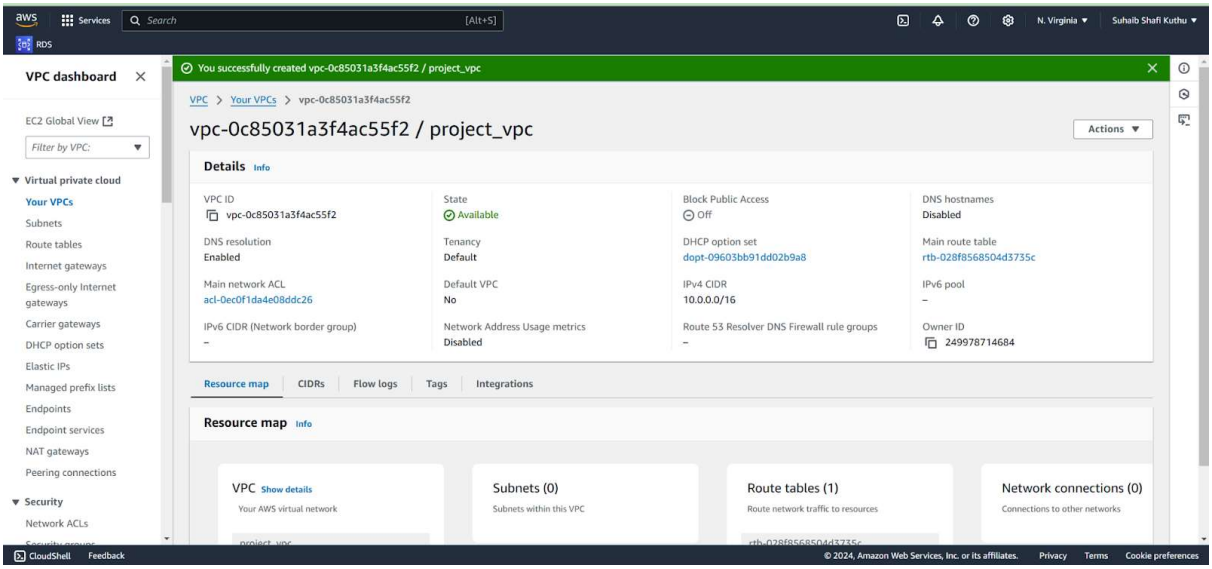
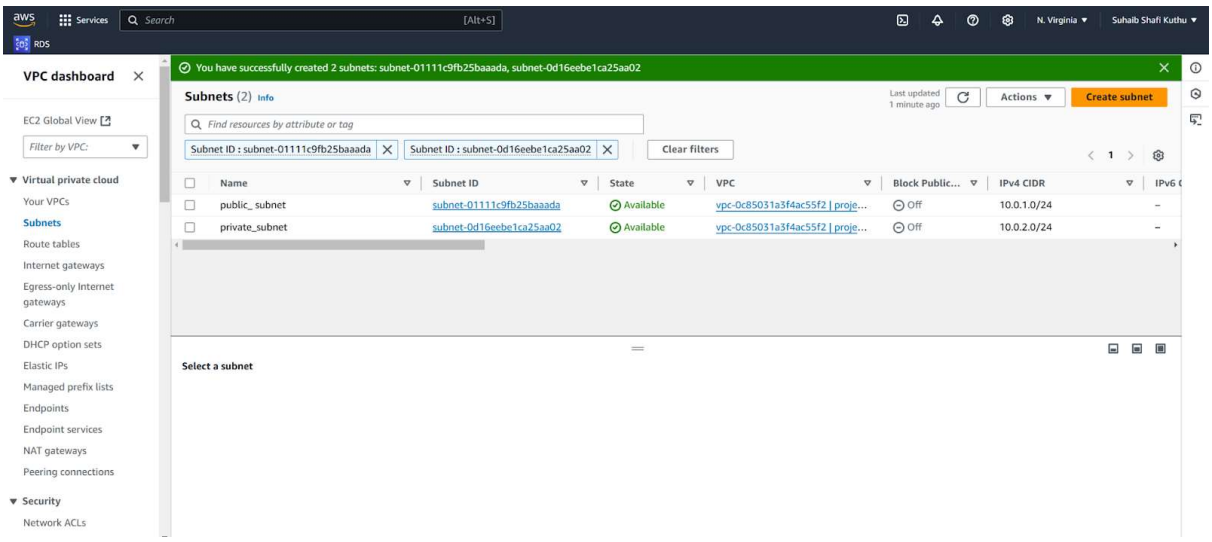
Project1

Below are the steps for the procedure to be followed to achieve the required architecture

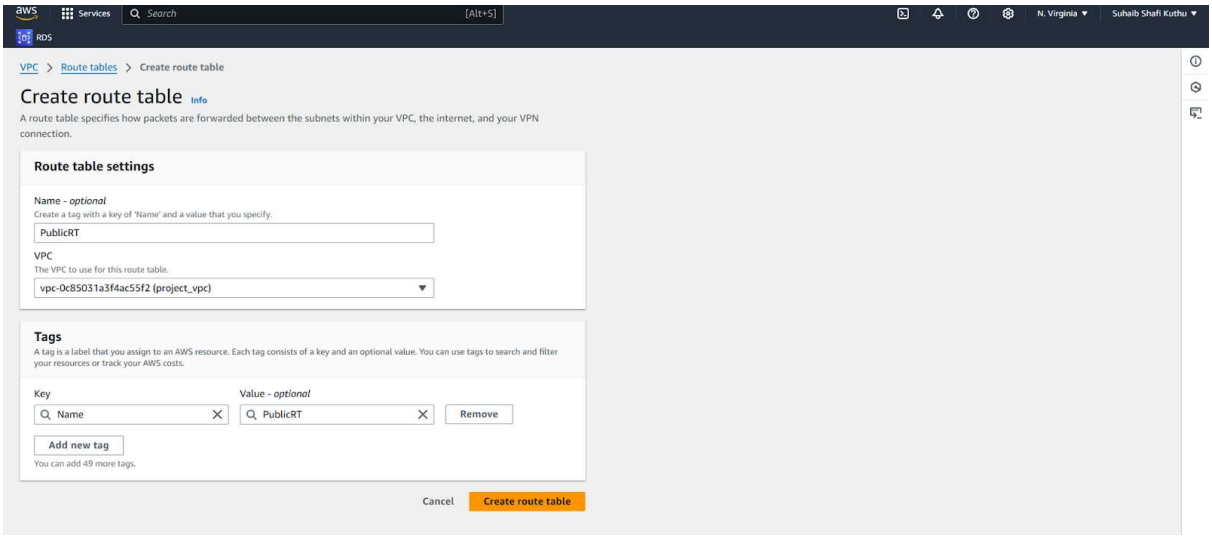
Step1:-
Create a vpc where the front end is kept in the public subnet and backend in private subnet.

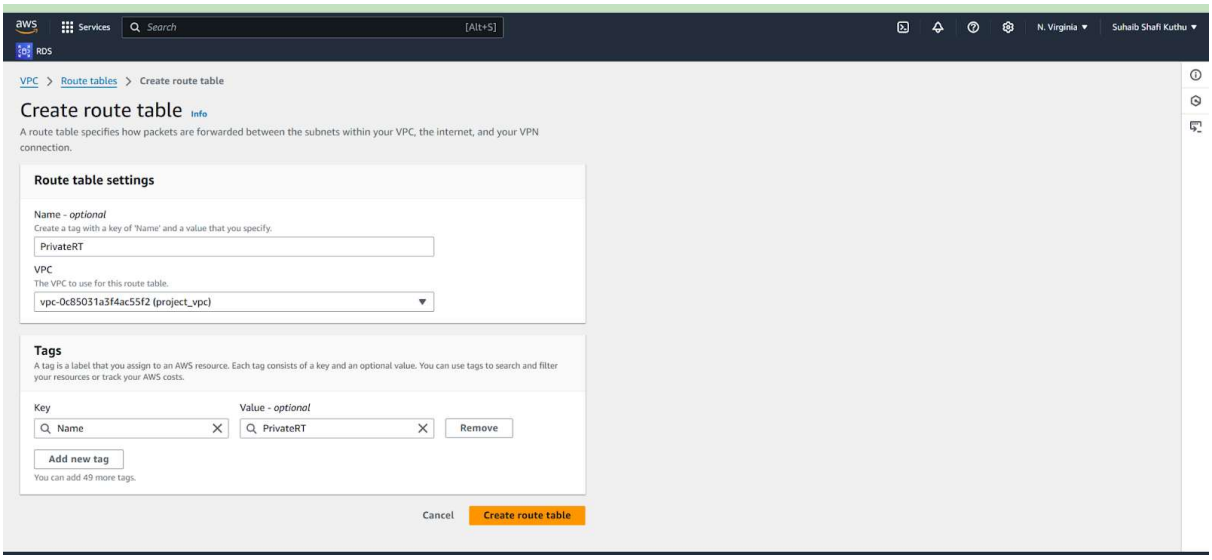


Step 2:- create subnets in the vpc with public access and private access



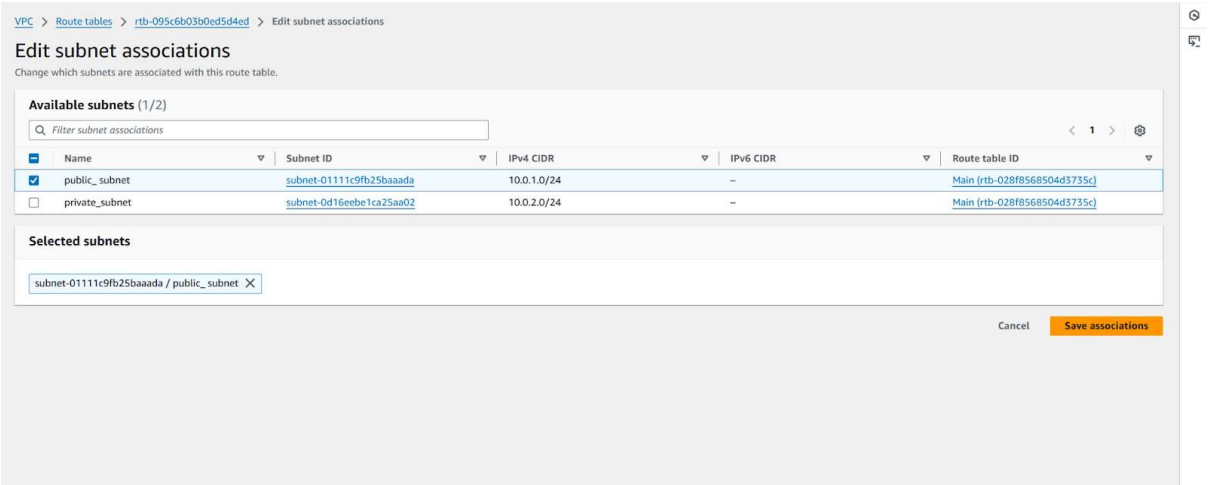
Step 3 :- create route tables for public and private subnets and associate accordingly

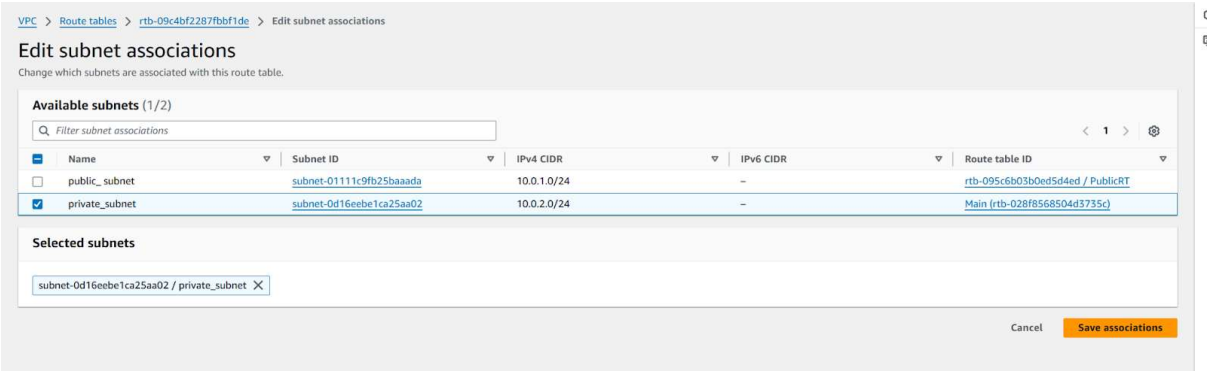




Step 4

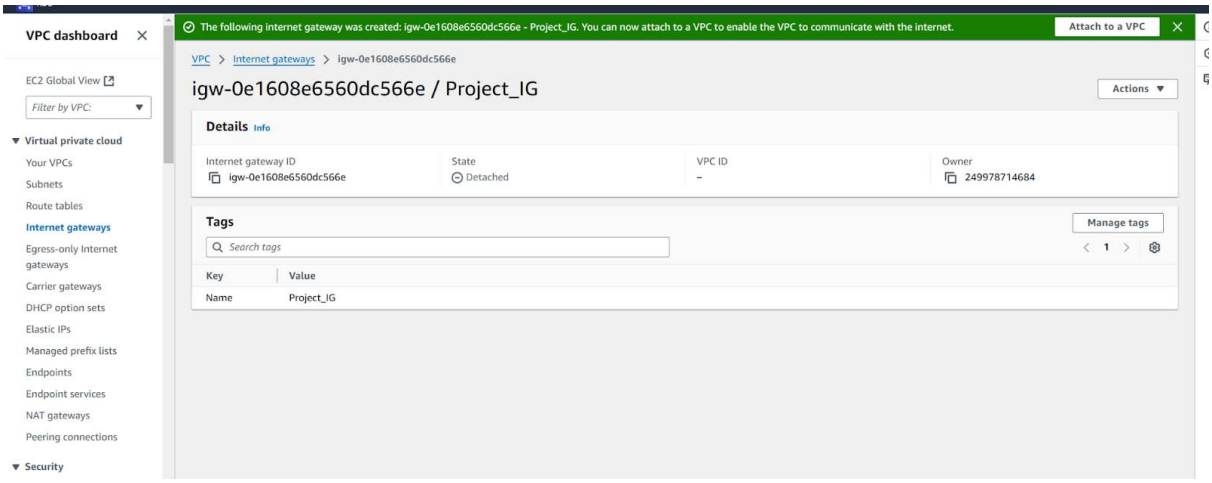
Subnet associations ,assoiciate public subnet with publicRt and private subnet for public rt



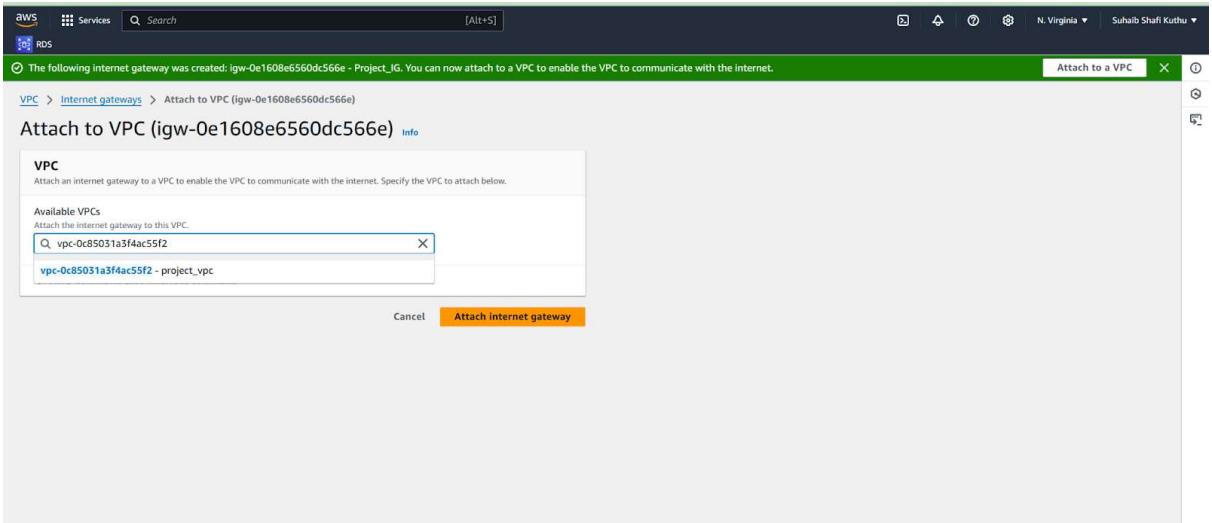


Step 5

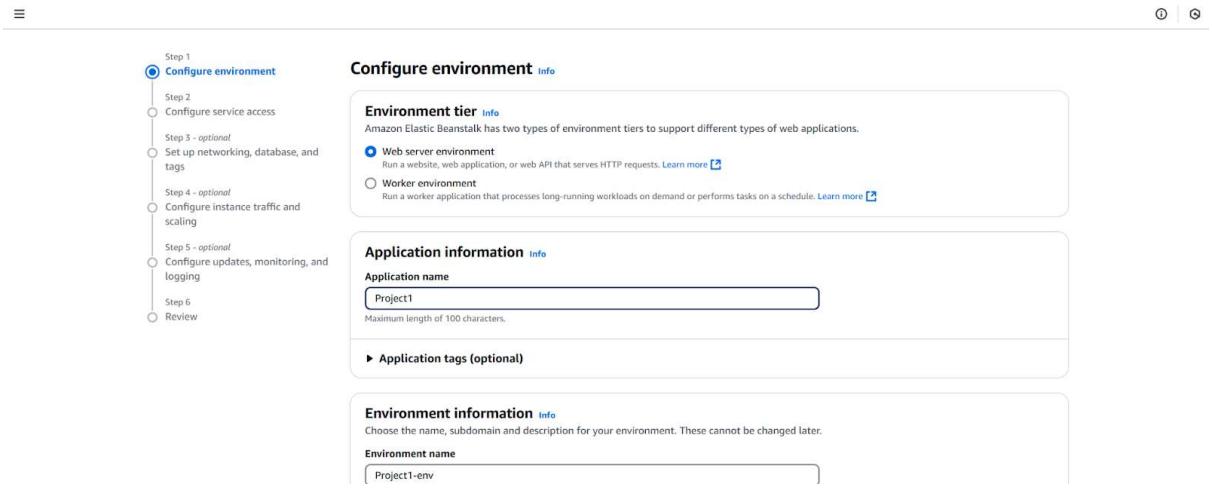
Create an internet gateway and attach it to vpc to make it available outside as public url.



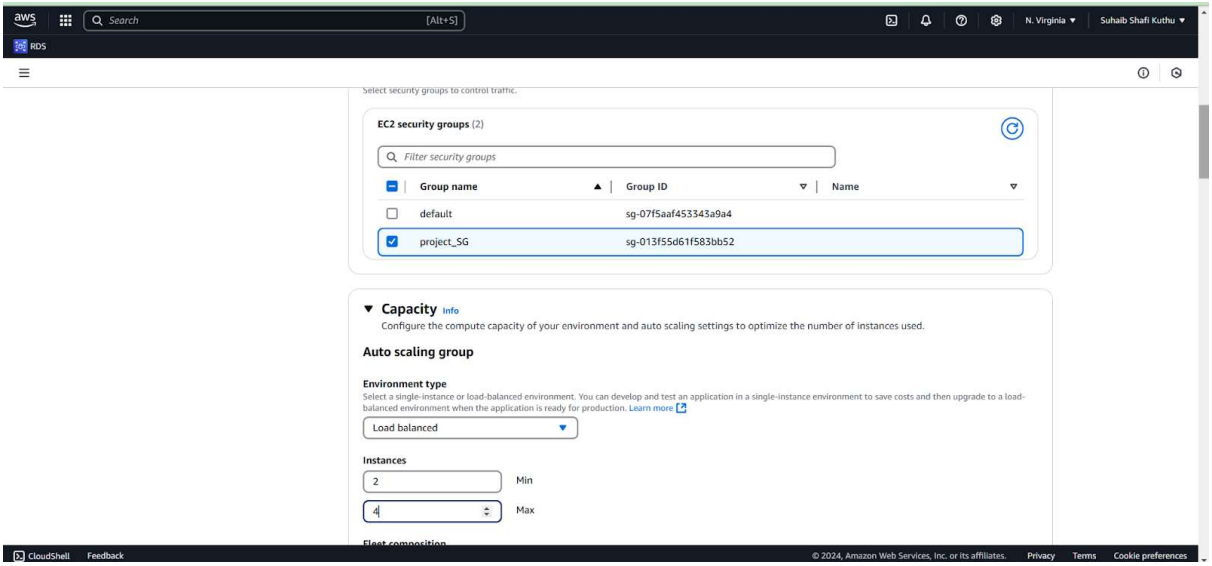
Attach it created vpc



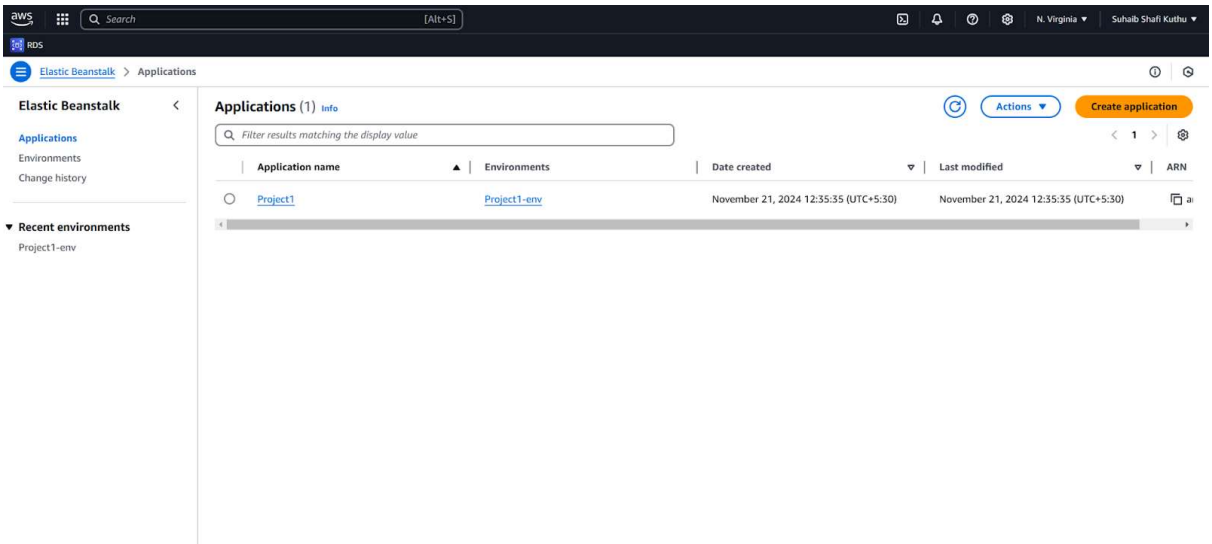
Step 6 :- go to elastic beanstalk and create an application with node.js environment



Configure networking and instance and load banacer settings as per the project

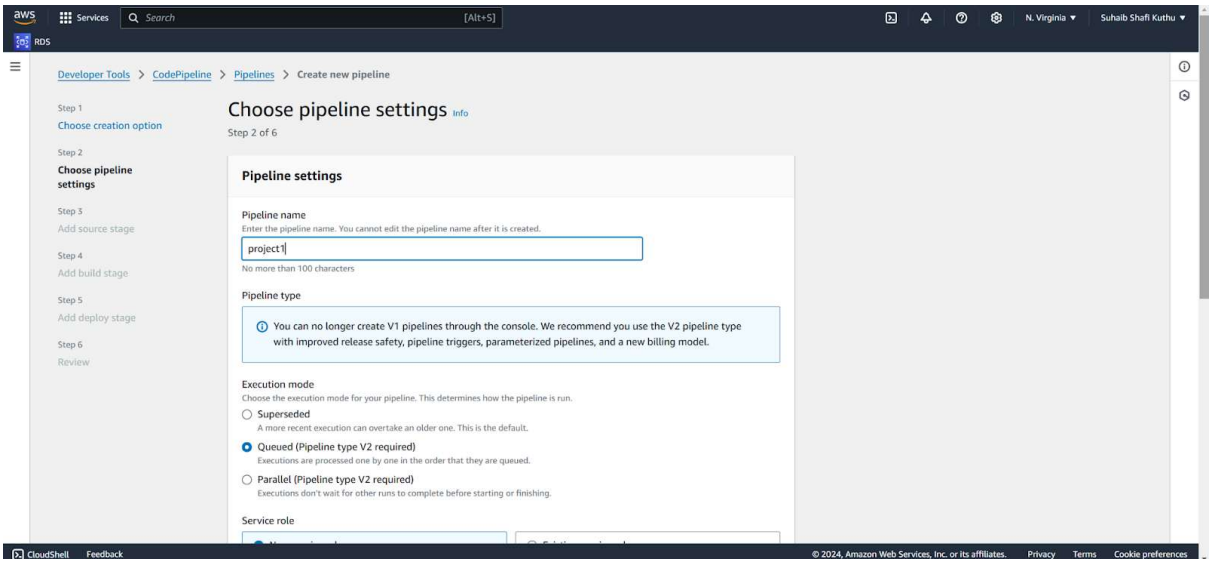


Project1

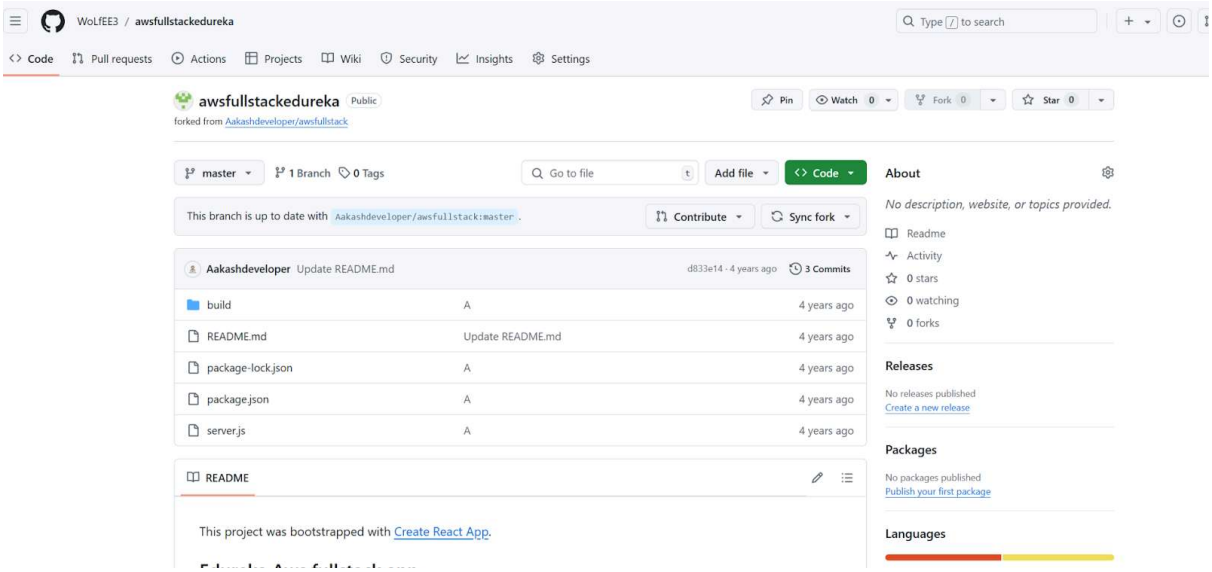


Step 7

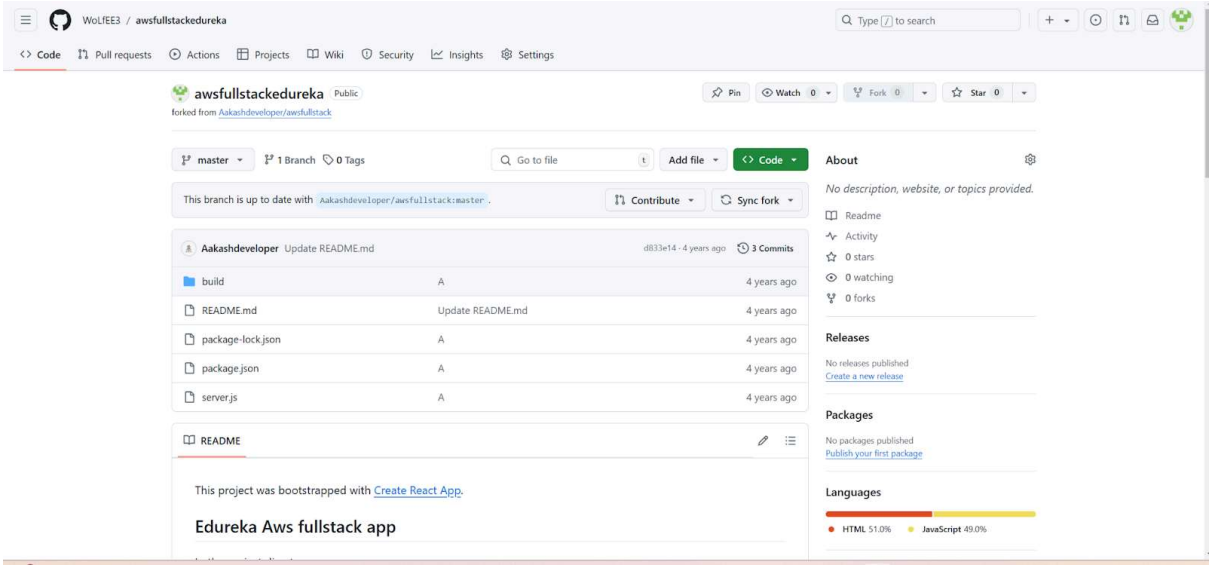
Now for CI/CD go to code pipe line

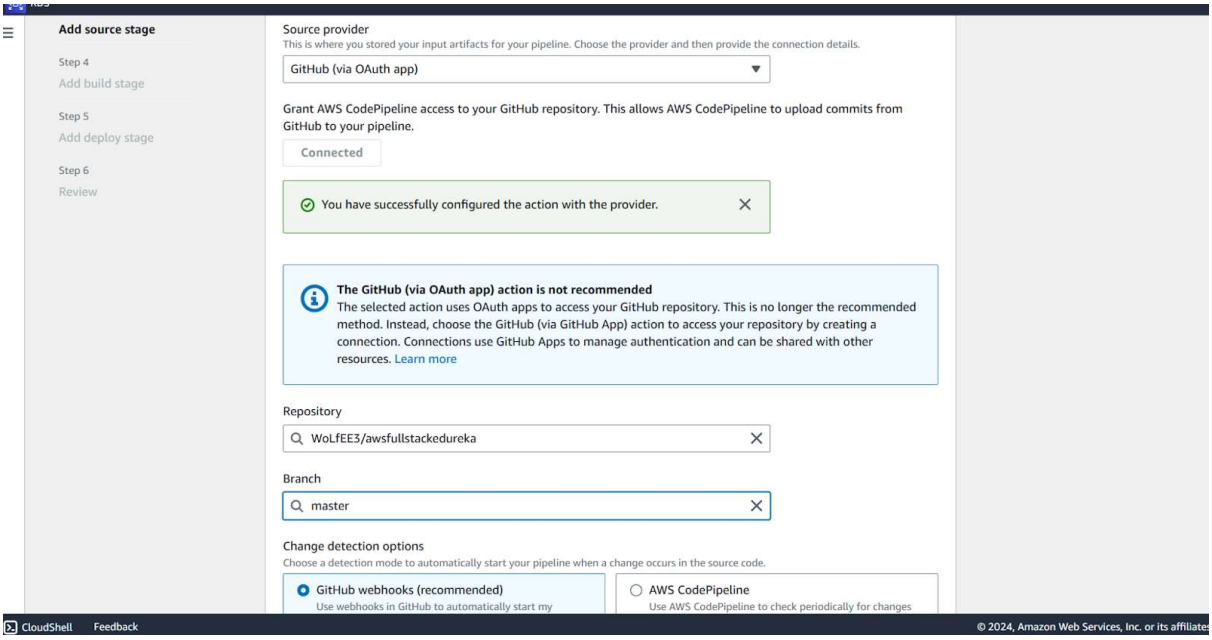


Step 8 Now in next step we will select the source from where we have to pick thee code and deploy

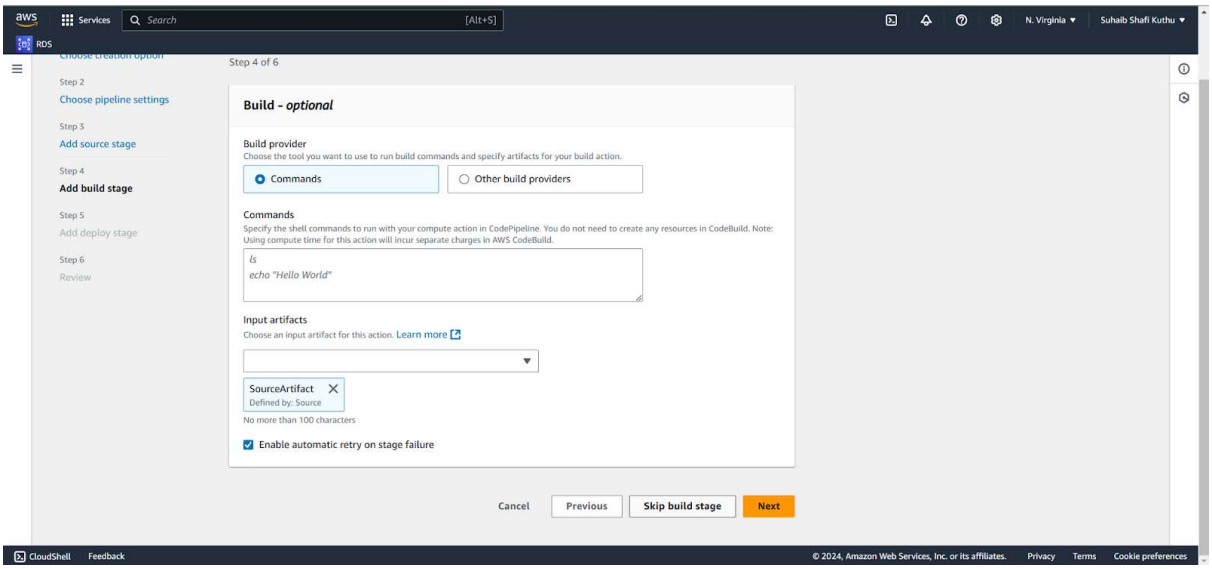


Step 9 Seletect the repo and branch used
The github branch



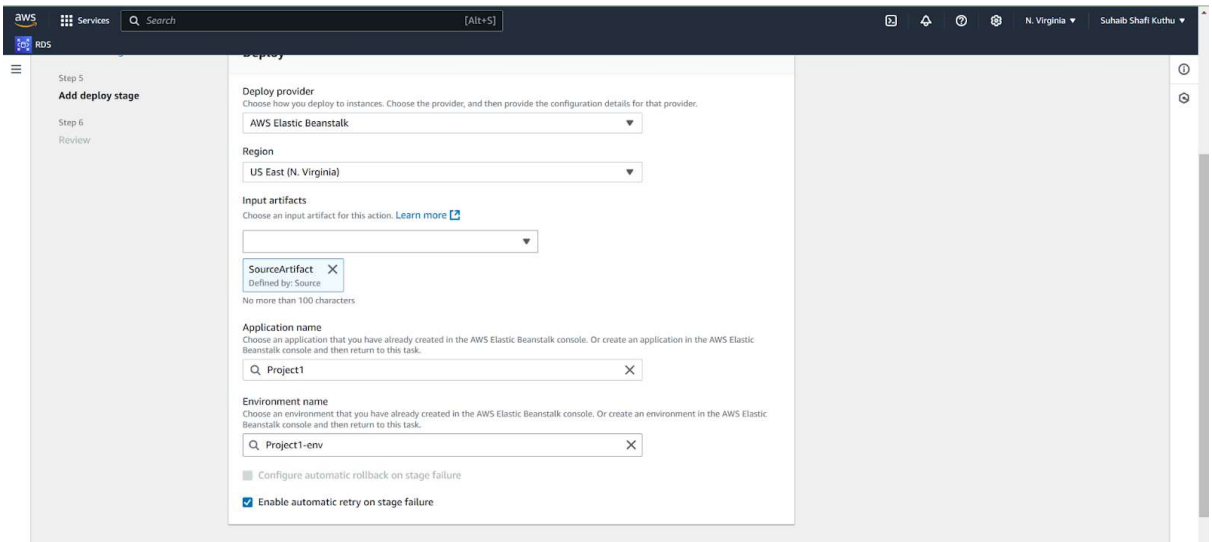


Step 10
For this demo we can skip the build stage of the application as it more like core script need to build app



Step 11

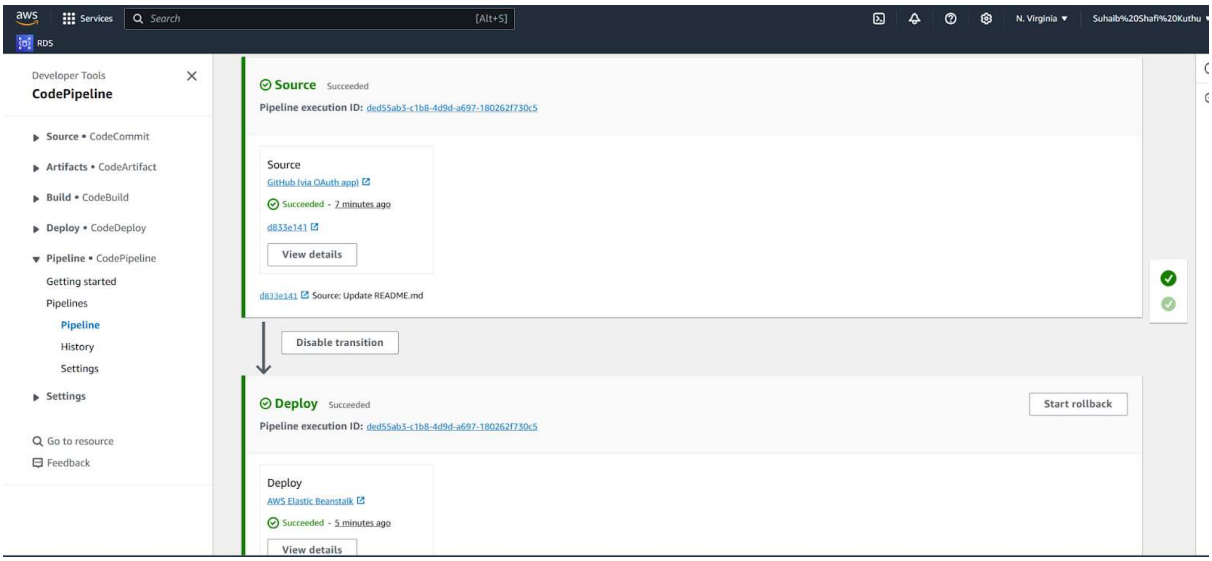
Now select the server where we want to deploy the application and i.e. beanstalk that we have launched



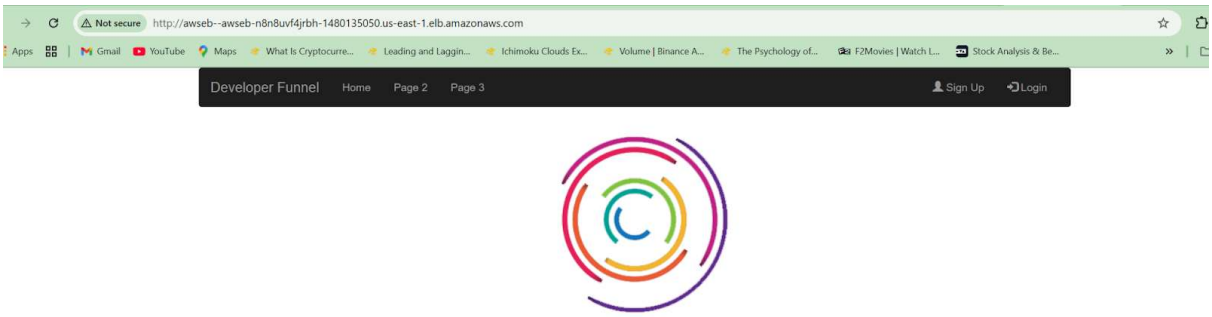
Step 12

Review the changes

Step 13 :-As soon as the code commit on the master branch this pipeline will execute and deploy code on the server.



As seen the app is deployed successfully

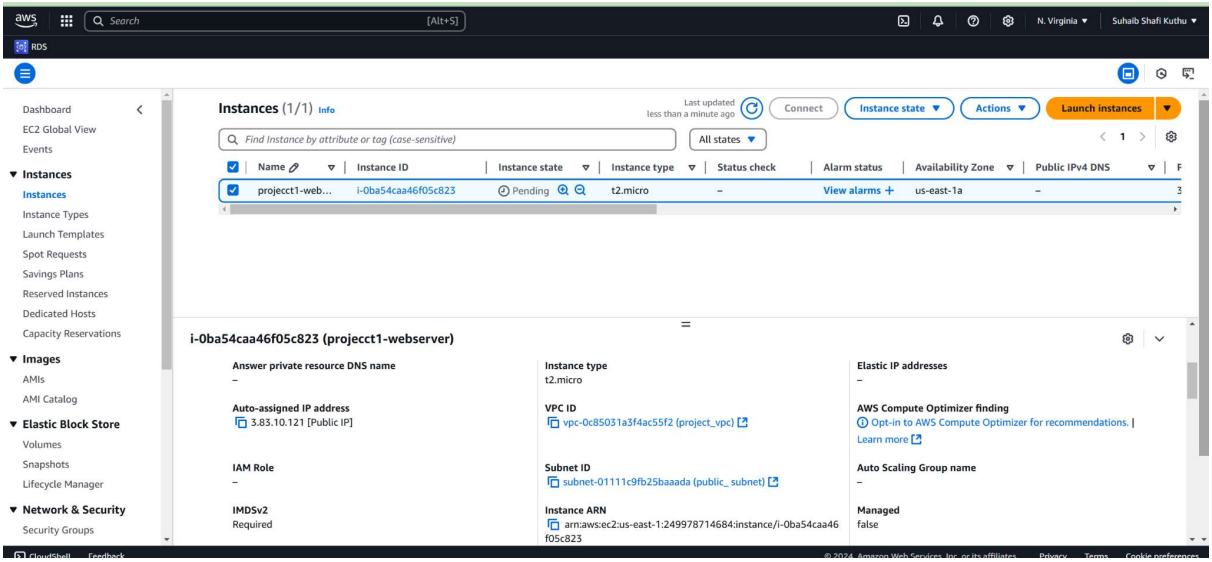


Now moving on to the backend part

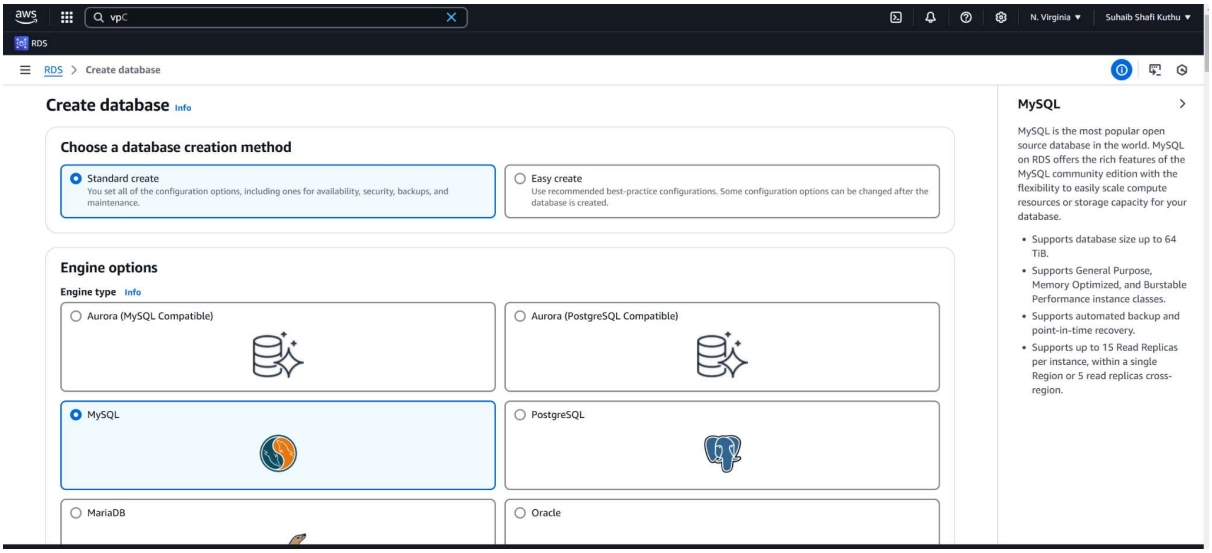
Step 1

Launch an ec2 instance in the public subnet

And a database in the private subnet so that it can only accessed by only the instance in public subnet ,as they are in the same vpc

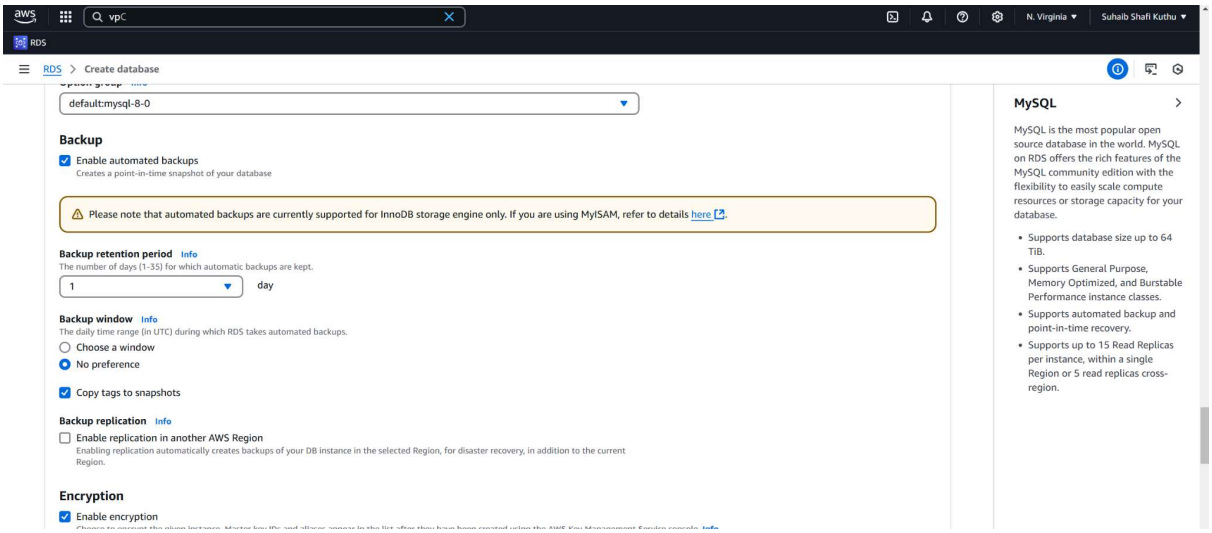


Now for db lauch go to RDS and launch a my sql database



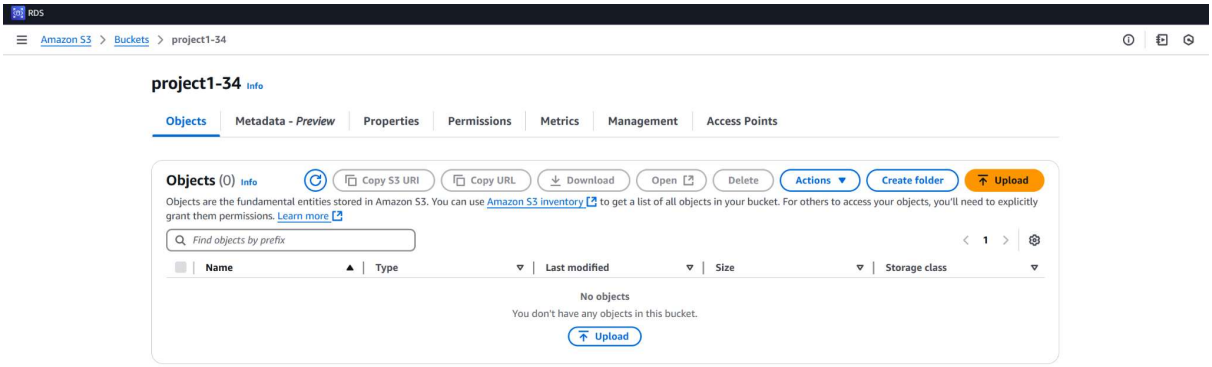
To retain the backup enable backups in database settings

Backups for your Web service may include creating snapshots of the instance or creating an Image of the Instance.

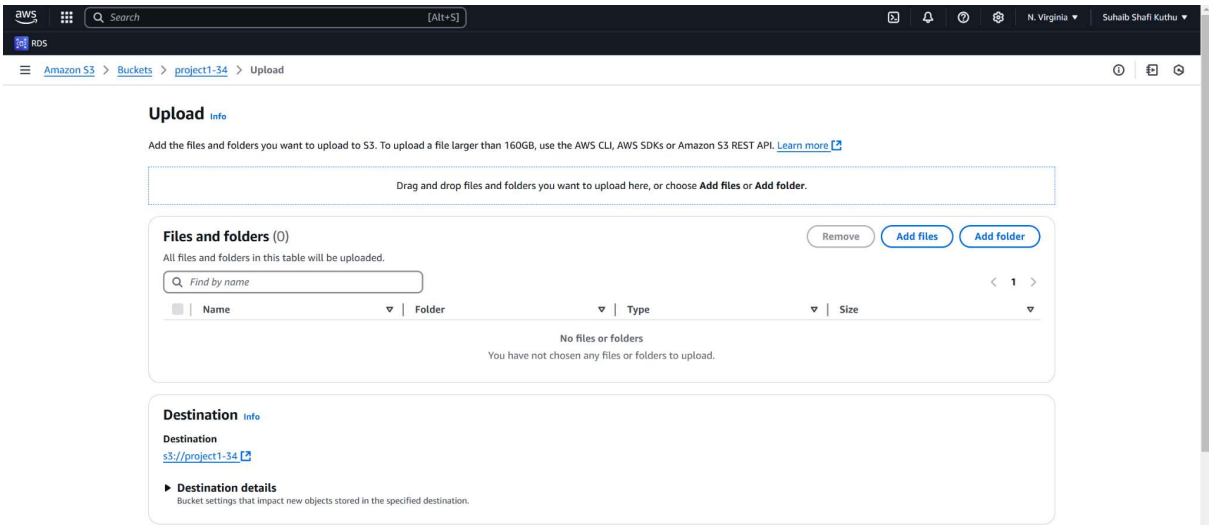


Ensure all the UI images served to the frontend application code are pr ovisioned via a secure storage unit

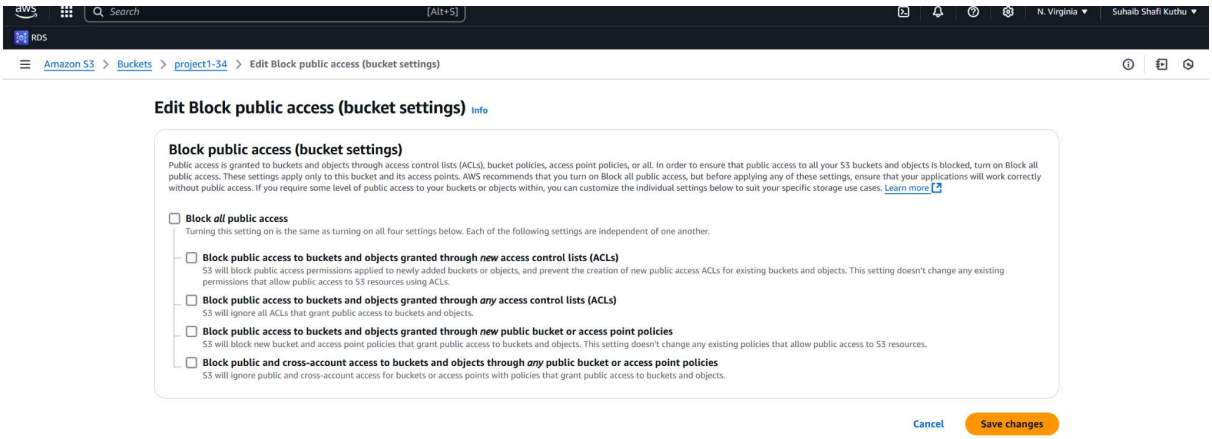
Create S3 Bucket



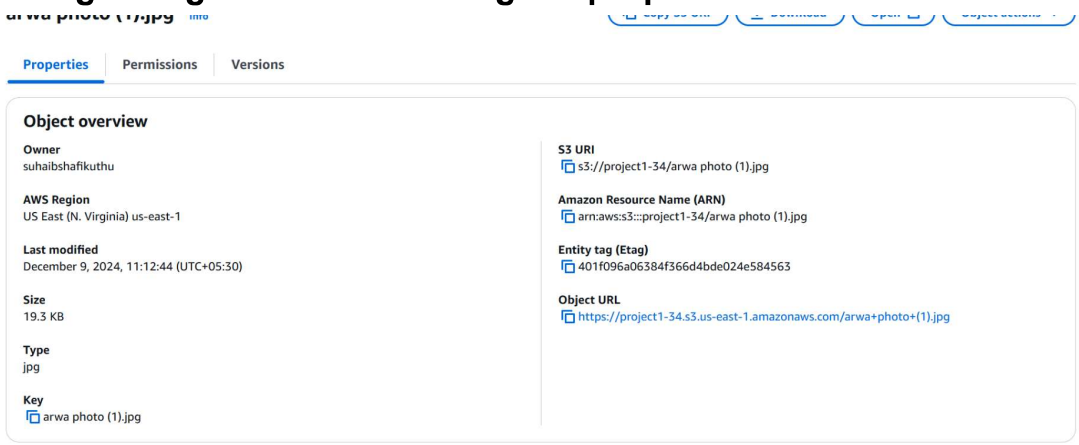
Upload files to s3 bucket



Make the Bucket Public So we can grab the images



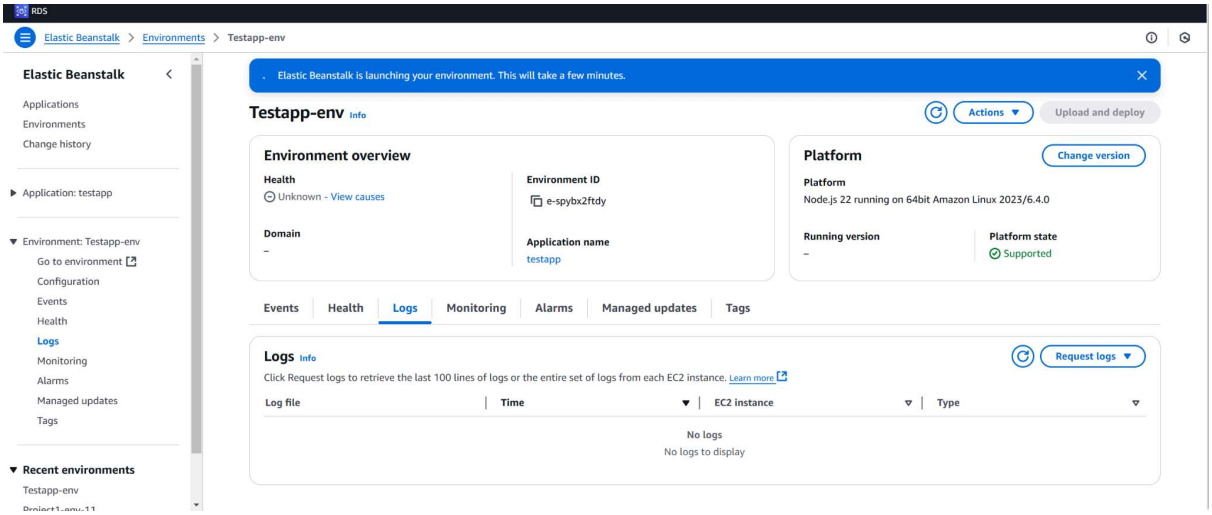
Find the URLs for your files by clicking on the bucket name, then selecting a single file and showing the properties



Automate the download all the activity logs into a CSV file, create a stream of data,
analyze it, and display it via a dashboard

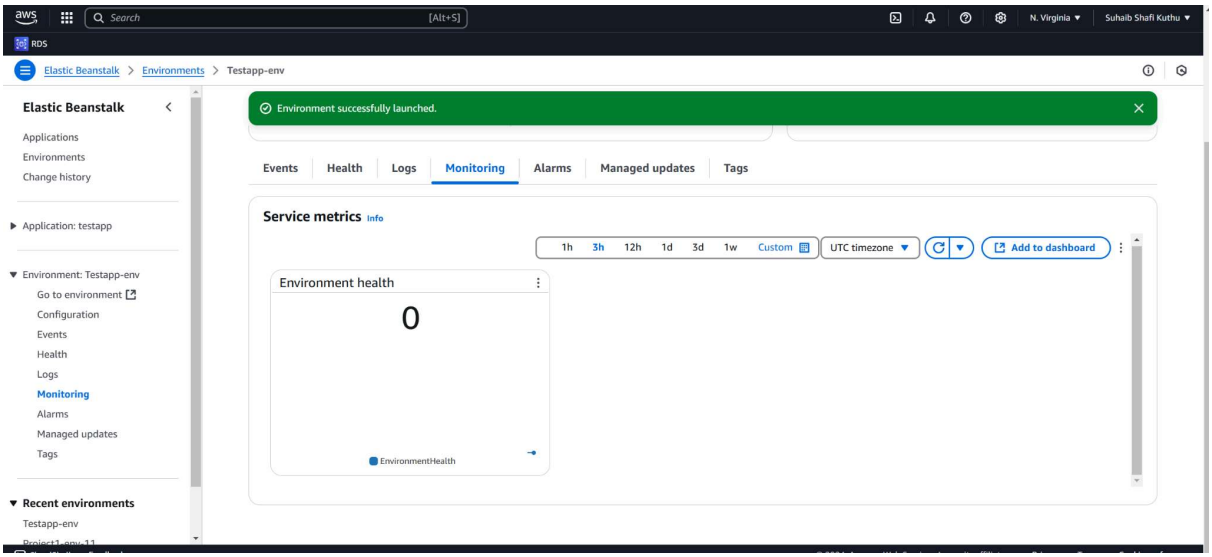
In the Elastic BeanStalk console, you will find the option to enable cloud watch, (Elastic Beanstalk-> Create application->configure more options). Now click on enable monitoring.

Then go to Cloudwatch and Create a dashboard to view and analyze the logs.



Now to display it on a dashboard

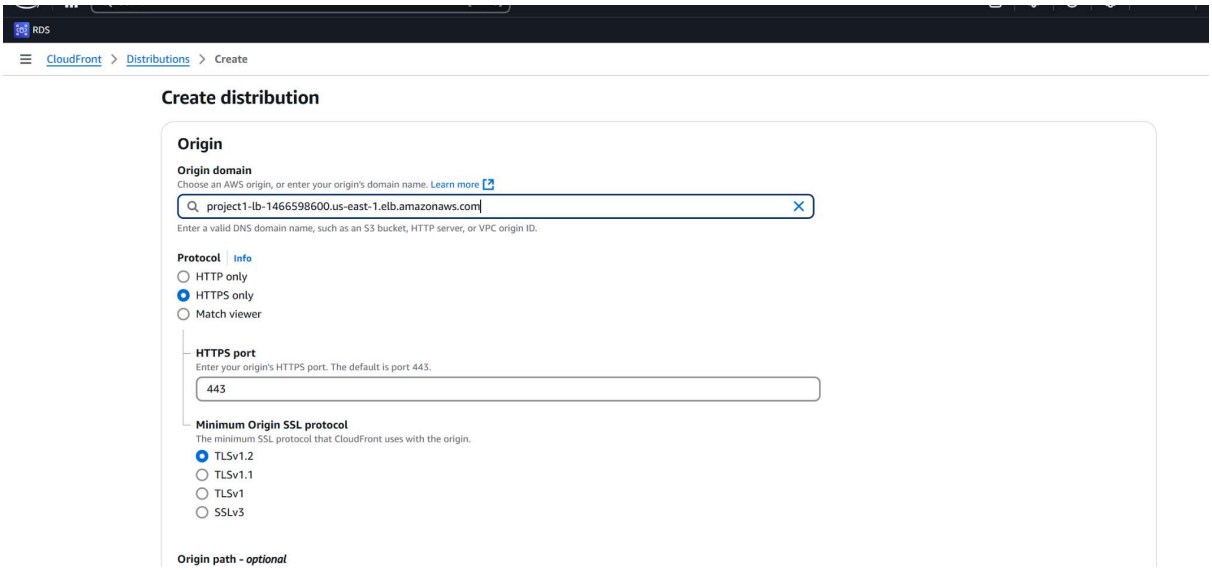
Go to monitoring and aad a dashboard



The Web application should also be cached globally, so users worldwide can access it with low latency

Create a CloudFront distribution to cache our application globally. Navigate to your CloudFront and click on create distribution Under web section click on get stared On the Create Distribution page,under

Origin Settings, choose the ELB that you created earlier Give the original path as project.php



Keep the rest as defaults and create a distribution Once your distribution is created, you will be able to see your website through your domain name of your distribution Check whether your infrastructure is working or not by terminating one of your instances, so that a new instance gets created by your auto scaling group

As elastic beanstalk allows us to create the load balancer and autoscaling group in their settings itself no separate creation of LB an Auto-scaling was shown.