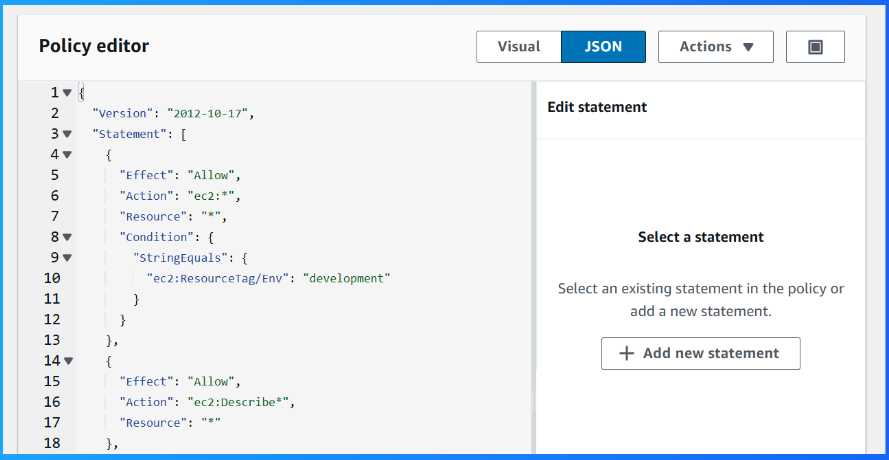
Cloud Security with AWS IAM



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Env": "development"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    },
```

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

Introducing today's project!

What is AWS IAM?

AWS IAM is a service that lets you manage who can access your AWS resources and what actions they can perform. It helps secure your environment by giving users only the permissions they need, supporting the principle of least privilege.

How I'm using AWS IAM in this project

In today's project, I used AWS IAM to create and manage users, groups, and policies to control access to specific AWS resources. I ensured that each user had the appropriate permissions, following the principle of least privilege for better security.

One thing I didn't expect...

One thing I didn’t expect in this project was how quickly IAM policies can become complex when managing multiple users and permissions.Ensuring correct level of access while avoiding overly permissive policies required careful planning and attention.

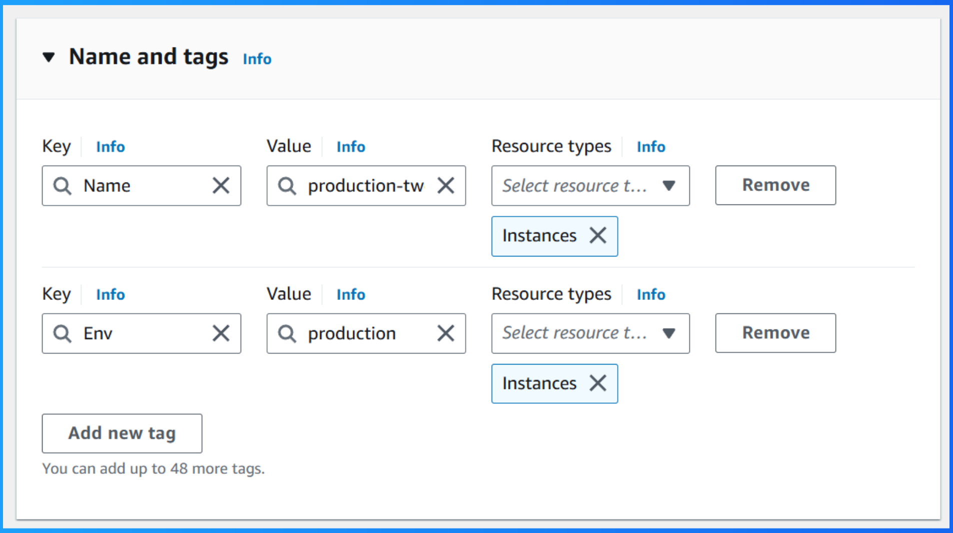
This project took me...

This project took me a few hours to complete. It involved setting up IAM users, groups, and policies, as well as testing access controls to ensure everything was configured correctly.

Tags

Tags are labels to help AWS Account users identify and manage their resources.Also for grouping,mass management and applying security policies.

The tag I've used on my EC2 instances is called Env. The value I've assigned for my instances are production and development.This reperesents the two different environments that we are using to build and release the app.



Name and tags Info

Key	Value	Resource types	
Name	production-tw	Instances	Remove
Env	production	Instances	Remove

Add new tag

You can add up to 48 more tags.

IAM Policies

An IAM policies are rules for who can do what with your AWS resources. It's all about giving permissions to IAM users, groups, or roles, saying what they can or can't do on certain resources, and when those rules kick in.

The policy I set up

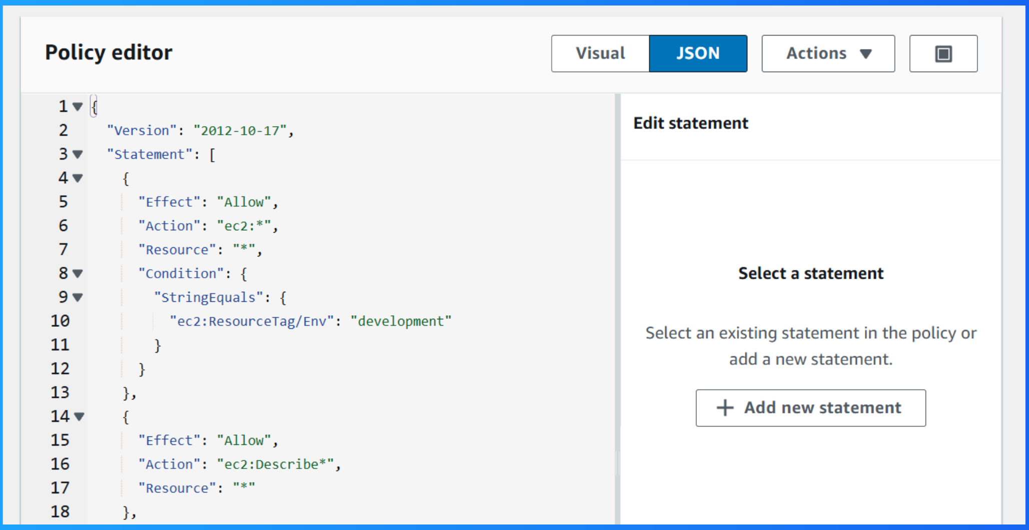
For this project, I've set up a policy using Json editor.

I've created a policy that restricts access to specific AWS resources by defining precise permissions. This enforces least-privilege access, limiting users to only necessary actions for improved security.

When creating a JSON policy, you have to define its Effect, Action and Resource.

The Effect, Action, and Resource attributes of a JSON policy mean controlling access: "Effect" allows or denies actions, "Action" specifies permitted operations, and "Resource" designates targeted resources. Together, they define precise permissions.

My JSON Policy



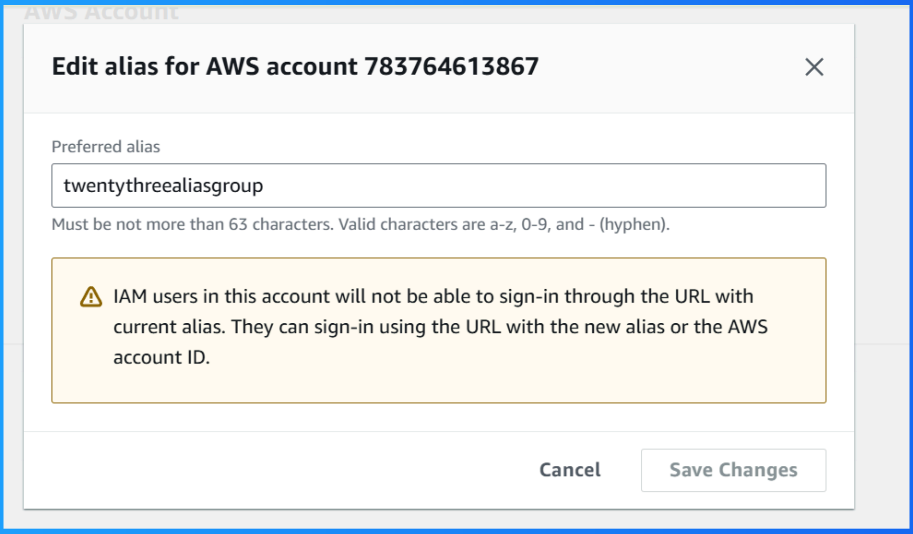
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Env": "development"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    },
```

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

Account Alias

An account alias is a customizable, user-friendly name for your AWS account, replacing the default account ID in the AWS console URL. This makes the URL easier to remember and share with users in your organization.

Creating an account alias took me just a few minutes. Now, my new AWS console sign-in URL is easier to remember and access with the customized alias.



Edit alias for AWS account 783764613867

Preferred alias

twentythreealiasgroup

Must be not more than 63 characters. Valid characters are a-z, 0-9, and - (hyphen).

IAM users in this account will not be able to sign-in through the URL with current alias. They can sign-in using the URL with the new alias or the AWS account ID.

IAM Users and User Groups

Users

IAM users are individuals or services with their own login credentials, granted specific permissions to access AWS resources.

User Groups

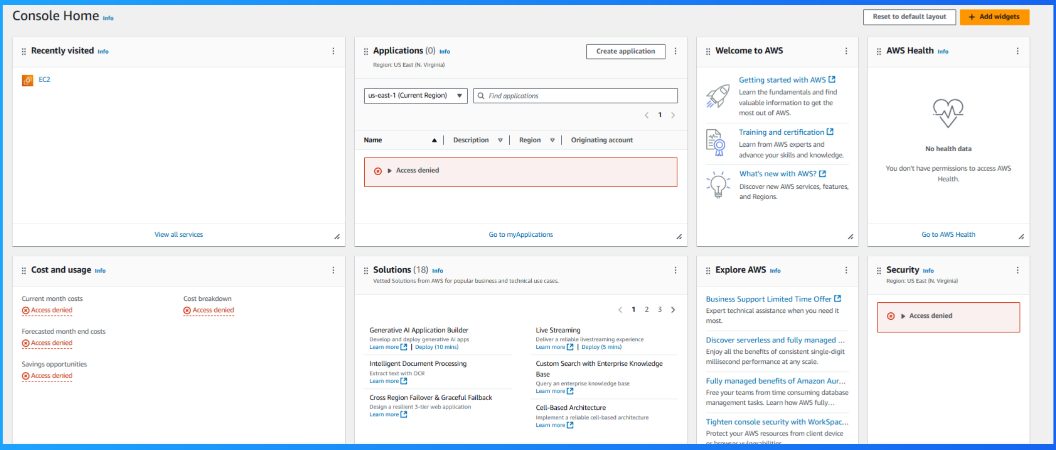
IAM users that simplify permission management by allowing policies to be applied to multiple users at once. This helps streamline access control, as permissions can be adjusted for the entire group rather than individual users.

I attached the policy I created to this user group, which means all users in the group get the same permissions. This simplifies managing access for multiple users at once.

Logging in as an IAM User

The first way is by sending the sign-in details via email, including the username and a temporary password. The second way is by providing the details through a secure messaging platform or shared document, ensuring the information remains private.

Once I logged in as my IAM user, I noticed limited access to AWS resources and services. This was because the IAM user only had the permissions granted by their assigned policies.



Recently visited Info

EC2

View all services

Applications (0) Info

Region: US East (N. Virginia)

Create application

us-east-1 (Current Region)

Find applications

Name	Description	Region	Originating account

Access denied

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

Solutions (18) Info

Vetted Solutions from AWS for popular business and technical use cases.

Generative AI Application Builder
Develop and deploy generative AI apps
Learn more | Deploy (10 mins)

Live Streaming
Deliver a reliable livestreaming experience
Learn more | Deploy (5 mins)

Intelligent Document Processing
Extract text with OCR
Learn more

Custom Search with Enterprise Knowledge Base
Query an enterprise knowledge base
Learn more

Cross Region Failover & Graceful Failback
Design a resilient 3-tier web application
Learn more

Cell-Based Architecture
Implement a reliable cell-based architecture
Learn more

Explore AWS Info

Business Support Limited Time Offer
Expert technical assistance when you need it most.

Discover serverless and fully managed ...
Enjoy all the benefits of consistent single-digit millisecond performance at any scale.

Fully managed benefits of Amazon Aur...
Free your teams from time consuming database management tasks. Learn how AWS fully...

Tighten console security with WorkSpac...
Protect your AWS resources from client device

Security Info

Region: US East (N. Virginia)

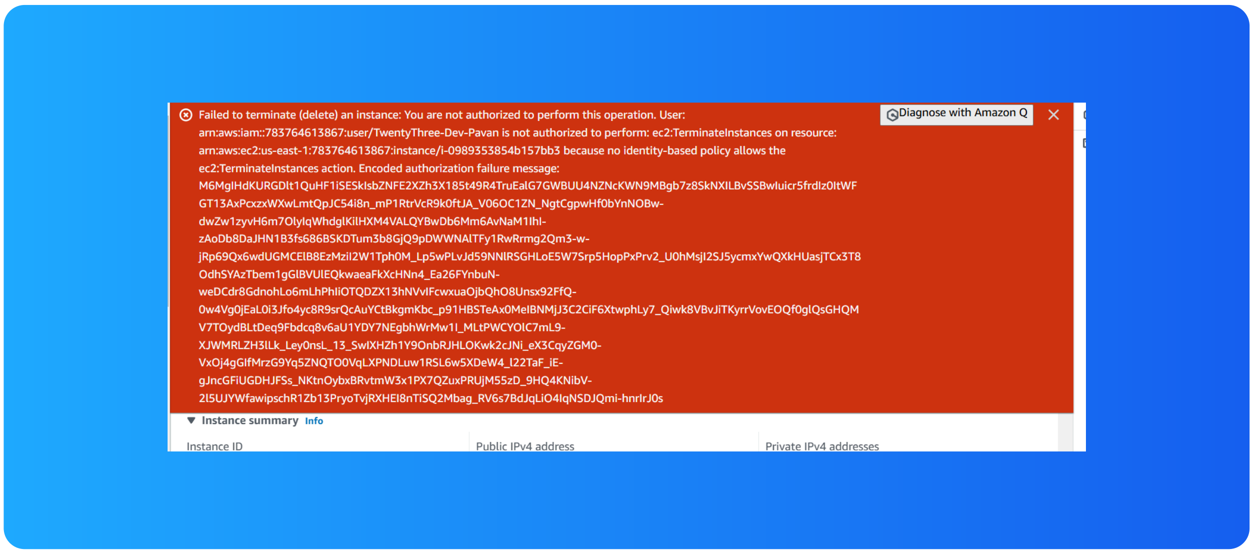
Access denied

Testing IAM Policies

I tested my JSON IAM policy by performing actions on my two EC2 instances, such as starting, stopping, or modifying their settings. This helped verify whether the permissions in the policy were correctly applied to the instances.

Stopping the production instance

When I tried to stop the production instance, I received an error or was denied access. This was because my IAM user didn't have the necessary permissions to manage or stop instances, as defined by the policies assigned to the user.

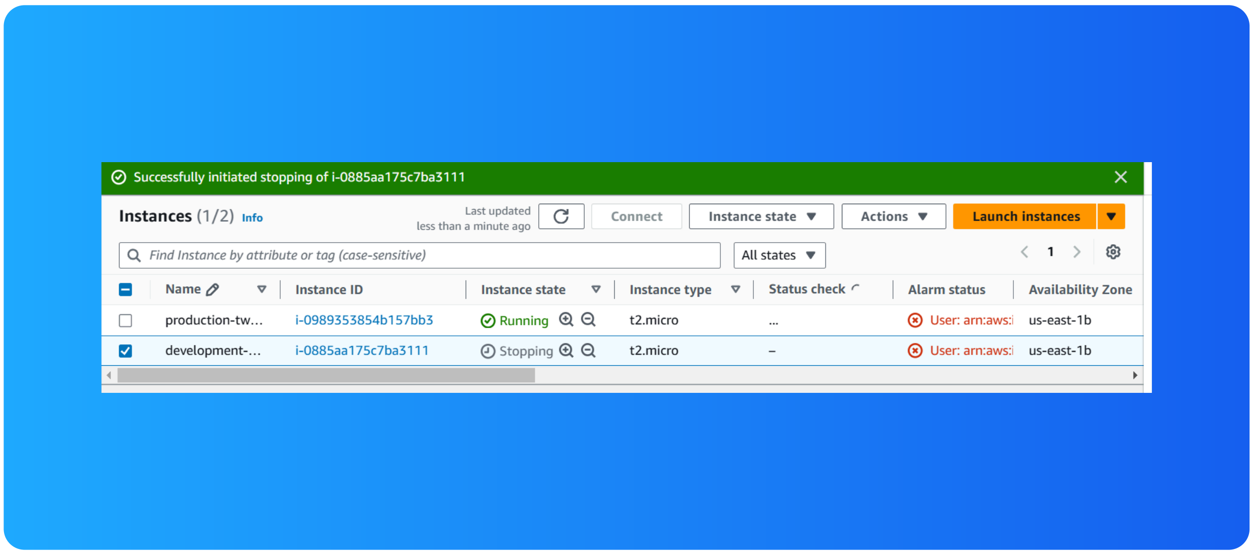


```
Failed to terminate (delete) an instance: You are not authorized to perform this operation. User:
arn:aws:iam::783764613867:user/TwentyThree-Dev-Pavan is not authorized to perform: ec2:TerminateInstances on resource:
arn:aws:ec2:us-east-1:783764613867:instance/i-0989353854b157bb3 because no identity-based policy allows the
ec2:TerminateInstances action. Encoded authorization failure message:
M6MgIHdKURGDlt1QuHF1iSESkIsbZNFE2XZh3X185t49R4TruEalG7GWBUU4NZNcKWN9MBgb7z8SkNXILBvSSBwIuicr5frdIz0ItWF
GT13AxPcxzxWXwLmtQpJC54i8n_mP1RtrVcR9k0ftJA_V06OC1ZN_NgtCgpwHf0bYnNOBw-
dwZw1zyvH6m7OlyIqWhdglKilHXM4VALQYBwDb6Mm6AvNaM1Ihl-
zAoDb8DaJHN1B3fs686BSKDTum3b8GjQ9pDWWNAlTFy1RwRrmg2Qm3-w-
jRp69Qx6wdUGMCElB8EzMziI2W1Tph0M_Lp5wPLvJd59NNlRSGHLoE5W7Srp5HopPxPrv2_U0hMsjI2SJ5ycmxYwQXkHUasjTCx3T8
OdhSYAzTbem1gGlBVUlEQkwaeaFkXcHNn4_Ea26FYnbuN-
weDCdr8GdnohLo6mLhPhIiOTQDZX13hNVvIFcwxuaOjbQhO8Unsx92FfQ-
0w4Vg0jEaL0i3Jfo4yc8R9srQcAuYCtBkgmKbc_p91HBSTeAx0MeIBNMjJ3C2CiF6XtwphLy7_Qiwk8VBvJiTKyrrVovEOQf0glQsGHQM
V7TOydBLtDeq9Fbdcq8v6aU1YDY7NEgbhWrMw1I_MLtPWCYOlC7mL9-
XJWMRLZH3lLk_Ley0nsL_13_SwIXHZh1Y9OnbRJHLOKwk2cJNi_eX3CqyZGM0-
VxOj4gGIfMrzG9Yq5ZNQTO0VqLXPNDLuw1RSL6w5XDeW4_l22TaF_iE-
gJncGFiUGDHJFSs_NKtnOybxBRvtmW3x1PX7QZuxPRUjM55zD_9HQ4KNibV-
2l5UJYWfawipschR1Zb13PryoTvjRXHEI8nTiSQ2Mbag_RV6s7BdJqLiO4IqNSDJQmi-hnrIrJ0s
```

Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance, I was either allowed or denied access based on my IAM policy permissions. This was because the policy assigned to my IAM user defined whether I had the necessary rights to stop the instance.



Successfully initiated stopping of i-0885aa175c7ba3111

Instances (1/2) Info

Last updated less than a minute ago

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☐	production-tw...	i-0989353854b157bb3	Running	t2.micro	...	User: arn:aws:i	us-east-1b
☑	development-...	i-0885aa175c7ba3111	Stopping	t2.micro	–	User: arn:aws:i	us-east-1b