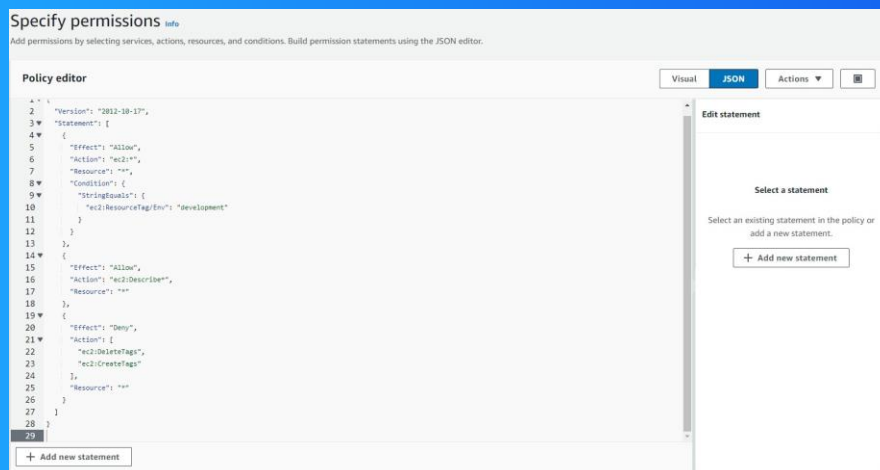




Cloud Security with AWS IAM



James Francis Facistol





Introducing today's project!

What is AWS IAM?

AWS IAM is a service that enables you to manage users, permissions, and roles securely in AWS. It's useful for controlling access to resources, ensuring security, and following best practices for user management in cloud environments.

How I'm using AWS IAM in this project

In today's project, I used AWS IAM to create user roles with specific permissions, ensuring secure access to resources. I configured policies to control access levels, enabling users to work effectively while maintaining security best practices.

One thing I didn't expect...

I didn't expect this project to be so easy. My background in Linux administration has really helped me grasp the concepts behind cloud services.

This project took me...

This project took me about an hour and a half as I focused on fully understanding the material.



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.

I've tagged my EC2 instances with "Env," assigning the values "production" and "development" to represent the two environments for building and releasing our new app.

▼ Name and tags [Info](#)

Key [Info](#)

Q Name X

Value [Info](#)

Q nextwork-develc X

Resource types [Info](#)

Select resource types ▼

Instances X

Remove

Key [Info](#)

Q Env X

Value [Info](#)

Q development X

Resource types [Info](#)

Select resource types ▼

Instances X

Remove

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 for instances tagged with "Env" set to "development." Additionally, it denies the ability to create or delete tags for all EC2 instances.

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

JSON policy attributes are: Effect: Specifies if the action is allowed or denied. Action: Defines the specific action. Resource: Identifies the resource or group affected.



My JSON Policy

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor Visual **JSON** Actions Grid

```
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](#)

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 assigned to your AWS account that replaces the account ID in the login URL.

Creating an account alias took me less than a minute - it is really easy and super-fast.

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

Create alias for AWS account 767398119712 ✕

Preferred alias

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

New sign-in URL

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

i 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 individual accounts within AWS Identity and Access Management (IAM) that enable users to securely access and manage AWS resources. Each IAM user has specific permissions and credentials.

User Groups

IAM user groups are collections of IAM users that allow you to manage permissions collectively. By assigning permissions to a group, you can easily control access for multiple users at once.

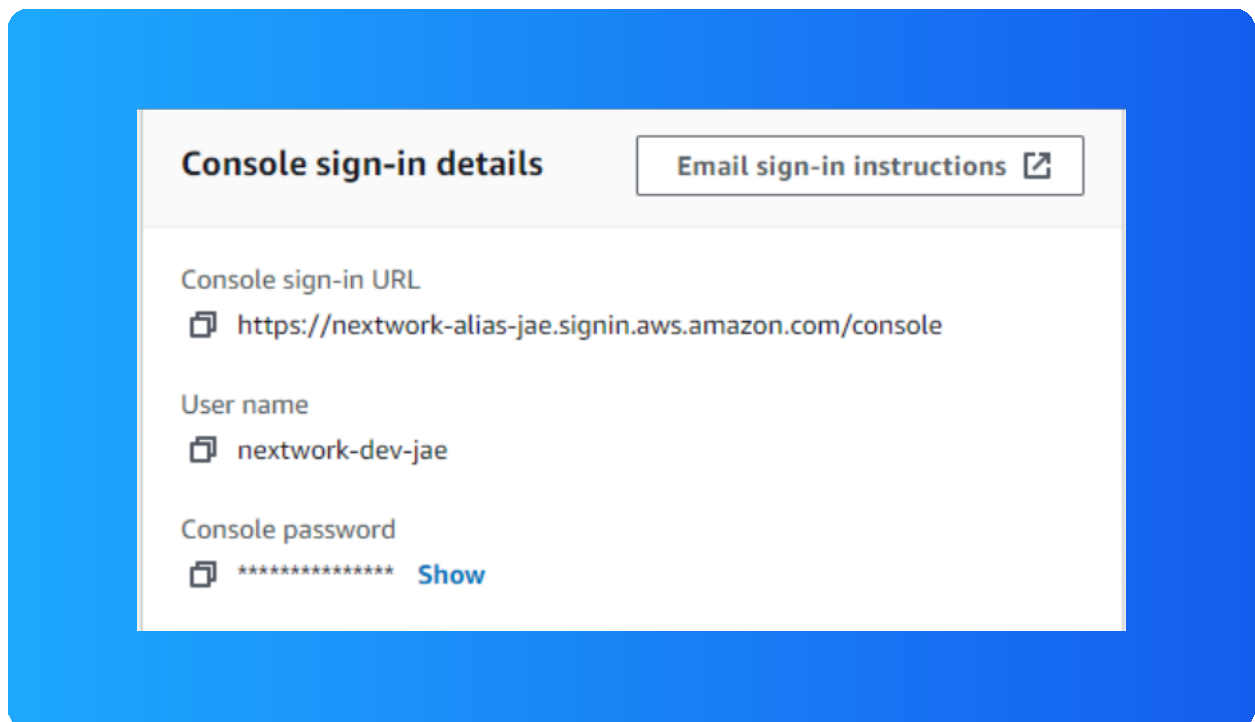
I attached the policy to this user group, so all members inherit its permissions. This simplifies management, as any changes to the policy automatically affect everyone in the group.



Logging in as an IAM User

The first way is to send the user an email containing their sign-in details securely. The second way is to provide the details via a secure messaging platform or a password manager that both you and the user have access to.

Once I logged in as my IAM user, I noticed that the dashboard was tailored to my permissions, providing access only to the resources relevant to my role. This made navigation easier and enhanced security by limiting exposure to unnecessary features.





Testing IAM Policies

I tested the JSON IAM policy I created by attempting to stop the development and production instances, specifically by triggering the StopInstances action.

Stopping the production instance

When I tried to stop the production instance, I got a permissions error, indicating my IAM policy didn't grant the necessary access, so I wasn't authorized to stop it.

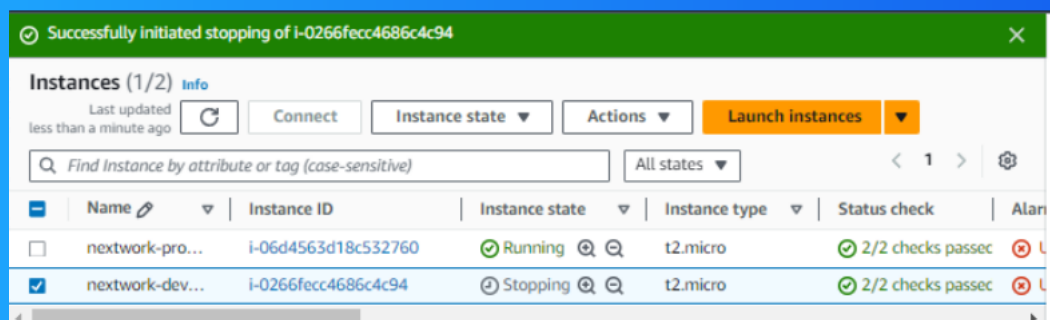
```
Failed to stop the instance i-06d4563d18c532760
You are not authorized to perform this operation. User: arn:aws:iam::767398119712:user/nextwork-dev-jae is not
authorized to perform: ec2:StopInstances on resource: arn:aws:ec2:ap-southeast-1:767398119712:instance/i-
06d4563d18c532760 because no identity-based policy allows the ec2:StopInstances action. Encoded authorization failure
message: nG7pGOuvJ_Kp5wzjs0Zt2HtHQPRNjJ0ejRKbH5SeJROTT5xiyqfYGIQL-ZKsS-
5Dm2pf2vrrPnOskCsD.NvFJKayymTUHOgmaz3QNbSKys1-T8Gz4WvSLVpcSQbzhd9YH8NoaIYSWroH4z55XIYgyV-
2HDW8yH29Go65dRpE17v6QSGHkvT6cmqAeK9TJbbNCYTnrcCwADsJXbT9NDsrpqModub_z7oALXaxL5f7TdQs73atwQITq0j
6sqXS-
xTZP8jpAGrcAy1CeFI26o_GHAPMUOIWByGLcwWp5wmvyLVpCry2h5UgR7x7PkVqssfoJGxHW9Pi1eku7oMEQkiiNjqeCW05qN
HXpCko9JB9R8r-QrWrzQ11c2xE73DdPeoFAKZ8Tt7QS1UYf4xLv8kTLZa0x75HJ_KJDpdcs5dGf_yw9n4SgrcRHwbnrfVn6bnkT-
0g0tv97r8p-DFWky1ZU6OeQv3FUba8Rz-205mBhqdGM-
ZOKyK6FYV2UihRHZeHynhrfy7d4QLgExurjoVMYIM0yJ137Is4nNIDpw46zrxuGdeow7FQlnia_W7Y5QejGKUUBfomXsl6cPN
VDLkTgOqoPE2iraLvoh-Jq4ilbPrPpNdEXI3-
dGsESkdBo4au9fcrzufcx4osB7G9aqtVcM5TtaUYcXYfA51pLA3YUdwt2Dsz2dhrOnoG-
mn4YkDqXwh7pgxYBhsVRnm2Pfi1bw04y-22JhvPTxW3U4efTzph8dOxqa92FP-
SRloW72FPueVCL7nhpmb8WjxIHp8BaxueTHDUfjuAcCoam5xlp09v3cSdR_qM5qrfThh_jrvT7Q9bc9dDq0t_3dOoCTzzZZMIi
KFx-sO1G4dlryem8YKdTy2tVuvRhAIW3OYgFAU2fnsfMm
```



Testing IAM Policies

Stopping the development instance

I successfully stopped the development instance because my policy allowed all EC2 actions on instances tagged with "Env: development," which I attached to my user group.





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

