VPC and Subnets

**VPC Creation**

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

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

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.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

* 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.

A screenshot of a computer

Description automatically generated

**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.

A screenshot of a computer

Description automatically generated

* 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.

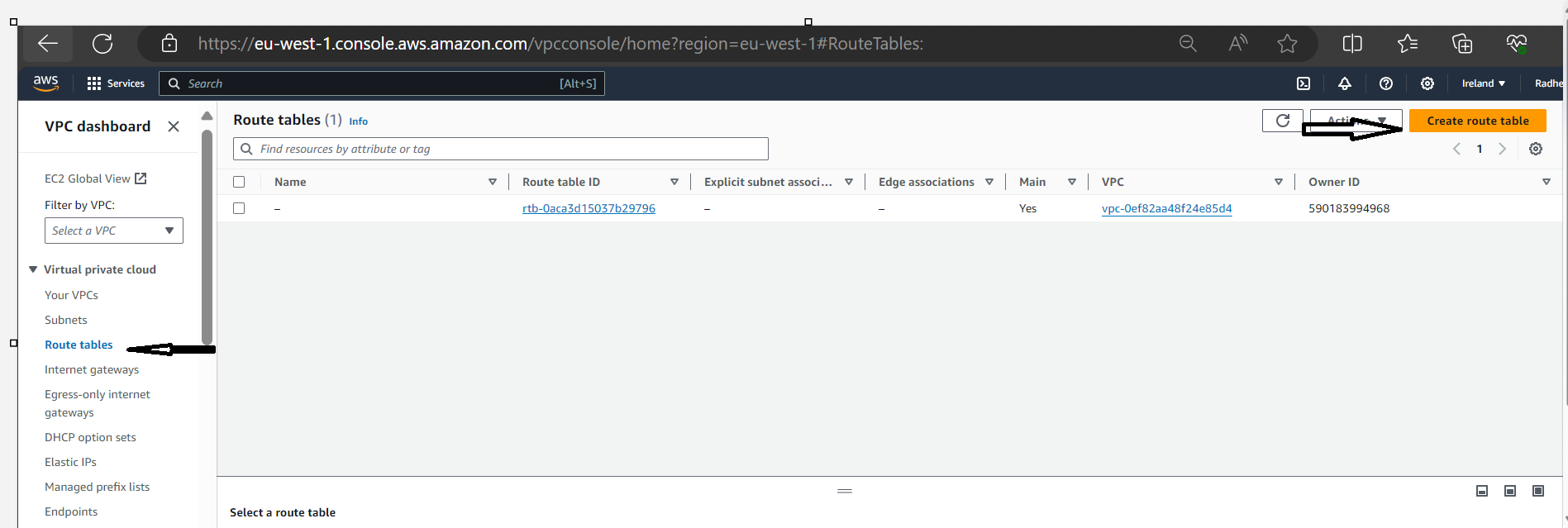
A screenshot of a computer

Description automatically generated

* 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.



A screenshot of a computer

Description automatically generated

* 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.

A screenshot of a computer

Description automatically generated

* 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.

A screenshot of a computer

Description automatically generated

* 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.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

* 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.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* 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.

A screenshot of a computer

Description automatically generated

* 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.

A screenshot of a computer

Description automatically generated

* 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.

A screenshot of a computer

Description automatically generated

* 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.

A screenshot of a computer

Description automatically generated

* 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.

A screenshot of a computer

Description automatically generated

* 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).

A screenshot of a computer

Description automatically generated

# IAM EC2 Instance Role Creation

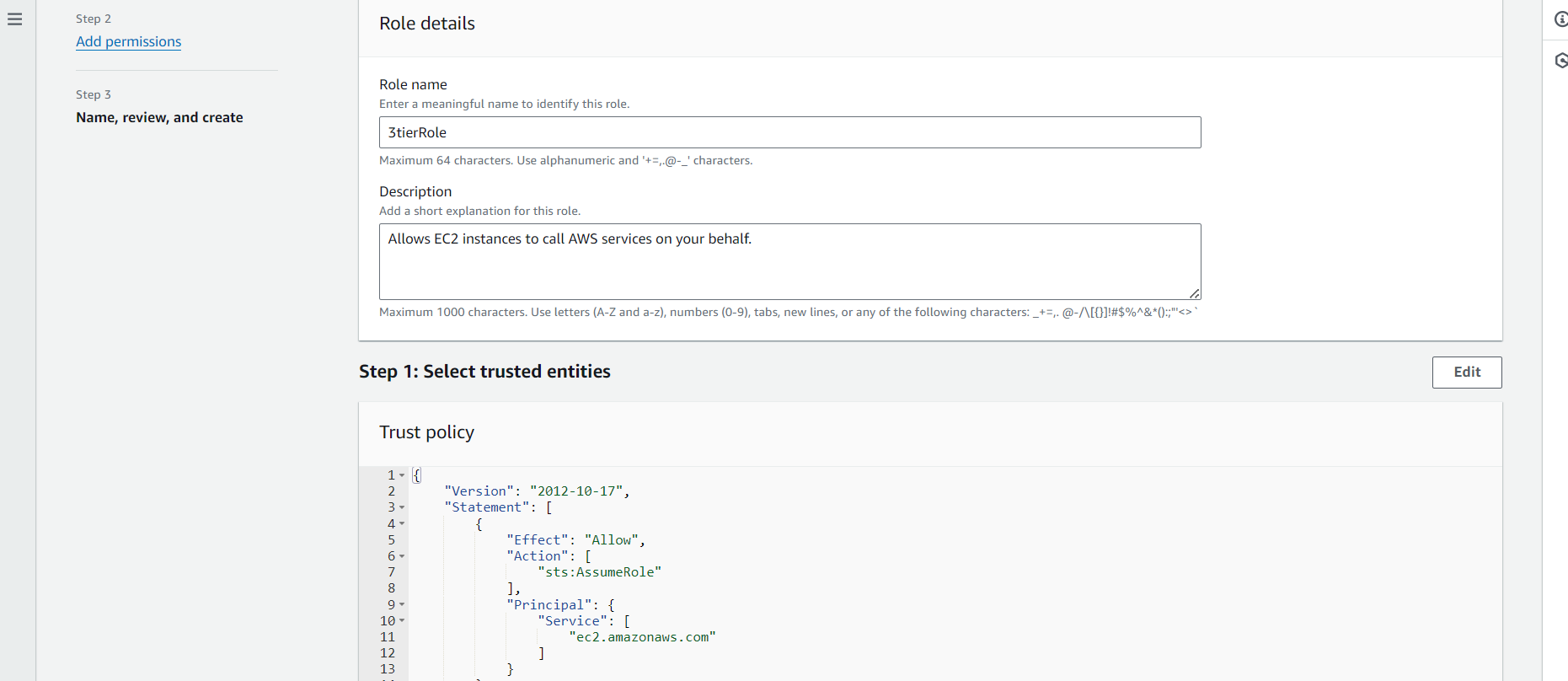
* Navigate to the IAM dashboard in the AWS console and create an EC2 role.

# A screenshot of a computer Description automatically generated

* Select EC2 as the trusted entity.

# A screenshot of a computer Description automatically generated

* 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



A screenshot of a computer

Description automatically generated

# 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.

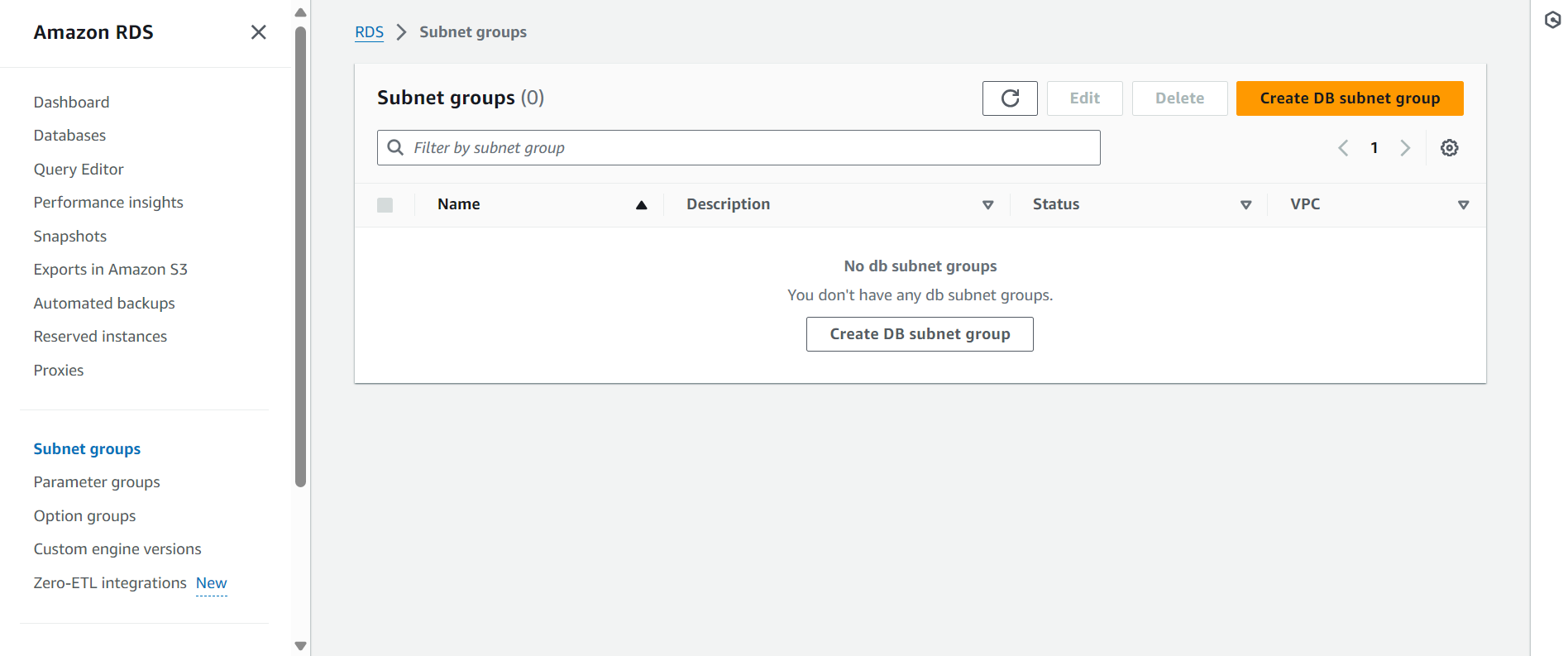
A screenshot of a computer

Description automatically generated

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.

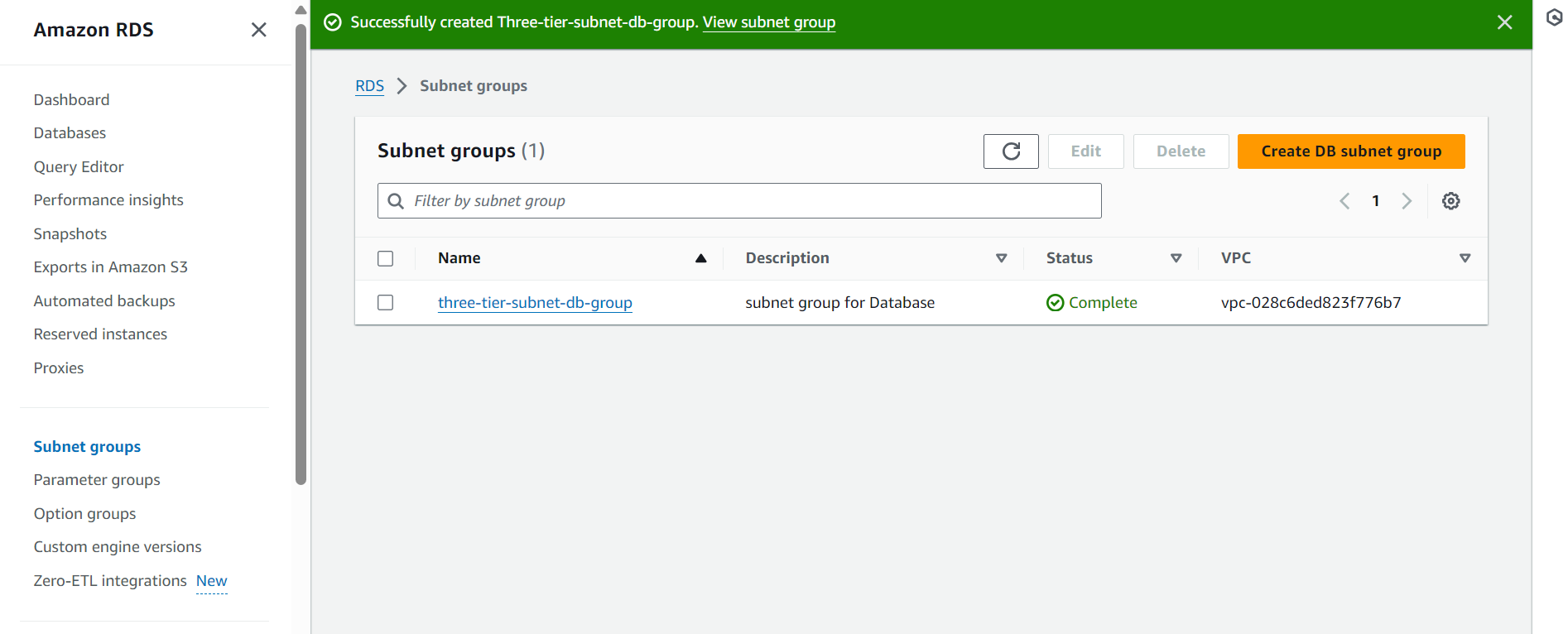
A screenshot of a computer

Description automatically generated

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.

A screenshot of a computer

Description automatically generated



# Database Deployment

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

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# App Instance Deployment

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

# A screenshot of a computer Description automatically generated

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

# A screenshot of a computer Description automatically generated

# A screenshot of a computer Description automatically generated

# A screenshot of a computer Description automatically generated

# A screenshot of a computer Description automatically generated

# A screenshot of a computer Description automatically generated

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

# Internal Load Balancing and Auto Scaling