AWS Cloud DevSecOps [Part1]

By Wilfredo Paulo A. Perez III

Overview

In this combined AWS project, I explored and implemented two core services:

1. **AWS IAM & EC2 for identity and access control**
2. **Amazon S3 for static website hosting**

Part 1: IAM Security and EC2 Management

Project Goal

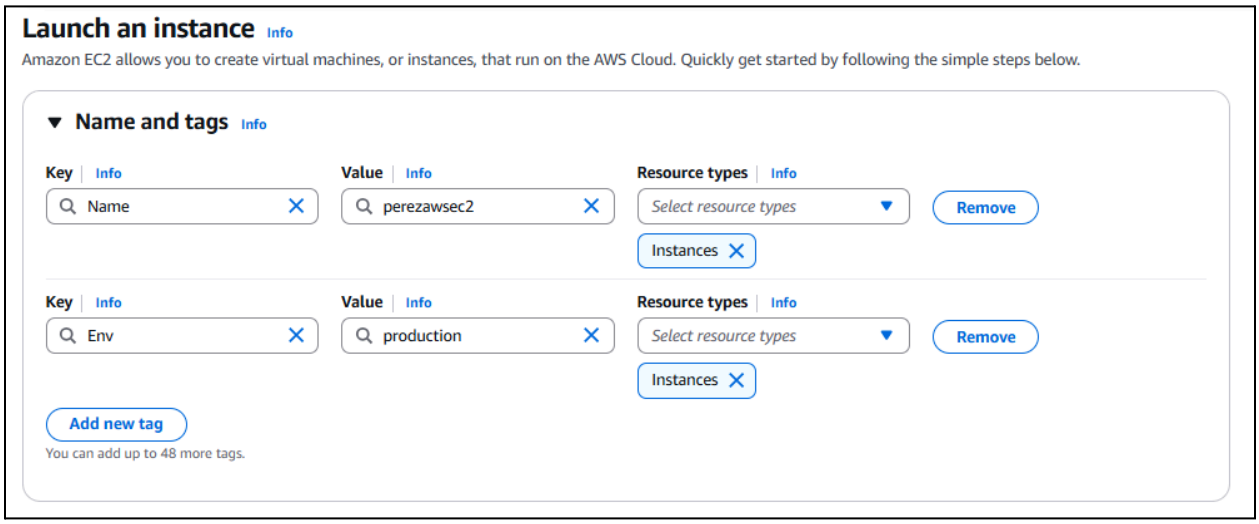
The objective was to learn how AWS IAM controls authentication and authorization across cloud resources, specifically focusing on EC2 instances tagged by environment (`development` or `production`).

Step 1: Launch EC2 Instances

I launched two Amazon EC2 instances:

- One tagged as `Environment: development`

- Another as `Environment: production`



Step 2: Tagging for Identification

Used **tags** to classify instances:

- `Key: Env`
- `Value: development` or `production`

Tags allow filtering and management of similar resources easily.

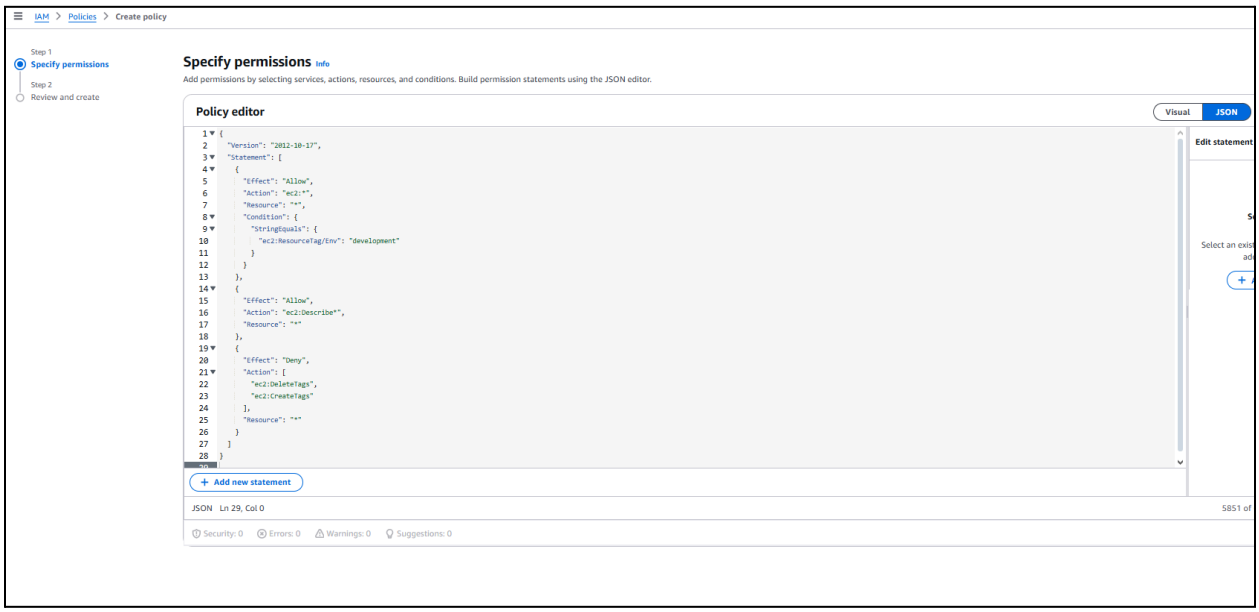
Step 3: Create IAM Policy with Tag-Based Access

I created a **custom IAM policy** using JSON, where:

- Users could `Start`, `Stop`, or `Describe` EC2 instances **only** if tagged as `development`.
- All users were **denied** the ability to create or delete tags.

IAM JSON Policy Structure:

- `Effect`: Allow/Deny
- `Action`: ec2:StartInstances, ec2:StopInstances, etc.
- `Resource`: Instances tagged as `Env=development`

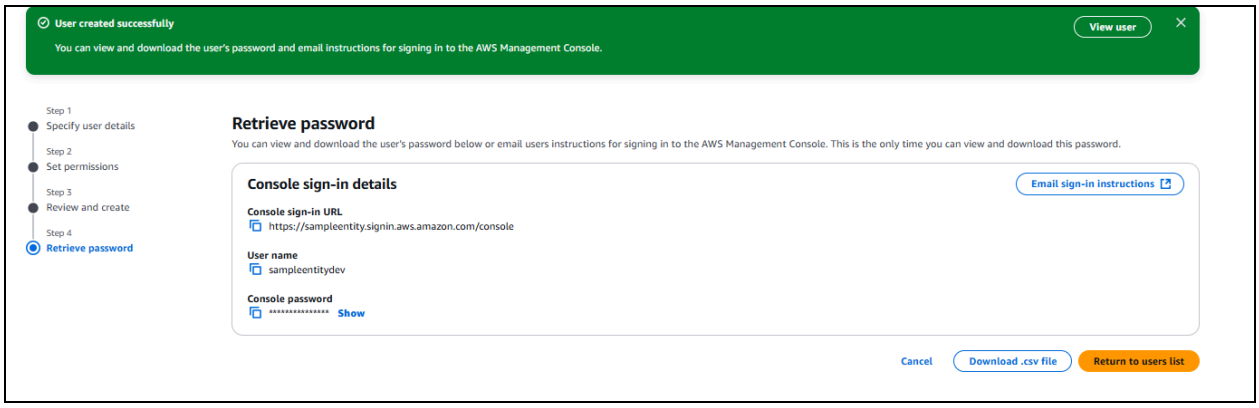


Step 4: IAM User and Group Setup

Created a new IAM user and added them to a user group with the above policy attached.

Benefits of using **groups**:

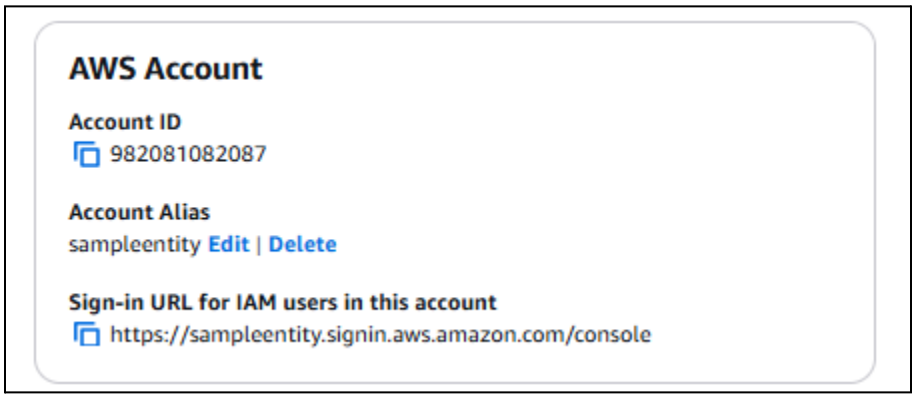
- Simplified permission management
- Scalability when managing multiple users



Step 5: Create Account Alias

I created a **friendly alias** for the AWS sign-in URL, making it easier to access:

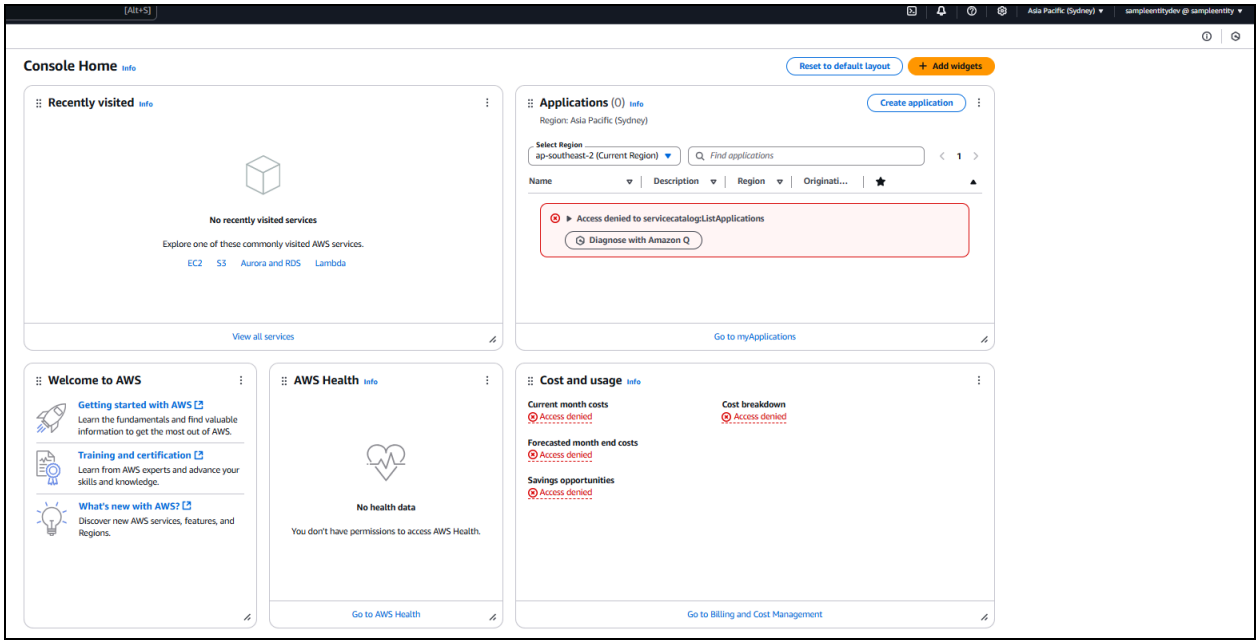
- `https://sampleentity.signin.aws.amazon.com/console` (no longer available to avoid charges)



Step 6: Testing IAM Access

Testing results:

- Able to stop the **development** EC2 instance.
- Blocked when attempting to stop the **production** EC2 instance (as expected).

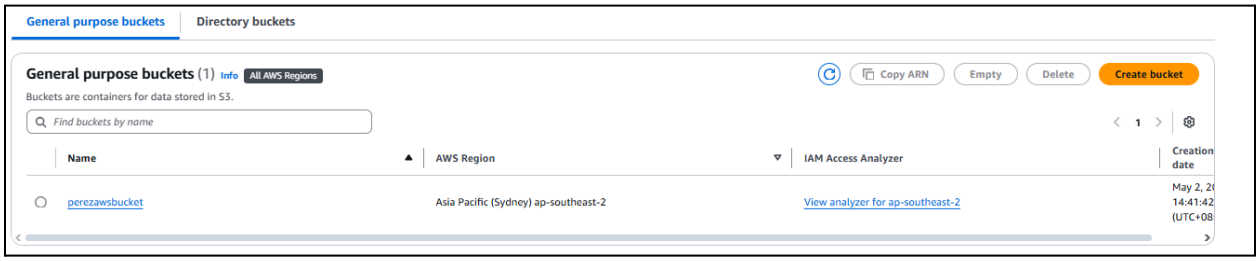

[Alt+S]
Asia Pacific (Sydney)
sampleentitydev @ sampleentity
Console Home Info
Reset to default layout
+ Add widgets
Recently visited Info
No recently visited services
Explore one of these commonly visited AWS services.
EC2 S3 Aurora and RDS Lambda
View all services
Applications (0) Info
Region: Asia Pacific (Sydney)
Create application
Select Region
ap-southeast-2 (Current Region)
Find applications
Name
Description
Region
Originati...
Access denied to servicecatalog:ListApplications
Diagnose with Amazon Q
Go to myApplications
Welcome to AWS
Getting started with AWS
Learn the fundamentals and find valuable information to get the most out of AWS.
Training and certification
Learn from AWS experts and advance your skills and knowledge.
What's new with AWS?
Discover new AWS services, features, and Regions.
AWS Health Info
No health data
You don't have permissions to access AWS Health.
Go to AWS Health
Cost and usage Info
Current month costs
Access denied
Cost breakdown
Access denied
Forecasted month end costs
Access denied
Savings opportunities
Access denied
Go to Billing and Cost Management


Failed to stop the instance i-0c9320aa130ea0f43
Diagnose with Amazon Q
You are not authorized to perform this operation. User: arn:aws:iam::640168412593:user/nextwork-dev-maya is not authorized to perform: ec2:StopInstances on resource: arn:aws:ec2:ap-southeast-2:640168412593:instance/i-0c9320aa130ea0f43 because no identity-based policy allows the ec2:StopInstances action. Encoded authorization failure message:
iLfdLoeBzLf0NSprCXyCJE1I123QJe3_4HZSdMGO9hZWR-
fEa10hjqKe6sAcnDmRrlbuePzHQpYQBKhKDJlVdovF8Z13e4mgFFDLJu45wFOrMnBLrCtyMTfbFWtcxMUJSHGrDFGeNWqMzf_UbB6
YQmjMZICjMKBBXolw9TLBr68KckpHFqBgGbxfPh7qtRi0TcSi0RHD32aZFqY6XMDc8HJFyfCWp1XCd4wTYm9p0jk5qxW4YbaJ2z9Jt-
iMQP41rg54661agBhngxRbfNarUzgBU6F4GA7k6veI5ln3OKLFD30sPlIDoKCNKiQbFxXlq9aczLz8ZlU-
UyFk1QFMNBwHv7XMcZXgz2ebjXmTZ5menftxKUcpf_MiPBb5wmMTmPjZzPQxDtVA_l8DX57f2VUZ0oDSRNNzQqcmkj015mQHb
NYg1fPsJNRZtd2m39GmqZJyIc-xsX-
satVk688fQvUbUanEHpgrabVMnmCDssZe4VoLfXhqLforBGgKWQuV_o6NG1d7b9fyb34iv99Cg9Hhpb8UJgxkyP8SRtOwxtIa1QfSo
DE4PGxAy2xdVVYxDQNKvt8AP1QVFtJnDsltNOTI2Xcmz8dSDkD9CSKV25OHVZPwn4kFD5lkWR0_cpVrqeIKvFxcCZS0drbAPvrH8Uz
WO5EiHFAbjuYUiGxj3f55XiKBolxmQPntSsBUCiH-I4JgKTGC3fHT9u8Vt0ghB3R81S4pAzt7HlSPK2QoEukw-
4HoRpWdgpoRk_LtkACs6uioTkB1HpmiTFZMrI7aNw-Tw-
jbcUx9E0YVQdyILAwGAqipvZCnqEzbiNfLwe9bbLBqIjTJ9sPcuSVDpXmrXDu1It-WV-
yJ8JWvgxHwaE7AfXg4JXhQH9RWnWND4zud762L3-Hq

Part 2: Hosting a Static Website on Amazon S3

Step 1: Create an S3 Bucket

- Created a new bucket with a globally unique name.
- Chose the **Asia Pacific (Sydney)** region (`ap-southeast-2`).

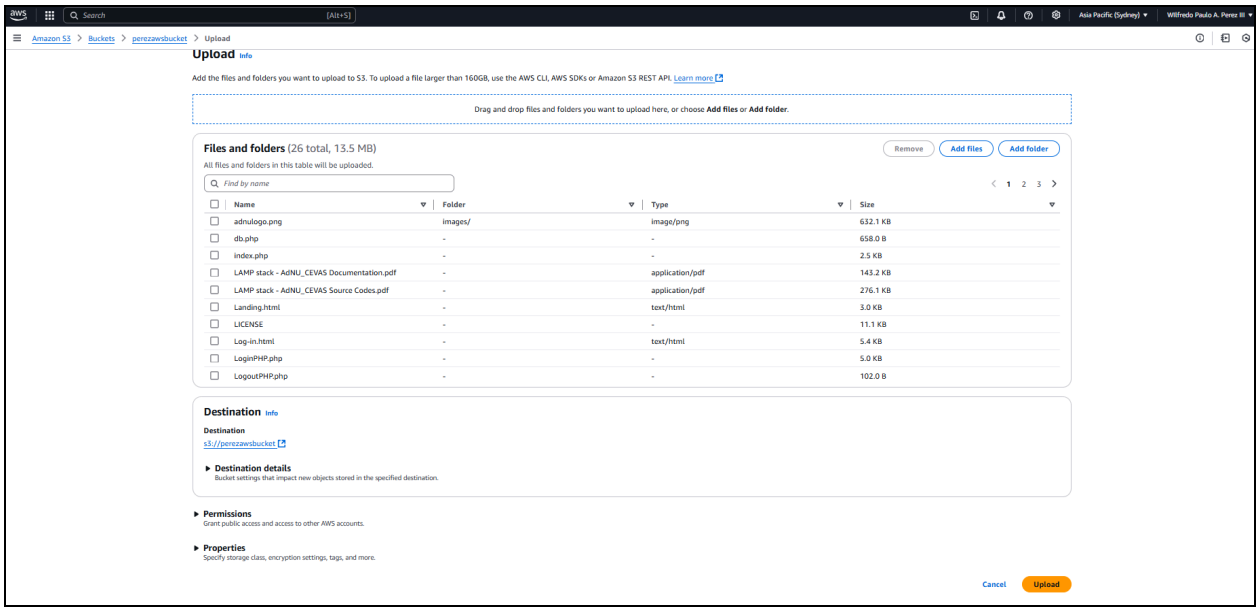


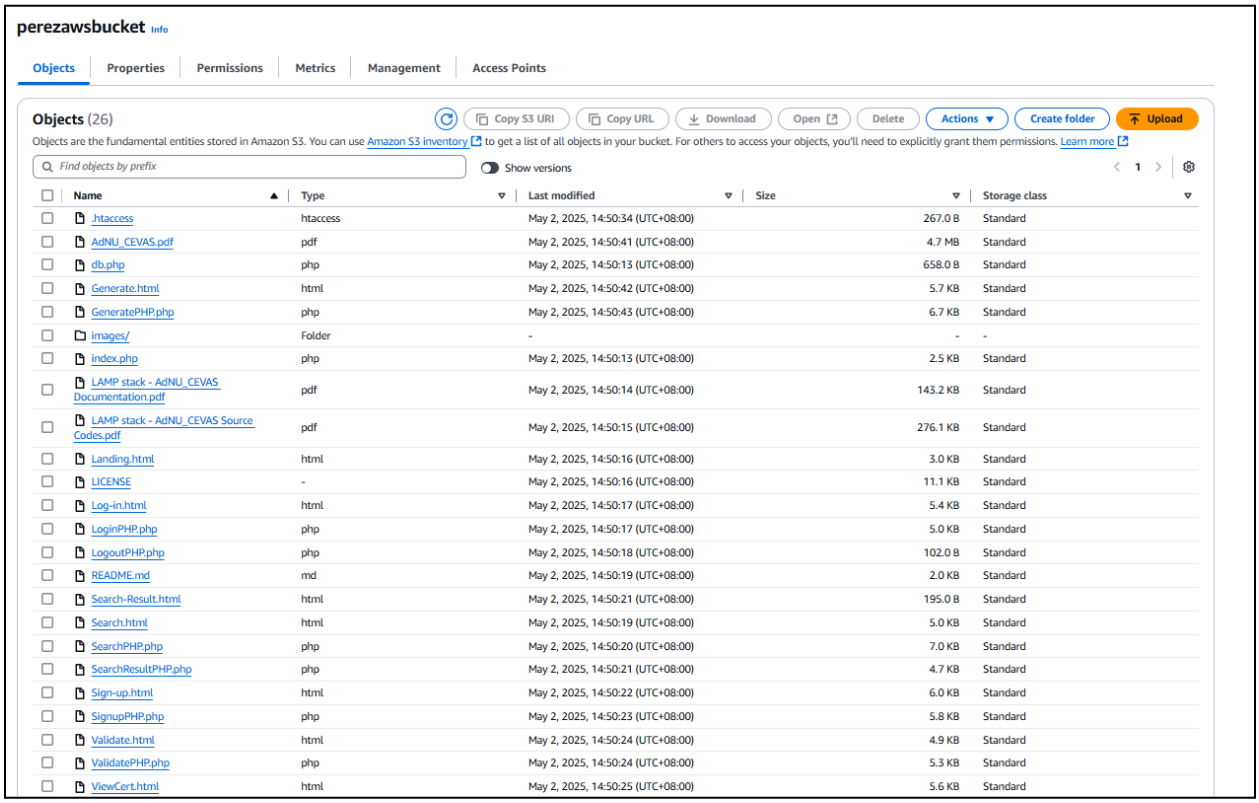
Step 2: Upload Website Files

Uploaded:

- `index.html`
- Several image files

These files formed the core of my static website.

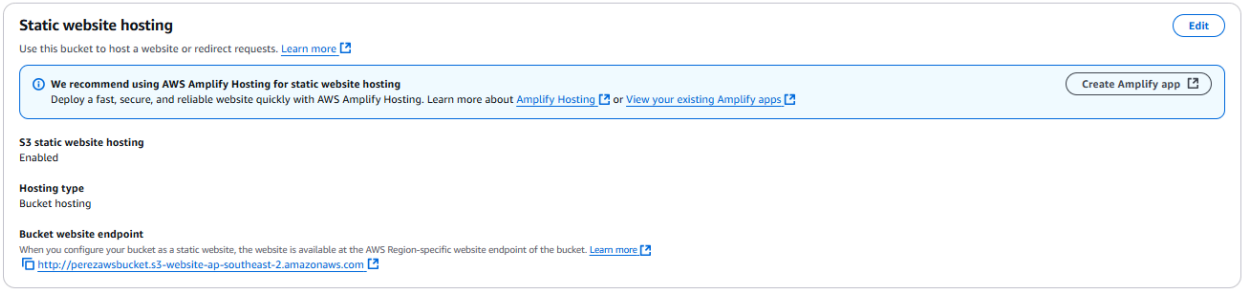




Step 3: Enable Static Website Hosting

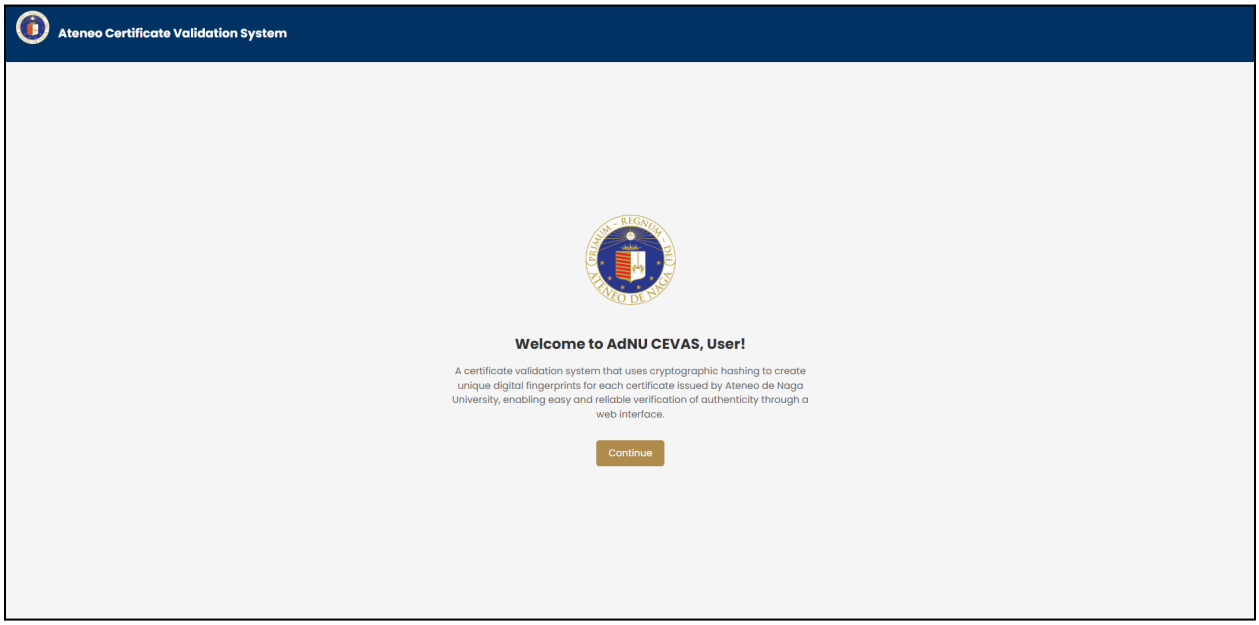
Enabled static website hosting on the bucket and got a **bucket endpoint URL**.

Initially encountered a `403 Forbidden` error because all S3 objects are private by default.



Step 4: Make Files Public via ACL

Changed the **Access Control List (ACL)** of the files to **public**, allowing the entire world to view my website.

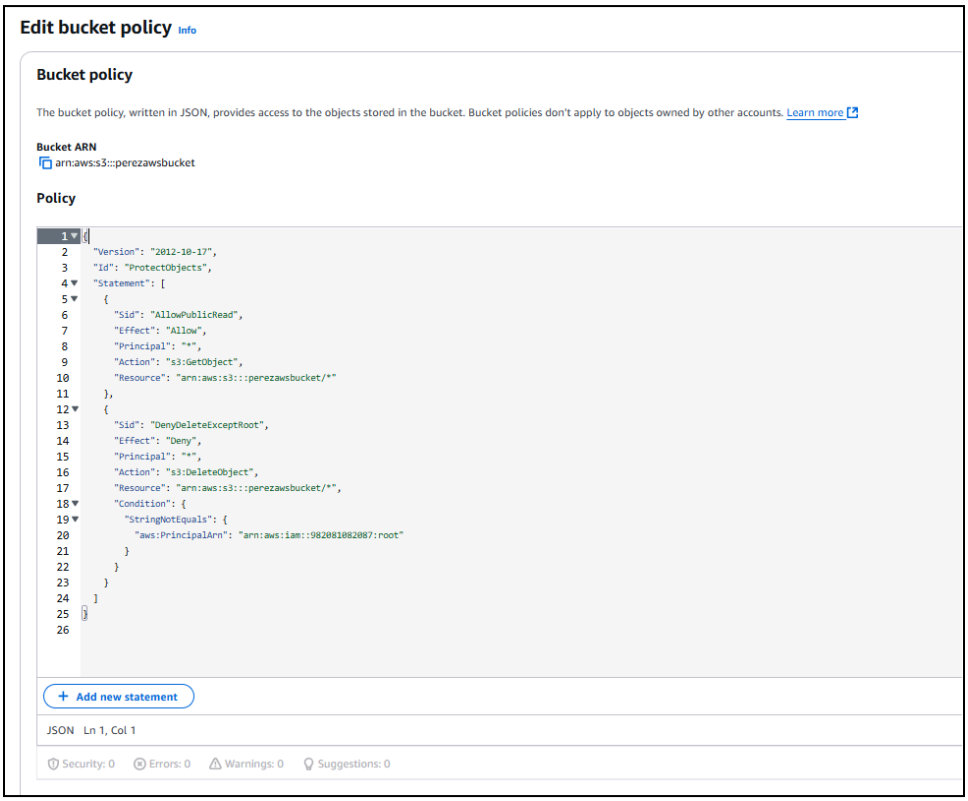


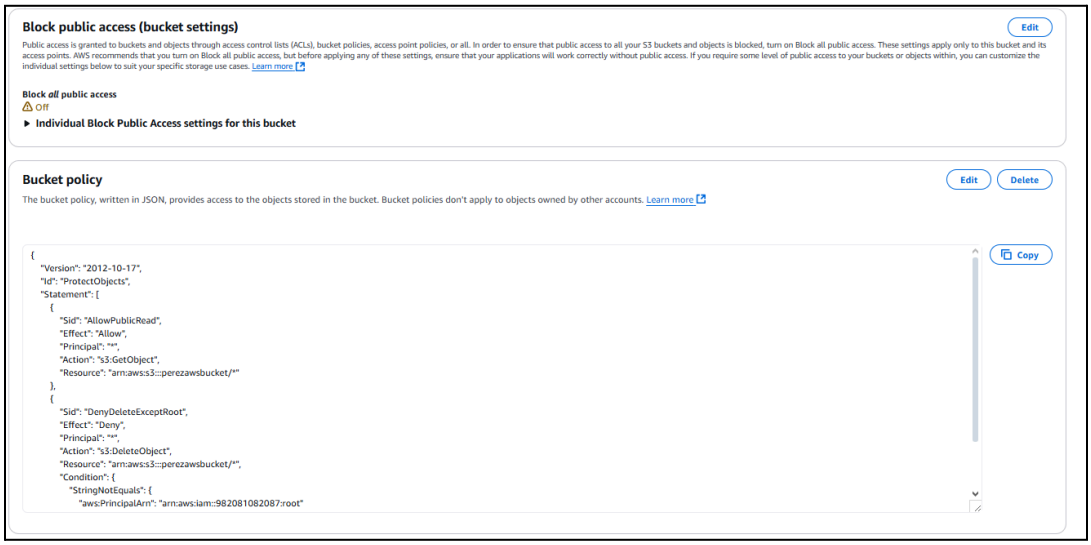
Step 5: Add Secure Bucket Policy

To secure the bucket:

- I added a **bucket policy** that **prevents deletion of files** by anyone except myself.

This protected the site from accidental or unauthorized deletions.





Project Reflection

IAM Security Project

- **Time taken**: ~30 minutes
- **Challenge**: Configuring the IAM user correctly
- **Reward**: Seeing the IAM restrictions in action

S3 Hosting Project

- **Time taken**: ~20 minutes
- **Challenge**: Crafting the bucket policy for secure file deletion
- **Reward**: Hosting a fully working public website

Insights

These two projects helped me:

- Understand **IAM roles, policies, and tag-based access control**
- Successfully **host a secure and public static website** on S3

Through hands-on experience, I now have a better grasp of AWS's identity management and storage capabilities.