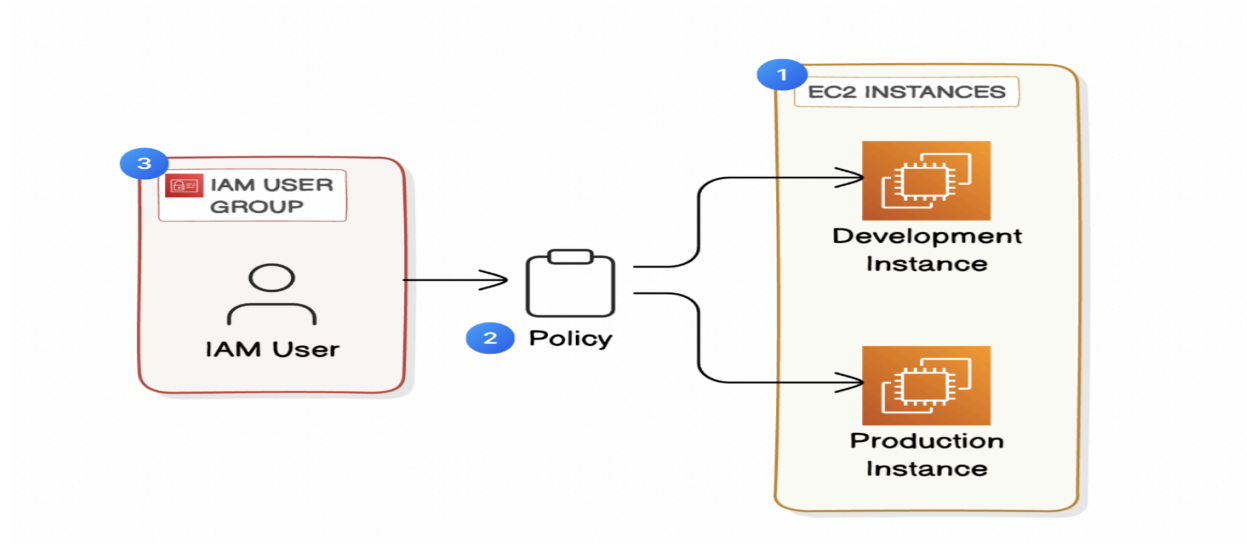




CLOUD SECURITY WITH AWS IAM

By Edikan Sam



PROJECT INTRODUCTION

WHAT IS AWS IAM?

AWS IAM (Identity and Access Management) is a service that securely manages user access to AWS resources. It helps control permissions, enhance security, and support compliance efforts.

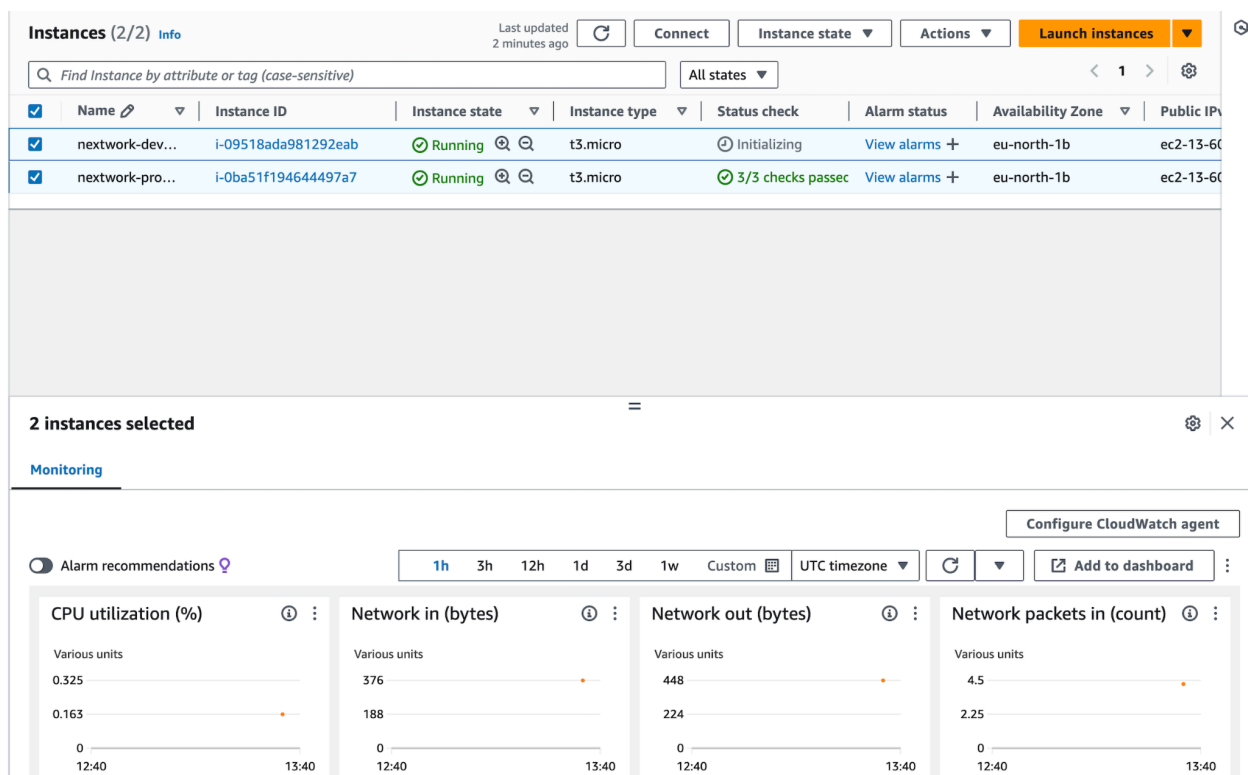
HOW I'M USING AWS IAM IN THIS PROJECT

I am using AWS IAM to create user groups, assign specific permissions to instances and attach policies.

One thing I didn't expect from this project was the intricate management of IAM policies and permissions, which demanded careful thought to provide users with appropriate access while maintaining security. This project took approximately 120 minutes to complete.

TAGS

EC2 tags function like labels that you can assign to your Amazon EC2 instances, which are virtual servers in the cloud. Each tag has a key and a value, making it easier to organize and manage your resources. For instance, you can use tags to indicate which instance is associated with a particular project, who oversees it, or its environment (such as development or production which I used on my EC2 instances). This simplifies the process of locating, sorting, and monitoring your servers, especially when you have numerous instances running in your AWS account.



IAM POLICIES

IAM policy defines who gets access to AWS resources and what actions they can perform. It grants users, groups, or roles permissions, specifying what they are allowed or not allowed to do and when these rules apply.

For this project, I have set up a policy using JSON.

I have created a policy that grants full EC2 access to resources tagged Env=development, allows view-only access to all EC2 resources and prevents users from modifying resources tags.

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 that the Effect defines if permission is allowed or denied; the Action specifies the operations permitted; and the Resources identifies the AWS resources affected.

The screenshot shows the AWS IAM console interface for creating a policy. The breadcrumb navigation at the top reads 'IAM > Policies > Create policy'. On the left, a sidebar indicates 'Step 1: Specify permissions' and 'Step 2: Review and create'. The main heading is 'Specify permissions' with an 'Info' link. Below the heading is a sub-header: 'Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.'

The 'Policy editor' section has two tabs: 'Visual' and 'JSON', with 'JSON' being the active tab. The JSON editor contains the following policy document:

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "ec2:*",
7       "Resource": "*",
8       "Condition": {
9         "StringEquals": {
10          "ec2:ResourceTag/Env": "development"
11        }
12      },
13    },
14    {
15      "Effect": "Allow",
16      "Action": "ec2:Describe*",
17      "Resource": "*"
18    },
19    {
20      "Effect": "Deny",
21      "Action": [
22        "ec2:DeleteTags",
23        "ec2:CreateTags"
24      ],
25      "Resource": "*"
26    }
27  ]
28 }
```

On the right side of the editor, there is a section titled 'Edit statement' with a sub-header 'Select a statement'. It includes the text 'Select an existing statement in the policy or add a new statement.' and a button labeled '+ Add new statement'.

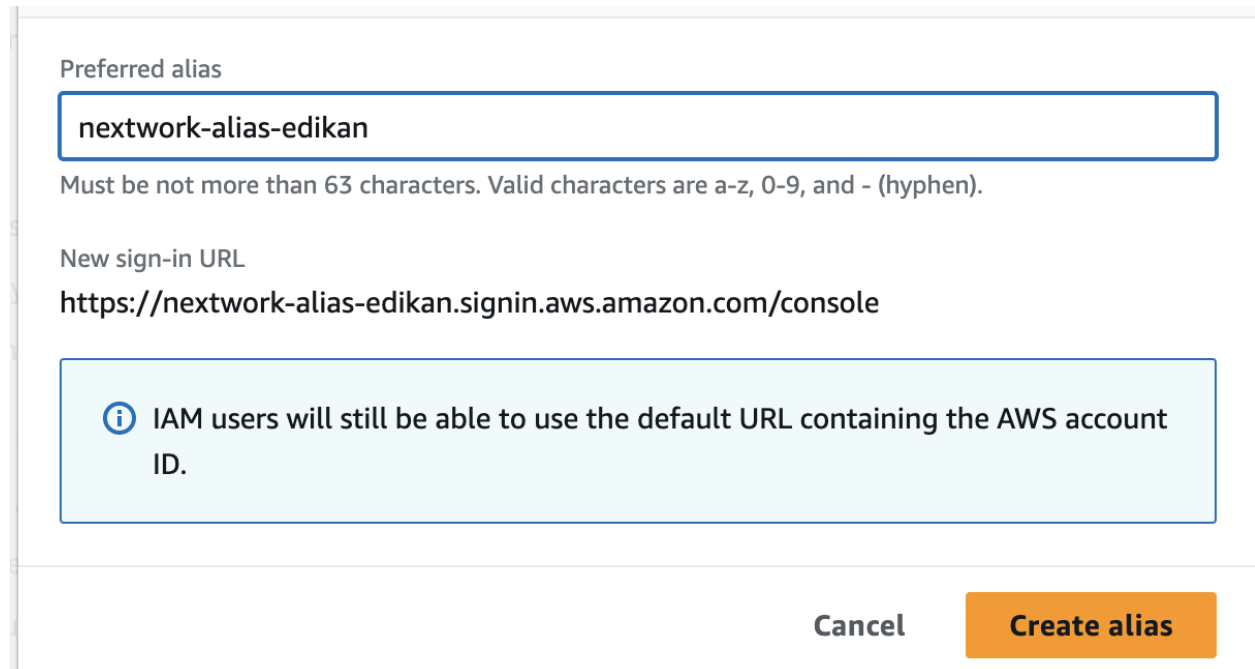
MY JSON POLICY

ACCOUNT ALIAS

An AWS Alias is a personalised name you use to replace the default account ID in your login URL. It takes less than a minute to create an account alias.

Now, my new AWS console sign-in URL is;

<https://nextwork-alias-edikan.signin.aws.amazon.com/console>



The screenshot shows the 'Create alias' dialog box in the AWS IAM console. It has a light gray border and a white background. At the top, the text 'Preferred alias' is in a small, gray font. Below it is a text input field with a blue border containing the text 'nextwork-alias-edikan'. Under the input field, a gray note states: 'Must be not more than 63 characters. Valid characters are a-z, 0-9, and - (hyphen)'. Below this, the text 'New sign-in URL' is in a small, gray font, followed by the URL 'https://nextwork-alias-edikan.signin.aws.amazon.com/console'. At the bottom of the dialog, there is a light blue information box with a blue 'i' icon and the text: 'IAM users will still be able to use the default URL containing the AWS account ID.'. At the very bottom of the dialog, there are two buttons: a gray 'Cancel' button and an orange 'Create alias' button.


Preferred alias

nextwork-alias-edikan

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

New sign-in URL

<https://nextwork-alias-edikan.signin.aws.amazon.com/console>

 IAM users will still be able to use the default URL containing the AWS account ID.

Cancel Create alias

IAM USERS AND USER GROUPS

Users

IAM users are like a login account for a person or application. Each has a name, password, and specific permissions, allowing them to access only certain AWS services perform actions you control, like read and write.


User Groups

An IAM user groups is a collection or grouping of IAM users. It enables you to manage permissions for all users within the group simultaneously by attaching policies to the group, instead of assigning them to each user individually.


I attached the policy I created to this user group, which means all users in the user group are granted the same permissions defined in the policy, simplifying access management.

Once I logged in as my IAM user, I noticed there were restrictions and i was denied access to IAM


Console sign-in details

Email sign-in instructions 


Console sign-in URL

 https://nextwork-alias-edikan.signin.aws.amazon.com/console

User name

 nextwork-dev-edikan

Console password

 ***** [Show](#)

Cancel

Download .csv file

Return to users list

TESTING IAM POLICIES

I tested my JSON IAM policy by stopping the production and development instances.

Stopping the production instance

When I tried to stop the production instance, I got an error message because of a lack of authorisation to access the instance.



Stopping the development instance

Next, when I tried to stop the development instance, it successfully stopped on first attempt which wasn't the case for the production instance.

Successfully initiated stopping of i-09518ada981292eab

Instances (1/2) Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input checked="" type="checkbox"/>	nextwork-dev...	i-09518ada981292eab	Stopped	t3.micro	-	User: arn:aws:iam::288761726154:user/nextwork-dev-edikan is not authorized to perform: ec2:StopInstances on resource: arn:aws:ec2:eu-north-1:288761726154:instance/i-0ba51f194644497a7 because no identity-based policy allows the ec2:StopInstances action.	eu-north-1b	-
<input type="checkbox"/>	nextwork-pro...	i-0ba51f194644497a7	Running	t3.micro	3/3 checks passed	User: arn:aws:iam::288761726154:user/nextwork-dev-edikan is not authorized to perform: ec2:StopInstances on resource: arn:aws:ec2:eu-north-1:288761726154:instance/i-0ba51f194644497a7 because no identity-based policy allows the ec2:StopInstances action.	eu-north-1b	ec2-13-6

AWS IAM is crucial for ensuring cloud security. It efficiently manages user access, permissions, and identity control, safeguarding your cloud environment from unauthorised access and promoting secure resource usage. Whether you're just starting out or are an experienced cloud professional, mastering IAM is essential for creating a secure cloud infrastructure.