



Cloud Security with AWS IAM

L

Louis Moyo

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions.

Policy editor

```
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}
```



Introducing Today's Project!

In this project, I will demonstrate identity and access control on AWS by launching an EC2 instance and using IAM policies, users, and groups to enforce least-privilege access. I'm doing this to learn how authentication and authorisation work together in practice.

Tools and concepts

Services I used were Amazon EC2 and AWS IAM. Key concepts I learnt include launching and tagging EC2 instances, creating IAM policies with conditions, using user groups and users for access control, and securing logins with an account alias.

Project reflection

This project took me approximately 1 hour. The most challenging part was understanding how IAM policies use conditions with tags. It was most rewarding to see the intern user successfully access only the development instance while being blocked from production.



Tags

Tags are key-value pairs that I can attach to AWS resources, such as EC2 instances. They are useful for organising and identifying resources, tracking costs, applying automation rules, and controlling access through IAM policies. Tags make it easier to manage resources at scale by grouping them by project, environment, or owner.

The tags I've used on my EC2 instances are called Name and Env. For the Name tag, the value I assigned is nextwork-dev-louis. For the Env tag, the value I assigned is development. These tags help me easily identify the instance by project and environment.

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Key <small>Info</small> <input type="text" value="Name"/> <small>X</small>	Value <small>Info</small> <input type="text" value="nextwork-dev-louis"/> <small>X</small>	Resource types <small>Info</small> <input type="button" value="Select resource types"/> <small>▼</small> <input type="button" value="Remove"/> <input type="button" value="Instances"/> <small>X</small>
Key <small>Info</small> <input type="text" value="Env"/> <small>X</small>	Value <small>Info</small> <input type="text" value="development"/> <small>X</small>	Resource types <small>Info</small> <input type="button" value="Select resource types"/> <small>▼</small> <input type="button" value="Remove"/> <input type="button" value="Instances"/> <small>X</small>

Add new tag

You can add up to 48 more tags.



IAM Policies

IAM Policies are rules that define what actions users or groups can perform on AWS resources.

The policy I set up

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

I've created a policy that lets the intern perform all EC2 actions on instances tagged with Env=development, while still allowing them to view (describe) all EC2 resources, and explicitly denying them the ability to create or delete tags.

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

Effect tells AWS whether the statement allows or denies the action. Action specifies which AWS operations are being controlled (e.g., ec2:StartInstances). Resource defines which resources the action applies to (e.g., a specific instance or * for all).



My JSON Policy

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions.

Policy editor

```
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 }
```



Account Alias

An account alias is a custom name that replaces the 12-digit AWS account ID in the login URL, making it easier for users to sign in.

Creating an account alias took me less than a minute. Now, my new AWS console sign-in URL is [https://<my-alias>.signin.aws.amazon.com/console](https://nextwork-alias-louis.signin.aws.amazon.com/console).

The screenshot shows the IAM Dashboard with a green success message at the top: "Alias for this account updated to nextwork-alias-louis." The dashboard includes sections for IAM resources (User groups: 0, Users: 2, Roles: 13, Policies: 7, Identity providers: 0) and AWS Account details (Account ID: 756716632322, Account Alias: nextwork-alias-louis, Sign-in URL: https://nextwork-alias-louis.signin.aws.amazon.com/console). A "What's new" section is also visible.



IAM Users and User Groups

Users

IAM users are individual identities in AWS with their own credentials. They represent people or applications that need access to AWS resources.

User Groups

IAM user groups are collections of IAM users that share the same permissions. By attaching policies to a group, all users in that group automatically inherit those permissions.

I attached the policy I created to this user group, which means every intern in the group can access and manage only the development EC2 instance while being restricted from production resources.



Logging in as an IAM User

The first way is to download the user's sign-in credentials as a .csv file and share it securely. The second way is to copy the sign-in link, username, and password from the console and provide them securely to the new user.

Once I logged in as my IAM user, I noticed that several dashboard panels showed "Access denied." This was because the user only has limited permissions from the policy I created, which restricts access to everything except the development EC2 instance.

Retrieve password
You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details	
Console sign-in URL	https://nextwork-alias-louis.signin.aws.amazon.com/console
User name	nextwork-dev-louis
Console password	***** Show

[Email sign-in instructions](#)

[Cancel](#) [Download .csv file](#) [Return to users list](#)

L

Louis Moyo

NextWork Student

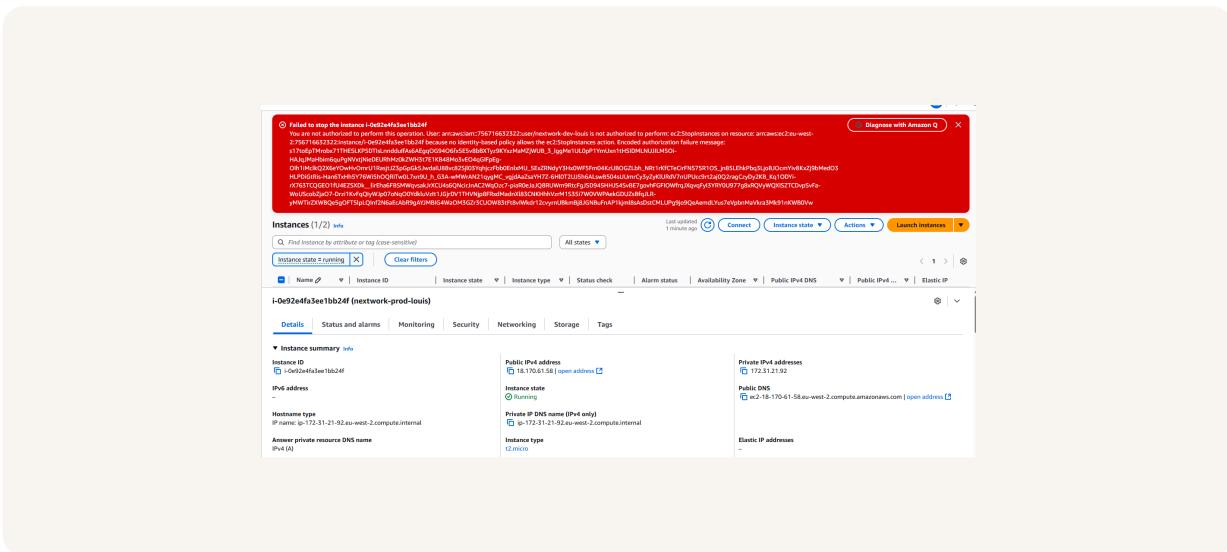
nextwork.org

Testing IAM Policies

I tested my JSON IAM policy by attempting to stop both the production and development EC2 instances.

Stopping the production instance

When I tried to stop the production instance, the action was denied. This was because the IAM policy I created only allows control of instances tagged with Env=development, so the user does not have permission to stop the production instance.





Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance, the action succeeded. This was because my IAM policy explicitly allows EC2 actions on resources tagged with Env=development.

The screenshot shows a CloudWatch Metrics Insights search results page. At the top, a message says "Successfully initiated stopping of i-0f5aae782fd6170b8". Below it, a table titled "Instances (1/2) Info" displays two rows of instance data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
nextwork-dev-louis	i-0f5aae782fd6170b8	Stopping	t2.micro	2/2 checks passed	User: arn:aws:	eu-west-2a	ec2-13-40-170-78.eu-w...	13.40.170.78	-
nextwork-prod-louis	i-0e92e4fa3ee1bb24f	Running	t2.micro	2/2 checks passed	User: arn:aws:	eu-west-2a	ec2-18-170-61-58.eu-w...	18.170.61.58	-



The IAM Policy Simulator

The IAM Policy Simulator is a tool that lets you test and validate IAM policies without affecting real AWS resources. It's useful for checking whether a user will be allowed or denied access to specific actions before trying them in production, saving time and avoiding disruption.

How I used the simulator

I set up a simulation for the StopInstances action on the development instance. The results were denied by default because the instance had no tags, so I manually added the Env=development tag. After that, the simulator confirmed access was allowed.

The screenshot shows the Policy Simulator interface. At the top, there are dropdown menus for 'Service' (set to 'Amazon EC2') and 'Action(s) selected' (set to '2 Action(s) sele...'), and buttons for 'Select All', 'Deselect All', 'Reset Contexts', 'Clear Results', and 'Run Simulation'. The 'Run Simulation' button is highlighted in blue.

Below the header, there is a 'Global Settings' section with a link to 'Show statement in NextWorkDevEnvironmentPolicy (IAM Policy)'.

The main area displays 'Action Settings and Results' for two actions:

Service	Action	Resource Type	Simulation Resource	Permission
Amazon EC2	DeleteTags	not required	*	denied 1 matching statements.
Show statement in NextWorkDevEnvironmentPolicy (IAM Policy)				
Resource You can specify the resource and context keys used to simulate this action. By default the simulation resource is ***.				
Add Resource				
Amazon EC2	StopInstances	instance	*	allowed 1 matching statements.
Show statement in NextWorkDevEnvironmentPolicy (IAM Policy)				
Resource You can specify the resource and context keys used to simulate this action. By default the simulation resource is ***.				
instance	*	ec2:resourcetag/env	development	



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

