Name: Arthur M. Artugue
Year and Section: IV-BCSAD

Assignment 2 : How-To Document Guide

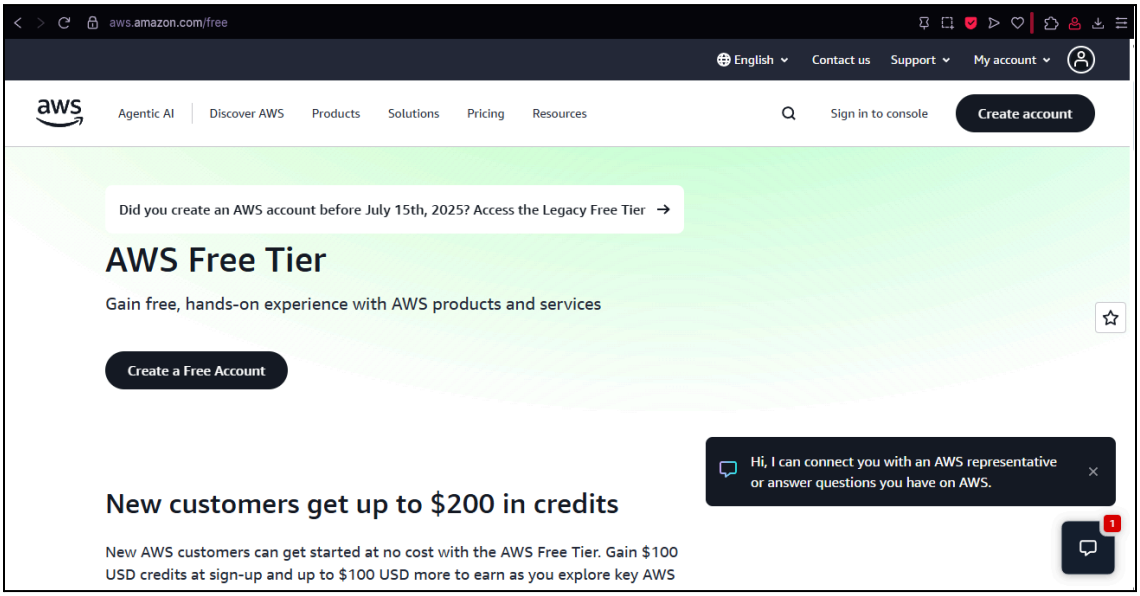
What is AWS and its Key Benefits?

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

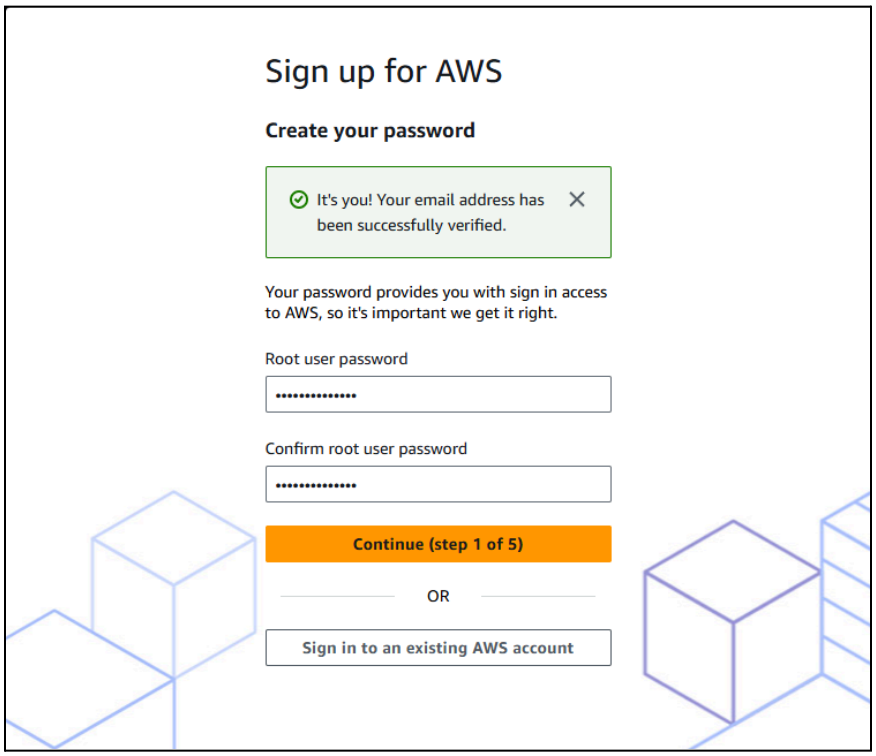
You can get started with AWS for free, which is perfect for this guide.

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

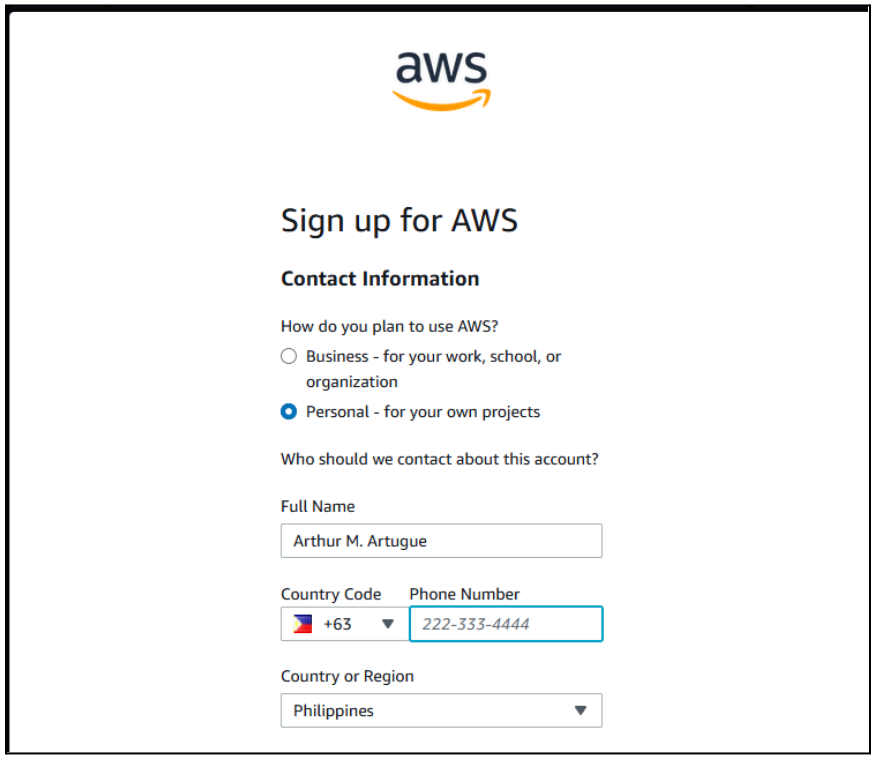


2. **Enter your email.** This email will be used to log in to your **root account**, which gives you full control of your aws account

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

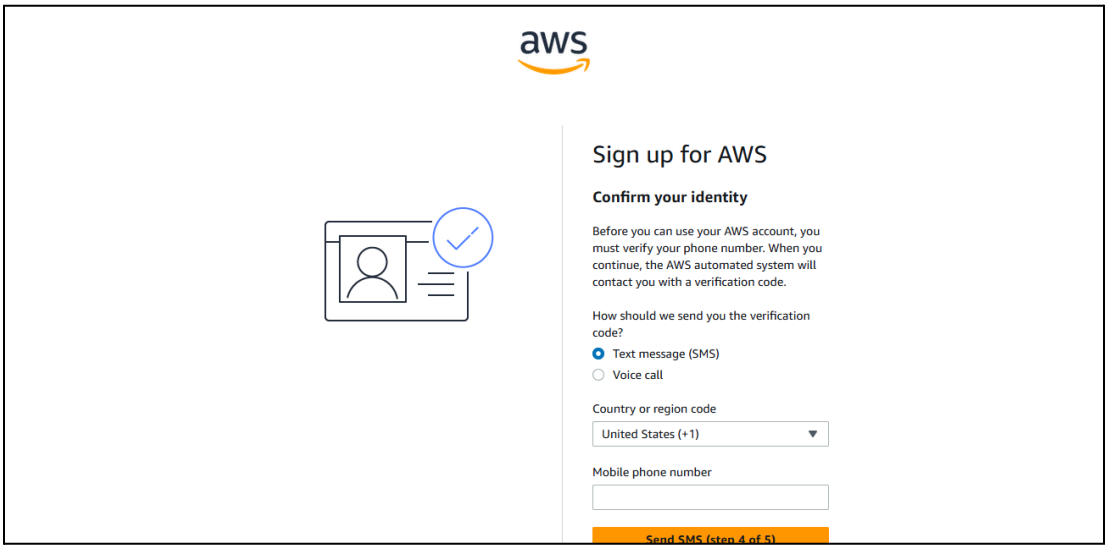


4. **Fill out the form.** Enter your contact information and agree to the terms.



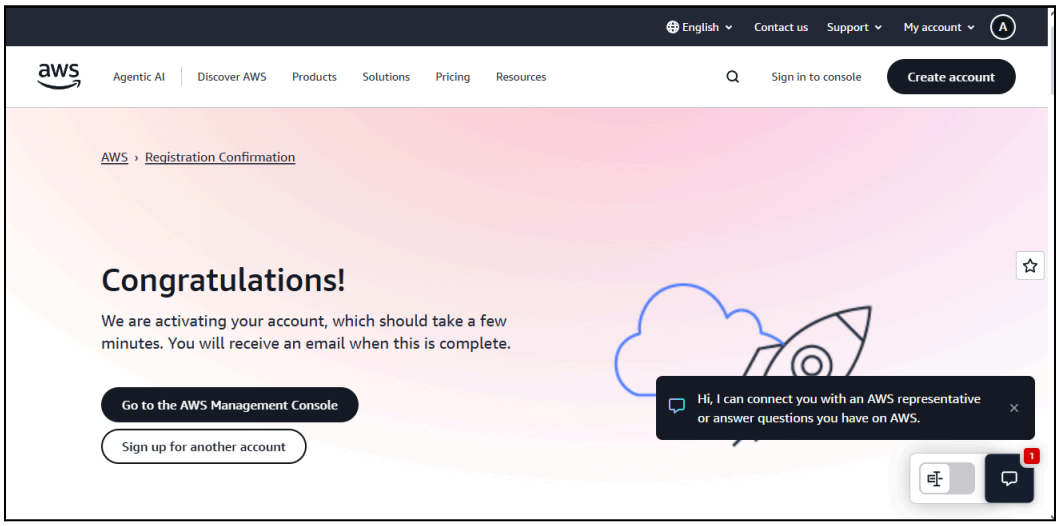
5. **Provide billing information.** You will need to enter your credit card information, but you won't be charged if you stay within the free tier limits.

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



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:

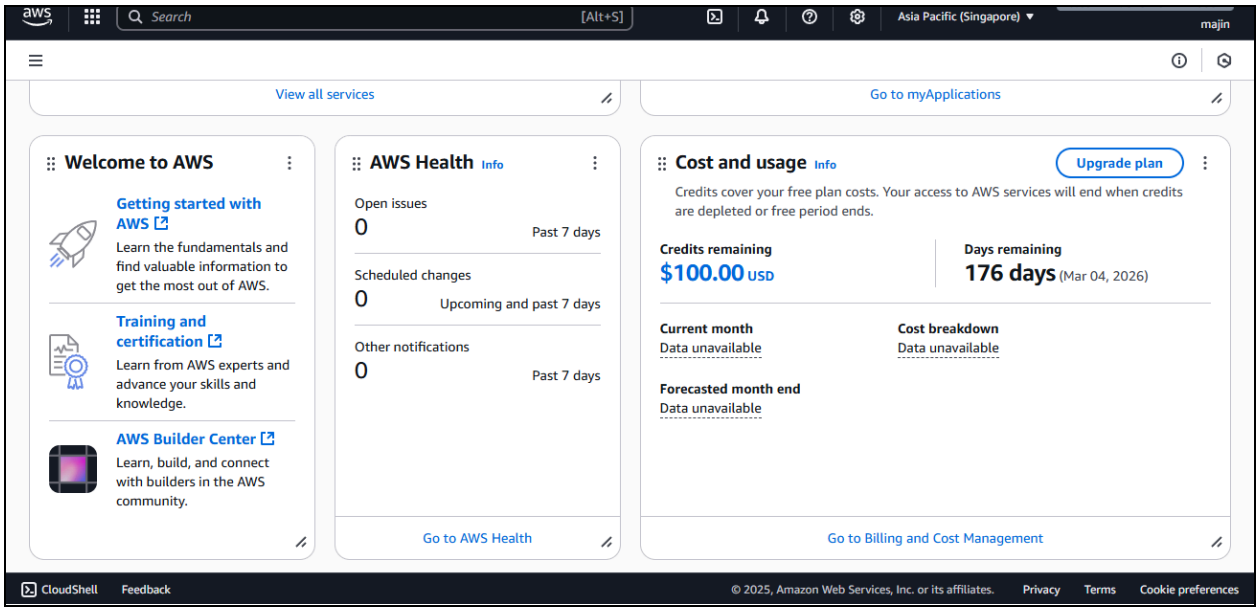
- **Compute resources** to host the web server.
- **Storage** for the macaroon images.
- A **database** to store inventory and track orders.

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**.



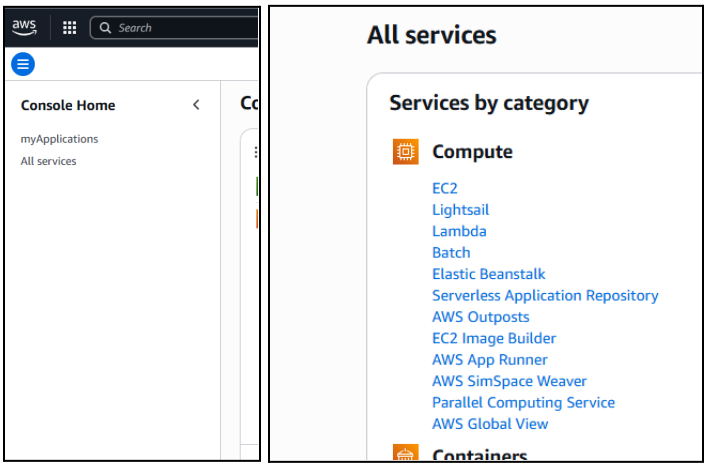
EC2 - Launching a Virtual Server

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.

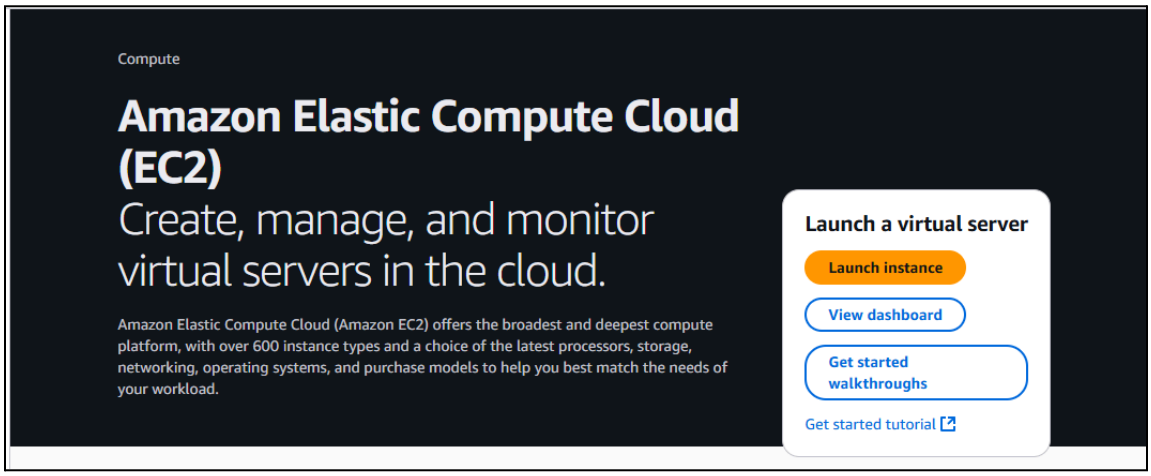
Note: The EC2 usage is billed in one-second increments with a minimum of 60 seconds in "On Demand Pricing" plan. Learn more about EC2 pricing here https://aws.amazon.com/ec2/pricing

Steps to Launch an EC2 Instance:

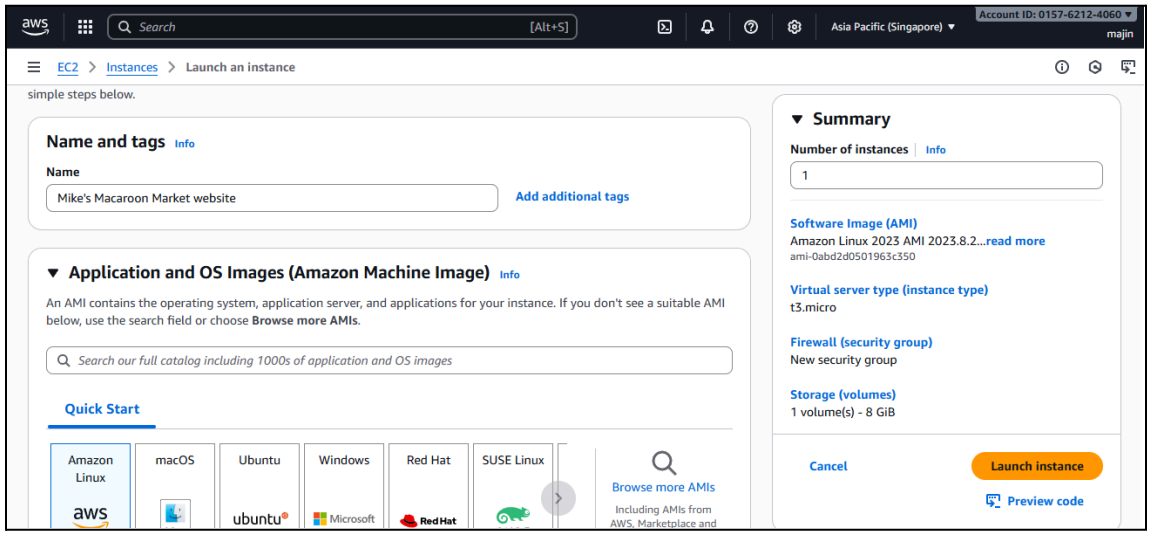
1. **Go to EC2.** From the AWS console, click on "View all services" and select "EC2".



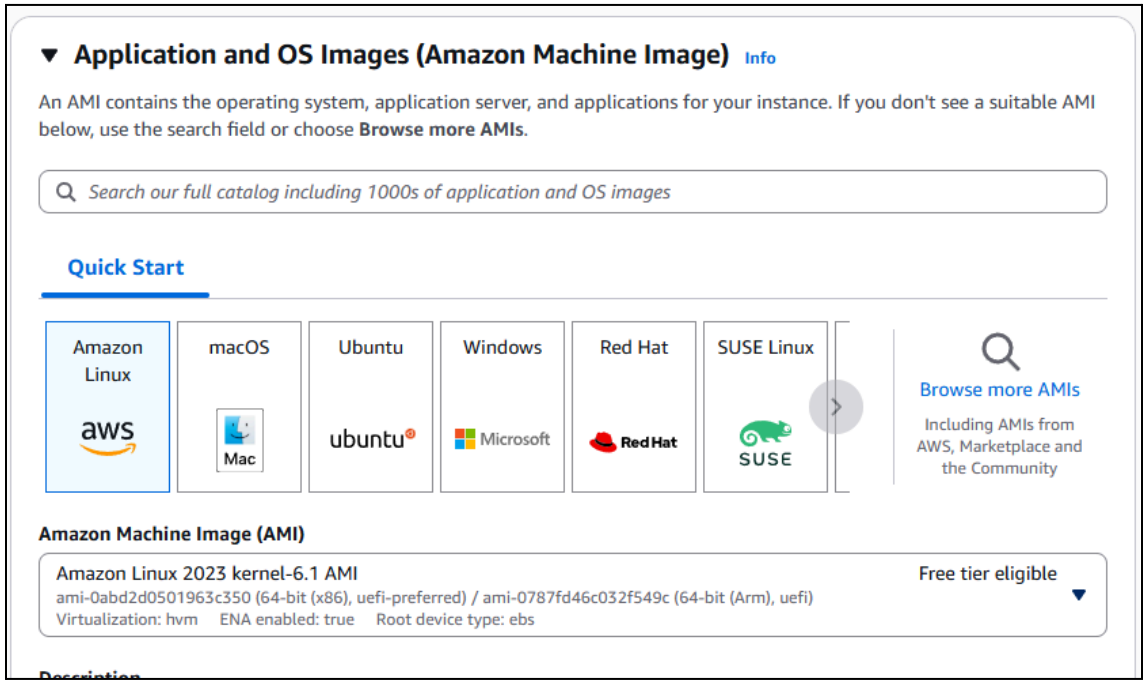
2. **Launch an instance.** Click the "Launch instance" button.



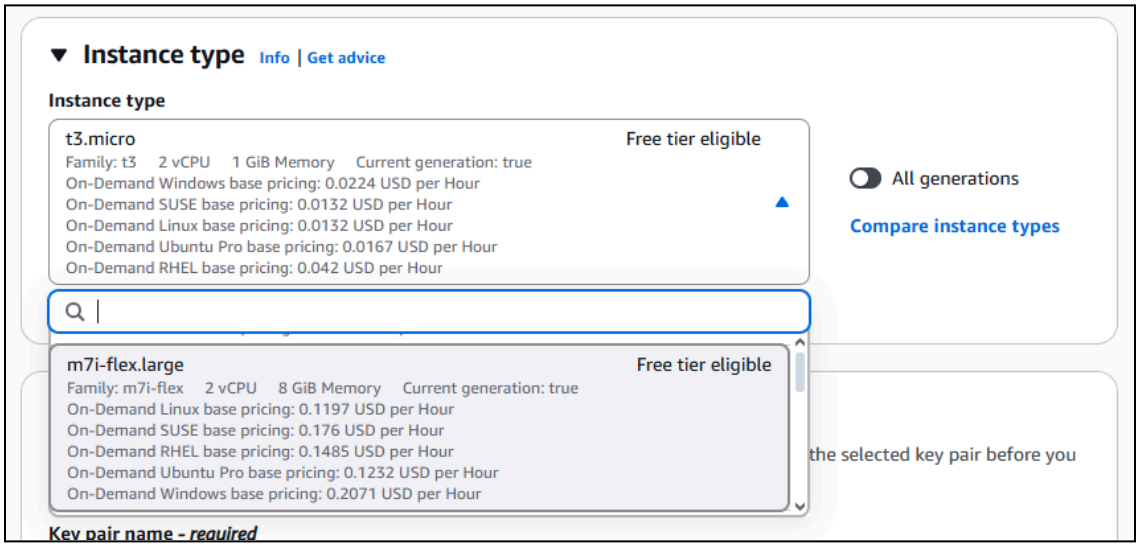
3. **Name the instance.** Give your instance a descriptive name, like "Mike's Macaroon Market website".



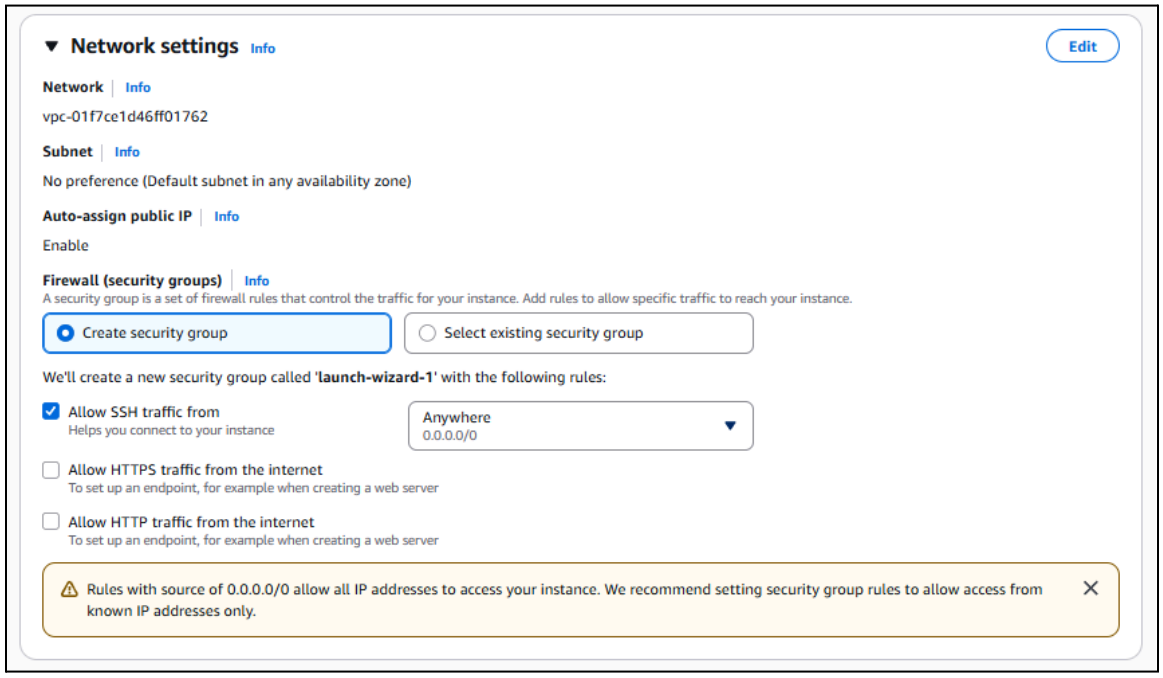
4. **Choose an operating system.** Select an **Amazon Machine Image (AMI)**, which is the operating system for your instance. The default, **Amazon Linux**, is a great choice and is eligible for the free tier.



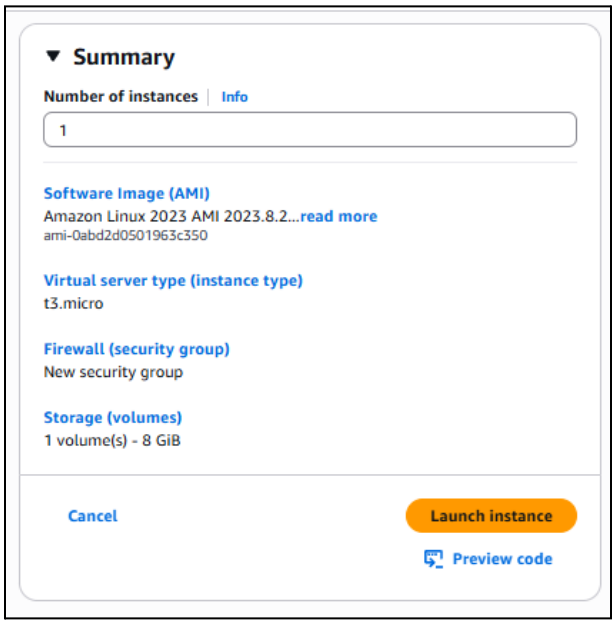
5. **Select an instance type.** This determines the amount of CPU and memory your virtual server will have. Stick with the default free tier-eligible option for this project.



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.



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



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

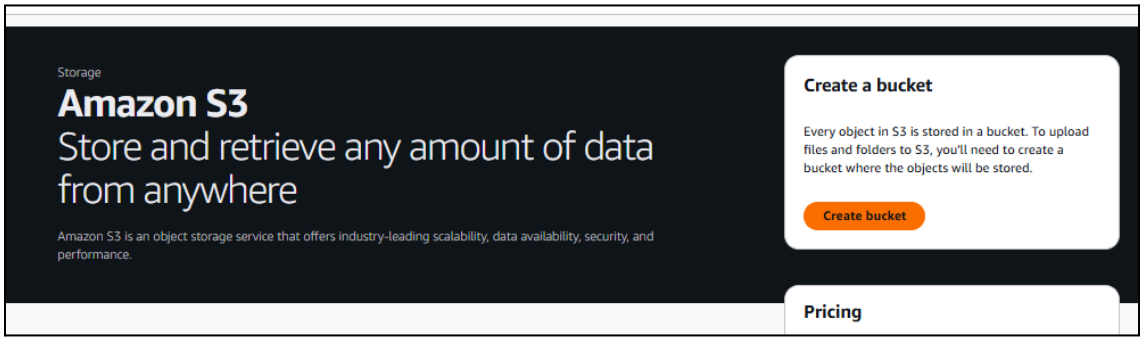
S3 - Configuring File Storage

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

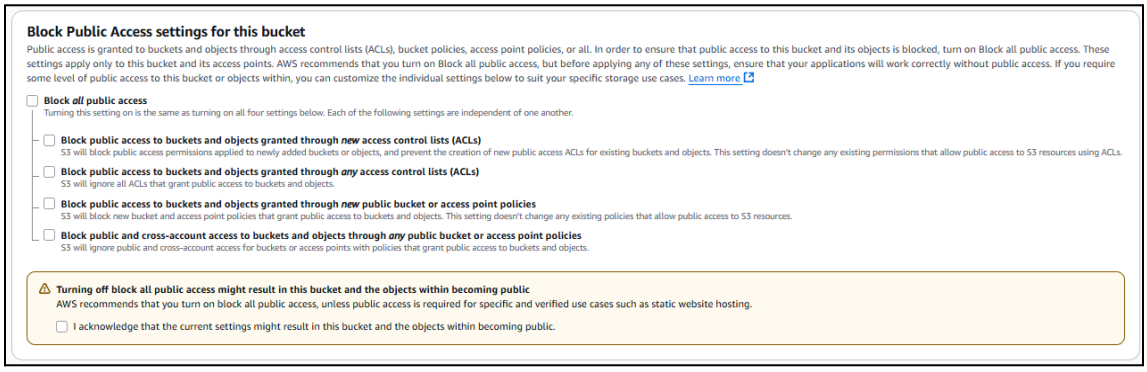
Steps to Create an S3 Bucket:

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

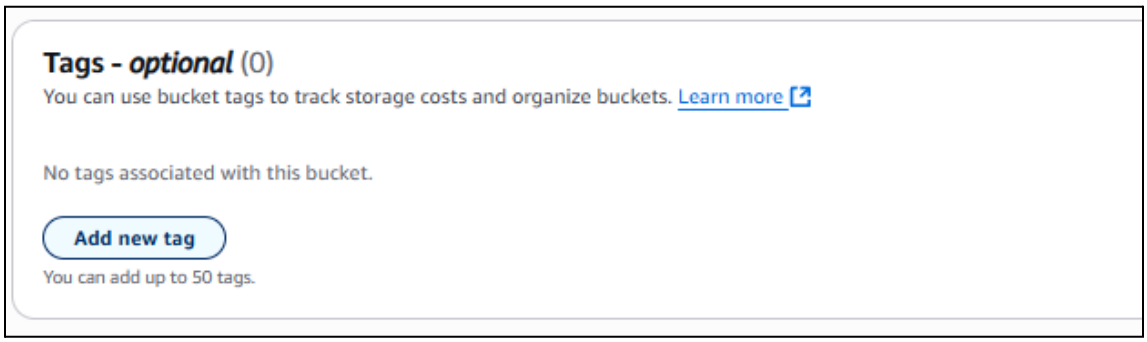


3. **Name your bucket.** Give the bucket a unique name. Since the name is part of the file's URL, it should be official-sounding and use hyphens instead of spaces (e.g., "mikes-macaroon-market").

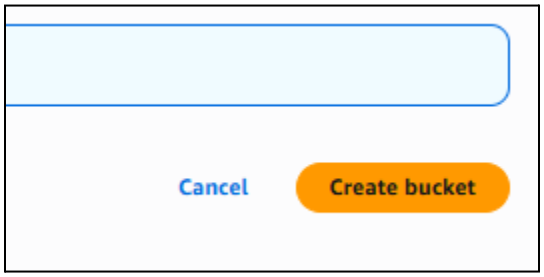
4. **Configure access.** Uncheck "Block all public access" to make the files in the bucket publicly available.



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.

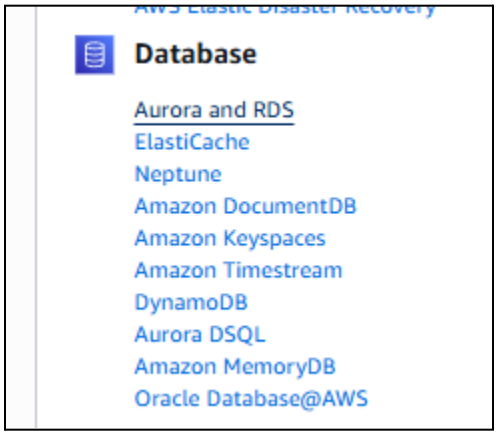
RDS - Creating a Database Instance

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.

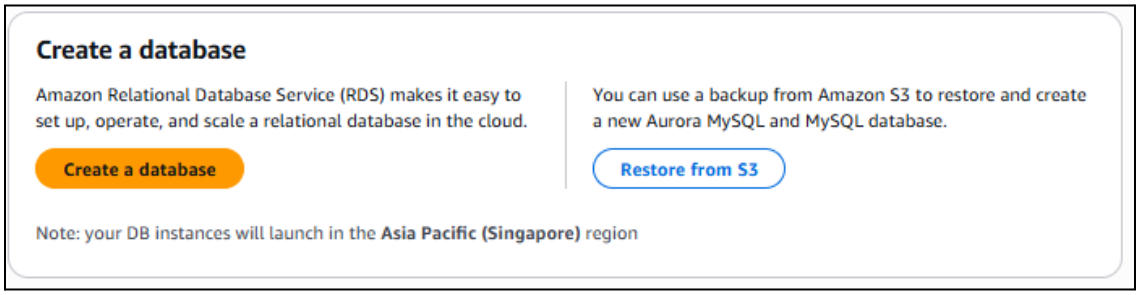
Note: Amazon Aurora has various pricing (pay-as-you-go, On-Demand, or Reserved Instance pricing) depending on your configuration and optional features you choose. Learn more about about its pricing here https://aws.amazon.com/rds/aurora/pricing

Steps to Create an RDS Database:

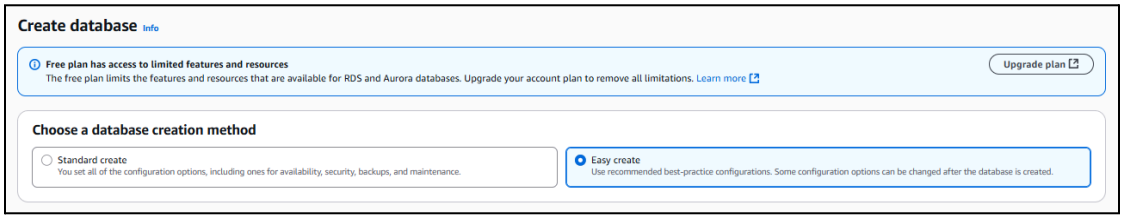
1. **Go to RDS.** Use the services menu (the nine-square button in the top left) and go to "Database," then "Aurora and RDS".



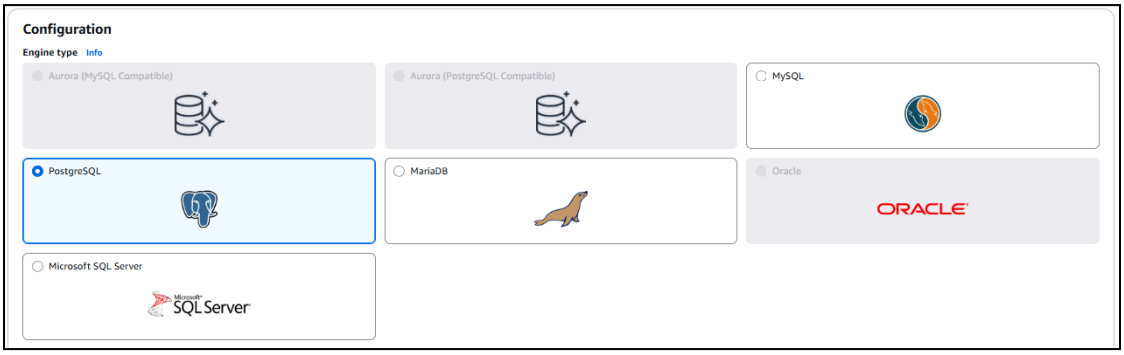
2. **Create a database.** Click the "Create Database" button.



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

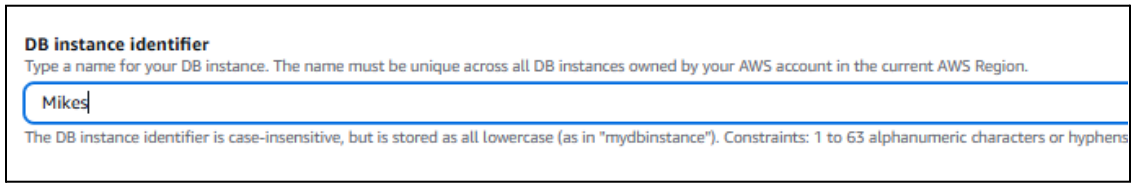


4. **Select a database type.** Choose the database type your application expects, such as **Postgres**.

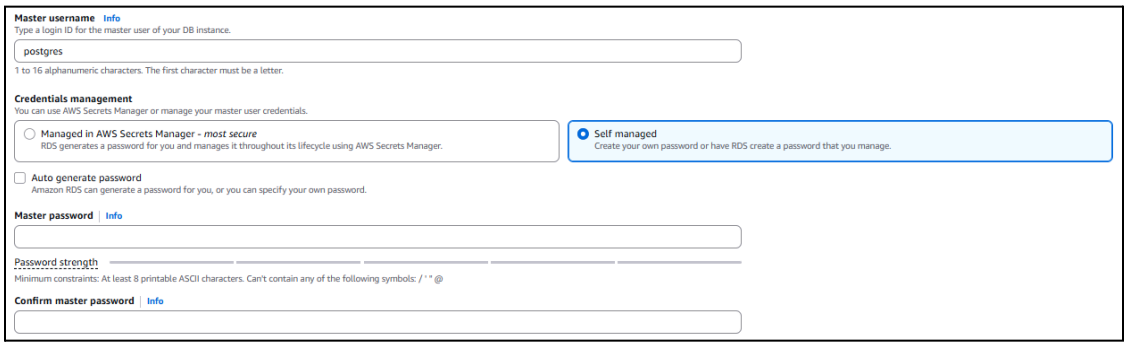


5. **Choose a size.** Select the smallest size to stay within the free tier.

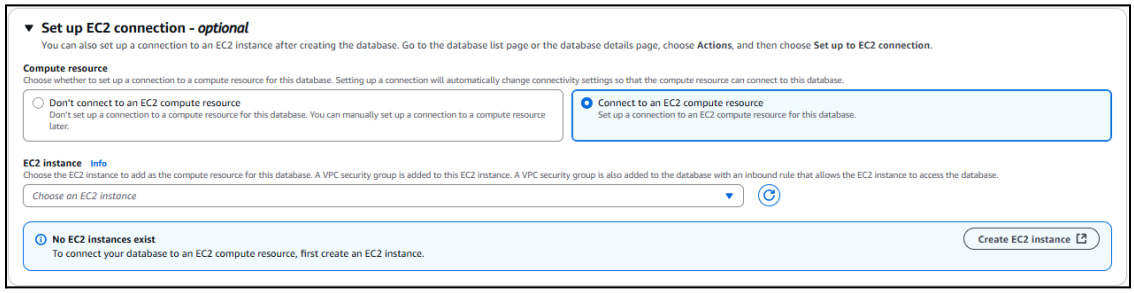
6. **Name the instance.** Give your database a name, like "Mike's Macaroon Market".



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.
    - **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`).

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.