

Sicat, Sean Russel

IV-ACSAD

Elec 3 Ass #2 Create a How-To document guide based on the AWS Tutorial video for Beginners Guide.

### **What AWS is and its key benefits**

Amazon Web Services (AWS) is a comprehensive cloud computing platform that provides on-demand access to computing resources, storage, databases, and other services over the internet. Instead of owning and maintaining physical servers, you can rent computing power and storage from AWS on a pay-as-you-use basis.

### **Step-by-Step Guide to Cloud Computing:**

#### **Signing up for AWS for free**

##### **1. Initial Setup**

- Go to aws.amazon.com and click Create a Free Account
- Enter your email address and choose an account name
- Verify your email address through the confirmation link

##### **2. Account Security**

- Create a strong password with uppercase, lowercase, numbers, and symbols
- Complete your personal information including full name and address

##### **3. Identity Verification**

- Verify your phone number with the SMS code sent by AWS
- This ensures account security and prevents automated signups

##### **4. Support Plan Selection**

- Select Basic Free Support
- Complete the Sign Up process

## **Setting a goal: Hosting a website on AWS**

For this tutorial, the goal is to host a website and it requires:

- **EC2 (Elastic Compute Cloud)** - Provides compute resources to run our web server
- **S3 (Simple Storage Service)** - Stores product images and static website file
- **RDS (Relational Database Service)** - Manages inventory data and customer orders

## **Exploring the AWS Management Console**

Getting Oriented:

### **1. Initial Login**

- Access the AWS Management Console using your account credentials
- You'll see the main dashboard with various widgets and quick access options

### **2. Region Selection**

- Choose the correct region (e.g., US East 1 – North Virginia)
- Select the region closest to your target users for better performance
- Different regions may have different pricing and service availability

### **3. Navigation Tools**

- Use the search bar to quickly find any AWS service
- Access services through the main Services menu organized by category
- Monitor cost usage and AWS health updates through dashboard widgets
- Pin frequently used services for quick access

## **EC2 - Launching a virtual server**

Virtual Server Setup Process:

### **1. Launch Configuration**

- Navigate to EC2 service and click Launch Instance
- Name your instance "Mike's Macaroon Market Website"

## **2. Operating System Selection**

- Choose Amazon Linux
- This provides a secure, optimized Linux environment

## **3. Instance Specifications**

- Select t2.micro as the instance type
- This provides 1 vCPU and 1 GB of memory, sufficient for our tutorial

## **4. Security Configuration**

- Proceed without a key pair for tutorial simplification
- Configure security group to allow HTTP traffic on port 8080
- This enables web traffic to reach your application

## **5. Storage and Launch**

- Keep default storage settings (8GB General Purpose SSD)
- Review configuration and click Launch Instance

## **S3 - Configuring file storage**

Bucket Creation and Configuration:

### **1. Create Storage Bucket**

- Search for S3 service and click Create Bucket
- Name the bucket uniquely (e.g., mikes-macaroon-market-yourname)
- Bucket names must be globally unique across all AWS users

### **2. Access Configuration**

- Allow public access so website images can be served to visitors
- This is necessary for a public-facing website with image content

### **3. Organization and Metadata**

- Add tags to organize resources and track costs
- Tags help categorize resources by project, environment, or team

#### **4. Security Setup**

- Create the bucket and edit its Bucket Policy to allow public read access
- This enables your website to display images stored in S3

#### **5. Content Management**

- Upload images manually through the console interface
- Alternatively, configure automatic uploads from your website application

#### **6. API Access**

- Create an Access Key and Secret Key for programmatic access
- These credentials allow your website to connect securely to S3

### **RDS - Creating a database instance**

#### **Database Service Setup:**

##### **1. Database Creation**

- Navigate to RDS service and click Create Database
- Choose Easy Mode for simplified configuration options

##### **2. Database Engine**

- Select PostgreSQL as the database engine
- PostgreSQL provides robust features for web applications

##### **3. Instance Configuration**

- Select the smallest available size to stay within Free Tier limits
- Free Tier provides 750 hours of database usage per month

##### **4. Database Identification**

- Name the database "mikes-macaroon-market"
- Use descriptive names that match your project structure

##### **5. Access Credentials**

- Generate username and password credentials automatically

- Copy and securely store these credentials for application configuration

## 6. Network Configuration

- Enable EC2 connectivity so your web server can communicate with the database
- This creates secure network connections between services

## 7. Deployment

- Click Create Database and wait for deployment (5-10 minutes)
- Copy the database endpoint URL when the instance becomes available

### Connecting to an EC2 instance and running commands

#### Server Access and Application Deployment:

##### 1. Server Connection

- Open EC2 Instance Connect through the AWS console
- Use the built-in SSH terminal that runs in your web browser
- No additional software installation required

##### 2. Development Environment Setup

- Install essential development tools: Git, Node.js, and npm
- These tools enable you to download and run the website application

##### 3. Application Download

- Clone the website's GitHub repository to your server
- This downloads all the website code and configuration files

##### 4. Environment Configuration

- Configure environment variables with your specific AWS resource details:
  - S3 bucket name and access credentials
  - Database endpoint URL and password
  - Any other application-specific settings

## **5. Application Installation**

- Install required Node.js packages using npm install
- This downloads all software dependencies needed by the website

## **6. Application Launch**

- Run the website using npm start & disown command
- The & disown ensures the application continues running after you disconnect

## **7. Website Access**

- Access your live website using the EC2 instance's public IP address on port 8080
- Share this URL to show others your deployed application

## **Tips for cost savings on AWS**

### **Cost Optimization Strategies:**

#### **Free Tier Management**

- Use Free Tier eligible services whenever possible
- Monitor Free Tier usage limits through the billing dashboard

#### **Resource Management**

- Shut down or terminate instances when not actively needed
- Stopping instances saves compute costs while preserving your data

#### **Geographic Optimization**

- Choose regions with lower pricing (avoid US West 1 which has higher costs)
- Balance cost savings with performance requirements for your users

#### **Financial Controls**

- Set up billing alerts and spending limits to prevent unexpected charges
- Configure notifications at multiple spending thresholds (\$5, \$10, \$25)

#### **Advanced Cost Strategies**

- Consider prepaying for Reserved Instances if you have predictable usage

- Use Spot Instances for development and testing workloads to save up to 90%

### **Continuing your AWS journey with AI, Machine Learning, and AWS Certifications**

After finishing the video it helped me understand that there is much more to learning AWS than just hosting a website. Learning the fundamentals is just the beginning, upon completing the basics of EC2, S3, and RDS, I can easily go into more advanced fields such as Artificial Intelligence and Machine Learning. Obtaining AWS Certifications can help me enhance my technical skills and help me cultivate new opportunities for future career growth in cloud computing.