

Name: Arthur M. Artugue

Year and Section: IV-BCSAD

Assignment 2 : How-To Document Guide

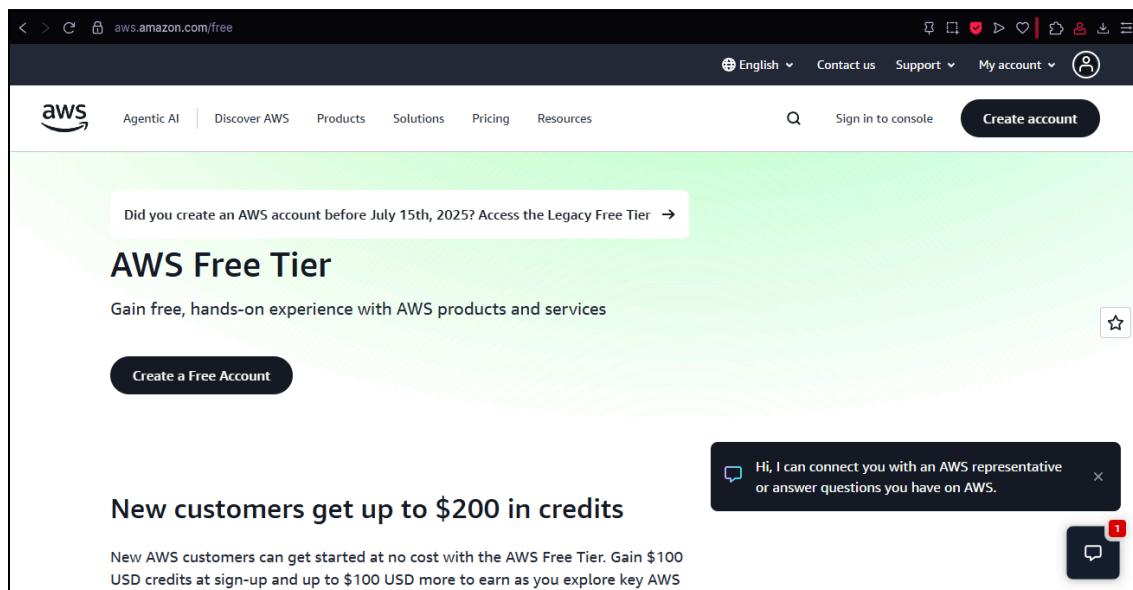
What is AWS and its Key Benefits?

Amazon Web Services (AWS) is a popular cloud platform that allows you to host websites and applications, scaling your servers up or down as needed and paying only for what you use. With AWS, your applications get high performance, reliability, security, and global reach automatically. AWS provides access to an expanding catalog of services, including cutting-edge AI and machine learning options.

Signing Up for AWS for Free

You can get started with AWS for free, which is perfect for this guide.

1. **Head to the website.** Go to <https://aws.amazon.com/free> and click the "Create a Free Account" button.



2. **Enter your email.** This email will be used to log in to your **root account**, which gives you full control of your aws account

3. **Choose a password.** Pick a strong password and click "Continue".

The screenshot shows the 'Create your password' step of the AWS sign-up process. At the top, a green box displays a success message: 'It's you! Your email address has been successfully verified.' Below this, a note states: 'Your password provides you with sign in access to AWS, so it's important we get it right.' There are two input fields: 'Root user password' and 'Confirm root user password', both containing masked text. A large orange 'Continue (step 1 of 5)' button is centered below them. To the left and right of the button are decorative 3D cube graphics. Below the cubes is a link 'Sign in to an existing AWS account'.

4. **Fill out the form.** Enter your contact information and agree to the terms.

The screenshot shows the 'Contact Information' step of the AWS sign-up process. At the top, the AWS logo is displayed. The page title is 'Sign up for AWS'. The section title 'Contact Information' is followed by the question 'How do you plan to use AWS?'. Two radio button options are shown: 'Business - for your work, school, or organization' (unchecked) and 'Personal - for your own projects' (checked). The next question is 'Who should we contact about this account?'. Below this, there is a 'Full Name' field containing 'Arthur M. Artigue'. Under 'Country Code' is a dropdown menu showing '+63' with a dropdown arrow, and next to it is a 'Phone Number' field containing '222-333-4444'. Finally, there is a 'Country or Region' dropdown menu showing 'Philippines'.

5. **Provide billing information.** You will need to enter your credit card information, but you won't be charged if you stay within the free tier limits.

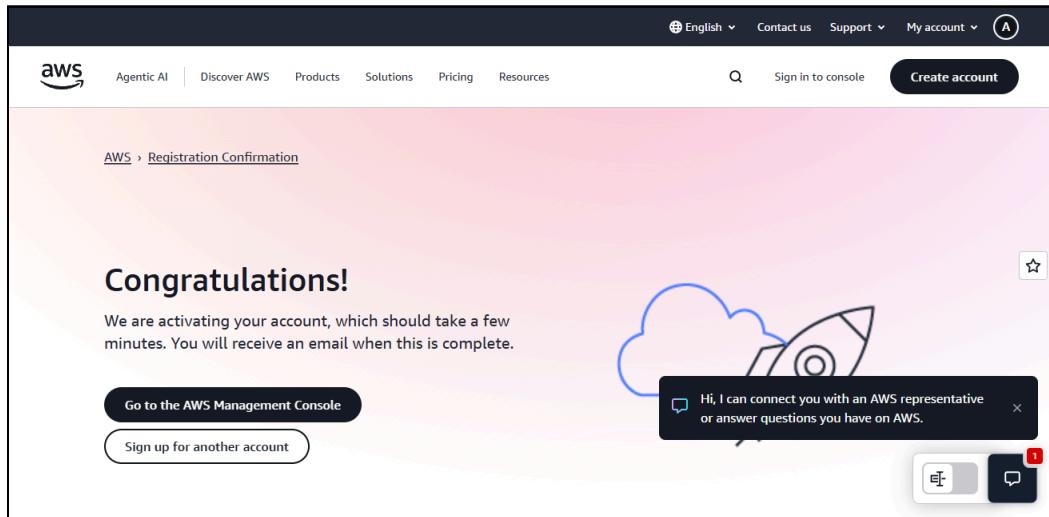
The screenshot shows the 'Sign up for AWS' process. On the left, there is a blue callout box titled 'Why is this required?' explaining that the verification process holds USD \$1 (or equivalent) for 3-5 days to verify the account and prevent fraud. It also notes that for the free plan, no charges occur until upgrade to a paid plan. Below this is a small icon of a house with a checkmark inside. On the right, the 'Billing Information' section is visible, featuring fields for 'Billing country' (set to Philippines), 'Credit or Debit card number' (with a placeholder field and icons for VISA, MasterCard, AMEX, and DISCOVER), 'Expiration date' (with dropdown menus for Month and Year), and 'Security code' (with a placeholder field). A note at the bottom states that AWS accepts most major credit and debit cards and provides a link to the FAQ.

6. **Verify your identity.** Enter your mobile phone number and verify the text message AWS sends you.

The screenshot shows the 'Sign up for AWS' process. On the left, there is an icon of a person's profile with a checkmark. On the right, the 'Confirm your identity' section is visible, which includes a note about verifying the phone number before using the AWS account. It asks how to send the verification code: 'Text message (SMS)' is selected, while 'Voice call' is an option. Below this are fields for 'Country or region code' (set to United States (+1)) and 'Mobile phone number'. At the bottom is a large orange button labeled 'Send SMS (step 4 of 5)'.

7. **Choose a support plan.** Select the free **Basic Support** plan and complete the sign-up.

8. **Account activation.** Your account activation may take a few minutes. You'll receive an email when it's ready.



9. **Sign in.** Once you get the email, click "Sign in to AWS" to go to the management console.
-

Setting a Goal: Hosting a Website on AWS

In this guide, we'll follow Mike's example of creating a website for his bakery, "Mike's Macaroon Market". This website will require three key components:

- **Compute resources** to host the web server.
 - **Storage** for the macaroon images.
 - A **database** to store inventory and track orders.
-

Exploring the AWS Management Console

The **AWS Management Console** is the central hub for your account, giving you an overview and access to all AWS services. Before you start, it's a good idea to set your **region**.

A region is a geographically distinct data center, and its location can affect service availability, pricing, and latency. Choosing a region close to your expected users is a good strategy. For example, the US East (N. Virginia) region is a good choice because it's generally more affordable.

From the console, you can also check your costs in the **Cost and Usage panel** and view service-related notifications in the **AWS Health panel**.

The screenshot shows the AWS Home page with the following sections:

- Welcome to AWS**: Includes links for "Getting started with AWS", "Training and certification", and "AWS Builder Center".
- AWS Health Info**: Shows 0 open issues and 0 scheduled changes.
- Cost and usage**: Shows credits remaining (\$100.00 USD) and days remaining (176 days). It also includes sections for Current month, Forecasted month end, and Cost breakdown.

At the bottom, there are links for "Go to AWS Health", "Go to Billing and Cost Management", and navigation links like CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

EC2 - Launching a Virtual Server

EC2 (Elastic Compute Cloud) is an AWS service that lets you create virtual servers, which are called **instances**. These instances are perfect for hosting applications, like the Node.js website for Mike's bakery. You can launch, manage, and resize them without worrying about physical hardware.

Note: The EC2 usage is billed in one-second increments with a minimum of 60 seconds in "On Demand Pricing" plan. Learn more about EC2 pricing here <https://aws.amazon.com/ec2/pricing>

Steps to Launch an EC2 Instance:

1. **Go to EC2.** From the AWS console, click on "View all services" and select "EC2".

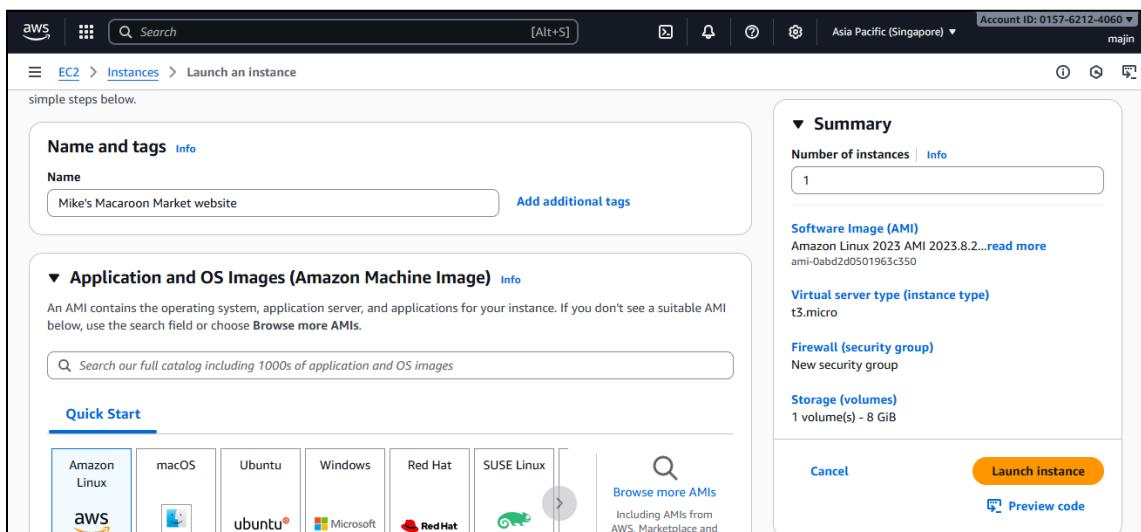
The screenshot shows the "All services" page under the "Services by category" section. The "Compute" section is selected, listing the following services:

- EC2
- Lightsail
- Lambda
- Batch
- Elastic Beanstalk
- Serverless Application Repository
- AWS Outposts
- EC2 Image Builder
- AWS App Runner
- AWS SimSpace Weaver
- Parallel Computing Service
- AWS Global View

2. **Launch an instance.** Click the "Launch instance" button.



3. **Name the instance.** Give your instance a descriptive name, like "Mike's Macaroon Market website".



4. **Choose an operating system.** Select an **Amazon Machine Image (AMI)**, which is the operating system for your instance. The default, **Amazon Linux**, is a great choice and is eligible for the free tier.

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

Quick Start

Search our full catalog including 1000s of application and OS images

Amazon Machine Image (AMI)

Amazon Linux	macOS	Ubuntu	Windows	Red Hat	SUSE Linux	Browse more AMIs
						Including AMIs from AWS, Marketplace and the Community

Amazon Linux 2023 kernel-6.1 AMI

ami-0abd2d0501963c350 (64-bit (x86), uefi-preferred) / ami-0787fd46c032f549c (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

5. **Select an instance type.** This determines the amount of CPU and memory your virtual server will have. Stick with the default free tier-eligible option for this project.

Instance type [Info](#) | Get advice

Instance type

t3.micro Free tier eligible

Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0224 USD per Hour
On-Demand SUSE base pricing: 0.0132 USD per Hour
On-Demand Linux base pricing: 0.0132 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0167 USD per Hour
On-Demand RHEL base pricing: 0.042 USD per Hour

m7i-flex.large Free tier eligible

Family: m7i-flex 2 vCPU 8 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.1197 USD per Hour
On-Demand SUSE base pricing: 0.176 USD per Hour
On-Demand RHEL base pricing: 0.1485 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.1232 USD per Hour
On-Demand Windows base pricing: 0.2071 USD per Hour

All generations

Compare instance types

the selected key pair before you

Key pair name - required

6. **Configure network settings.** Leave the default network settings, but you will need to allow HTTP traffic by modifying the security group later. The web application will use port 8080, which is a higher-numbered port that avoids some Linux setup steps needed for lower-numbered ports like 80.

The screenshot shows the 'Network settings' configuration step of the AWS Launch Wizard. It includes sections for Network, Subnet, Auto-assign public IP, and Firewall (security groups). The Firewall section is expanded, showing options to 'Create security group' (selected) or 'Select existing security group'. Below this, it says 'We'll create a new security group called 'launch-wizard-1' with the following rules:' followed by three checkboxes: 'Allow SSH traffic from Anywhere' (selected), 'Allow HTTPS traffic from the internet', and 'Allow HTTP traffic from the internet'. A warning message at the bottom states: '⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' An 'Edit' button is located in the top right corner.

7. **Launch.** After reviewing your settings, click "Launch Instance".

The screenshot shows the 'Summary' screen of the AWS Launch Wizard. It displays the following configuration:

- Number of instances: 1
- Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2... [read more](#) ami-0abd2d0501963c350
- Virtual server type (instance type): t3.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

At the bottom, there are 'Cancel', 'Launch instance' (highlighted in orange), and 'Preview code' buttons.

After the instance is created, you can configure the security group to allow incoming HTTP traffic on port 8080.

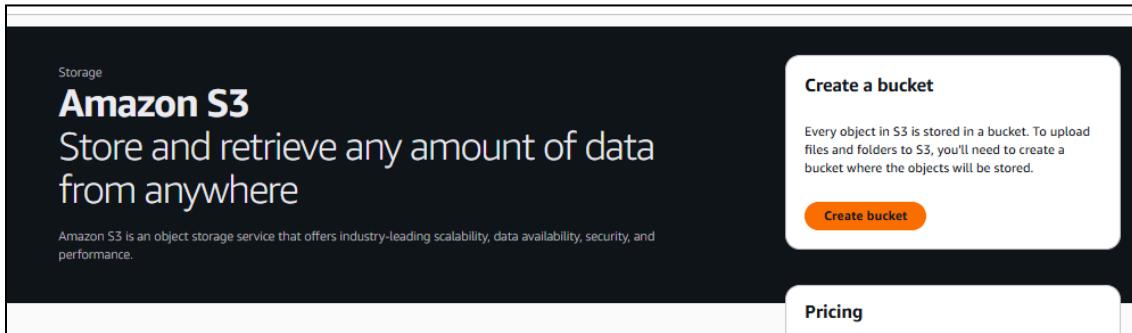
S3 - Configuring File Storage

S3 (Simple Storage Service) is a service for storing files. You can store files privately or make them publicly accessible, which is what's needed for the website's macaroon images. Files are stored in **buckets**, which are like root folders.

Note: There is no minimum charge in S3 it is "Pay only for what you use" (Same as the pay as you go in google cloud). But keep in mind that pricing will vary depending on "cost components", for example size of objects stored, data retrieval and data transfer. Learn more about S3 pricing here <https://aws.amazon.com/s3/pricing>

Steps to Create an S3 Bucket:

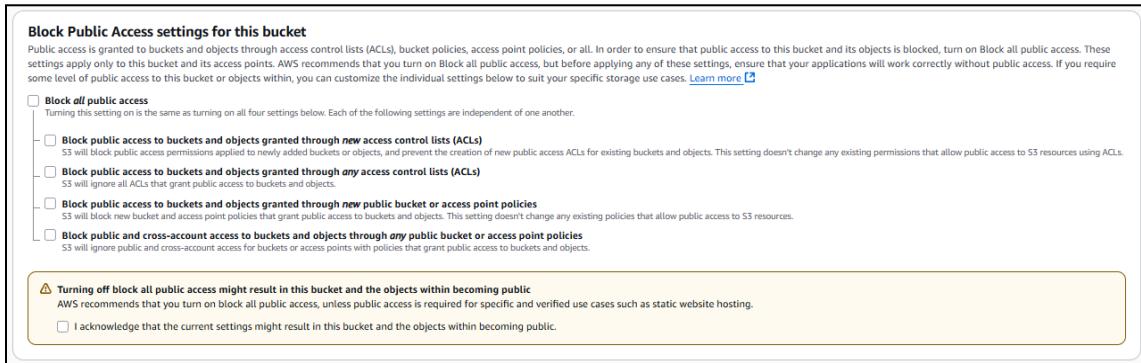
1. **Navigate to S3.** Use the search bar in the AWS console and type "S3".
2. **Create a bucket.** Click the "Create Bucket" button.



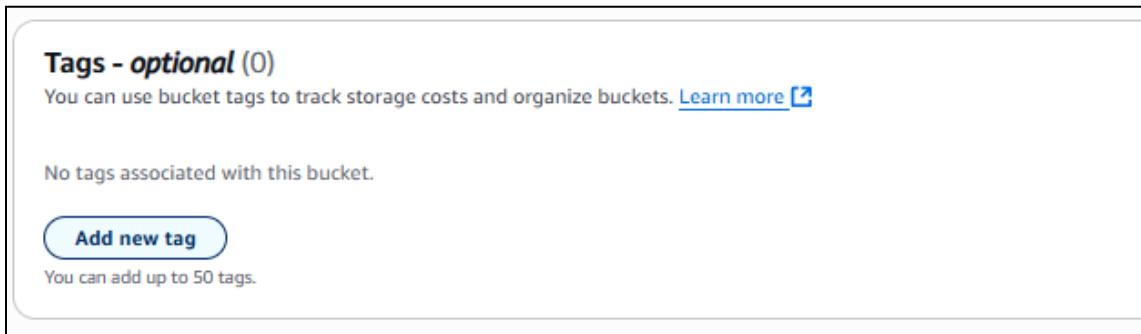
3. **Name your bucket.** Give the bucket a unique name. Since the name is part of the file's URL, it should be official-sounding and use hyphens instead of spaces (e.g., "mikes-macaroon-market").

A screenshot of the 'Create bucket' configuration page in the AWS console. The top section shows the title 'Create bucket' with a 'Info' link and a note that buckets are containers for data stored in S3. Below this is a 'General configuration' section. It includes fields for 'AWS Region' (set to Asia Pacific (Singapore) ap-southeast-1), 'Bucket type' (with 'General purpose' selected, indicated by a blue border), and 'Bucket name' (set to 'mikes-macaroon-market'). There is also a 'Directory' option with a note about its use cases. Further down, there is a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button and a note about copying settings. At the bottom, there is a note about the bucket name format: 'Format: s3://bucket/prefix'.

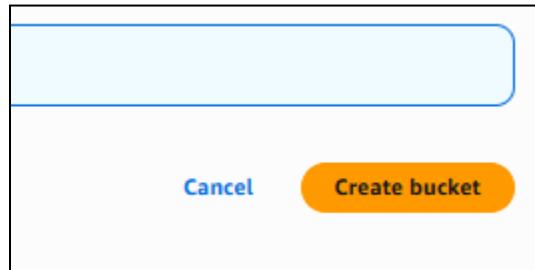
4. **Configure access.** Uncheck "Block all public access" to make the files in the bucket publicly available.



5. **Add tags.** Tags are labels that help you organize and group your resources, especially as you add more applications and environments (like production or dev).



6. **Create.** Click "Create Bucket".



7. **Set the bucket policy.** To make the bucket truly public, go to the **Permissions** tab and edit the **Bucket Policy** to allow everyone to read the files.
8. **Create an access key.** For your website to have permission to upload files, you'll need an access key and secret access key. Go to **Security Credentials** from the account menu and click "Create Access Key".
Be sure to save these keys securely! They will not be visible again after you leave the page.

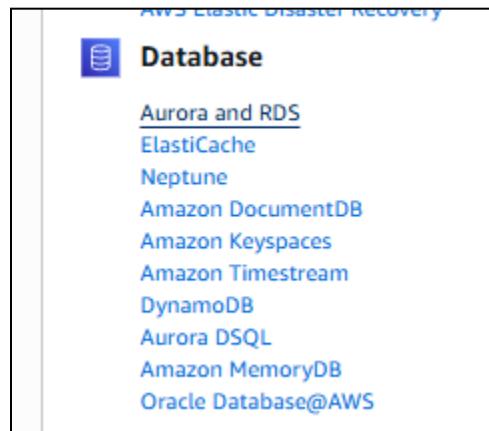
RDS - Creating a Database Instance

RDS (Relational Database Service) is a service for setting up a database. Mike's website needs a database to store product information and customer orders.

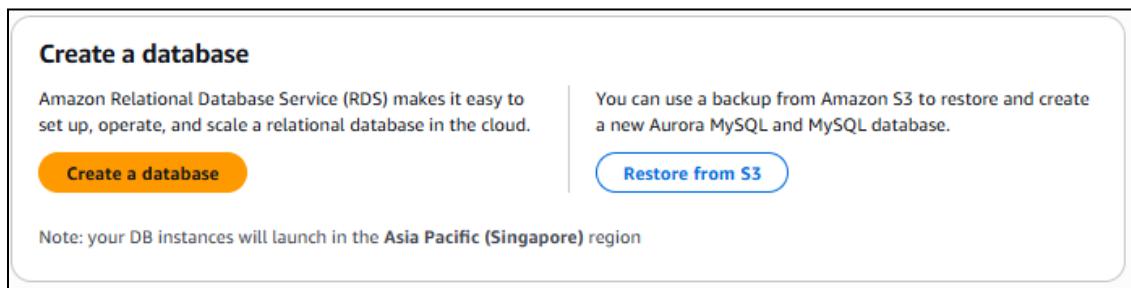
Note: Amazon Aurora has various pricing (pay-as-you-go, On-Demand, or Reserved Instance pricing) depending on your configuration and optional features you choose. Learn more about about its pricing here <https://aws.amazon.com/rds/aurora/pricing>

Steps to Create an RDS Database:

1. **Go to RDS.** Use the services menu (the nine-square button in the top left) and go to "Database," then "Aurora and RDS".



2. **Create a database.** Click the "Create Database" button.



3. **Choose a method.** Select **Easy Mode** to simplify the process and let AWS choose recommended settings.

Create database [Info](#)

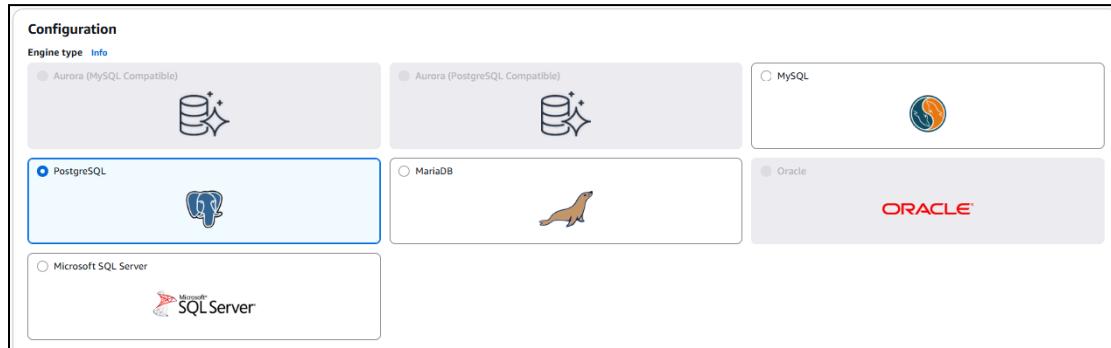
Free plan has access to limited features and resources
The free plan limits the features and resources that are available for RDS and Aurora databases. Upgrade your account plan to remove all limitations. [Learn more](#) [Upgrade plan](#)

Choose a database creation method

Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

4. **Select a database type.** Choose the database type your application expects, such as **Postgres**.



5. **Choose a size.** Select the smallest size to stay within the free tier.



6. **Name the instance.** Give your database a name, like "Mike's Macaroon Market".

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

Mikes

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens

7. **Set credentials.** You can keep the default username and have AWS generate a password for you.

8. **Connect to EC2.** Expand the "Set up EC2 Connection" section, check the "Connect to an EC2 compute resource" option, and select your EC2 instance from the dropdown menu. This automatically sets up the network and security to allow your EC2 instance to access the database.

9. **Create.** Click "Create Database".
 10. **Save credentials.** After creation, a notification will appear with your username and the generated password. Copy and save these for later, as they're needed to configure your website.

Connecting to an EC2 Instance and Running Commands

Now that all the AWS resources are created, the final step is to install and run the web application on the EC2 instance.

1. **Connect to your instance.** Go back to the EC2 dashboard, select your instance, and click the "Connect" button.
2. **Use EC2 Instance Connect.** The simplest way to connect is with **EC2 Instance Connect**, which lets you SSH to your instance directly from the browser. Just click the "Connect" button.
3. **Run commands.** A terminal window will open. Here you can run commands to download and configure your website.
 - o **Install Git:** Use `sudo dnf install git -y` to install the program needed to download the website code.

- **Download code:** Use `git clone` to download the website's repository.
 - **Set environment variables:** Use the `export` command to set environment variables with your AWS credentials (S3 bucket name, access key, secret key, database hostname, and password). This tells the website how to connect to your AWS resources.
 - **Install Node.js:** Use `sudo dnf install nodejs -y` to install the required software.
 - **Install dependencies:** Use the `npm install` command to install all the necessary packages for your website.
 - **Start the website:** Use the `npm start &` command to run your website in the background. Then, use the `disown` command to keep the process running even after you close the terminal.
4. **Access your website.** Your website is now live! You can find the public IP address of your server in the AWS console and browse to it on port 8080 (e.g., [http://\[your IP address\]:8080](http://[your IP address]:8080)).
-

Tips for Cost Savings on AWS

- **Choose affordable regions:** Stick to regions in the US and Europe, which are generally the most affordable, with the exception of US West 1.
 - **Prepay for resources:** If you know you'll need specific resources long-term, you can prepay for them at a discount.
 - **Use flexible bidding:** You can bid on Amazon's extra capacity for cheaper resources if your usage is flexible.
 - **Set spending limits:** To avoid unexpected bills, be sure to set a spending limit on your account.
-

Continuing Your AWS Journey with AI, Machine Learning, and AWS Certifications

Hosting a website is a great start, but AWS offers endless possibilities with services for artificial intelligence (AI), machine learning, computer vision, fraud detection, and more. If you're looking to validate your expertise and advance your career in the cloud industry, pursuing an **AWS certification** is an excellent next step.