



Cloud Security with AWS IAM



Abhishek sanap

The screenshot shows the AWS IAM Policy editor interface. On the left, there is a large text area containing JSON code for a policy. On the right, a modal window titled "Edit statement" is open, prompting the user to "Select a statement" or "Add new statement". The JSON code in the main area is as follows:

```
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 }
```

At the bottom of the JSON code area, there is a button labeled "+ Add new statement". The status bar at the bottom indicates "5851 of 6144 characters remaining". Below the status bar, there are links for "Security: 0", "Errors: 0", "Warnings: 0", and "Suggestions: 0".



Introducing today's project!

What is AWS IAM?

It is a web service for securely controlling access to AWS resources.

How I'm using AWS IAM in this project

AWS IAM enables you to securely control access to AWS services and resources for your users. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

One thing I didn't expect...

Json code i didnt expect because my process was different with natasha

This project took me...

the project takes necessary time between 1:30 to 1:49 hours

Abhishek sanap
NextWork Student

NextWork.org

Tags

Tags are labels to help AWS Account users identify and manage their resources. Tags are useful 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 represents the two different environments that we are using to build and release the NextWork app.

▼ Name and tags [Info](#)

Key Info <input type="text" value="Name"/> X	Value Info <input type="text" value="nextwork-devel"/> X	Resource types Info <input type="text" value="Select resource ty..."/> ▼ Remove
Instances X		
Key Info <input type="text" value="Env"/> X	Value Info <input type="text" value="development"/> X	Resource types Info <input type="text" value="Select resource ty..."/> ▼ Remove
Instances X		
Add new tag		
You can add up to 48 more tags.		



IAM Policies

IAM Policies are rules that help to allow/deny users'/resources' permissions to perform certain actions to my AWS Account's resources.

The policy I set up

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

I've created a Policy that allows all EC2- related actions to all EC2 instances that have the Environment ("Env") tag "development". But, it also denies creating and deleting tags for ALL EC2 instances.

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

When writing JSON Policy statements, you have to specify the: Effect: ie. Allow or Deny. Action: ie the specific action that we are wanting to allow or deny. Resource: the specific resource/group of resources in my AWS Account that this policy will



Abhishek sanap
NextWork Student

NextWork.org

My JSON Policy

Policy editor

Visual **JSON** Actions ▾

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 }
+ Add new statement

5851 of 6144 characters remaining

JSON Ln 3, Col 24

⌚ Security: 0 ⚡ Errors: 0 ⚠ Warnings: 0 🌐 Suggestions: 0

Edit statement

Select a statement

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

+ Add new statement

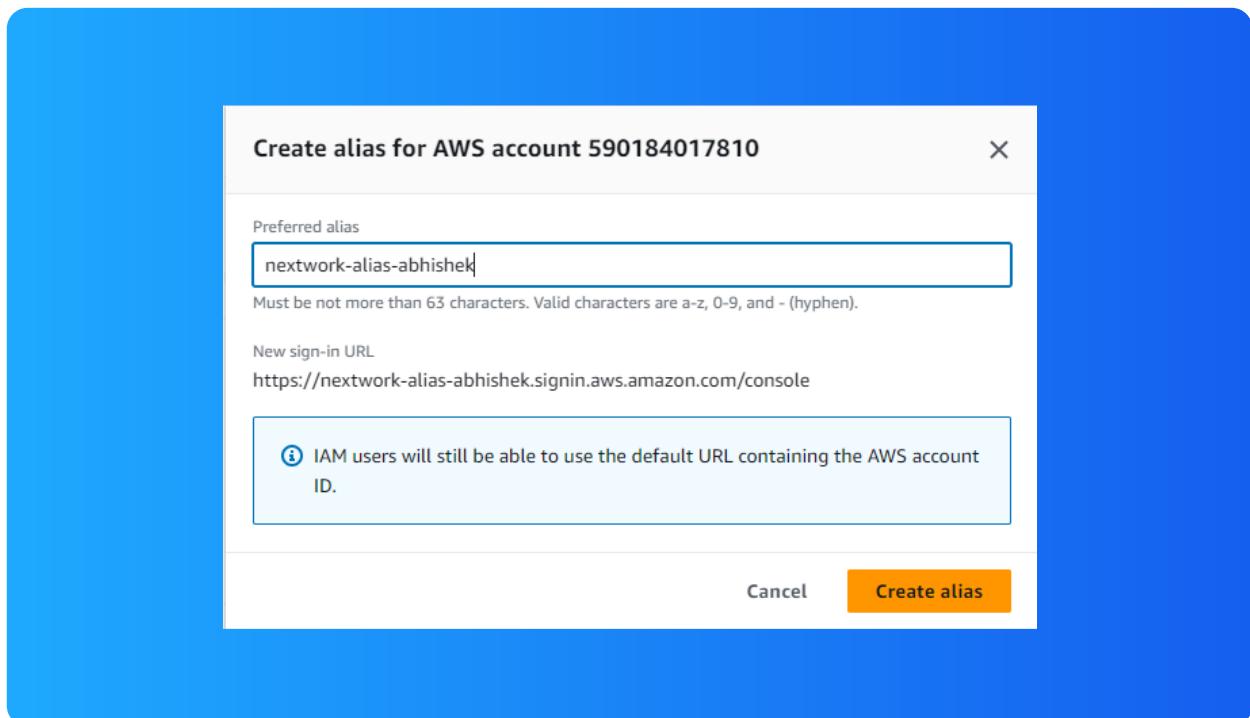


Account Alias

An account alias is a custom name that I can assign to my AWS Account. This custom name would replace my Account ID in my Account's log-in URL

Creating an account alias took me less than a minute-super fast!

Now, my new AWS console sign-in URL is <https://nextwork-alias-abhishek.signin.aws.amazon.com/console>





IAM Users and User Groups

Users

IAM Users are other log-ins/people who have access to my AWS Account. These users are created by myself using the AWS IAM service! I can designate my IAM users' access to my AWS Account's resources/services

User Groups

I also created a User Group. User Groups are useful for grouping and managing users' permissions at a group level. They act similarly to folders when it comes to mass assigning permission/policies

My User Group is called nextwork-dev-group. I attached the Policy I created to this User Group, which means all users that are added to that user group will inherit the user group's access permissions



Logging in as an IAM User

Once my new user was set up, there were two ways I could share its sign-in details: firstly, emailing sign-in instructions; secondly, downloading a .csv file. My new user had a unique sign-in URL - this is my Accoun work!

Once I logged in, I noticed that a lot of panels displayed "Access denied". This was a clear difference to the dashboard I usually see in my AWS Account (where I had unrestricted access to resources and wasn't denied access to anything)

Console sign-in details

Email sign-in instructions [\[link\]](#)

Console sign-in URL
<https://nextwork-alias-abhishek.signin.aws.amazon.com/console>

User name
[\[link\]](#) nextwork-dev-abhishek

Console password
[\[link\]](#) ***** [Show](#)

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

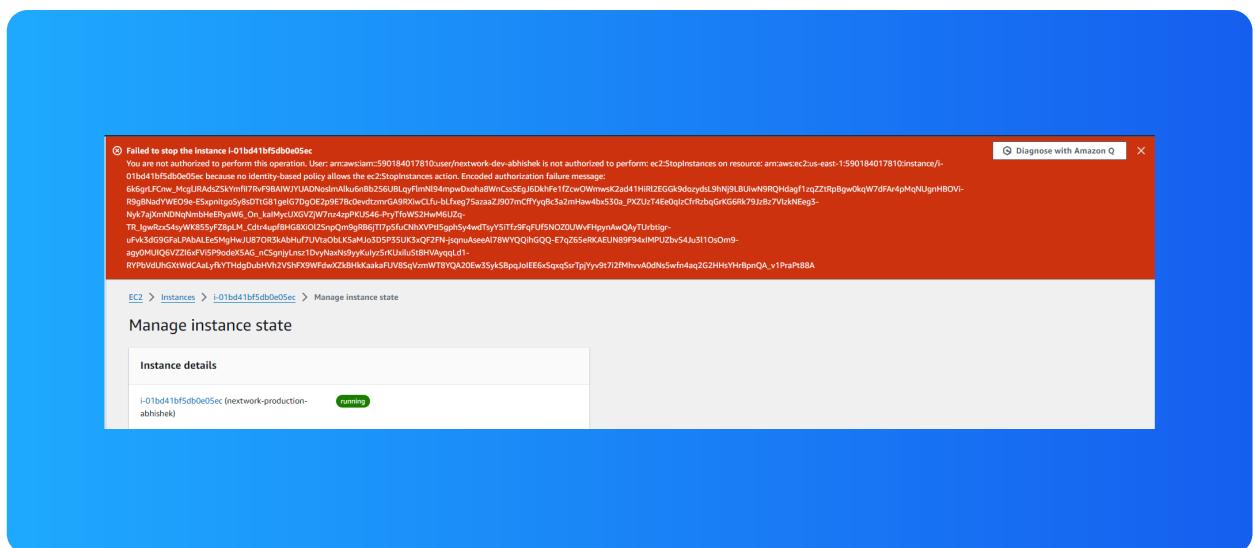


Testing IAM Policies

I tested the JSON IAM policy I set up by trying to Stop the development and production instances ie. triggering the StopInstances action.

Stopping the production instance

When I tried to stop the production instance, an error message stopped me and explained that I am not authorised to stop the production instance.

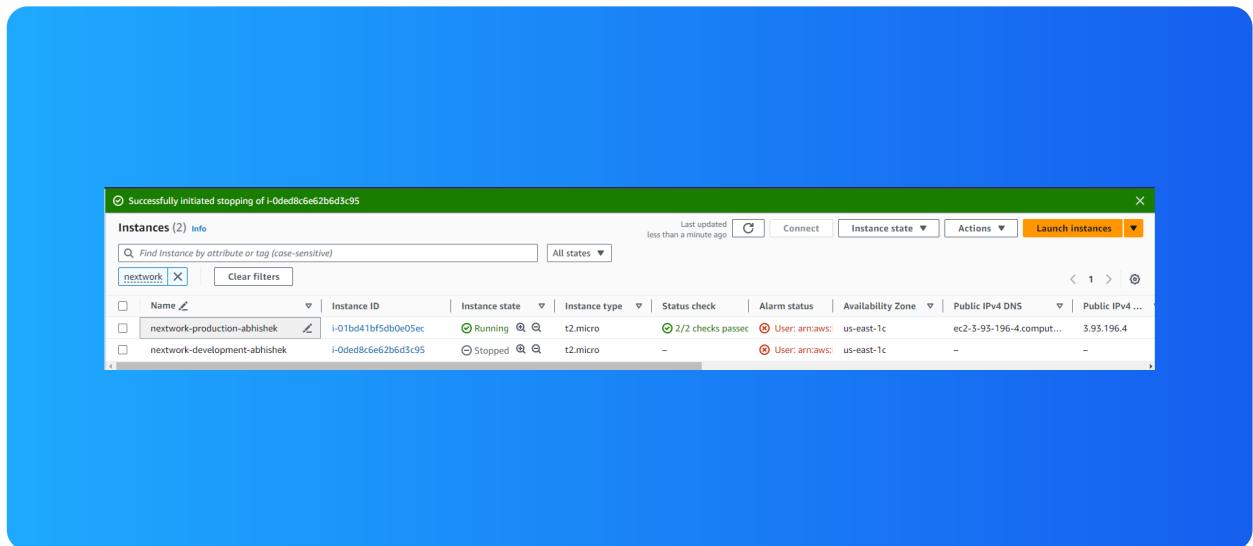




Testing IAM Policies

Stopping the development instance

when I tried to stop the development instance, the development instance could be stopped! This was because the Policy I created (and attached to the User Group that my User is a part of) allowed all EC2 related actions to all EC2 instances/resources.





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

