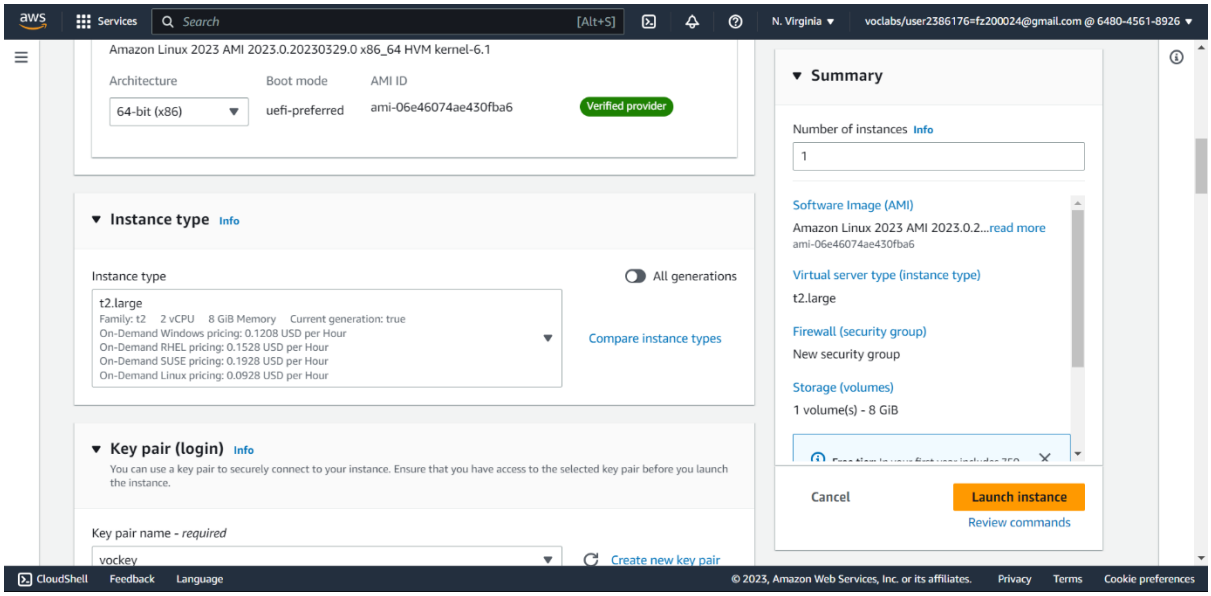
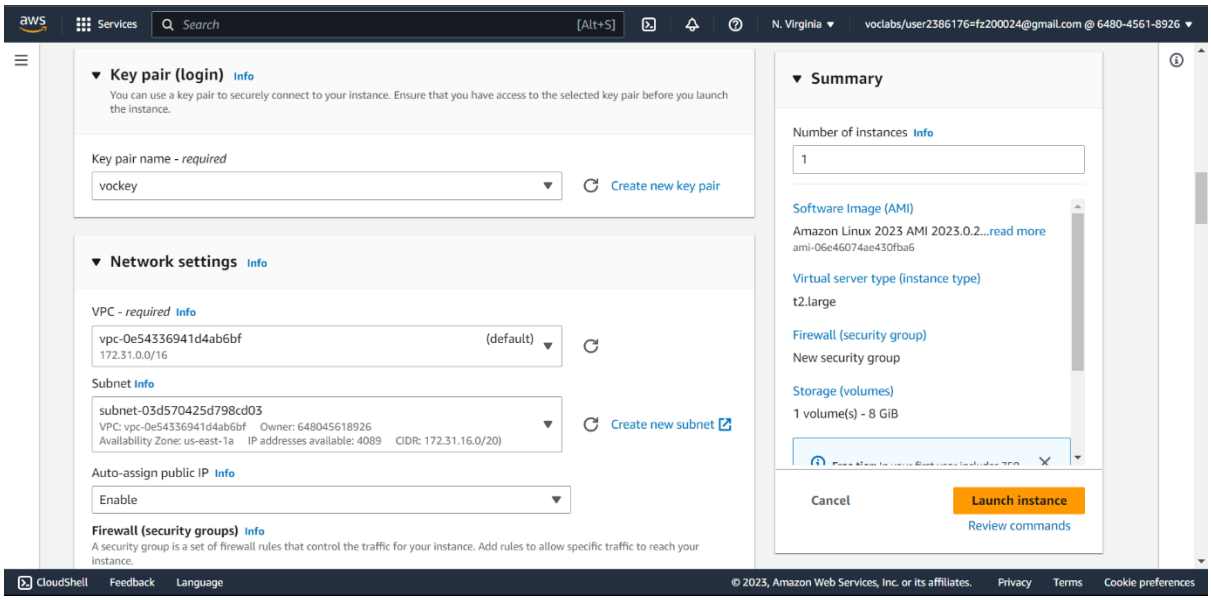
PART A:

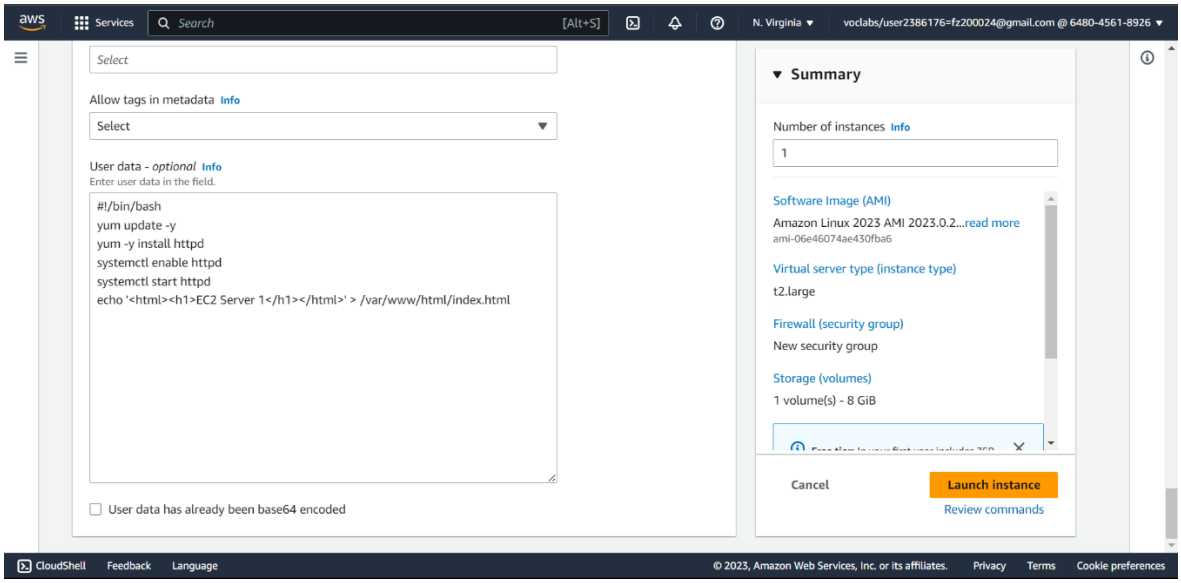
To begin the creation of the three-server load-balanced system, three instances of the EC2 server, namely EC2 Server 1, EC2 Server 2 and EC2 Server 3 are first to be created with the t2.large instance type being dual-core with 8GBs of RAM.



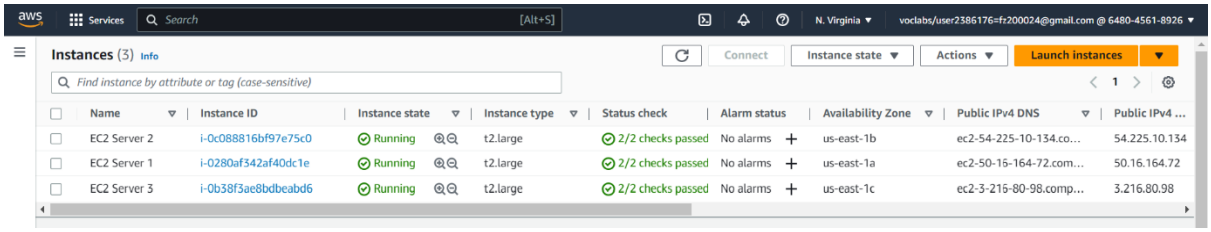
Since all three servers need to be in separate availability zones, each of them are assigned to the zones, us-east-1a, us-east-1b, & us-east-1c respectively through the subnet drop-down list.



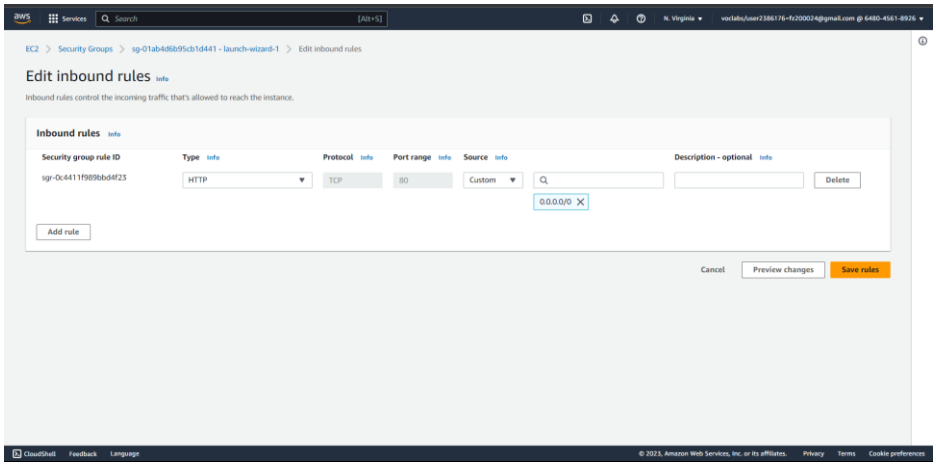
In the user data text field, the following HTML code is entered for the server to be able to display HTML pages that indicate their names for each page (given is only for EC2 Server 1, text for the server number is changed according to the server name in which website is being added)



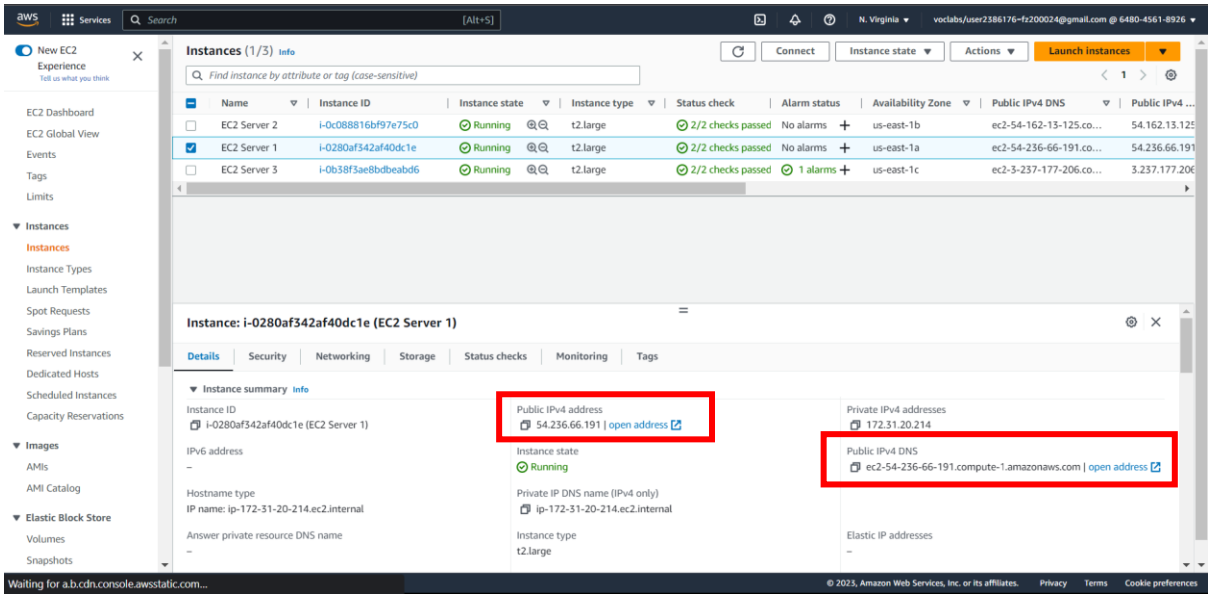
The three EC2 instances have been created

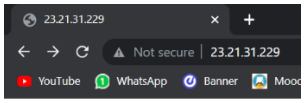


After creation of the three EC2 instances, on the left panel under the Network and Security, in the Security Groups page, the three new security policies that were created upon the creation of the EC2 instances are each modified to be of type HTTP that allow for any and all inbound IPv4 addresses and traffic to reach the instance. This makes the website in the instance public for everyone to view.

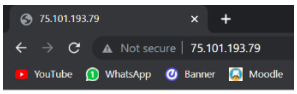


Following is the result of accessing each instance's respective websites by pasting their respective public IP address into the browser search bar:

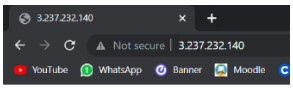




EC2 Server 1

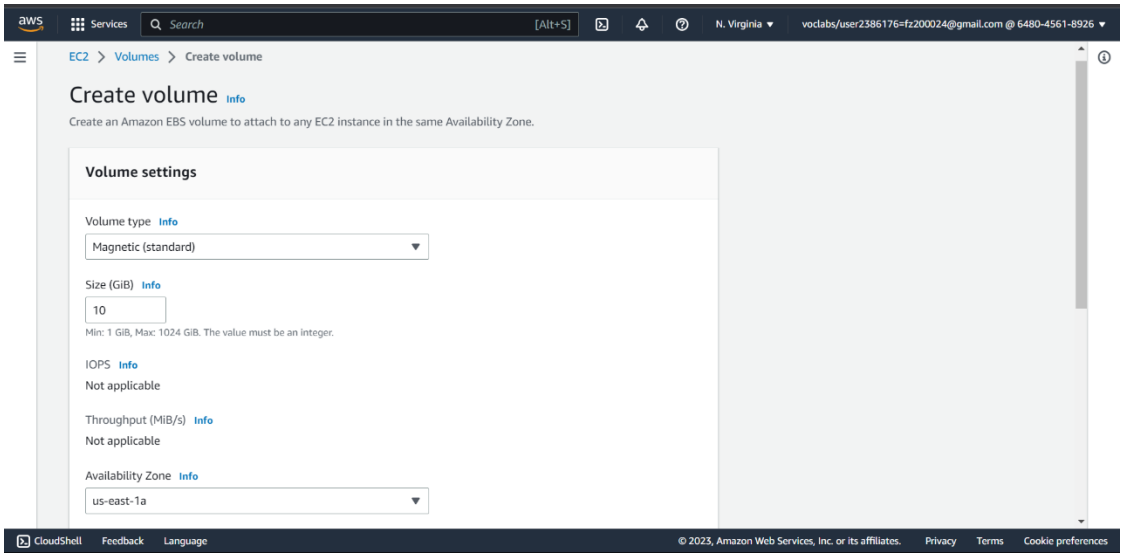


EC2 Server 2

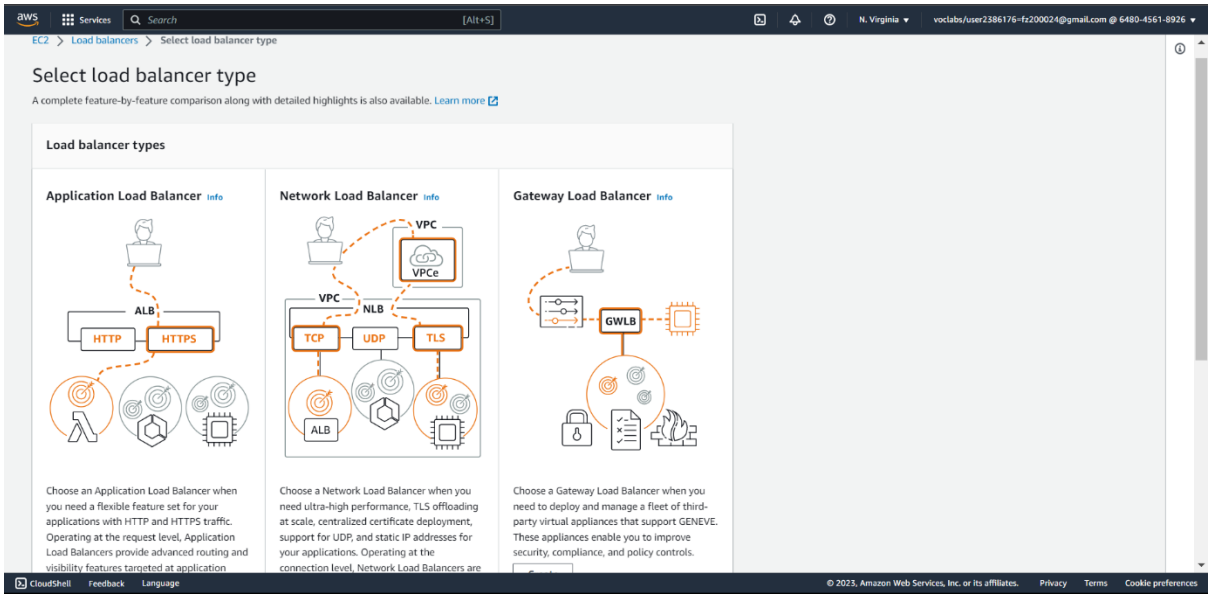


EC2 Server 3

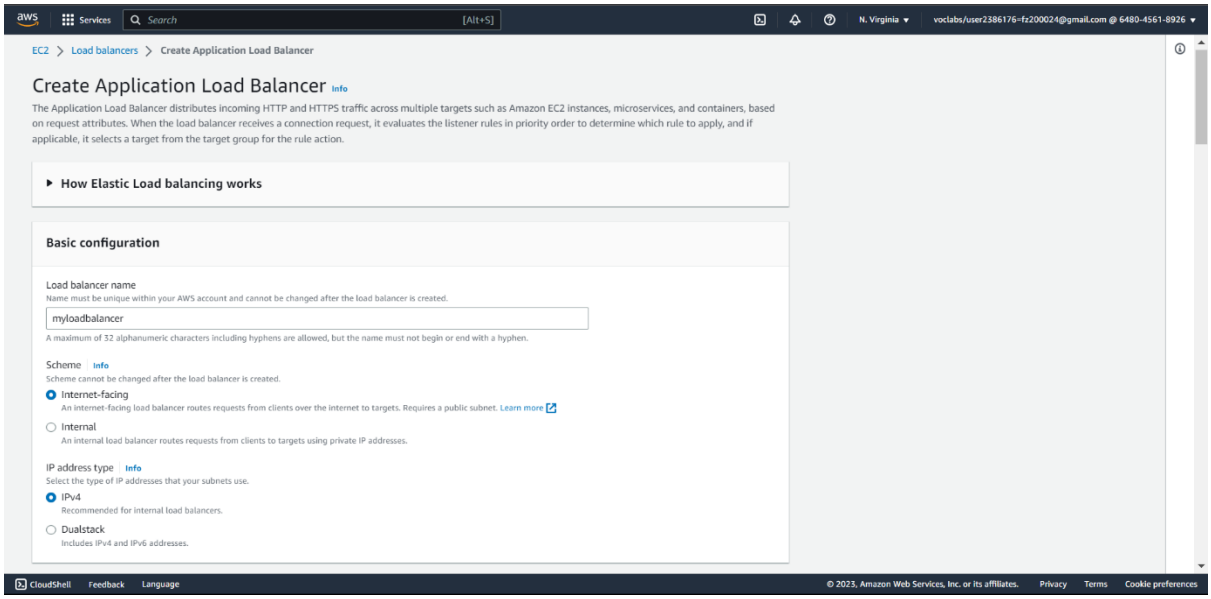
Now, three EBS Volumes by the name of EBS Volume 1, EBS Volume 3.1, and EBS Volume 3.2 are created with each volume having 10GBs of Magnetic (standard) storage each. The volumes are assigned different availability zones based on the EC2 instance that they are to be attached to. For EBS Volume 1, availability zone is configured as us-east-1a since it will be attached to EC2 Server 1, whereas for EBS Volume 3.1 and 3.2, the availability zone is configured as us-east-1c for both since they will both be attached to EC2 Server 3.

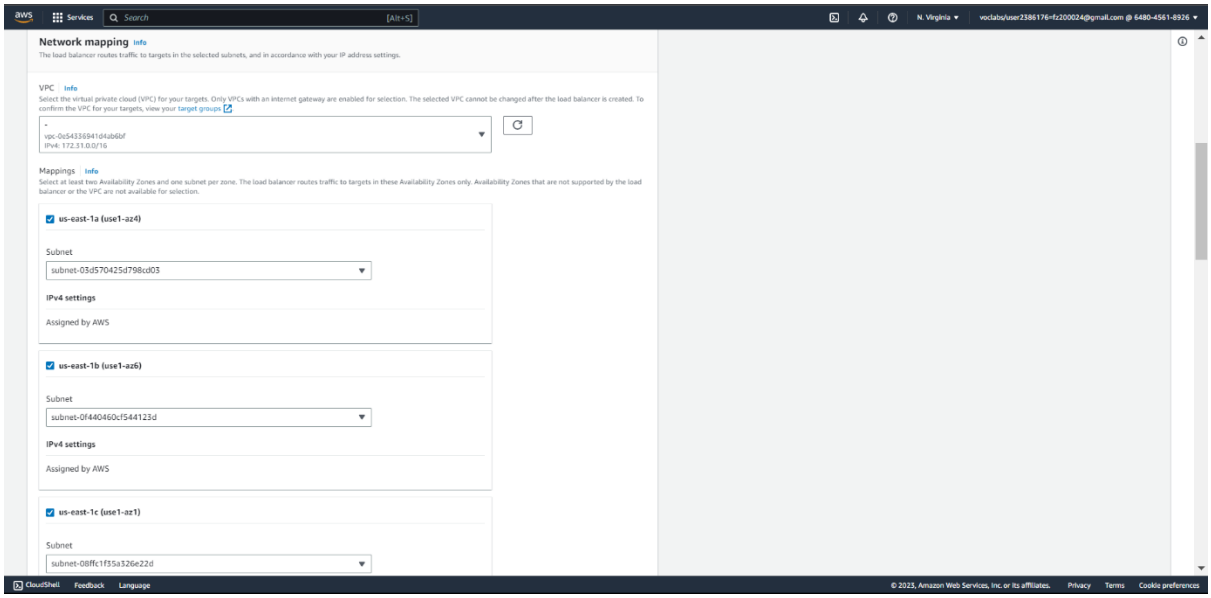


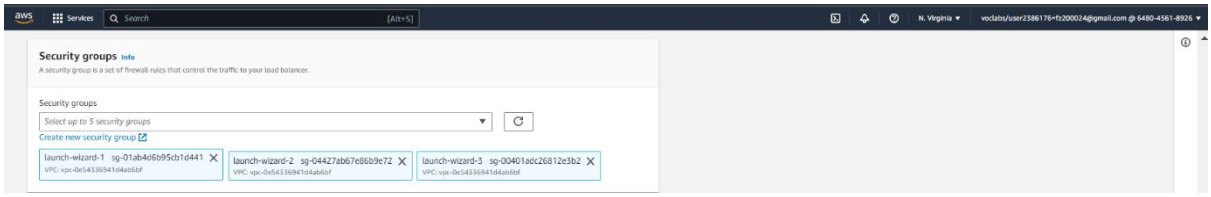
Next, an Elastic Load Balancer is to be created to handle the traffic load for the three servers, from the Load Balancers page that is accessed from the EC2 left hand pane. An application load balancer is chosen to be created, because this load balancer service is specifically designed for being capable of distributing incoming traffic across multiple targets, including EC2 instances. Considering that in this case EC2 instances have been used for server creation and static website hosting, an Application Load Balancer is the best choice in this scenario.



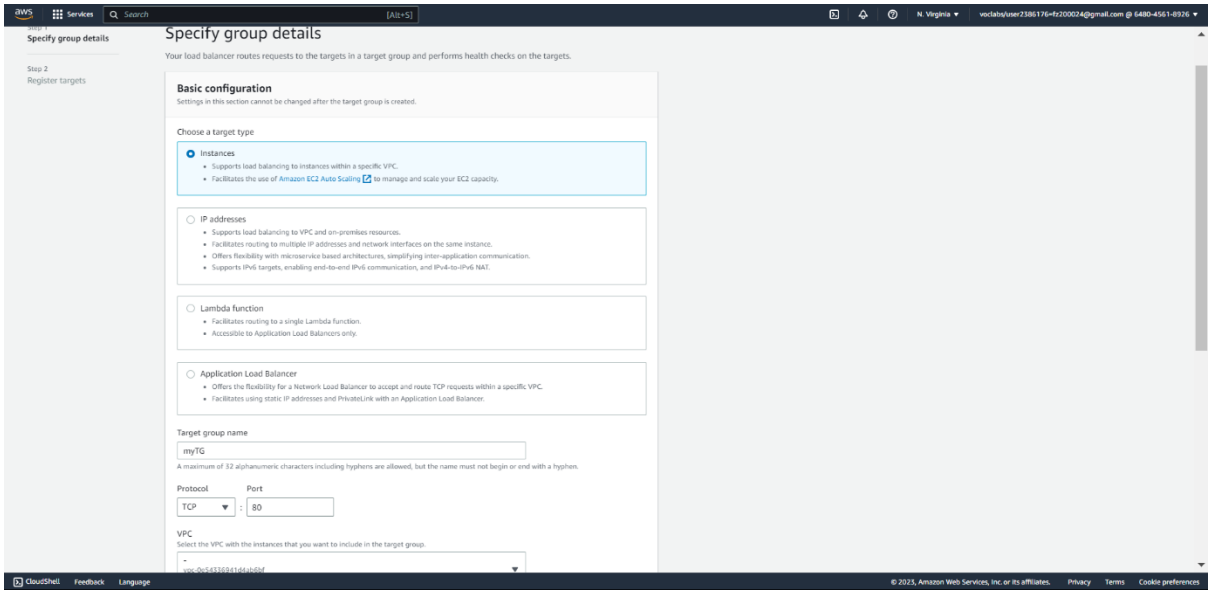
The load balancer is assigned a name **myloadbalancer** and the three availability zones of the EC2 instances EC2 Server 1, EC2 Server 2 and EC2 Server 3.

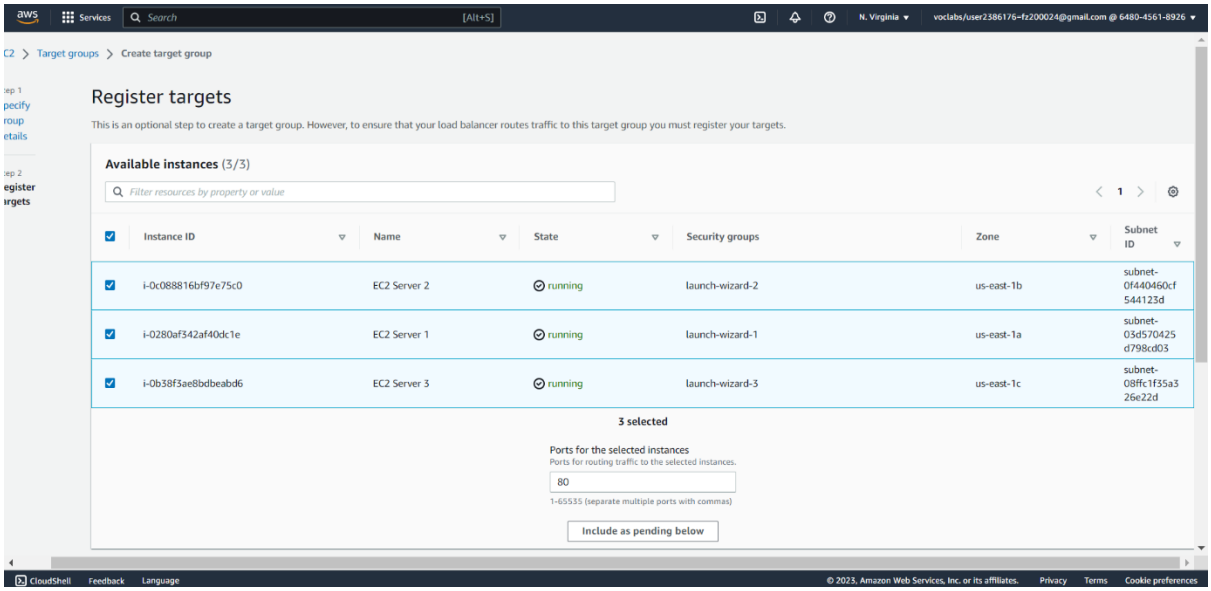


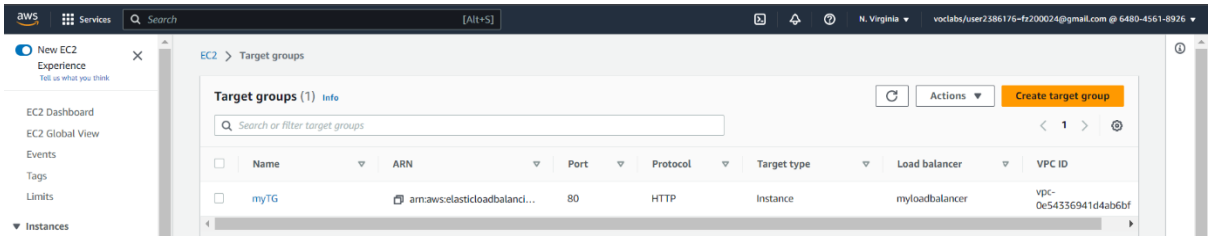




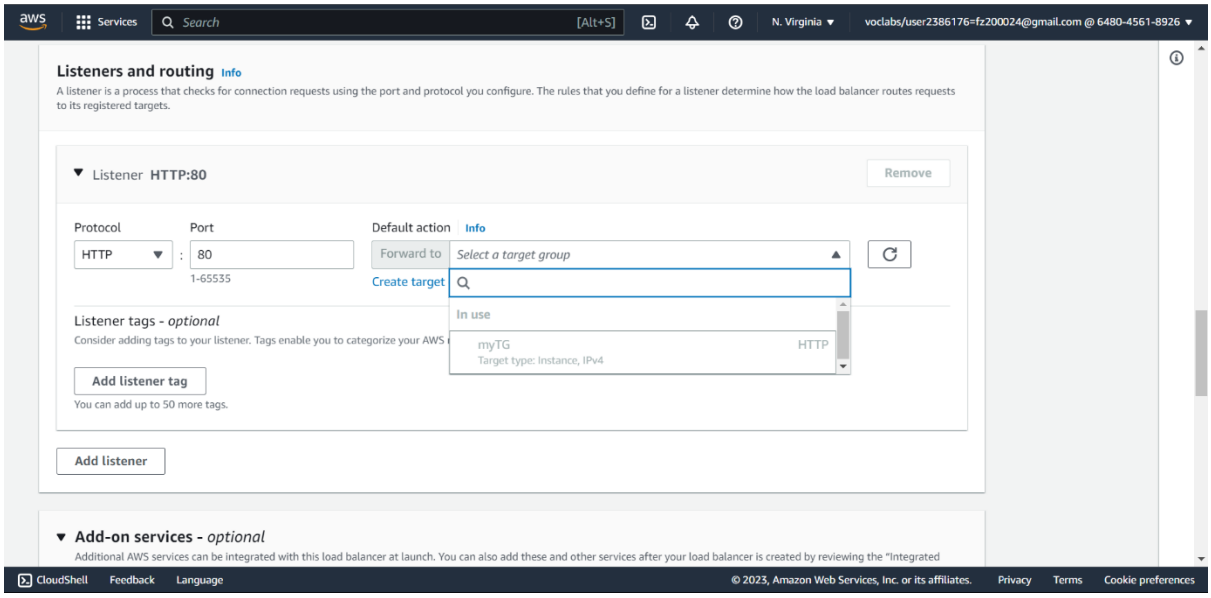
A target group with the name of **myTG** is created to be added a listener for the load balancer.



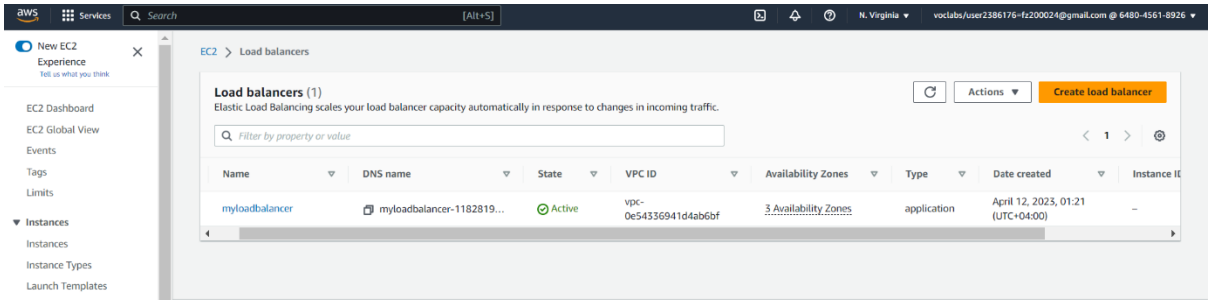


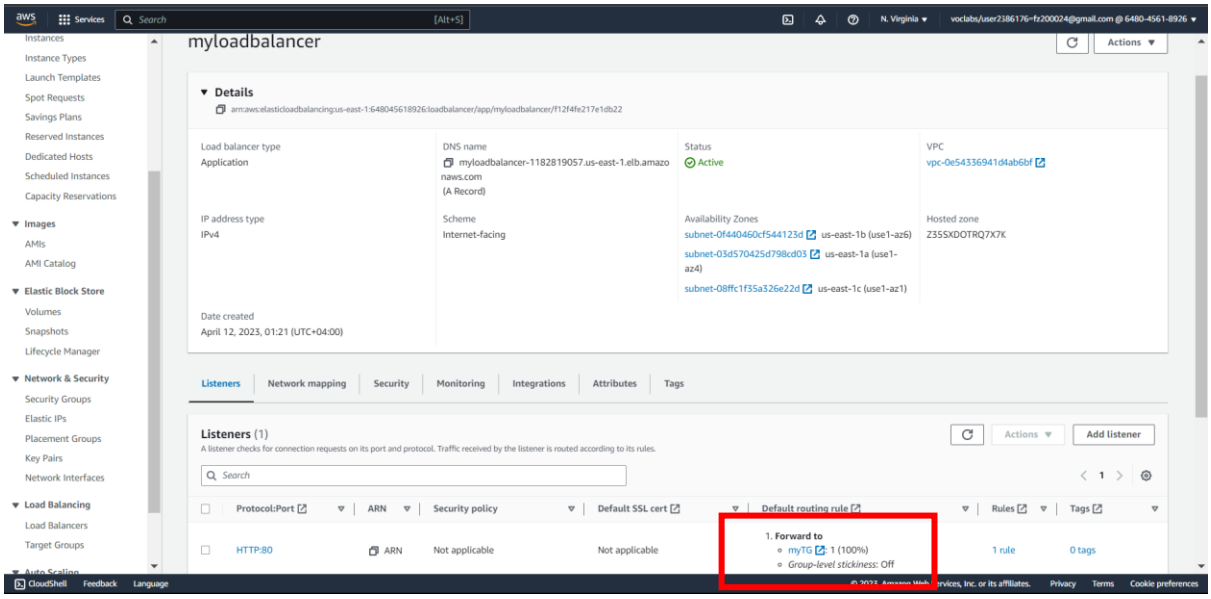


Note: I forgot to screenshot this screen when initially selecting the target group, so I'm just showing the reselection again for the already selected target group which is why it is greyed out.

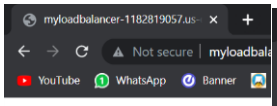


The load balancer has been created and is tested out with the following results after refreshing every couple of times:

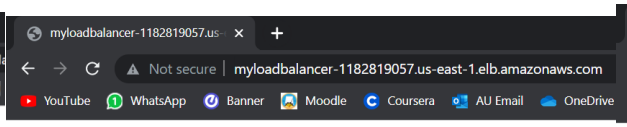




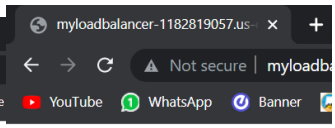
Results:



EC2 Server 2

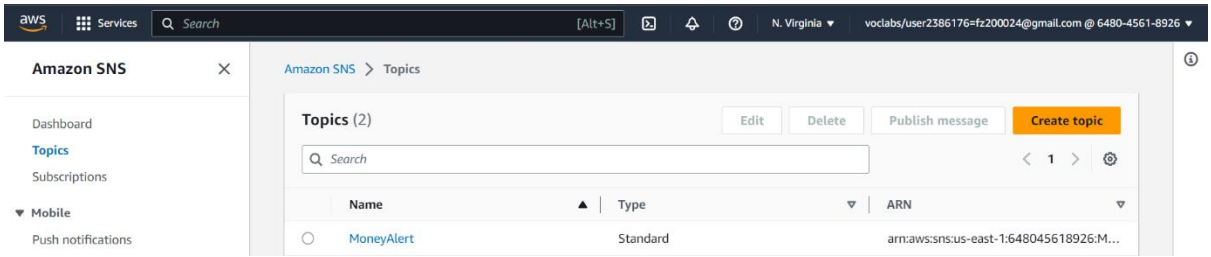


EC2 Server 1

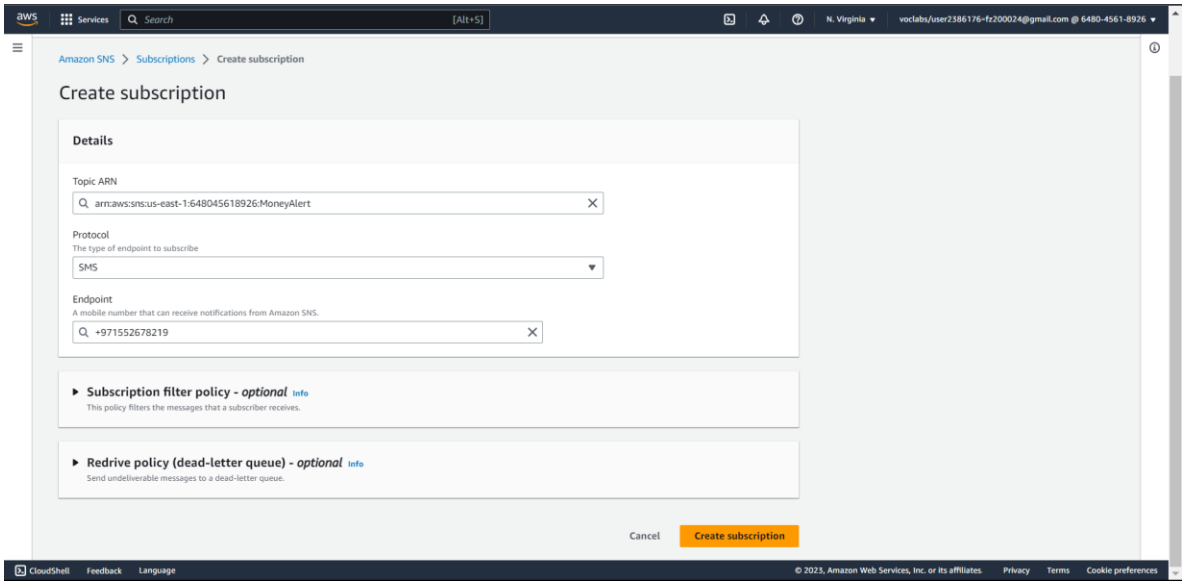


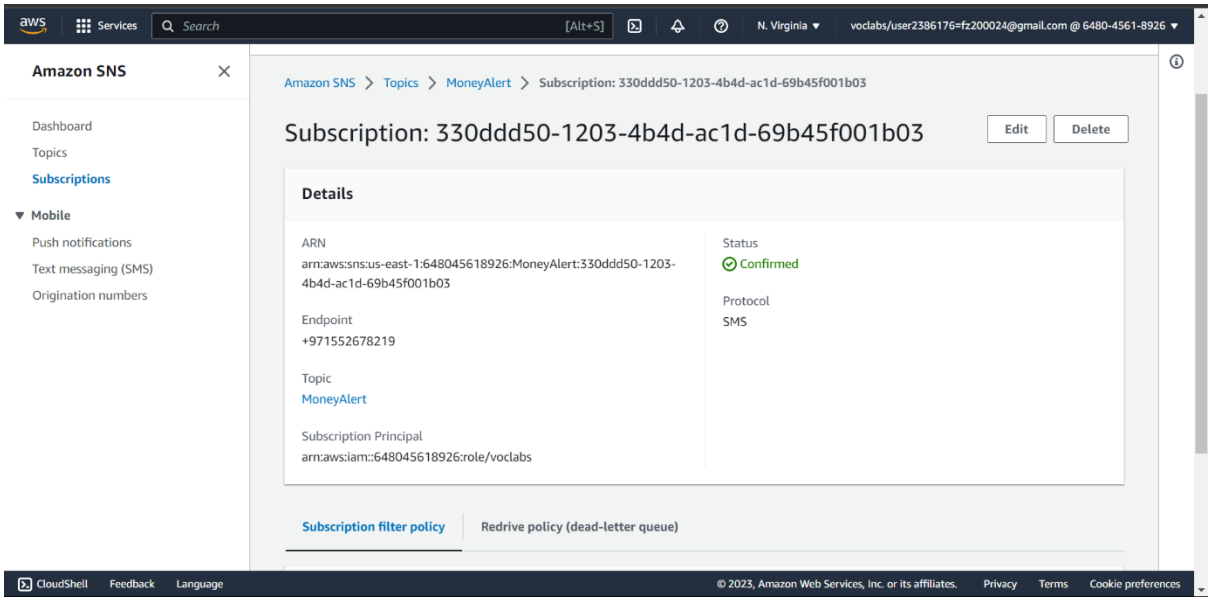
EC2 Server 3

An alarm service is to be created now for monitoring the estimated charge of EC2. First, an SNS (Simple Notification Service) Topic by the name of MoneyAlert is created.

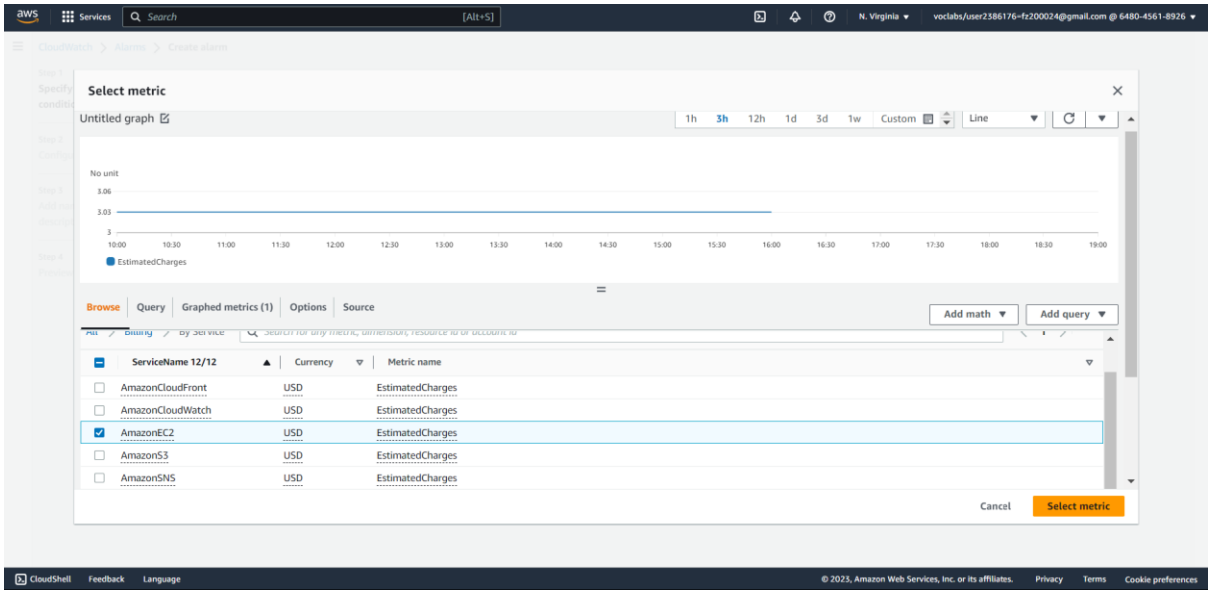


Then a subscription for the MoneyAlert topic is created using the SMS protocol, and a verified phone number.

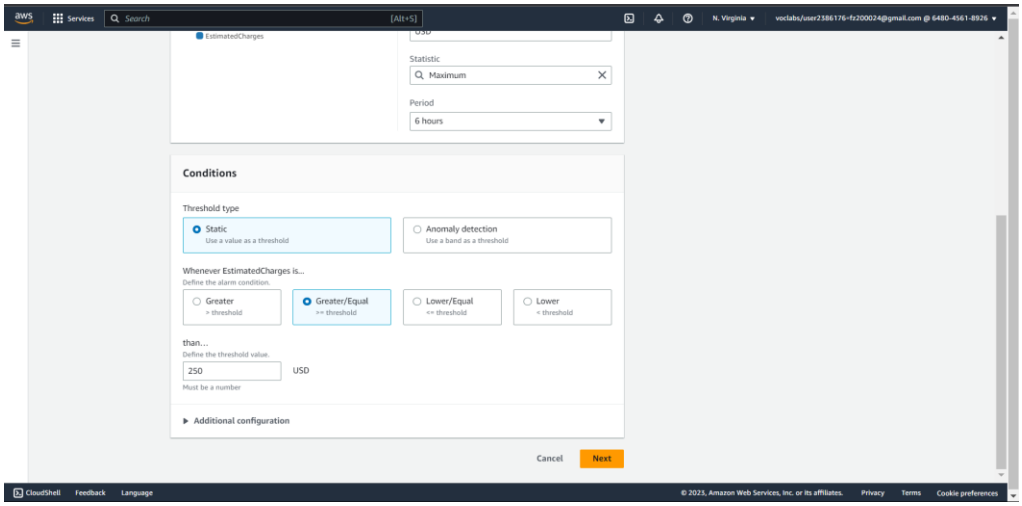




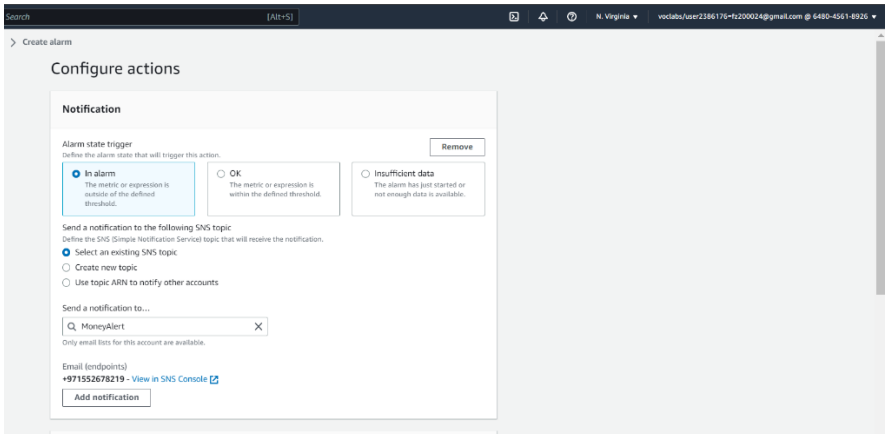
After this, a CloudWatch alarm is created. Since the alarm should monitor the estimated charge for EC2 only, therefore when selecting a metric, and then going to Billing>>By Services, The AmazonEC2 metric is selected.



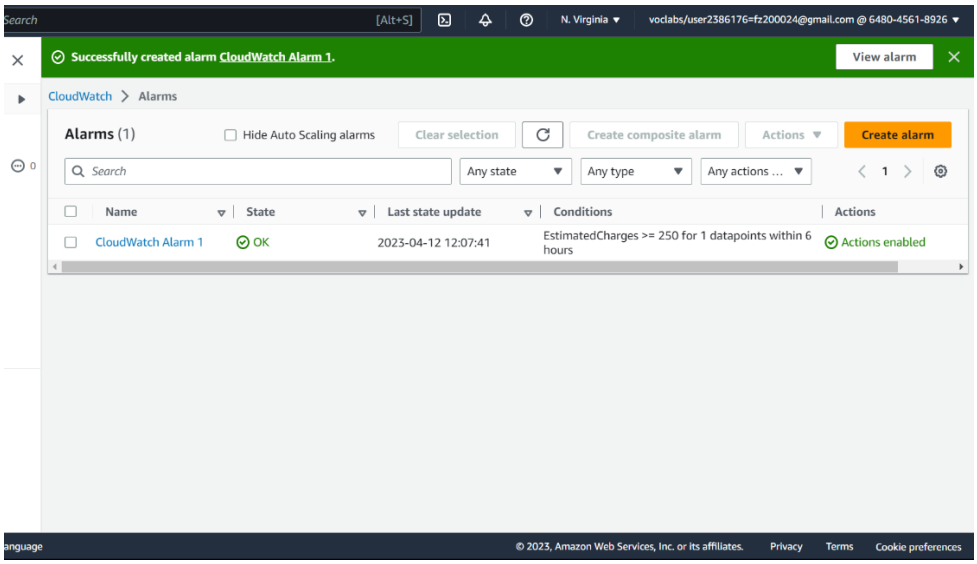
The alarm is then configured with a static threshold type, and the threshold value is defined as greater/equal to 250 USD.

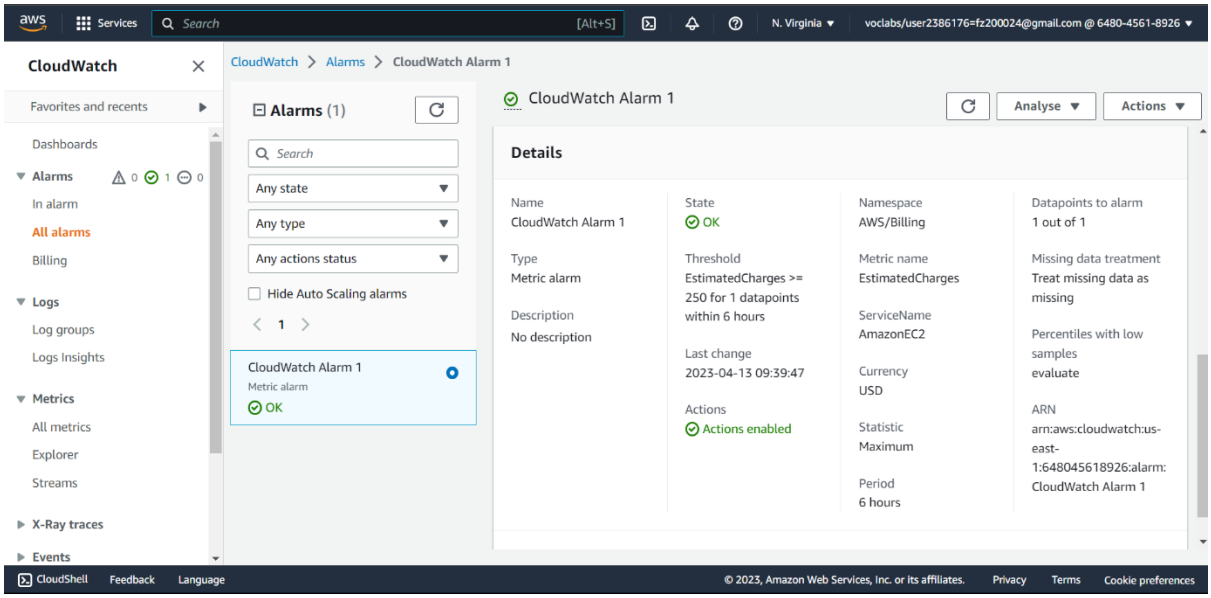


The alarm is meant to be created for the existing SNS topic of MoneyAlert.

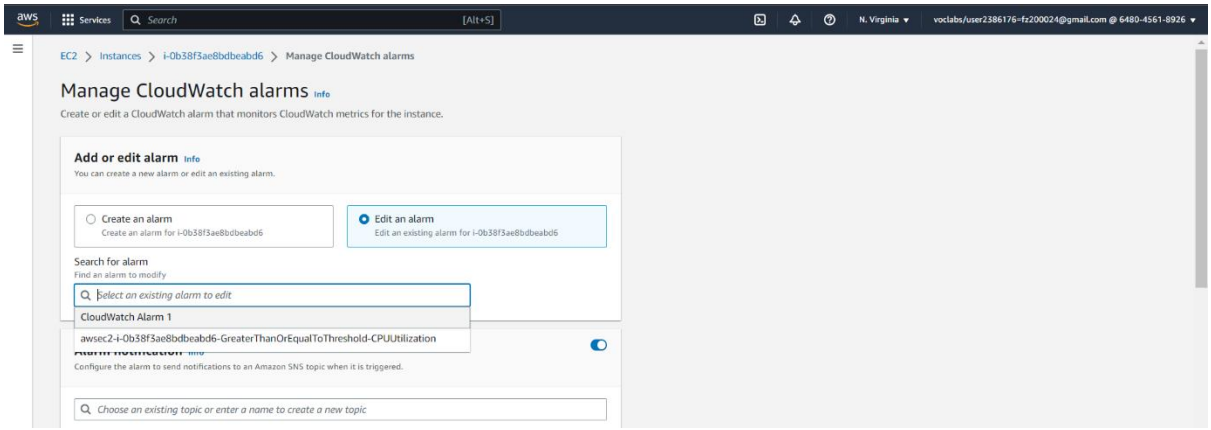


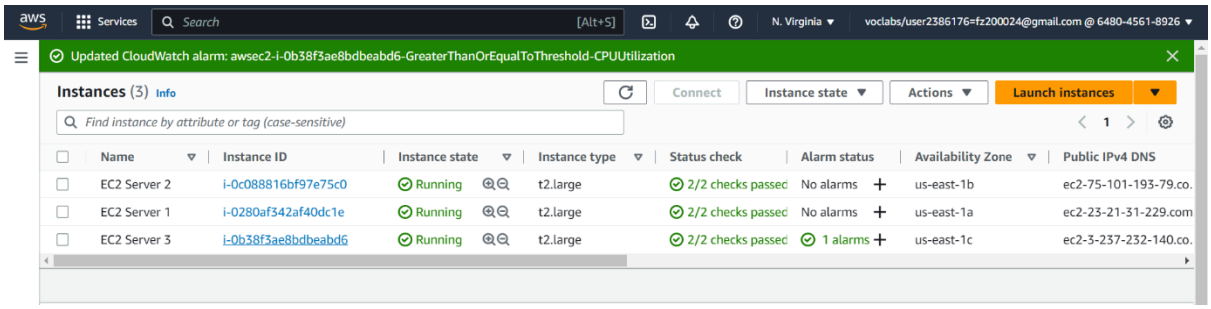
After finally configuring the alarm name as CloudWatch Alarm 1, the following is shown for cloud watch alarm:





This alarm is now to be attached to the EC2 Server 3 instance. This is done by adding an alarm to the EC2 Server 3 instance.

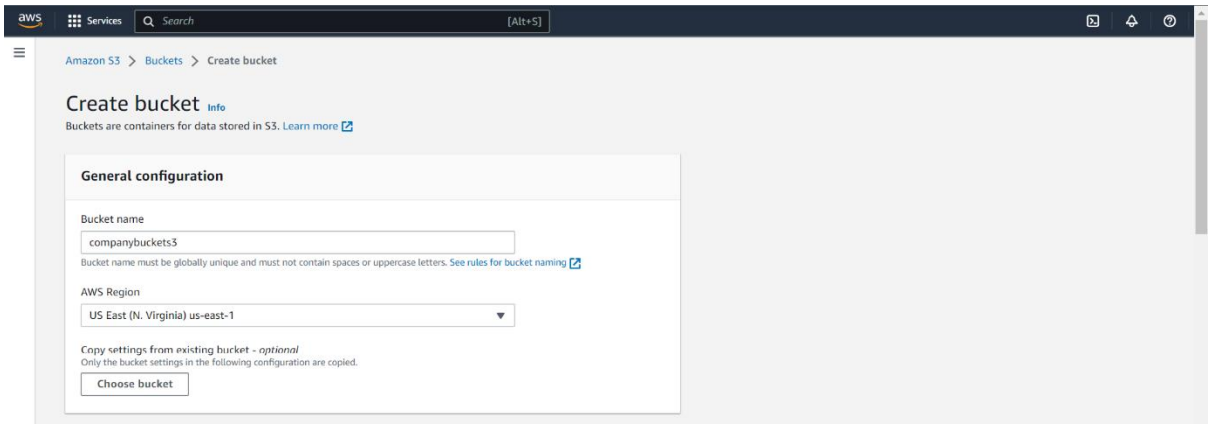


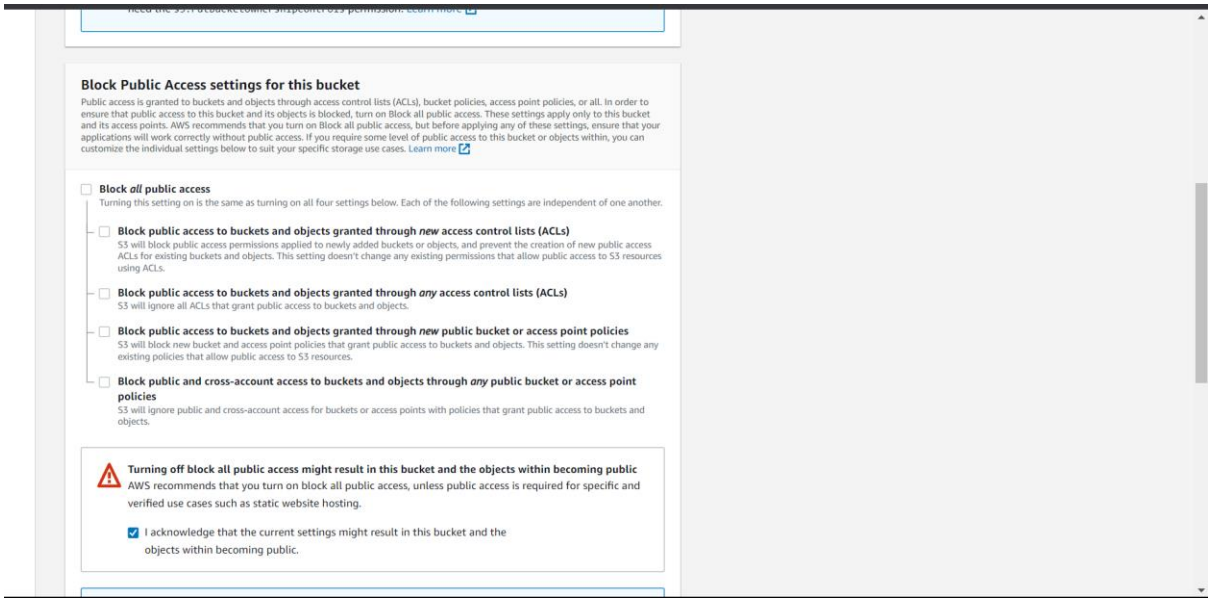


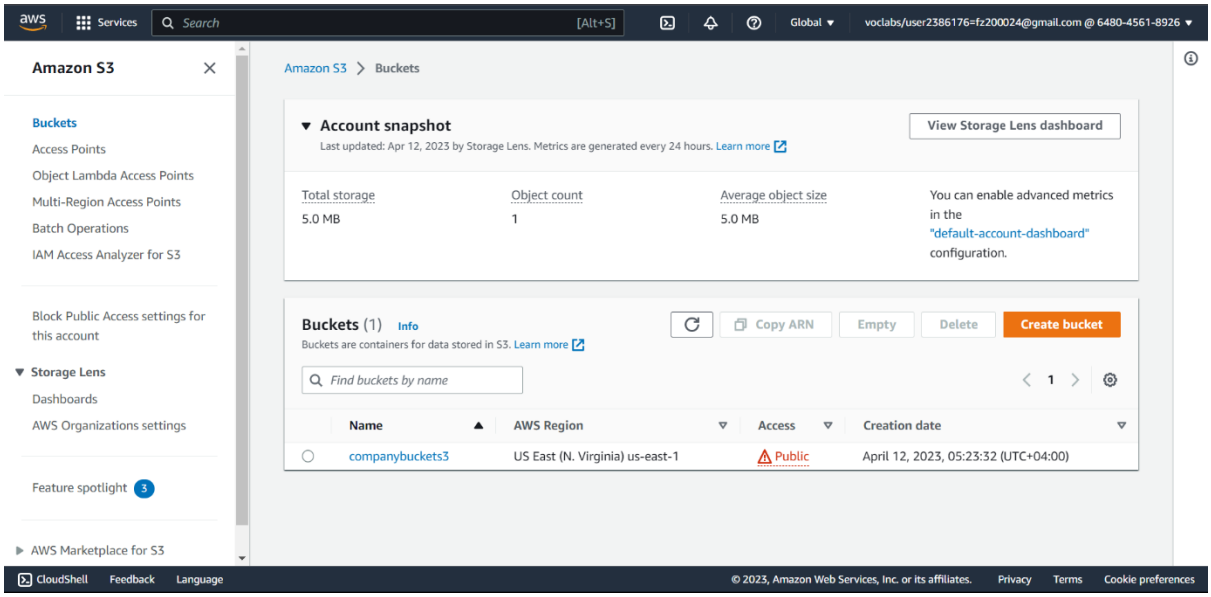
The three server load balance system creation is complete.

PART B:

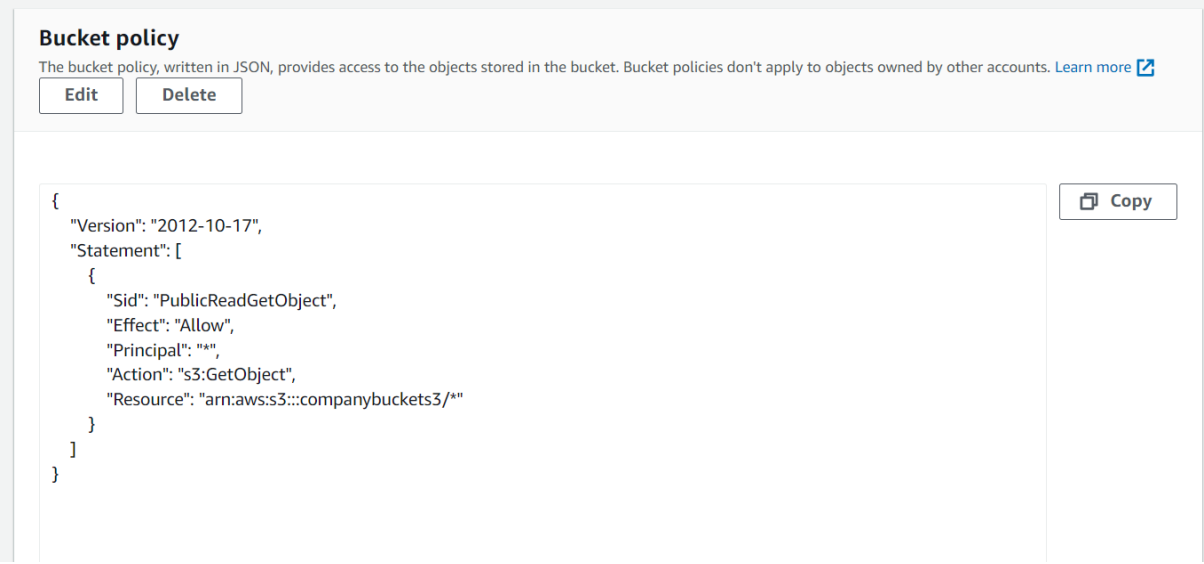
To begin the creation of an S3 bucket with company specific data, an S3 bucket is created first with the name **companybuckets3,** and granted public access.



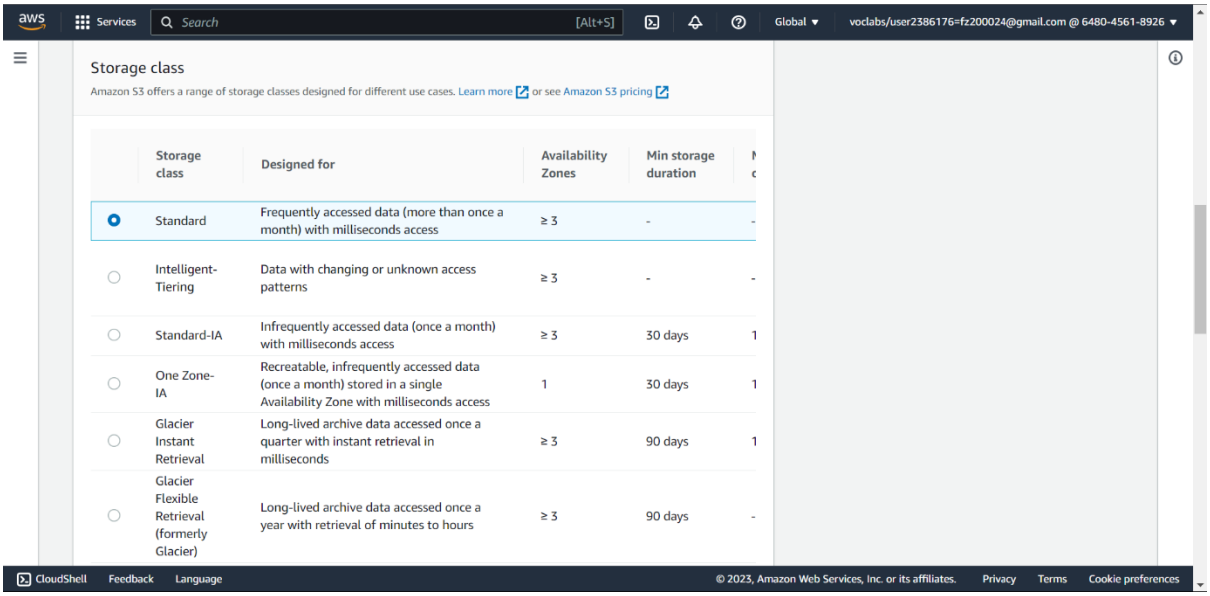




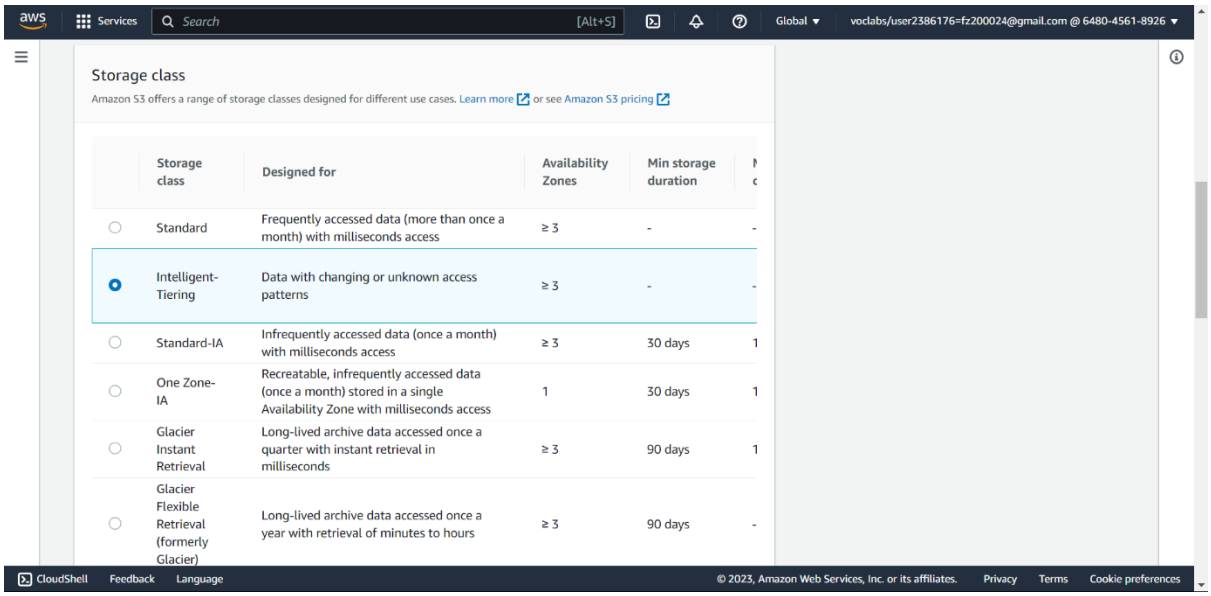
The bucket permissions are then accessed, and modified by entering the following JSON code in the bucket policy text field to provide public access to the objects stored in the bucket.



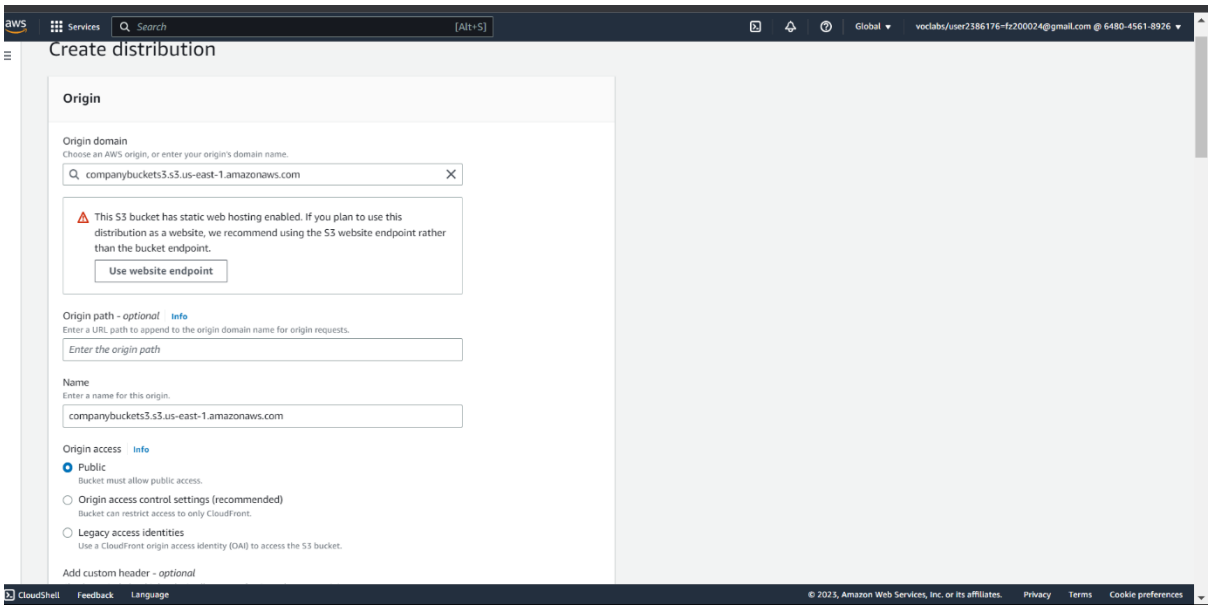
After this, files and folders are uploaded to the bucket. The relevant files and folders of the company website are uploaded to the S3 bucket with the Standard storage class.



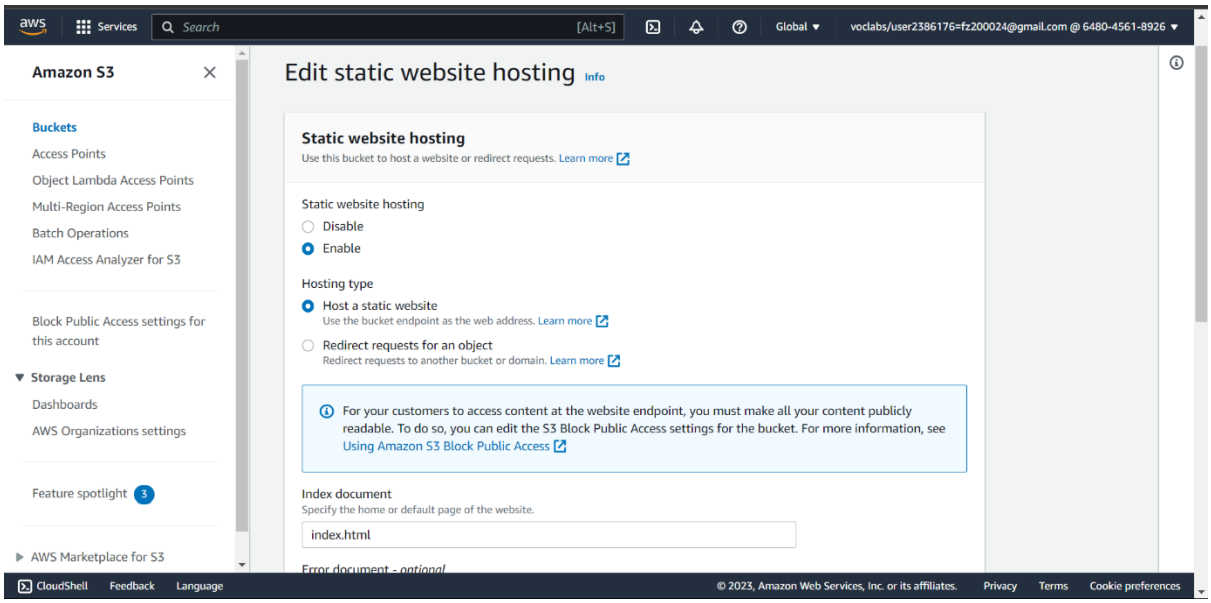
Then the salary log and the relevant files and folders of the media archives are uploaded to the S3 bucket with the Intelligent-Tiering storage class, since these objects consist of data with changing or unknown access patterns.



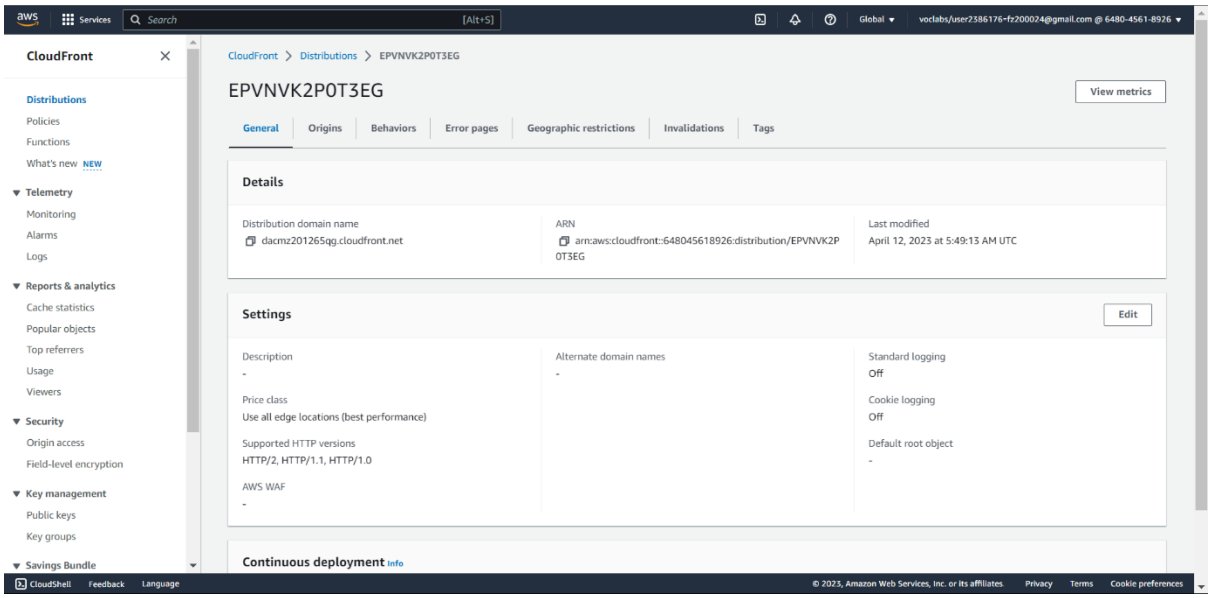
The S3 bucket files are then made to be publicly accessible via a CDN. A CloudFront Distribution is created for this purpose using the S3 bucket origin domain.



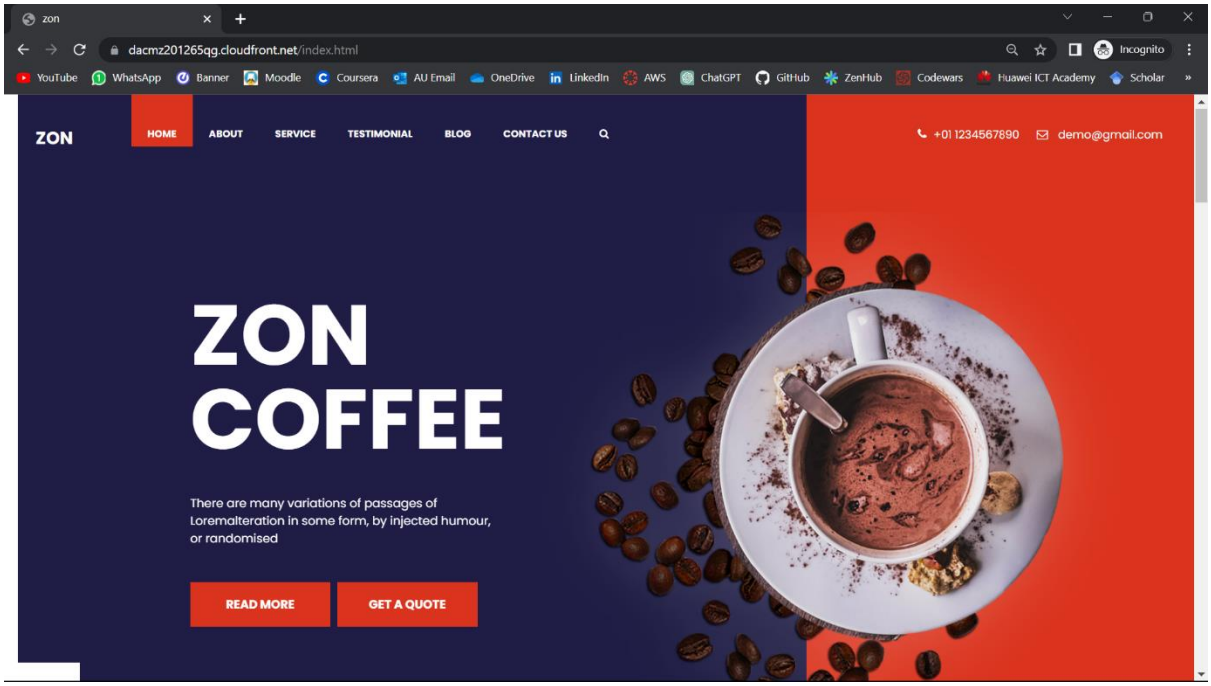
Going to the companybuckets3 properties tab, the static website hosting settings are edited to be enabled for the bucket hosting type, and the index document is specified as index.html, which is the home page of the website.



From the recently created CloudFront distribution, the distribution domain name is copied and pasted into the search bar of the browser, and the path index.html is appended to it as it is the homepage of the website also specified while modifying the static web hosting setting in the S3 bucket, making the link look like: https://dacmz201265qg.cloudfront.net/index.html



This link makes the website in the company’s S3 bucket publicly accessible via CDN.



The creation S3 bucket for company data is complete.

The cloud architecture has successfully been implemented according to the requirements given.