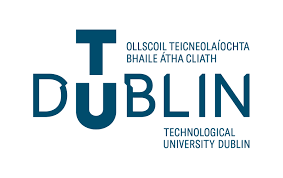
**Cloud Computing CMPU3007: 2020 – 21**

**Assignment 2**

****

Submission Date: 20/12/2020

Lecturer Name: Dr Basel Magableh

**Student**

Philip Herweling

C18470774

C18470774@mytudublin.ie

TU856 Year 3

Contents

[Introduction: 3](#_Toc59362100)

[Three Tier Web application: 4](#_Toc59362101)

[1. A VPC named (project\_vpc) in the (N. Virginia) 5](#_Toc59362102)

[VPC: 5](#_Toc59362103)

[2. The web and app tiers are hosted in three availability zones (Create private subnets for each tier is recommended) 5](#_Toc59362104)

[Subnets: 5](#_Toc59362105)

[Route Tables: 6](#_Toc59362106)

[3. Host the DB in Multi AZ DB in private subnets 7](#_Toc59362107)

[Database: 7](#_Toc59362108)

[4. Use the following script for the web tier with Amazon Linux 2 8](#_Toc59362109)

[EC2 Instances: 8](#_Toc59362110)

[5. Use Amazon Linux 2 for the app tier with the following script 9](#_Toc59362111)

[6. Make Sure that the web and app tier are scalable and available 10](#_Toc59362112)

[Images: 10](#_Toc59362113)

[Snap Shots: 10](#_Toc59362114)

[Elastic load balancers 10](#_Toc59362115)

[Launching configurations: 11](#_Toc59362116)

[Auto Scaling Groups: 11](#_Toc59362117)

[EC2 Instances: 12](#_Toc59362118)

[7. Required Screenshots: 13](#_Toc59362119)

[Subnets: 13](#_Toc59362120)

[Load Balancers: 13](#_Toc59362121)

[Autoscaling Groups: 14](#_Toc59362122)

[Final Step Showing Access: 14](#_Toc59362123)

# **Introduction:**

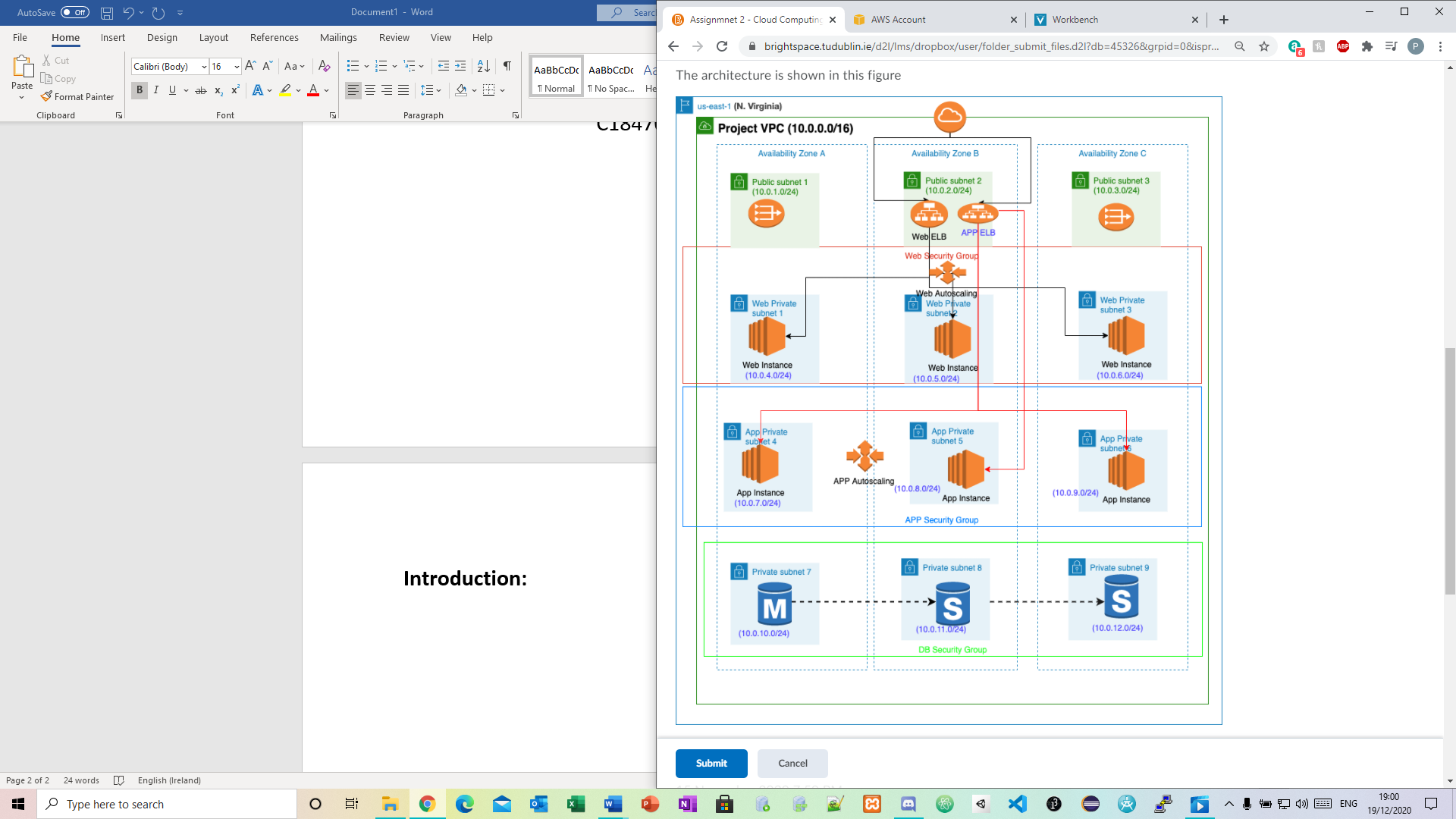
For this assignment I had to design a three-tier web application to be hosted in AWS cloud. The architecture as you can see in the figure below has a multi-AZ web server. For the assignment I started off by creating a vpc which I named project\_vpc. The region I used was N. Virginia as specified in the brief. After making the vpc I made the 12 subnets needed I.e. the three public subnets in the three availability zones us-east-1a, 1b and 1c and the nine private subnets (three private web, database and app subnets each). These were also in the availability zones us-east-1a, 1b and 1c. After creating the subnets, I made two route table called ‘Project public route table’ and ‘Project private route table’. The public route table has three subnet associations which are public subnet 1, 2 and 3. And the private route table has 9 subnet associations which are all nine private subnets.

The next thing I done was create a database subnet group called project\_dp\_subnet\_group. This subnet group stored all three database subnets which I created earlier. After making the database subnet group I made my database. I used MySQL to make the database.

After creating my project database, I made two EC2 instance called web tier and app server I used these when I was creating my elastic load balancers. After that then I made to AMI’s one for the web tier and one for the app tier. I then needed to create two load balancers one for the app tier and one for the web tier. I called these two load balancers WebELB and AppELB. After creating the two elb’s I needed to make two autoscaling groups but before I could do that, I needed to make two launch configurations first, one for the web auto scaling group and one for the app autoscaling group. After making these two launch configurations I made the two autoscaling groups.

The final step for me was to test the load balancers and make sure the connection worked. Gladly it did work, and you can find the screenshots in the section titled Required Screenshots.

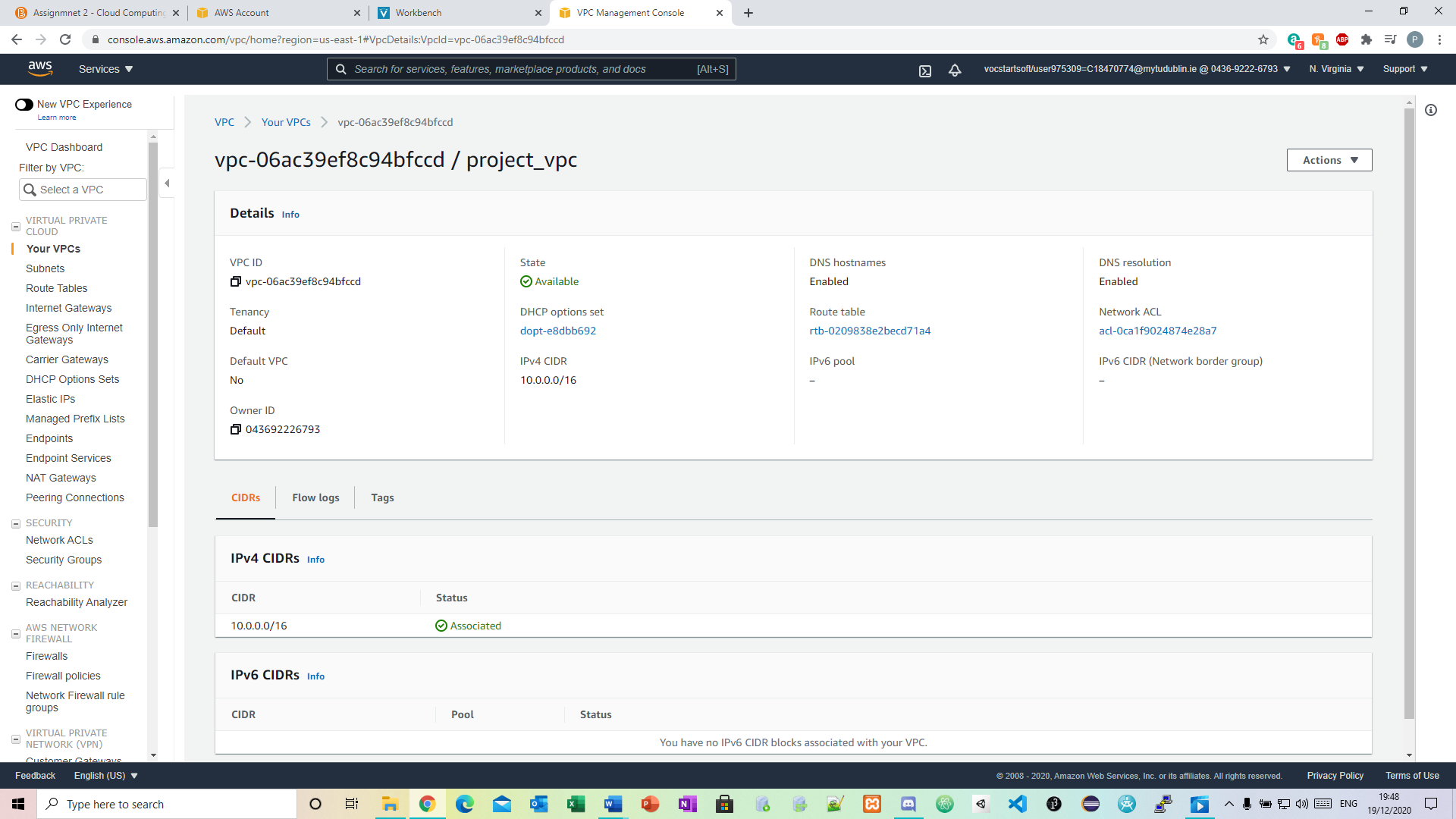
# **Three Tier Web application:**



# **A VPC named (project\_vpc) in the (N. Virginia)**

## VPC:

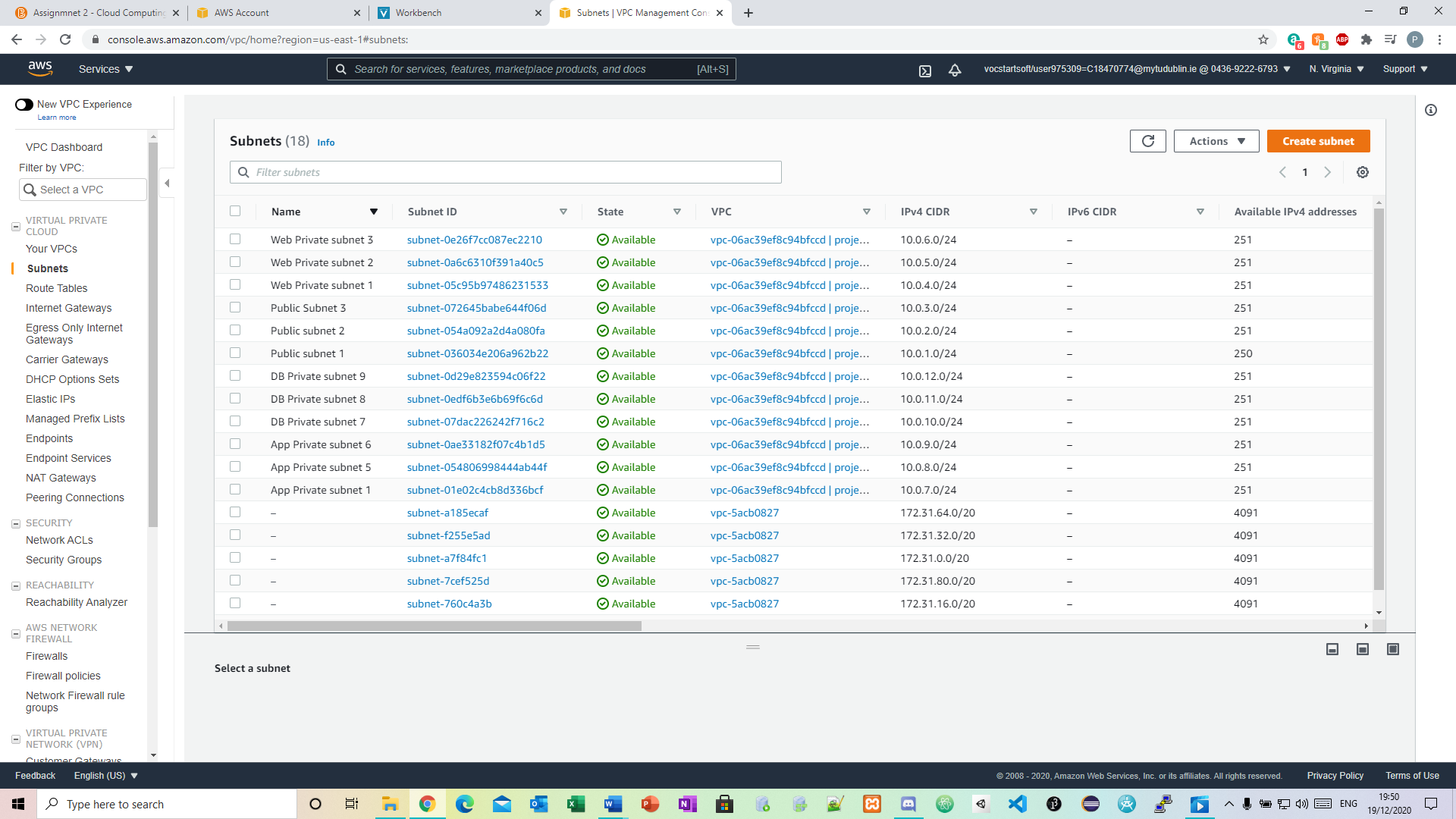
Figure 1: project\_vpc



# **The web and app tiers are hosted in three availability zones (Create private subnets for each tier is recommended)**

## Subnets:

Figure 2: 12 subnets created



## Route Tables:

Figure 3: Project Private route table

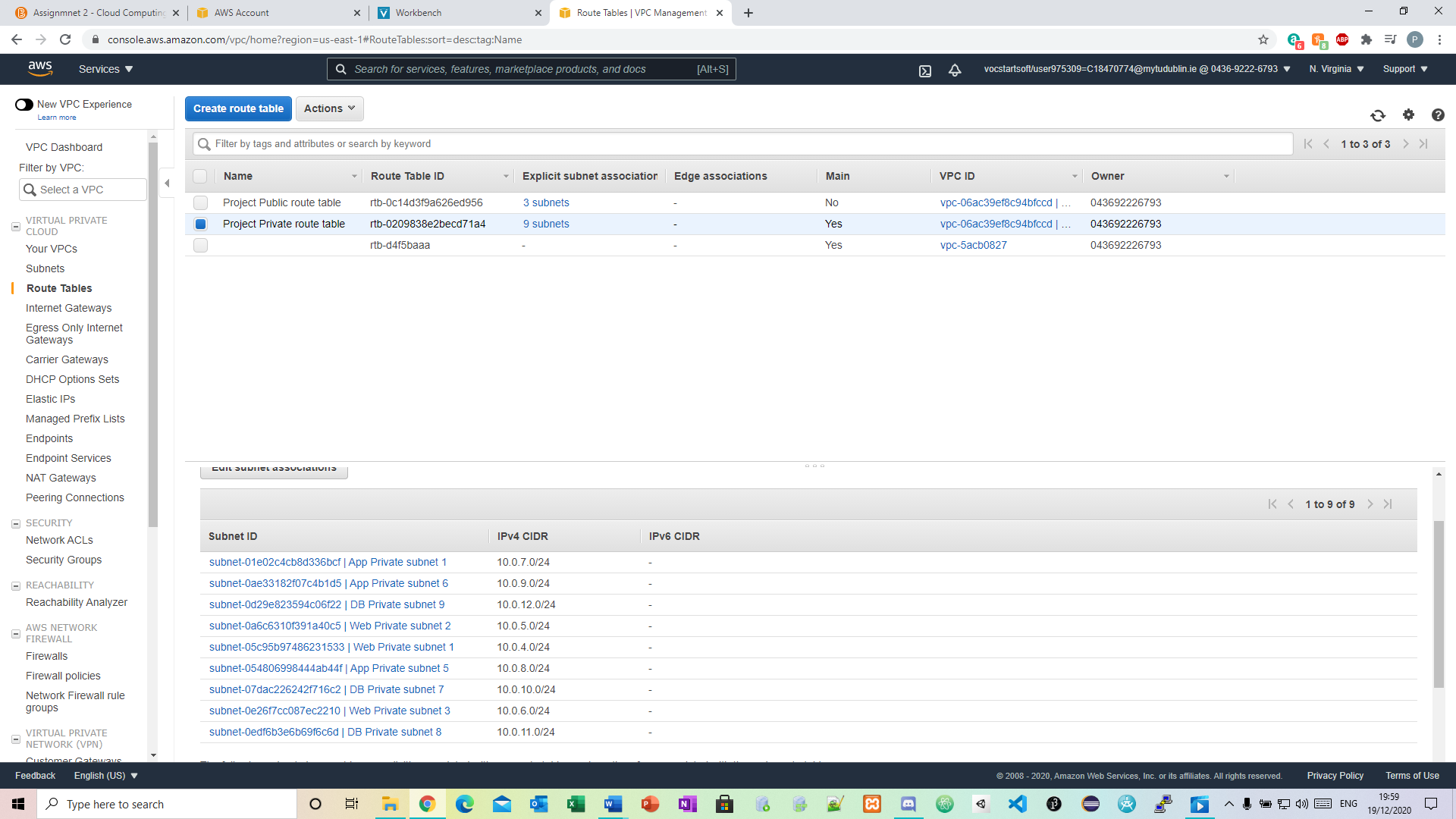
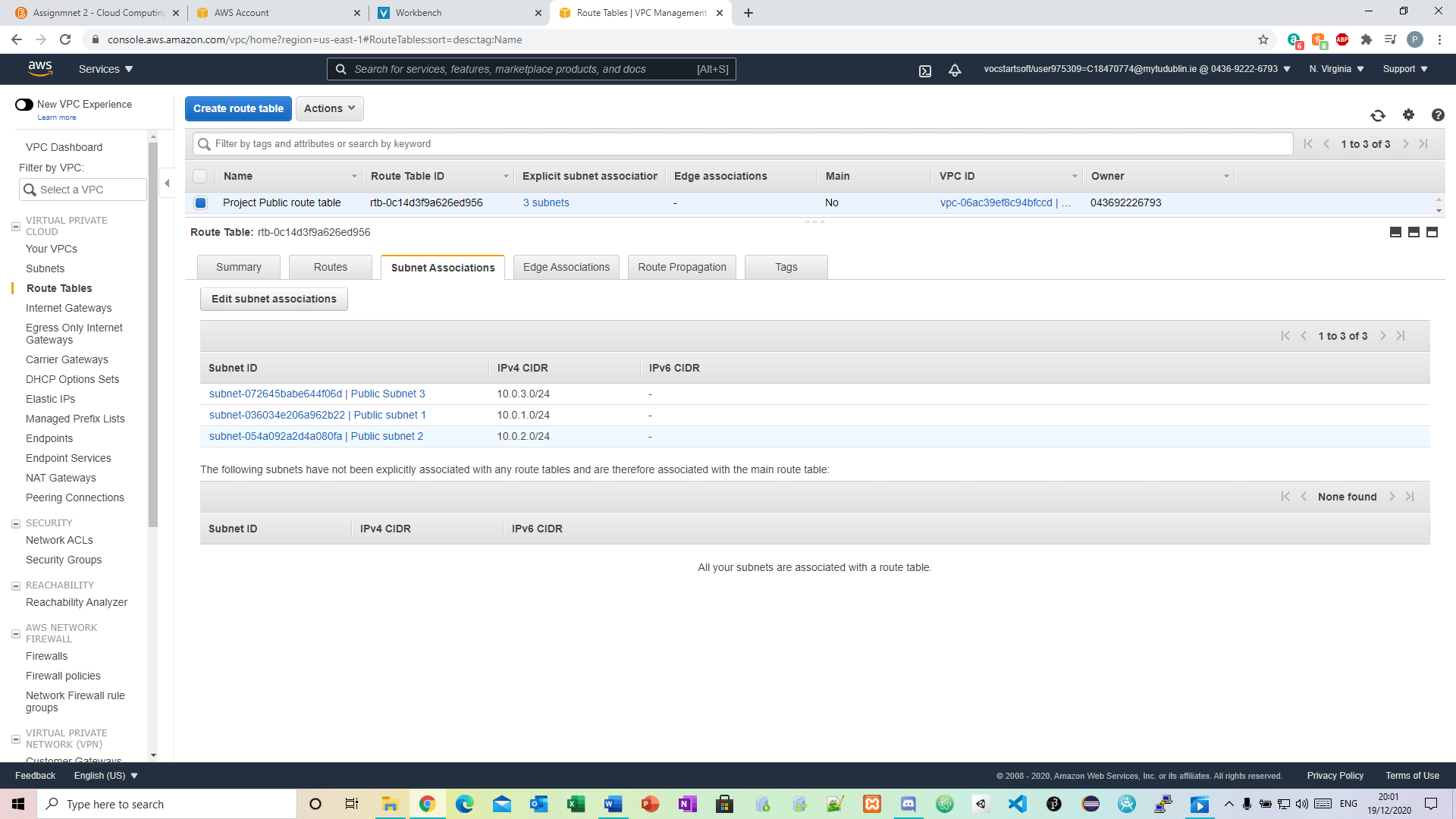


Figure 4: Project Public route table



# **Host the DB in Multi AZ DB in private subnets**

## Database:

Figure 5: Database subnet group

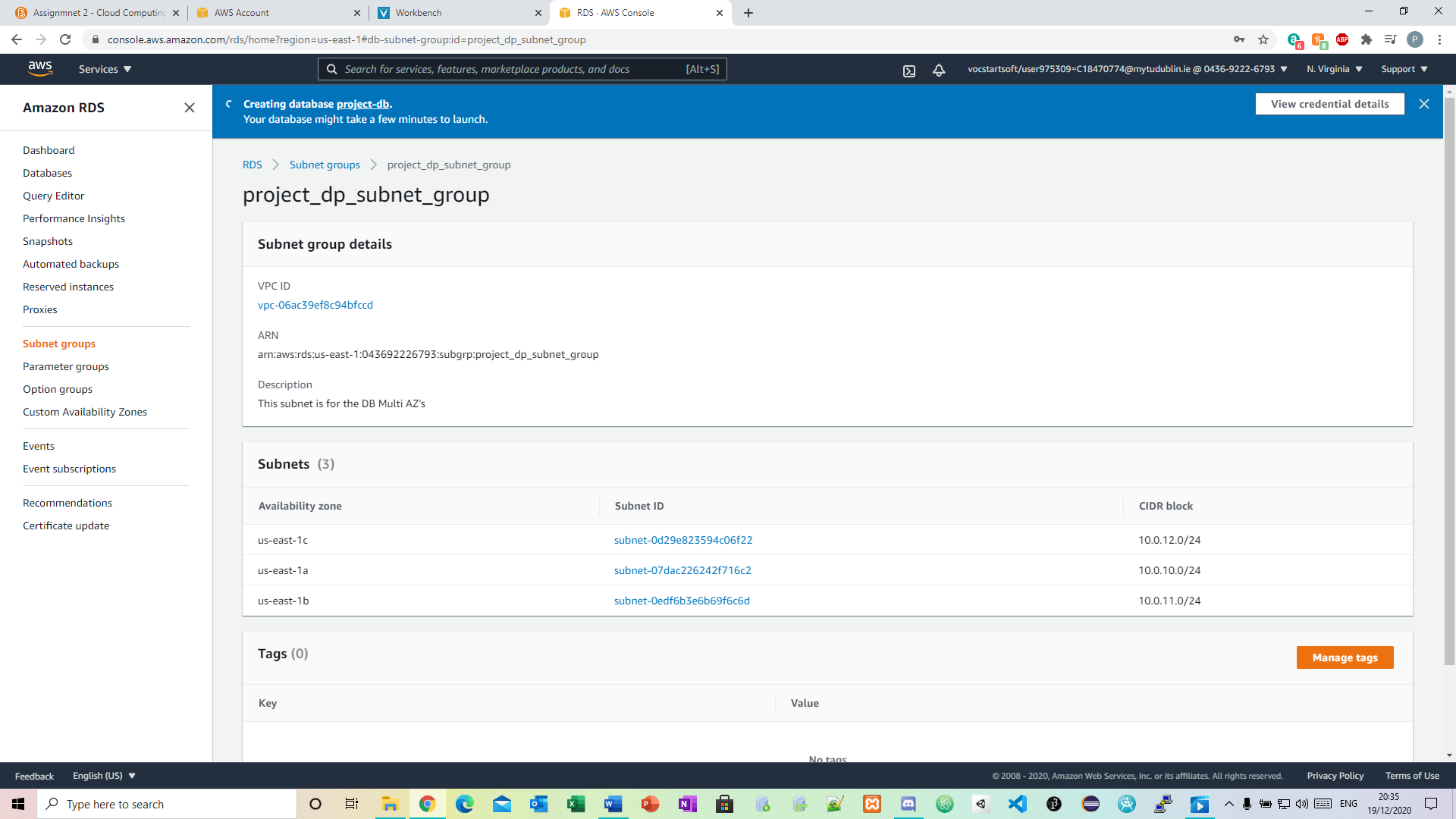
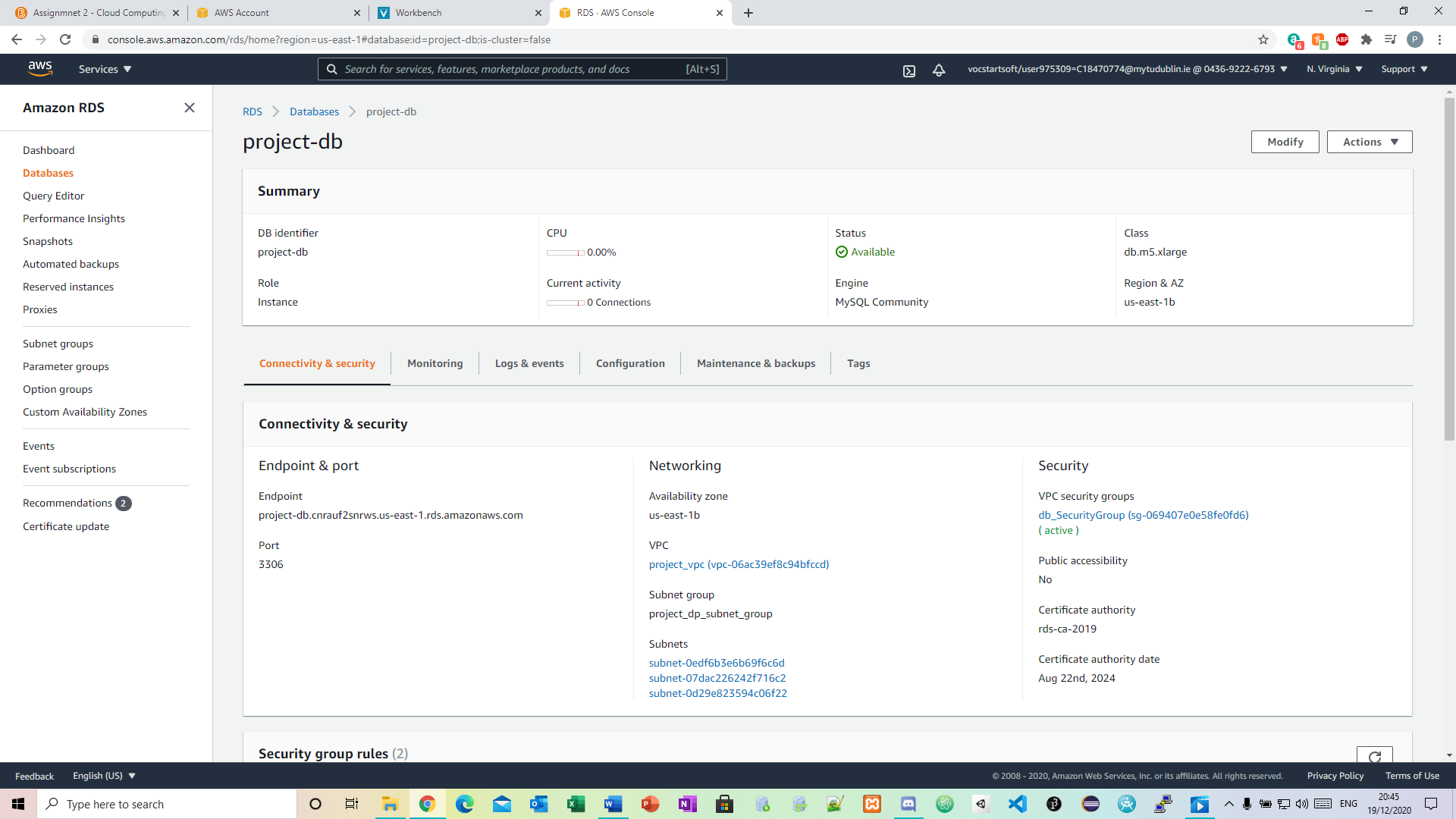


Figure 6: Database



# **Use the following script for the web tier with Amazon Linux 2**

## EC2 Instances:

Figure 7.1: Web Tier Instance

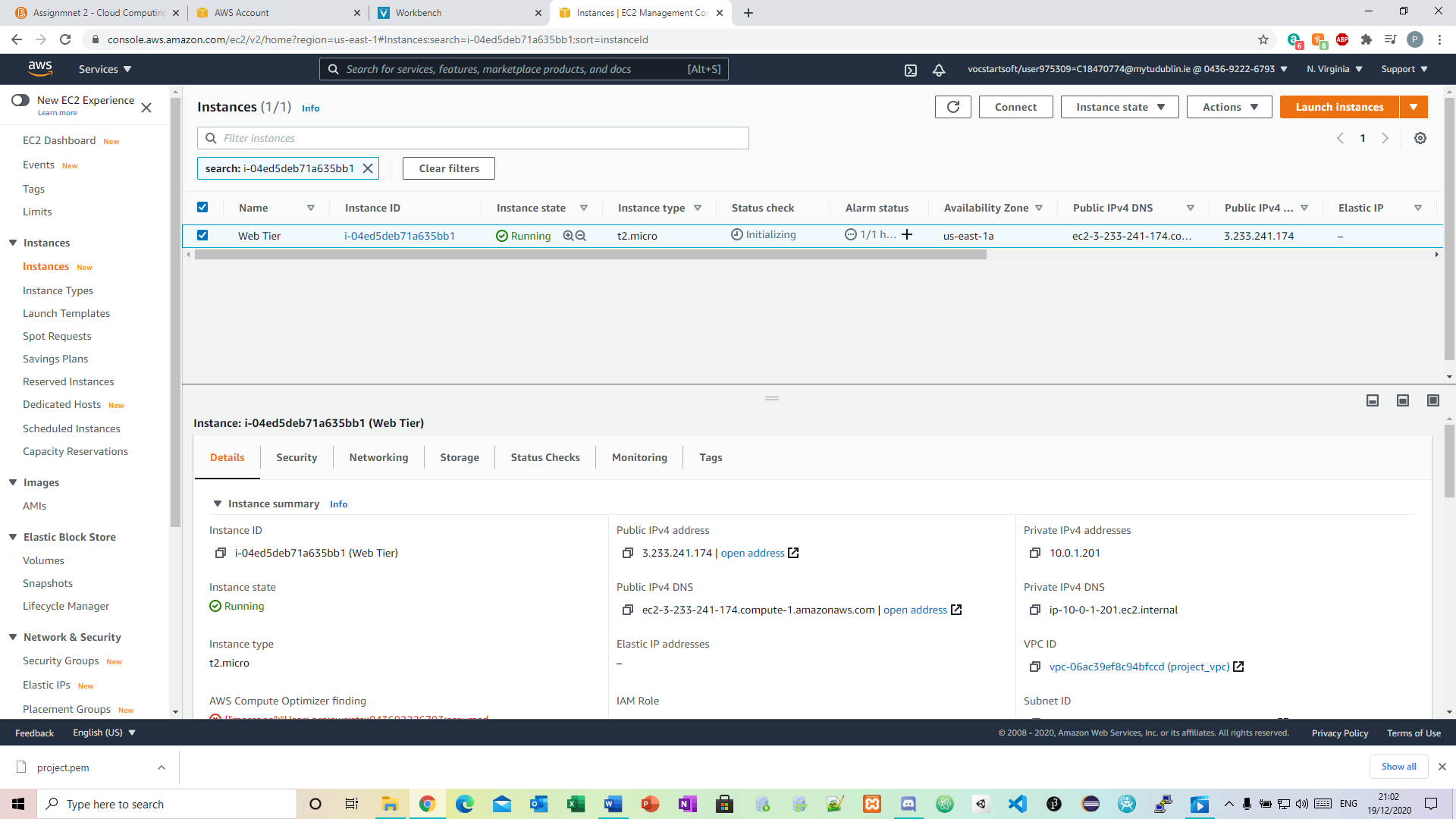
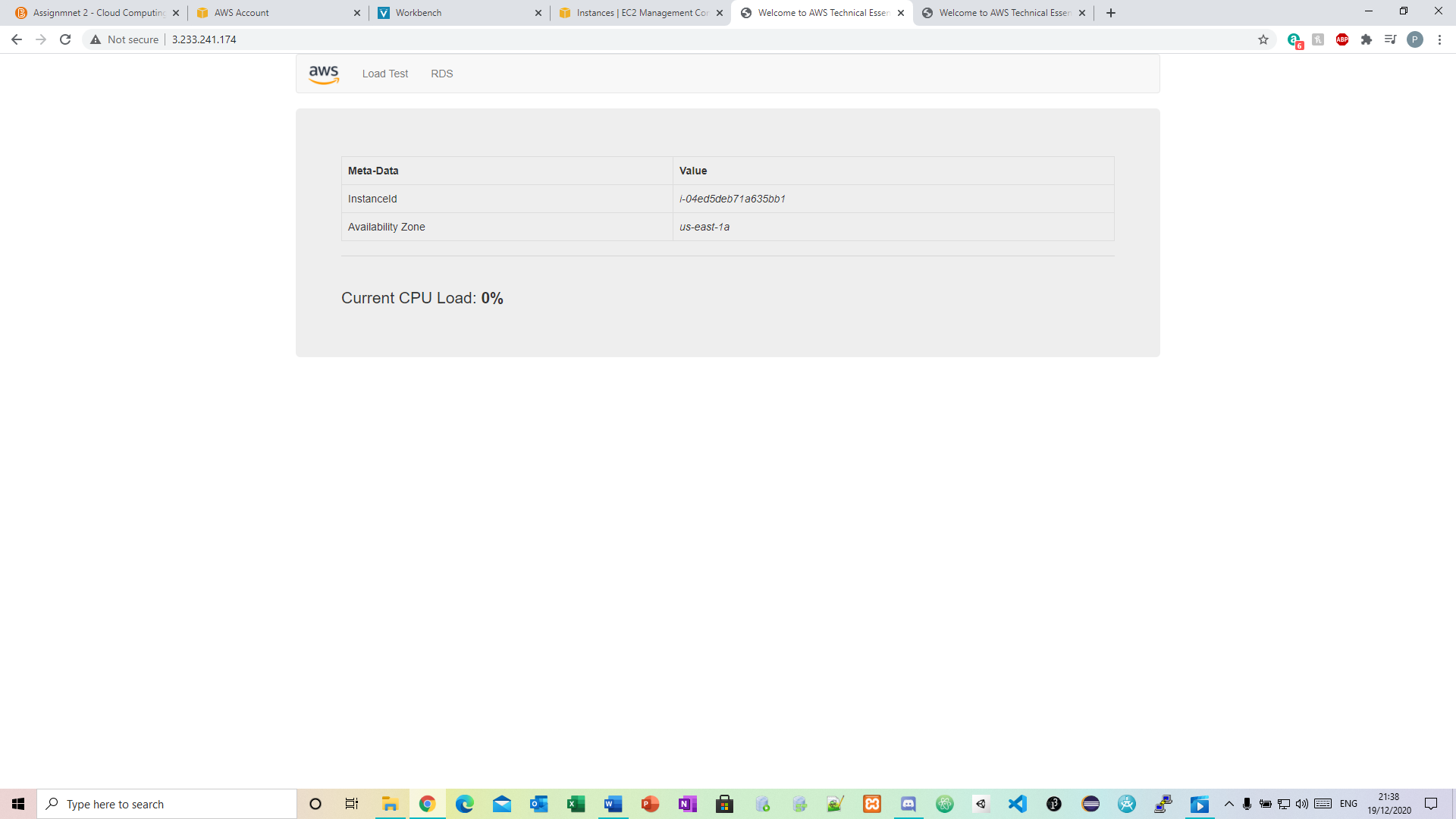


Figure 7.2:



# **Use Amazon Linux 2 for the app tier with the following script**

Figure 8.1: App Server Instance

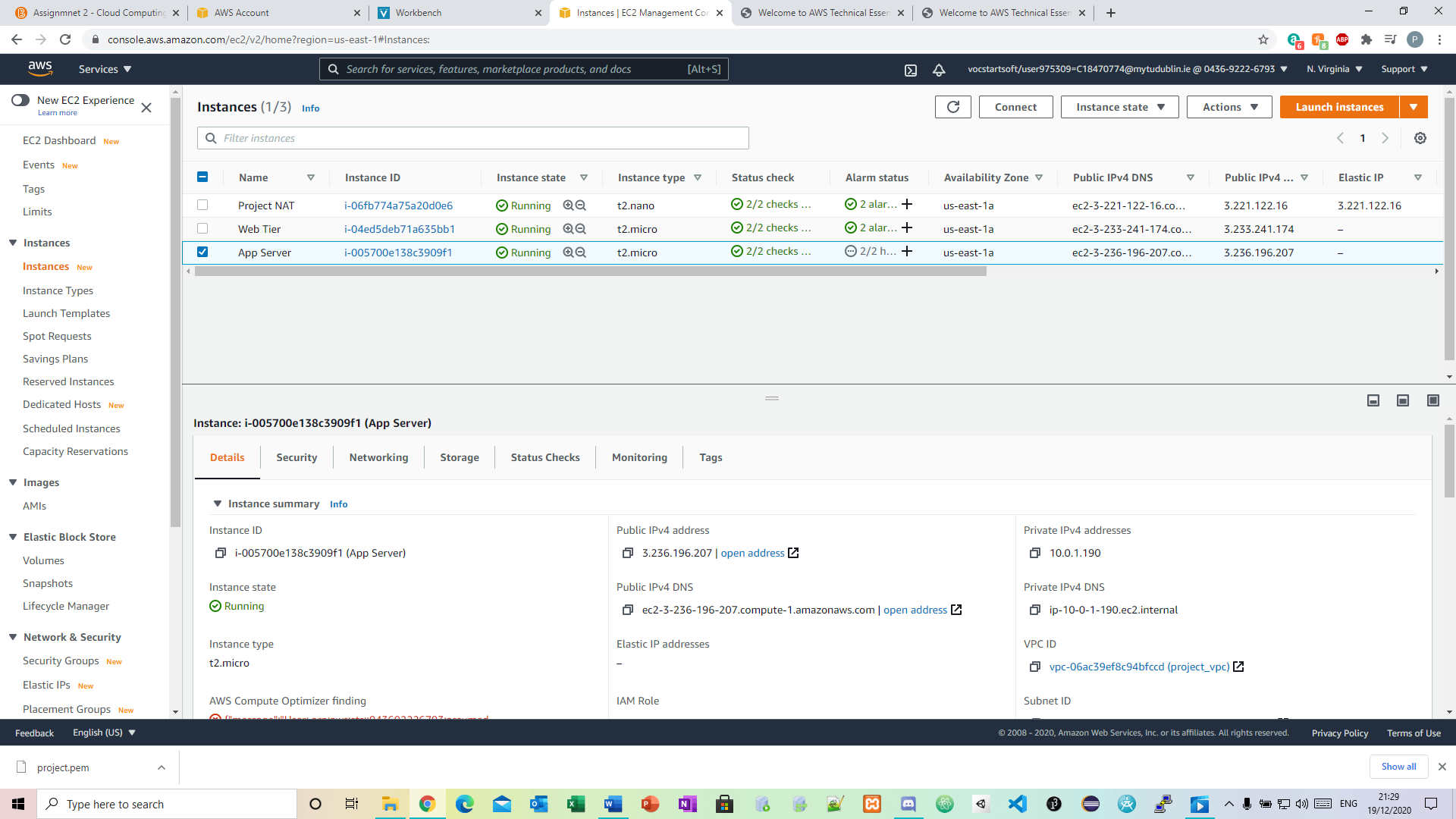


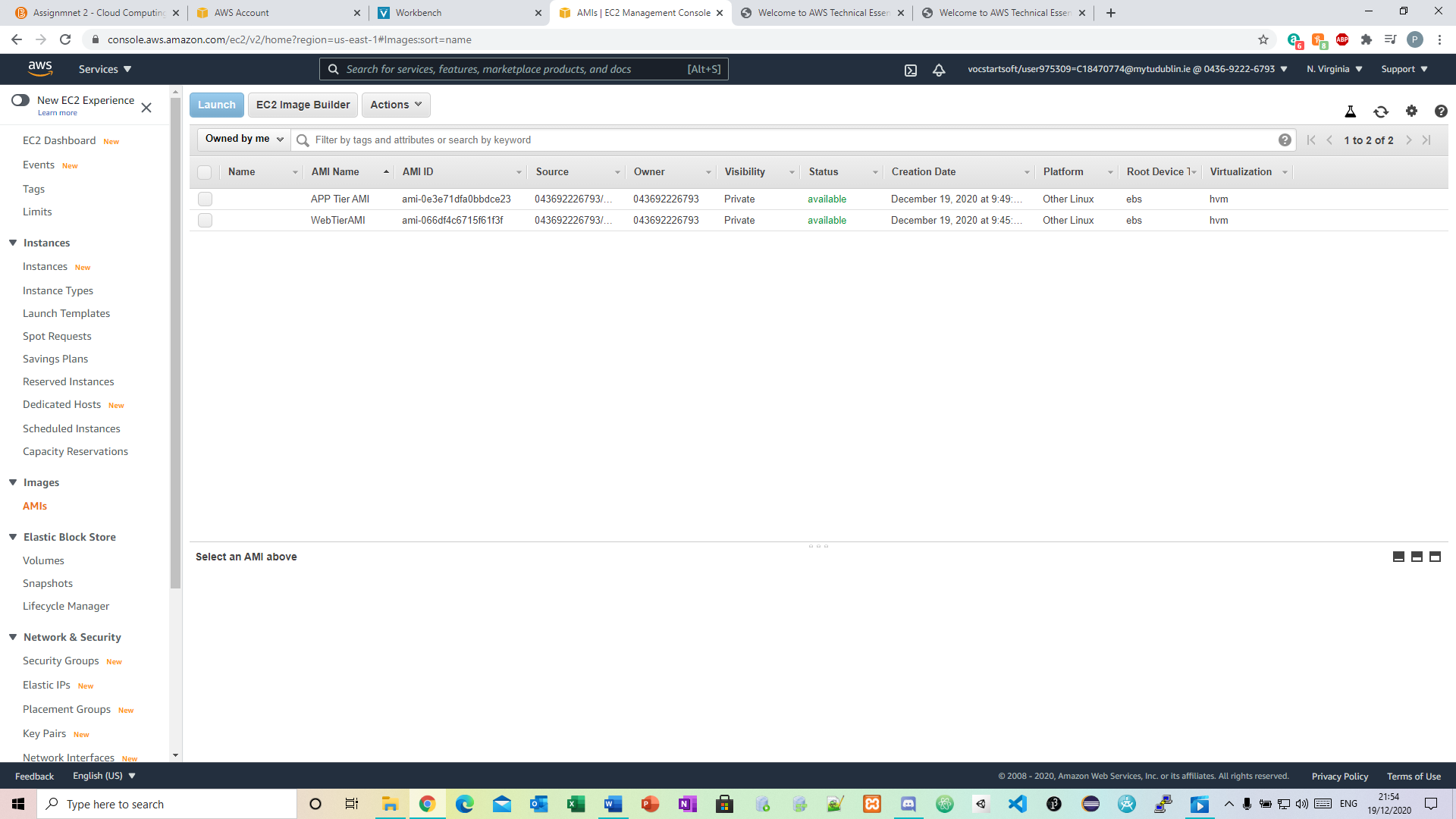
Figure 8.2:



# **Make Sure that the web and app tier are scalable and available**

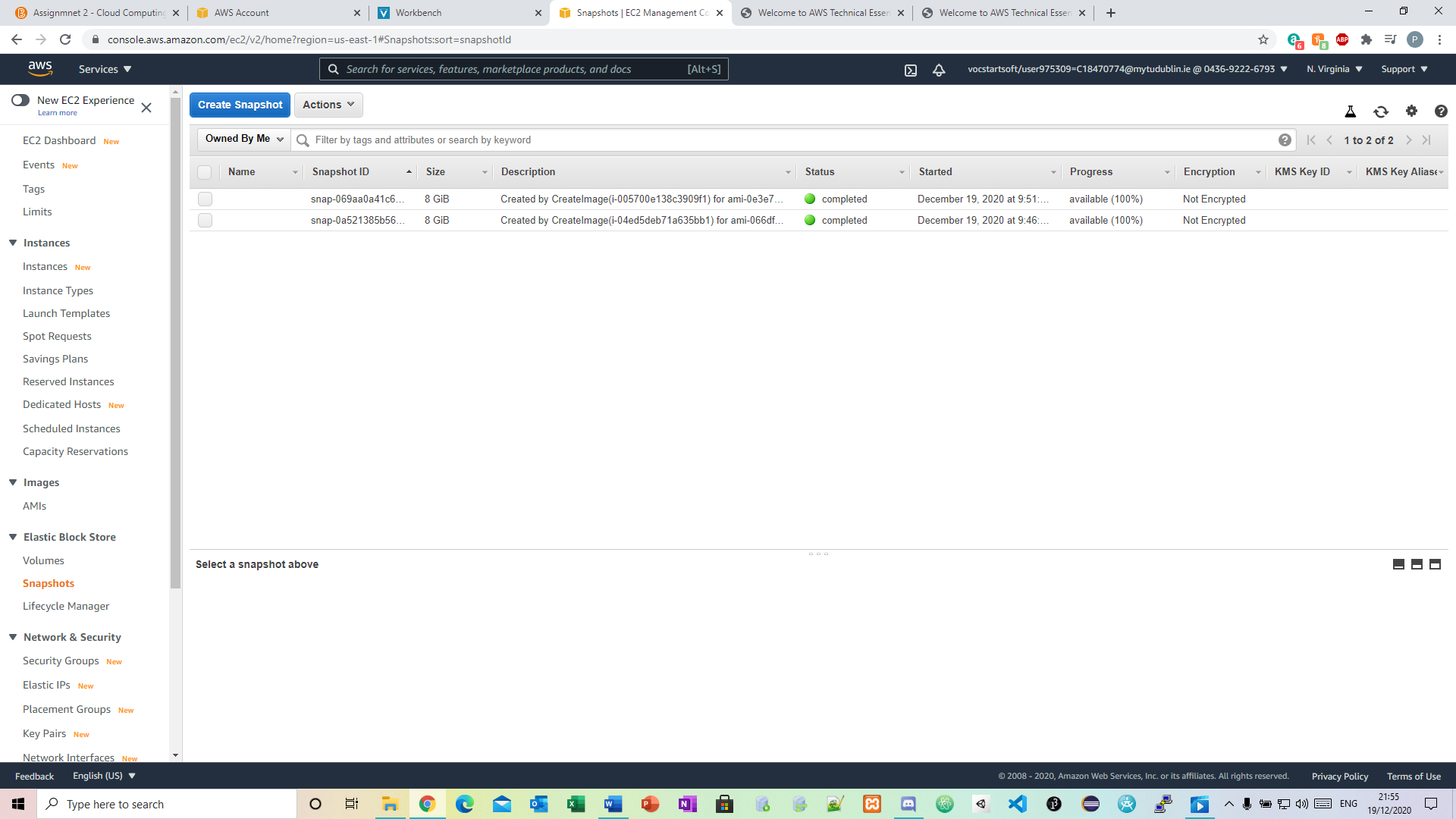
## Images:

Figure 9: Web Tier AMI and App Tier AMI



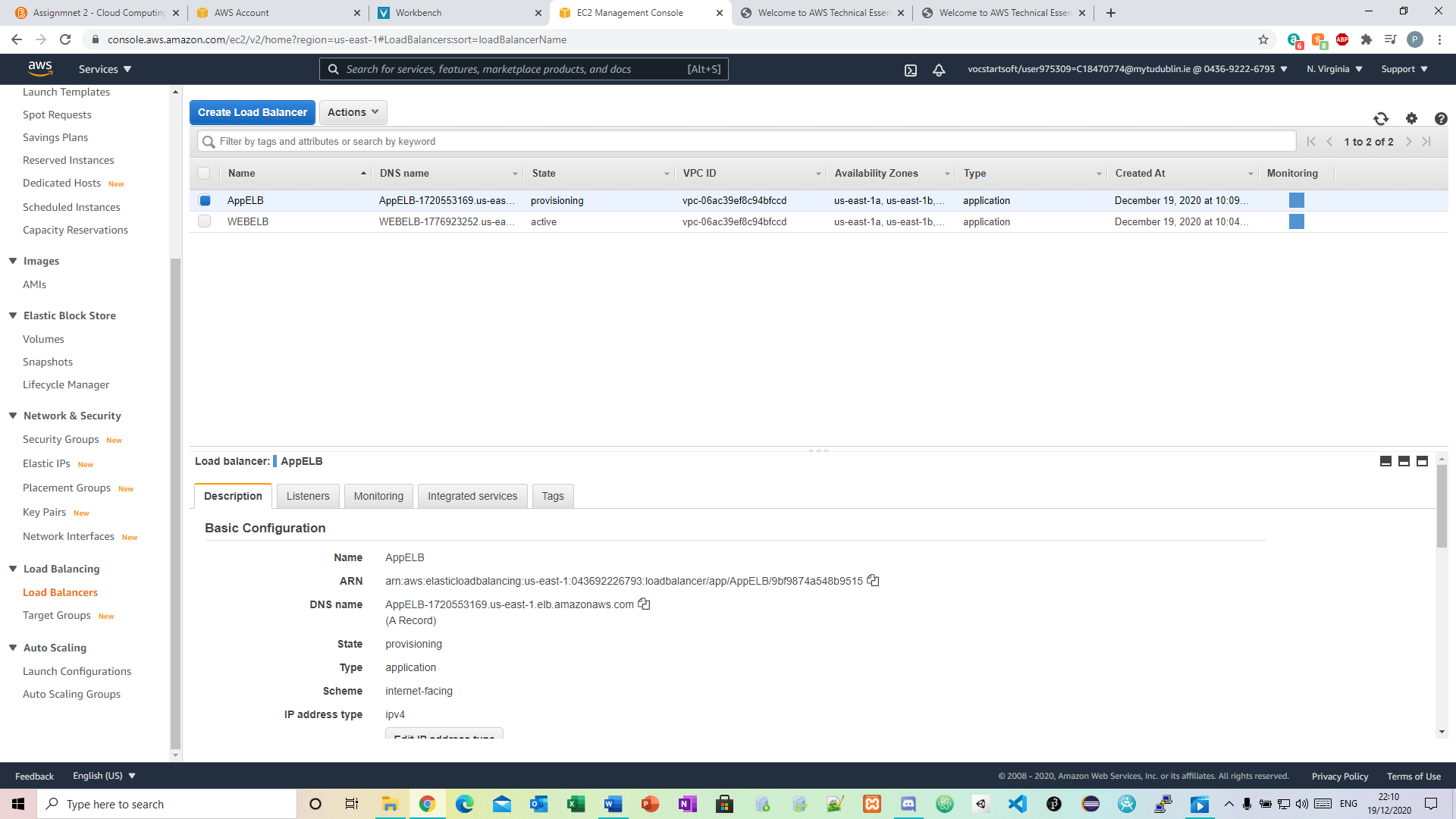
## Snap Shots:

Figure 10: snapshots



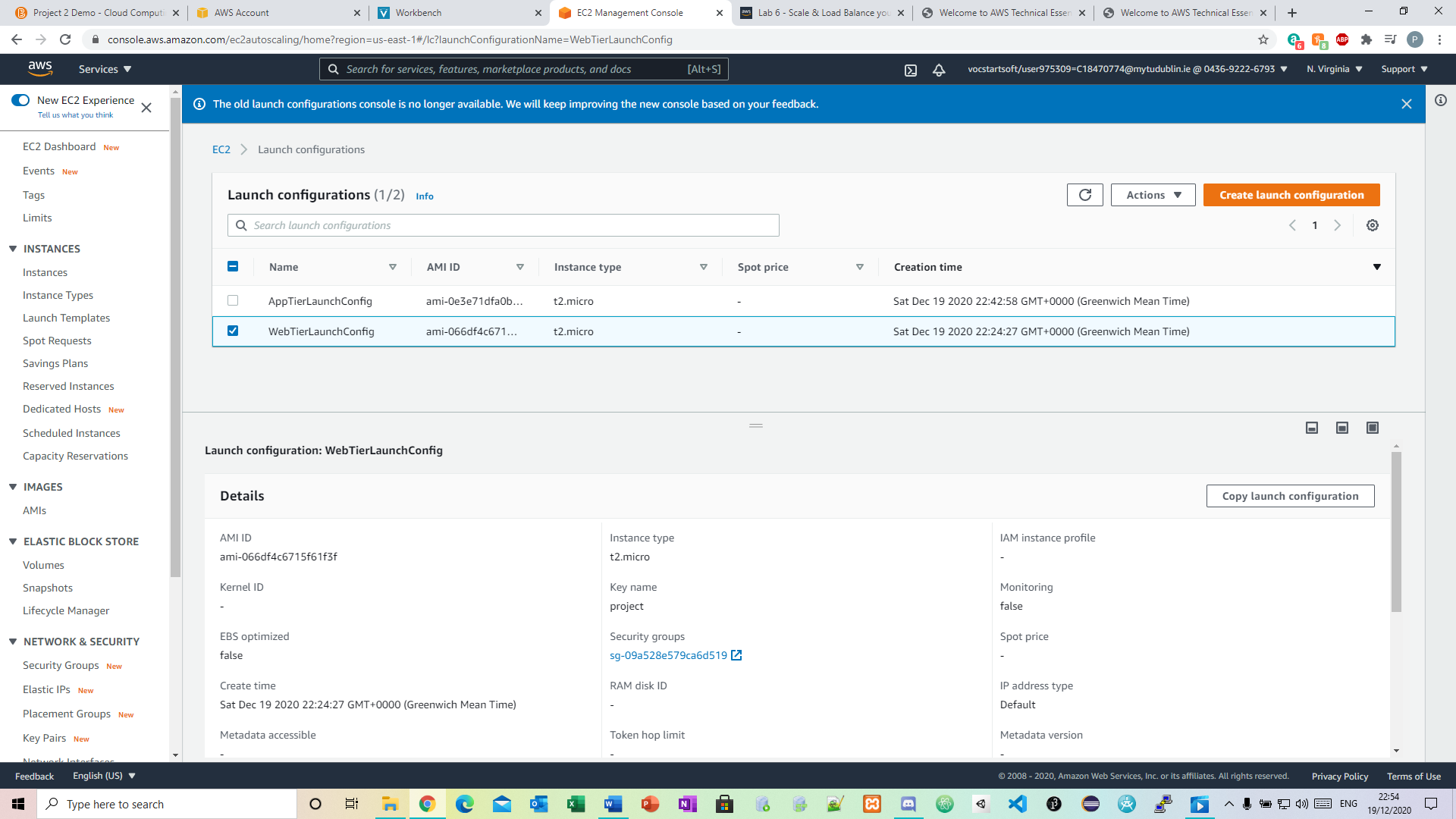
## Elastic load balancers

Figure 11: Web and App ELB’s



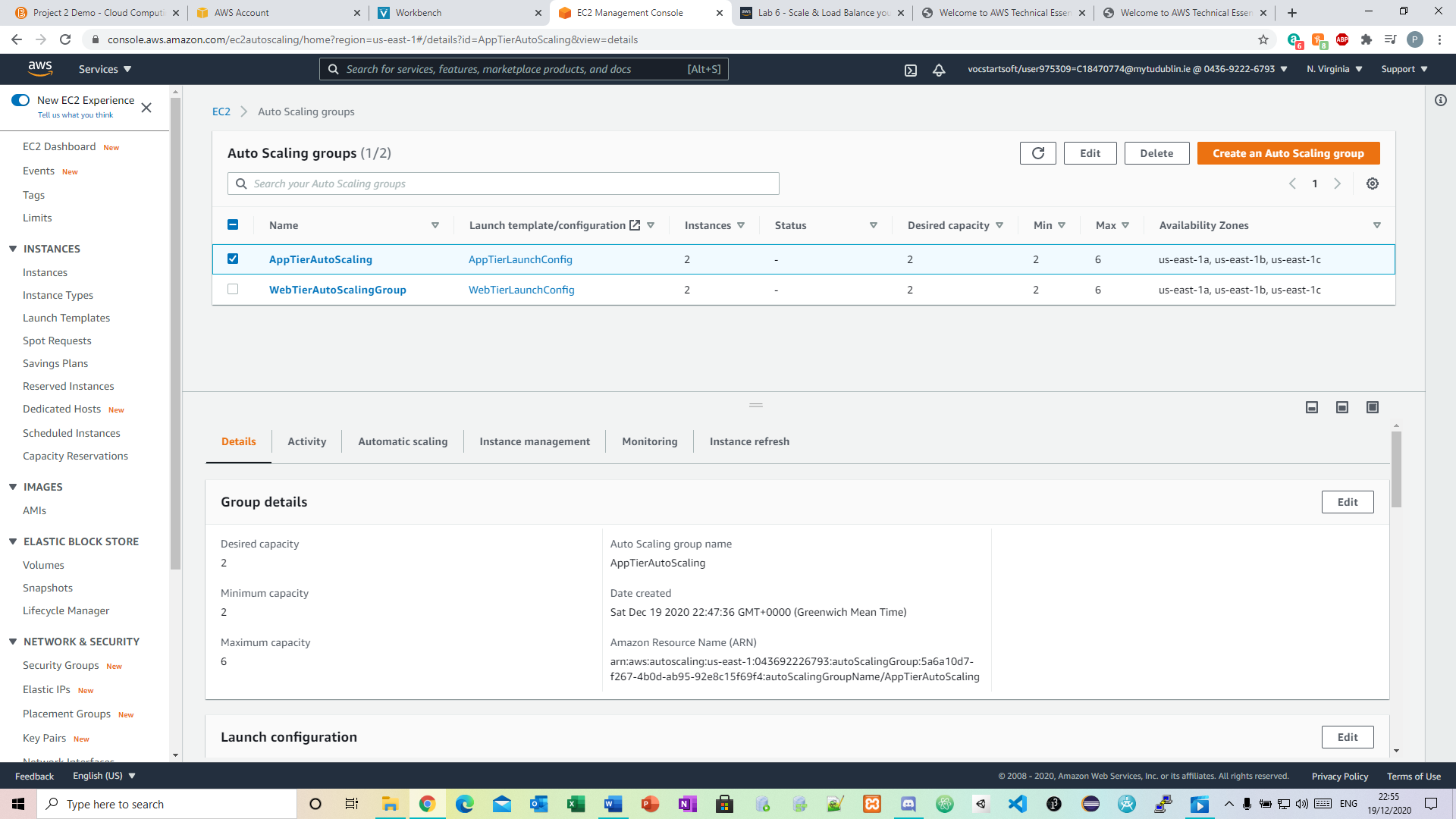
## Launching configurations:

Figure 12: App Tier launch configuration and Web Tier launch configuration



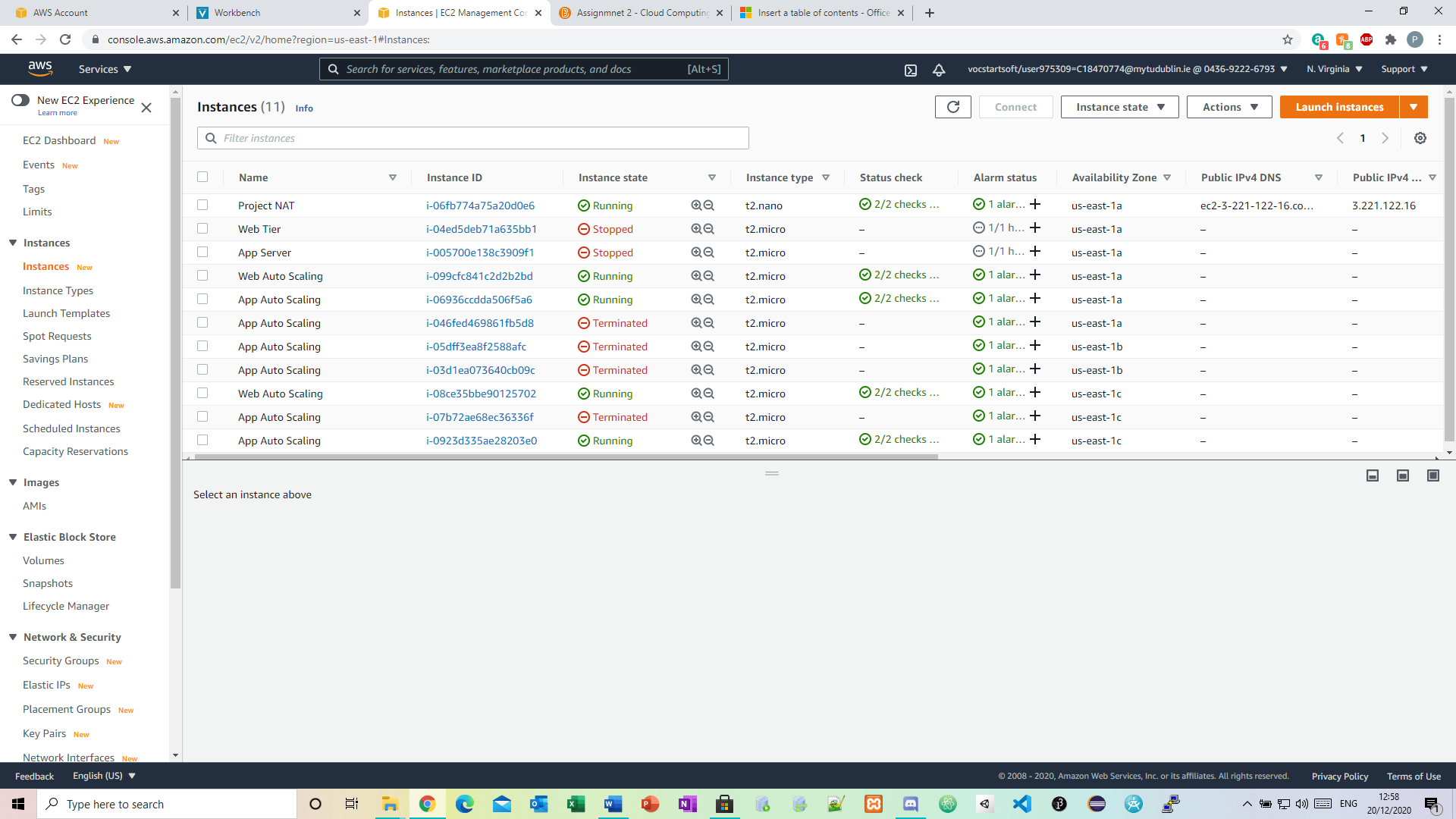
## Auto Scaling Groups:

Figure 13: App Tier Auto Scaling and Web Tier Auto Scaling Groups



## EC2 Instances:

Figure 14: Instances after load test was done

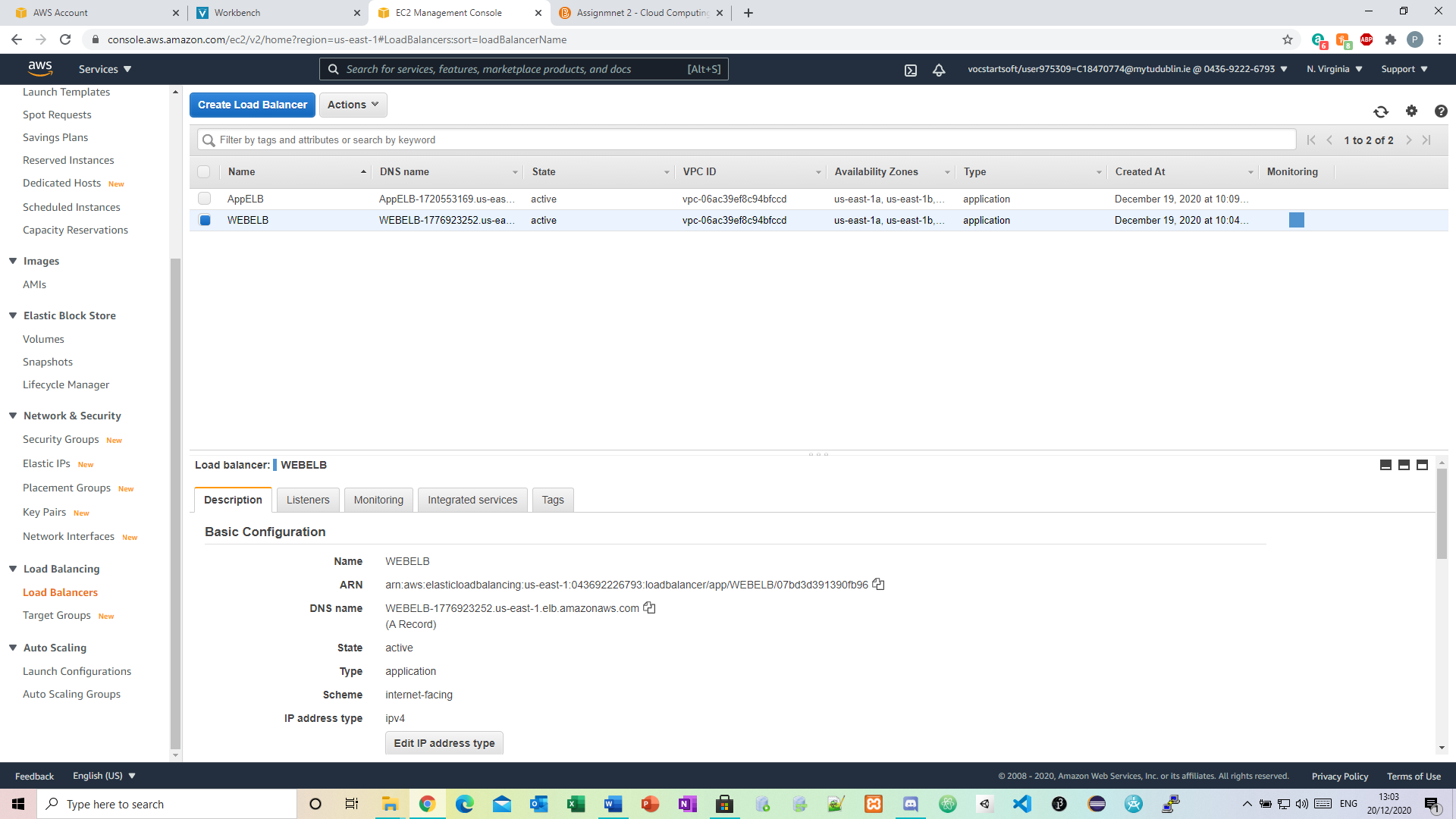


# **Required Screenshots:**

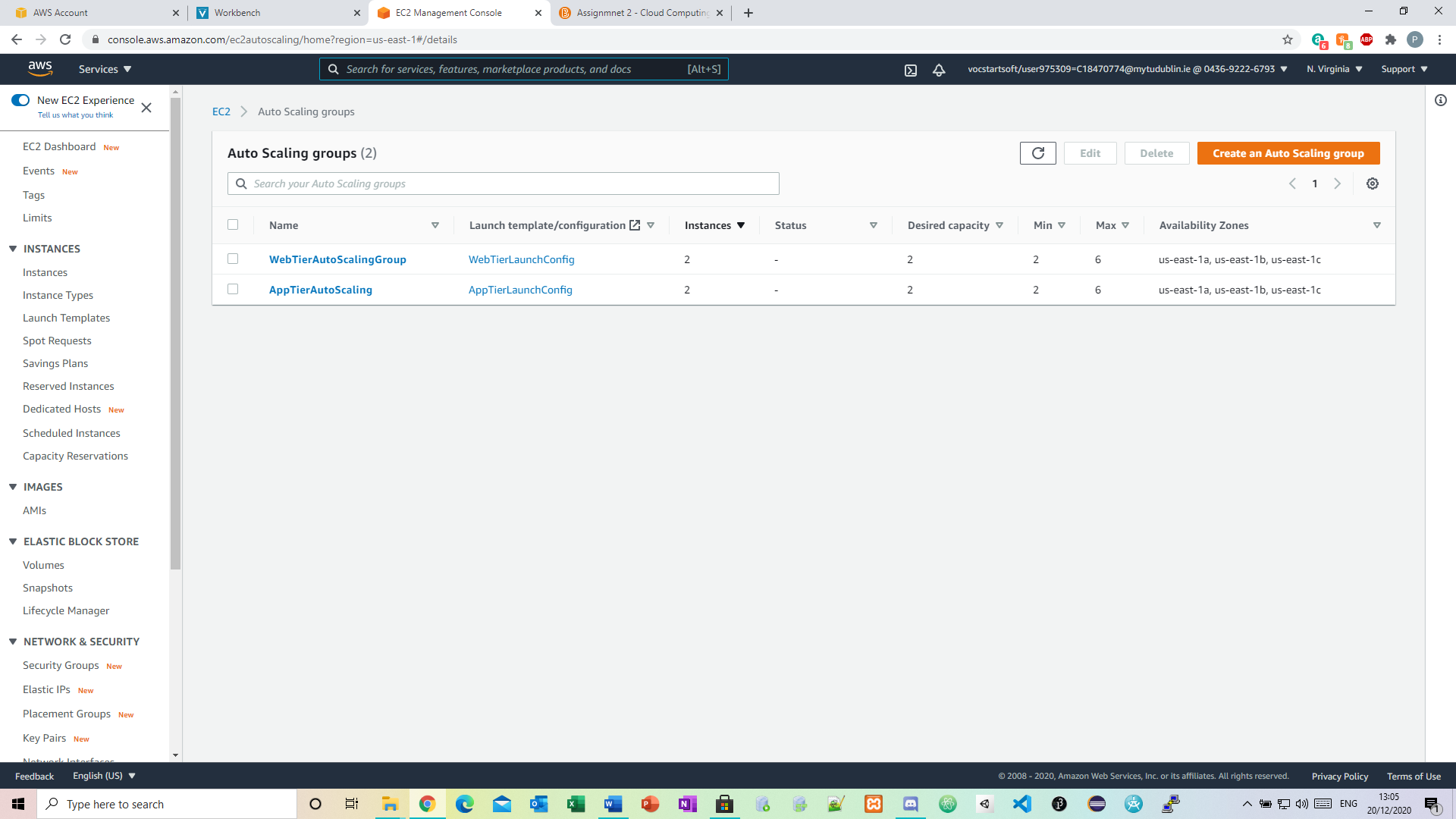
## Subnets:



## Load Balancers:



## Autoscaling Groups:



## Final Step Showing Access:

