

PROJECT

3 - TIER ARCHITECTURE

Name: M. Sreenath

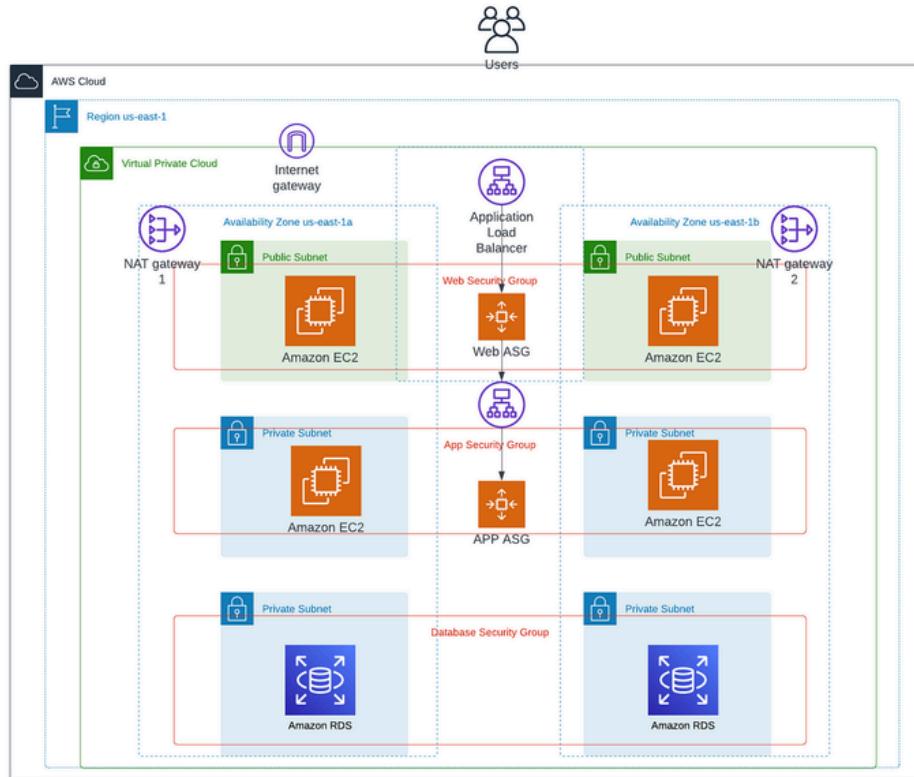


Figure: 3 - Tier Architecture

Before We Dive In...

Every grand structure rests on a foundation you rarely see — but without it, nothing stands. This project isn't just servers and subnets; it's a carefully crafted bridge between user curiosity and data precision

1. Architecture Overview

This section explains the purpose and high-level design of the 3-tier architecture.

The 3-tier architecture is designed to separate concerns and improve scalability, security, and manageability.

- **Web Tier (Presentation Layer):** Handles user interactions. Hosts the front-end or web servers that users access via a browser.
- **Application Tier (Logic Layer):** Contains business logic and processes the data between web and database layers.
- **Database Tier (Data Layer):** Stores and manages structured data securely. Only the app tier can communicate with this layer.

Steps to Create the 3-Tier Architecture:

1. Create VPC, Subnets – 6, Internet gate way – 1, Route tables – 2, Nat gate way – 1.
2. Launch an EC2 instance.
3. Create Load Balancer
4. Create an AMI (image).
5. Create Autoscaling group, Create launch template.
6. Create Subnet group.
7. Create Database (RDS).
8. Establish connection.

Step: 1 Create VPC and its components.

1. Create VPC:

The screenshot shows the AWS VPC dashboard. In the top navigation bar, the 'VPC' tab is selected. Below it, the 'Your VPCs (1)' section displays a table with one row. The row contains the name '3-tier-vpc', VPC ID 'vpc-0979ce9ab31158a9f', state 'Available', and a 'Block Public...' switch set to 'Off'. A message at the bottom says 'Select a VPC above'.

2. Subnet Setup: 2 Public, 4 Private

The screenshot shows the AWS Subnets page. The top navigation bar has the 'VPC' tab selected. The main table lists six subnets: 'public-subnet-1' and 'private-subnet-1' are public; 'public-subnet-2', 'private-subnet-2', 'private-subnet-3', and 'private-subnet-4' are private. All subnets are in an 'Available' state and are associated with the VPC 'vpc-0979ce9ab31158a9f'. The 'Block Public.' column shows all switches are off. A specific subnet, 'subnet-0e562ce6f100fd96f / private-subnet-4', is selected and shown in more detail below the table.

3. Internet Gateway Setup and attach to new VPC.

The screenshot shows the AWS Internet Gateways page. The top navigation bar has the 'VPC' tab selected. The table shows one internet gateway named '3-tier-igw' with Internet gateway ID 'igw-0da3695db8266c3d87'. It is listed as 'Attached' to the VPC 'vpc-0979ce9ab31158a9f'. The 'Owner' column shows the ID '47466'.

4. Create Route tables

The screenshot shows the AWS VPC Route Tables page. On the left, there's a sidebar with 'VPC dashboard' and 'Virtual private cloud' sections. The main area displays a table titled 'Route tables (3)'. The columns include 'Name', 'Route table ID', 'Explicit subnet associ...', 'Edge associations', 'Main', and 'VPC'. The table contains three rows:

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
public-route-table	rtb-006056561fa7df5cb	-	-	No	vpc-0979ce9ab
-	rtb-03ce18cce52a06608	-	-	Yes	vpc-0979ce9ab
private-route-table	rtb-0c83fa10eef4fc978	-	-	No	vpc-0979ce9ab

5. Associate Subnets with Route Tables

The screenshot shows the 'Edit subnet associations' page for route table 'rtb-06553919aa507c49b'. It lists two subnets under 'Available subnets (2/6)': 'public-subnet-2' and 'public-subnet-1'. Both are checked and associated with the route table.

The screenshot shows the 'Edit subnet associations' page for route table 'rtb-093dad6ac8c13026c'. It lists five subnets under 'Available subnets (4/6)': 'private-subnet-3', 'private-subnet-1', 'private-subnet-2', 'private-subnet-4', and 'public-subnet-1'. All are checked and associated with the route table.

6. Attach Public Route Table to Internet Gateway (via Edit Routes)

The screenshot shows the 'Edit routes' page for route table 'rtb-06553919aa507c49b'. It displays one route entry:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

Below this, there's a section for adding a new route with a 'Add route' button. At the bottom right are 'Cancel', 'Preview', and 'Save changes' buttons.

7. Create NAT gateway

The screenshot shows the 'Create NAT gateway' settings page. It includes fields for 'Name' (3Tire-NAT), 'Subnet' (selected: subnet-0b982de60b13cc63c (public-subnet-1)), 'Connectivity type' (Public selected), and 'Elastic IP allocation ID' (eipalloc-00b5371c9ae3d81c). A blue 'Allocate Elastic IP' button is visible.

8. Attach Private Route Table to NAT Gateway

The screenshot shows the 'Edit routes' page for route table rtb-093dad6ac8c13026c. It lists two routes: one to 10.0.0.0/16 targeting 'local' (Status: Active, Propagated: No) and another to 0.0.0.0/0 targeting 'NAT Gateway' (Status: Active, Propagated: No). A 'Remove' button is shown for the second route.

Step: 2 Launch an EC2 instance.

1. Launch EC2 Instances: 2 Public & 2 Private

The screenshot shows the EC2 Instances page with four instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
public-instance-1	i-0293694eca82b66af	Running	t2.micro	Initializing	View alarms +	us-east-1b
private-instance-2	i-02b482b6aa3df6b63	Running	t2.micro	Initializing	View alarms +	us-east-1b
public-instance-2	i-0be59e7cb1b245462	Running	t2.micro	Initializing	View alarms +	us-east-1b
private-instance-1	i-093189606fd9e36db	Running	t2.micro	Initializing	View alarms +	us-east-1b

Step 3: Create Load Balancer

1. Create Two Target Groups

- Public Target Group
- Private Target Group

The screenshot shows the AWS EC2 Target groups page. On the left, there's a navigation sidebar with links for Lifecycle Manager, Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), and Auto Scaling (Auto Scaling Groups). The main content area is titled "Target groups (2) Info". It has a search bar and a table with columns: Name, ARN, Port, Protocol, Target type, and Load balancer. Two entries are listed:

Name	ARN	Port	Protocol	Target type	Load balancer
private-tg	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/private-tg/54321	80	HTTP	Instance	None associated
public-tg	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/public-tg/54321	80	HTTP	Instance	None associated

Below the table, it says "0 target groups selected" and "Select a target group above."

2. Associate EC2 Instances with Their Respective Target Groups

This screenshot shows the "public-tg" target group details. The left sidebar is identical to the previous one. The main content area is titled "Targets" and "Registered targets (2) Info". It includes a table with columns: Instance ID, Name, Port, Zone, Health status, and Health status details. Two instances are registered:

Instance ID	Name	Port	Zone	Health status	Health status details
i-0293694eca82b66af	public-instance-1	80	us-east-1b (us...)	Unused	Target group is not co...
i-0be59e7cb1b245462	public-instance-2	80	us-east-1b (us...)	Unused	Target group is not co...

This screenshot shows the "private-tg" target group details. The left sidebar is identical to the previous ones. The main content area is titled "Targets" and "Registered targets (2) Info". It includes a table with columns: Instance ID, Name, Port, Zone, Health status, and Health status details. Two instances are registered:

Instance ID	Name	Port	Zone	Health status	Health status details
i-093189606fd9e36db	private-instance-1	80	us-east-1b (us...)	Unused	Target group is not co...
i-02b482b6aa3df6b63	private-instance-2	80	us-east-1b (us...)	Unused	Target group is not co...

3. Create Application Load Balancers: Public & Private

- Public Load Balancer – Internet-Facing
- Private Load Balancer – Internal-Facing

EC2 > Load balancers > Create Application Load Balancer

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► How Application Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme | Info
Scheme can't be changed after the load balancer is created.

Internet-facing

- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name resolves to public IPs.
- Requires a public subnet.

Internal

- Serves internal traffic.
- Has private IP addresses.
- DNS name resolves to private IPs.
- Compatible with the IPv4 and Dualstack IP address types.

EC2 VPC S3 IAM

EC2 > Load balancers > Create Application Load Balancer

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► How Application Load Balancers work

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme | Info
Scheme can't be changed after the load balancer is created.

Internet-facing

- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name resolves to public IPs.
- Requires a public subnet.

Internal

- Serves internal traffic.
- Has private IP addresses.
- DNS name resolves to private IPs.
- Compatible with the IPv4 and Dualstack IP address types.

Step 4: Create an Amazon Machine Image (AMI)

EC2 VPC S3 IAM

EC2 > Instances

Capacity Reservations		Instances (1/2) <small>Info</small>			
		Last updated less than a minute ago			
		Name	Instance ID	Instance state	Instance type
<input checked="" type="checkbox"/>	public-instance-1	i-0293694eca82b66af	Running	t2.micro	2/2 ch
<input type="checkbox"/>	public-instance-2	i-0be59e7cb1b245462	Running	t2.micro	2/2 ch

Actions ▾ **Launch instances** ▾

- Instance diagnostics
- Instance settings
- Networking
- Security
- Image and templates** ▾
- Monitor and troubleshoot

Instances (1/2) Info

Actions ▾ **Launch instances** ▾

Create image

Create template from instance

Launch more like this

Create image Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Image details

Instance ID
 i-0293694eca82b66af (public-instance-1)

Image name

Maximum 127 characters. Can't be modified after creation.

Image description - optional

Maximum 255 characters

Reboot instance
When selected, Amazon EC2 reboots the instance so that data is at rest when snapshots of the attached volumes are taken. This ensures data consistency.

Step 5: Create an Auto Scaling Group

1. Create launch template - Public

EC2 > Launch templates > Create template from instance

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Source instance
i-0293694eca82b66af

Launch template name - required
3Tier-template
Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description
allow/
Max 255 chars

Auto Scaling guidance | Info
Select this if you intend to use this template with EC2 Auto Scaling
 Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

Summary

Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-05ffe3c48a9991133

Virtual server type (instance type)
t2.micro

Firewall (security group)
allow-all

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or 17 micro instances or 13 micro instance equivalents)

Create launch template

EC2 > Launch templates > Create launch template

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Q. Search our full catalog including 1000s of application and OS images

Recents | My AMIs | Quick Start

Don't include in launch template | Recently launched | Currently in use

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Summary

Software Image (AMI)

Virtual server type (instance type)

Firewall (security group)
3tire-sg

Storage (volumes)
1 volume(s)

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or 17 micro instances or 13 micro instance equivalents)

Create launch template

2. Create Auto Scaling Group - Public

EC2 VPC S3 IAM

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1

Choose launch template

Step 2
Choose instance launch options

Step 3 - optional
Integrate with other services

Step 4 - optional
Configure group size and scaling

Step 5 - optional
Add notifications

Step 6 - optional
Add tags

Step 7
Review

Choose launch template Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name
Enter a name to identify the group.
public-autoscaling
Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1 Choose launch template
 Step 2 **Choose instance launch options**
 Step 3 - optional Integrate with other services
 Step 4 - optional Configure group size and scaling
 Step 5 - optional Add notifications
 Step 6 - optional Add tags
 Step 7 Review

Choose instance launch options Info

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Instance type requirements Info

Reset to launch template

Specify instance attributes
 Provide your compute requirements. We fulfill your desired capacity with matching instance types based on your allocation strategy selection.

Manually add instance types
 Add one or more instance types. Any of the instance types may be launched to fulfill your desired capacity based on your allocation strategy selection.

Required instance attributes
 Enter your compute requirements in virtual CPUs (vCPUs) and memory.

vCPUs
 Enter the minimum and maximum number of vCPUs per instance.
 0 minimum 100 maximum

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1 Choose launch template
 Step 2 **Choose instance launch options**
 Step 3 - optional Integrate with other services
 Step 4 - optional **Configure group size and scaling**
 Step 5 - optional Add notifications
 Step 6 - optional Add tags
 Step 7 Review

Configure group size and scaling - optional Info

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size Info
 Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type
 Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances) ▾

Desired capacity
 Specify your group size.
 2

EC2 > Auto Scaling groups > Create Auto Scaling group

Scaling Info
 You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits
 Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity 2	Max desired capacity 3
Equal or less than desired capacity	
Equal or greater than desired capacity	

Automatic scaling - optional
 Choose whether to use a target tracking policy | Info
 You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies
 Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy
 Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

3. Create launch template - Private

4. Create Autoscaling group - Private

5. Auto Scaling Launches 4 Additional Instances (2 Public, 2 Private)

EC2 > Instances

Network & Security		Instances (8) Info					
		Last updated 1 minute ago		Connect	Instance state	Actions	Launch instances
		Find Instance by attribute or tag (case-sensitive)		All states			
		Instance state = running		Clear filters			
		Name	Instance ID	Instance state	Instance type	Status check	Alarm status
		public-instance-1	i-020c492fe96d8fe2e	Running	t2.micro	2/2 checks passed	View alarm
		private-instance-1	i-08f567eabf4c334e1	Running	t2.micro	2/2 checks passed	View alarm
			i-02bb748f0553a5154	Running	t1.micro	2/2 checks passed	View alarm
			i-04173f7f7635c8c9f	Running	t2.micro	Initializing	View alarm
		public-instance-2	i-0eee20228b7f2ff9d	Running	t2.micro	2/2 checks passed	View alarm
		private-instance-2	i-0a67832e25c9d7085	Running	t2.micro	2/2 checks passed	View alarm
			i-0b47b8446512d7fd3	Running	t2.micro	Initializing	View alarm
			i-06c7efb1f3aee22b8	Running	t1.micro	2/2 checks passed	View alarm

Step 6: Create Subnet Group

[Aurora and RDS](#) > [Subnet groups](#) > Create DB subnet group

Create DB subnet group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

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.

[Aurora and RDS](#) > [Subnet groups](#) > Create DB subnet group

Add subnets

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

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

private-subnet-3 Subnet ID: subnet-04d0ce06b48370a98 CIDR: 10.0.4.0/25	private-subnet-4 Subnet ID: subnet-0ef82664eeb12703b CIDR: 10.0.5.0/26
---	---

Step 7: Create Database (RDS Instance)

Create database info

Choose a database creation method

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



PostgreSQL



MariaDB



Oracle



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

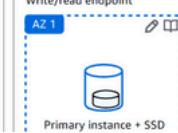
Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the [Amazon RDS service level agreement \(SLA\)](#).

Multi-AZ DB cluster deployment (3 instances)

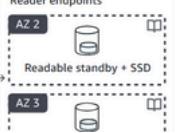
Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides:

- 99.95% uptime
- Redundancy across Availability Zones
- Increased read capacity
- Reduced write latency

Write/read endpoint



Reader endpoints

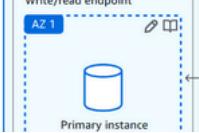


Multi-AZ DB instance deployment (2 instances)

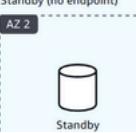
Creates a primary DB instance with a non-readable standby instance in a separate Availability Zone. This setup provides:

- 99.95% uptime
- Redundancy across Availability Zones

Write/read endpoint



Standby (no endpoint)



Single-AZ DB instance deployment (1 instance)

Creates a single DB instance without standby instances. This setup provides:

- 99.5% uptime
- No data redundancy

Write/read endpoint



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.

database-3-tier

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 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.

admin

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management

You can use AWS Secrets Manager or manage your master user credentials.

Managed in AWS Secrets Manager - most secure

RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

Self managed

Create your own password or have RDS create a password that you manage.

Auto generate password

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

Master password Info

Aura and RDS > Create database

Public access [Info](#)

Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

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

New VPC security group name
newsfor-database

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-rsa2048-g1 (default)
Expiry: May 26, 2061

Step 8: Establish Connection

EC2 VPC S3 IAM

EC2 > Instances > i-013d6c46f4e679f76 > Connect to instance

Connect [Info](#)
Connect to an instance using the browser-based client.

EC2 Instance Connect Session Manager **SSH client** EC2 serial console

Instance ID [i-013d6c46f4e679f76 \(public-server-1\)](#)

- Open an SSH client.
- Locate your private key file. The key used to launch this instance is awskey.pem
- Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "awskey.pem"
- Connect to your instance using its Public DNS:
 ec2-34-230-11-188.compute-1.amazonaws.com

Example:
 ssh -i "awskey.pem" ec2-user@ec2-34-230-11-188.compute-1.amazonaws.com

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

After connecting to the server, run:

- sudo -i
- apt update -y
- sudo apt install mysql-server -y

```
[root@ip-192-168-2-27 ec2-user]# sudo yum install mysql -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package mariadb.x86_64 1:5.5.68-1.amzn2.0.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version            Repository      Size
=====
Installing:
 mariadb          x86_64   1:5.5.68-1.amzn2.0.1    amzn2-core    8.8 M

Transaction Summary
=====
Install 1 Package

Total download size: 8.8 M
Installed size: 49 M
Downloading packages:
```

```
[root@ip-192-168-2-27 ec2-user]# mysql -h database-1.c380a08uukyc.ap-south-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 28
Server version: 8.0.35 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)]> 
```

```
[root@ip-192-168-2-27 ec2-user]# mysql -h database-1.c380a08uukyc.ap-south-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 28
Server version: 8.0.35 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)]> CREATE DATABASE webappdb;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
| webappdb       |
+-----+
5 rows in set (0.00 sec)

MySQL [(none)]> 
```

```
| information_schema |
| mysql           |
| performance_schema |
| sys             |
| webappdb        |
+-----+
5 rows in set (0.00 sec)

MySQL [(none)]> USE webappdb;
Database changed
MySQL [webappdb]> clear
MySQL [webappdb]> CREATE TABLE IF NOT EXISTS transactions(
    ->     id INT NOT NULL AUTO_INCREMENT,
    ->     amount DECIMAL(10,2),
    ->     description VARCHAR(100),
    ->     PRIMARY KEY(id)
    -> );
Query OK, 0 rows affected (0.04 sec)

MySQL [webappdb]> SHOW TABLES;
+-----+
| Tables_in_webappdb |
+-----+
| transactions      |
+-----+
1 row in set (0.02 sec)

MySQL [webappdb]> 
```