

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.

Try AWS at no cost for up to 6 months

Start with USD \$100 in AWS credits, plus earn up to USD \$100 by completing various activities.

aws

Sign up for AWS

Root user email address
Used for account recovery and as described in the [AWS Privacy Notice](#)

nardhodo1026@gmail.com

AWS account name
Choose a name for your account. You can change this name in your account settings after you sign up.

nardhodo

Verify email address

OR

Sign in to an existing AWS account

3. Create a strong password

Sign up for AWS

Create your password

It's you! Your email address has been successfully verified.

Your password provides you with sign in access to AWS, so it's important we get it right.

Root user password

Confirm root user password

Continue (step 1 of 5)

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

Sign up for AWS

Choose your account plan



Free (6 months)
Learn, experiment, and build prototypes

- ✓ Receive up to \$200 in credits
- ✓ Includes free usage of select services
- ✗ Workloads scale beyond credit thresholds
- ✗ Access to all AWS services and features

ⓘ After the 6 month free period or when all credits are used, you can choose to upgrade to a paid plan. Otherwise, your account closes automatically.

[Choose free plan](#)



Paid
Develop production-ready workloads

- ✓ Receive up to \$200 in credits
- ✓ Includes free usage of select services
- ✓ Workloads scale beyond credit thresholds
- ✓ Access to all AWS services and features

ⓘ After all of your credits are used, you are charged using pay-as-you-go pricing.

[Choose paid plan](#)

5. Enter Billing Information

Sign up for AWS

Billing Information

Billing country
Your billing country determines the payment methods available to you to pay for AWS services.

Philippines ▾

Credit or Debit card number

AWS accepts most major credit and debit cards. To learn more about payment options, review our [FAQ](#).

Expiration date

Month ▾ Year ▾

Security code ⓘ

Cardholder's name

6. aws AWS Account Successfully Created

AWS Home | Services | Products | Solutions | Training | Resources

Sign in to console Create account

AWS > Registration Confirmation

Congratulations!

We are activating your account, which should take a few minutes. You will receive an email when this is complete.

[Go to the AWS Management Console](#)

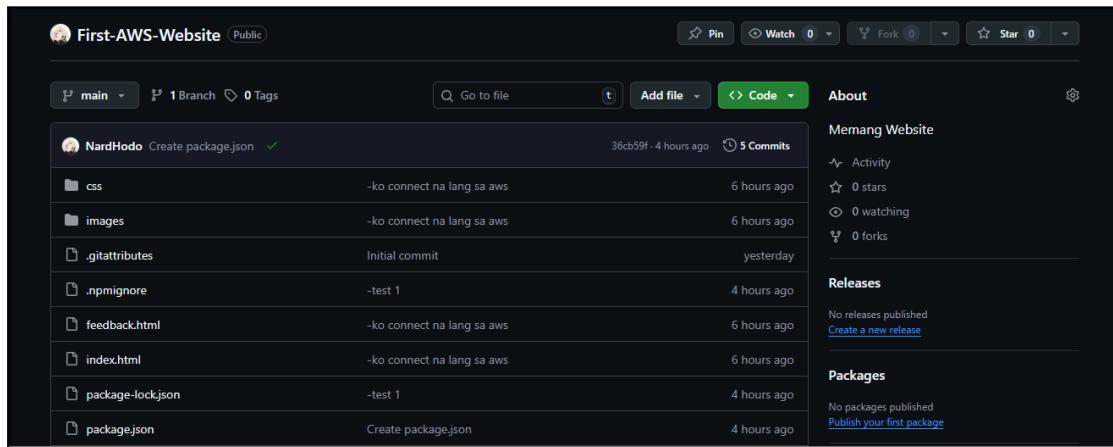
[Sign up for another account](#)



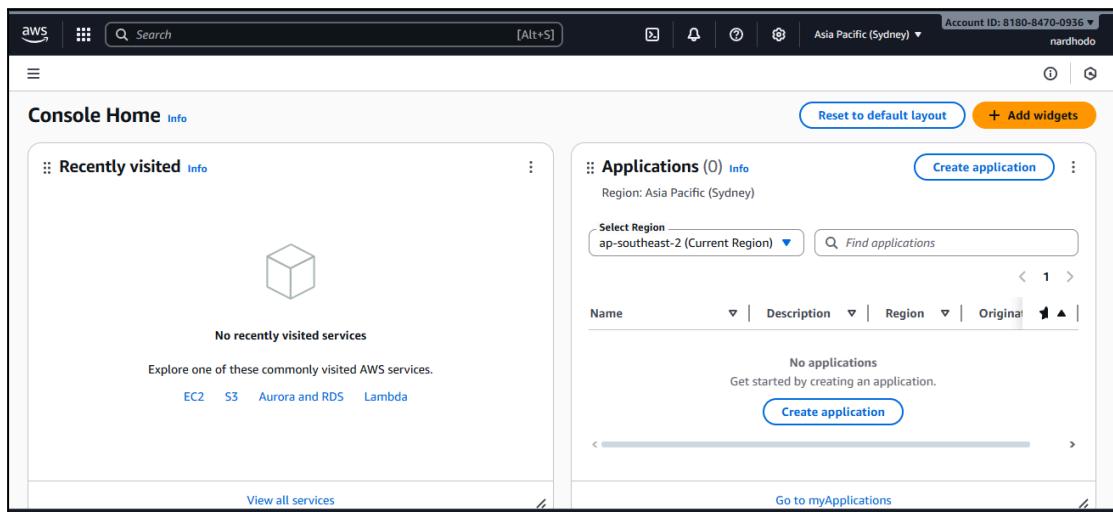
☆

Setting a Goal: Hosting a Website on AWS

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

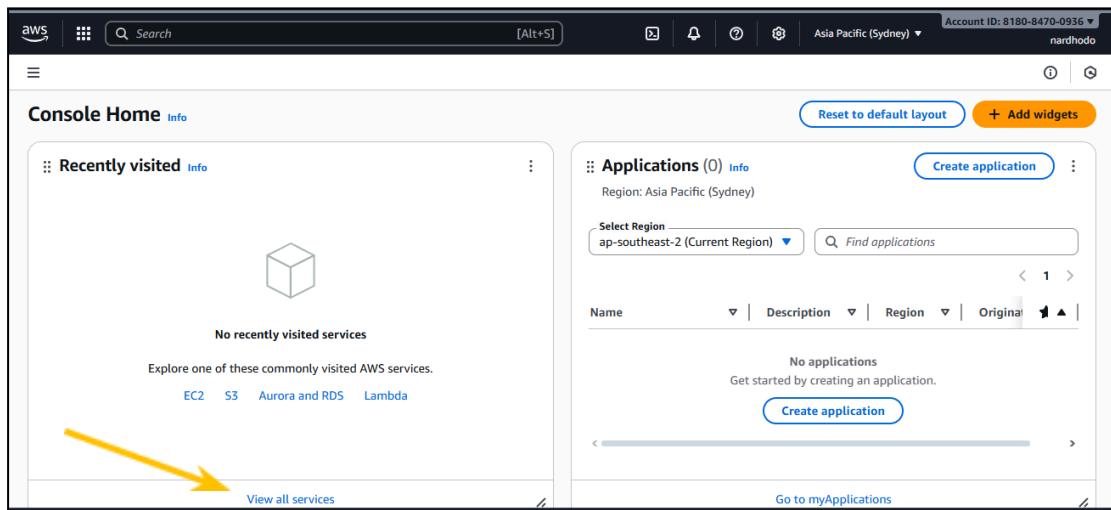


Exploring the AWS Management Console

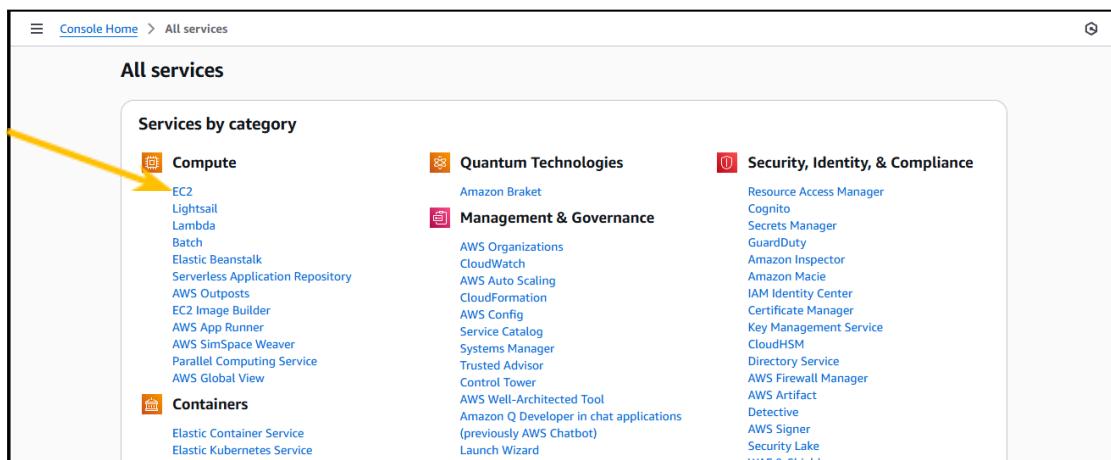


Launching a new EC2 Instance

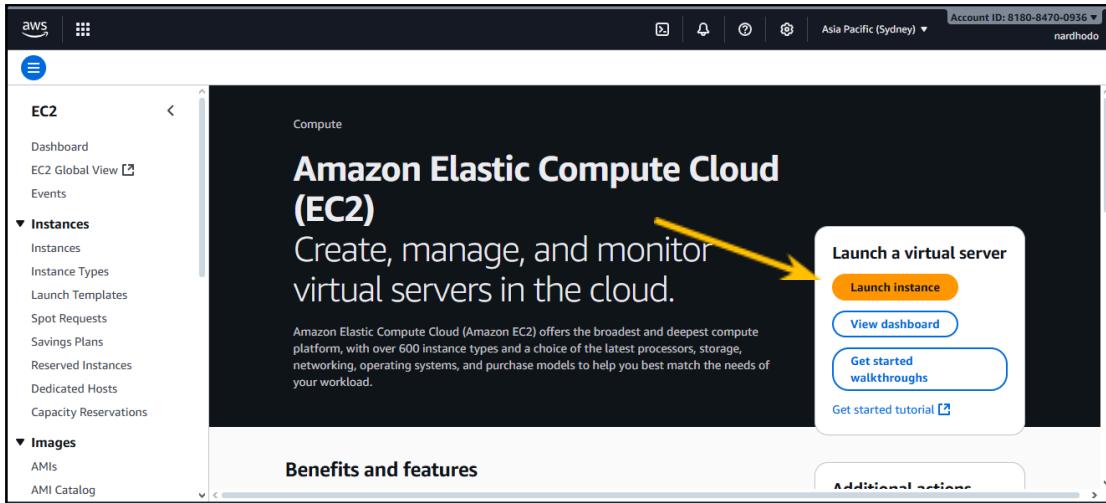
1. Scroll down within the console and search for “View All Services”.



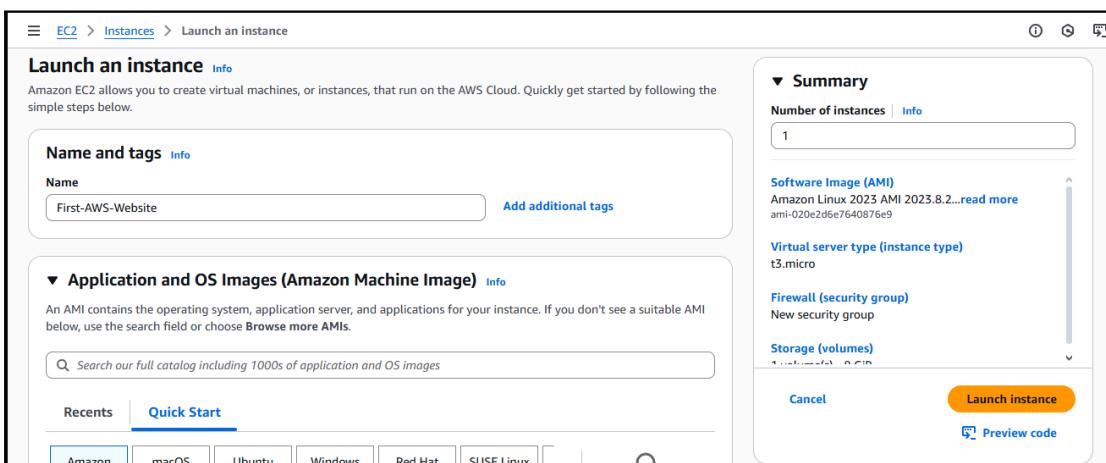
2. Select EC2



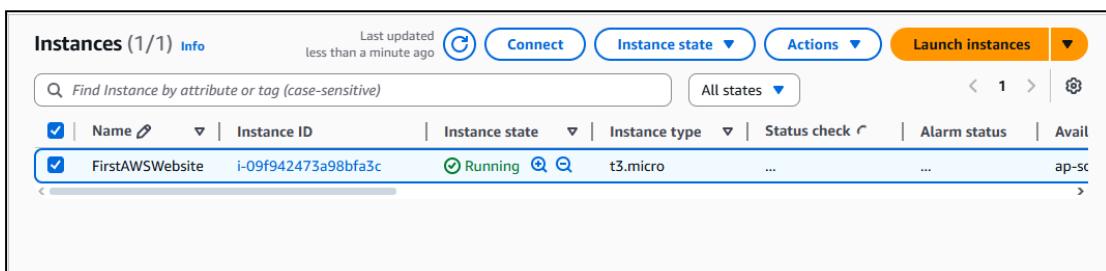
3. Click “Launch Instance”



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

| Instance summary for i-09f942473a98bfa3c (FirstAWSWebsite) Info | | |
|---|---|--|
| Connect Instance state Actions | | |
| Refreshing instance data | | |
| Instance ID i-09f942473a98bfa3c | Public IPv4 address 3.25.147.119 open address | Private IPv4 addresses 172.31.44.164 |
| IPv6 address - | Instance state Running | Public DNS ec2-3-25-147-119.ap-southeast-2.compute.amazonaws.com open address |
| Hostname type IP name: ip-172-31-44-164.ap-southeast-2.compute.internal | Private IP DNS name (IPv4 only) ip-172-31-44-164.ap-southeast-2.compute.internal | Elastic IP addresses - |
| Answer private resource DNS name IPv4 (A) | Instance type t3.micro | AWS Compute Optimizer finding - |
| Auto-assigned IP address - | VPC ID vpc-0bc4e7baec6b92983 | |

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

| Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags |
|---|--|--|----------|------------|---------|------|
| ▼ Security details | | | | | | |
| IAM Role - | Owner ID 818084700936 | Launch time Thu Sep 11 2025 15:57:42 GMT+0800 (Philippine Standard Time) | | | | |
| Security groups sg-038aa43fd33142dca (launch-wizard-1) | | | | | | |

11. Click on the “Edit Inbound Rules” button

| Inbound rules | Outbound rules | Sharing - new | VPC associations - new | Tags |
|---|---|--|--|---|
| Inbound rules (2) | | | | |
| <input type="checkbox"/> Name <input type="checkbox"/> - | <input type="checkbox"/> Security group rule ID <input type="checkbox"/> - | <input type="checkbox"/> IP version <input type="checkbox"/> IPv4 | <input type="checkbox"/> Type <input type="checkbox"/> Custom TCP | <input type="checkbox"/> Protocol <input type="checkbox"/> TCP |

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

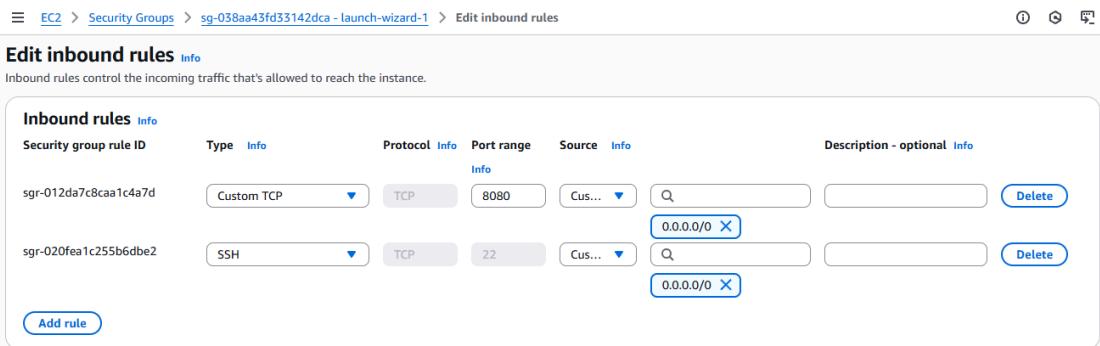
☰ EC2 > Security Groups > sg-038aa43fd33142dca - launch-wizard-1 > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

| Security group rule ID | Type | Protocol | Port range | Source | Description - optional | Actions |
|------------------------|------------|----------|------------|----------|--------------------------------|-------------------------|
| sgr-012da7c8caa1c4a7d | Custom TCP | TCP | 8080 | Cus... ▾ | <input type="text"/> 0.0.0.0/0 | <button>Delete</button> |
| sgr-020fea1c255b6dbe2 | SSH | TCP | 22 | Cus... ▾ | <input type="text"/> 0.0.0.0/0 | <button>Delete</button> |

[Add rule](#)

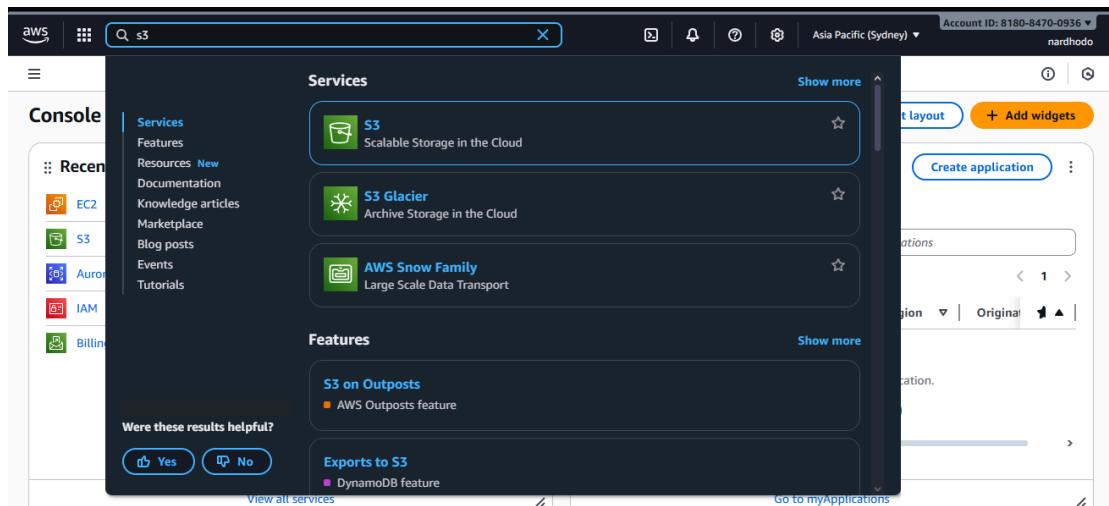


13. Click “Save Rules”
14. The EC2 Instance is fully configured and running.

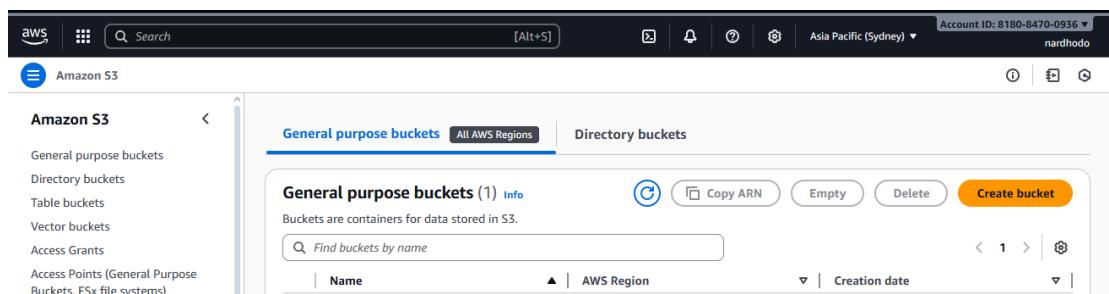
Configuring File Storage using S3 Buckets

A. Creating the Bucket

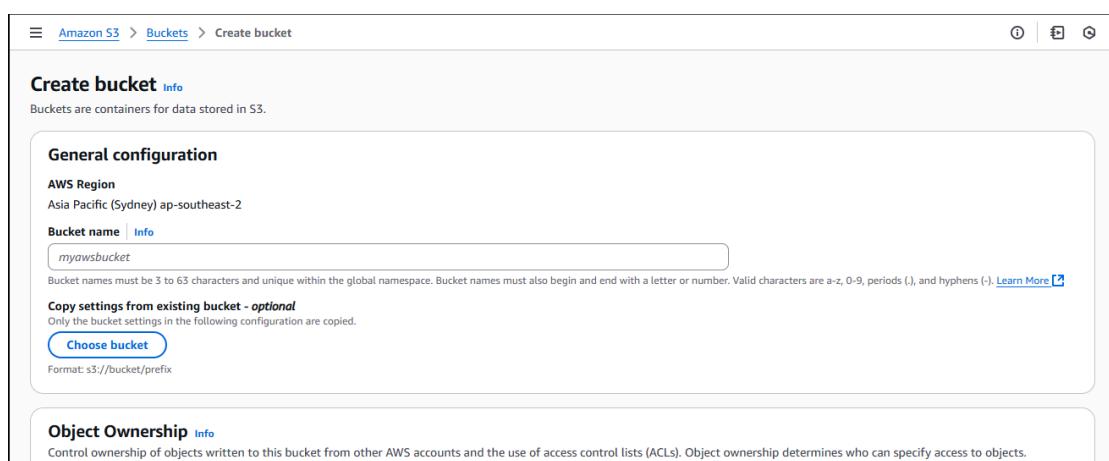
1. Click “View all Services” or you could search “S3” from the search bar above.



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.

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

⚠️ Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

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.

Tags - optional (2)
You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

| Key | Value - optional | Remove |
|-------------|------------------|------------------------|
| application | ricochi-shrine | Remove |
| environment | production | Remove |

[Add new tag](#)
You can add up to 48 more tags.

Default encryption [Info](#)
Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).

Server-side encryption with Amazon S3 managed keys (SSE-S3)

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

B. Modifying the Bucket

1. Click on the created bucket

The screenshot shows the AWS S3 console interface. At the top, there are two tabs: "General purpose buckets" (which is selected) and "All AWS Regions". Below the tabs, there is a search bar labeled "Find buckets by name" and a table with two columns: "Name" and "AWS Region". A single row is visible, showing the bucket "ricochi-shrine" in the "Asia Pacific (Sydney) ap-southeast-2" region. To the right of the table are buttons for "Copy ARN", "Empty", "Delete", and "Create bucket". Below the table, there are navigation arrows and a settings gear icon.

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

The screenshot shows the "Permissions" tab for the bucket "ricochi-shrine". The tab also includes links for "Objects", "Properties", "Metrics", "Management", and "Access Points". Under the "Permissions overview" section, there is a note about "Access finding" and a link to "View analyzer for ap-southeast-2". In the "Block public access (bucket settings)" section, there is a note about public access being granted through various methods. It shows that "Block all public access" is currently "Off" and has an "Edit" button. There is also a link to "Individual Block Public Access settings for this bucket".

The screenshot shows the "Bucket policy" page for the bucket "ricochi-shrine". It includes a note about the bucket policy providing access to objects stored in the bucket. It has "Edit" and "Delete" buttons at the top right. There is also a link to "Learn more".

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

Edit bucket policy Info

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts.
[Learn more](#)

Bucket ARN
arn:aws:s3:::ricochi-shrine

Policy

```
1 Version: "2012-10-17",
2   "Statement": [
3     {
4       "Effect": "Allow",
5       "Principal": "*",
6       "Action": "s3:GetObject",
7       "Resource": "arn:aws:s3:::ricochi-shrine/*"
8     }
9   ]
10 ]
11 ]
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

[+ Add new statement](#)

4. Click "Save Changes"

C. Uploading Files to the Bucket

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

The screenshot shows the 'Objects' tab for the 'ricochi-shrine' bucket. There is one object listed:

| Name | Type | Last modified | Size | Storage class |
|-------------|------|--|----------|---------------|
| image_1.jpg | jpg | September 11, 2025, 16:08:02 (UTC+08:00) | 133.1 KB | Standard |

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

The screenshot shows the 'Upload' interface. It has three main sections:

- Upload**: A large area for dragging and dropping files or choosing files/folders.
- Files and folders**: Shows one file selected: 'image_1.jpg' (133.1 KB). It includes a 'Remove' button and 'Add files/Add folder' buttons.
- Destination**: Set to 's3://ricochi-shrine'.

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”

The screenshot shows two screenshots of the AWS IAM interface. The top screenshot is the 'Access keys' page under 'Security credentials'. It displays one active access key: 'AKIA346NF4MEHDAHWGDO' (Created 6 hours ago, Last used None, Region N/A, Status Active). Below it is a 'CloudFront key pairs' section. The bottom screenshot is a 'Create access key' wizard. Step 1 is selected ('Alternatives to root user access keys'). It contains a warning about root user access keys and suggests using IAM roles or IAM Identity Center. Step 2 ('Retrieve access key') is shown below. At the bottom, there's a checkbox for understanding the risks and a 'Create access key' button.

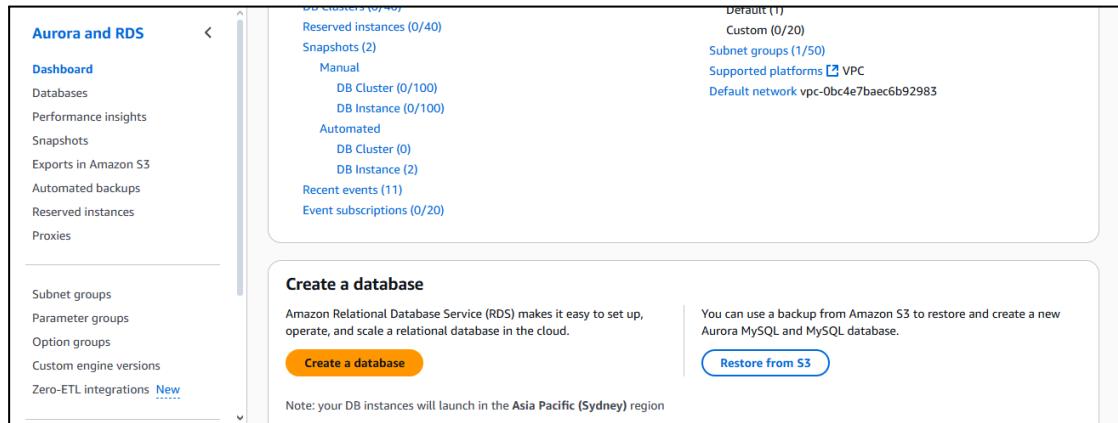
4. An Access key will be generated, make sure to copy and save it or download the csv file.

This screenshot shows the generated access key details. It includes the 'Access key' (AKIA346NF4MEHDAHWGDO) and the 'Secret access key' (represented by a series of asterisks). A 'Show' button is available to view the full secret key.

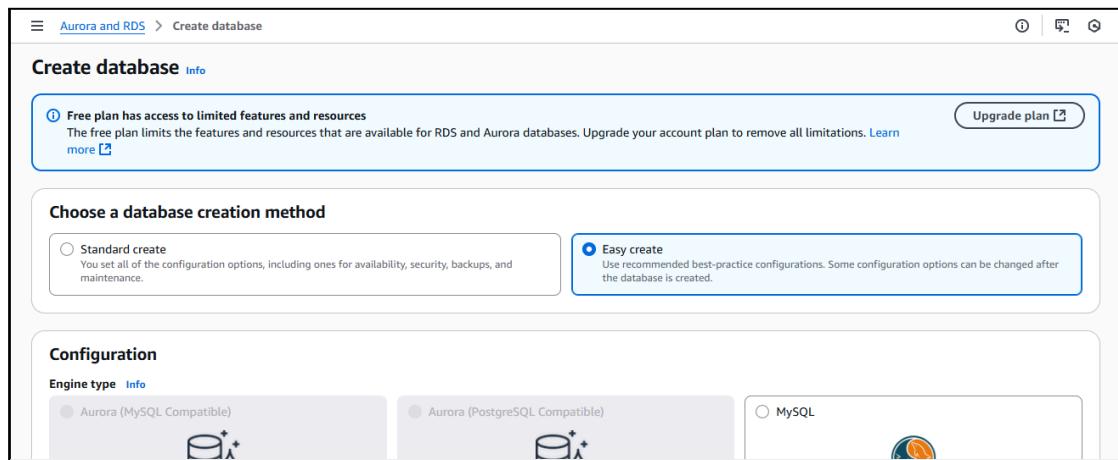
5. Click “Done”

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.



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

The screenshot shows the 'Create database' configuration page. Under 'Engine type', 'PostgreSQL' is selected. Under 'DB instance size', 'Free tier' is selected. Both options are highlighted with blue outlines.

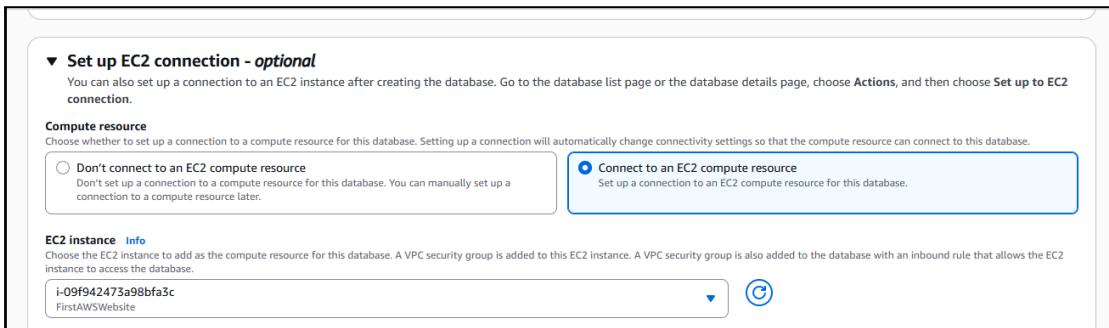
| Engine type | DB instance size |
|--------------------------------|--|
| Aurora (MySQL Compatible) | Production db.r7g.large 4 vCPUs 32 GB RAM 400 GB 2.203 USD/hour |
| Aurora (PostgreSQL Compatible) | Dev/Test db.r7g.large 2 vCPUs 16 GB RAM 200 GB 0.325 USD/hour |
| PostgreSQL | Free tier db.t4g.micro 2 vCPUs 1 GB RAM 20 GB 0.029 USD/hour |
| MariaDB | |
| MySQL | |
| Microsoft SQL Server | |
| Oracle | |

- 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.

The screenshot shows the 'Create database' configuration page. Under 'DB instance identifier', 'database-1' is entered. Under 'Master user credentials', 'Self managed' is selected. Both fields have blue outlines indicating they are selected.

| DB instance identifier | Master user credentials |
|------------------------|--|
| database-1 | Self managed Create your own password or have RDS create a password that you manage. |

- Scroll down and find “Setup EC2 Connection”
- Expand the tab, and check the “Connect to an EC2 compute resource” option.
- 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.

The screenshot shows the AWS EC2 Instances page. At the top, there's a header with 'Instances (1) Info' and a 'Connect' button. Below the header is a search bar with placeholder text 'Find Instance by attribute or tag (case-sensitive)' and a dropdown menu set to 'All states'. The main table lists one instance: 'FirstAWSWebsite' with Instance ID 'i-09f942473a98bfa3c', which is 'Running' (indicated by a green checkmark), of type 't3.micro', and has 3/3 checks passed. The table also includes columns for Status check, Alarm status, and Availability.

2. Click on the instance
3. Click on the “Connect Button”

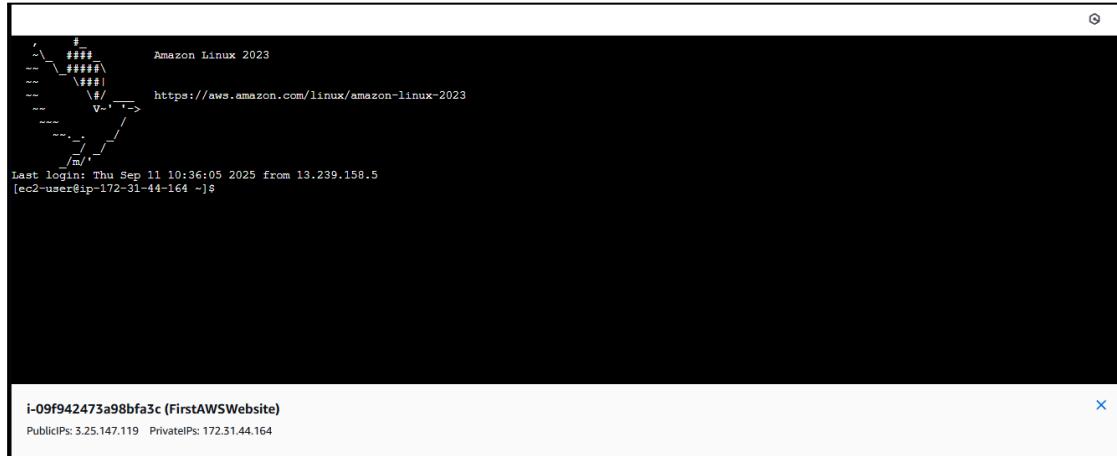
The screenshot shows the 'Instance summary for i-09f942473a98bfa3c (FirstAWSWebsite)' page. It displays various details about the instance, including its ID, state, and network information. Key details include:

- Instance ID:** i-09f942473a98bfa3c
- Instance state:** Running
- Public IPv4 address:** 3.25.147.119
- Private IPv4 addresses:** 172.31.44.164
- Public DNS:** ec2-3-25-147-119.ap-southeast-2.compute.amazonaws.com
- Private IP DNS name (IPv4 only):** ip-172-31-44-164.ap-southeast-2.compute.internal
- Instance type:** t3.micro
- VPC ID:** vpc-0bc4e7baec6b92983

The screenshot shows the 'Connect' dialog box for the instance. It has tabs for 'EC2 Instance Connect', 'Session Manager', 'SSH client', and 'EC2 serial console'. The 'EC2 Instance Connect' tab is active. The form fields include:

- Instance ID:** i-09f942473a98bfa3c (FirstAWSWebsite)
- Connection type:**
 - Connect using a Public IP (selected)
 - Connect using a Private IP
- Username:** ec2-user (pre-filled in the input field)
- Note:** A note at the bottom states: 'Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.'
- Buttons:** 'Cancel' and 'Connect' (highlighted in orange)

4. You will be directed to the terminal.



The screenshot shows a terminal window titled 'Amazon Linux 2023' with a stylized logo. The prompt is '[ec2-user@ip-172-31-44-164 ~]\$'. The output includes a welcome message from the OS, a timestamp ('Last login: Thu Sep 11 10:36:05 2025 from 13.239.158.5'), and the user's session identifier ('[ec2-user@ip-172-31-44-164 ~]\$'). At the bottom of the terminal, there is a status bar with the text 'i-09f942473a98bfa3c (FirstAWSWebsite)' and '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 ~]$ 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=AKIA346NF4MEHDAHWGQ
[ec2-user@ip-172-31-44-164 ~]$ export S3_SECRET_KEY=JF5F+aO3rvb7GnM7230iMDqNv+CikvTNxRxa2oo7
[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.