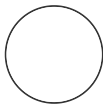


Cloud Security with AWS IAM



Mahmoud Alshaer

```
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 how to use AWS IAM to control access and permission settings in my AWS account. I'm doing this project to learn about cloud security.

Tools and concepts

Services I used were Amazon EC2 and AWS IAM. Key concepts I learnt include IAM users, policies, user groups and account aliases. I also learnt how to launch an EC2 instance and how to tag the instance.

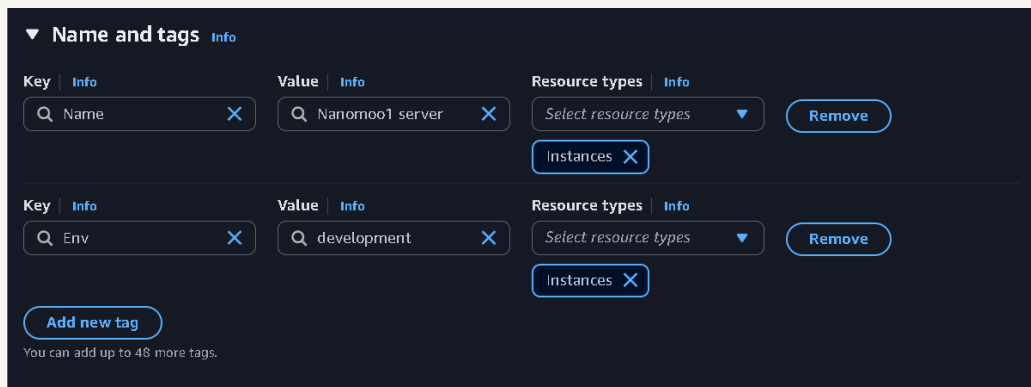
Project reflection

This project took me approximately 1 hour and 30 mins. The most challenging part was understanding the IAM policy since it was written in JSON. It was most rewarding to see permission denied when the intern tried to delete the production instance.

Tags

Tags are organizational tools that let us label our resources. They are helpful for grouping resources, cost allocation, and applying policies for all resources with the same tag.

The tag I've used on my EC2 instances is called Env, which stands for environment. The value I've assigned for my instances are production and development.



IAM Policies

IAM Policies are rules that determine who can do what in our AWS account. I'm using policies today to control who has access to the production/development instance.

The policy I set up

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

I've created a policy that allows the policy holder (i.e. the intern) to have permission to do anything they want to any instance tagged with "development".

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

The Effect, Action, and Resource attributes of a JSON policy means whether or not the policy is allowing/denying action (the effect); what the policy holder can or can't do (action); the specific AWS resources that the policy relates to (resource).

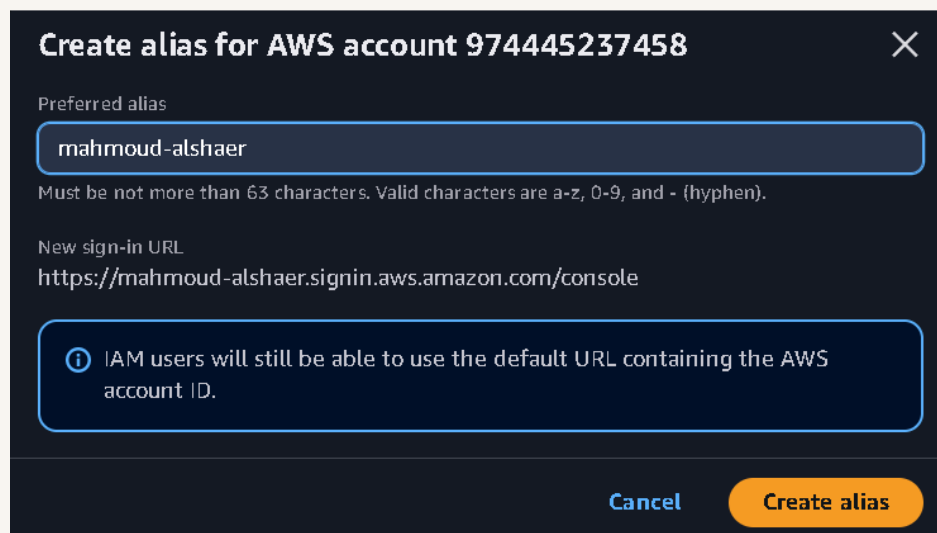
My JSON Policy

```
Policy editor Vis
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 nickname for an AWS account.

Creating an account alias took me 30 seconds. Now, my new AWS console sign-in URL uses the alias instead of my account ID.



The screenshot shows a dark-themed dialog box titled "Create alias for AWS account 974445237458" with a close button (X) in the top right corner. Inside the dialog, there is a section for "Preferred alias" with a text input field containing "mahmoud-alshaer". Below the input field, a note states: "Must be not more than 63 characters. Valid characters are a-z, 0-9, and - (hyphen)." Below this, the "New sign-in URL" is displayed as "https://mahmoud-alshaer.signin.aws.amazon.com/console". A blue information icon (i) is followed by the text: "IAM users will still be able to use the default URL containing the AWS account ID." At the bottom right, there are two buttons: a blue "Cancel" button and an orange "Create alias" button.

IAM Users and User Groups

Users

IAM users are people or entities that have access/can login to your AWS account.

User Groups

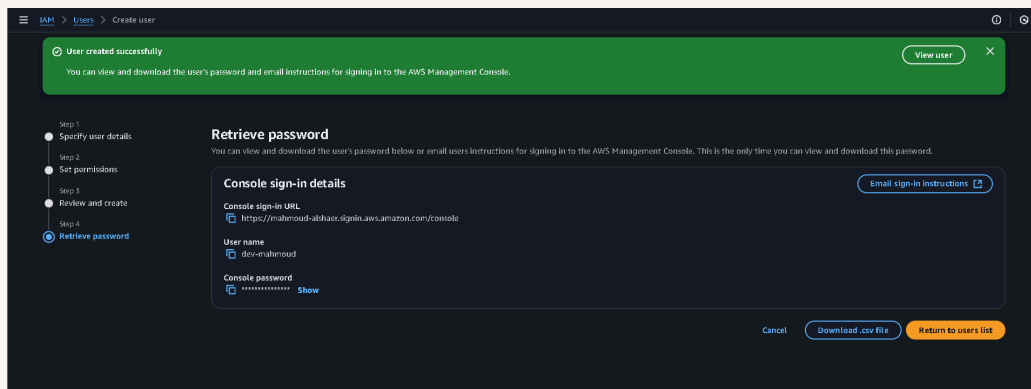
IAM user groups are folders that collect IAM users so that you can apply permission settings at the group level.

I attached the policy I created to this user group, which means any user created inside this group will automatically get the permissions attached to the custom policy.

Logging in as an IAM User

The first way is to email sign in instructions to the user, while the second way is to download the CSV file with the sign in instructions inside.

Once I logged in as my IAM user, I noticed that the user is already denied access to panels on the AWS console dashboard. This was because I only setup permissions to the EC2 development instance, so the intern wouldn't have access to anything else.



Testing IAM Policies

I tested my JSON IAM policy by attempting to stop both instances.

Stopping the production instance

When I tried to stop the production instance I was met with an error. This was because the production instance is tagged with the production label which is outside of the scope of the permission policy.



Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance I successfully saw the instance state change from stopping to stopped. This was because the permission policy allows the intern to stop instances.

