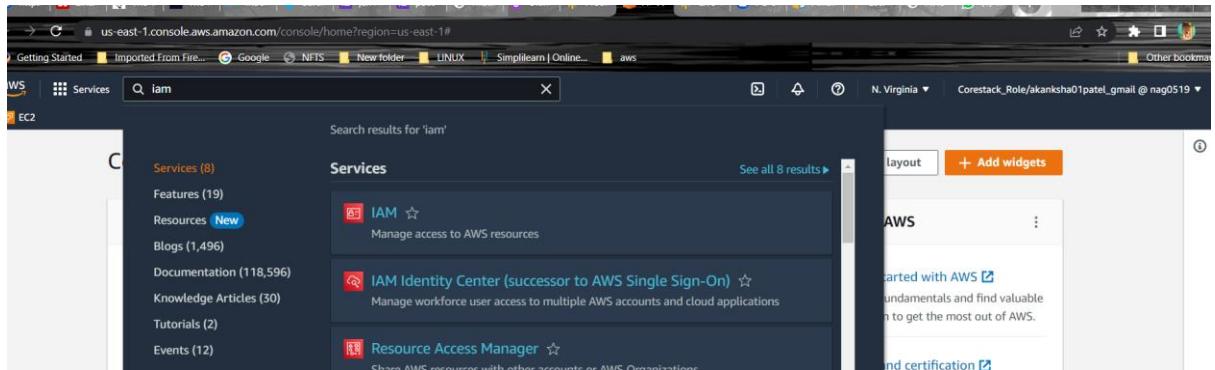


## Configure and connect a MySQL Database Instance with a Web Server

To begin with any of the activity we first need to set up IAM i.e., permissions to the users on the AWS resources. So, we will provide RDB full access and EC2instance access to our user.

### i. Navigate to the AWS lab and search for IAM:



1. Select IAM.
2. IAM dashboard appears.
3. We create a user to whom we will assign Policies.
4. As IAM Policies are global so we don't need to worry of the region we have selected currently.

### ii. Creating a user:

1. Click on Users on left side of IAM dashboard.

# Identity and Access Management (IAM)

Unable to load search

## Dashboard

### ▼ Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

### ▼ Access reports

2. Click on Add Users.

The screenshot shows the AWS IAM Users list interface. At the top, there is a blue banner with the text "Introducing the new Users list experience" and "We've redesigned the Users list experience to make it easier to use. Let us know what you think." Below the banner, the page title is "IAM > Users". A sub-header says "Users (1) Info" and describes an IAM user as "An IAM user is an identity with long-term credentials that is used to interact with AWS in an account." On the right side of the header, there are "Add users" and "Delete" buttons. Below the header, there is a search bar with the placeholder "Find users by username or access key". Underneath the search bar is a table with the following columns: "User name", "Groups", "Last activity", "MFA", "Password age", and "Active key age". The table contains one row for the user "corestack-9852b", which has "None" in all columns except "Last activity" where it says "Never".

3. Enter a name for the user.
4. Enable console access.
5. Select autogenerated password.
6. And disable the option to create a new password for the user in next-sign.

IAM > Users > Create user

Step 1  
Specify user details

Step 2  
Set permissions

Step 3  
Review and create

Step 4  
Retrieve password

### Specify user details

User details

User name: UserForMySQL

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = . @ \_ - (hyphen)

Enable console access - optional

Enables a password that allows users to sign in to the AWS Management Console.

Console password:

Autogenerated password

You can view the password after you create the user.

Custom password

Enter a custom password for the user.

Show password

Users must create a new password at next sign-in (recommended).

Users automatically get the IAMUserChangePassword policy to allow them to change their own password.

7. Then click Next.

Step 2  
Set permissions

Step 3  
Review and create

Step 4  
Retrieve password

### User details

User name: UserForMySQL

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = . @ \_ - (hyphen)

Enable console access - optional

Enables a password that allows users to sign in to the AWS Management Console.

Console password:

Autogenerated password

You can view the password after you create the user.

Custom password

Enter a custom password for the user.

Show password

Users must create a new password at next sign-in (recommended).

Users automatically get the IAMUserChangePassword policy to allow them to change their own password.

For programmatic access, you can generate access keys after you create the user. [Learn more](#)

Cancel **Next**

8. Then Set Permissions page comes up. Here we need to select Attach policies directly as we are aware of the policies we need to assign to our user.

IAM > Users > Create user

Step 1  
Specify user details

Step 2  
Set permissions

Step 3  
Review and create

Step 4  
Retrieve password

### Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

#### Permissions options

Add user to group

Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions

Copy all group memberships, attached managed policies, and inline policies from an existing user.

Attach policies directly

Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (8)

[Edit](#) [Create group](#)

Now we see a page with various Permissions policies to choose from:

IAM > Users > Create user

Step 1  
Specify user details

Step 2  
**Set permissions**

Step 3  
Review and create

Step 4  
Retrieve password

## Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

### Permissions options

- Add user to group  
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions  
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly  
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

### Permissions policies (1040)

Choose one or more policies to attach to your new user.

Policy name	Type	Attached entities
<input type="checkbox"/> AccessAnalyzerServiceRolePolicy	AWS managed	0
<input type="checkbox"/> AdministratorAccess	AWS managed - job function	3

[Create policy](#)

## 9. Now we need to choose 2 policies :

1) RDSFullAccess

➤ We type RDS in search and press enter

Step 2  
**Set permissions**

Step 3  
Review and create

Step 4  
Retrieve password

### Permissions options

- Add user to group  
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions  
Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly  
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

### Permissions policies (1040)

Choose one or more policies to attach to your new user.

Policy name	Type	Attached entities
<input type="checkbox"/> AccessAnalyzerServiceRolePolicy	AWS managed	0
<input type="checkbox"/> AdministratorAccess	AWS managed - job function	3
<input type="checkbox"/> AdministratorAccess-Amplify	AWS managed	0
<input type="checkbox"/> AdministratorAccess-AWSElasticBea...	AWS managed	0

[Create policy](#)

➤ Now we have a list of all policies related to RDS.

➤ We select RDSFullAccess Policy

The screenshot shows the AWS IAM 'Permissions policies' search interface. A yellow box highlights the 'rds' filter applied in the search bar. The results table shows various AWS managed policies, with one policy, 'AmazonRDSFullAccess', selected and highlighted with a blue border.

Policy name	Type	Attached entities
AmazonRDSBetaServiceRolePolicy	AWS managed	0
AmazonRDSCustomPreviewService...	AWS managed	0
AmazonRDSCustomServiceRolePolicy	AWS managed	0
AmazonRDSDataFullAccess	AWS managed	0
AmazonRDSDirectoryServiceAccess	AWS managed	0
AmazonRDEnhancedMonitoringRole	AWS managed	0
<b>AmazonRDSFullAccess</b>	AWS managed	2
AmazonRDSPerformanceInsightsRe...	AWS managed	0

- Now we remove the RDS selected filter by clicking on the cross button .

The screenshot shows the AWS IAM 'Permissions options' interface. The search bar now contains 'rds' with a crossed-out icon, indicating the filter has been removed. The results table shows the same list of AWS managed policies as the previous screenshot, but without the 'rds' filter applied.

Policy name	Type	Attached entities
AmazonRDSBetaServiceRolePolicy	AWS managed	0
AmazonRDSCustomPreviewService...	AWS managed	0
AmazonRDSCustomServiceRolePolicy	AWS managed	0
AmazonRDSDataFullAccess	AWS managed	0
AmazonRDSDirectoryServiceAccess	AWS managed	0
AmazonRDEnhancedMonitoringRole	AWS managed	0
<b>AmazonRDSFullAccess</b>	AWS managed	2
AmazonRDSPerformanceInsightsRe...	AWS managed	0
AmazonRDSPreviewServiceRolePolicy	AWS managed	0

- We now proceed to search for EC2 full access policy by typing EC2 and press enter .

Step 3  
Review and create

---

Step 4  
Retrieve password

Add user to group  
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions  
Copy all group memberships, attached managed policies, and inline policies from an existing user.

Attach policies directly  
Attach a managed policy directly to a user. best practice, we recommend attaching po a group instead. Then, add the user to the appropriate group.

**Permissions policies (1/1040)**  
Choose one or more policies to attach to your new user.

Policy name	Type	Attached entities
<input type="checkbox"/> AccessAnalyzerServiceRolePolicy	AWS managed	0
<input type="checkbox"/> AdministratorAccess	AWS managed - job function	3
<input type="checkbox"/> AdministratorAccess-Amplify	AWS managed	0
<input type="checkbox"/> AdministratorAccess-AWSElasticBea...	AWS managed	0
<input type="checkbox"/> AlexaForBusinessDeviceSetup	AWS managed	0

**Permissions policies (1/1040)**  
Choose one or more policies to attach to your new user.

Policy name	Type	Attached entities
<input type="checkbox"/> AmazonEC2ContainerRegistryFullAc...	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerRegistryPowe...	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerRegistryRead...	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceAutosc...	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceEvents...	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceforEC2...	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceRole	AWS managed	0
<input checked="" type="checkbox"/> AmazonEC2FullAccess	AWS managed	2
<input type="checkbox"/> AmazonEC2ReadOnlyAccess	AWS managed	0
<input type="checkbox"/> AmazonEC2RoleforAWSCodeDeploy	AWS managed	0
<input type="checkbox"/> AmazonEC2RoleforAWSCodeDeplov...	AWS managed	0

➤ We need to select the required policy as shown:

## 2) EC2FullAccess

**Permissions policies (2/1040)**  
Choose one or more policies to attach to your new user.

Policy name	Type	Attached entities
<input type="checkbox"/> AmazonEC2ContainerRegistryFullAc...	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerRegistryPowe...	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerRegistryRead...	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceAutosc...	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceEvents...	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceforEC2...	AWS managed	0
<input type="checkbox"/> AmazonEC2ContainerServiceRole	AWS managed	0
<input checked="" type="checkbox"/> AmazonEC2FullAccess	AWS managed	2
<input type="checkbox"/> AmazonEC2ReadOnlyAccess	AWS managed	0
<input type="checkbox"/> AmazonEC2RoleforAWSCodeDeploy	AWS managed	0
<input type="checkbox"/> AmazonEC2RoleforAWSCodeDeplov...	AWS managed	0

- Now we have selected all of our policies, we will scroll to the bottom and click Next.

AWS managed

- AmazonEC2ReadOnlyAccess
- AmazonEC2RoleforAWSCodeDeploy
- AmazonEC2RoleforAWSCodePipelineRole
- AmazonEC2RoleforDataPipelineRole
- AmazonEC2RoleforSSM
- AmazonEC2RolePolicyForLaunchWi...
- AmazonEC2SpotFleetAutoscaleRole
- AmazonEC2SpotFleetTaggingRole
- AmazonElasticMapReduceforEC2Role
- AmazonSSManagedEC2InstanceD...
- AWSApplicationAutoscalingEC2Spo...
- AWSApplicationMigrationEC2Access

**Permissions boundary - optional**

Set a permissions boundary to control the maximum permissions for this user. Use this advanced feature used to delegate permission management to others. [Learn more](#)

Cancel Previous **Next**

- We now need to review the details we added and click on Create User.

Name	Type	Used as
AmazonRDSFullAccess	AWS managed	Permissions policy
AmazonEC2FullAccess	AWS managed	Permissions policy

**Tags - optional**

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Previous **Create user**

- Once user is created we need to download the csv file that contains the password details for the user to login.
- Then return to user list.

⌚ User created successfully  
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

IAM > Users > Create user

Step 1 Specify user details

Step 2 Set permissions

Step 3 Review and create

Step 4 Retrieve password

### Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details	
Console sign-in URL	<a href="https://nago519.signin.aws.amazon.com/console">https://nago519.signin.aws.amazon.com/console</a>
User name	<a href="#">UserForMySQL</a>
Console password	<a href="#">***** Show</a>

[Email sign-in instructions](#) [Download .csv file](#) [Return to users list](#)

⌚ User created successfully  
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

IAM > Users > Create user

Step 1 Specify user details

Step 2 Set permissions

Step 3 Review and create

Step 4 Retrieve password

### Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details	
Console sign-in URL	<a href="https://nago519.signin.aws.amazon.com/console">https://nago519.signin.aws.amazon.com/console</a>
User name	<a href="#">UserForMySQL</a>
Console password	<a href="#">***** Show</a>

[Email sign-in instructions](#) [Download .csv file](#) [Return to users list](#)

➤ We can now view the user we just created.

Identity and Access Management (IAM)

Unable to load search

Dashboard

Access management

User groups

**Users**

Roles

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Introducing the new Users list experience  
We've redesigned the Users list experience to make it easier to use. Let us know what you think.

⌚ User created successfully  
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

IAM > Users

Users (2) Info

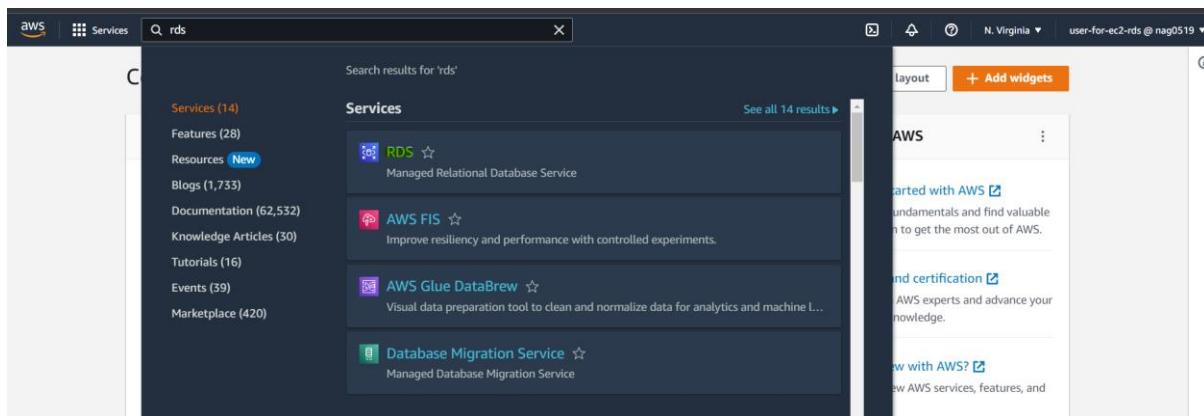
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Groups	Last activity	MFA	Password age	Active key age
corestack-9852b	None	Never	None	870 days ago	-
UserForMySQL	None	Never	None	None	-

- We will create all our services in default VPC, so our region will be N.Virginia.

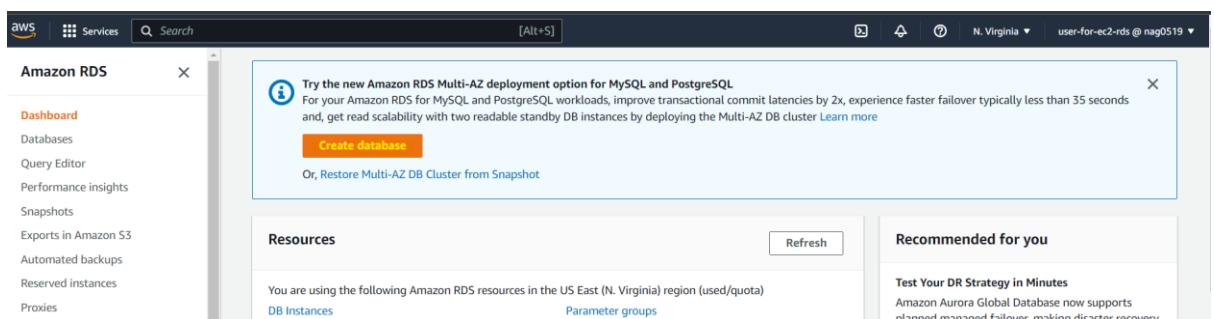
### iii. Creating MySQL Database:

#### 1. On Aws dashboard search for RDS:



Select RDS, its dashboard opens.

#### 2. Select Create Database



The create database screen appears.

S Services Search [Alt+S] N. Virginia user-for-ec2-rds @ nag0519

We listened to your feedback! Now, create a database with a single click using our pre-built configurations! Or choose your own configurations.

RDS > Create database

### Create database

Choose a database creation method [Info](#)

Standard create You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type [Info](#)

Aurora (MySQL Compatible) 

Aurora (PostgreSQL Compatible) 

MySQL 

Share your feedback X

3. Select Standard Database
4. In Engine options choose: MySQL

aws Services Search [Alt+S] N. Virginia user-for-ec2-rds @ nag0519

We listened to your feedback! Now, create a database with a single click using our pre-built configurations! Or choose your own configurations.

RDS > Create database

### Create database

Choose a database creation method [Info](#)

Standard create You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type [Info](#)

MySQL 

Aurora (MySQL Compatible) 

Aurora (PostgreSQL Compatible) 

MySQL 

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Share your feedback X

**Edition**

MySQL Community

**Known issues/limitations**  
Review the [Known issues/limitations](#) to learn about potential compatibility issues with specific database versions.

**Hide filters**

Show versions that support the Multi-AZ DB cluster [Info](#)  
Create a Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

Show versions that support the Amazon RDS Optimized Writes [Info](#)  
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

**Engine Version**

MySQL 8.0.28

**Templates**  
Choose a sample template to meet your use case.

**MySQL**

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

## 5. In Availability Choose: Free Tier

**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](#)

**Availability and durability**

**Deployment options** [Info](#)  
The deployment options below are limited to those supported by the engine you selected above.

- Multi-AZ DB Cluster - new  
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.
- Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)  
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.
- Single DB instance (not supported for Multi-AZ DB cluster snapshot)  
Creates a single DB instance with no standby DB instances.

**MySQL**

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

## 6. In DB Instance Identifier type a name.

## 7. In Credential Setting type a username of your choice/requirement.

**Settings**

**DB instance identifier** [Info](#)  
Type a name for your DB instance. The name must be unique across 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. 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.

**Manage master credentials in AWS Secrets Manager**  
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

**If you manage the master user credentials in Secrets Manager, some RDS features aren't supported.** [Learn more](#)

**MySQL**

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

## 8. Type a password and retype the same

The screenshot shows the AWS RDS MySQL instance configuration page. On the left, there are fields for 'Master password' and 'Confirm master password', both containing masked text. Below these are 'Instance configuration' settings for 'DB instance class' (set to db.t3.micro), which includes 2 vCPUs, 1 GiB RAM, and 2,085 Mbps network. A checkbox for 'Include previous generation classes' is also present. On the right, a sidebar titled 'MySQL' provides information about the database engine, including its popularity, features like automated backup, and support for up to 15 read replicas.

9.

## 10. Selecting the default configurations

The screenshot shows the AWS RDS MySQL instance configuration page. It includes 'Storage' settings for 'Storage type' (General Purpose SSD gp2) and 'Allocated storage' (set to 20 GiB). Below this is a section for 'Storage autoscaling' with a note about dynamic scaling support. The right sidebar contains the same MySQL information as the previous screenshot.

**Maximum storage threshold** **Info**  
Charges will apply when your database autoscales to the specified threshold  
1000 GiB  
The minimum value is 22 GiB and the maximum value is 6,144 GiB

**Connectivity** **Info**

Compute resource  
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource  
Set up a connection to an EC2 compute resource for this database.

**Virtual private cloud (VPC)** **Info**  
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.  
Default VPC (vpc-0d5ce8925d0fe9cc3)  
Only VPCs with a corresponding DB subnet group are listed.

**MySQL**

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

## 11. Selecting default VPN and proceeding:

**Maximum storage threshold** **Info**  
The minimum value is 22 GiB and the maximum value is 6,144 GiB

**Connectivity** **Info**

Compute resource  
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource  
Set up a connection to an EC2 compute resource for this database.

**Virtual private cloud (VPC)** **Info**  
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.  
Default VPC (vpc-0d5ce8925d0fe9cc3)  
Only VPCs with a corresponding DB subnet group are listed.

**MySQL**

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

**Maximum storage threshold** **Info**  
The minimum value is 22 GiB and the maximum value is 6,144 GiB

**Connectivity** **Info**

Compute resource  
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource  
Set up a connection to an EC2 compute resource for this database.

**Virtual private cloud (VPC)** **Info**  
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.  
Default VPC (vpc-0d5ce8925d0fe9cc3)  
Only VPCs with a corresponding DB subnet group are listed.

**MySQL**

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

**No**  
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

**VPC security group (firewall) Info**  
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

**Choose existing**  
Choose existing VPC security groups

**Create new**  
Create new VPC security group

**Existing VPC security groups**  
Choose one or more options ▾

**default** X

**Availability Zone Info**  
No preference ▾

**RDS Proxy**  
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

**Create an RDS Proxy Info**  
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy Pricing](#).

**Certificate authority - optional Info**  
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-2019 (default)  
If you don't select a certificate authority, RDS chooses one for you.

**MySQL**

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

**Database authentication**

**Database authentication options Info**

**Password authentication**  
Authenticates using database passwords.

**Password and IAM database authentication**  
Authenticates using the database password and user credentials through AWS IAM users and roles.

**Password and Kerberos authentication**  
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

**Monitoring**

**Monitoring**

**Enable Enhanced monitoring**  
Enabling Enhanced monitoring metrics are useful when you want to see how different processes or threads use the CPU.

**Error loading IAM Roles**  
User: arn:aws:iam::206336842862:user/user-for-ec2-rds is not authorized to perform: iam>ListRoles on resource: arn:aws:iam::206336842862:role/ because no identity-based policy allows the iam>ListRoles action

**MySQL**

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

## 12. Select Create Database

**Additional configuration**

Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

**Estimated monthly costs**

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier.](#)

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing](#) page.

**MySQL**

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Once we hit Create Database it takes some time to get the database Status as Available.

#### iv. Creating a web Server:

##### 1. Let's search for EC2 in aws search bar:

Search results for 'ec2'

**Services** (12)

- EC2 ★ Virtual Servers in the Cloud
- EC2 Image Builder ★ A managed service to automate build, customize and deploy OS images
- Amazon Inspector ★ Continual vulnerability management at scale
- AWS Firewall Manager ★ Central management of firewall rules

**Features** (51)

- Dashboard EC2 feature
- Limits EC2 feature

##### 2. Click on Launch Instance.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like 'EC2 Dashboard', 'Instances', 'Images', etc. The main area has a 'Resources' summary table and a 'Launch instance' section. In the 'Launch instance' section, the 'Launch Instance' button is highlighted with a yellow box. To the right, there's an 'Account attributes' panel and an 'Explore AWS' panel.

Instances (running)	1	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	1	Key pairs	1
Load balancers	0	Placement groups	1	Security groups	2
Snapshots	0	Volumes	1		

### 3. Click Launch Instance

This screenshot is identical to the one above, showing the AWS EC2 Dashboard with the 'Launch instance' button highlighted in yellow.

Instances (running)	1	Auto Scaling Groups	0	Dedicated Hosts	0
Elastic IPs	0	Instances	1	Key pairs	1
Load balancers	0	Placement groups	1	Security groups	2
Snapshots	0	Volumes	1		

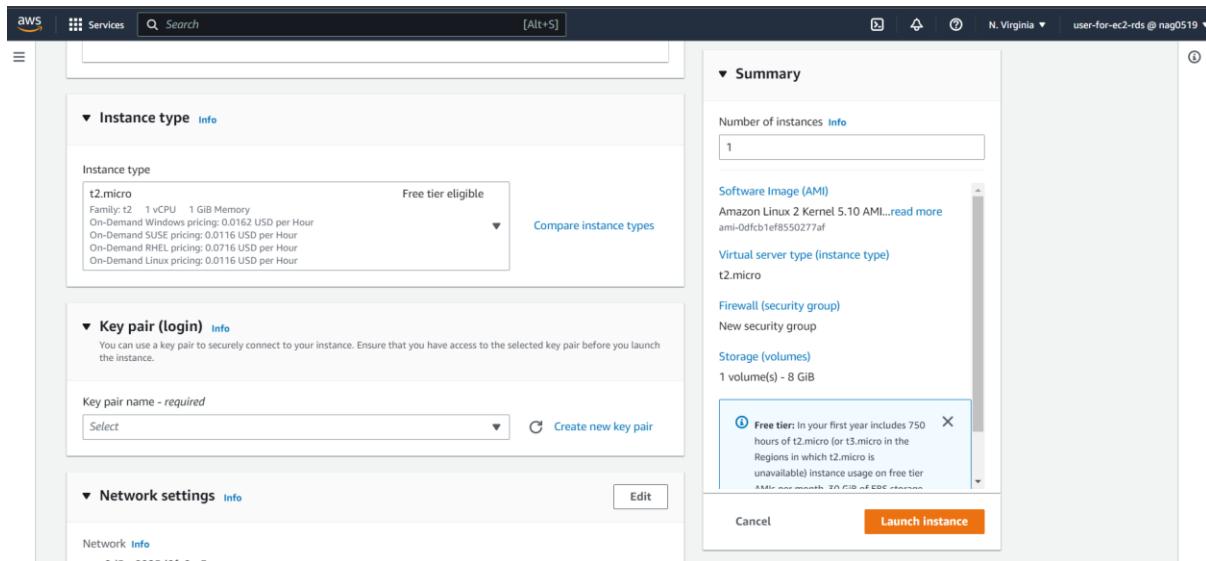
### 4. Enter the VM name:

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Summary' section, the number of instances is set to 1. The software image (AMI) is 'Amazon Linux 2 Kernel 5.10 AMI...'. The virtual server type (instance type) is 't2.micro'. A tooltip for the free tier indicates it includes 750 hours of t2.micro usage in the N. Virginia region. The storage is 1 volume(s) - 8 GiB. The 'Launch Instance' button is visible.

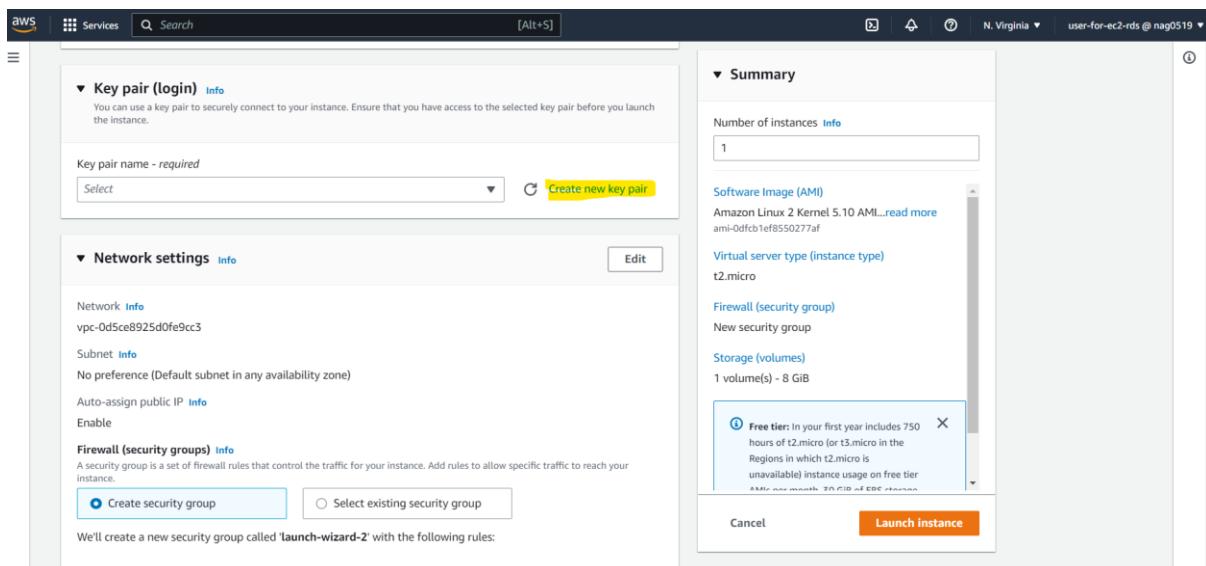
5. Select the VM Configuration as per your requirements, I will choose Linux VM with default settings.

The screenshot shows the 'AMIs' step of the EC2 wizard. It lists various operating systems: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and S. The 'Amazon Linux' option is selected. Below it, the 'Amazon Machine Image (AMI)' details are shown: 'Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type'. The AMI ID is 'ami-0dfcb1ef8550277af'. The instance type is set to 't2.micro' in the 'Instance type' dropdown. A tooltip for the free tier indicates it includes 750 hours of t2.micro usage in the N. Virginia region. The 'Launch Instance' button is visible.

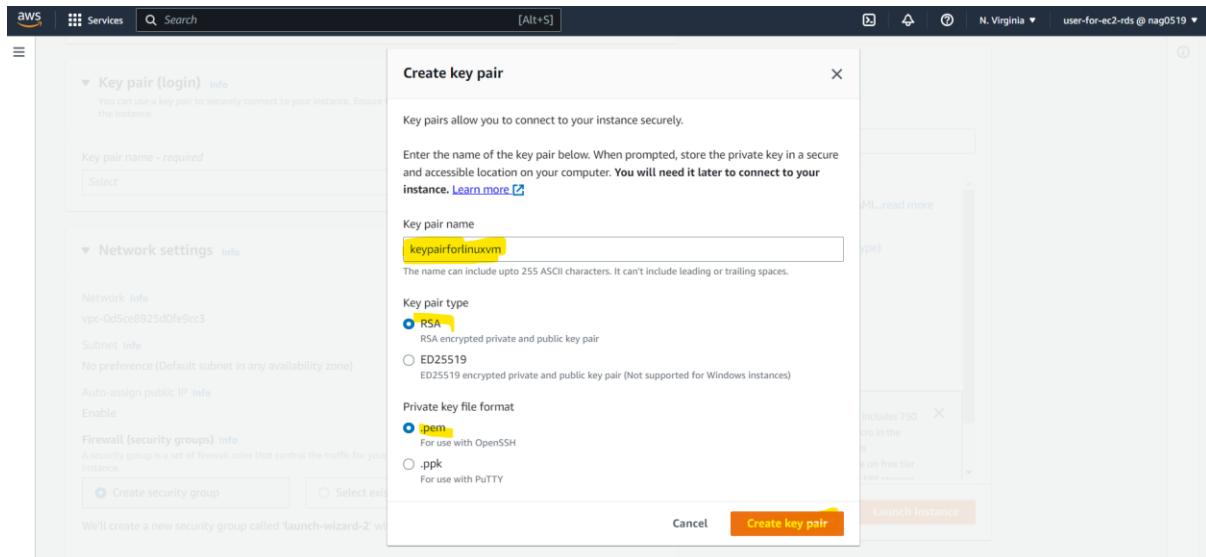
6. Choose the instance type as t2 micro:



## 7. Create Key-pair for VM:



- Type name for your key pair.
- Select .pem file as file format



- Then click Create key pair.
  - The key pair file gets downloaded in our local system.
8. Under Network setting:
- Click edit.
  - Choose Default VPN.
  - For Subnet choose the subnet region which is exactly same as the region of the MySQL, RDB we created.

DB identifier	Instance	Engine	Region & AZ	Size	Status	Actions
admin	Instance	MySQL Community	us-east-1d	db.t3.micro	Available	1 Action

- Auto assign Public IP: Enable

- In Firewall group:
- Give a security group name and note it aside for further use.

The screenshot shows the AWS EC2 Launch Wizard. On the left, under 'Firewall (security groups)', a new security group is being created with the name 'launch-wizard-2'. A single inbound rule is defined: Type 'ssh', Protocol 'TCP', Port range '22', and Source type 'Anywhere'. On the right, the 'Summary' panel shows the configuration: 1 instance, Amazon Linux 2 Kernel 5.10 AMI, instance type t2.micro, and a 'Launch Instance' button.

- In Inbound Security Group rules :
- Let SSH setting be present(default)

The screenshot shows the 'Inbound security groups rules' step. The same SSH rule is listed. A warning message is displayed: '⚠ Rules with source of 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.' The 'Advanced' link is visible at the bottom. The summary panel on the right is identical to the previous screenshot.

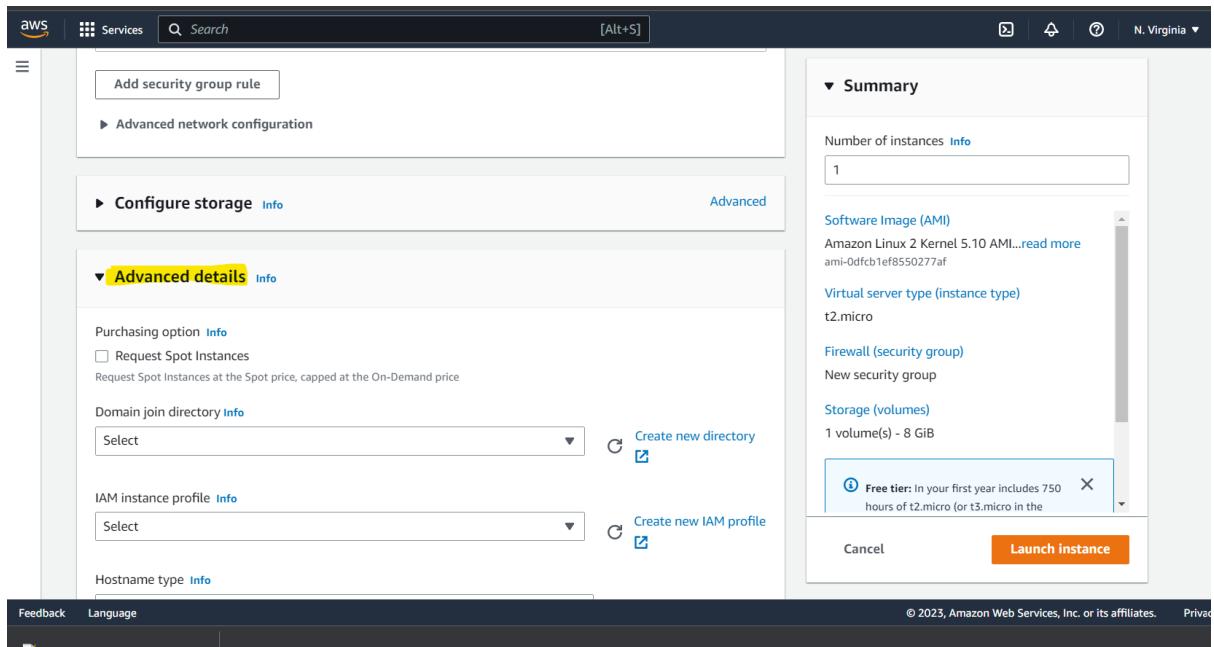
- Click on Add security group rule.
- In Type: HTTP
- Source: Internet i.e., 0.0.0.0/0

The screenshot shows the AWS EC2 instance creation wizard. On the left, under 'Security group rule 2 (TCP, 80, 0.0.0.0/0)', the 'Type' dropdown is set to 'HTTP', 'Protocol' is 'TCP', and 'Port range' is '80'. The 'Source type' is 'Custom' and the 'Source' field contains '0.0.0.0/0'. A warning message at the bottom states: '⚠ 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.' On the right, the 'Summary' section shows 'Number of instances' as 1, 'Software Image (AMI)' as 'Amazon Linux 2 Kernel 5.10 AMI...', 'Virtual server type (instance type)' as 't2.micro', and 'Storage (volumes)' as '1 volume(s) - 8 GiB'. There is also a note about the 'Free tier'.

- Then add another Security group rule:
- Type: HTTPS
- Source:0.0.0/0

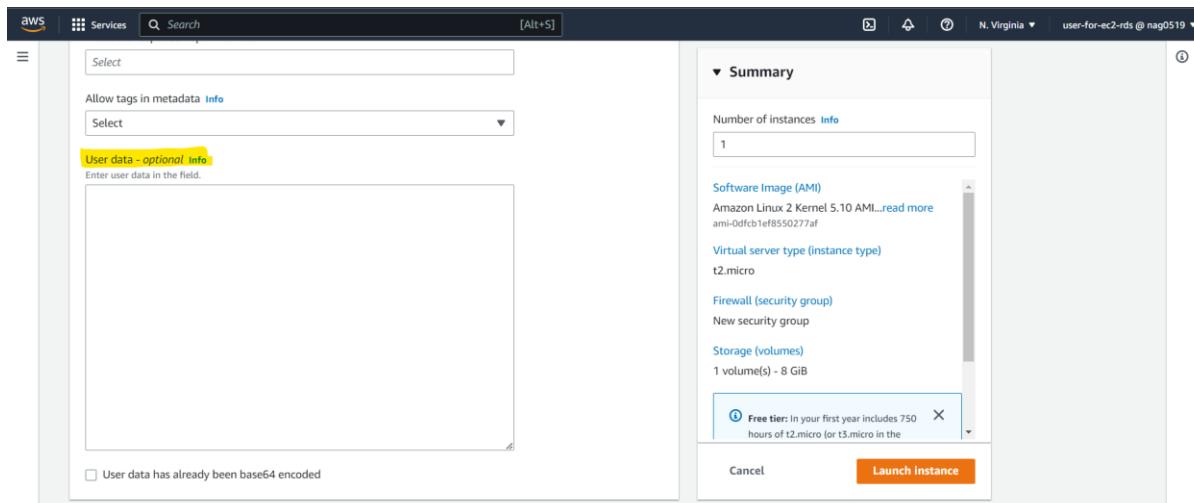
The screenshot shows the continuation of the AWS EC2 instance creation wizard. A new security group rule is being added: 'Security group rule 3 (TCP, 443, 0.0.0.0/0)'. The 'Type' dropdown is set to 'HTTPS', 'Protocol' is 'TCP', and 'Port range' is '443'. The 'Source type' is 'Custom' and the 'Source' field contains '0.0.0.0/0'. A warning message at the bottom states: '⚠ 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.' Below this, there are sections for 'Advanced network configuration', 'Configure storage', and 'Advanced details'. On the right, the 'Summary' section shows 'Number of instances' as 1, 'Software Image (AMI)' as 'Amazon Linux 2 Kernel 5.10 AMI...', 'Virtual server type (instance type)' as 't2.micro', and 'Storage (volumes)' as '1 volume(s) - 8 GiB'. A note about the 'Free tier' is also present. At the bottom right is a 'Launch Instance' button.

9. Then in Advance Details scroll to the bottom

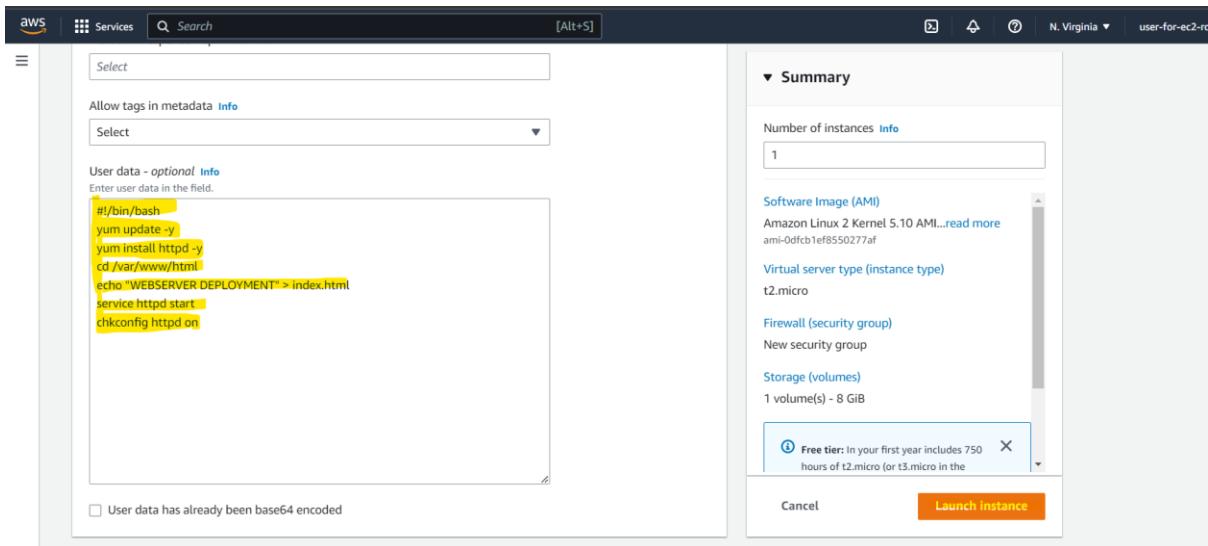


10. In User Data we need to type the following script :

```
#!/bin/bash
yum update -y
yum install httpd -y
cd /var/www/html
echo "WEBSERVER DEPLOYMENT" > index.html
service httpd start
chkconfig httpd on
```



11. Type the script and click on **Launch Instance**.



Our VM with Web Server is created successfully.



- Now we go to Instance dashboard to check the status of our VM.
- We will check our VM by Connecting it to the terminal.

Now we have created the MySQL DB and our Linux VM with Web Server.

#### v. Creating a connection between the database and the VM.

1. Navigate to the RDS dashboard and select our admin DB.

The screenshot shows the AWS RDS Databases page. A green banner at the top indicates "Successfully created database admin2". Below it, a message encourages creating a Blue/Green Deployment. The main table lists two databases: "admin" and "admin2".

DB identifier	Role	Engine	Region & AZ	Size	Status	Actions
admin	Instance	MySQL Community	us-east-1d	db.t3.micro	Available	1 Action
admin2	Instance	MySQL Community	us-east-1f	db.t3.micro	Available	1 Action

The screenshot shows the AWS RDS Database details page for "admin". The "Summary" section displays basic information like CPU usage (3.28%), status (Available), and engine (MySQL Community). The "Connectivity & security" section shows the endpoint: "admin.c1rekxklwmzp.us-east-1.rds.amazonaws.com".

- Here we need to save our endpoint for further usage.
- We need to scroll down to get the endpoint.

The screenshot shows the AWS Amazon RDS console. On the left, there's a sidebar with various navigation options like Dashboard, Databases, Query Editor, etc. The main area displays a MySQL instance named 'admin'. Key details shown include:

DB identifier	CPU	Status	Class
admin	3.28%	Available	db.t3.micro
Role	Current activity	Engine	Region & AZ
Instance	1 Connections	MySQL Community	us-east-1d

Below this, the 'Connectivity & security' tab is selected. It provides detailed information about the endpoint and port settings, networking, and security groups.

Endpoint	Networking	Security
admin.cfrekkklwmzp.us-east-1.rds.amazonaws.com	Availability Zone: us-east-1d, VPC: vpc-0d5ce8925d0fe9cc3	VPC security groups: default (sg-00e816d03650538be) - Active
Port: 3306	Subnet group: default-vpc-0d5ce8925d0fe9cc3	Publicly accessible: No
	Subnets: subnet-083f1a55fe02e976f, subnet-015a25d2fb21c1140	Certificate authority: rds-ca-2019
		Certificate authority date: August 22, 2024, 22:38 (UTC+05:30)

2. Once noted the endpoint we need to click the VPC Security Groups:
3. Security Groups dashboard opens:

The screenshot shows the AWS Security Groups dashboard. The left sidebar includes options like EC2 Dashboard, Instances, and Images. The main area displays the 'Security Groups (1/1)' table with one entry:

Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-00e816d03650538be	default	vpc-0d5ce8925d0fe9cc3	default VPC security gr...	206336842862

Below the table, the details for the 'sg-00e816d03650538be - default' security group are shown, including tabs for Details, Inbound rules, Outbound rules, and Tags. A message at the bottom says, "You can now check network connectivity with Reachability Analyzer".

4. We need to scroll down and edit Inbound Rules.

The screenshot shows the AWS EC2 Security Groups page. A search bar at the top has the query "sg-00e816d03650538be". Below it, a table lists the security group details: Name (sg-00e816d03650538be), Security group ID (sg-00e816d03650538be), VPC ID (vpc-0d5ce8925d0fe9cc3), Description (default VPC security gr...), and Owner (206336842862). The table has columns for Name, Security group ID, Security group name, VPC ID, Description, and Owner. Below the table, a section titled "sg-00e816d03650538be - default" shows tabs for Details, Inbound rules, Outbound rules, and Tags. The Inbound rules tab is selected, displaying a message about using the Reachability Analyzer and a button to "Run Reachability Analyzer".

Edit Inbound Rules page opens

The screenshot shows the "Edit inbound rules" page for the default security group. The URL is "https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#security-groups:sg-00e816d03650538be/default/inbound-rules". The page title is "Edit inbound rules". It displays a table of inbound rules with one entry: "sg-0a15368f778e23572" (Security group rule ID), "MySQL/Aurora" (Type), "TCP" (Protocol), "3306" (Port range), "Custom" (Source), and an empty "Description - optional" field. A "Delete" button is visible next to the source field. At the bottom are "Add rule", "Cancel", "Preview changes", and "Save rules" buttons.

5. We need to delete all the inbound rules .

6. Now we need to click on Add Rule

The screenshot shows the "Edit inbound rules" page again, but this time the table is empty, indicating "This security group has no inbound rules." The "Add rule" button is highlighted with a yellow box. At the bottom are "Cancel", "Preview changes", and "Save rules" buttons.

- In Type: Select MySQL/Aurora
- In Source: From the security groups present select the security group of the VM we created.

## VM's Security Group:

The screenshot shows the AWS EC2 Instances page. Two instances are listed: 'vmforrds' (Running, t2.micro) and 'vmformysqlconnection' (Running, t2.micro). The second instance is selected. Below the table, the 'Security' tab is active in the instance details panel. It shows the IAM Role (none), Owner ID (206336842862), and Launch time (Wed Feb 22 2023 02:42:37 GMT+0530 (India Standard Time)). Under 'Security groups', the group 'sg-0f2ecbd4d6cb6c344 (launch-wizard-2)' is listed. A dropdown menu is open over this group, showing other security groups: 'sg-0d23734f2c4cc75bb', 'sg-0f2ecbd4d6cb6c344', and 'default | sg-00e816d03650538be'. At the bottom right of the instance details panel are 'Cancel', 'Preview changes', and 'Save rules' buttons.

The screenshot shows the 'Edit inbound rules' page for the security group 'sg-00e816d03650538be - default'. The 'Inbound rules' table has one rule: a MySQL/Aurora rule on port 3306. The 'Source' dropdown menu is open, showing the selected group 'sg-0f2ecbd4d6cb6c344' and other available groups like 'sg-0d23734f2c4cc75bb' and 'default | sg-00e816d03650538be'. At the bottom right are 'Cancel', 'Preview changes', and 'Save rules' buttons, with 'Save rules' being the active button.

## 7. Now click on Save Rules

The screenshot shows the 'Edit inbound rules' page after saving the changes. The 'Inbound rules' table remains the same, but the 'Source' dropdown now shows the selected group 'sg-0f2ecbd4d6cb6c344'. The 'Save rules' button is highlighted. At the bottom right are 'Cancel', 'Preview changes', and 'Save rules' buttons, with 'Save rules' being the active button.

➤ Our Inbound Rule is displayed upon saving.

The screenshot shows the AWS EC2 Security Groups page. At the top, a green banner displays the message: "Inbound security group rules successfully modified on security group (sg-00e816d03650538be | default)". Below this, the "Security Groups (1/3) Info" section lists two groups: "launch-wizard-1" and "launch-wizard-2". The "Inbound rules (1/1)" section shows one rule for "sgr-0921c669b2e3568...". This rule is for "MySQL/Aurora" on "TCP" port "3306".

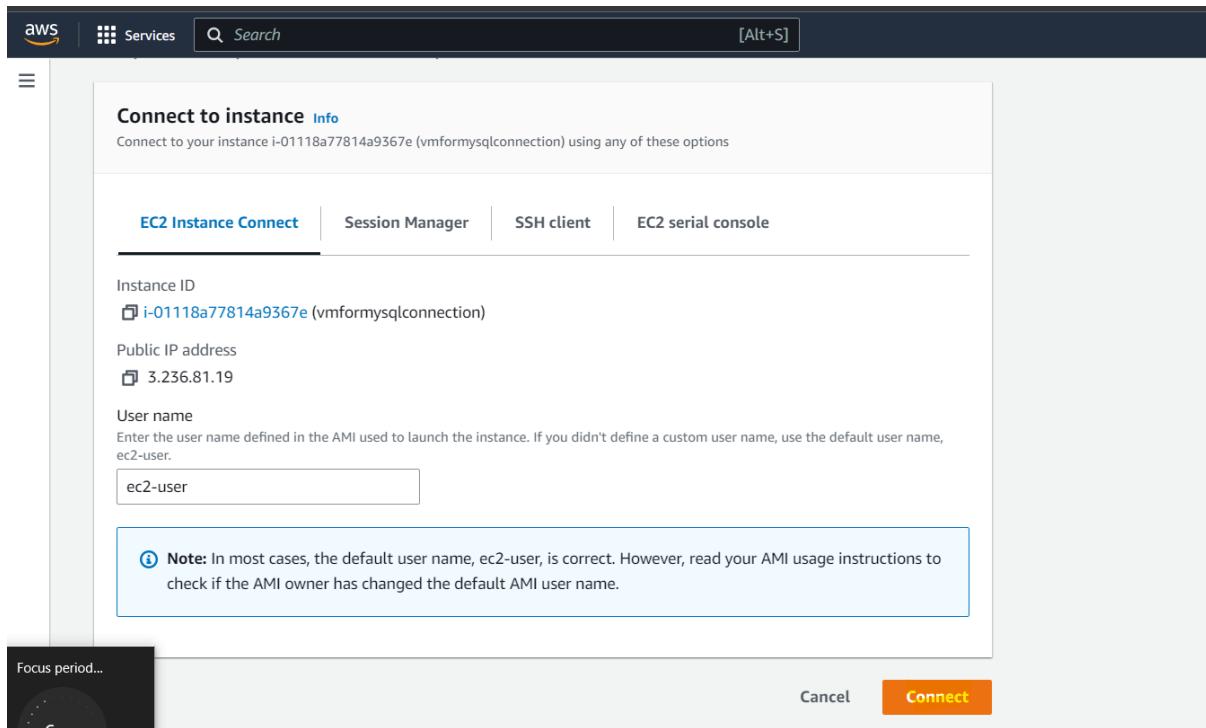
- Now we are ready to connect our VM with MYSQL and run the commands.
- Let's navigate to our VM.

#### vi. Launching our VM terminal:

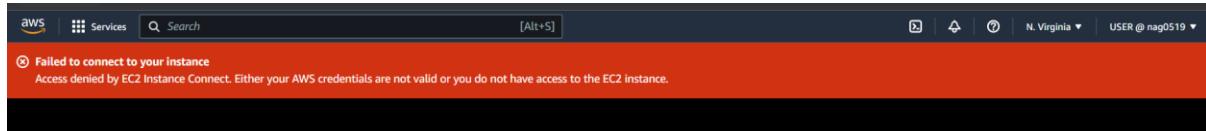
##### 1. Select our VM and click on Connect.

The screenshot shows the AWS EC2 Instances page. It displays a single instance named "vmformysqlco..." with the ID "i-01118a77814a9367e". The instance is currently "Running". The "Connect" button is highlighted in yellow.

##### 2. Again, click on connect.



- After hitting connect we were getting the error:



3. So, then we gave user one more permission:

The screenshot shows the 'Permissions policies' section of the AWS IAM console. It lists three policies: 'AmazonEC2FullAccess', 'AmazonRDSFullAccess', and 'EC2InstanceConnect'. The 'EC2InstanceConnect' policy is highlighted with a yellow background. The table columns are 'Policy name', 'Type', and 'Attached via'. The 'Permissions boundary (not set)' section is also visible at the bottom.

Policy name	Type	Attached via
AmazonEC2FullAccess	AWS managed	Directly
AmazonRDSFullAccess	AWS managed	Directly
EC2InstanceConnect	AWS managed	Directly

- Now we are able to connect to the VM

```
i-01118a77814a9367e (vmformysqlconnection)
PublicIPs: 3.236.81.19 PrivateIPs: 172.31.71.55
```

Now lets make MySQL connection .

Now we will enter the following command:

1.sudo su: This command is used to become the Super user

```
i-01118a77814a9367e (vmformysqlconnection)
PublicIPs: 3.236.81.19 PrivateIPs: 172.31.71.55
```

2.yum install mysql: to install mysql in Linux terminal.

```

[root@ip-172-31-71-55 ec2-user]# install mysql
install: missing destination file operand after 'mysql'
try 'install --help' for more information.
[root@ip-172-31-71-55 ec2-user]# yum install mysql
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
-->> Package mariadb.x86_64 1:5.5.68-1.amzn2 will be installed
-->> Finished Dependency Resolution

Dependencies Resolved

Transaction Summary
====

Install 1 Package

Total download size: 8.8 M

i-01118a77814a9367e (vmformysqlconnection)
PublicIPs: 3.236.81.19 PrivateIPs: 172.31.71.55

```

Type y:

```

Installing:
mariadb           x86_64          1:5.5.68-1.amzn2          amzn2-core          8.8 M
Transaction Summary
====

Install 1 Package

Total download size: 8.8 M
Installed size: 49 M
Is this ok [y/N]: y
Downloading packages:
mariadb-5.5.68-1.amzn2.x86_64.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:mariadb-5.5.68-1.amzn2.x86_64
  Verifying   : 1:mariadb-5.5.68-1.amzn2.x86_64
                                                               1/1
                                                               1/1

Installed:
  mariadb.x86_64 1:5.5.68-1.amzn2

Complete!
[root@ip-172-31-71-55 ec2-user]#

```

3. mysql -h endpoint\_of\_rds -P 3306 -u rds\_username -p

This command is used to connect database with the VM.

In our case the command can be changed as:

mysql -h admin.c1rekxklwmzp.us-east-1.rds.amazonaws.com -P 3306 -u admin -p

```

[reached]
[root@ip-172-31-71-55 ec2-user]# mysql -h admin.c1rekxklwmzp.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.28 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)]> []

```

i-01118a77814a9367e (vmformysqlconnection)

PublicIPs: 3.236.81.19 PrivateIPs: 172.31.71.55

- After we type the above command, we are prompted to enter our password.
- Now we type our RDS password but we don't see our pointer moving as we type our password.
- After typing the password, we have successfully connected our VM with the MySQL database.

- Now we can create a database inside the mysql db.

Show databases: this command is used to see all databases created.

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

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

We will create our own database.

Create database users.

```
MySQL [(none)]> create database users;
Query OK, 1 row affected (0.02 sec)

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

Now we will use our database.

Use Users

```
MySQL [(none)]> use users;
Database changed
MySQL [users]> show tables;
Empty set (0.00 sec)

MySQL [users]> []
```

Show tables: To display all tables inside the database.

Now we can create table:

```
MySQL [users]> create table employees( name varchar (800), empID varchar(10), city varchar(500) );
Query OK, 0 rows affected (0.02 sec)

MySQL [users]> []
```

```
MySQL [users]> show tables;
+-----+
| Tables_in_users |
+-----+
| employees       |
+-----+
1 row in set (0.01 sec)
```

```
MySQL [users]> insert into employees values ('ANAMIKA','201','Banglore');
Query OK, 1 row affected (0.01 sec)

MySQL [users]> select * from employees
-> ;
+-----+-----+-----+
| name | empID | city   |
+-----+-----+-----+
| ANAMIKA | 201 | Banglore |
+-----+-----+
1 row in set (0.00 sec)

MySQL [users]> []
```

This is how we can configure and connect MYSQL instance to a web server in AWS