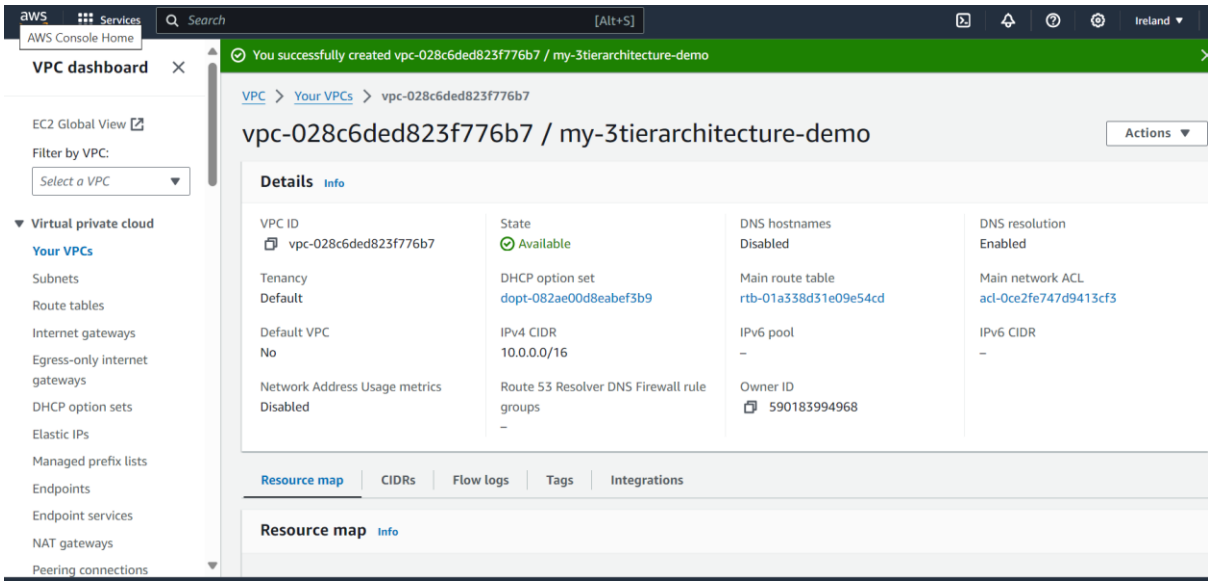
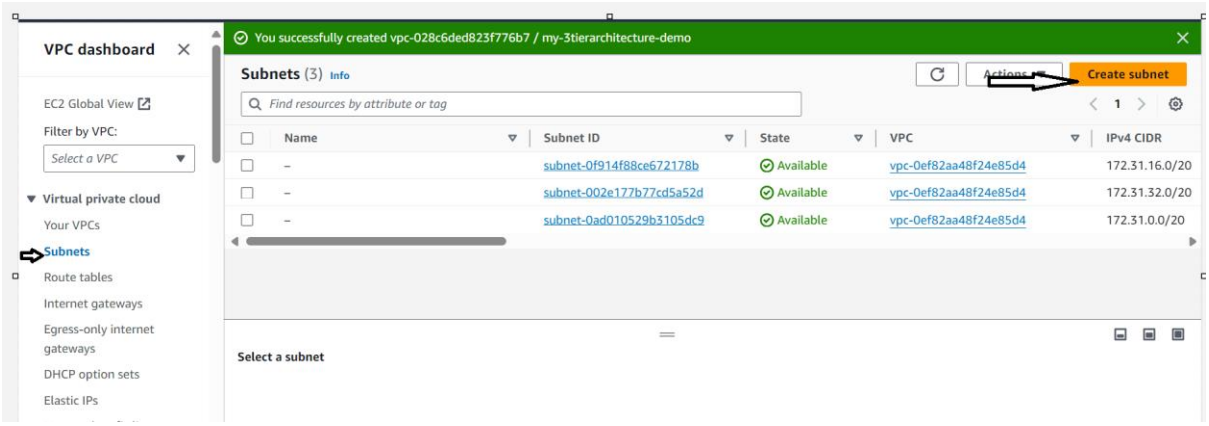
VPC and Subnets

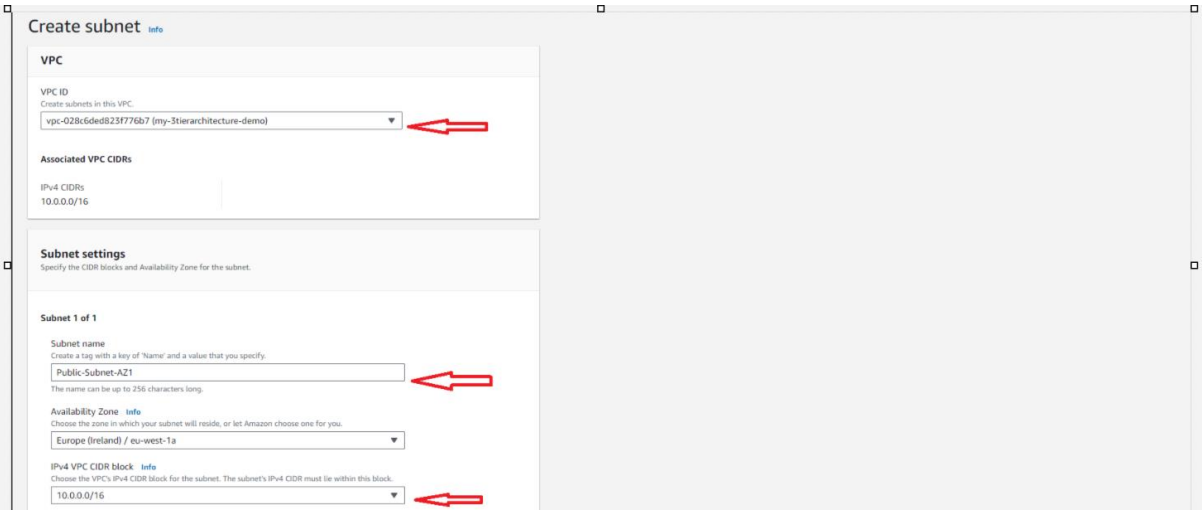
VPC Creation

- Navigate to the VPC dashboard in the AWS console and navigate to **Your VPCs** on the left-hand side.

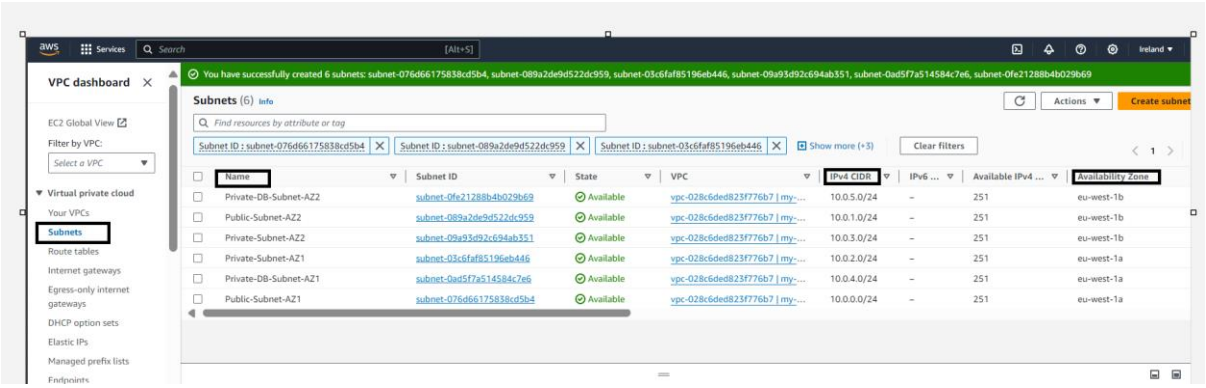




We will need six subnets across two availability zones. That means that three subnets will be in one availability zone, and three subnets will be in another zone. Each subnet in one availability zone will correspond to one layer of our three-tier architecture.



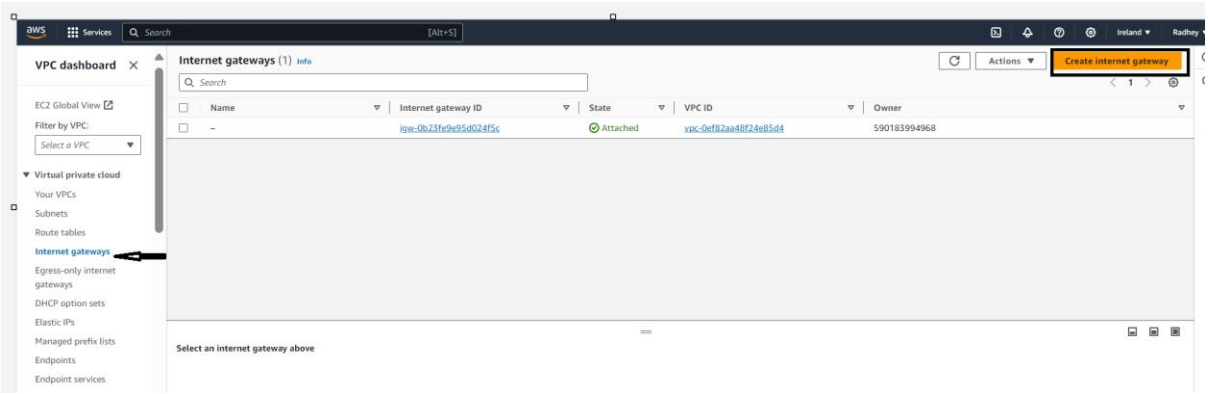
Your final subnet setup should be similar to this. Verify that you have 3 subnets across 2 different availability zones.



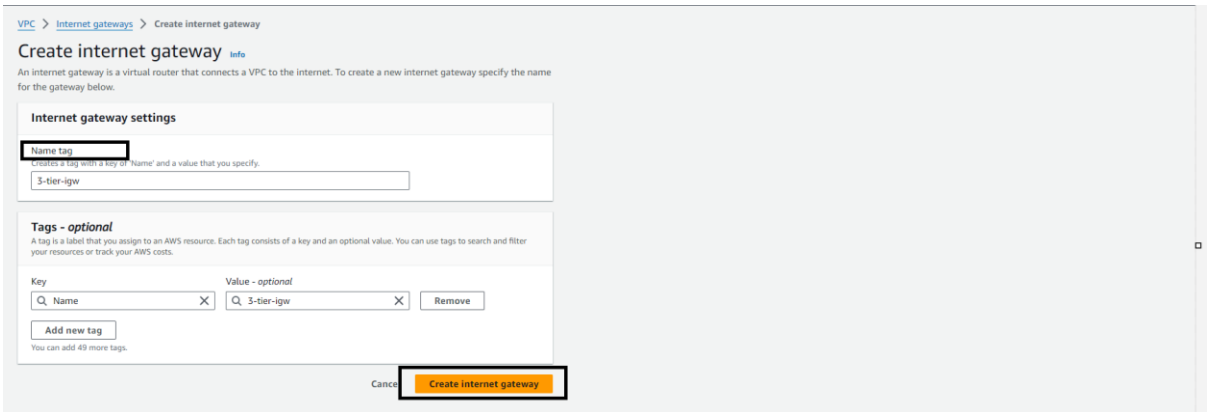
Internet Connectivity

Internet Gateway:

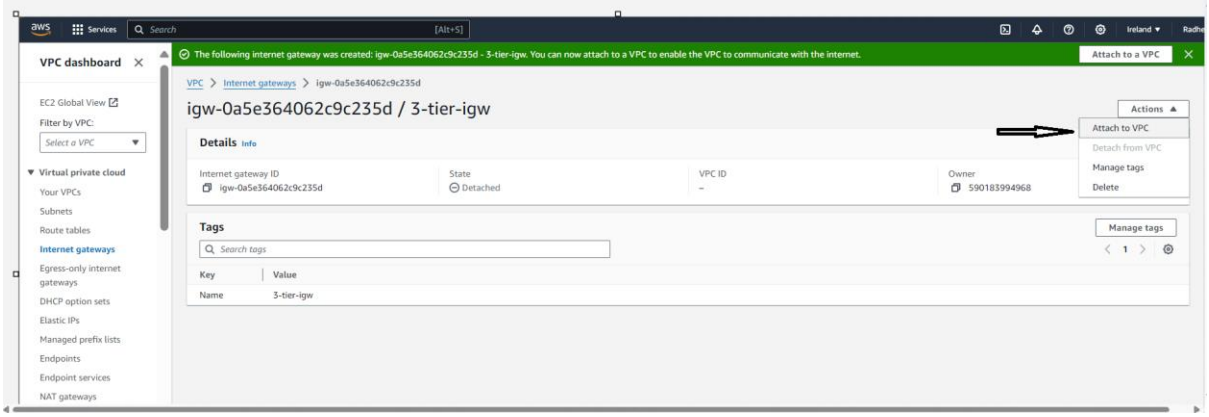
- In order to give the public subnets in our VPC internet access we will have to create and attach an Internet Gateway. On the left-hand side of the VPC dashboard, select Internet Gateway.



- Create your internet gateway by simply giving it a name and clicking Create internet gateway.

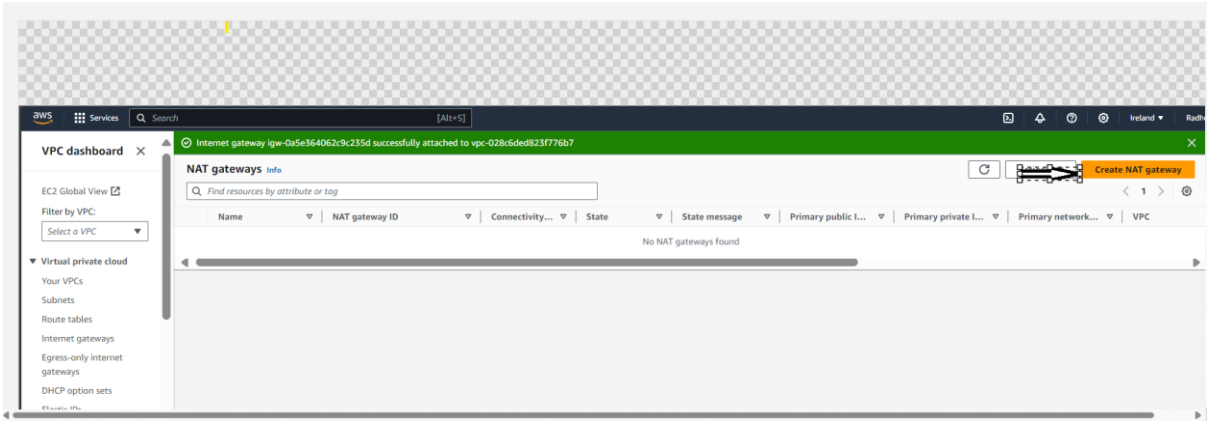


- After creating the internet gateway, attach it to your VPC that you create in the VPC and Subnet Creation step of the workshop. You have a couple options on how to do this, either with the creation success message or the Actions drop down.

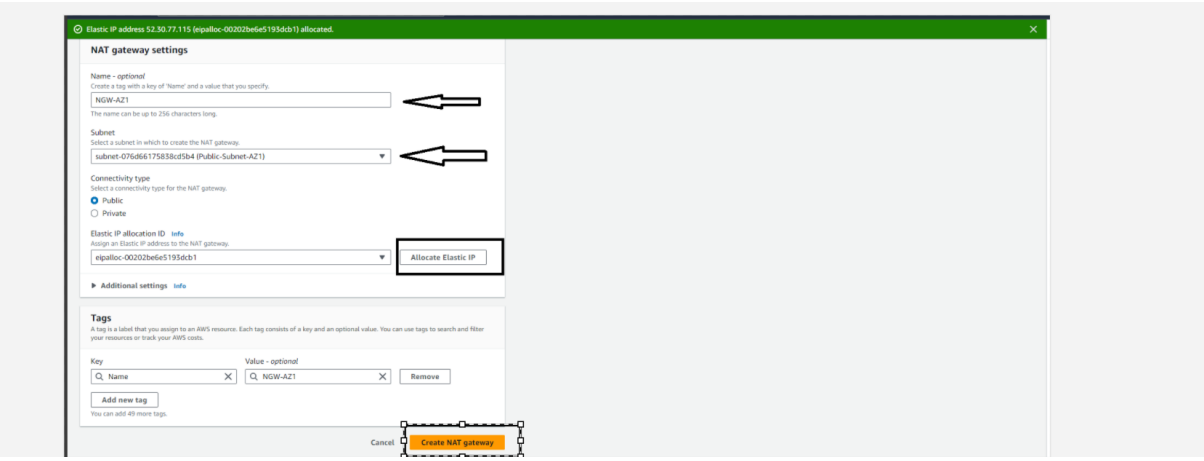


NAT Gateway

- In order for our instances in the app layer private subnet to be able to access the internet they will need to go through a NAT Gateway. For high availability, you'll deploy one NAT gateway in each of your public subnets. Navigate to NAT Gateways on the left side of the current dashboard and click Create NAT Gateway.



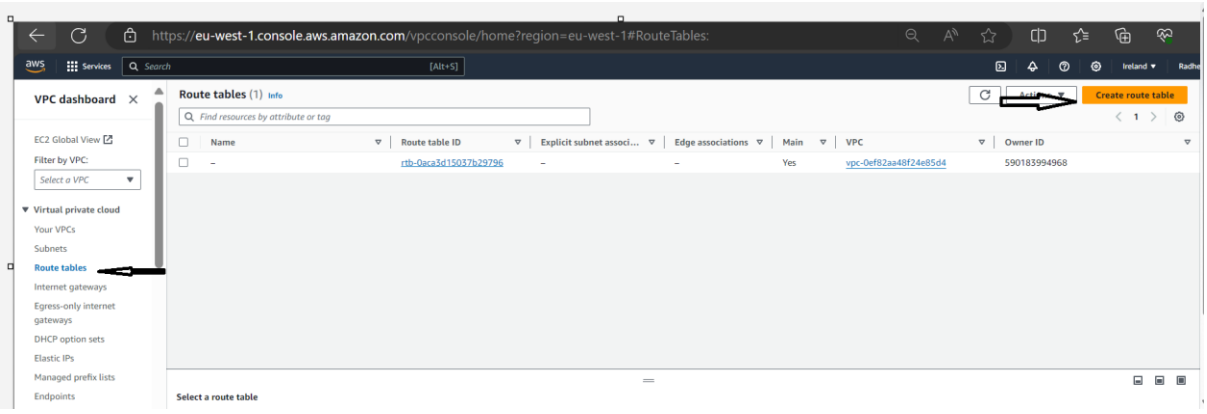
- Fill in the Name, choose one of the public subnets you created in part 2, and then allocate an Elastic IP. Click Create NAT gateway.

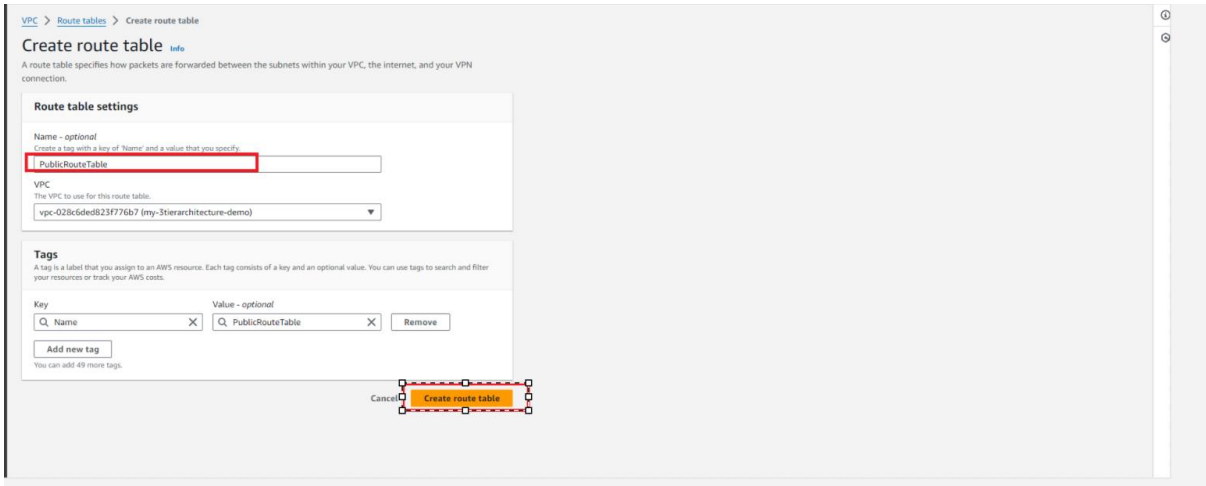


- Repeat step 1 and 2 for the other subnet.

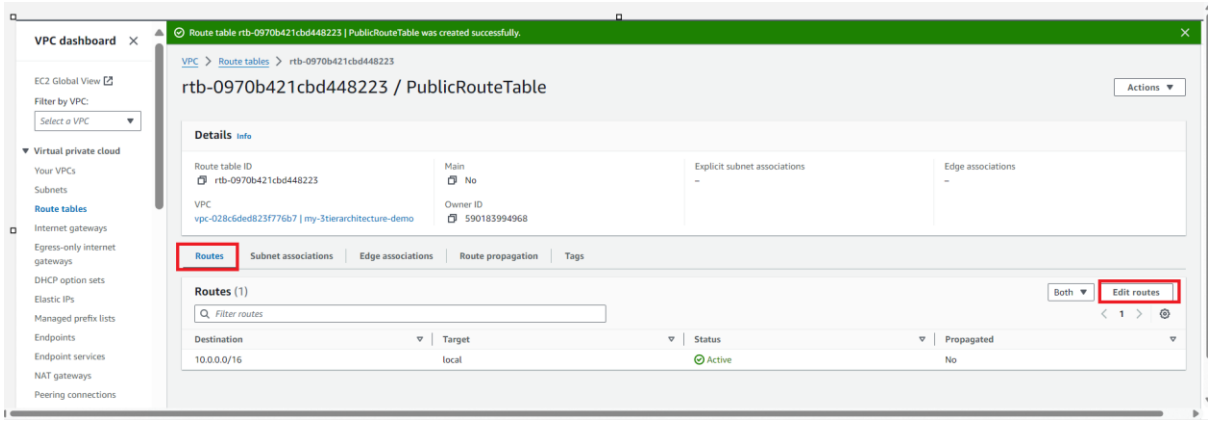
Routing Configuration

- Navigate to Route Tables on the left side of the VPC dashboard and click Create route table First, let's create one route table for the web layer public subnets and name it accordingly.

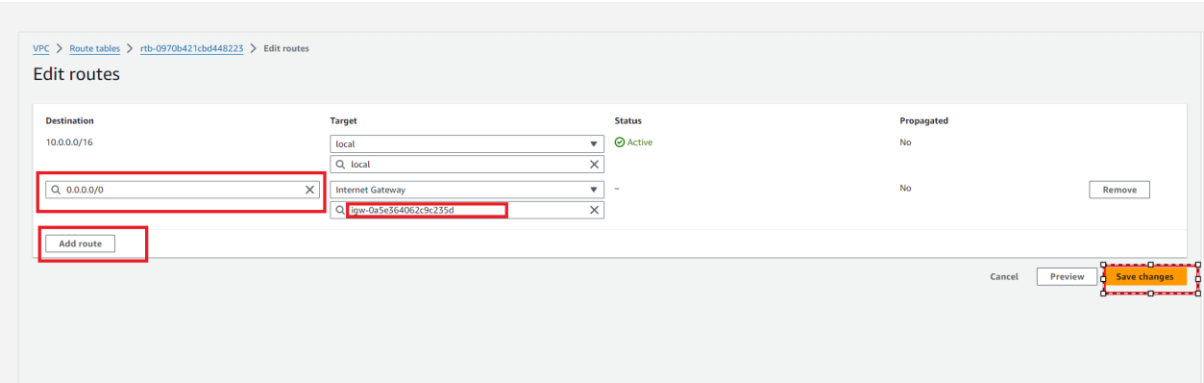




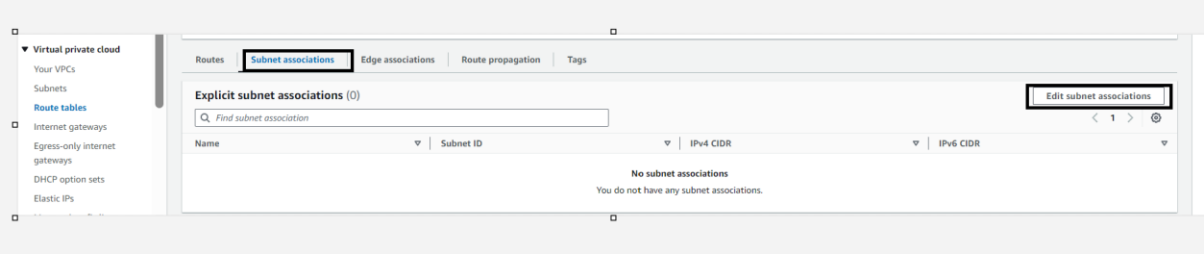
- After creating the route table, you'll automatically be taken to the details page. Scroll down and click on the Routes tab and Edit routes.



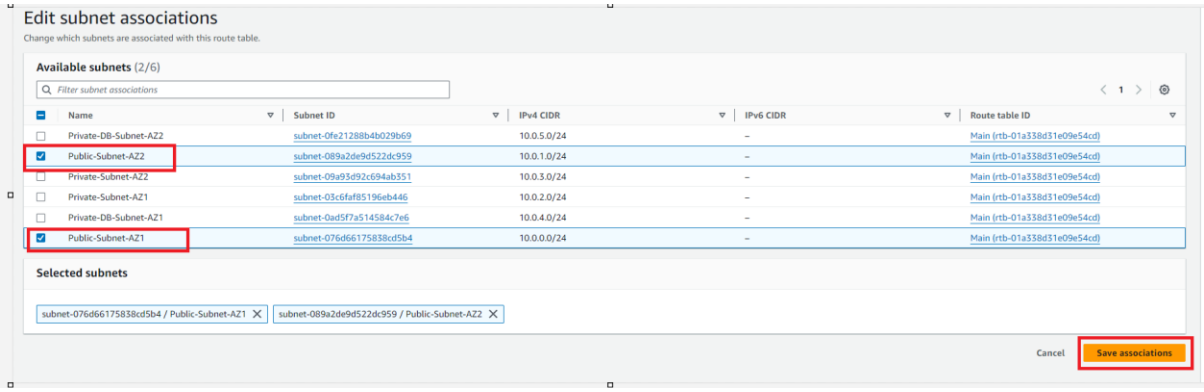
- Add a route that directs traffic from the VPC to the internet gateway. In other words, for all traffic destined for IPs outside the VPC CDIR range, add an entry that directs it to the internet gateway as a target. Save the changes.



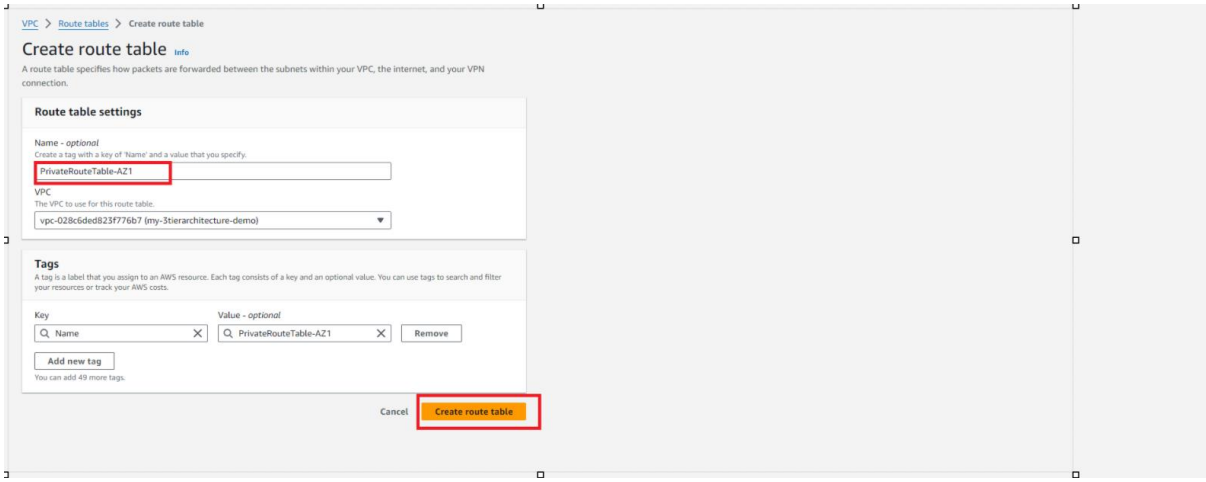
- Edit the Explicit Subnet Associations of the route table by navigating to the route table details again. Select Subnet Associations and click Edit subnet associations.

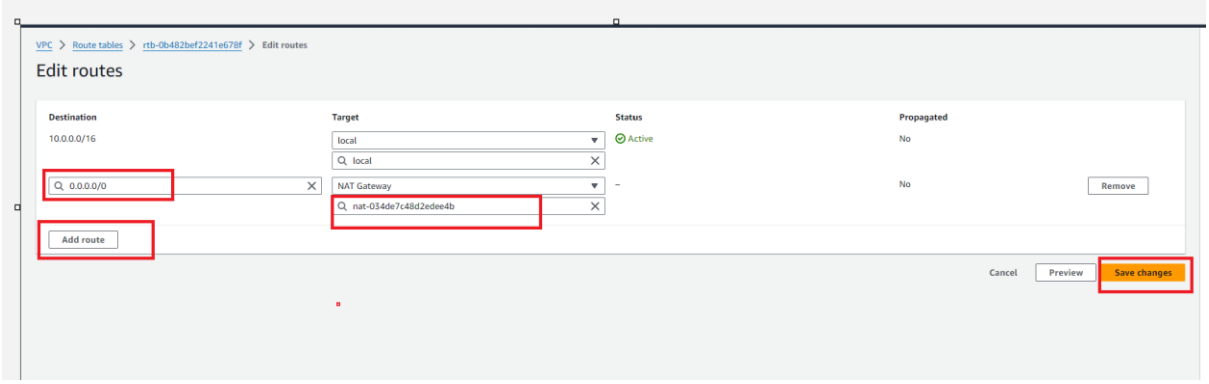


- Select the two web layer public subnets you created eariler and click **Save associations**.



- Now create 2 more route tables, one for each app layer private subnet in each availability zone. These route tables will route app layer traffic destined for outside the VPC to the NAT gateway in the respective availability zone, so add the appropriate routes for that.

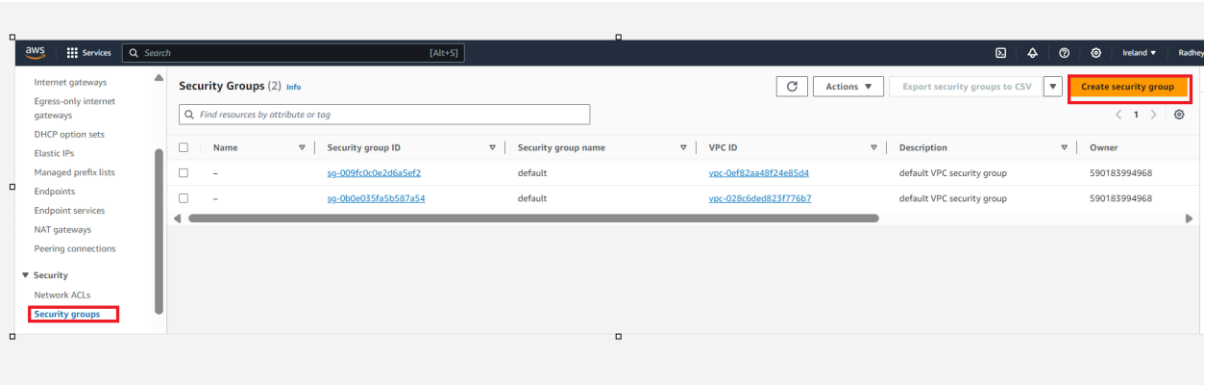




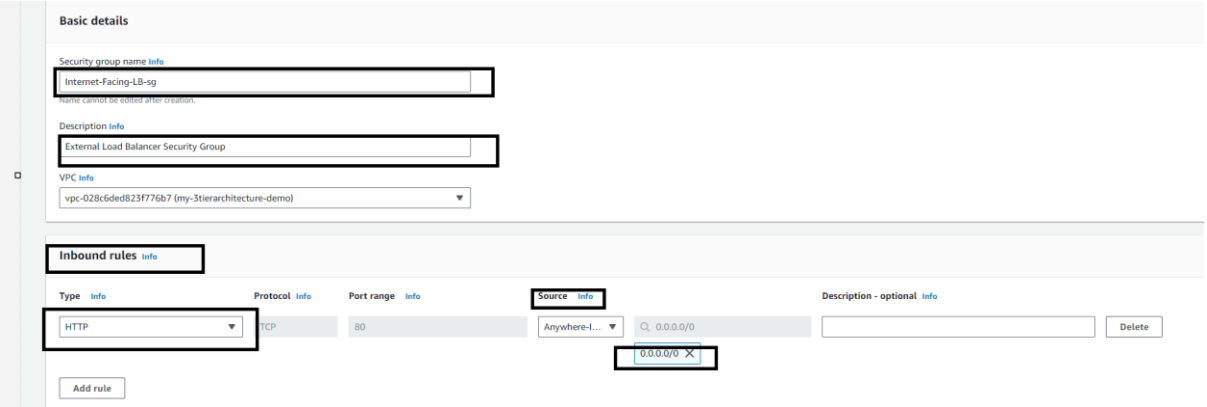
- Once the route tables are created and routes added, add the appropriate subnet associations for each of the app layer private subnets.

Security Groups

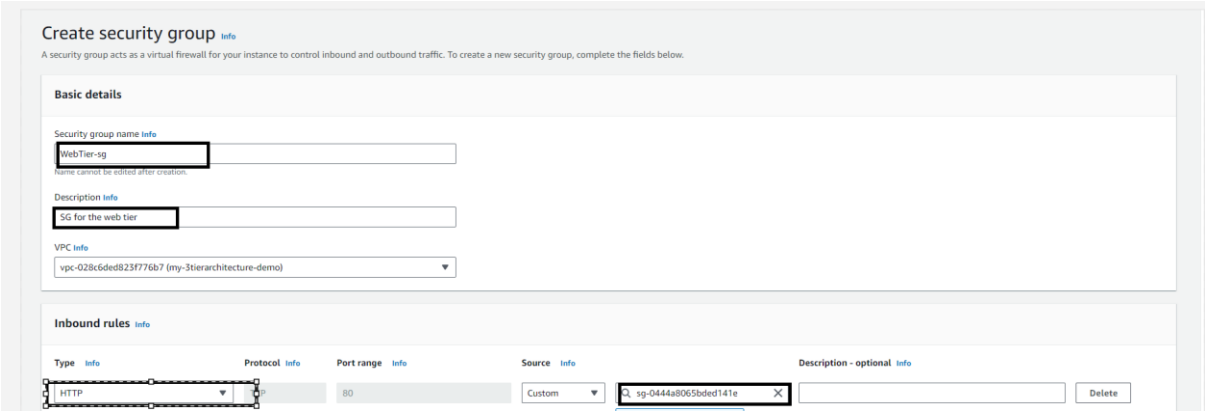
- Security groups will tighten the rules around which traffic will be allowed to our Elastic Load Balancers and EC2 instances. Navigate to Security Groups on the left side of the VPC dashboard, under Security.



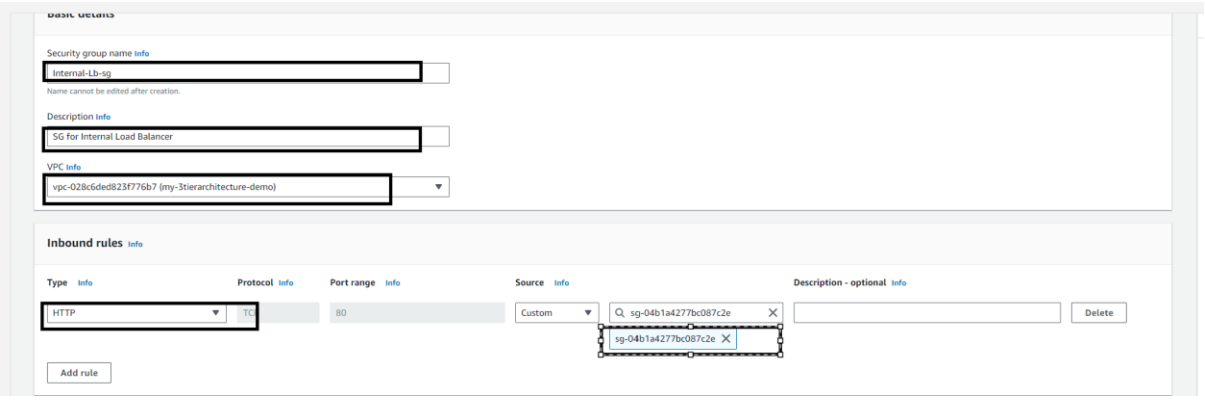
- The first security group you'll create is for the public, internet facing load balancer. After typing a name and description, add an inbound rule to allow HTTP type traffic.



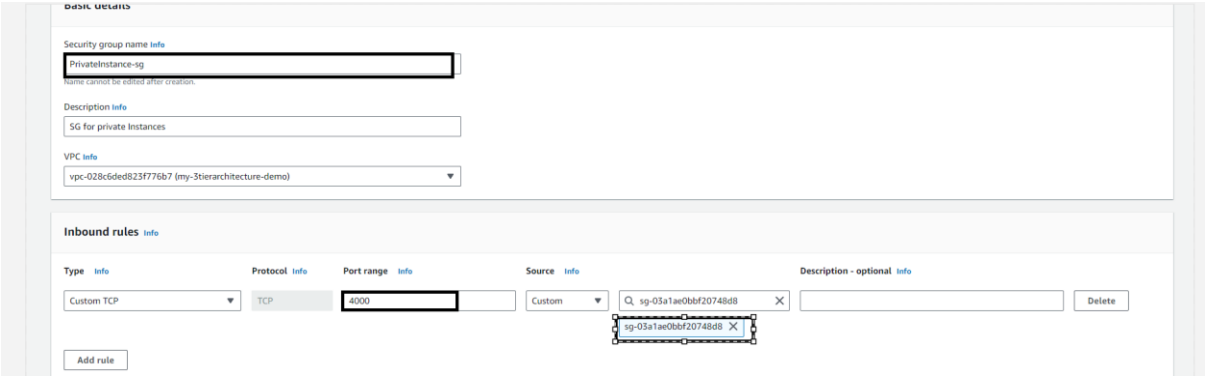
- The second security group you'll create is for the public instances in the web tier. After typing a name and description, add an inbound rule that allows HTTP type traffic from your internet facing load balancer security group you created in the previous step. This will allow traffic from your public facing load balancer to hit your instances.



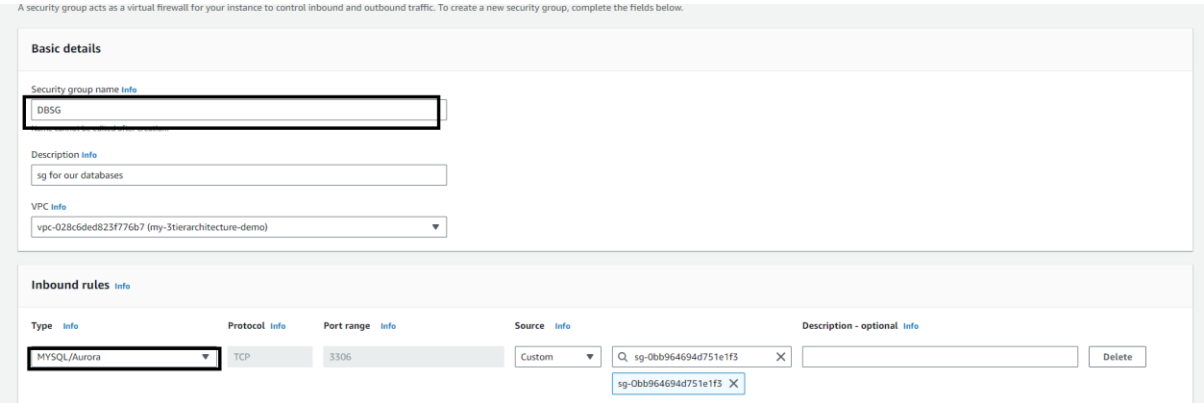
- The third security group will be for our internal load balancer. Create this new security group and add an inbound rule that allows HTTP type traffic from your public instance security group. This will allow traffic from your web tier instances to hit your internal load balancer.



- The fourth security group we'll configure is for our private instances. After typing a name and description, add an inbound rule that will allow TCP type traffic on port 4000 from the internal load balancer security group you created in the previous step. This is the port our app tier application is running on and allows our internal load balancer to forward traffic on this port to our private instances.

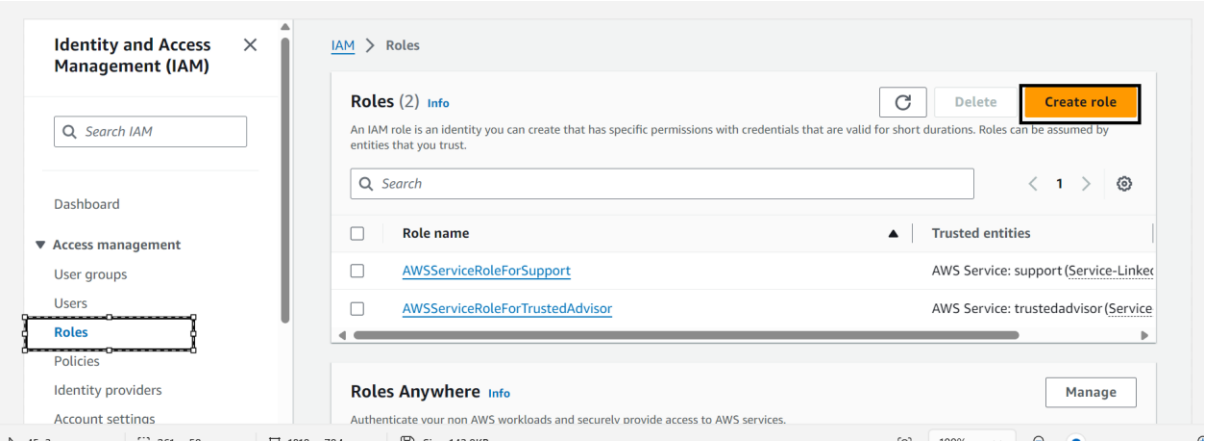


- The fifth security group we'll configure protects our private database instances. For this security group, add an inbound rule that will allow traffic from the private instance security group to the MYSQL/Aurora port (3306).

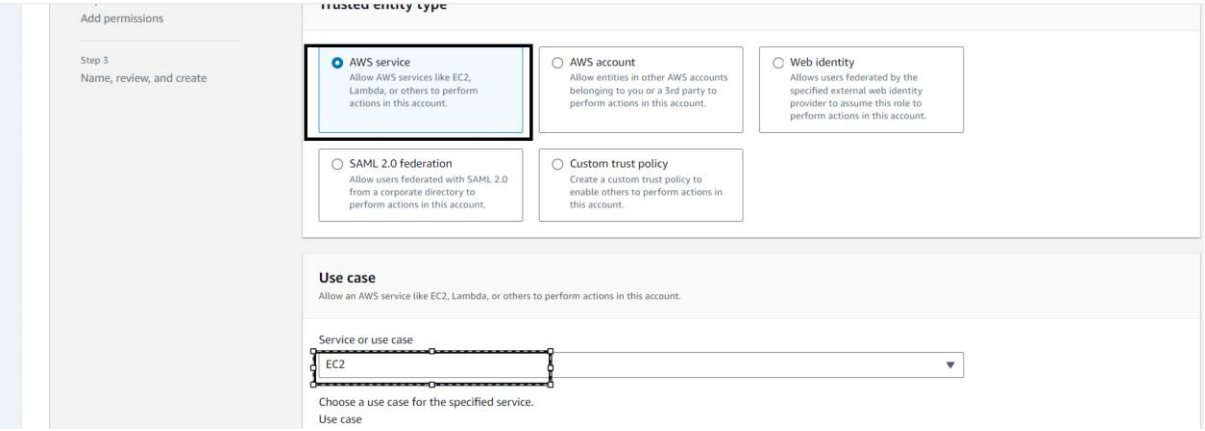


IAM EC2 Instance Role Creation

- Navigate to the IAM dashboard in the AWS console and create an EC2 role.

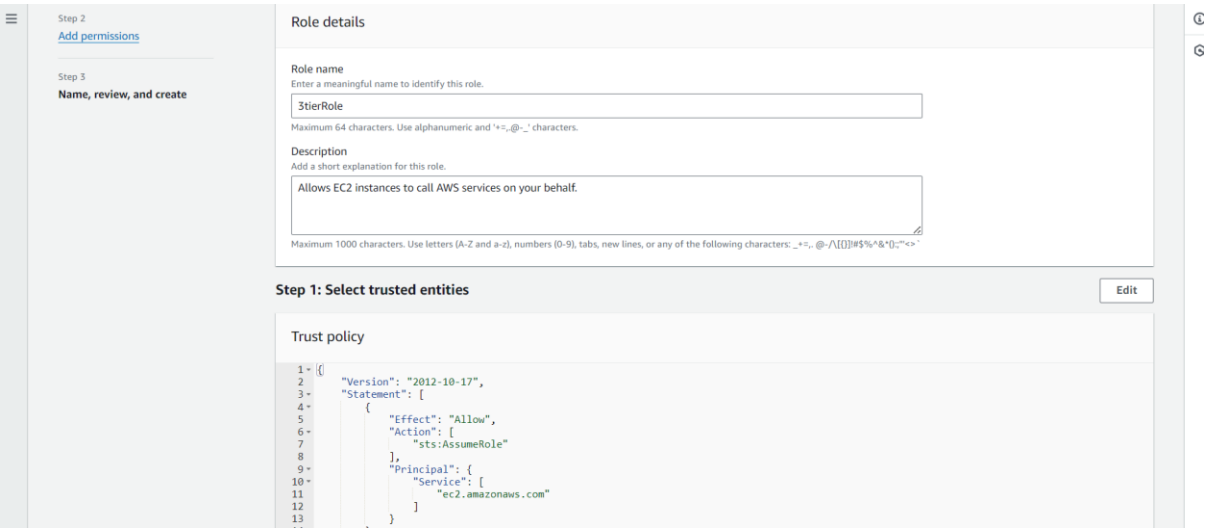


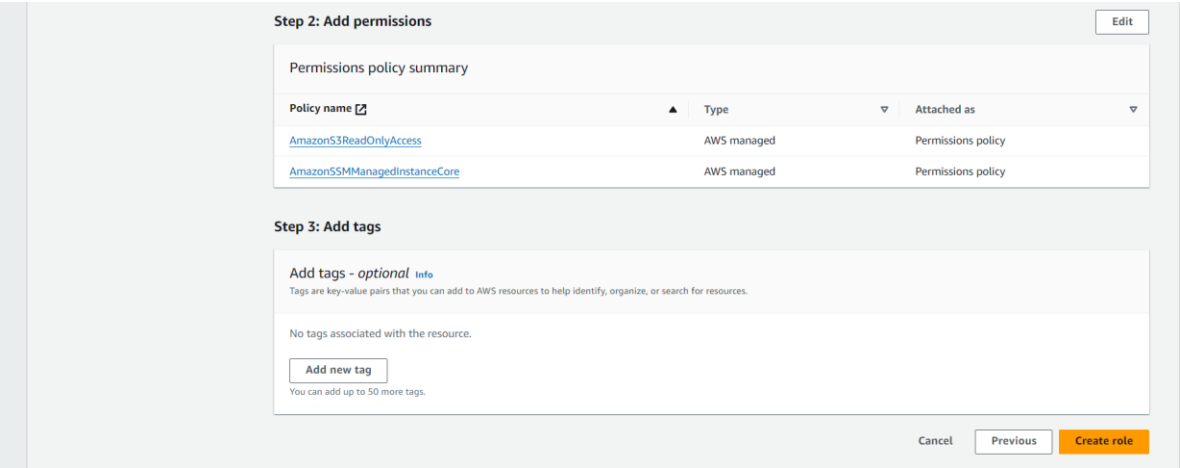
- Select EC2 as the trusted entity.



- When adding permissions, include the following AWS managed policies. You can search for them and select them. These policies will allow our instances to download our code from S3 and use Systems Manager Session Manager to securely connect to our instances without SSH keys through the AWS console.

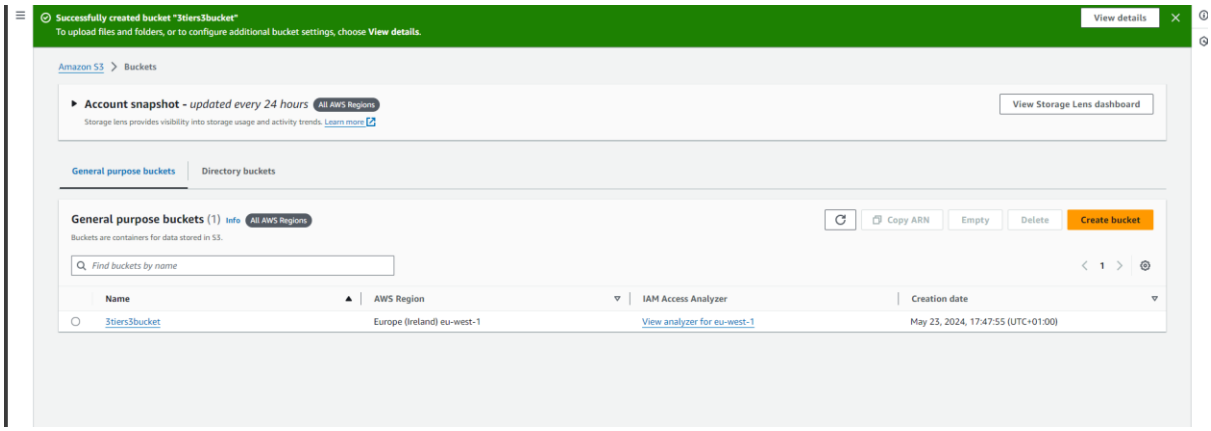
  - ✓ AmazonSSMManagedInstanceCore
  - ✓ AmazonS3ReadOnlyAccess





S3 Bucket Creation

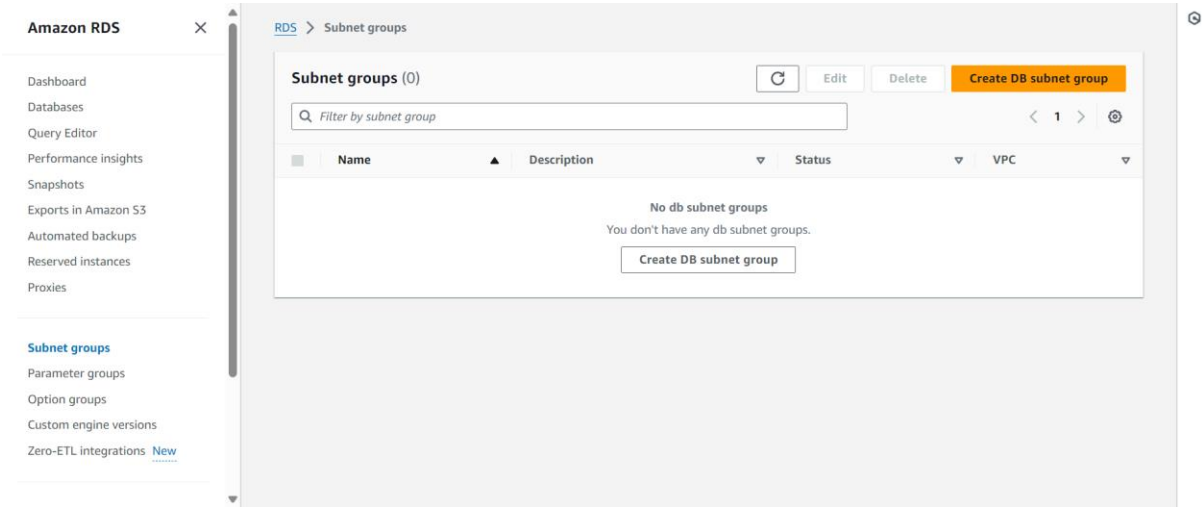
- Navigate to the S3 service in the AWS console and create a new S3 bucket.
- Give it a unique name, and then leave all the defaults as in. Make sure to select the region that you intend to run this whole lab in. This bucket is where we will upload our code later.



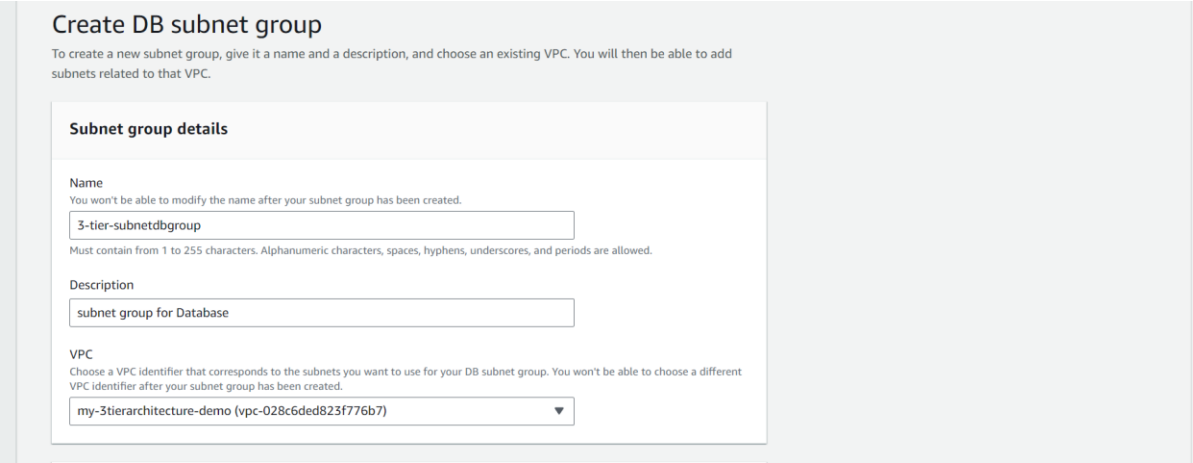
DATABASE DEPLOYMENT:

Subnet Groups

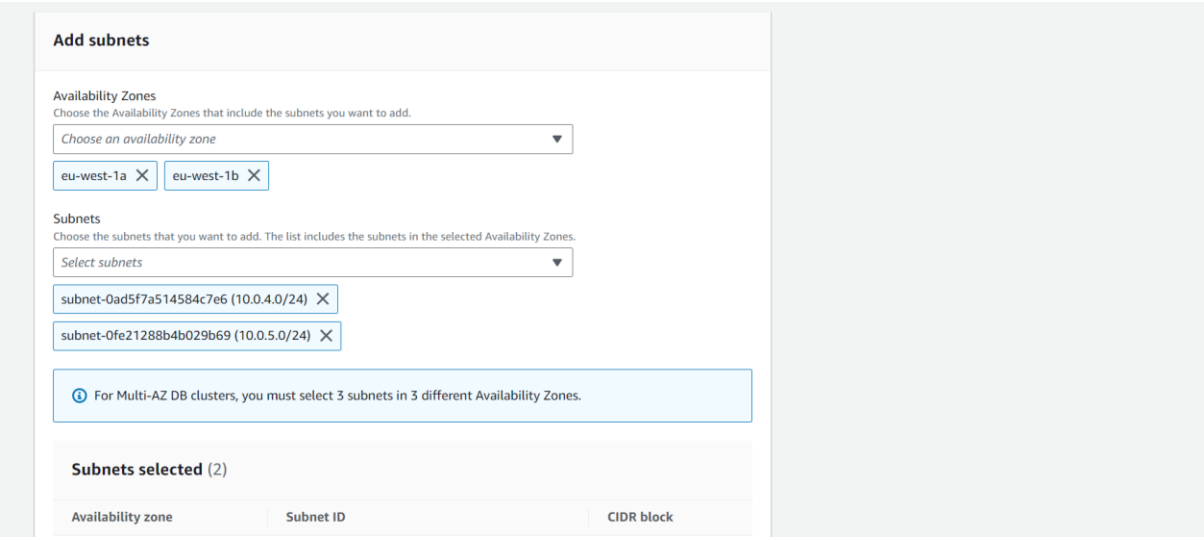
- Navigate to the RDS dashboard in the AWS console and click on Subnet groups on the left hand side. Click Create DB subnet group.

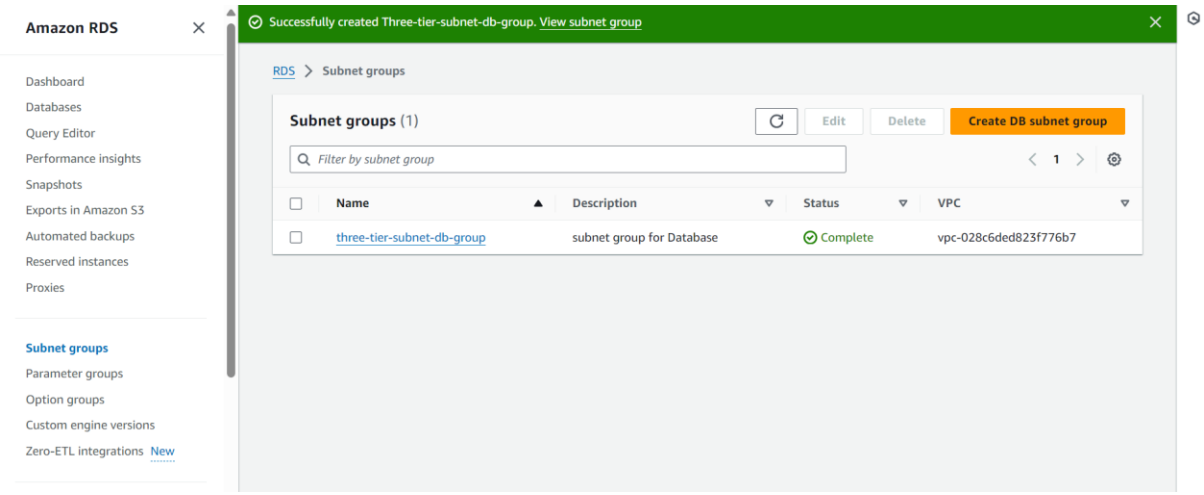


- Give your subnet group a name, description, and choose the VPC we created.



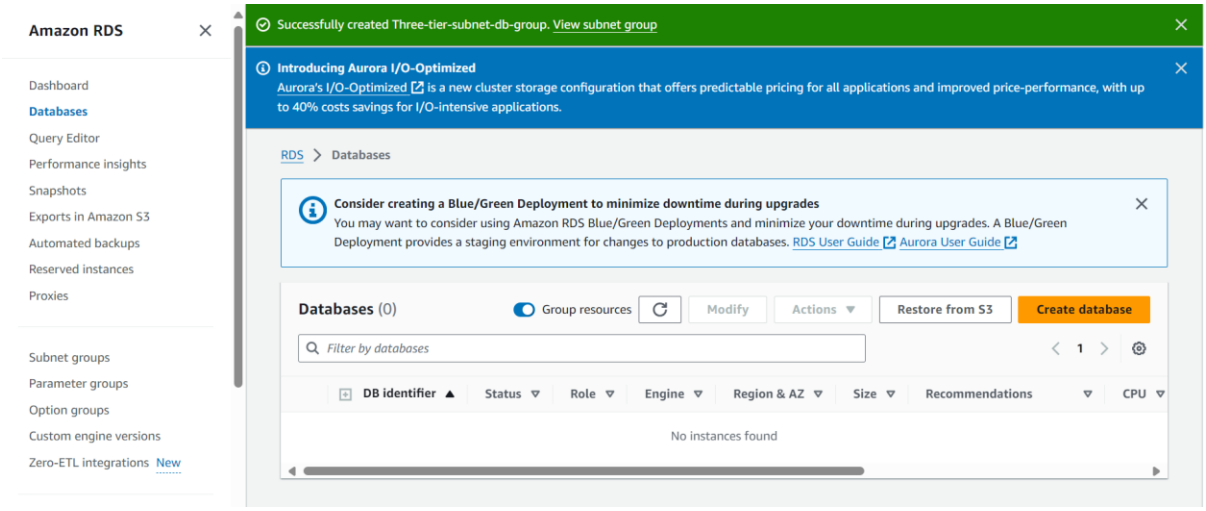
When adding subnets, make sure to add the subnets we created in each availability zone specificaly for our database layer. You may have to navigate back to the VPC dashboard and check to make sure you're selecting the correct subnet IDs.

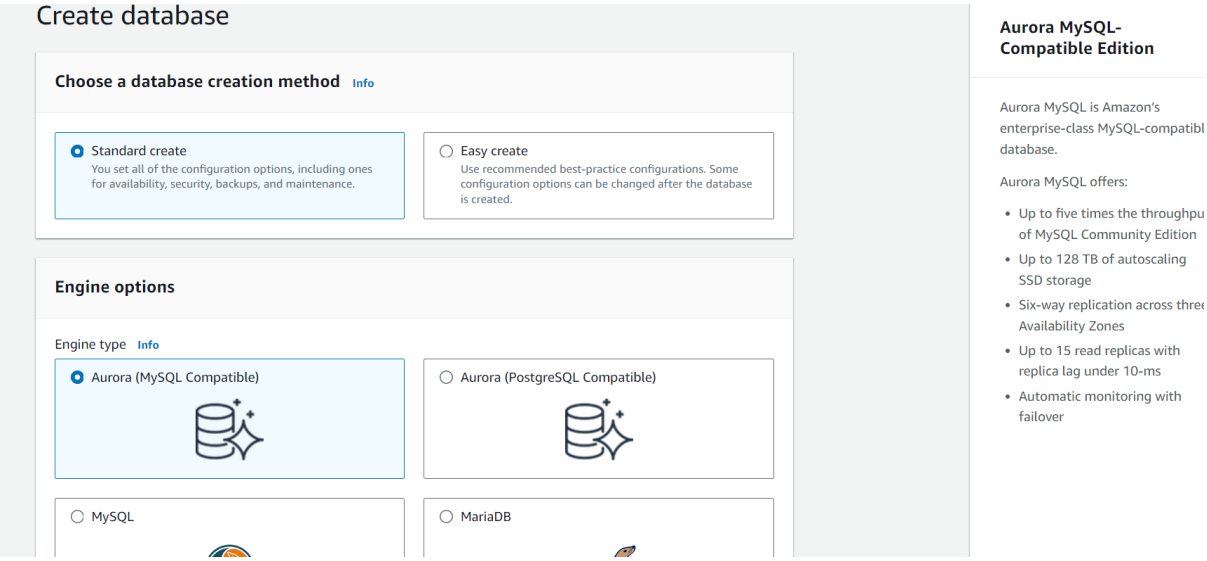




Database Deployment

Navigate to Databases on the left hand side of the RDS dashboard and click Create database.


Amazon RDS
Dashboard
Databases
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 New
Successfully created Three-tier-subnet-db-group. View subnet group
Introducing Aurora I/O-Optimized
Aurora's I/O-Optimized is a new cluster storage configuration that offers predictable pricing for all applications and improved price-performance, with up to 40% costs savings for I/O-intensive applications.
RDS > Databases
Consider creating a Blue/Green Deployment to minimize downtime during upgrades
You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide
Databases (0)
Group resources
Modify
Actions
Restore from S3
Create database
Filter by databases
DB identifier
Status
Role
Engine
Region & AZ
Size
Recommendations
CPU
No instances found


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
MariaDB
Aurora MySQL-Compatible Edition
Aurora MySQL is Amazon's enterprise-class MySQL-compatibl database.
Aurora MySQL offers:
Up to five times the throughpu of MySQL Community Edition
Up to 128 TB of autoscaling SSD storage
Six-way replication across thre Availability Zones
Up to 15 read replicas with replica lag under 10-ms
Automatic monitoring with failover

Aurora MySQL 3.05.2 (compatible with MySQL 8.0.32) - default for major version 8.0

⚠ Parallel query is off by default. To enable it, use a DB instance parameter group with the aurora_parallel_query parameter enabled. Learn more

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.

Settings

DB cluster identifier **Info**

Enter a name for your DB cluster. The name must be unique across all DB clusters owned by your AWS account in the current AWS Region.

database-1

The DB cluster identifier is case-insensitive, but is stored as all lowercase (as in "mydbcluster"). 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.

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 32 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**

••••••••••

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

Confirm master password **Info**

••••••••••

Aurora MySQL-Compatible Edition

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

Configuration options

Database instance, storage, and I/O charges vary depending on the configuration. Learn more

- Aurora Standard
  - Cost-effective pricing for many applications with moderate I/O usage (I/O costs <25% of total database costs).
  - Pay-per-request I/O charges apply. DB instance and storage prices don't include I/O usage.
- Aurora I/O-Optimized
  - Predictable pricing for all applications. Improved price performance for I/O-intensive applications (I/O costs >25% of total database costs).
  - No additional charges for read/write I/O operations. DB instance and storage prices include I/O usage.

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class Info

Hide filters

Include previous generation classes

- Serverless v2
- Memory optimized classes (includes r classes)
- Burstable classes (includes t classes)

db.r6g.2xlarge
8 vCPUs 64 GiB RAM Network: 4,750 Mbps

Aurora MySQL-Compatible Edition

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

Multi-AZ deployment Info

- Create an Aurora Replica or Reader node in a different AZ (recommended for scaled availability)
  Creates an Aurora Replica for fast failover and high availability.
- Don't create an Aurora Replica

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.

Network type Info

To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

- IPv4
  Your resources can communicate only over the IPv4 addressing protocol.
- Dual-stack mode
  Your resources can communicate over IPv4, IPv6, or both.

Virtual private cloud (VPC) Info

Choose the VPC. The VPC defines the virtual networking environment for this DB cluster.

Aurora MySQL-Compatible Edition

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

Choose the VPC. The VPC defines the virtual networking environment for this DB cluster.

my-3tierarchitecture-demo (vpc-028c6ded823f776b7)
6 Subnets, 2 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 cluster can use in the VPC that you selected.

three-tier-subnet-db-group
2 Subnets, 2 Availability Zones

Public access Info

- Yes
  RDS assigns a public IP address to the cluster. Amazon EC2 instances and other resources outside of the VPC can connect to your cluster. Resources inside the VPC can also connect to the cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.
- No
  RDS doesn't assign a public IP address to the cluster. Only Amazon EC2 instances and other resources inside the VPC can connect to your cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.

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.

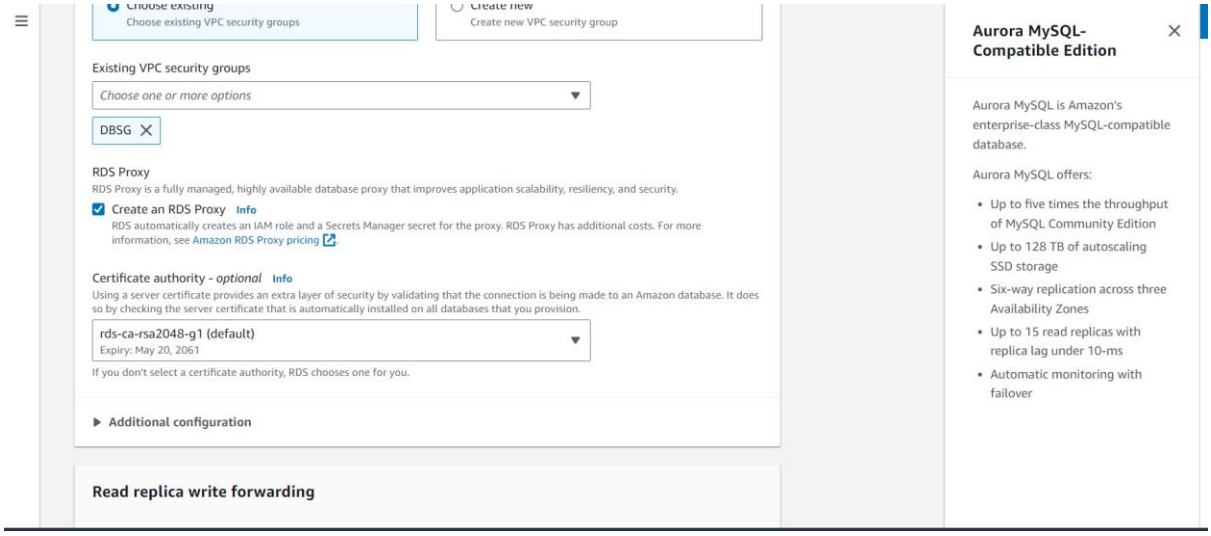
- Choose existing
  Choose existing VPC security groups
- Create new
  Create new VPC security group

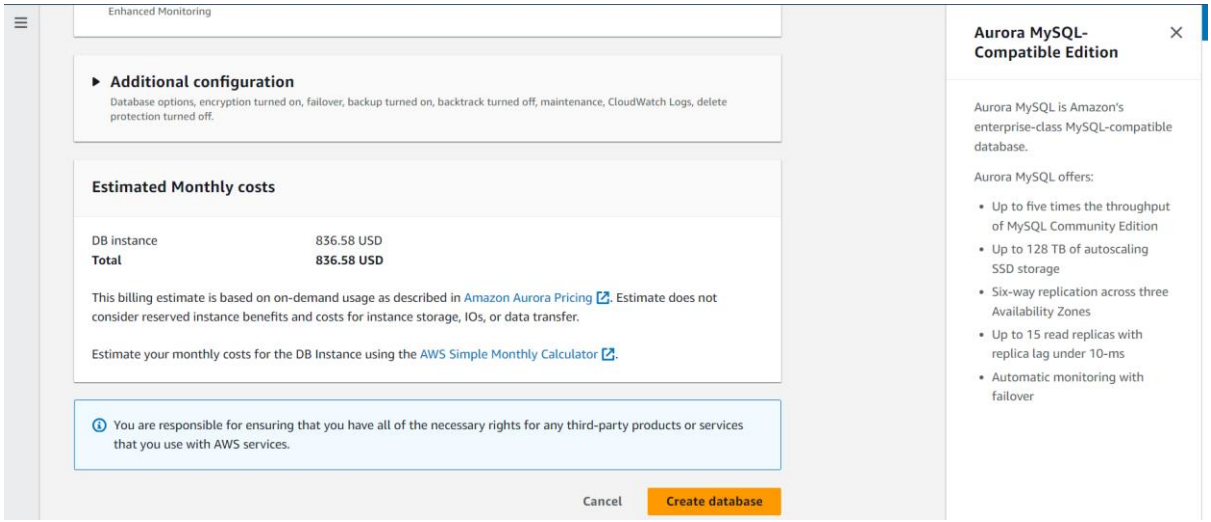
Aurora MySQL-Compatible Edition

Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

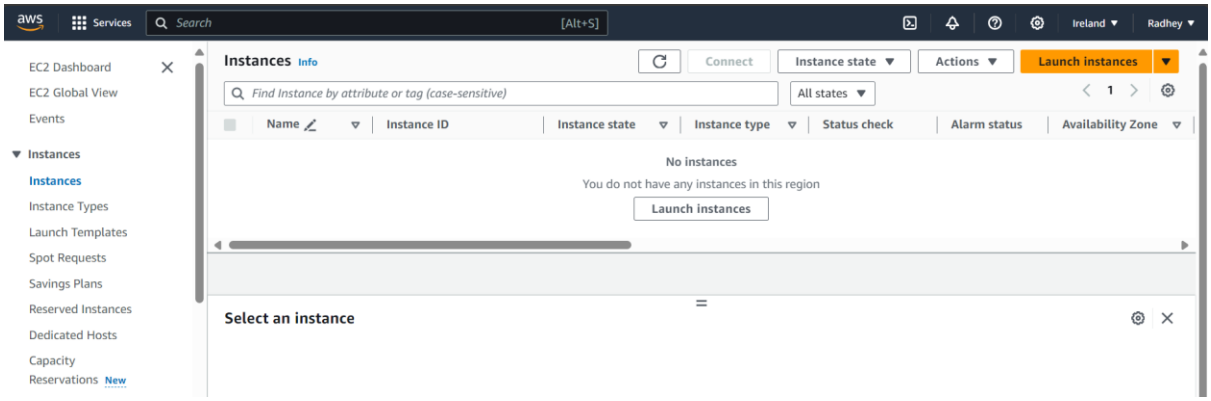
- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover



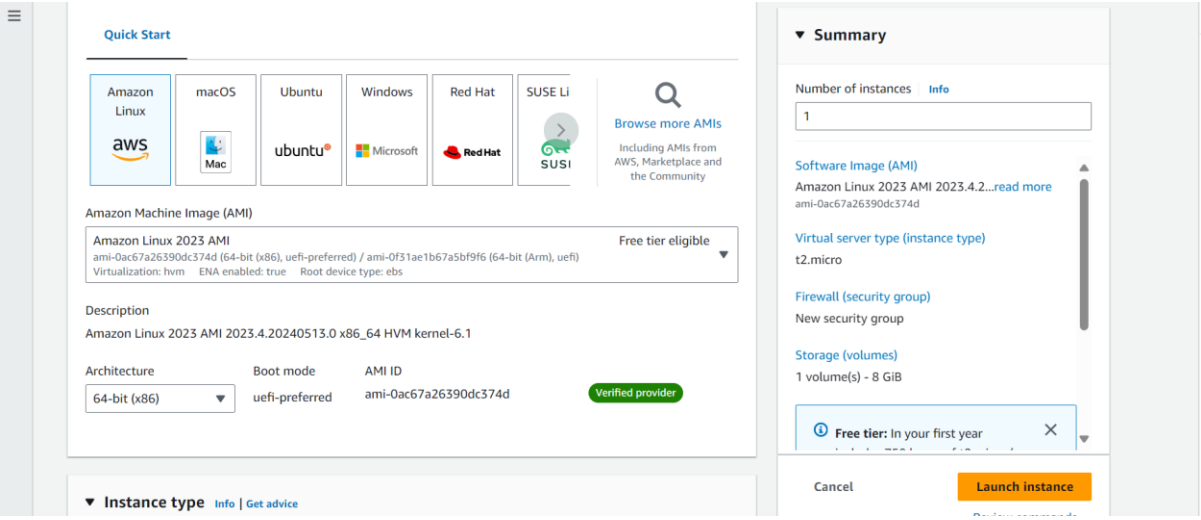


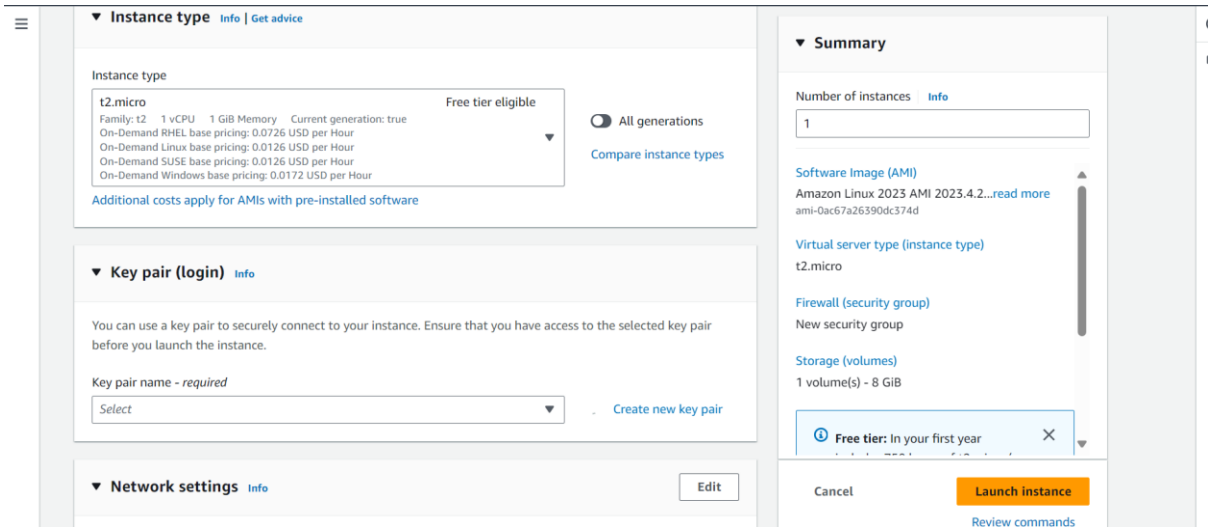
App Instance Deployment

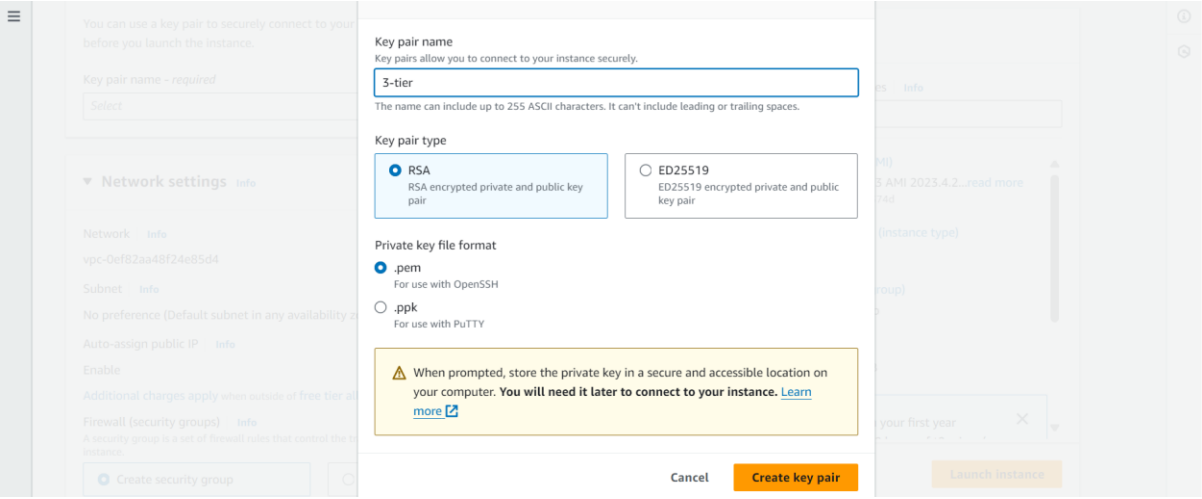
1. Navigate to the EC2 service dashboard and click on **Instances** on the left hand side. Then, click **Launch Instances**.

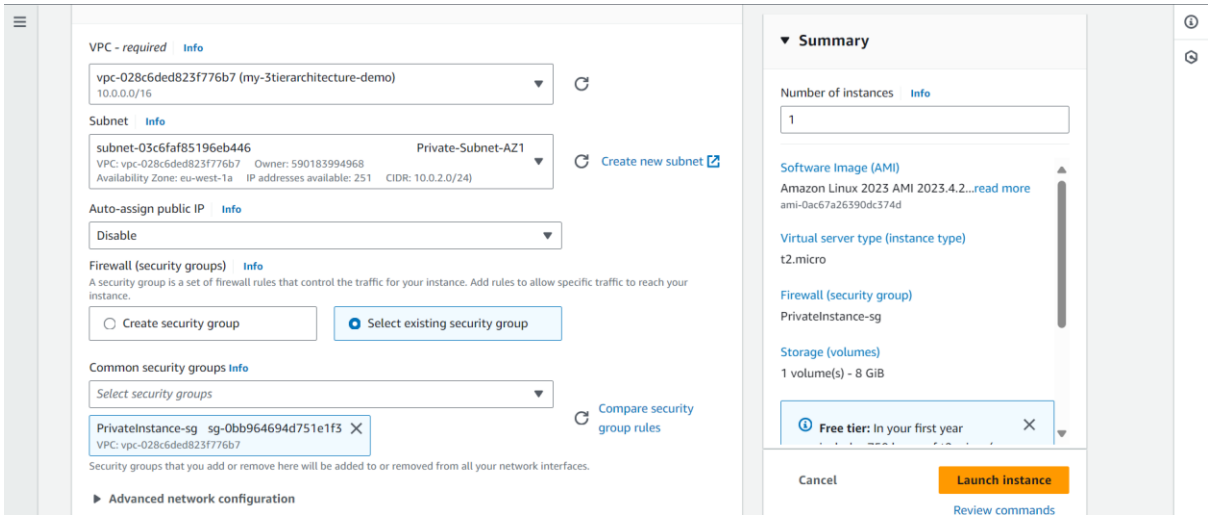


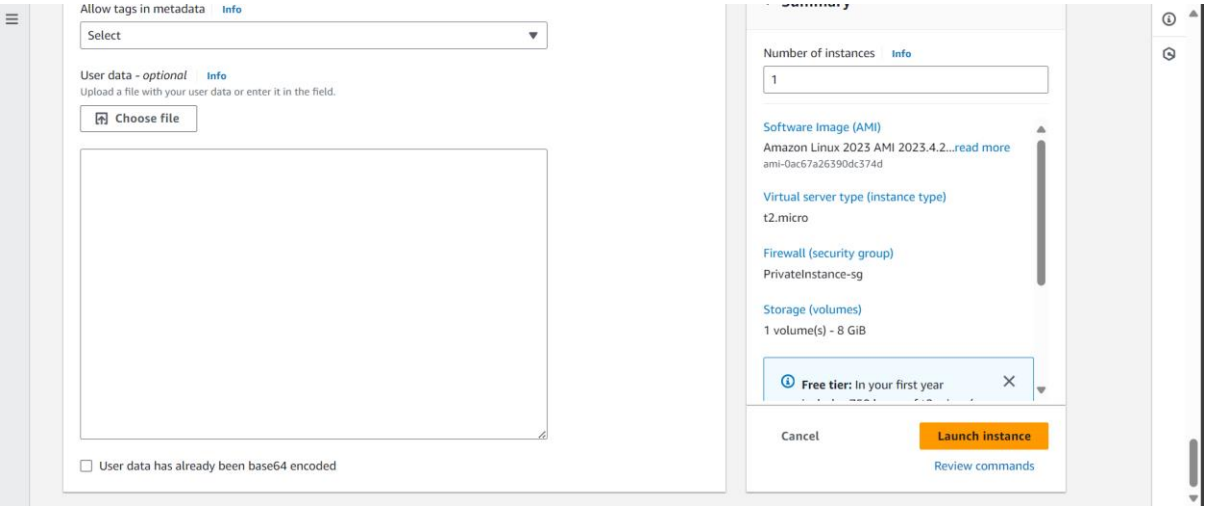
2. Select the first **Amazon Linux 2 AMI**






Key pair name
Key pairs allow you to connect to your instance securely.
3-tier
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.
Key pair type
RSA
RSA encrypted private and public key pair
ED25519
ED25519 encrypted private and public key pair
Private key file format
.pem
For use with OpenSSH
.ppk
For use with PuTTY
When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. Learn more
Cancel
Create key pair


VPC - required Info
vpc-028c6ded823f776b7 (my-3tierarchitecture-demo)
10.0.0.0/16
Subnet Info
subnet-03c6faf85196eb446
Private-Subnet-AZ1
VPC: vpc-028c6ded823f776b7 Owner: 590183994968
Availability Zone: eu-west-1a IP addresses available: 251 CIDR: 10.0.2.0/24)
Create new subnet
Auto-assign public IP Info
Disable
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
Common security groups Info
Select security groups
PrivateInstance-sg sg-0bb964694d751e1f3
VPC: vpc-028c6ded823f776b7
Compare security group rules
Security groups that you add or remove here will be added to or removed from all your network interfaces.
Advanced network configuration
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.4.2...read more
ami-0ac67a26390dc374d
Virtual server type (instance type)
t2.micro
Firewall (security group)
PrivateInstance-sg
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year
Cancel
Launch instance
Review commands



Once the instance successfully Launched, login to EC2 insstance and configure database and application and test Application Tier.

Internal Load Balancing and Auto Scaling