

AWS Fundamentals - Course-End Project 1

Creating a VPC with Database and EC2 Instances

Objective

To design and construct an Amazon Virtual Private Cloud (VPC) architecture that includes an EC2 instance within a public subnet and a database instance within a private subnet

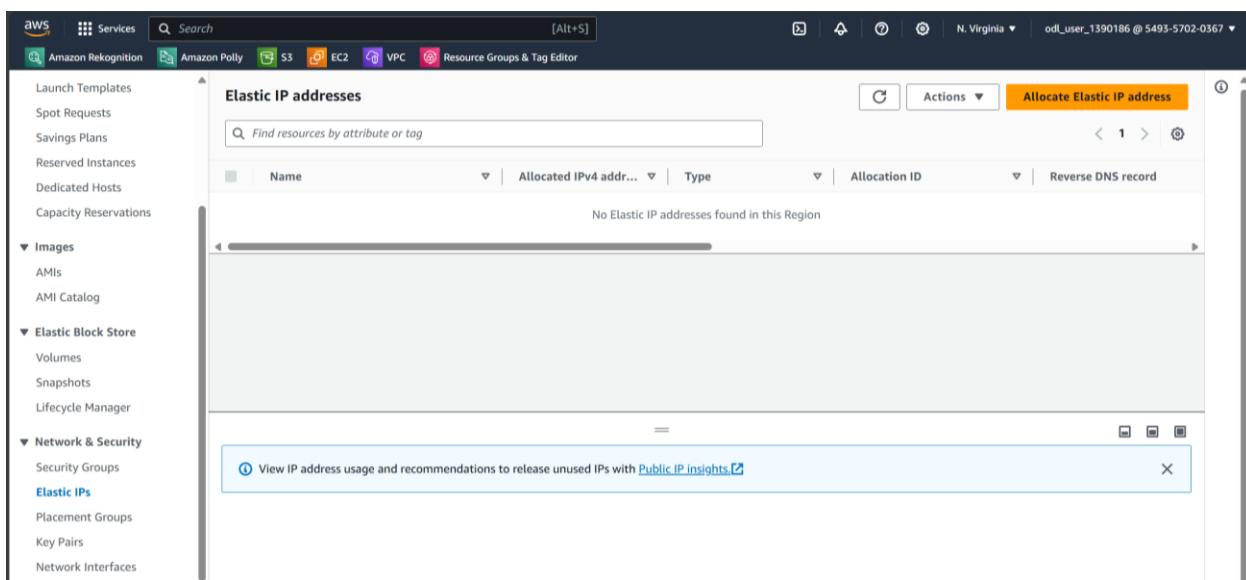
Tasks:

1. Allocating an Elastic IP address
2. Creating a VPC with public and private subnets
3. Creating an additional private subnet
4. Creating a VPC security group
5. Creating a VPC security group for a private DB instance
6. Creating a DB subnet group
7. Creating a DB instance in the private subnet
8. Creating an EC2 instance in the public subnet

Step 1: Allocating an Elastic IP address

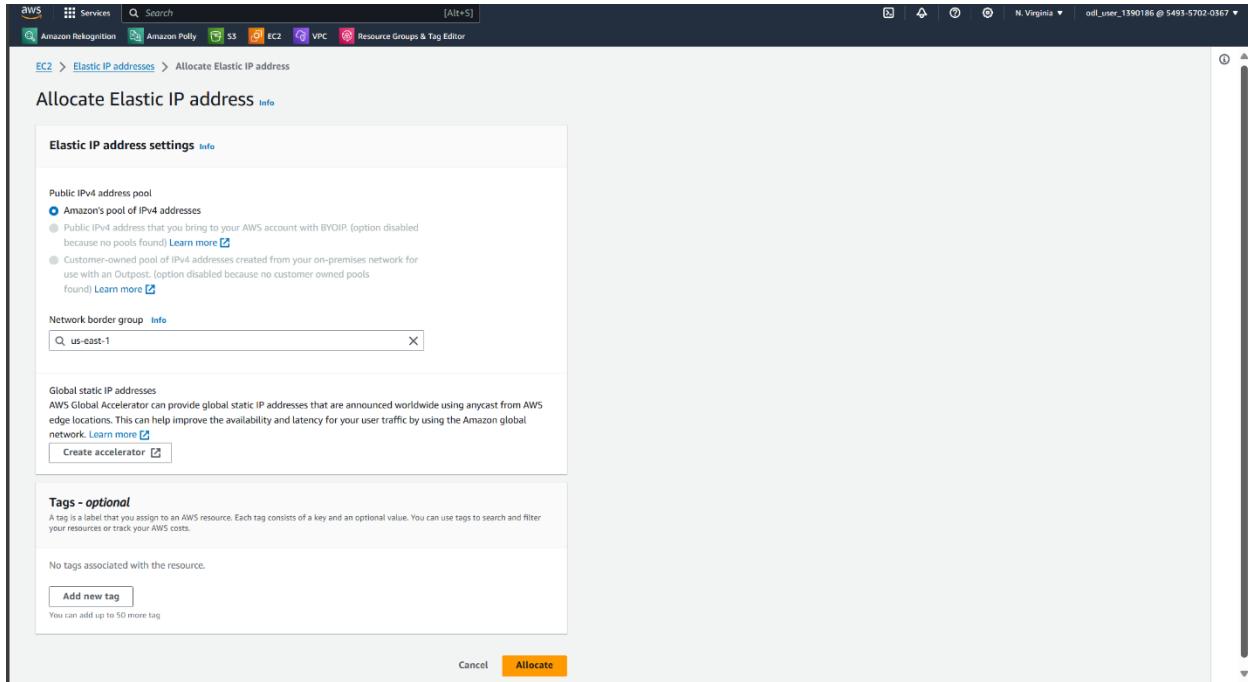
Go to EC2 Dashboard. To your left under Network & Security, Click on Elastic IP.

To create an elastic IP, click on **Allocate Elastic IP address**:

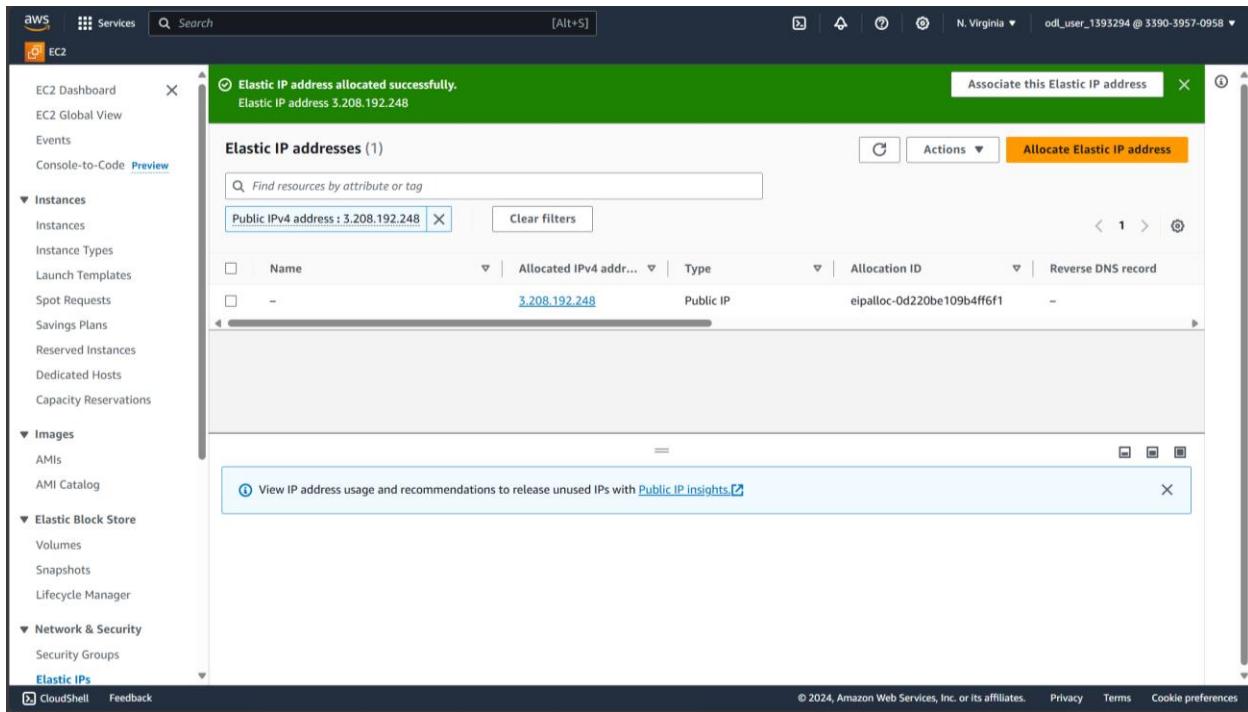


In Allocate Elastic IP address, be sure that Amazon's pool of IPv4 addresses is selected under Public IPv4 addresses.

Your network border group should match with the Availability Zone you are under.



Click on **Allocate**.

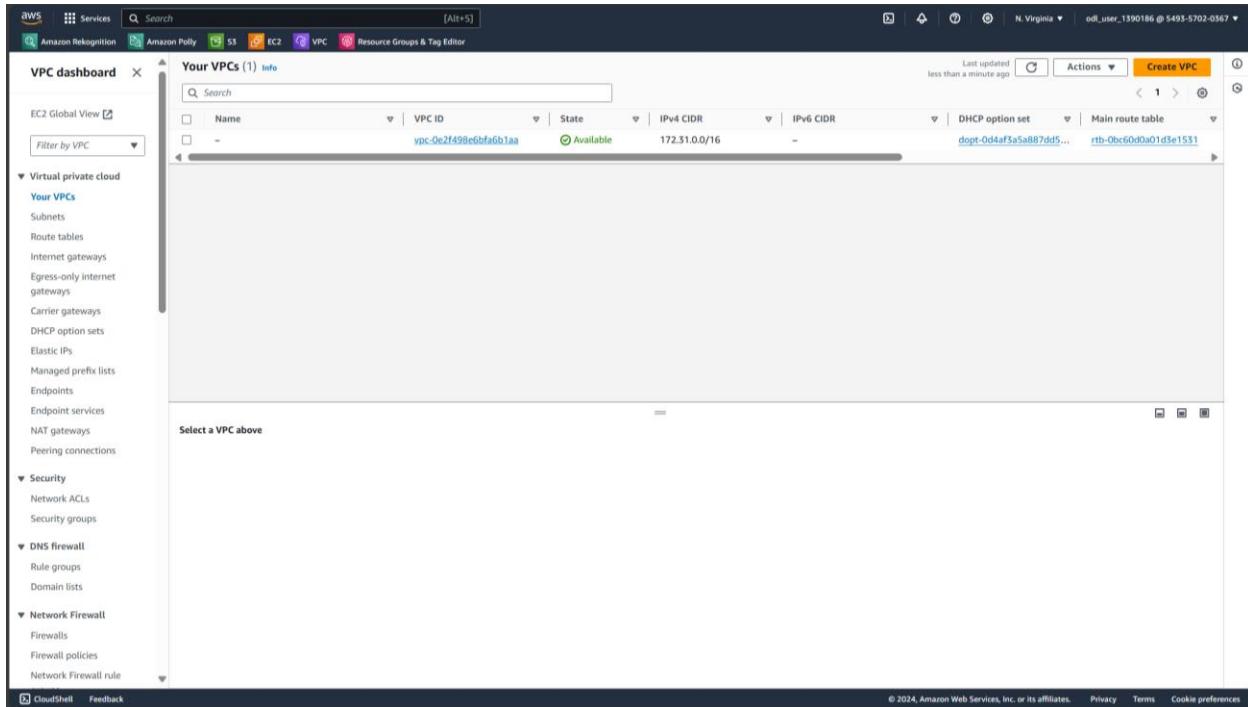


You have your Elastic IP address all set.

Step 2: Creating a VPC with public and private subnets

Most of these relate to Lesson 6 LEP.

Go to the VPC dashboard in AWS. Already you have a default VPC.



The screenshot shows the AWS VPC dashboard. On the left, there is a sidebar with various navigation options under "Virtual private cloud" and "Security". The main area displays a table titled "Your VPCs (1) info" with one row of data. The table columns are: Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, DHCP option set, and Main route table. The single entry is "vpc-0e2f498e6bfa6b1aa" with state "Available", IPv4 CIDR "172.31.0.0/16", and Main route table "rtb-0bc60d0a01d3e1531". A message at the bottom says "Select a VPC above".

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table
-	vpc-0e2f498e6bfa6b1aa	Available	172.31.0.0/16	-	dopt-0d4af1a5a887d615...	rtb-0bc60d0a01d3e1531

To create a VPC, click on **Create VPC**.

You have multiple options from VPC only that allows you to choose to create the VPC by itself and the subnets separately. Or to save time, you can choose “VPC and More” where you create a VPC and a set of subnets.

In this case, we choose our IPv4 CIDR block for our VPC to be 10.0.0.0/20

Number of Availability Zones (AZs) Info

Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

1	2	3
---	---	---

▼ Customize AZs

First availability zone

us-east-1a

Second availability zone

us-east-1b

Number of public subnets [Info](#)
The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0	2	
---	----------	--

Number of private subnets [Info](#)
The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0	2	4
---	----------	---

▼ Customize subnet CIDR blocks

Public subnet CIDR block in us-east-1a
10.0.0.0/24 256 IPs

Public subnet CIDR block in us-east-1b
10.0.1.0/24 256 IPs

Private subnet CIDR block in us-east-1a
10.0.8.0/24 256 IPs

Private subnet CIDR block in us-east-1b
10.0.9.0/24 256 IPs

NAT gateways (\$) [Info](#)
Choose the number of Availability Zones (AZs) in which to create NAT gateways.
Note that there is a charge for each NAT gateway

None	In 1 AZ	1 per AZ
-------------	---------	----------

VPC endpoints [Info](#)
Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None	S3 Gateway
------	-------------------

DNS options [Info](#)
 Enable DNS hostnames
 Enable DNS resolution

► Additional tags

[Cancel](#) **Create VPC**

Our customize block comes out as 10.0.0.0/24, 10.0.1.0/24, 10.0.8.0/24, 10.0.9.0/24. This will be useful when creating an additional subnet.

Once done, select **Create VPC**.

VPC ID: [vpc-030bc1797173e1125](#)

Status: Available

Default: Default

Default VPC: No

Network Address Usage metrics: Disabled

DNS hostnames: Enabled

Main route table: rtb-04580ad52847566

IPv6 CIDR: 10.0.0.0/20

Route 53 Resolver DNS Firewall rule groups: -

IPv6 pool: -

Owner ID: 549357020367

Resource map

- Subnets (4)**
 - us-east-1a: CourseProject1-subnet-public1-us...
 - us-east-1a: CourseProject1-subnet-private1-u...
 - us-east-1b: CourseProject1-subnet-public2-us...
 - us-east-1b: CourseProject1-subnet-private2-u...
- Route tables (4)**
 - CourseProject1-rtb-private2-us-east-1b
 - CourseProject1-rtb-private1-us-east-1a
 - CourseProject1-rtb-public
 - CourseProject1-rtb-private1-us-east-1a
- Network connections (2)**
 - CourseProject1-igw
 - CourseProject1-vpcx-s3

VPC created

Step 3. Creating Additional Subnet

Following on the Subnet page, go to Create Subnet.

Name	Subnet ID	State	VPC	IPv4 CIDR
-	subnet-0df70d34819cc6ed6	Available	vpc-0e2f498e6bfa6b1aa	172.31.16.0/20
-	subnet-01bf4a1290523fad	Available	vpc-0e2f498e6bfa6b1aa	172.31.80.0/20
-	subnet-daa29c731169196dc	Available	vpc-0e2f498e6bfa6b1aa	172.31.0.0/20
-	subnet-05bb50c12fcbae37	Available	vpc-0e2f498e6bfa6b1aa	172.31.32.0/20
-	subnet-055f4bfbc2363e097	Available	vpc-0e2f498e6bfa6b1aa	172.31.48.0/20
-	subnet-0fc2b5617995b1260	Available	vpc-0e2f498e6bfa6b1aa	172.31.64.0/20

Select a subnet

Select the VPC you want to connect to. In this case, “CourseProject1”. Give your subnet a name of your choosing.

The screenshot shows the AWS VPC console with the "Create subnet" wizard. In the "VPC" section, the VPC ID "vpc-030bc1797173e1125 (CourseProject1-vpc)" is selected. Under "Associated VPC CIDRs", the IPv4 CIDR "10.0.0.0/20" is listed. In the "Subnet settings" section, the "Subnet name" is set to "my-subnet-01". The "Availability Zone" dropdown is set to "No preference".

Choose an AZ that hasn't been taken:

As your first four are us-east-1b, us-east-1b, us-east-1a, us-east-1a

Or select No preference. In this case, we will do us-east-1c

The screenshot shows the AWS VPC console with the "Create subnet" wizard. In the "VPC" section, the VPC ID "vpc-030bc1797173e1125 (CourseProject1-vpc)" is selected. Under "Associated VPC CIDRs", the IPv4 CIDR "10.0.0.0/20" is listed. In the "Subnet settings" section, the "Subnet name" is set to "Subnet3-CP1". The "Availability Zone" dropdown is set to "US East (N. Virginia) / us-east-1c".

Choose your IPv4 VPC CIDR block. Make sure it's higher than the other subnets.

The screenshot shows the AWS VPC Subnet creation interface. The 'Subnet 1 of 1' section is active. It includes fields for 'Subnet name' (Subnet3-CP1), 'Availability Zone' (US East (N. Virginia) / us-east-1c), 'IPv4 VPC CIDR block' (10.0.0.0/20), and 'IPv4 subnet CIDR block' (10.0.10.0/26). A 'Tags - optional' section contains a single tag 'Name: Subnet3-CP1'. The 'Create subnet' button is highlighted in orange at the bottom right.

Create Subnet.

The screenshot shows the AWS VPC dashboard. The left sidebar is expanded to show 'Virtual private cloud' and 'Subnets'. The main area displays a success message: 'You have successfully created 1 subnet: subnet-016f51c46c25250a9'. Below this, the 'Subnets (1)' table lists the newly created subnet: Subnet ID: subnet-016f51c46c25250a9, State: Available, VPC: vpc-030bc1797173e1125, and IPv4 CIDR: 10.0.10.0/26.

Name	Subnet ID	State	VPC	IPv4 CIDR
Subnet3-CP1	subnet-016f51c46c25250a9	Available	vpc-030bc1797173e1125 Cou...	10.0.10.0/26

You can see it shown in your resource map for your project

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with various navigation options like EC2 Global View, Virtual private cloud, Security, DNS firewall, Network Firewall, and more. The main area displays 'Your VPCs (1/2) Info' with a table showing one VPC named 'CourseProject1-vpc'. Below this, the 'Resource map' tab is selected, showing a hierarchical diagram of the VPC components:

- VPC**: Your AWS virtual network.
- Subnets (5)**: Subnets within this VPC, including us-east-1a, us-east-1b, and us-east-1c.
- Route tables (4)**: Route tables that route network traffic to resources, including CourseProject1-rtb-private2-us-east-1b, CourseProject1-rtb-private1-us-east-1a, and CourseProject1-rtb-public.
- Network connections (2)**: Connections to other networks, including CourseProject1-lgw and CourseProject1-vpce-s3.

Step 4. Creating a VPC security group

Go to Security Groups on the left in the VPC Dashboard under Security. Select Create Security Group.

The screenshot shows the AWS Security Groups page. The left sidebar includes options for EC2 Global View, Virtual private cloud, Security, DNS firewall, Network Firewall, and CloudShell/Feedback. The main content area lists 'Security Groups (2) Info' with the following details:

Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-0f1135ed45a611637	default	vpc-030bc1797173e1125	default VPC security group	549357020367
-	sg-0e94365cf09ee7ba	default	vpc-0e2f498e6bfa6b1aa	default VPC security group	549357020367

In Create Security Group, type in the name you want to call your security group (Option to provide a description info) and select the VPC you want to attach it to.

VPC > Security Groups > Create security group

Create security group Info

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

Basic details

Security group name Info
 Name cannot be edited after creation.

Description Info

VPC Info

Inbound rules Info

This security group has no inbound rules.

[Add rule](#)

For Inbound Rules

Inbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
SSH	TCP	22	Anyw... <input type="button" value="▼"/>	<input type="text" value="0.0.0.0/0"/> <input type="button" value="X"/>
HTTP	TCP	80	Anyw... <input type="button" value="▼"/>	<input type="text" value="0.0.0.0/0"/> <input type="button" value="X"/>

[Add rule](#)

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

While it would be safer to have your SSH be your IP address. For this example, will have it be Anywhere-IPv4 0.0.0.0/0

Outbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Destination <small>Info</small>	Description - optional <small>Info</small>
All traffic	All	All	Custom <input type="button" value="▼"/> <input type="text" value="0.0.0.0/0"/> <input type="button" value="X"/>	<input type="button" value="Delete"/>

[Add rule](#)

⚠ Rules with destination of 0.0.0.0/0 or ::/0 allow all IP addresses to leave the instance. We recommend setting security group rules to leave the instance from known IP addresses only.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

[Add new tag](#)
You can add up to 50 more tags

[Cancel](#) [Create security group](#)

Be sure outbound rules have the destination set to Custom and be 0.0.0.0/0

Click on **Create security group.**

The screenshot shows the AWS VPC dashboard with a success message: "Security group (sg-0009da19d7ef4d720 | Project1-SecurityA) was created successfully". The "Details" section shows the security group name is "Project1-SecurityA", owner is "549357020367", security group ID is "sg-0009da19d7ef4d720", and it has 2 permission entries. The "Inbound rules" tab is selected, displaying two rules: one for port 80 (HTTP) and one for port 22 (SSH). The "Outbound rules" and "Tags" tabs are also present.

The screenshot shows the AWS VPC dashboard with a success message: "Security group (sg-0009da19d7ef4d720 | Project1-SecurityA) was created successfully". The "Security Groups (3) Info" section lists three security groups: "default" (VPC ID: vpc-0f1135ed45a611657), "Project1-SecurityA" (VPC ID: vpc-030bc1797173e1125), and "sg-0e94365cf0d09ee7ba" (VPC ID: vpc-0e2f498e6bfaf6b1aa). The "Create security group" button is visible at the top right. The left sidebar shows the navigation menu for the VPC dashboard.

Step 5. Creating a VPC security group for a private DB instance

Repeat the same process. Go to Create Security Group. In the page, type your custom name, description, and the VPC you want your security group to be affiliated with.

The screenshot shows the 'Create security group' wizard. In the 'Basic details' section, the security group name is 'Project1-SecurityGroupB' and the description is 'VPC security group for a private DB instance'. The VPC is set to 'vpc-030bc1797175e1125 (CourseProject1-vpc)'. Under 'Inbound rules', it says 'This security group has no inbound rules.' and there is a 'Add rule' button.

For Inbound Rules, since we want this to be for a private DB instance, we can have it set to “MySQL/Aurora” for Type and the source to your custom Security Group.

The screenshot shows the 'Create security group' wizard with the 'Inbound rules' section open. The 'Type' dropdown is set to 'MySQL/Aurora', 'Protocol' to 'TCP', 'Port range' to '3306', and 'Source' to 'Custom'. The 'Source' dropdown shows 'Project1-SecurityGroupB'. On the right, there are lists for 'CIDR blocks' (empty) and 'Security Groups' (showing 'default | sg-0f1135ed45a611637' and 'Project1-SecurityA | sg-0009da19d7ef4d720'). Below these are 'Prefix lists' (empty).

Once set, be sure Outbound Rules are the same as before

The screenshot shows the AWS Security Groups creation wizard. It has two main sections: 'Inbound rules' and 'Outbound rules'. In the 'Inbound rules' section, a rule is defined for MySQL/Aurora (TCP port 3306) from a specific security group (sg-0009da19d7ef4d720). In the 'Outbound rules' section, a rule allows all traffic (All traffic) to go to all IP addresses (0.0.0.0/0). A note at the bottom of the page advises against allowing all IP addresses and recommends setting security group rules to leave the instance from known IP addresses only. There is also a 'Tags - optional' section where no tags are currently assigned.

Click on **Create security group**.

The screenshot shows the AWS VPC dashboard. On the left, there is a navigation sidebar with various VPC-related options like Virtual private cloud, Security groups, and Network Firewall. The main area displays a success message: 'Security group (sg-0425c88c9b21a7969 | Project1-SecurityGroupB) was created successfully'. Below this, the details for the security group 'sg-0425c88c9b21a7969 - Project1-SecurityGroupB' are shown, including its name, ID, owner, and VPC ID. Under the 'Inbound rules' tab, one rule is listed: a MySQL/Aurora rule (TCP port 3306) from the security group sg-0009da19d7ef4d720.

Step 6: Creating a DB subnet group

Go to RDS. Under the RDS Dashboard, select Subnet groups.

Select '**Create DB subnet group**'.

The screenshot shows the AWS RDS Subnet groups page. On the left, there's a sidebar with links like Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups (selected), Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, and Recommendations. The main content area is titled "Subnet groups (0)" and contains a table with columns for Name, Description, Status, and VPC. A message says "No db subnet groups" and "You don't have any db subnet groups." There's a prominent orange "Create DB subnet group" button at the bottom right of the table.

In Create DB subnet group:

Type in the name for your DB subnet group and a description if you want to. Be sure to select the VPC you want to connect to “CourseProject1-vpc”.

This screenshot shows the "Create DB subnet group" wizard. Step 1: Subnet group details. It has two sections: "Subnet group details" and "Add subnets". In "Subnet group details", there's a "Name" field containing "Project1-DBSubnet1A" and a "Description" field containing "DB subnet 1A for private subnet in Project 1". In "Add subnets", there's a "Availability Zones" section with a dropdown menu set to "Choose an availability zone" and a "Subnets" section with a dropdown menu set to "Select subnets". A note at the bottom says "For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones."

To add subnets, first select the Availability Zones you want to connect to. Be sure they are the AZs that your subnets for your VPC are in.

Add subnets

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

Choose an availability zone ▾

- us-east-1a
- us-east-1b
- us-east-1c
- us-east-1d
- us-east-1e
- us-east-1f

Next select the subnets you wish to connect to. Be sure they are the subnets that match with your VPC.

Add subnets

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

Choose an availability zone ▾

us-east-1a X us-east-1b X

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

Select subnets ▾

- us-east-1a
 - subnet-0ff6ff2bff025f625 (10.0.0.0/24)
 - subnet-0d1225170fefb6b2a (10.0.8.0/24)
- us-east-1b
 - subnet-0e7c251cbcd685ab9 (10.0.9.0/24)
 - subnet-0ed11c9b811429594 (10.0.1.0/24)

Subnets selected (2)

Availability zone	Subnet ID	CIDR block
us-east-1a	subnet-0d1225170fefb6b2a	10.0.8.0/24
us-east-1b	subnet-0e7c251cbcd685ab9	10.0.9.0/24

In this case, we are creating a private subnet group for the following step. Make sure you select the correct subnets that are your private subnets from VPC.

- Us-east-1a, 10.0.8./24
- Us-east-1b, 10.0.9.0/24

Creating a multi-AZ DB clusters.

Add subnets

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

Choose an availability zone ▾

us-east-1a X us-east-1b X

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

Select subnets ▾

subnet-0d1225170fefb6b2a (10.0.8.0/24) X
subnet-0e7c251cb685ab9 (10.0.9.0/24) X

ⓘ For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Subnets selected (2)

Availability zone	Subnet ID	CIDR block
us-east-1a	subnet-0d1225170fefb6b2a	10.0.8.0/24
us-east-1b	subnet-0e7c251cb685ab9	10.0.9.0/24

Cancel Create

Click **Create**.

Successfully created Project1-DBSubnet1A. View subnet group

Subnet groups (2)

Name	Description	Status	VPC
project1-dbsubnet1	DB subnet group for Project 1	Complete	vpc-030bc1797173e1125
project1-dbsubnet1a	DB subnet 1A for private subnet in Project 1	Complete	vpc-030bc1797173e1125

Your DB subnet is created among other subnets.

Step 7. Creating a DB instance in the private subnet

Go to RBS Dashboard. Click on “Create database”

The screenshot shows the AWS RDS Dashboard. At the top, there is a green banner with the message "Successfully created Project1-DBSubnet1. View subnet group". Below the banner, there is a notification about "Introducing Aurora I/O-Optimized". The main area is titled "Resources" and lists various Amazon RDS resources. In the "Create database" section, there is a button labeled "Create database". To the right, there are sections for "Recommended services", "Recommended for you", and "Additional information".

In Create Database:

Have ‘Standard create’ selected. For Engine Options, click on for your choice of engine. For this example, we are choosing “MySQL”.

The screenshot shows the "Create database" wizard. In the "Choose a database creation method" step, the "Standard create" option is selected. In the "Engine options" step, the "MySQL" engine type is selected. Other engine options shown include Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), MariaDB, PostgreSQL, Oracle, Microsoft SQL Server, and IBM Db2.

Edition

MySQL Community

Engine version [Info](#)
View the engine versions that support the following database features.

[▼ 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 8.0.35

Enable RDS Extended Support [Info](#)
Amazon RDS Extended Support is a paid offering [\[?\]](#). By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the [RDS for MySQL documentation](#) [\[?\]](#).

Be sure to choose “MySQL Community” is selected and you have the latest version of your engine selected.

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
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.

To save cost, select Free tier under Templates.

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. 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](#)

Password strength Very strong

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / \ * @

Confirm master password [Info](#)

In Settings, type a name for DB instance identifier. Under Credentials Settings, create your username. To create and manage your own password, select “Self Managed” and unselect “Auto generate password”. Type in your own password and confirm.

The screenshot shows the 'Instance configuration' section of the AWS RDS console. In the 'DB instance class' dropdown, 'db.t3.micro' is selected, showing 2 vCPUs, 1 GiB RAM, and Network: 2,085 Mbps. Below this, under 'Storage type', 'General Purpose SSD (gp2)' is selected. The 'Allocated storage' field is set to 20 GiB. A note indicates that after modifying storage, the DB instance will be in storage-optimization.

In instance configuration, select db.t3.micro.

This screenshot provides a more detailed view of the instance configuration. It shows the 'Storage' section with 'Allocated storage' set to 20 GiB and a note about storage optimization. The 'Connectivity' section includes options for connecting to an EC2 compute resource, with 'Don't connect to an EC2 compute resource' selected.

For Storage, have Storage type be gp2 and allocated storage 20 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.

CourseProject1-vpc (vpc-030bc1797173e1125)
▼
5 Subnets, 3 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

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

project1-dbsubnet1
▼
2 Subnets, 2 Availability Zones

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.

Under Connectivity, be sure your compute resource is NOT connected to EC2 compute resource.

Connect to your VPC “CourseProject1”. Connect to your DB subnet group. In this case, since it is for a private subnet, select the one you have set for your private subnets in VPC.

Click on “No” for Public Access.

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
▼
Project1-SecurityA X Project1-SecurityGroupB 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-rsa2048-g1 (default)
▼
Expiry: May 25, 2061

If you don't select a certificate authority, RDS chooses one for you.

Additional configuration

Choose your existing VPC security group. Here, we select both “Project1-SecurityA” for SSH and HTTP access and “Project1-SecurityGroupB” that is set for a private DB instance.

The screenshot shows the AWS RDS configuration interface. The 'Tags - optional' section indicates no tags are associated with the resource, with an option to add new tags. The 'Database authentication' section shows 'Password authentication' selected, while 'Password and IAM database authentication' and 'Password and Kerberos authentication' are also listed. The 'Monitoring' section has an unchecked checkbox for 'Enable Enhanced Monitoring'.

Be sure your Database authentication is set to Password authentication.

This screenshot shows the detailed configuration for a database instance. Under 'Database options', the 'Initial database name' is set to 'db1'. The 'DB parameter group' is 'default.mysql8.0' and the 'Option group' is 'default:mysql-8-0'. In the 'Backup' section, 'Enable automated backups' is checked. A note states that automated backups are supported for InnoDB storage engine only. Under 'Backup retention period', a value of '1' day is selected. The 'Backup window' is set to 'No preference'. The 'Backup replication' section includes an option to enable replication in another AWS Region, which is unchecked. Finally, under 'Encryption', 'Enable encryption' is checked.

In Database options, create a name for your initial database name. The rest can be left alone.

Estimated Monthly costs

DB instance	12.41 USD
Storage	2.30 USD
Total	14.71 USD

This billing estimate is based on on-demand usage as described in Amazon RDS Pricing [\[?\]](#). Estimate does not include costs for backup storage, I/Os (if applicable), or data transfer.

Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#) [\[?\]](#).

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](#) [\[?\]](#)

ⓘ You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

[Cancel](#) [Create database](#)

Click on **Create Database**.

The screenshot shows the AWS RDS console. On the left, there's a sidebar with options like Dashboard, Databases (which is selected), Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, Recommendations, and Certificate update. The main area has a green banner at the top stating "Successfully created database database-1". Below it, there's a message about Aurora I/O-Optimized and a note about Blue/Green Deployments. The "Databases (1)" table lists "database-1" with details: Status: Backing-up, Role: Instance, Engine: MySQL Community, Region & AZ: us-east-1a, Size: db.t3.micro, Recommendations: none, CPU: none, Current activity: none, Maintenance: none, and VPC: vpc-030bc179. There are buttons for Group resources, Modify, Actions, Restore from S3, and Create database.

Step 8: Creating an EC2 instance in the public subnet

To create instance, go to EC2. Under the EC2 dashboard, select Instances. In Instances, select “Launch Instances” to begin process.

Screenshot of the AWS EC2 Instances page showing no instances and a "Launch instances" button.

Instances Info

No instances
You do not have any instances in this region

Select an instance

Launch an instance

Name and tags

Name: Project1-DBInstance1

Application and OS Images (Amazon Machine Image)

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.

Quick Start

- Amazon Linux
- macOS
- Ubuntu
- Windows
- Red Hat
- SUSE Linux Enterprise Server
- Browse more AMIs

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023.5.2...read more

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of bandwidth.

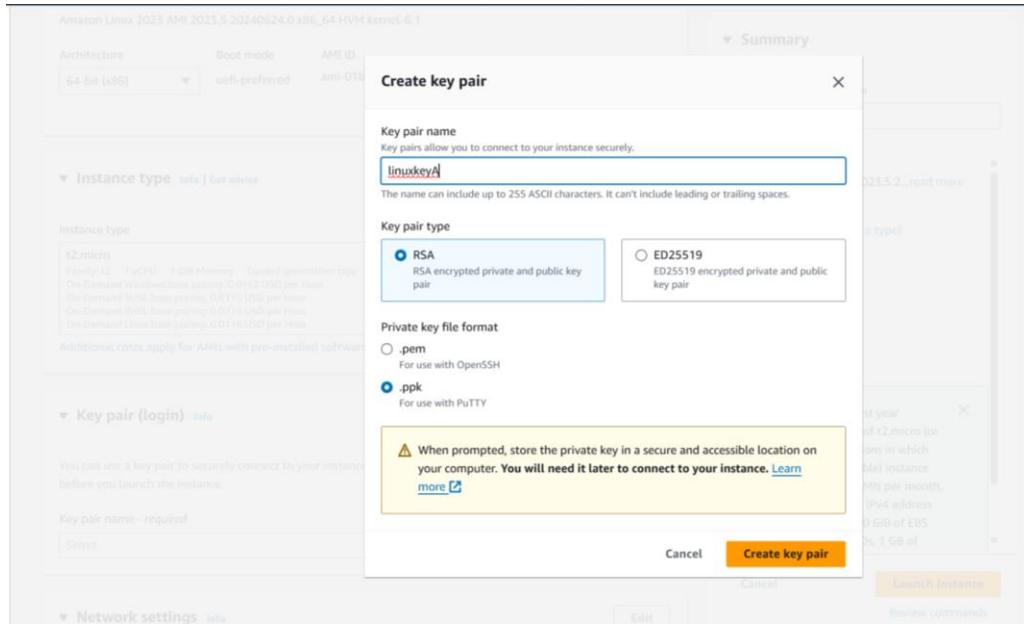
Launch instance

Create a name for your instance.

Select your AMI for your instance. For this example, we will be using Amazon Linux. Be sure it is Free tier eligible.

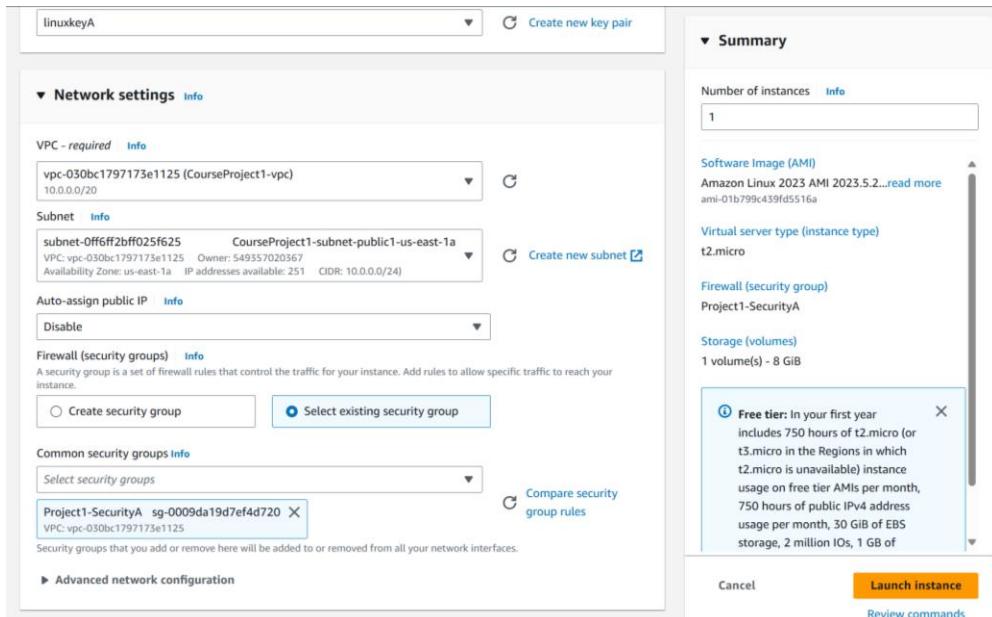
Select your instance type. In this case, you want t2.micro to be free tier.

Create a key pair for your instance.

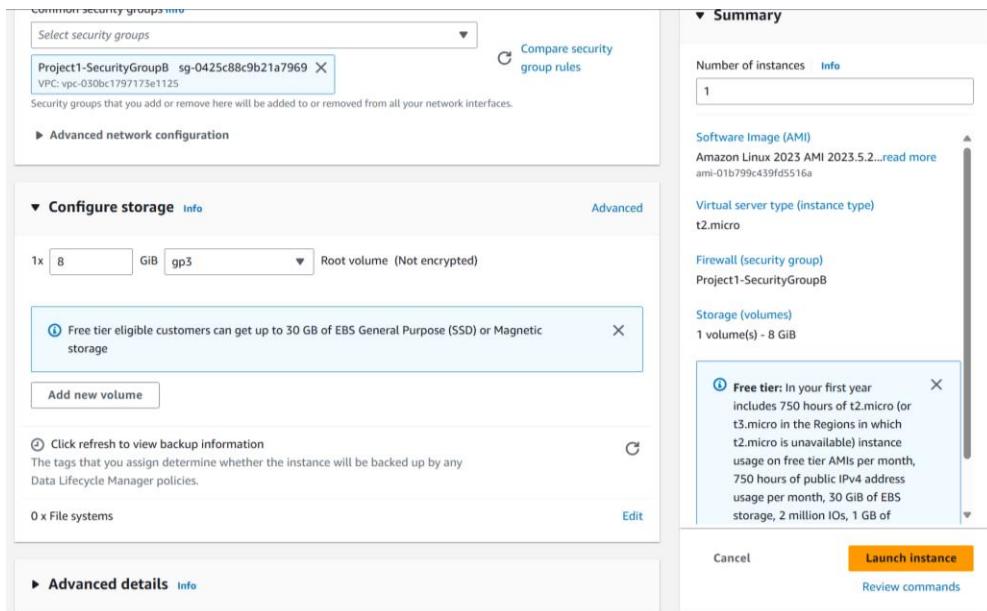


Create a name for your key pair. For the Linux AMI, we will be using PuTTY to log-in so have file format in “.ppk”. Create key pair.

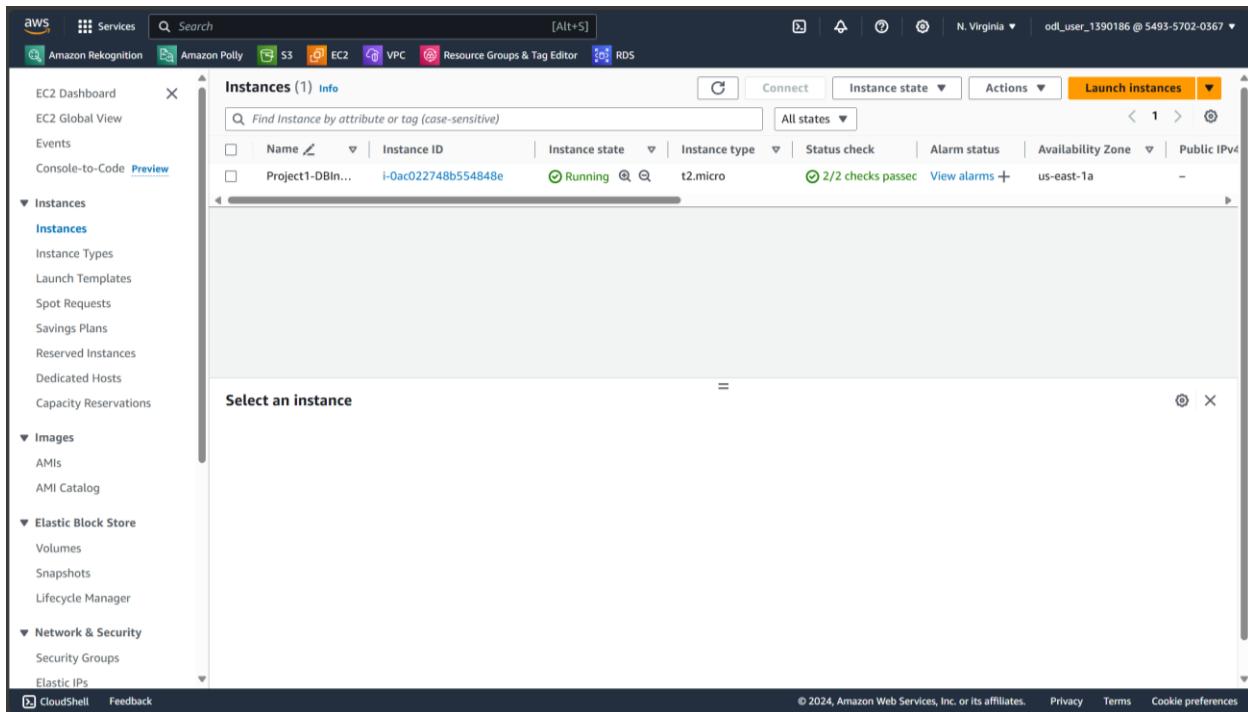
Go to Network Settings. Select Edit in order to connect to your VPC.



Select your VPC. For subnet, choose the public subnet of your VPC. For security group choose the security group you made from Step 5.



Have Configure Storage set at gp3. Review all your settings. End with “Launch instance”.



In a few minutes, the instance should have “Running” for Instance State and “2/2 checks passed” for Status check.

Returning to Step 1 to provide a public IPv4 address for instance

The screenshot shows the AWS EC2 console with the 'Elastic IP addresses' section selected. A single IP address, 3.208.192.248, is listed. The 'Actions' menu is open, and 'Associate Elastic IP address' is highlighted. Other options in the menu include 'View details', 'Release Elastic IP addresses', 'Update reverse DNS', 'Enable transfers', 'Disable transfers', and 'Accept transfers'. Below the table, there's a summary card for the IP address.

Go to Elastic IPs and select your elastic IP. Under Actions, select “Associate Elastic IP address.”

Elastic IP address: 3.208.192.248

Resource type
Choose the type of resource with which to associate the Elastic IP address.
 Instance
 Network interface

Reassociation
Specify whether the Elastic IP address can be reassigned to a different resource if it is already associated with a resource.
 Allow this Elastic IP address to be reassigned

Select Instance and select the instance you want to associate the IP address with. Once selected, click on “Associate”.

The screenshot shows the AWS EC2 Instances page. A single instance, "Project1-DBinst...", is listed as "Running" with the instance type "t2.micro". The public IP address is 3.208.192.248 and the private IP address is 10.0.2.39. The instance is in the "us-east-1a" availability zone. The AWS Compute Optimizer finding section on the right suggests opting-in for recommendations.

Now your instance has a public address from elastic IP.

Step 9: Run VPC on PuTTY

PuTTY App

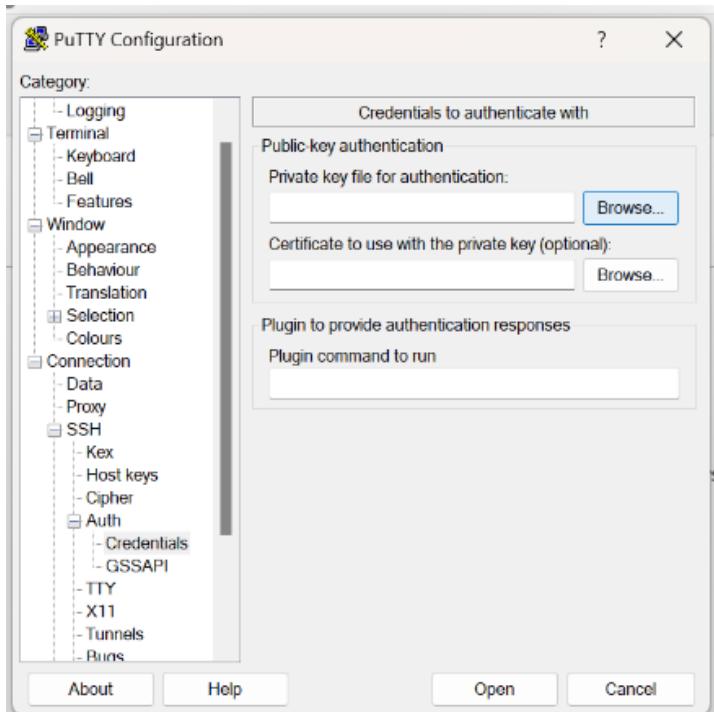
Open the PuTTY app.

The screenshot shows the AWS EC2 Instances page with the PuTTY Configuration window overlaid. The PuTTY window displays a session configuration for connecting to the instance's public IP address, 3.208.192.248, on port 22 via SSH. The "Saved Sessions" section is empty, and the "Default Settings" section is also empty. The "Connection type" is set to SSH.

Be sure your instance is Running. Copy your public IP and paste in PuTTY in Host Name

Have it at Port 22 and Select SSH.

Under SSH, under Authorization and Credentials. Browse for your private key (ppk) for Linux and install it. Then click open.

A screenshot of a terminal window titled 'ec2-user@ip-10-0-0-239:~'. The window displays the following text:

```
login as: ec2-user
Authenticating with public key "linuxkeyA"
,#
~\### Amazon Linux 2023
~~\_###\
~~ \##|
~~ \#/ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~'-->
~~ ./
~~ /m/ [ec2-user@ip-10-0-0-239 ~]$
```

The terminal prompt '[ec2-user@ip-10-0-0-239 ~]\$' is visible at the bottom.

Once open type in your user: ec2-user

Download SQL

Following these steps from AWS:

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ConnectToInstance.html#my-sql-install-cli

- sudo dnf install mariadb105 to install MySQL command-line client on Amazon Linux 2023
- be sure to type “y” for yes to install

```
[ec2-user@ip-10-0-0-239 ~]$ sudo dnf install mariadb105
Last metadata expiration check: 1:53:36 ago on Fri Jun 28 19:44:44 2024.
Dependencies resolved.
=====
Package           Architecture Version       Repository   Size
=====
Installing:
  mariadb105          x86_64    3:10.5.23-1.amzn2023.0.1
  mariadb-connector-c      x86_64    3:1.13-1.amzn2023.0.3
  mariadb-connector-c-config  noarch   3:1.13-1.amzn2023.0.3
  mariadb105-common        x86_64    3:10.5.23-1.amzn2023.0.1
  perl-Sys-Hostname        x86_64    1.23-477.amzn2023.0.6
=====
Transaction Summary
Install 5 Packages

Total download size: 1.8 M
Installed size: 1.9 M
Is transaction test? [y/n]: y
Downloaded Packages:
(1/5): mariadb-connector-c-config-3.1.13-1.amzn2023.0.3.noarch.rpm 138 kB/s | 9.2 kB  00:00
(2/5): mariadb-connector-c-3.1.13-1.amzn2023.0.3.x86_64.rpm 2.5 MB/s | 196 kB  00:00
(3/5): mariadb105-common-10.5.23-1.amzn2023.0.1.x86_64.rpm 1.2 MB/s | 30 kB  00:00
(4/5): perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64.rpm 862 kB/s | 18 kB  00:00
(5/5): mariadb105-10.5.23-1.amzn2023.0.1.x86_64.rpm 13 MB/s | 1.6 MB  00:00
=====
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing:
    mariadb-connector-c-config-3.1.13-1.amzn2023.0.3.noarch 1/1
  Installing:
    mariadb-connector-c-3.1.13-1.amzn2023.0.1.x86_64 1/5
    mariadb105-common-3:10.5.23-1.amzn2023.0.1.x86_64 2/5
    mariadb105-10.5.23-1.amzn2023.0.1.x86_64 3/5
    perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64 4/5
    mariadb105-3:10.5.23-1.amzn2023.0.1.x86_64 5/5
  Running scriptlets:
    mariadb-connector-c-3.1.13-1.amzn2023.0.3.x86_64 5/5
  Verifying:
    mariadb-connector-c-3.1.13-1.amzn2023.0.3.noarch 1/1
    mariadb-connector-c-3.1.13-1.amzn2023.0.1.x86_64 2/5
    mariadb105-3:10.5.23-1.amzn2023.0.1.x86_64 3/5
    mariadb105-common-3:10.5.23-1.amzn2023.0.1.x86_64 4/5
    perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64 5/5
=====
Installed:
  mariadb-connector-c-3.1.13-1.amzn2023.0.3.x86_64
  mariadb105-common-3:10.5.23-1.amzn2023.0.1.x86_64
=====
mariadb-connector-c-config-3.1.13-1.amzn2023.0.3.noarch
perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64
mariadb105-3:10.5.23-1.amzn2023.0.1.x86_64
=====
Complete!
[ec2-user@ip-10-0-0-239 ~]$
```

To check if mySQL is installed run: mysql –version

```
[ec2-user@ip-10-0-0-239 ~]$ mysql --version
mysql  Ver 15.1 Distrib 10.5.23-MariaDB, for Linux (x86_64) using EditLine wrapper
[ec2-user@ip-10-0-0-239 ~]$
```

You can run “man mysql” to get a manual on you MySQL

Connect DB

You can follow these steps from AWS

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ConnectToInstance.html

database-1

Summary

DB identifier	Status	Role	Engine	Recommendations
database-1	Available	Instance	MySQL Community	
CPU	Class	Current activity	Region & AZ	
0.00%	db.t3.micro	0 Connections	us-east-1a	

Connectivity & security

Endpoint & port	Networking	Security
Endpoint copied database-1.cmsay2svp7hq.us-east-1.rds.amazonaws.com Port: 3306	Availability Zone: us-east-1a VPC: CourseProject1-vpc (vpc-0d9f378d1502c9c8a) Subnet group: project1-dbsubnet Subnets: subnet-0d2e6a5e116fe1826, subnet-0f9b5f27d84ac8fd3 Network type: IPv4	VPC security groups: Project1-SecurityA (sg-0fb288547f8aecc28) (Active), Project1-SecurityGroupB (sg-011043bfe6fad7c35) (Active) Publicly accessible: No Certificate authority: rds-ca-rsa2048-g1 Certificate authority date: May 25, 2061, 16:34 (UTC-07:00) DB instance certificate expiration date: June 28, 2025, 12:10 (UTC-07:00)

Copy the endpoint address of your database. Then in the terminal type in:

```
mysql -h endpoint_address -P port_number -u master_username_for_DB -p
```

```
[ec2-user@ip-10-0-0-239:~]
Installing : mariadb-connector-c-3.1.13-1.amzn2023.0.3.x86_64
Installing : mariadb105-common-3:10.5.23-1.amzn2023.0.1.x86_64
Installing : perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64
Installing : mariadb105-3:10.5.23-1.amzn2023.0.1.x86_64
Running scriptlet: mariadb105-3:10.5.23-1.amzn2023.0.1.x86_64
Verifying  : mariadb-connector-c-3.1.13-1.amzn2023.0.3.x86_64
Verifying  : mariadb-connector-c-config-3.1.13-1.amzn2023.0.3.noarch
Verifying  : mariadb105-3:10.5.23-1.amzn2023.0.1.x86_64
Verifying  : mariadb105-common-3:10.5.23-1.amzn2023.0.1.x86_64
Verifying  : perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64

Installed:
mariadb-connector-c-3.1.13-1.amzn2023.0.3.x86_64           mariadb-connector-c-config-3.1.13-1.amzn2023.0.3.noarch
mariadb105-common-3:10.5.23-1.amzn2023.0.1.x86_64        perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64

Complete!
[ec2-user@ip-10-0-0-239 ~]$ mysql --version
mysql Ver 15.1 Distrib 10.5.23-MariaDB, for Linux (x86_64) using EditLine wrapper
[ec2-user@ip-10-0-0-239 ~]$ man mysql
[ec2-user@ip-10-0-0-239 ~]$ mysql --version
mysql Ver 15.1 Distrib 10.5.23-MariaDB, for Linux (x86_64) using EditLine wrapper
[ec2-user@ip-10-0-0-239 ~]$ mysql -h database-1.cmsay2svp7hq.us-east-1.rds.amazonaws.com -P 3306 -u admin1 -p
Enter password: [REDACTED]
```

Type in your password for administrator of your DB.

```
[ec2-user@ip-10-0-0-239:~]
Verifying  : perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64

Installed:
mariadb-connector-c-3.1.13-1.amzn2023.0.3.x86_64           mariadb-connector-c-config-3.1.13-1.amzn2023.0.3.noarch
mariadb105-common-3:10.5.23-1.amzn2023.0.1.x86_64        perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64
mariadb105-3:10.5.23-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-10-0-0-239 ~]$ mysql --version
mysql Ver 15.1 Distrib 10.5.23-MariaDB, for Linux (x86_64) using EditLine wrapper
[ec2-user@ip-10-0-0-239 ~]$ man mysql
[ec2-user@ip-10-0-0-239 ~]$ mysql --version
mysql Ver 15.1 Distrib 10.5.23-MariaDB, for Linux (x86_64) using EditLine wrapper
[ec2-user@ip-10-0-0-239 ~]$ mysql -h database-1.cmsay2svp7hq.us-east-1.rds.amazonaws.com -P 3306 -u admin1 -p
Enter password: [REDACTED]

Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 56
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)]>
```

You are connected to your MySQL database through terminal.

```
MySQL [(none)]> \h
General information about MariaDB can be found at
http://mariadb.org

List of all client commands:
Note that all text commands must be first on line and end with ';'
?
  ((?) Synonym for help.
clear
  ((c) Clear the current input statement.
connect
  ((n) Connect to the server. Optional arguments are db and host.
delimiter
  ((d) Set statement delimiter.
edit
  ((e) Edit command with $EDITOR.
ego
  ((G) Send command to MariaDB server, display result vertically.
exit
  ((q) Exit mysql. Same as quit.
go
  ((g) Send command to MariaDB server.
help
  ((h) Display this help.
noautocommit
  ((N) Don't write into outfile.
pager
  ((P) Set PAGER [to pager]. Print the query results via PAGER.
print
  ((p) Print current command.
prompt
  ((R) Change your mysql prompt.
quit
  ((q) Quit mysql.
rehash
  ((#) Rebuild completion hash.
source
  ((.) Execute an SQL script file. Takes a file name as an argument.
status
  ((s) Show status information from the server.
system
  ((!) Execute a system shell command.
tee
  ((T) Set outfile [to outfile]. Append everything into given outfile.
use
  ((u) Use another database. Takes database name as argument.
charset
  ((C) Switch to another charset. Might be needed for processing binlog with multi-byte charsets.
warnings
  ((W) Show warnings after every statement.
nowarning
  ((w) Don't show warnings after every statement.

For server side help, type 'help contents'

MySQL [(none)]> \q
Bye
[ec2-user@ip-10-0-0-239 ~]$
```

Can run \h as an option.

For HTTP Access

Run Http/Apache Web services to check if you can access it on browser.

[Directions: Compiling and Installing - Apache HTTP Server Version 2.4](#)

Run: sudo yum install httpd

```
[ec2-user@ip-10-0-0-239 ~]$ sudo yum http
No such command: http. Please use 'ls' to find an yum command
It is hard to be a YUM plugin command, try "yum install dnf-command(http)"
[ec2-user@ip-10-0-0-239 ~]$ sudo yum install http
Last metadata expiration check: 2:34:46 ago on Fri Jun 28 19:44:44 2024.
No match for argument: http
Error: Unable to find a match: http
[ec2-user@ip-10-0-0-239 ~]$ sudo yum install httpd
Last metadata expiration check: 2:38:03 ago on Fri Jun 28 19:44:44 2024.
Dependencies resolved.

=====
| Package           | Architecture | Version      | Repository | Size |
|=====|
| Installing:      |             |             |            |       |
| httpd            | x86_64       | 2.4.59-2.amzn2023 | amazonlinux | 47 k |
|=====|
| Installing dependencies: |             |             |            |       |
| apr               | x86_64       | 1.7.2-2.amzn2023.0.2 | amazonlinux | 129 k |
| apr-util          | x86_64       | 1.6.3-1.amzn2023.0.1 | amazonlinux | 98 k |
| generic-logos-httpd | noarch     | 18.0.0-12.amzn2023.0.3 | amazonlinux | 19 k |
| httpd-core        | x86_64       | 2.4.59-2.amzn2023 | amazonlinux | 1.4 M |
| httpd-filesystem | noarch     | 2.4.59-2.amzn2023 | amazonlinux | 14 k |
| httpd-tools       | x86_64       | 2.4.59-2.amzn2023 | amazonlinux | 81 k |
| libbrotli         | x86_64       | 1.0.9-4.amzn2023.0.2 | amazonlinux | 315 k |
| mailcap           | noarch     | 2.1.49-3.amzn2023.0.3 | amazonlinux | 33 k |
| installing weak dependencies: |             |             |            |       |
| axkit-util-spnssl | x86_64       | 1.6.3-1.amzn2023.0.1 | amazonlinux | 17 k |
| mod_http2         | x86_64       | 2.0.27-1.amzn2023.0.2 | amazonlinux | 166 k |
| mod_lua            | x86_64       | 2.4.59-2.amzn2023 | amazonlinux | 61 k |
|=====|
| Transaction Summary |             |             |            |       |
| install 12 Packages |             |             |            |       |
| Total download size: 2.3 M |             |             |            |       |
| Installed size: 6.9 M |             |             |            |       |
| is this ok [y/N]: |             |             |            |       |
[ec2-user@ip-10-0-0-239 ~]$
```

After installed, check your status of your httpd/Apache Web server.

Run: status httpd service

```
Complete!
[ec2-user@ip-10-0-0-239 ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[ec2-user@ip-10-0-0-239 ~]$
```

To enable program, run:

- sudo systemctl enable httpd
- sudo systemctl start httpd

HTTPD is enabled and allow access to database by copying your IPv4 address of instance and paste it on your browser.

The following steps are how to create a VPC and connect to an EC2 Instance and Database.

Finish