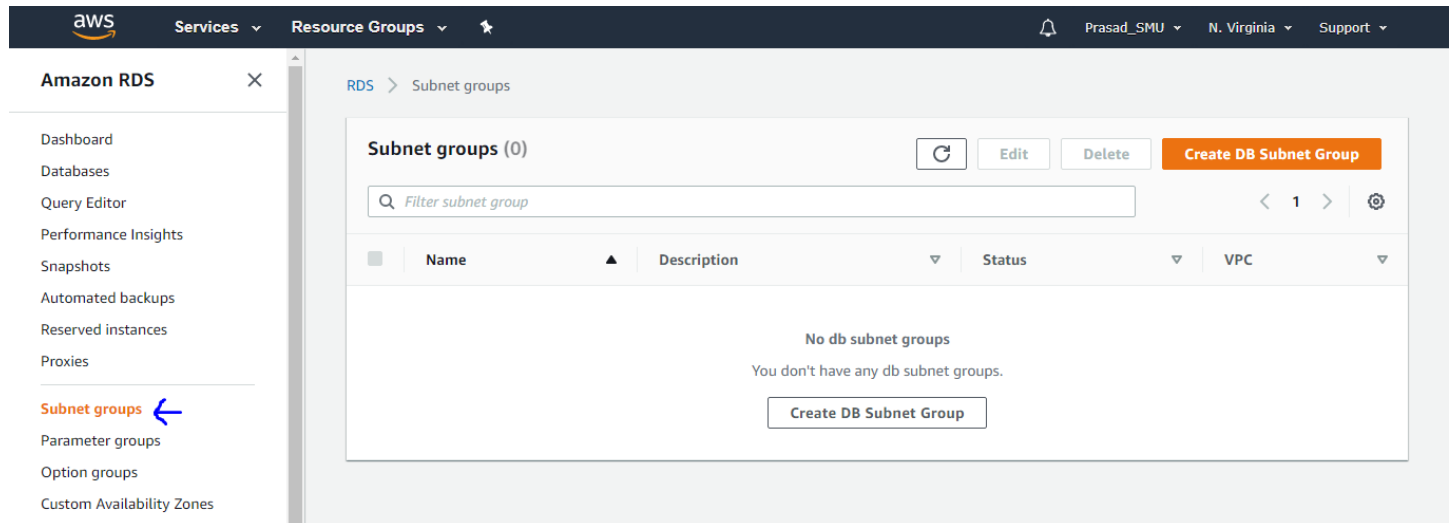


Below is list of Tasks:

- Task 1: Create RDS Subnet Group for Private Subnets (10.192.20.0/24, 10.192.21.0/24)
- Task 2: Create Security Group for RDS Instance
- Task 3: Launch an RDS-MySQL Instance
- Task 4: Launch a Front-End EC2 Instance
- Task 5: Configure the EC2 Instance with MySQL
- Task 6: Connect RDS-MySQL DB Instance to EC2 Instance
- Task 7: MySQL Administration
- Task 8: Test the RDS Failover

Task 1: Create RDS Subnet Group for Private Subnets (10.192.20.0/24, 10.192.21.0/24)

Navigate to Relational Database Service and click on Subnet Group.



Click on Create DB Subnet Group. Give Subnet Group Name of your choice. Select the VPC that we've created in Lab 1.

Subnet group details

Name
You won't be able to modify the name after your subnet group has been created.

Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description

VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

Select the Availability Zones as “us-east-1a” and “us-east-1b” wherein we’ve deployed our entire architecture. Select the Private Subnets from both the Availability Zones and click on Create.

Availability Zones
Choose the Availability Zones that include the subnets you want to add.

Choose an availability zone ▼

us-east-1a ✕ us-east-1b ✕

Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

Select subnets ▼

subnet-053e2caa77fb5cfec (10.192.20.0/24) ✕

subnet-08f8f698c2c4a13d9 (10.192.21.0/24) ✕

Subnets selected (2)

| Availability zone | Subnet ID | CIDR block |
|-------------------|--------------------------|----------------|
| us-east-1a | subnet-053e2caa77fb5cfec | 10.192.20.0/24 |
| us-east-1b | subnet-08f8f698c2c4a13d9 | 10.192.21.0/24 |

Cancel Create

Subnet Group is now finally created.

Subnet groups (1) Refresh Edit Delete Create DB Subnet Group

Filter subnet group < 1 > ⚙

| <input type="checkbox"/> | Name ▲ | Description ▼ | Status ▼ | VPC ▼ |
|--------------------------|------------------------------|----------------------------------|----------|-----------------------|
| <input type="checkbox"/> | subnetgroupforprivatesubnets | Subnet Group for Private Subnets | Complete | vpc-062814d035612343e |

Task 2: Create Security Group for RDS Instance

Navigate to VPC Service, Click on Security Groups and then Click on Create Security Group.

Give the Security Group Name & Description of your choice, select the VPC as Custom VPC and click on Create.

Create security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group fill in the fields below.

Security group name* ⓘ

Description* ⓘ

VPC ⓘ

* Required

Cancel Create

Select the newly created Security Group. At the bottom, click on Inbound Rules and click on Edit Rules.

Filter by tags and attributes or search by keyword |< < 1 to 7 of 7 > >

| <input type="checkbox"/> | Name | Group ID | Group Name | VPC ID |
|-------------------------------------|-----------------------|----------------------|-----------------------|--------------|
| <input type="checkbox"/> | Custom VPC-Default SG | sg-0138fa39c03c2ca04 | default | vpc-062814d0 |
| <input checked="" type="checkbox"/> | RDS-DB Security Group | sg-0234c6165a9ff20e1 | RDS-DB Security Group | vpc-062814d0 |

Security Group: sg-0234c6165a9ff20e1

Description Inbound Rules Outbound Rules Tags

Edit rules

| Type ⓘ | Protocol ⓘ | Port Range ⓘ | Source ⓘ | Description ⓘ |
|----------------------------------|------------|--------------|----------|---------------|
| This security group has no rules | | | | |

Click on Add Rule. Select Rule Type as “MYSQL/Aurora” and Source as “Custom” with Public Subnet 1 CIDR (10.192.10.0/24). In the Public Subnet 1, we are going to deploy a front-end server which will interact with RDS Databases in Private Subnets. Finally, click on Save Rules. Keep Outbound Rule as Default.

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

| Type ⓘ | Protocol ⓘ | Port Range ⓘ | Source ⓘ | Description ⓘ | |
|--------------|------------|--------------|----------|----------------|----------------------------|
| MYSQL/Aurora | TCP | 3306 | Custom | 10.192.10.0/24 | e.g. SSH for Admin Desktop |

Add Rule

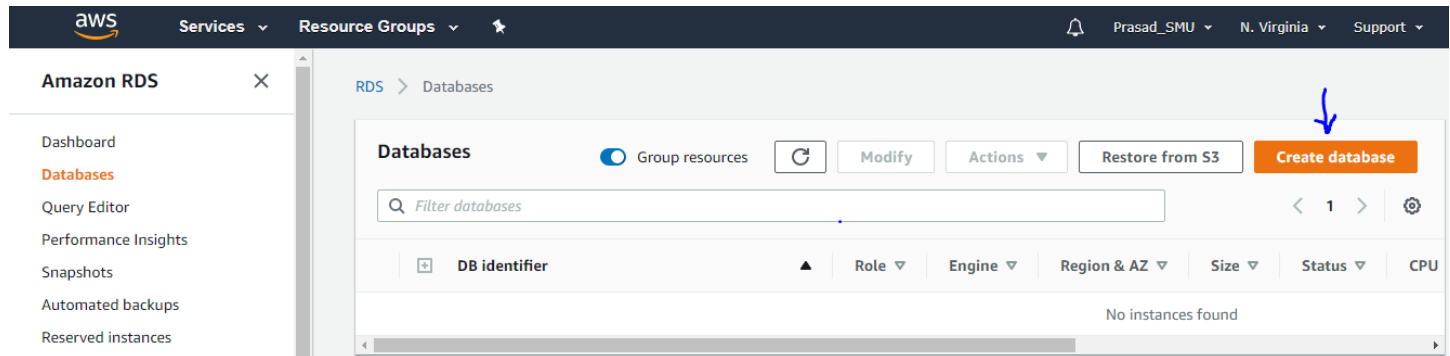
NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

* Required

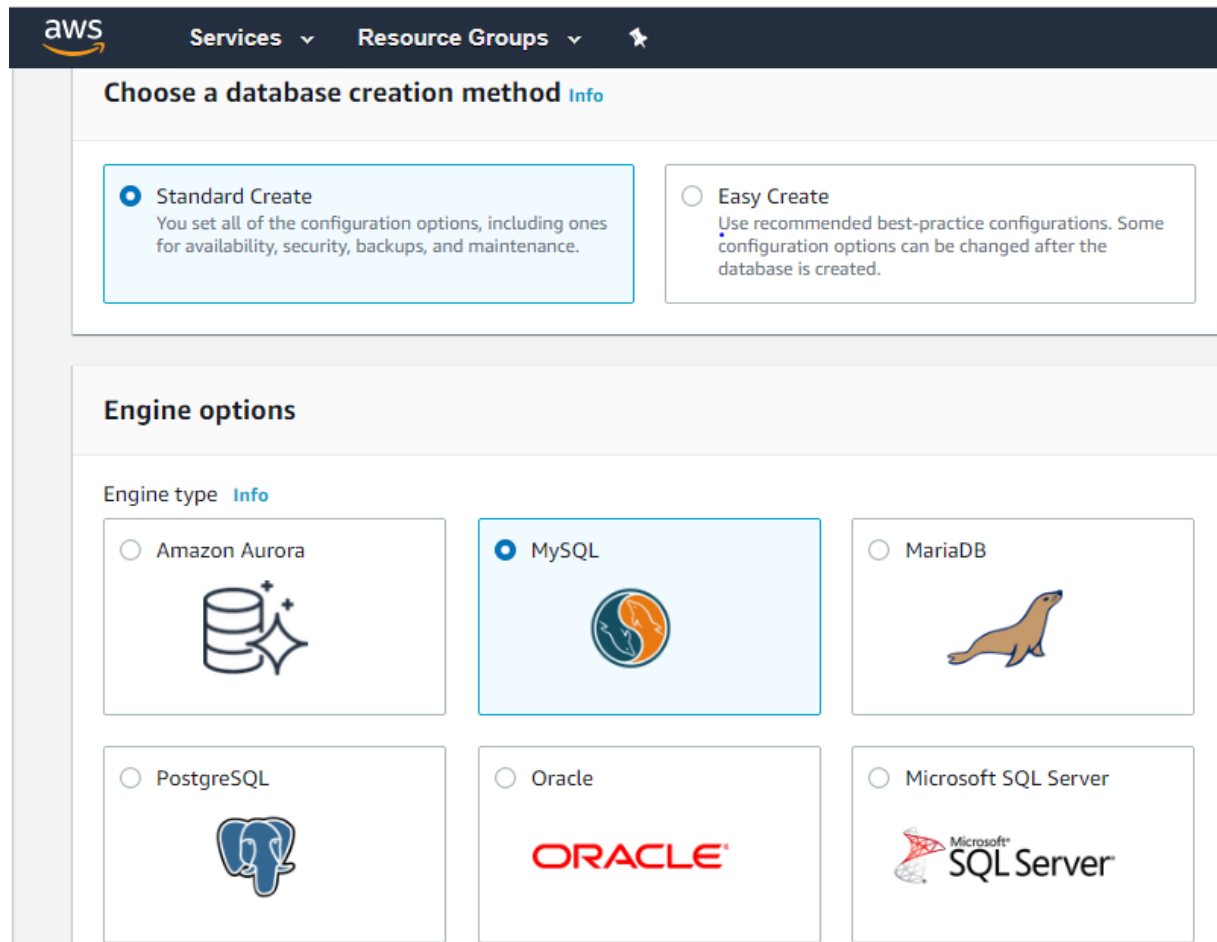
Cancel Save rules

Task 3: Launch an RDS-MySQL Instance

Navigate to Relational Database Service (RDS) and click on Create Database.



Select the Database Creation Method as Standard and select the Database Engine as MySQL.



Then choose Template as Dev/Test.

Templates

Choose a sample template to meet your use case.

☐ **Production**
Use defaults for high availability and fast, consistent performance.

☒ **Dev/Test**
This instance is intended for development use outside of a production environment.

☐ **Free tier**
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
[Info](#)

Under Availability and Durability, select “Create a standby instance”. This creates a replica of the Master DB Instance in another Availability Zone for failover.

Availability & durability

Multi-AZ deployment [Info](#)

☒ **Create a standby instance (recommended for production usage)**
Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.

☐ Do not create a standby instance

Under DB Instance Size, click on Burstable Classes and select “db.t2.micro”.

DB instance size

DB instance class [Info](#)
Choose a DB instance class that meets your processing power and memory requirements. The DB instance class options below are limited to those supported by the engine you selected above.

☐ Standard classes (includes m classes)

☐ Memory Optimized classes (includes r and x classes)

☒ Burstable classes (includes t classes)

db.t2.micro

1 vCPUs 1 GiB RAM Not EBS Optimized

☐ Include previous generation classes

Now give DB Instance Identifier Name, Master Username and Password of your choice. Make a note of Master Username and Password, you'll need this in further tasks.

Settings

DB instance identifier [Info](#)

Type a name for your DB instance. The name must be unique cross all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username [Info](#)

Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter

☐ Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), "(double quote) and @ (at sign).

Confirm password [Info](#)

Under the Connectivity Option, select the Custom VPC. Select the Subnet Group which we've created in the Task 1. Select the Publicly Accessible as NO, since we're deploying DB Instances in the Private Subnets. This is a best practice to consider.

Connectivity

Virtual Private Cloud (VPC) [Info](#)
VPC that defines the virtual networking environment for this DB instance.

Custom VPC (vpc-062814d035612343e) ▼

Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change the VPC selection.

▼ Additional connectivity configuration

Subnet group [Info](#)
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected. •

subnetgroupforprivatesubnets ▼

Publicly accessible [Info](#)

☐ Yes
Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

☒ No
RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

Select the Security Group which we've created in Task 2. Make sure to deselect the Default Security Group.

VPC security group
Choose one or more RDS security groups to allow access to your database. Ensure that the security group rules allow incoming traffic from EC2 instances and devices outside your VPC. (Security groups are required for publicly accessible databases.)

☒ Choose existing
Choose existing VPC security groups

☐ Create new
Create new VPC security group

Existing VPC security groups

Choose VPC security groups ▼

RDS-DB Security Group X ←

Database port [Info](#)
TCP/IP port the database will use for application connections.

3306

Under Additional Configuration, give the Initial Database Name of your Choice.

▼
Additional configuration

Database options, backup enabled, backtrack disabled, Enhanced Monitoring enabled, maintenance, CloudWatch Logs, delete protection disabled

Database options

Initial database name
Info

mydb

If you do not specify a database name, Amazon RDS does not create a database.

Keep rest all the settings as Default.

Finally, review all the settings once again and click on Create Database.

RDS-Database creation process can take up to 5-10 minutes.

Task 4: Launch a Front-End EC2 Instance

Navigate to EC2 Service and Click on Launch Instance.

Resource Groups
Prasad_SMU
N. Virginia
Support

Launch Instance
Connect
Actions

Filter by tags and attributes or search by keyword
1 to 3 of 3

| <input type="checkbox"/> | Name | Instance ID | Instance Type | Availability Zone | Instance State | Status Checks | Alarm Status |
|--------------------------|--------------------|---------------------|---------------|-------------------|----------------|-------------------|--------------|
| <input type="checkbox"/> | Windows Server-IIS | i-0cb4711711c5a18d8 | t2.micro | us-east-1a | running | 2/2 checks passed | None |
| <input type="checkbox"/> | Server 2 | i-060878cf7930ebee6 | t2.micro | us-east-1a | running | 2/2 checks passed | None |
| <input type="checkbox"/> | Server 3 | i-01043fe380368098b | t2.micro | us-east-1b | running | 2/2 checks passed | None |

Select the first, Amazon Linux 2 AMI (HVM).

Free tier eligible

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0323c3dd2da7fb37d (64-bit x86) / ami-0ce2e5b7d27317779 (64-bit Arm)

2/2 checks passed

2/2 checks passed

2/2 checks passed

64-bit (x86)

64-bit (Arm)

Select

Keep the Instance Type as General Purpose-t2.micro and click Next.

Select the Network as Custom VPC. Select the Subnet as Public Subnet 1. Recollect that we've created a Security Group for DB Instances which allows DB traffic from any EC2 Instance in this Subnet (Public Subnet 1). Enable termination protection and click Next.

Network *i* vpc-062814d035612343e | Custom VPC [Create new VPC](#)

Subnet *i* subnet-01ee44283bcd09e5c | Public Subnet 1 | us-e [Create new subnet](#)
248 IP Addresses available

Auto-assign Public IP *i* Use subnet setting (Enable)

Placement group *i* ☐ Add instance to placement group

Capacity Reservation *i* Open [Create new Capacity Reservation](#)

IAM role *i* None [Create new IAM role](#)

Shutdown behavior *i* Stop

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

Enable termination protection *i* ☒ Protect against accidental termination

Monitoring *i* ☐ Enable CloudWatch detailed monitoring
Additional charges apply.

Keep the Storage Settings as Default, click Next. You can Add Tags if you wish, else click Next.

Under Configure Security Group, create a Security Group with the Name of your choice, select Type as SSH, Source as "My IP" and click Review and Launch.

Security group name: SG for front end server

Description: launch-wizard-1 created 2020-04-25T02:25:07.794-05:00

| Type <i>i</i> | Protocol <i>i</i> | Port Range <i>i</i> | Source <i>i</i> | Description <i>i</i> |
|---------------|-------------------|---------------------|-------------------------|----------------------------|
| SSH | TCP | 22 | My IP 70.123.124.218/32 | e.g. SSH for Admin Desktop |

Review the Instance Settings and click on Launch.

Create a new key pair for this EC2 Instance, download this key Pair & save it on your Desktop.

Finally, click on LAUNCH INSTANCE.

Wait for some time, EC2 Instance now has successfully launched.

Filter by tags and attributes or search by keyword

| | Name | Instance ID | Instance Type | Availability Zone | Instance State | Status Checks |
|-------------------------------------|------------------------|---------------------|---------------|-------------------|----------------|-------------------|
| <input type="checkbox"/> | Windows Server-IIS | i-0cb4711711c5a18d8 | t2.micro | us-east-1a | running | 2/2 checks passed |
| <input type="checkbox"/> | Server 2 | i-060878cf7930e8ea6 | t2.micro | us-east-1a | running | 2/2 checks passed |
| <input checked="" type="checkbox"/> | Front End-Linux Server | i-036d96a88689caafa | t2.micro | us-east-1a | running | 2/2 checks passed |
| <input type="checkbox"/> | Server 3 | i-01043fe380368098b | t2.micro | us-east-1b | running | 2/2 checks passed |

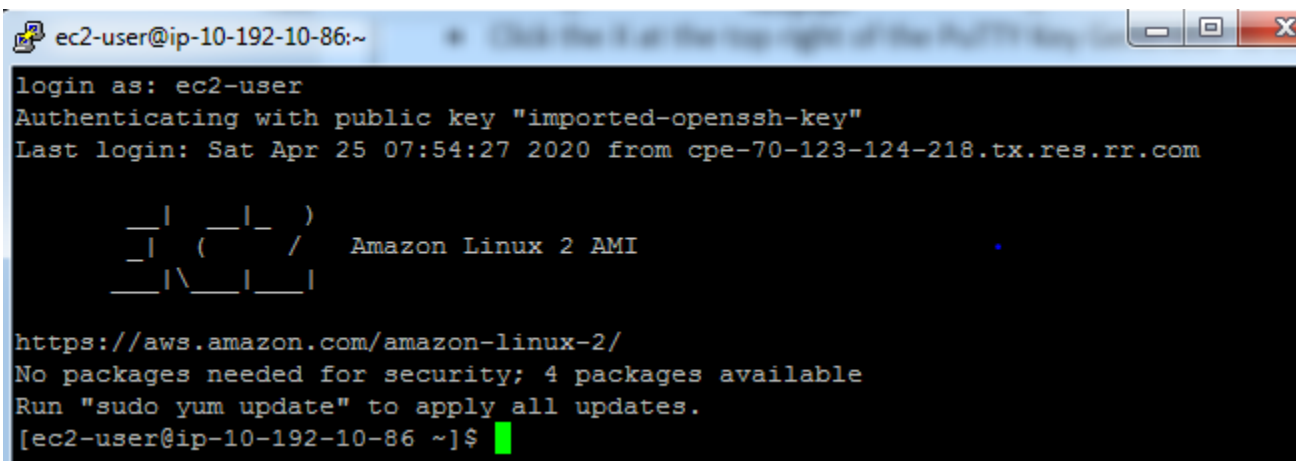
Task 5: Configure the EC2 Instance with MySQL

Convert the pem file which you've downloaded in the previous task, convert it to a ppk file.

- Open puttygen.exe
- In the PuTTY Key Generator panel, choose File > Load private key.
- At the bottom of the Load private key panel, click on the drop-down menu that displays *PuTTY Private Key Files (.ppk) and choose All Files**.
- Still in same panel, browse to the directory where you downloaded the .pem file (for example the Downloads directory).
- Select. pem and click Open.
- A PuTTYgen Notice screen should display, indicating that the key was successfully imported. Click OK.
- Click Save private key, then click Yes to save it without a passphrase.
- Give it the filename and click Save.
- Click the X at the top right of the PuTTY Key Generator to close it.

To connect to your Linux EC2 Instance, follow the below steps.

- Open PuTTY software.
- Give the Hostname as the Public IP Address of the Linux EC2 Instance.
- Click on Connections, then click on SSH, then click on Auth.
- Browse the .ppk file and hit Open.
- For certificates validation, click on YES.
- Provide the username as "ec2-user".



```
ec2-user@ip-10-192-10-86:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
Last login: Sat Apr 25 07:54:27 2020 from cpe-70-123-124-218.tx.res.rr.com  
  
  _ | _ | _ )  
  _ | ( _ | /  Amazon Linux 2 AMI  
  _ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-2/  
No packages needed for security; 4 packages available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-10-192-10-86 ~]$
```

You're Linux EC2 Instance has been successfully launched.

Now install the MySQL client on the Linux EC2 Instance.

Command: Sudo yum install mysql

```
[ec2-user@ip-10-192-10-86 ~]$ sudo yum install mysql
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 2.4 kB 00:00
Resolving Dependencies
--> Running transaction check
---> Package mariadb.x86_64 1:5.5.64-1.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
mariadb x86_64 1:5.5.64-1.amzn2 amzn2-core 9.0 M
Transaction Summary
=====
Install 1 Package

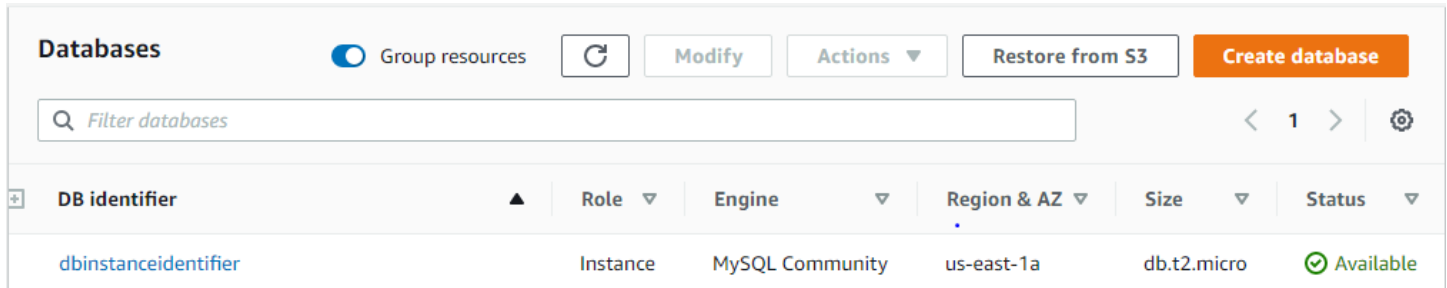
Total download size: 9.0 M
Installed size: 49 M
Is this ok [y/d/N]: y
Downloading packages:
mariadb-5.5.64-1.amzn2.x86_64.rpm | 9.0 MB 00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : 1:mariadb-5.5.64-1.amzn2.x86_64 1/1
Verifying : 1:mariadb-5.5.64-1.amzn2.x86_64 1/1

Installed:
mariadb.x86_64 1:5.5.64-1.amzn2

Complete!
```

Task 6: Connect RDS-MySQL DB Instance to EC2 Instance

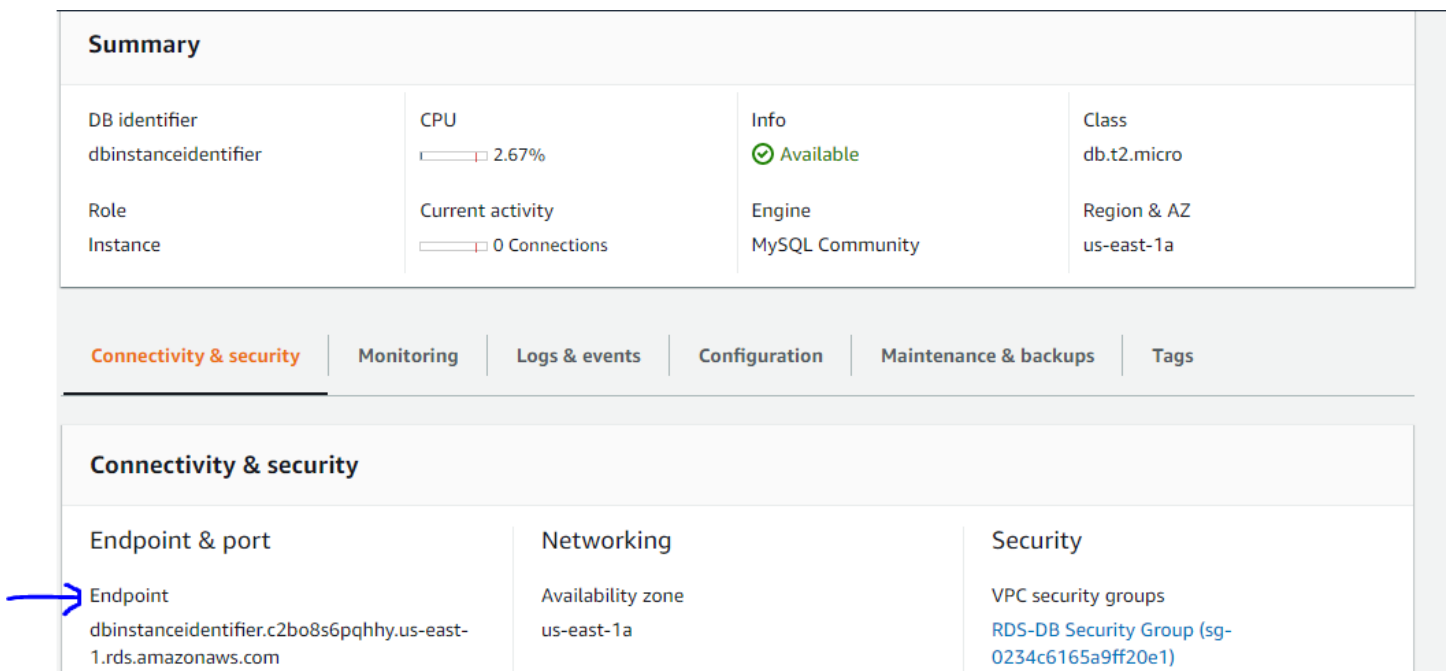
Hopefully now your DB Instance is now Available.



The screenshot shows the AWS RDS 'Databases' console. At the top, there are buttons for 'Group resources', 'Refresh', 'Modify', 'Actions', 'Restore from S3', and 'Create database'. Below these is a search bar labeled 'Filter databases'. A table lists the database instances with columns: DB identifier, Role, Engine, Region & AZ, Size, and Status. One instance is listed: 'dbinstanceidentifier' with Role 'Instance', Engine 'MySQL Community', Region & AZ 'us-east-1a', Size 'db.t2.micro', and Status 'Available' (indicated by a green checkmark).

| DB identifier | Role | Engine | Region & AZ | Size | Status |
|----------------------|----------|-----------------|-------------|-------------|-----------|
| dbinstanceidentifier | Instance | MySQL Community | us-east-1a | db.t2.micro | Available |

Click on the DB Instance and copy the Endpoint.



The screenshot shows the 'Connectivity & security' tab for the database instance. It is divided into three sections: 'Endpoint & port', 'Networking', and 'Security'. The 'Endpoint & port' section shows the 'Endpoint' as 'dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com', which is highlighted by a blue arrow. The 'Networking' section shows the 'Availability zone' as 'us-east-1a'. The 'Security' section shows the 'VPC security groups' as 'RDS-DB Security Group (sg-0234c6165a9ff20e1)'.

| Summary | | | |
|---------------------------------------|-----------------------------------|---------------------------|---------------------------|
| DB identifier dbinstanceidentifier | CPU 2.67% | Info Available | Class db.t2.micro |
| Role Instance | Current activity 0 Connections | Engine MySQL Community | Region & AZ us-east-1a |

| Connectivity & security | | |
|--|---|---|
| Endpoint & port Endpoint dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com | Networking Availability zone us-east-1a | Security VPC security groups RDS-DB Security Group (sg-0234c6165a9ff20e1) |

Go back to the Putty Session and run the following command.

Command:

```
mysql -h dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com -u master -P 3306 -p
```

Give the password that you've set.

```
[ec2-user@ip-10-192-10-86 ~]$ mysql -h dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com -u master -P 3306 -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.7.22-log Source distribution

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)]>
MySQL [(none)]>
MySQL [(none)]>
MySQL [(none)]>
MySQL [(none)]>
MySQL [(none)]>
```

That's it, you've now connected DB Instance to Front End EC2 Instance.

Task 7: MySQL Administration

Now check the existing Databases. Run the following Command.

Command: show databases;

```
MySQL [(none)]> show databases;
+-----+
| Database                |
+-----+
| information_schema       |
| innodb                  |
| mydb                     |
| mysql                   |
| performance_schema      |
| sys                      |
+-----+
6 rows in set (0.00 sec)
```

You'll notice "mydb" database which you specified at the start of RDS DB configurations.

Let's get into the "mydb" database and create tables.

To get into the get into the "mydb" database, run the following command.

Command: use mydb;

```
MySQL [(none)]> use mydb;
Database changed
MySQL [mydb]>
```

To create tables, run the following command.

Command:

create table prasadtraining (NAME VARCHAR(30), BATCH INT(2), PHONE VARCHAR(10));

```
MySQL [mydb]> create table prasadtraining (NAME VARCHAR(30), BATCH INT(2), PHONE VARCHAR(10));
Query OK, 0 rows affected (0.04 sec)

MySQL [mydb]>
```

Enter the values in the table as follows.

Commands:

insert into prasadtraining values ('PRASAD BHAVSAR',1,'1234567899');

insert into prasadtraining values ('SCOTT KINGSLEY',2,'9876543210');

```
MySQL [mydb]> insert into prasadtraining values ('PRASAD BHAVSAR',1,'1234567899');
Query OK, 1 row affected (0.01 sec)

MySQL [mydb]>
MySQL [mydb]> insert into prasadtraining values ('SCOTT KINGSLEY',2,'9876543210');
Query OK, 1 row affected (0.02 sec)

MySQL [mydb]>
```

Let's now verify the Table. Run the below command.

Command: select * from prasadtraining;

```
MySQL [mydb]> select * from prasadtraining;
+-----+-----+-----+
| NAME          | BATCH | PHONE          |
+-----+-----+-----+
| PRASAD BHAVSAR |      1 | 1234567899     |
| SCOTT KINGSLEY |      2 | 9876543210     |
+-----+-----+-----+
2 rows in set (0.00 sec)

MySQL [mydb]>
```

Task 8: Test the RDS Failover

Make a note of your DB Instance's current Availability Zone.

dbinstanceidentifier

ModifyActions ▾

Summary

| | | | |
|---------------------------------------|---|------------------------------|-----------------------------|
| DB identifier dbinstanceidentifier | CPU <div><div></div>2.67%</div> | Info <div>Available</div> | Class db.t2.micro |
| Role Instance | Current activity <div><div></div>0 Connections</div> | Engine MySQL Community | Region & AZ us-east-1a ← |

Come back to PuTTY session, exit from DB by pressing CTRL+Z and run the following command.

Command:

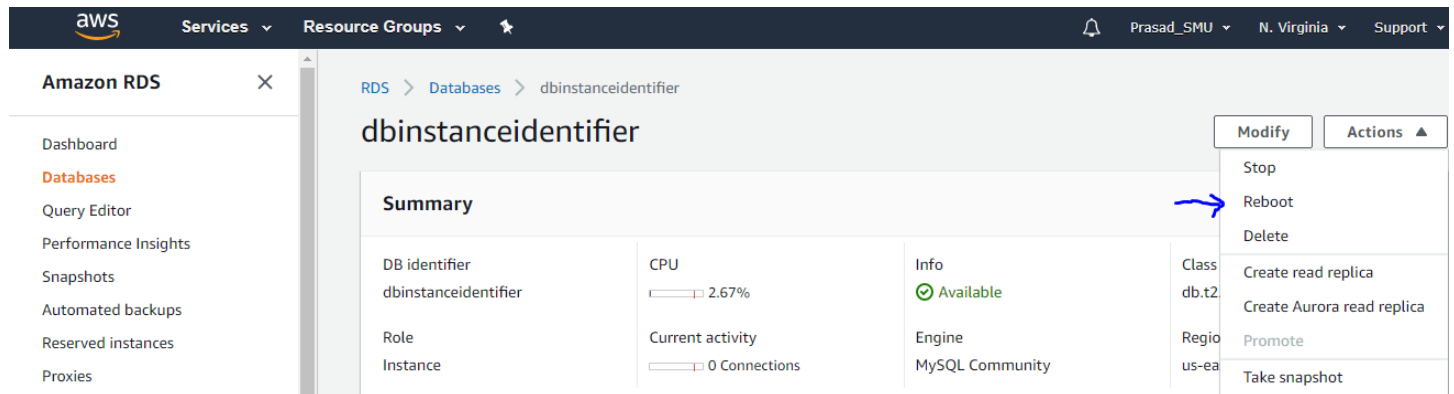
```
while true; do host dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com ; sleep 3; done
```

Replace the highlighted context with the endpoint of your DB Instance.

```
[ec2-user@ip-10-192-10-86 ~]$ while true; do host dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com ; sleep 3; done
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.20.9
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.20.9
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.20.9
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.20.9
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.20.9
```

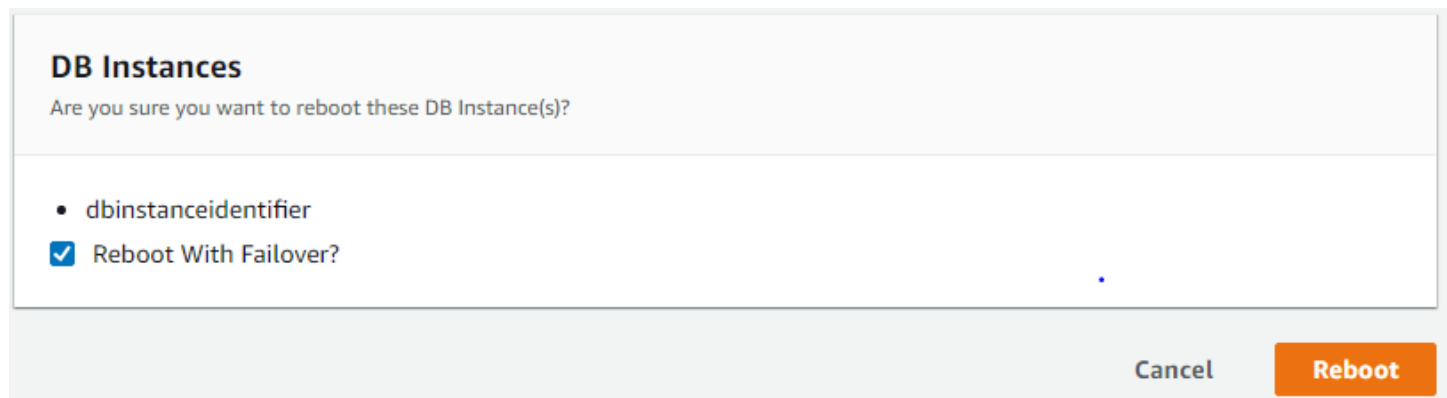
It indicates that the DB Instance for the EC2 Instance is getting served from the subnet 10.192.20.9/24, which is our Private Subnet 1 in Availability Zone (us-east-1a).

Now come back to the RDS console, select the DB Instance. Click on Actions and click on Reboot.



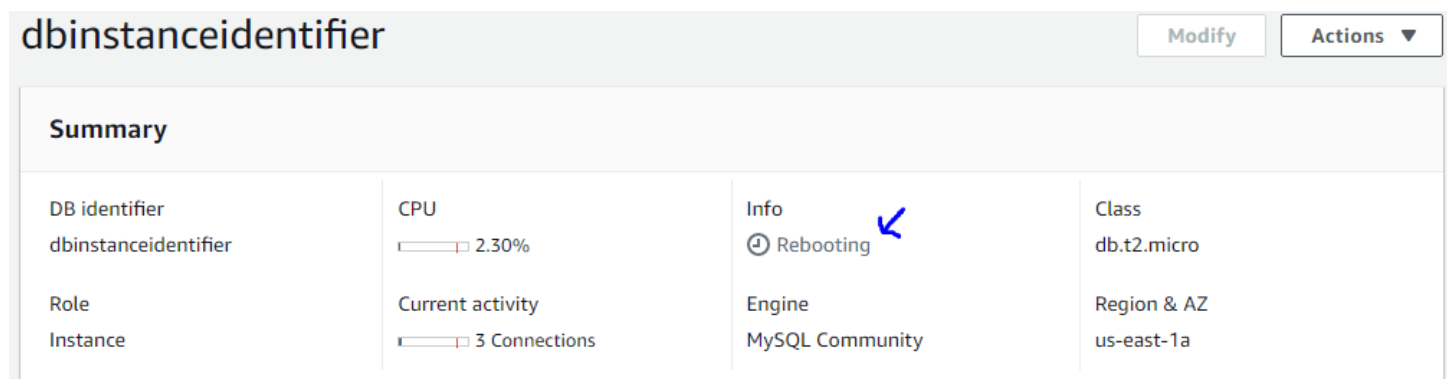
The screenshot shows the AWS RDS console interface. The left sidebar contains navigation links: Dashboard, Databases (selected), Query Editor, Performance Insights, Snapshots, Automated backups, Reserved instances, and Proxies. The main content area displays the details for the DB instance 'dbinstanceidentifier'. The 'Summary' section shows the DB identifier, CPU usage (2.67%), Role, and Instance. The 'Info' section shows the status as 'Available' with a green checkmark, and the 'Engine' as 'MySQL Community'. The 'Actions' dropdown menu is open, showing options: Stop, Reboot (highlighted with a blue arrow), Delete, Create read replica, Create Aurora read replica, Promote, and Take snapshot.

Check Reboot with Failover and Click on Reboot.



The screenshot shows the 'DB Instances' page in the AWS RDS console. The heading 'DB Instances' is followed by the question 'Are you sure you want to reboot these DB Instance(s)?'. Below this, a list of instances is shown, with 'dbinstanceidentifier' selected. The 'Reboot With Failover?' checkbox is checked. At the bottom right, there are 'Cancel' and 'Reboot' buttons. The 'Reboot' button is highlighted with an orange background.

In the Summary, Info status changes to Rebooting.



The screenshot shows the AWS RDS console interface for the DB instance 'dbinstanceidentifier'. The 'Summary' section shows the DB identifier, CPU usage (2.30%), Role, and Instance. The 'Info' section shows the status as 'Rebooting' with a blue arrow pointing to it, and the 'Engine' as 'MySQL Community'. The 'Class' is 'db.t2.micro' and the 'Region & AZ' is 'us-east-1a'.

Now immediately click on Events on the Right-Hand Side and look for the Multi-AZ Instance Failover Events. You'll notice that the Multi-AZ Failover has been started and completed.

Amazon RDS

- Dashboard
- Databases
- Query Editor
- Performance Insights
- Snapshots
- Automated backups
- Reserved instances
- Proxies
- Subnet groups
- Parameter groups
- Option groups
- Custom Availability Zones
- Events**
- Event subscriptions

RDS > Events

Events (41)

Filter events

| Source | Type | Time | Message |
|---|-----------|----------------------------------|---|
| dbinstanceidentifier | Instances | Sat Apr 25 03:45:37 GMT-500 2020 | <u>Multi-AZ instance failover completed</u> |
| dbinstanceidentifier | Instances | Sat Apr 25 03:45:08 GMT-500 2020 | DB instance restarted |
| dbinstanceidentifier | Instances | Sat Apr 25 03:45:02 GMT-500 2020 | Multi-AZ instance failover started. |
| dbinstanceidentifier | Instances | Sat Apr 25 02:49:37 GMT-500 2020 | Finished updating DB parameter group |
| dbinstanceidentifier | Instances | Sat Apr 25 02:48:35 GMT-500 2020 | Finished DB Instance backup |
| rds:dbinstanceidentifier-2020-04-25-07-46 | Snapshots | Sat Apr 25 02:48:34 GMT-500 2020 | Automated snapshot creation |
| rds:dbinstanceidentifier-2020-04-25-07-46 | Snapshots | Sat Apr 25 02:46:31 GMT-500 2020 | Creating automated snapshot |
| dbinstanceidentifier | Instances | Sat Apr 25 02:46:31 GMT-500 2020 | Backing up DB instance |

Events (40)

Filter events

| Type | Time | Message |
|-----------|----------------------------------|--|
| Instances | Sat Apr 25 03:45:37 GMT-500 2020 | Multi-AZ instance failover completed ← |
| Instances | Sat Apr 25 03:45:08 GMT-500 2020 | DB instance restarted |
| Instances | Sat Apr 25 03:45:02 GMT-500 2020 | Multi-AZ instance failover started. ← |

Now check the DB Instance Availability Zone. You'll notice that it has been changed to another Availability Zone (us-east-1b).

dbinstanceidentifier

ModifyActions ▼

Summary

| | | | |
|---------------------------------------|--|---------------------------|-----------------------------|
| DB identifier dbinstanceidentifier | CPU <div>2.83%</div> | Info Available | Class db.t2.micro |
| Role Instance | Current activity <div>0 Connections</div> | Engine MySQL Community | Region & AZ us-east-1b ← |

Also, come back to PuTTY now.

```
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.20.9
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.20.9
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.20.9
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.20.9
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.20.9 ←
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.21.86
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.21.86
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.21.86
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.21.86
dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com has address 10.192.21.86
```

You'll observe that the DB Instance for the EC2 Instance is now getting served from the subnet 10.192.21.9/24, which is our Private Subnet 2 in Availability Zone (us-east-1b).

This proves that Failover has been successfully achieved.

Now let's check whether we lost any data or not?

Comeback to PuTTY session, stop the trace (CTRL+Z).

Login to your DB Instance and check the tables which you've created earlier.

```
[ec2-user@ip-10-192-10-86 ~]$ mysql -h dbinstanceidentifier.c2bo8s6pqhhy.us-east-1.rds.amazonaws.com -u master -P 3306 -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.7.22-log Source distribution

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)]> use mydb;
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 [mydb]> select * from prasadtraining;
+-----+-----+-----+
| NAME          | BATCH | PHONE      |
+-----+-----+-----+
| PRASAD BHAVSAR |      1 | 1234567899 |
| SCOTT KINGSLEY |      2 | 9876543210 |
+-----+-----+-----+
2 rows in set (0.01 sec)

MySQL [mydb]>
```

All data in the table is available, Failover has been successfully tested!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

I hope you have now gained better understanding of RDS Multi-AZ Failover.

For Questions, contact me on pbhavsar@smu.edu.