



Cloud Security with AWS IAM



Kiprono Yegon

Screenshot of the AWS IAM 'Create policy' interface showing the 'Specify permissions' step.

The JSON code in the Policy editor is as follows:

```
1 { "Version": "2012-10-17",  
2   "Statement": [  
3     {  
4       "Effect": "Allow",  
5       "Action": "ec2:Describe*",  
6       "Resource": "*"  
7     },  
8     {  
9       "Effect": "Allow",  
10      "Action": "ec2:CreateTags",  
11      "Resource": "arn:aws:ec2:  
12        ${AWSRegion}:*:instance/*",  
13      "Condition": {  
14        "StringEquals": {  
15          "ec2:ResourceTag/Env": "development"  
16        }  
17      }  
18    },  
19    {  
20      "Effect": "Allow",  
21      "Action": "ec2:DeleteTags",  
22      "Resource": "arn:aws:ec2:  
23        ${AWSRegion}:*:instance/*",  
24      "Condition": {  
25        "StringNotEquals": {  
26          "ec2:ResourceTag/Env":  
27        }  
28      }  
29  ]  
30}
```

The right side of the screen shows the 'Edit statement' interface with a sidebar for selecting or adding new statements.

A circular profile picture of a person with dark hair and a beard, wearing a blue shirt.

Introducing Today's Project!

In this project, I will demonstrate using AWS IAM to control access to AWS services to specific users. I'm doing this project to learn more about cloud security and best practices if the cloud.

Tools and concepts

Services I used were Amazon EC2 and Amazon IAM. Key concepts I learnt include IAM users, policies, user groups and account aliases. I learned how JSON policies worked.

Project reflection

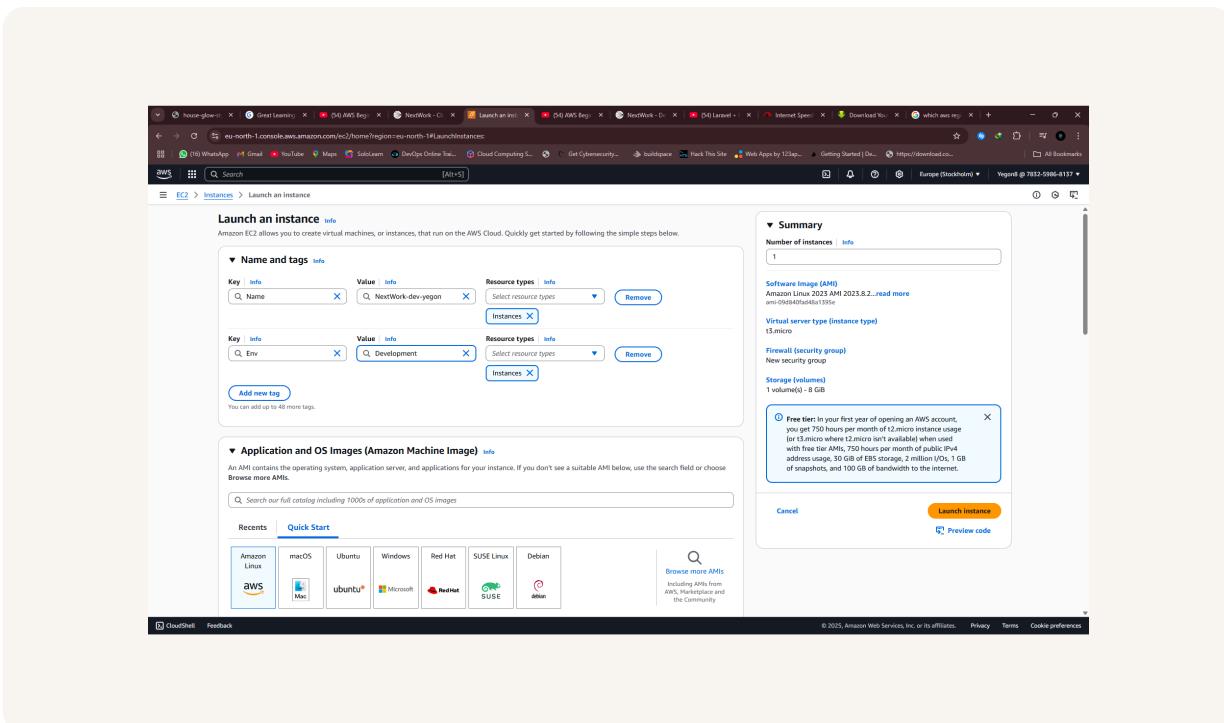
This project took me approximately one and a half hour to complete. The most challenging part was attaching policies. It was most rewarding to finally create instances and to delete them.



Tags

Tags are labels you attach to AWS resources for organization. I used the tag "Env" to label the production and development resources.

The tag I've used on my EC2 instances is called Environment (Env). The value I've assigned for my instances are development and production.





IAM Policies

IAM Policies are rules for who can do what with given AWS resources. It gives permissions to IAM users, groups and roles and specify what changes they can do to specific AWS resources.

The policy I set up

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

I've created a policy that allows all actions on the EC2 instances

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; that the Effect can deny or allow a certain action and the action defines the list of actions that the policy allows or denies and the resource shows which resource the policy apply.



My JSON Policy

Screenshot of the AWS IAM Policy editor showing a JSON policy document.

Step 1: Specify permissions

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

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/Evn": "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 }
```

Actions

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

Visual JSON Actions [Edit statement](#)

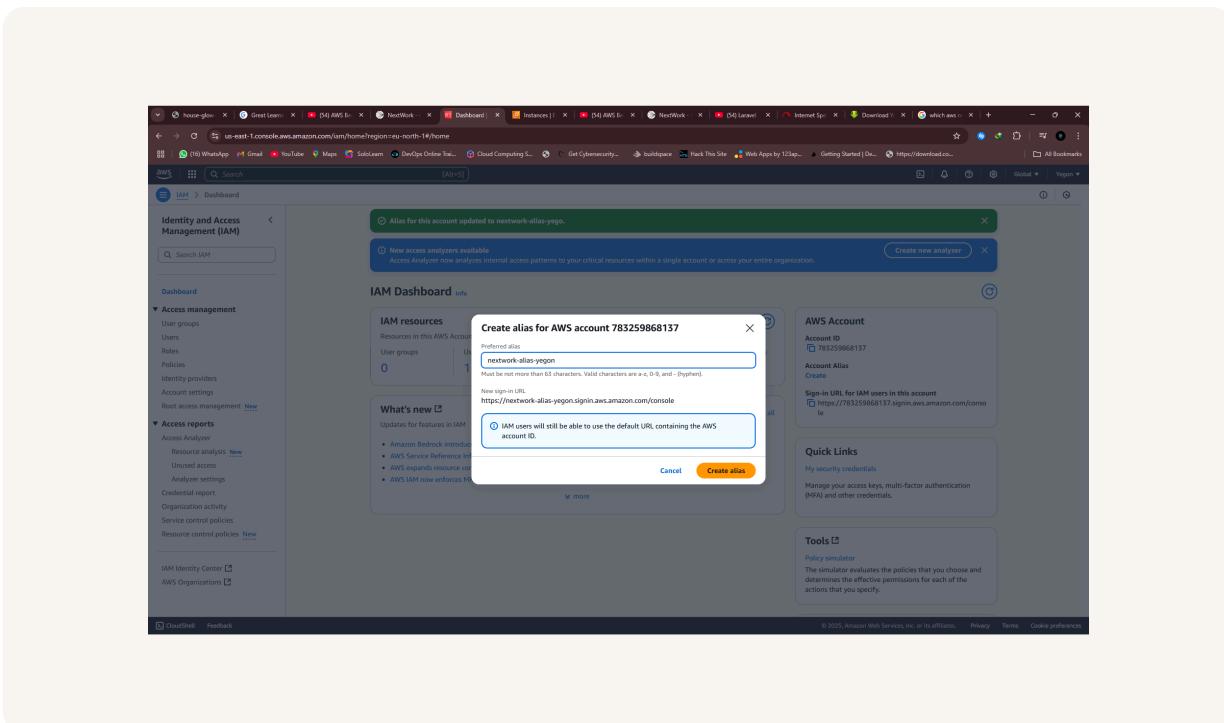
JSON Ln 29, Col 0 5851 of 6144 characters remaining



Account Alias

An account alias is a unique user friendly name that you use for an account instead of the account ID to sign in to the AWS Management Console.

Creating an account alias took me less than two minutes. Now, my new AWS console sign-in URL is <https://nextwork-alias-yegon.signin.aws.amazon.com/console>



A circular profile picture of a person with dark hair and a beard, wearing a blue shirt.

IAM Users and User Groups

Users

IAM users are the people or accounts that will get to use the AWS resources that are specified. Users are given permissions to access the resources mentioned in the policy.

User Groups

IAM user groups are collection of IAM users that have the same permissions to perform specific tasks to given resources.

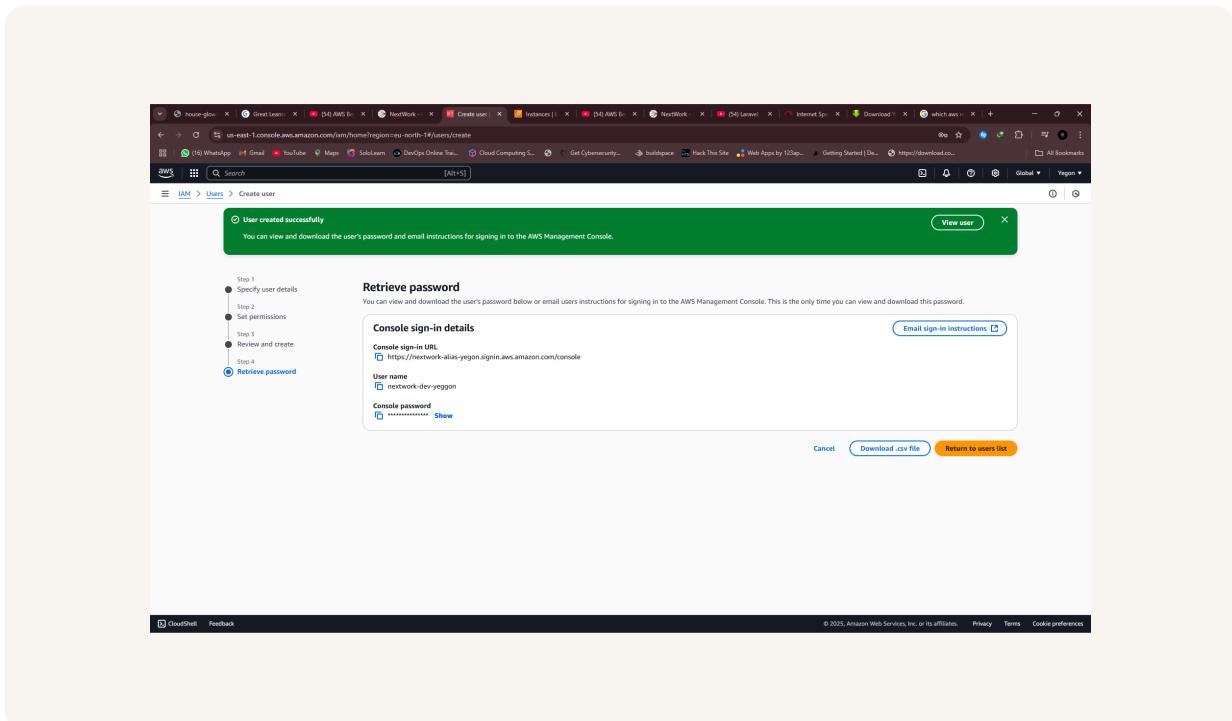
I attached the policy I created to this user group, which means that the users in that group are given permissions by the policy on the resources they will work with



Logging in as an IAM User

The first way to share a user's sign in details is by downloading the csv file and sharing it with the new user or by emailing the sign in instructions to the user.

Once I logged in as my IAM user, I noticed that some of the dashboard panels are showing access denied. This was because of the IAM policy that we gave to the user group that the user is in.





Testing IAM Policies

I tested my JSON IAM policy by trying to stop the two EC2 instances. The JSON IAM policy only allows the stopping of one EC2 instance policy; the development instance.

Stopping the production instance

When I tried to stop the production instance there was a red banner that showed an error. This was because the intern does not have access to the production instance as stated in the IAM policy.

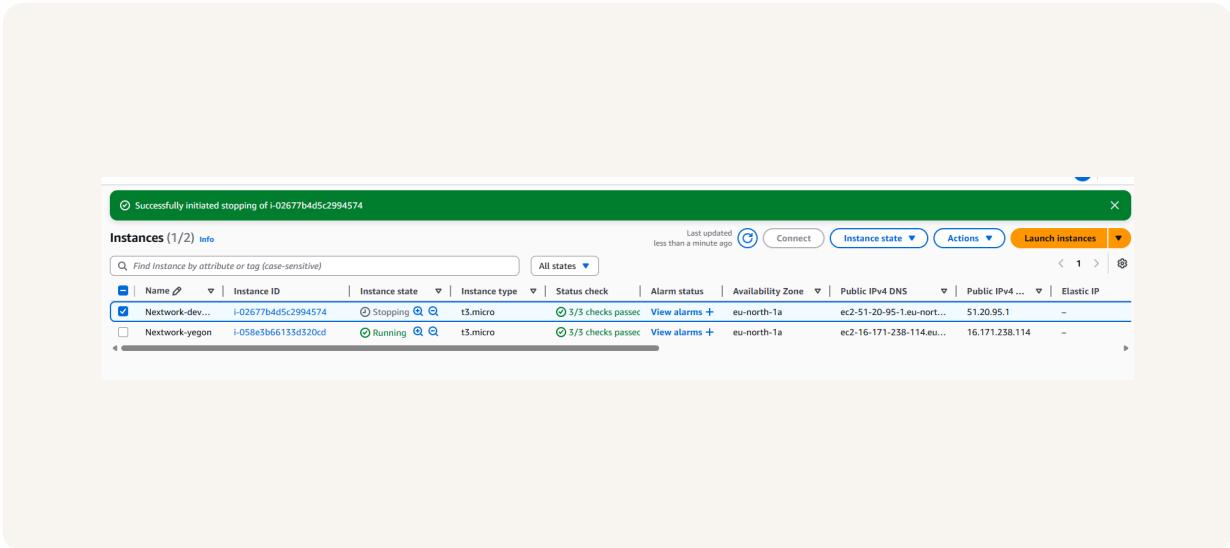




Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance it stopped. This was because the intern has access to manipulate the development instance according to the JSON IAM policy





nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

