Llenard C. Diama
IV-BCSAD

Assignment No. 2

What is AWS and it’s key benefits?

AWS or Amazon Web Services is a cloud computing platform offering on-demands IT resources like compute power, storage, and databases through the internet.

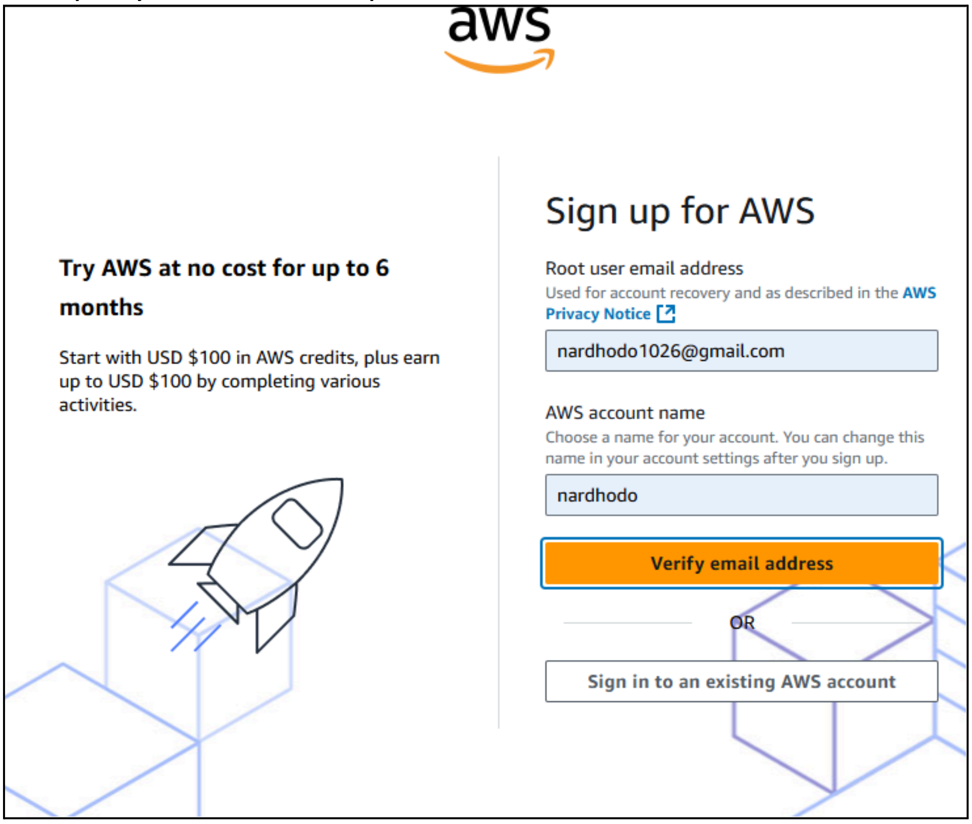
Key Benefits of AWS:

1. Cost Effective - AWS offers paying option for you to only pay for specific resources that you will utilize in your app projects.
2. Scalability - AWS features an Auto Scaling and Elastic Load Balancing services that allows you to scale your applications based on the demands.
3. Security - AWS provides strong security with compliance certifications, encryption, Identity and Access Management (IAM), etc.
4. Backup and Disaster Recovery - AWS provides affordable and efficient disaster recovery solutions with cross-region replications and backup storage.

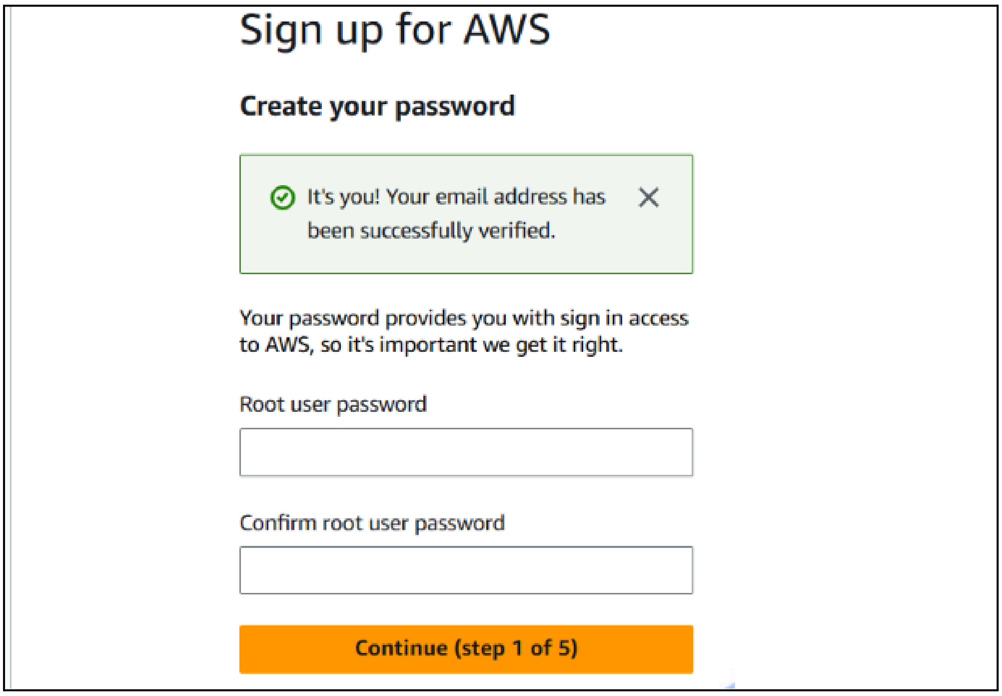
Signing up for AWS for Free

1. Go to https://signin.aws.amazon.com/signup?request_type=register to signup for an AWS Account

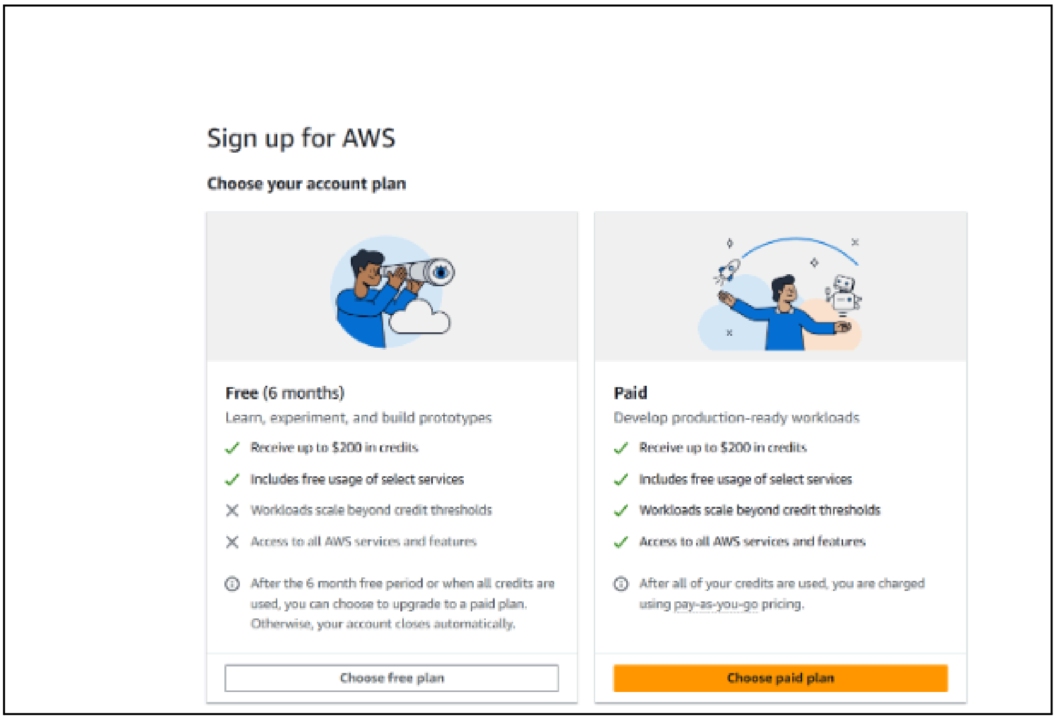
2. Input your email and preferred AWS account name.



3. Create a strong password

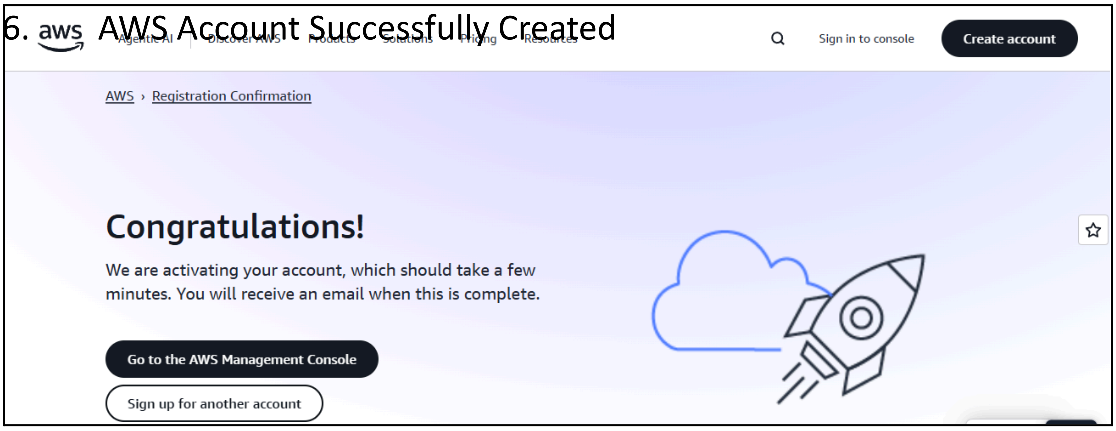


4. Choose a subscription plan (For this guide, it is recommended to pick the Free Plan)



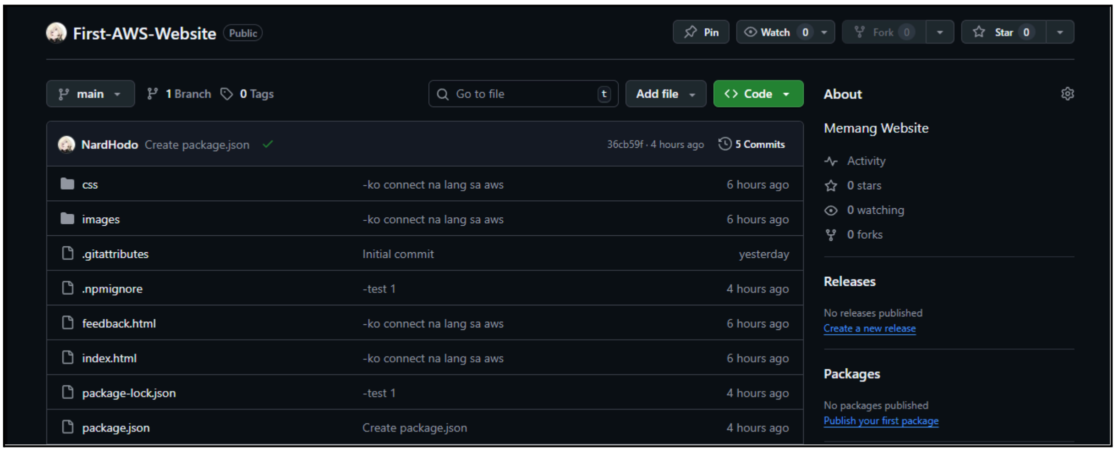
5. Enter Billing Information

6. AWS Account Successfully Created

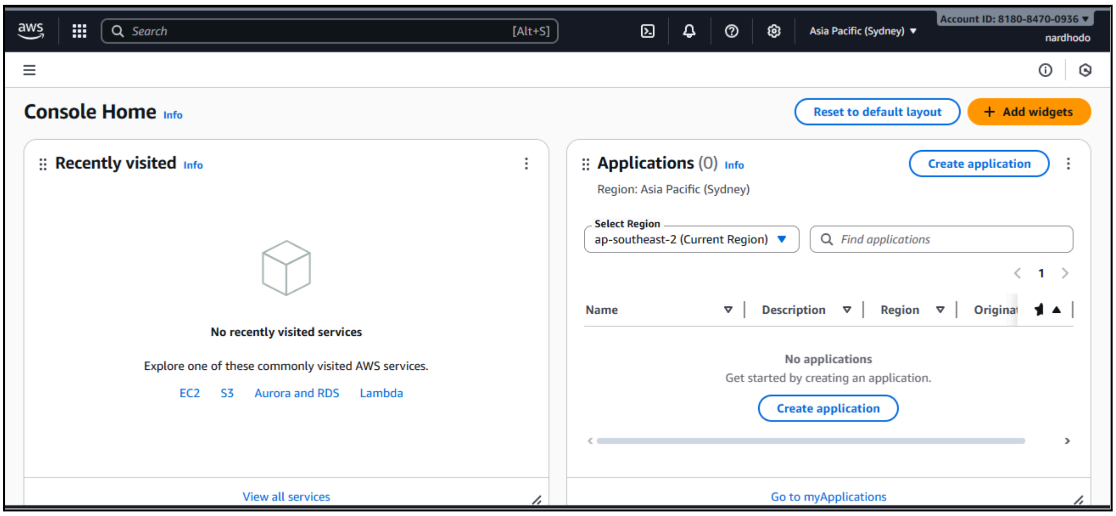


Setting a Goal: Hosting a Website on AWS

Sample website to use for this example:
https://github.com/NardHodo/First-AWS-Website

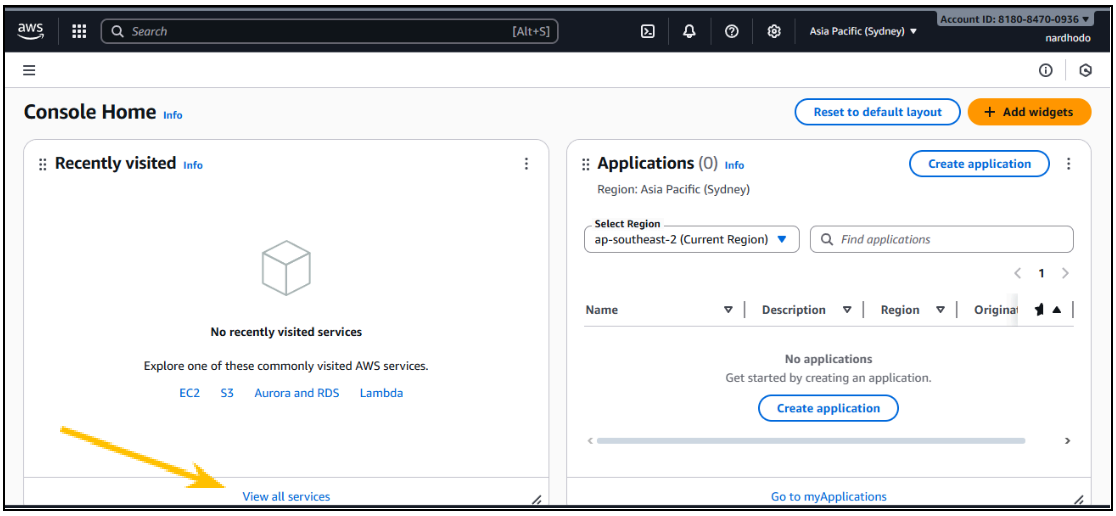


Exploring the AWS Management Console

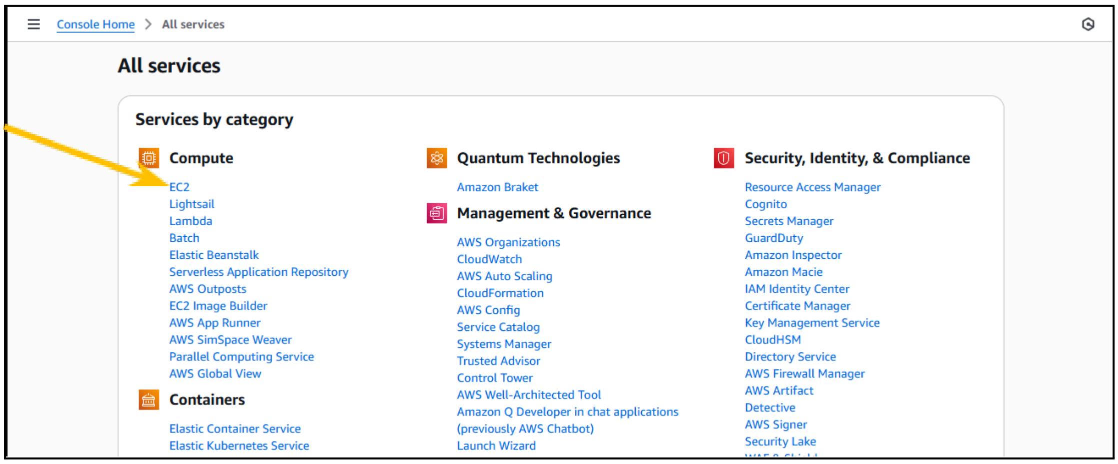


Launching a new EC2 Instance

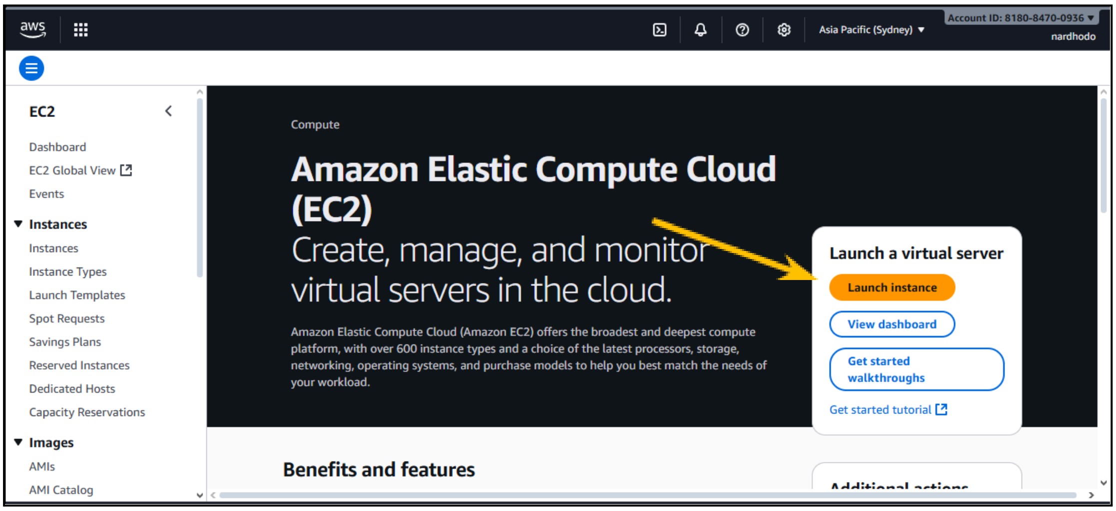
1. Scroll down within the console and search for “View All Services”.



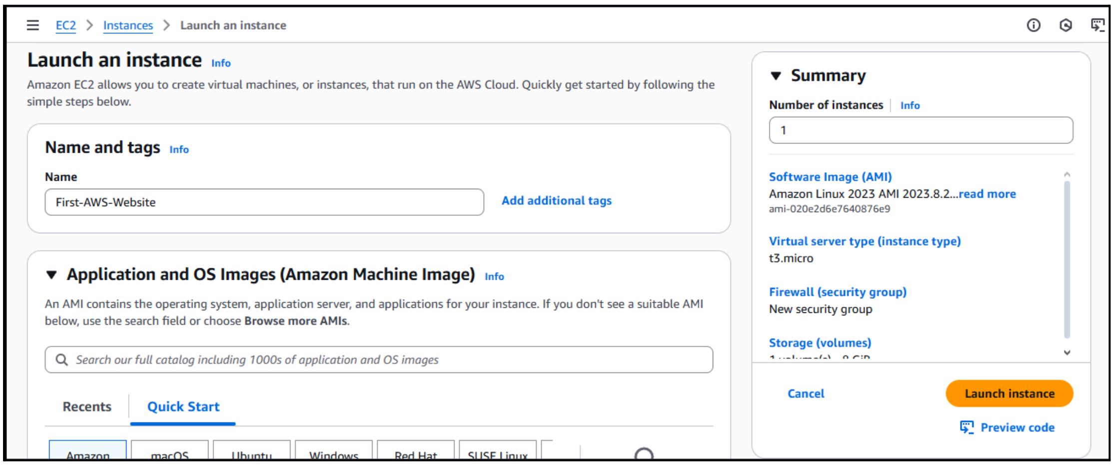
2. Select EC2



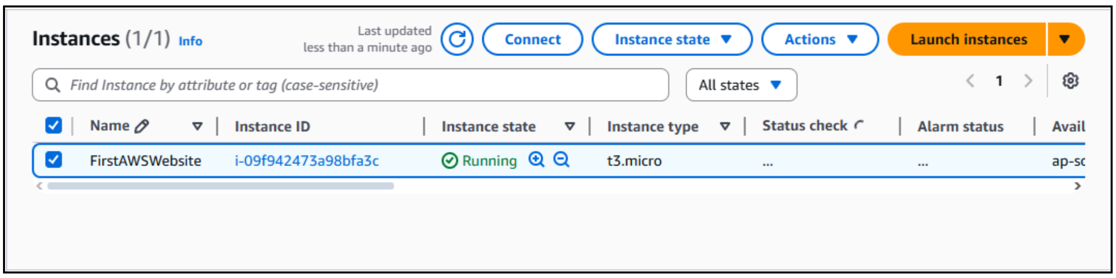
3. Click “Launch Instance”



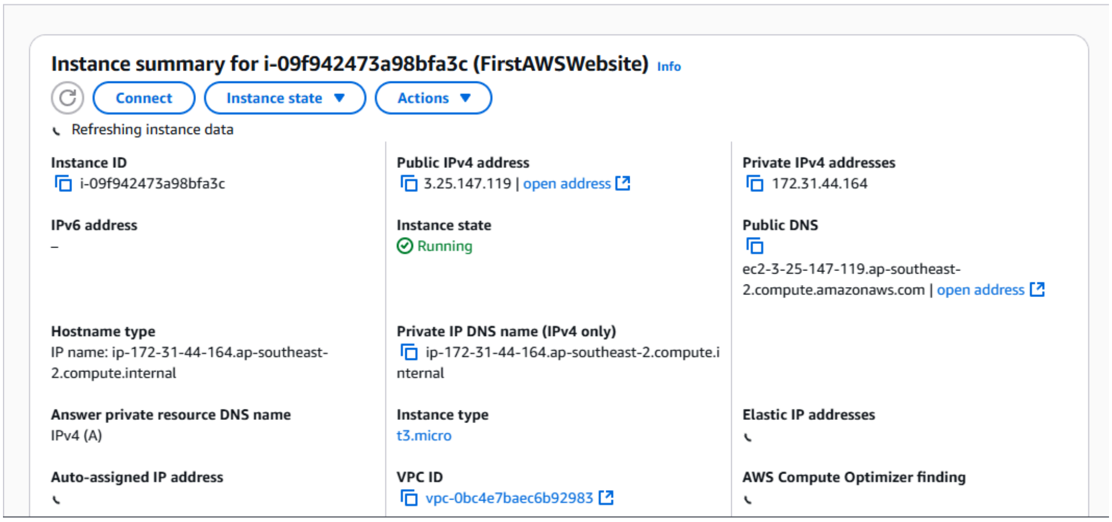
4. Provide a name for the instance.
5. Leave most of the settings by its default value.
6. Click “Launch Instance”.



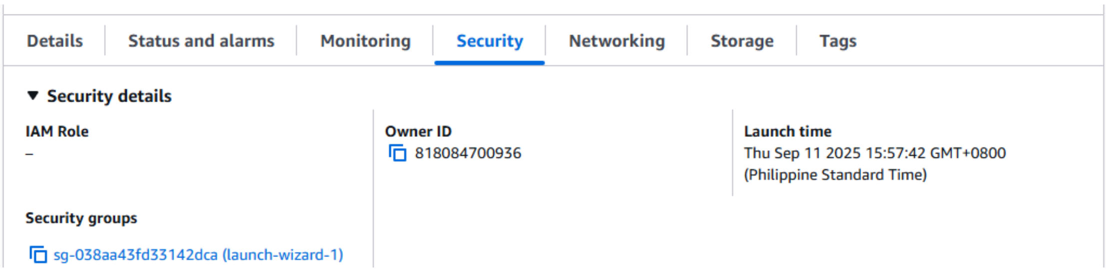
7. Once launched , you'll be able to view it on the Instances tab.
8. Select Instance



9. Scroll down and go to the Security Tab

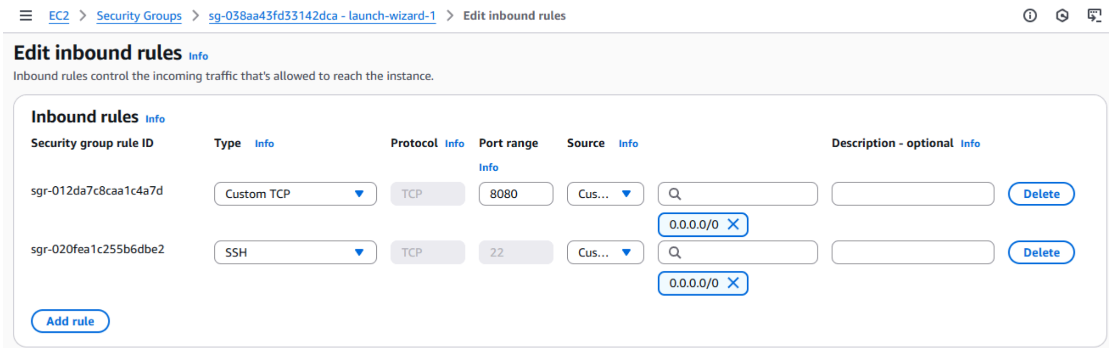


10. Click on the security group named “launch-wizard-1” to set up a new rule for http traffic.



11. Click on the “Edit Inbound Rules” button

12. Click “Add Rule” with the port number 8080 and choose the default range of IP addresses

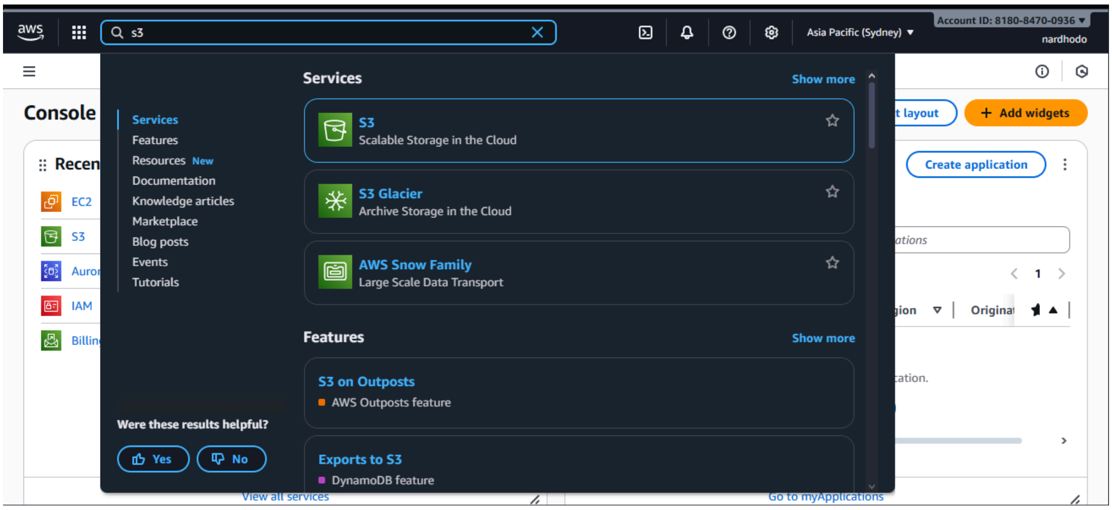


13. Click “Save Rules”
14. The EC2 Instance is fully configured and running.

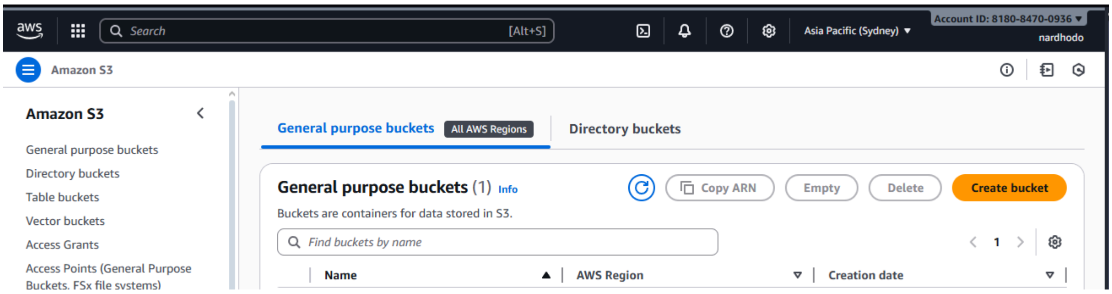
Configuring File Storage using S3 Buckets

A. Creating the Bucket

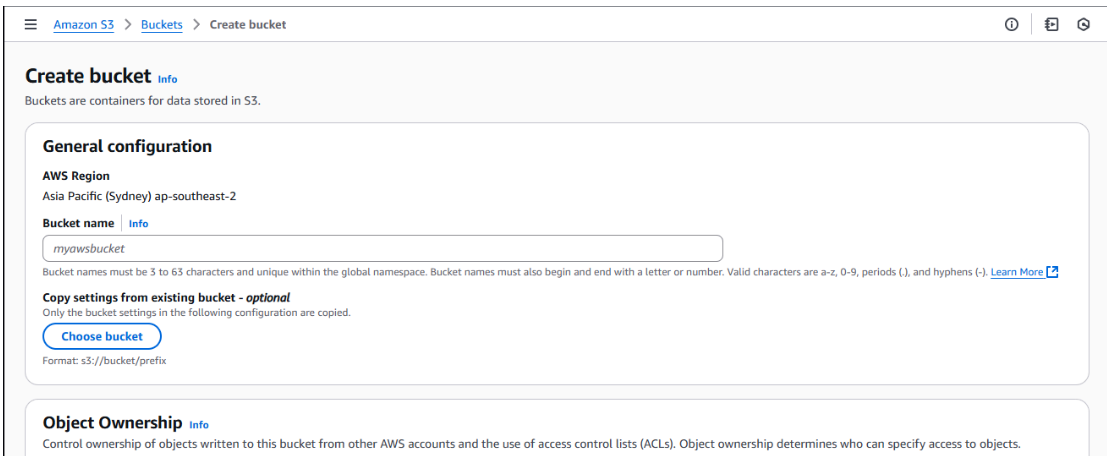
1. Click “View all Services” or you could search “S3” from the search bar above.



2. Click “Create a Bucket”

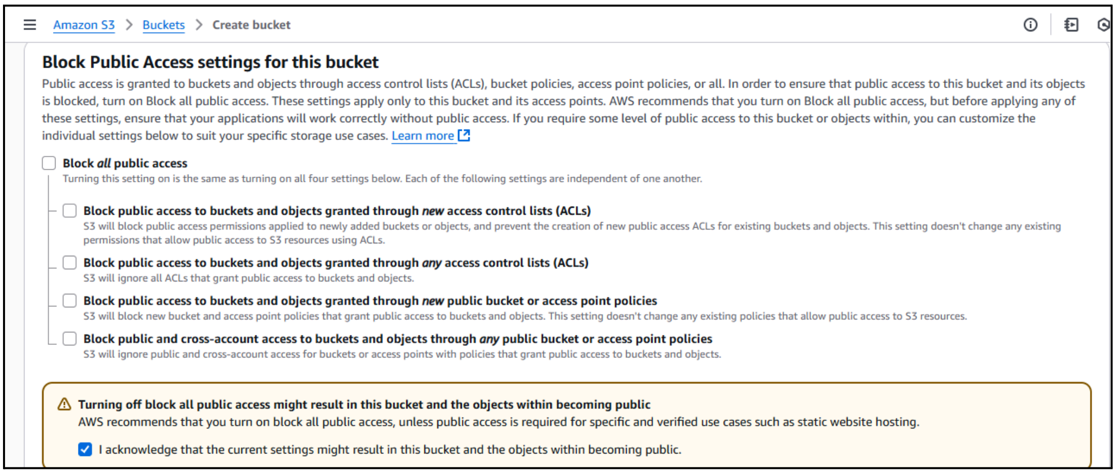


3. Setup Bucket Storage

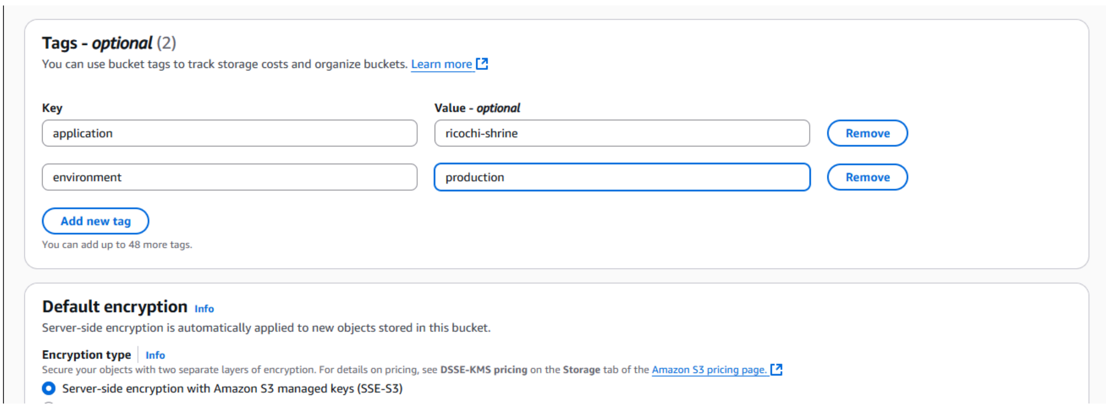


4. Uncheck the “Block all public access” checkbox. For this example, we need our bucket to be public so it could be accessed by everyone.

5. Acknowledge the settings you just modified.



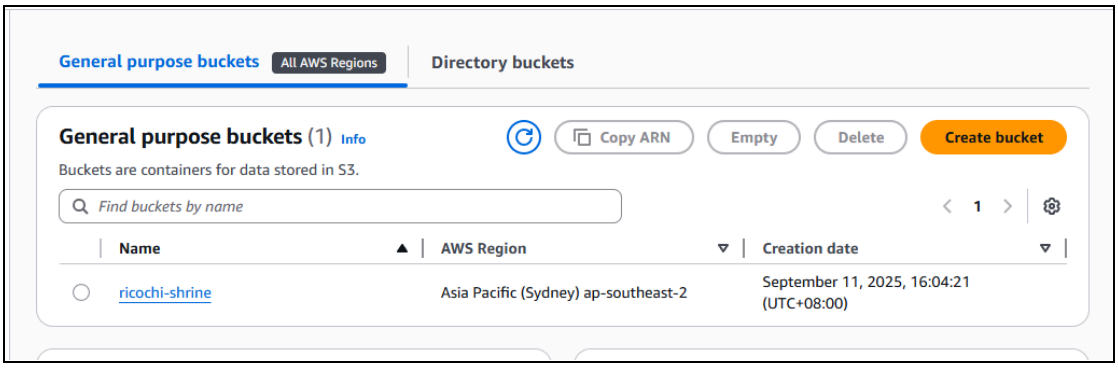
6. For this example, let’s add 2 bucket tags to make sure that the bucket is being used and our website is in production.



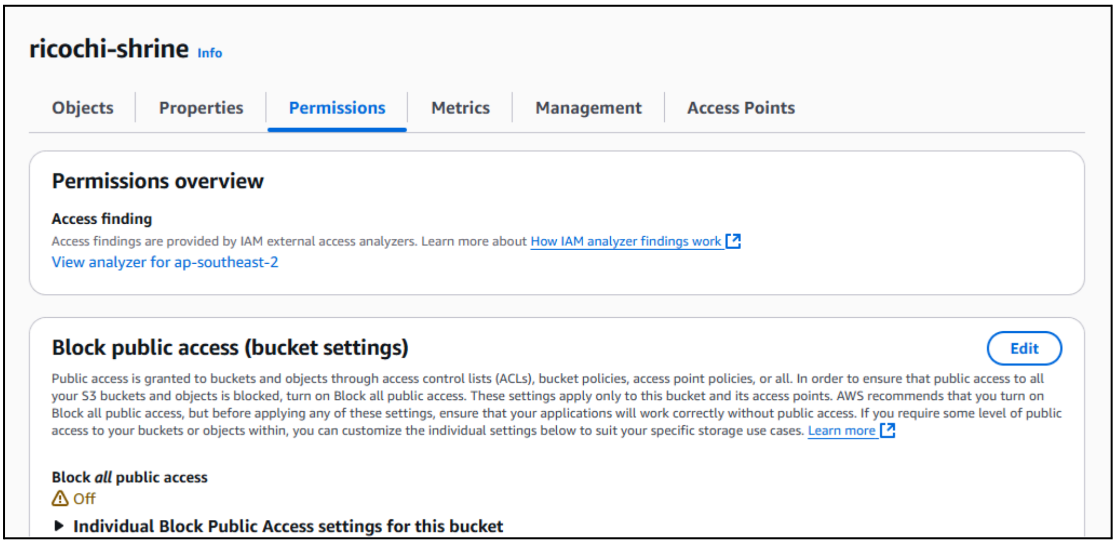
7. Leave other settings on default and click “Create Bucket”

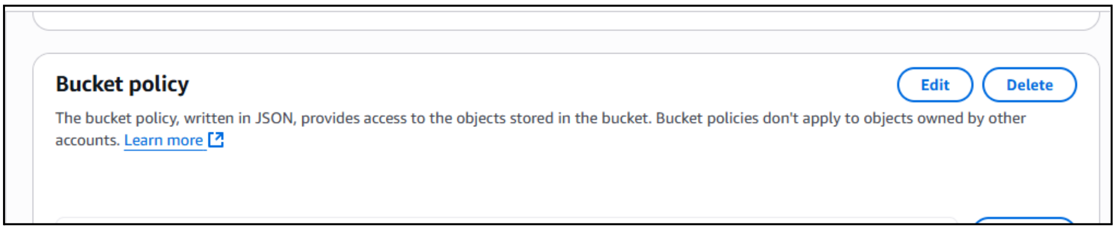
B. Modifying the Bucket

1. Click on the created bucket

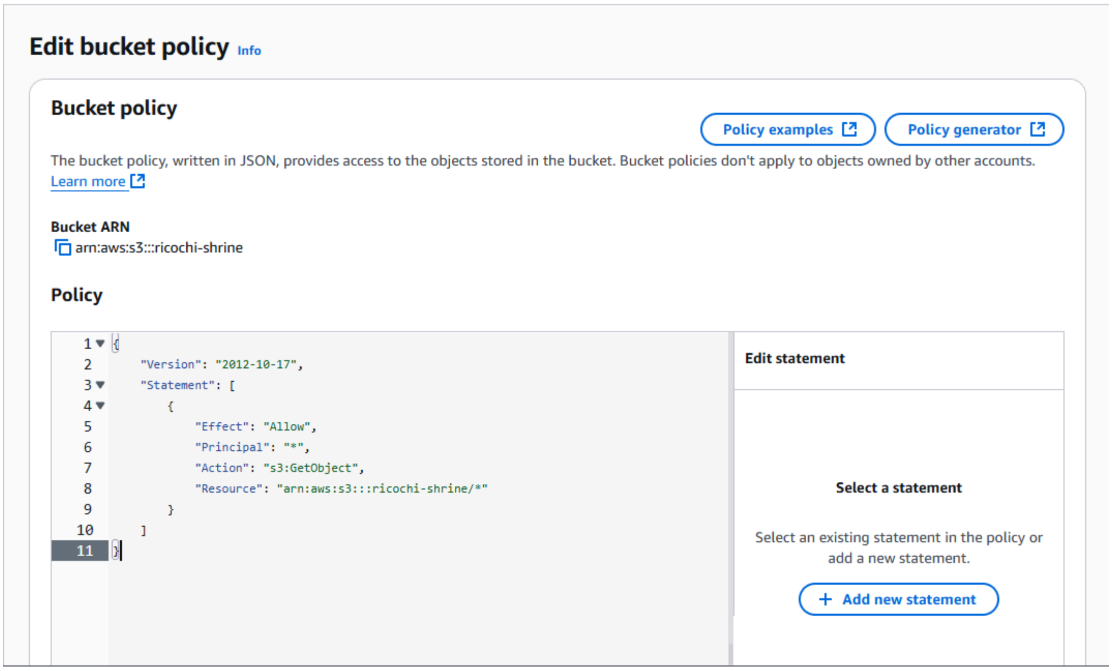


2. Go to the Permissions Tab and Select "Edit Bucket Policy"





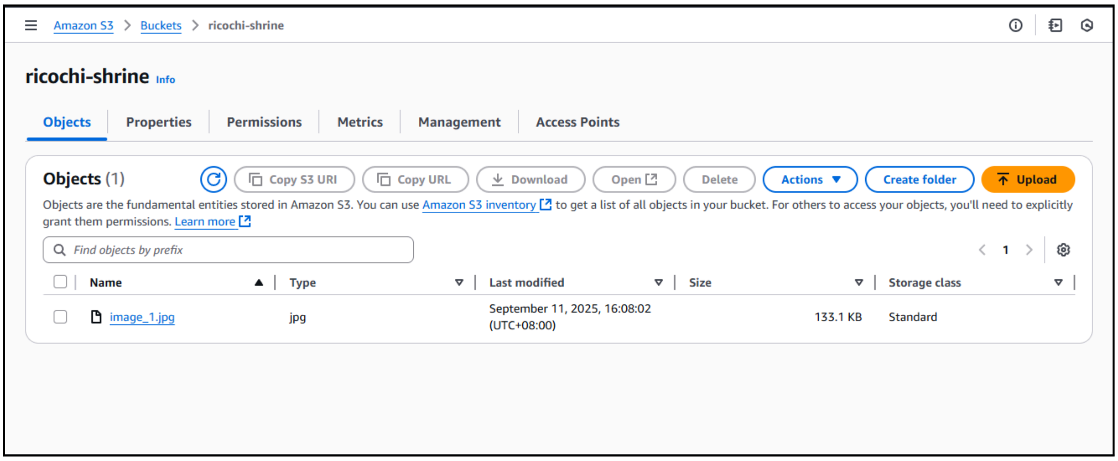
3. Enter the following code. Make sure that the name on the Resource is the same name as your bucket



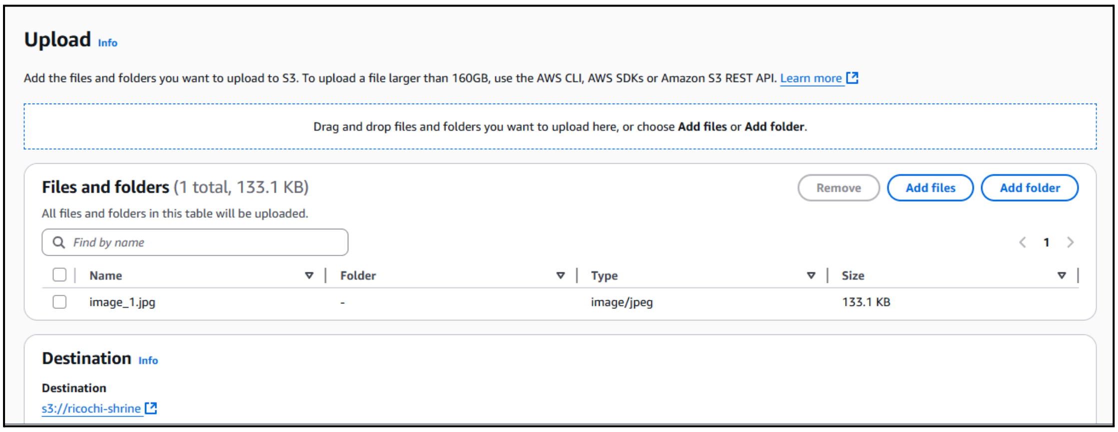
4. Click “Save Changes"

C. Uploading Files to the Bucket

1. Click the created bucket
2. Go to "Objects" Tab
3. Click "Upload"

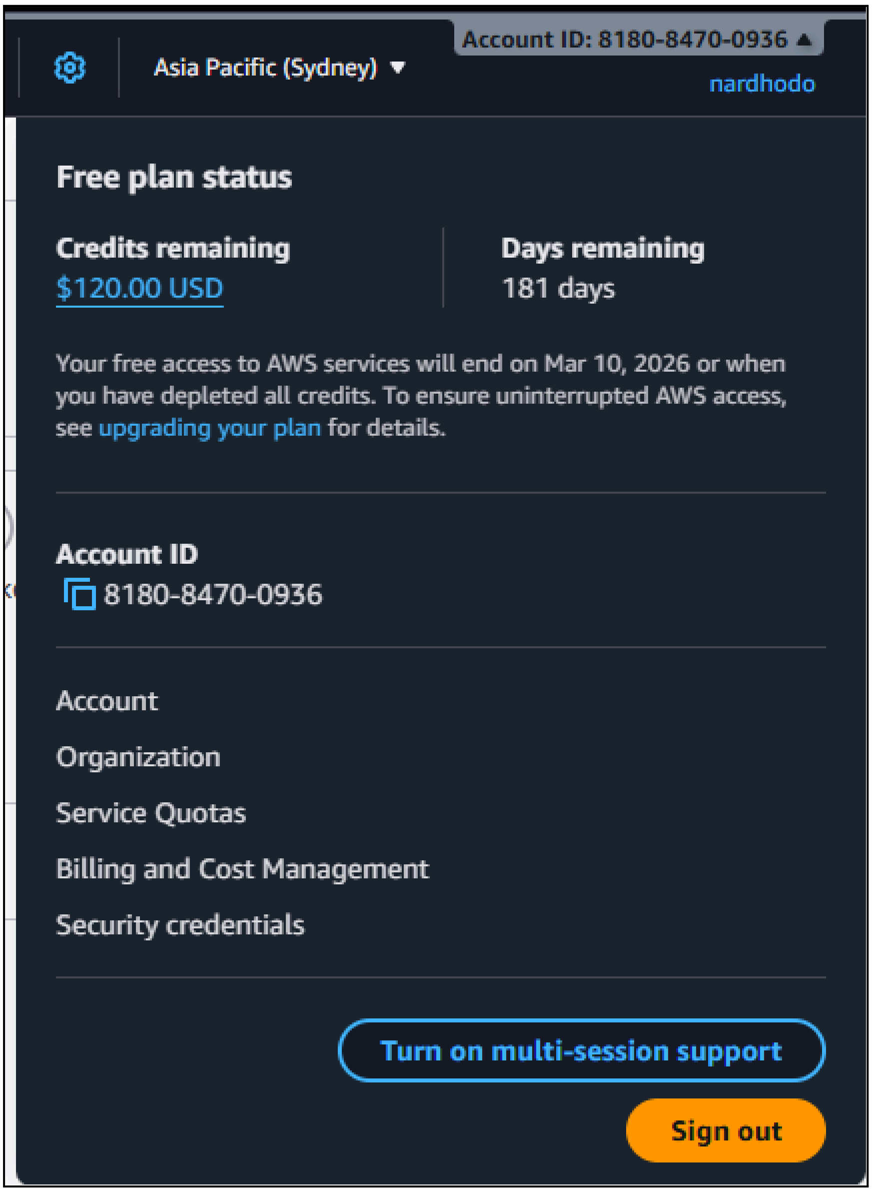


4. Click "Add Files"
5. Choose a file of your choice
6. Click "Upload"

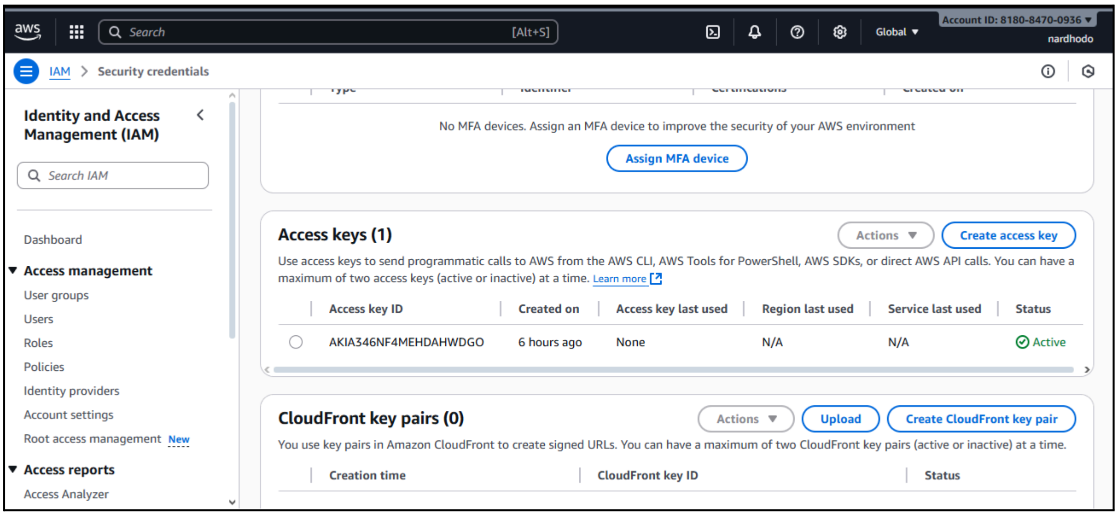


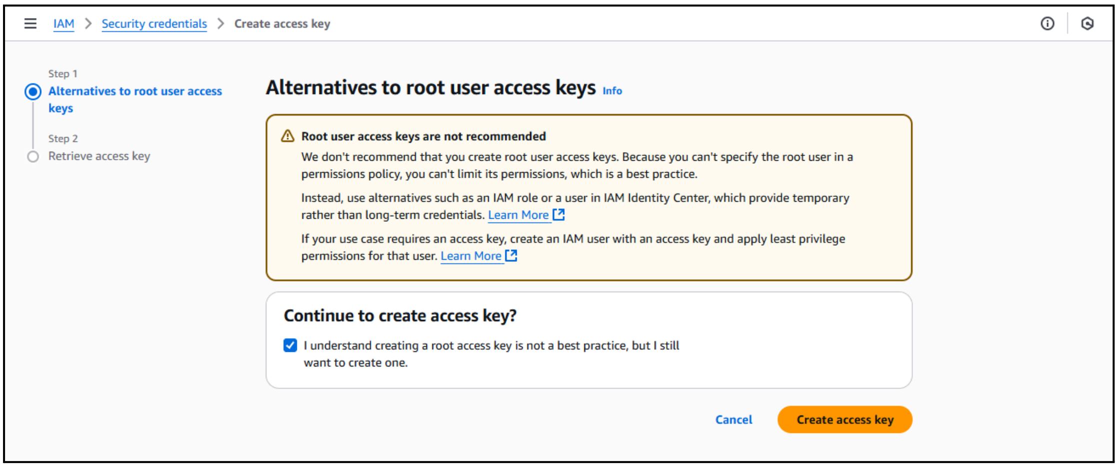
D. Obtaining Access Keys to access the Bucket

1. Click on the "Account Menu" on the top right of the screen and click "Security Credentials"



2. Scroll down to the “Access Keys” section
3. Click “Create Access Key”



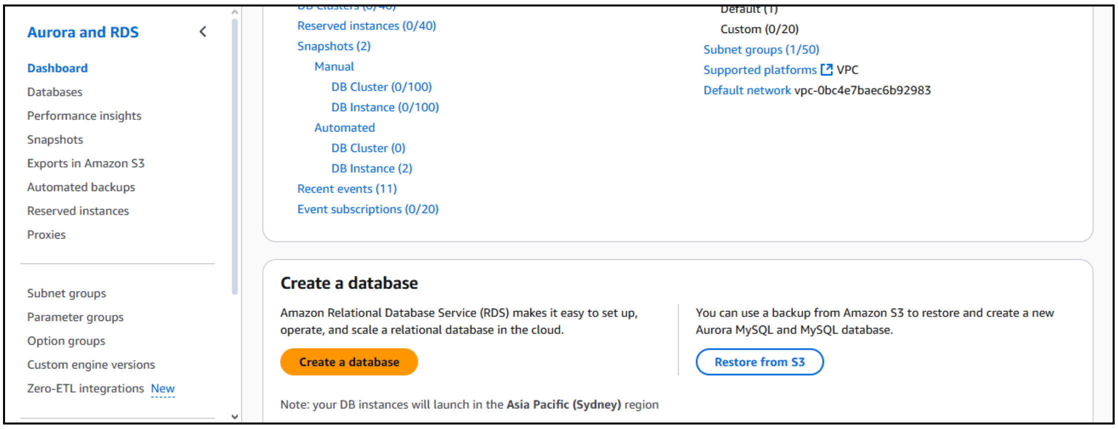


4. An Access key will be generated, make sure to copy and save it or download the csv file.

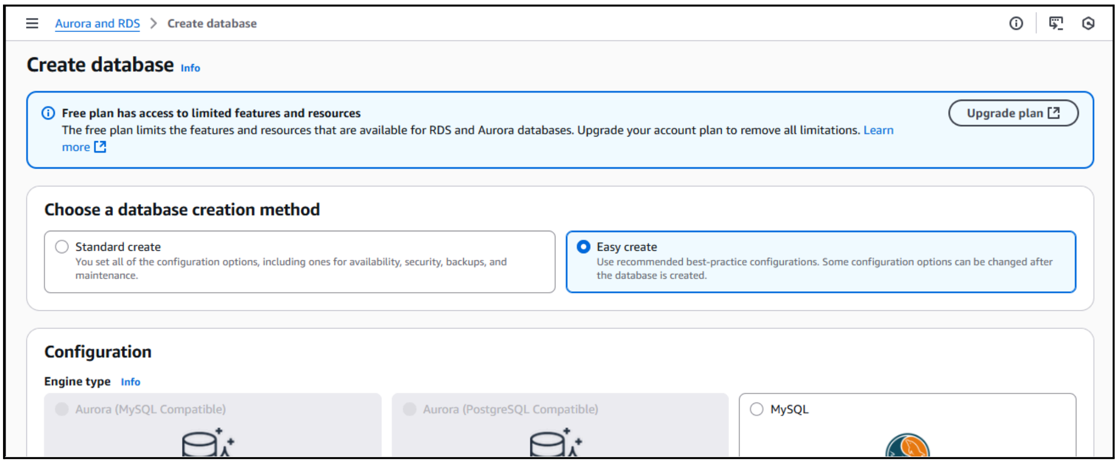
5. Click “Done”

Creating a Database Instance

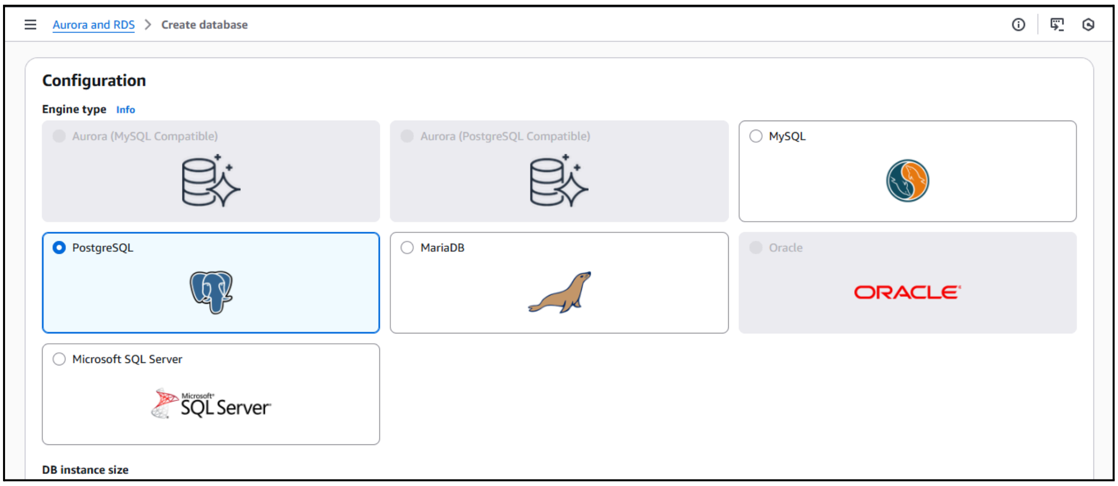
1. Search for Aurora and RDS service from the search bar.
2. Click “Create Database”

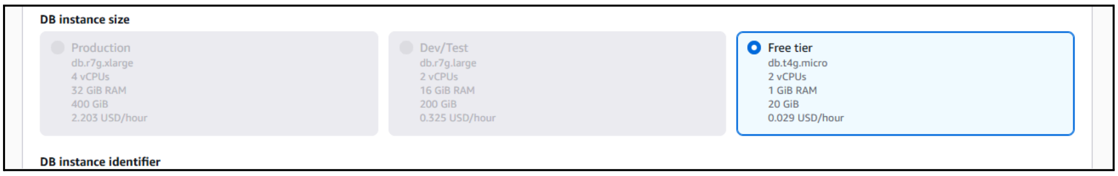


3. Setup Database
4. Choose “Easy Mode” to hide advanced database settings.

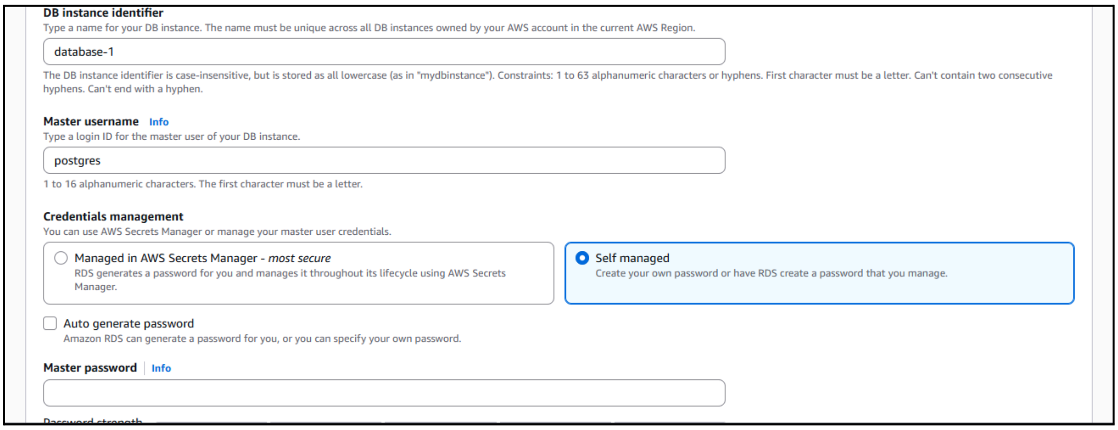


5. Choose “PostgreSQL” as database type and make sure that you are on “Free Tier”

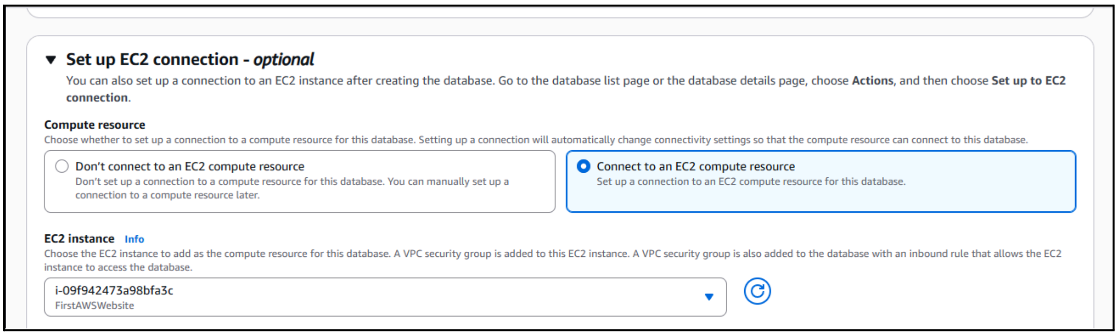




6. Set a name for your Database Instance. You could also set a username for your database or leave it as default. You could also set a password or let AWS generate a password for you.



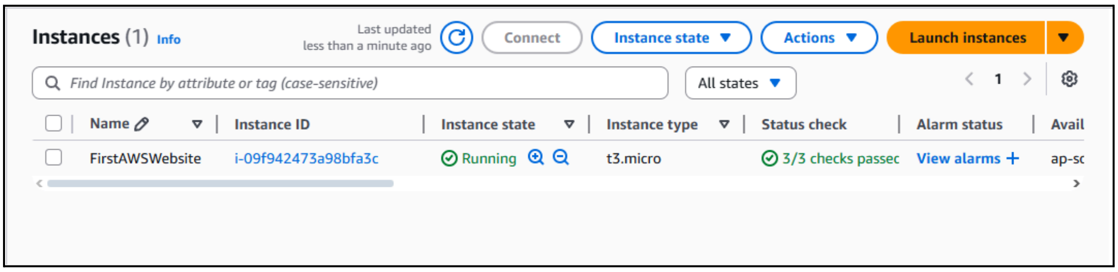
7. Scroll down and find “Setup EC2 Connection”
8. Expand the tab, and check the “Connect to an EC2 compute resource” option.
9. Choose the EC2 Instance that you’ve created earlier



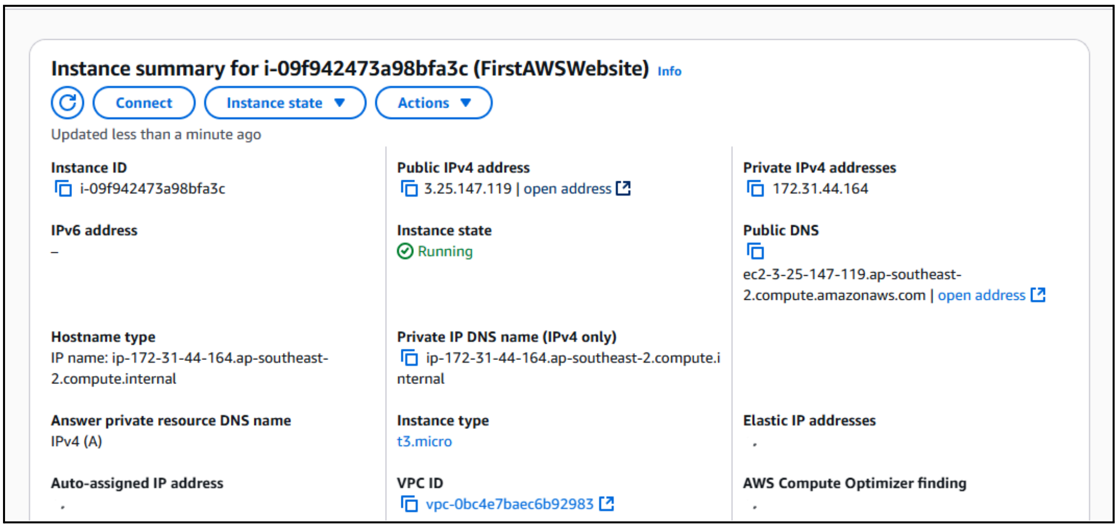
10. Click the “Create Database” button.
11. (If you decide to use a generated password) While the database is being created, click the “View Credentials” button at the top right to view the auto generated password.

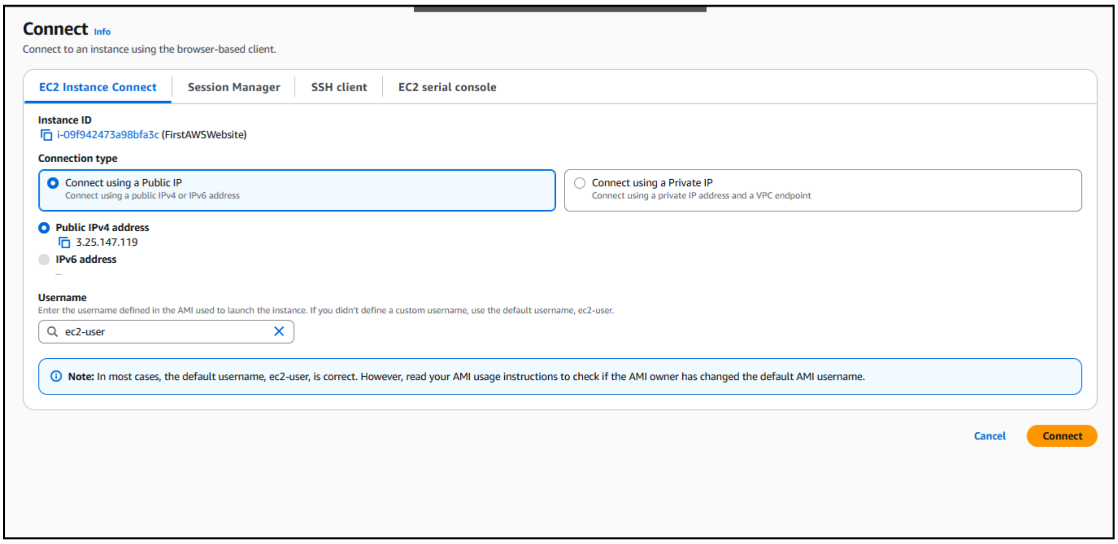
Hosting the Website by running Commands

1. Head back to the EC2 Instance you've created earlier.



2. Click on the instance
3. Click on the "Connect Button"





4. You will be directed to the terminal.

```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Thu Sep 11 10:36:05 2025 from 13.239.158.5
[ec2-user@ip-172-31-44-164 ~]$
```

i-09f942473a98bfa3c (FirstAWSWebsite)
PublicIPs: 3.25.147.119 PrivateIPs: 172.31.44.164

5. The website is hosted on Github, so we need to download the source code from there.

6. Enter the following commands:

    sudo dnf install git
    git clone https://github.com/NardHodo/First-AWS-Website

7. Set Environment variables for the website

8. Enter the following commands:
9. The website also requires nodejs so install it.

```
[ec2-user@ip-172-31-44-164 ~]$
[ec2-user@ip-172-31-44-164 ~]$ git clone https://github.com/NardHodo/First-AWS-Website
fatal: destination path 'First-AWS-Website' already exists and is not an empty directory.
[ec2-user@ip-172-31-44-164 ~]$ export S3_BUCKET=ricochi-shrine
[ec2-user@ip-172-31-44-164 ~]$ export S3_REGION=ap-southeast-2
[ec2-user@ip-172-31-44-164 ~]$ export S3_ACCESS_KEY=AKIA346NF4MEHDAHWDGO
[ec2-user@ip-172-31-44-164 ~]$ export S3_SECRET_KEY=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
[ec2-user@ip-172-31-44-164 ~]$ export DB_HOST=ricochi-shrine.czwg800qstir.ap-southeast-2.rds.amazonaws.com
[ec2-user@ip-172-31-44-164 ~]$ export DB_USER=postgres
[ec2-user@ip-172-31-44-164 ~]$ export DB_PASS=imZQBQAl7ZYhKs7FsAid
[ec2-user@ip-172-31-44-164 ~]$ sudo dnf install nodejs
```

10. Install dependencies
11. Input the command ‘npm install’

12. Enter the following command to start the website in the background and make sure it’s still running after you exit the terminal.

```
Complete!
[ec2-user@ip-172-31-44-164 First-AWS-Website]$ npm install

up to date, audited 1 package in 388ms

found 0 vulnerabilities
[ec2-user@ip-172-31-44-164 First-AWS-Website]$ npm start &
```

13. Now that the process is running, enter the command “disown” to disconnect the job from the session. This makes sure that the process still runs after you terminate the session.

Website Address

3.25.147.119:8080

Tips for Cost Savings in AWS

1. Use AWS Cost Explorer to analyze underutilized resources and switch to smaller instance types if possible.
2. Set cost and usage budgets. Get email or SNS alerts if you’re nearing limits.
3. Share resources like Reserved Instances across multiple accounts.
4. Use Free Tier and Credits. Most AWS Services have a 12-Month Free Tier.