**How-To build A Highly Available 3-Tier Architecture**

Graphical user interface

Description automatically generated

3-Tier Architecture diagram

**Now what is a 3-Tier Architecture?**

A 3-Tier Architecture is the most popular implementation of a multi-tier architecture and consist of a single web tier, application tier, and data tier. Each tier runs on its own infrastructure. Each tier can be developed simultaneously by a separate development team, and can be updated or scaled without impacting other tiers.

**Prerequisites:**

* AWS Account(Using free-tier here)
* Knowledge of using the AWS CLI
* Patience( This will be a long process and lots of topics covered)
* As always EAGERNESS TO LEARN!!



**Objectives:**

* Create a VPC
* Create a Web Tier
* Create a Application Tier
* Create a Database Tier

I have also created a diagram that shows my work which is the header of this article. I used the website <https://www.diagrams.net/> This has all of the AWS symbols available for you to create a diagram so I would create this as well to show your work.

**Step 1: Creating a VPC ( This is needed for the multi tier architecture)**

Once logged into your AWS Account you will type in VPC at the top of the search bar. You will then click on the VPC service. Once you are in the VPC dashboard click the orange “Create VPC” button.

Graphical user interface, application

Description automatically generated

When we are inside the VPC creator we will be selecting the VPC and more option. This allows us to create everything we need to associate with the VPC. We will select the Name for our VPC. I have name mine “Tier3Project”

The CIDR block address can stay as default which is how I left it. We will click 2x for the number of availability zone. We will choose 2x for the public subnets. We will choose 4x for the private subnets. We will go over this more as to why we are choosing these specific options.

Graphical user interface, text, application

Description automatically generated

We will choose the 1x per AZ for the NAT gateways. For the VPC endpoint make sure you choose none. The last step on this is clicking the orange button that says “Create VPC”

Graphical user interface, application

Description automatically generated

The VPC Creation will take a few minutes. Once you see the Success in green at the top you have created your own VPC. Click the view VPC at the bottom.

Graphical user interface, application

Description automatically generated

You will see the VPC that is created. It also shows the state as “Available”. Since the VPC created the Public & Private subnets we want to make some changes for them. This can be done by click on the Subnets tab on the left side.

Graphical user interface, application, Teams

Description automatically generated

We can see the 2x Public subnets that were created along with the 4x private subnets that were created.

Graphical user interface, table

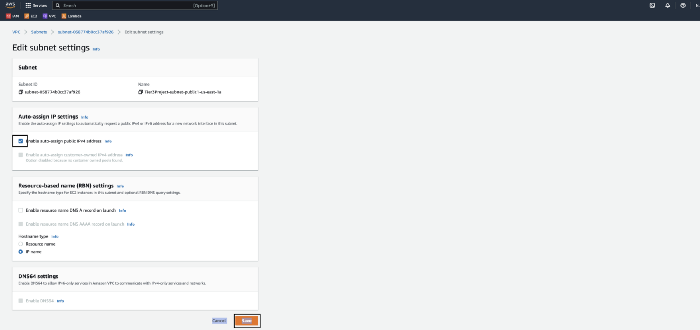
Description automatically generated

Now we will want to click on each of the public subnets and change the “Auto-assign public IPV4 Address” As you can see below when the subnet was created it has “No” under the “Auto-assign public IPV4 Address” in order to achieve this we will go to the top right of the page and click the “Action” button. we will then see a drop down box and we will want to choose the “Edit subnet settings” button.

Graphical user interface, application

Description automatically generated

We will then click the “Enable auto-assign IPv4 address” & Click the orange save button. Next we will start the web tier creation.



Graphical user interface, application

Description automatically generated

**Step 2: Creating a Web-Tier**

Listed below is what the Web-Tier will have

* 2x Public Subnets
* 2x EC2 Instances- Can have an OS of your choice
* EC2 Web server security group allowing inbound permission from the internet
* Boot strap static web page( We will be creating a script to achieve this)

We will start off by creating an Auto Scaling Group. This can be done by going into the EC2 dashboard. You can search for EC2 at the top of the search bar and click on EC2. When you are in the dashboard you will see at the bottom of the left pane Auto Scaling and under that section you should see Auto Scaling Group. Click the Auto Scaling Group button.

Graphical user interface, application

Description automatically generated

Click the Orange “Create Auto Scaling Group” button

Graphical user interface, application, website

Description automatically generated

Once we are inside the Auto Scaling We will create an Auto Scaling group name. Once we input a name we will then create a Launch template. You will then click the blue button that says “Create a launch template” Once we are in the Create Launch Template section we will name our launch template. You can name it to your liking but I names mine “Tier3ASG” Make sure ti select the Auto scaling guidance box. This will help set up a template to use with EC2 Auto Scaling. For the AMI image I click the Amazon Linux.

Graphical user interface, text, application, Word

Description automatically generated

We will then select the instance type. I chose the t2.micro because that falls under the Free-Tier. We will then create a new keypair. Make sure you remember where this key is stored for later. We will then create a new security group. We will name that security group which is required. We will then put a description of this SG I put to allow SSH & HTTP. We will also want to make sure this SG is linked to our VPC. We will then click add security group role. You will add the SSH & the HTTP rules. These can be found under “Type”. We will also click the “source type” which will be anywhere. If you performed this correctly you should see port 80 for HTTP and port 22 for SSH. When we allow anywhere you should see the CIDR address 0.0.0.0

Graphical user interface, application

Description automatically generated

Once you create the SG inbound rules you will then add a network network interface. Under the network interface you will choose the “Enable” option for IPv4 address.

Graphical user interface, text, application, email

Description automatically generated

After you have completed the above steps you will then go to the bottom of the page and click the “User data” option. Then scroll to the bottom and you will see the “User data” here is where we will create a script to install a web server from start up of the instance. Below is the scripts. After enter the script click the orange “Create launch template” button

Graphical user interface, application

Description automatically generated

#!/bin/bash  
yum update -y  
yum install -y httpd  
systemctl start httpd  
systemctl enable httpd  
echo "<html><body><h1>CONGRATULATIONS! BLACK TEAM IS THE BEST!!!</h1></body></html>" > /var/www/html/index.html

Graphical user interface, application

Description automatically generated

You should see a success message.

Graphical user interface, text, application, email

Description automatically generated

Now we will go back to the Auto scaling page we started creating before. You will then refresh the Launch template and should be able to choose the instance we just created. After that click the orange “Next” button

Graphical user interface, application

Description automatically generated

We should now see the Network section of Auto Scaling. We want to make sure our VPC we created is selected. If we don’t choose the VPC we created we wont see the different subnets we created. Make sure to choose both of the public subnets. Now pay attention because I first chose a public and a private. If we do this we wont be able to complete the tasks at hand. If you are all done click the orange “Next” button to move forward.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

We will now Attach a new load balancer as well. We are doing this now instead of taking the extra steps later. Choose the option “Attach to a new load balancer”. Under Load balancer type click the “Application Load Balancer”. We will now name our Load balancer. You can pick any name you would like. After picking a name then you will choose the Scheme type. This will be Internet-Facing.

Graphical user interface, application, Word

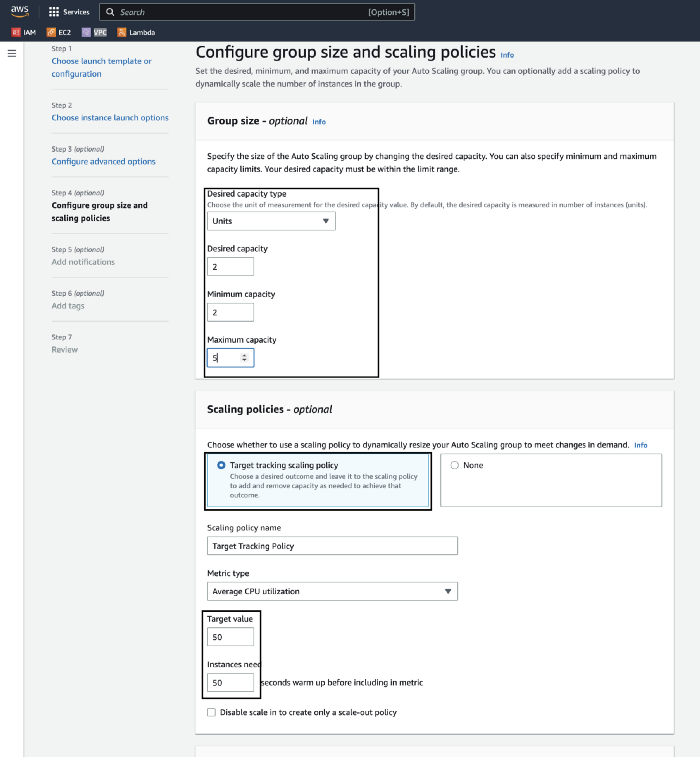
Description automatically generated

We will want to make sure under the Listeners and routing section we create a target group. I left the name to what AWS chose but you can pick anything you want. Also if you want to monitor the Auto Scaling you can click the “enable” box for CloudWatch. After this is completed click the orange “Next” button.

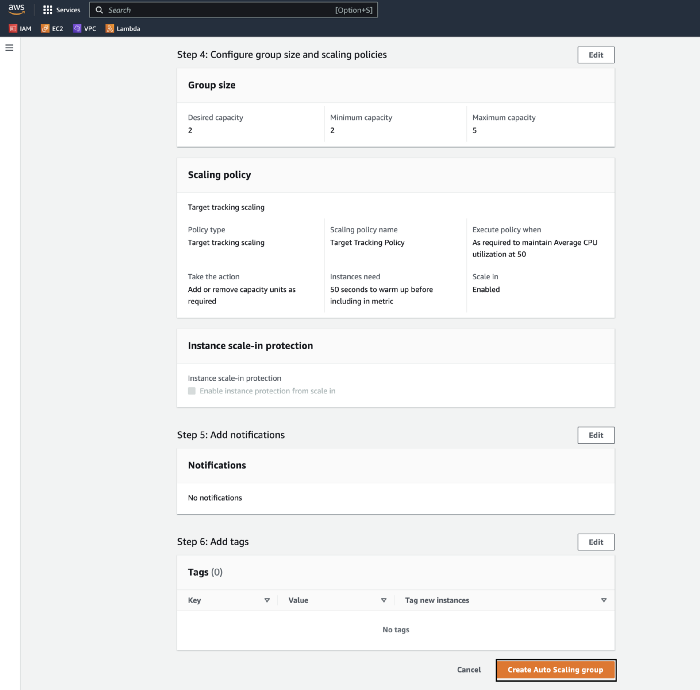
Graphical user interface, text, application, email

Description automatically generated

For the desired capacity type we want to choose the options below. The capacity will be 2x the minimum capacity will be 2x as well. The maximum capacity we will set at 5. Now you can skip to the review since we wont be doing anything with the step 5 or step 6.



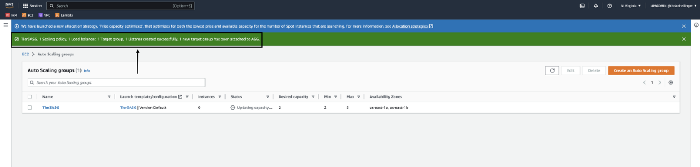
Click the “Create Auto Scaling group” button.



Ok now I know this has been a lot and seems like we just started so come on LET’S KEEP GOING! YOU CAN DO IT!



Once we Create the Auto Scaling group we should see a Success message at the top of the screen.



Now go to the EC2 instance that is running and enter the public IP Address into a browser window and see if your web page shows up.

Graphical user interface

Description automatically generated

Now as you can see I entered the word “Black” incorrectly by accident so I will now edit the user data and modify my script. So I will now go to the instance I boot strapped. I then clicked the action tab to the top right section of the page. After clicking action I then clicked the instance settings tab. Then another drop down box shows “Edit user” data and I then clicked that.

Graphical user interface, application

Description automatically generated

I now spelled everything correctly. So now you know how to edit the user data after the instance has already been created.

Graphical user interface, text, application, email

Description automatically generated

**Step 3: Create a Application-Tier**

Listed below is what the Application-Tier will have

* 2x private subnets
* 2x EC2 instances(We will be picking the same OS as the Web-Tier)
* EC2 Application server security group allowing inbound permission from the Web Server security group

Note: This is not a true application tier as we don’t have any provided code to run on the EC2 instances.

We will basically do some of the same steps as we did with the Web-tier section. So we will go to the Auto Scaling section in the EC2 dashboard and create another Auto scaling group and create another instance. Now we will be doing this for 2x of our private subnets. So we will name the Launch template. I named mine “Tier3Privategroup” This is because it will have the private subnets associated with it. We will be using the same AMI which is Amazon Linux.

Graphical user interface, text, application

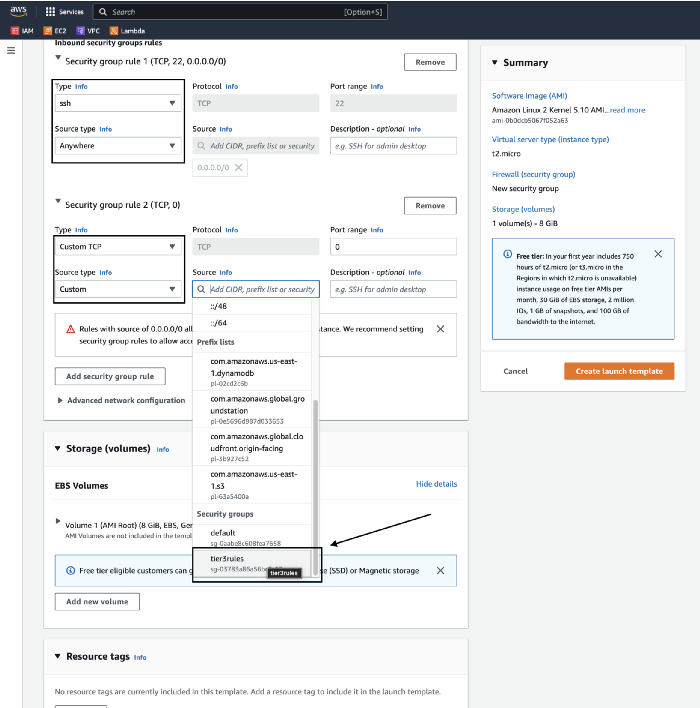
Description automatically generated

Here we are doing the same thing which is using a t2.micro instance. We will create a new key pair for this instance. Again remember where you store the key. We will then create a security group like before. The security name is required along with the description. Make sure our VPC we created for this project is chosen.

Graphical user interface, text, application, email

Description automatically generated

We will be adding an inbound security group rule. We will choose SSH and the source type as anywhere. For the next security group rule we will be adding our Web-Tier SG as well. This will be in your drop down menu so make sure to click that specific SG

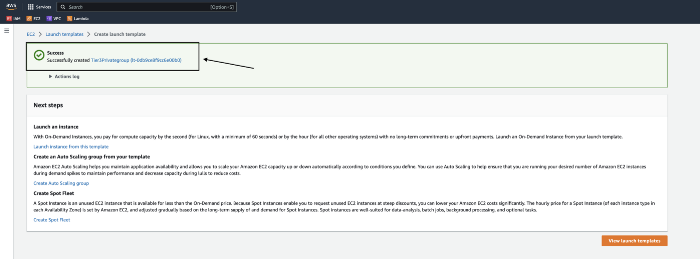


After everything is checked and picked correctly create the launch template.

Graphical user interface, application

Description automatically generated

We see a Success message that we created the launch template.



Here we will see the launch template we just created. We can also now see the SG we have chosen for this launch template. So we will continue on for the next steps.

Graphical user interface

Description automatically generated

Now for the network options of this Auto Scaling group we will choose 2x private subnets. After we choose the 2x subnets we will click the “Next” button.

Graphical user interface, text, application

Description automatically generated

We will create another new load balancer for this launch template. We don’t technically need it but I will be creating one. As before make sure the correct private subnets are selected.

Graphical user interface, application, Word

Description automatically generated

We will create a new target group name as well. After this is completed click the “Next” button.

Graphical user interface, text, application

Description automatically generated

Here wi will be choosing the same options as the Web-Tier Group size auto scaling section.

Graphical user interface, application

Description automatically generated

After we create the Application-Tier Auto Scaling group we should now see a total of 2x Auto Scaling groups.

Graphical user interface, application

Description automatically generated

Now let’s go to the EC2 Instance and we should see a total of 4x instances running. This is 2x from the Web-Tier and 2x from the Application-Tier.

Graphical user interface, table

Description automatically generated

**Step 4: Create a Database-Tier**

Here is a list of what the Database-tier will have

* Free-Tier MySQL RDS Database
* Database security group allowing inbound traffic for MySQL from the Application server security group
* 2x Private subnets

We will be going to the search bar in the AWS console and enter RDS. This is the database we will be creating. Now when we are in the RDS dashboard we will be creating a new subnet group. So click Subnet groups on the left pane of the RDS Dashboard.

Graphical user interface, text, application

Description automatically generated

We will be naming our Subnet group. We will put a description in as well. In the Details section we want to make sure we have our VPC picked. We should then add the Availability zones. I have picked 2x different AZ’s to keep it highly available. Now I have picked the last 2x private subnets that haven’t been allocated yet. Once all of the above steps have been taken click the orange “Create” button.

Graphical user interface, application

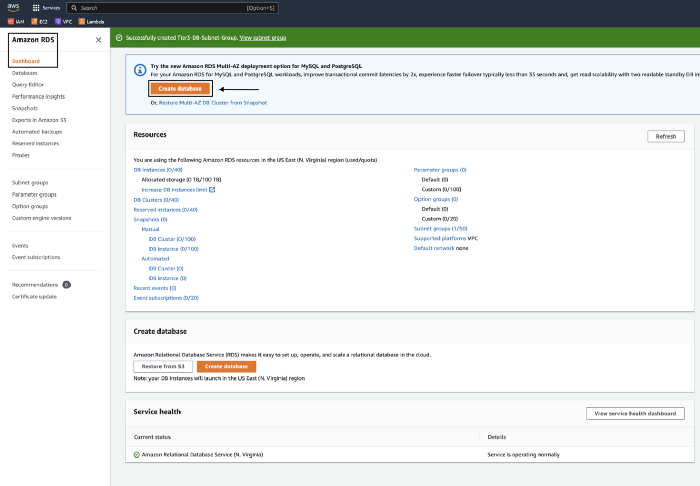
Description automatically generated

We will now see the Subnet group we just created.

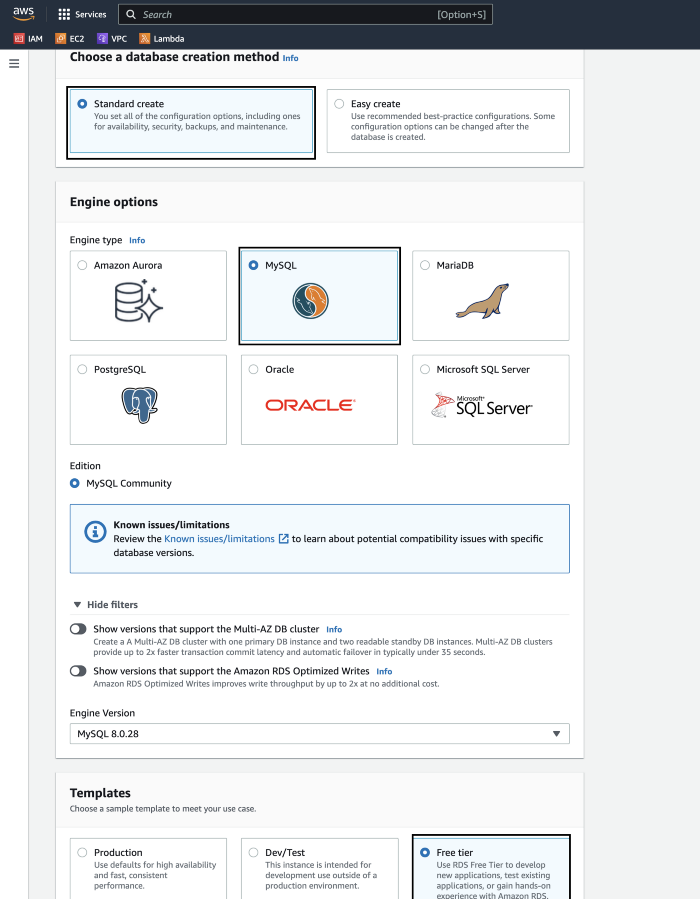
Graphical user interface, application

Description automatically generated

We will now create the Database. Once we are back in the RDS dashboard we will see the orange “Create database” button. So click that which will begin the process.



We will choose the Standard Create option. We will then choose the MySQL Engine option. Now at the bottom of my screen shot you will see the Templates section. We will choose the Free-Tier. If you choose any other option it will charge you.

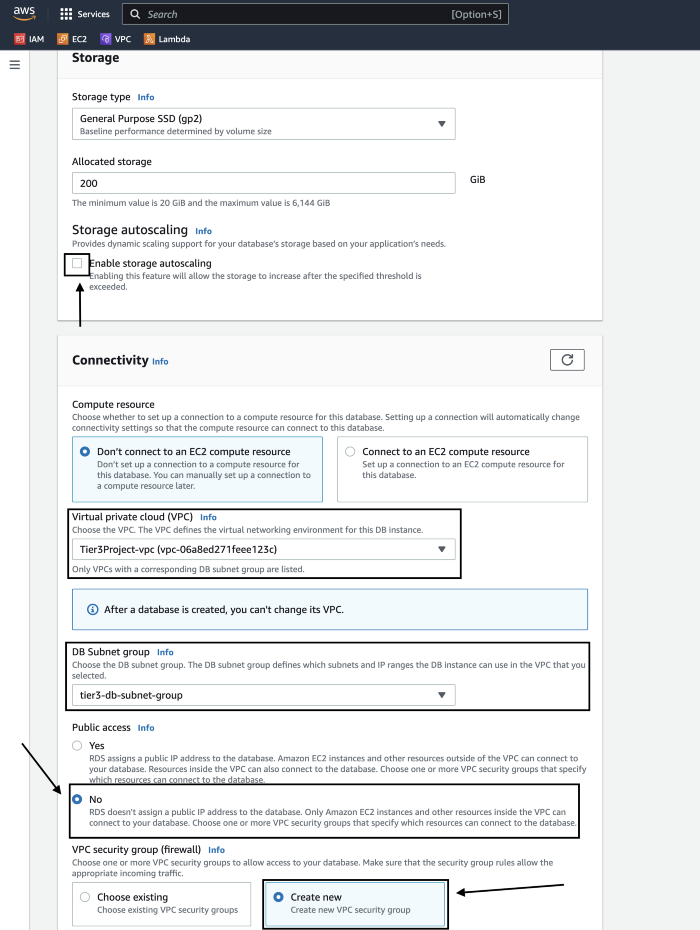


Once we have completed the above section we will now create a password for the credentials setting section. We will then make sure we find the t2.micro instance which I had to click the options and find it.

Graphical user interface, application

Description automatically generated

Below the storage section is very important to pay attention too. We will make sure the box under storage autoscaling is unchecked. This is for Enabling storage autoscaling. Now under the Connectivity section we will Choosing the DB Subnet Group. We will also be choosing “No” for Public access and this is because this Database will be private.



Next we will Create a new VPC security group for this database. I have named my security group and you can choose whatever name you would like. I have also picked the AZ us-east-1a. After you have done all of this you will then go to the bottom of the page and click “Create Database”.

Graphical user interface, text, application, email

Description automatically generated

You will see a Success message if you have created it.

Graphical user interface

Description automatically generated

We will then create the Inbound security rules for the database security group. This will allow inbound access from the application-tier we created earlier. So we will make sure we are on the security group and then click the actions tab on the top right of the page. We will then see a drop down box that says “Edit inbound rules”.

Graphical user interface, text, application, email

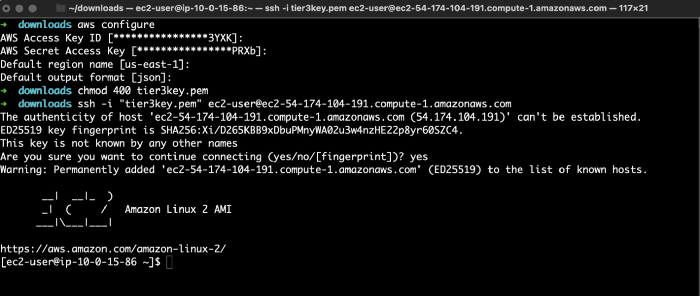
Description automatically generated

Once we are in the Edit inbound rules we will create a customer rules with the SG we created from the application-tier. Now I have having issues with it modifying the existing rules so I had to add the new Inbound rule and then delete the old. Once that is completed make sure to click the save button.

Graphical user interface, application

Description automatically generated

Since I’m using a MAC I will have to allow permissions to my keys. Once I allow permissions I can enter my EC2 Instance.



Here we will Ping the Private IP Address. You will see that it was successful.

Text

Description automatically generated

So this is how you create a highly available 3-tier architecture. Even though this was a long process and a lot of information I would tear everything down which you should do anyways because if not you could be charged. I actually performed this multiple times before writing this article so It does take sometime.

The harder the process the more we learn. I thank everyone for reading this article and learning something new. Please give this article a clap or give me a follow. Thanks!