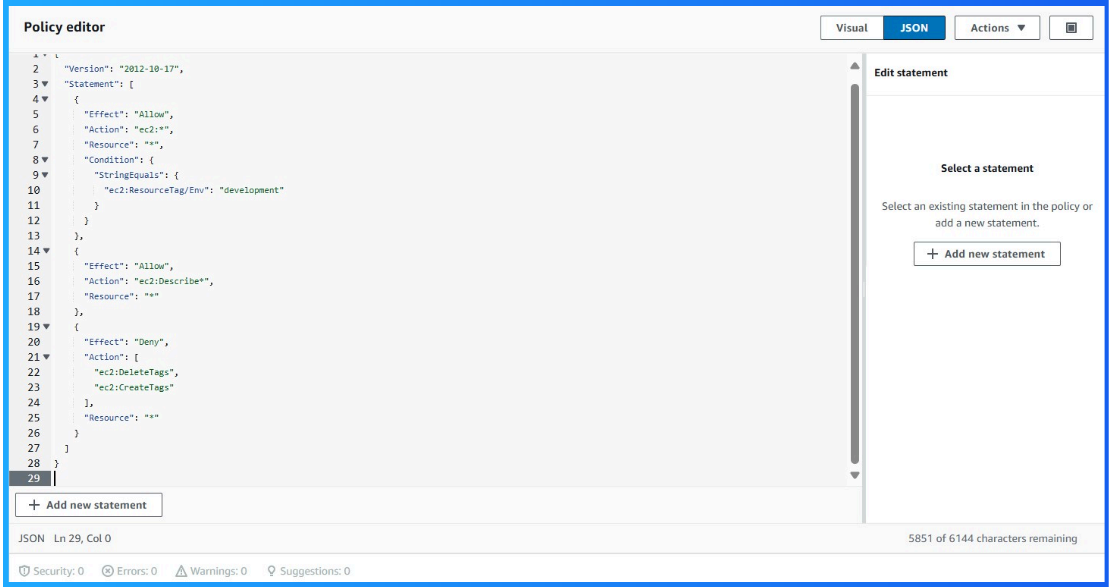
Cloud Security with AWS IAM

Gabriel Mazer



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Env": "development"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": [
        "ec2:DeleteTags",
        "ec2:CreateTags"
      ],
      "Resource": "*"
    }
  ]
}
```

JSON Ln 29, Col 0

5851 of 6144 characters remaining

Security: 0 Errors: 0 Warnings: 0 Suggestions: 0

Introducing today's project!

What is AWS IAM?

AWS IAM is a service that controls access to AWS resources by managing permissions and user authentication. It's essential for securing AWS environments and granting appropriate levels of access.

How I'm using AWS IAM in this project

I used AWS IAM to create users, groups, and policies, applying role-based access to control which actions could be performed on my EC2 instances based on tags and environment.

One thing I didn't expect...

One thing I didn’t expect was how IAM policies can be so precise, allowing me to test permission rules like stopping instances only in specific environments.

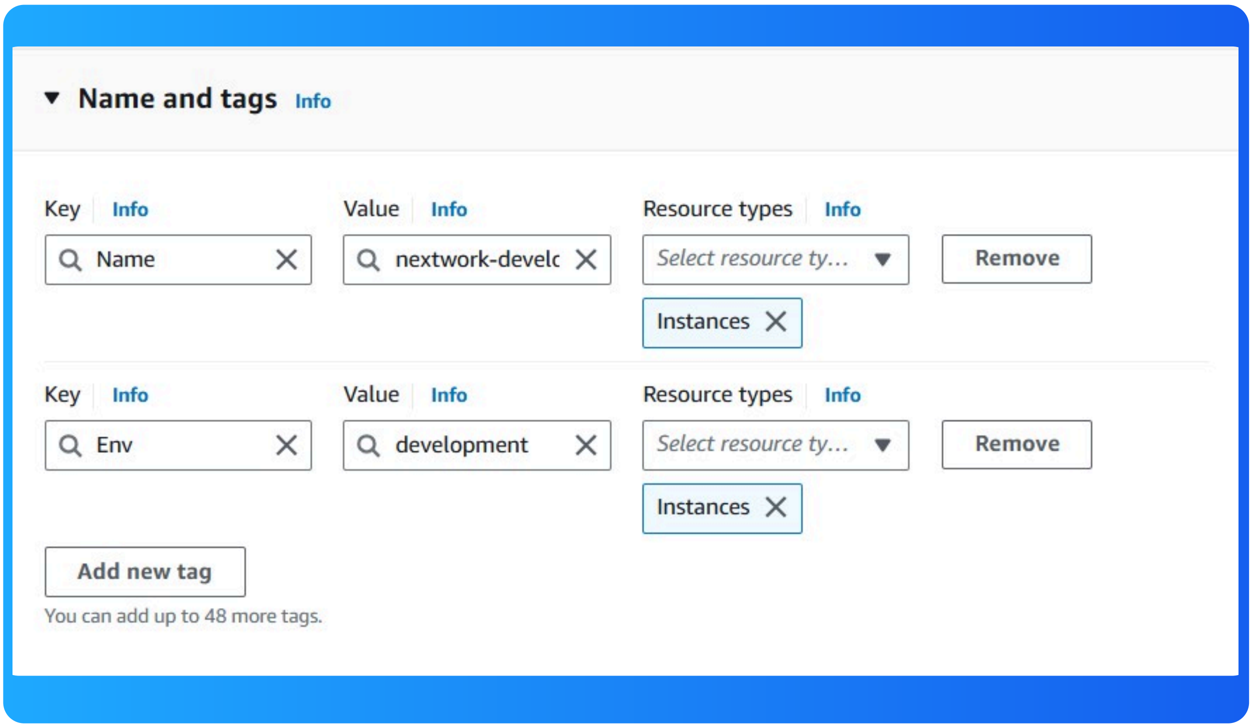
This project took me...

This project took me approximately one hour to complete.

Tags

Tags are key-value pairs assigned to AWS resources for easier identification, organization, cost tracking, and applying management policies across environments.

The tag I’ve used on my EC2 instances is called "Env." The values I’ve assigned for my instances are "production" for the production environment and "development" for the development environment.



▼ Name and tags Info

Key	Value	Resource types	
Name	nextwork-develc	Select resource ty… ▼ Instances ✕	Remove
Env	development	Select resource ty… ▼ Instances ✕	Remove

You can add up to 48 more tags.

IAM Policies

IAM Policies are rules that define permissions for users or services, specifying what actions they can perform on AWS resources.

The policy I set up

For this project, I’ve set up a policy using the JSON editor, allowing precise control over permissions and conditions.

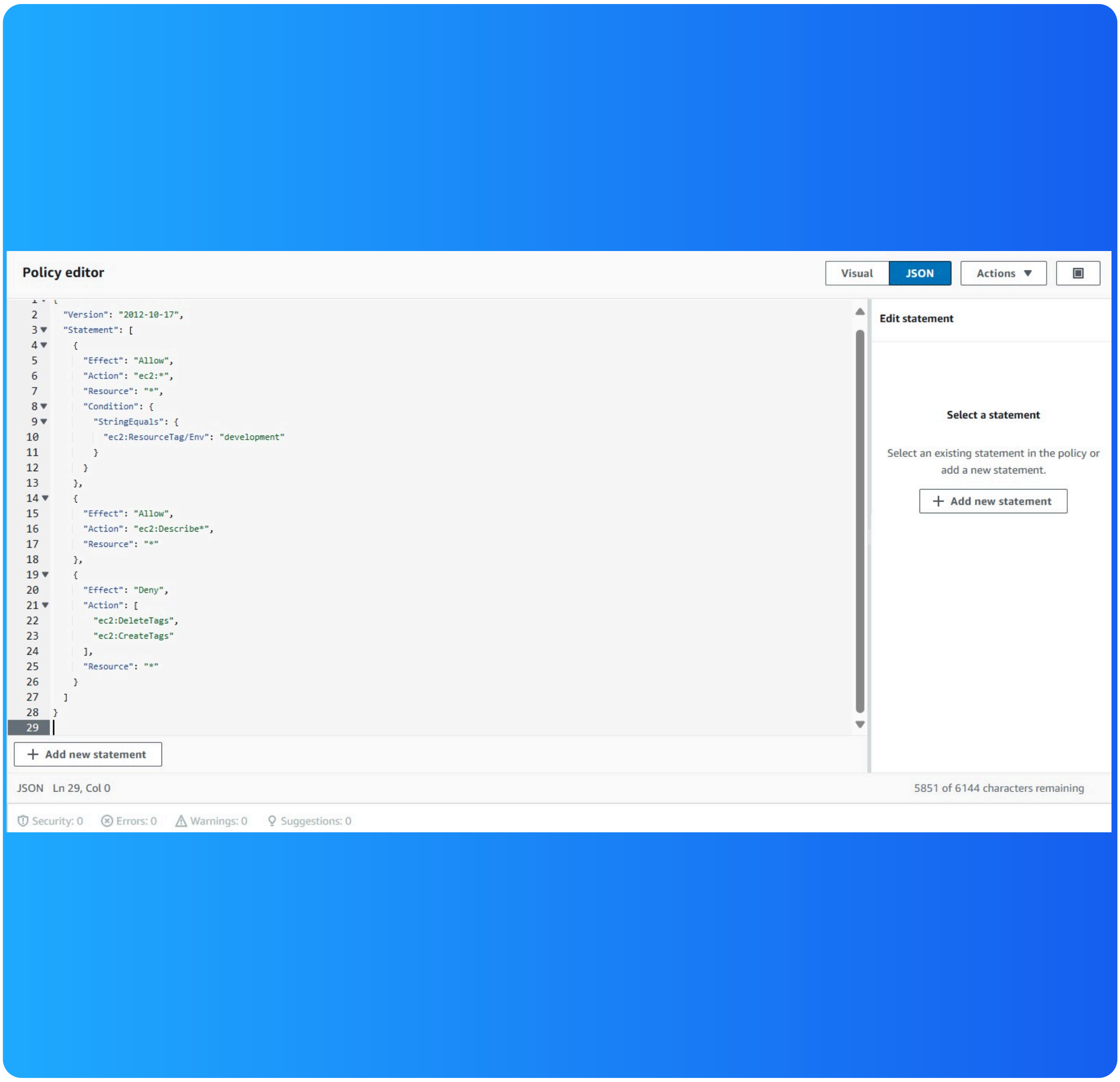
I’ve created a policy that allows EC2 actions on instances tagged as "development" while restricting certain actions like deleting tags.

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

The Effect, Action, and Resource attributes of a JSON policy mean whether actions are allowed or denied, the specific AWS actions permitted, and the resources the actions apply to.

Gabriel Mazer

My JSON Policy



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Env": "development"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": [
        "ec2:DeleteTags",
        "ec2:CreateTags"
      ],
      "Resource": "*"
    }
  ]
}
```

JSON Ln 29, Col 0

5851 of 6144 characters remaining

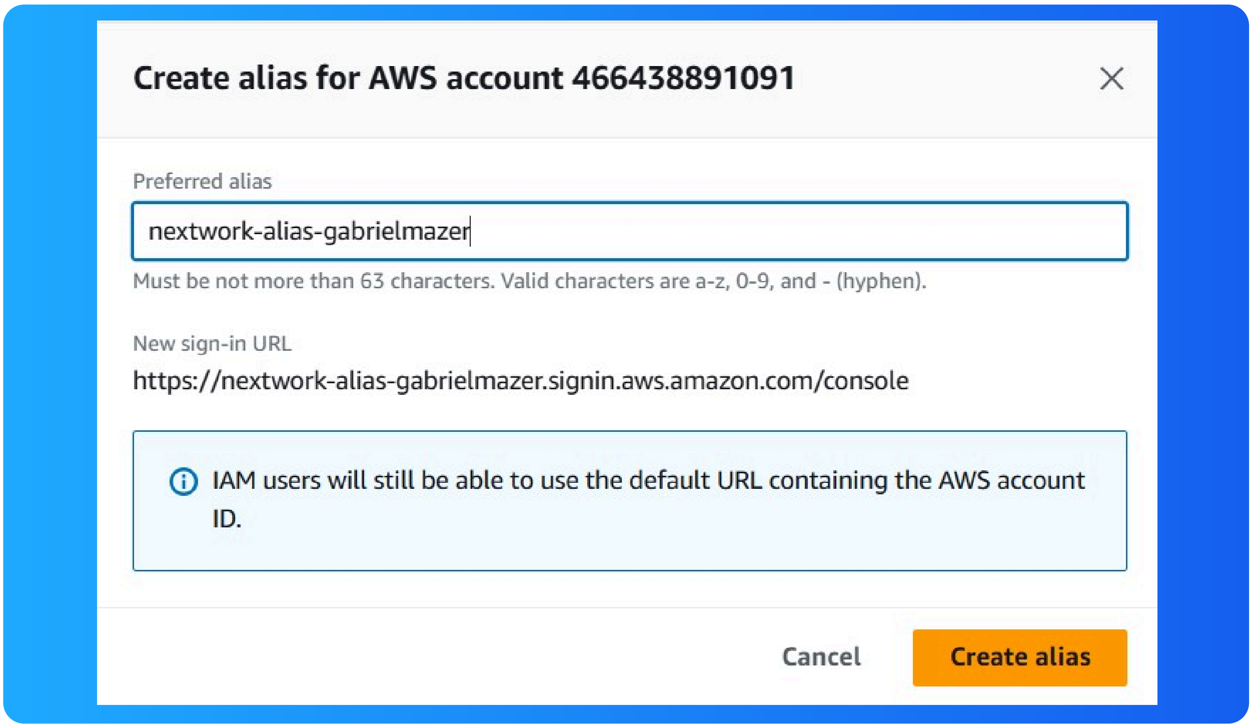
Security: 0 Errors: 0 Warnings: 0 Suggestions: 0

Account Alias

An account alias is a user-friendly name that replaces the default AWS account ID in your console login URL, making it easier to share and remember.

Creating an account alias took me just a few minutes to set up and apply.

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



Create alias for AWS account 466438891091

Preferred alias

nextwork-alias-gabrielmazer

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

New sign-in URL

https://nextwork-alias-gabrielmazer.signin.aws.amazon.com/console

ⓘ IAM users will still be able to use the default URL containing the AWS account ID.

IAM Users and User Groups

Users

IAM users are individual identities in AWS with specific permissions, allowing them to access and perform actions on AWS resources.

User Groups

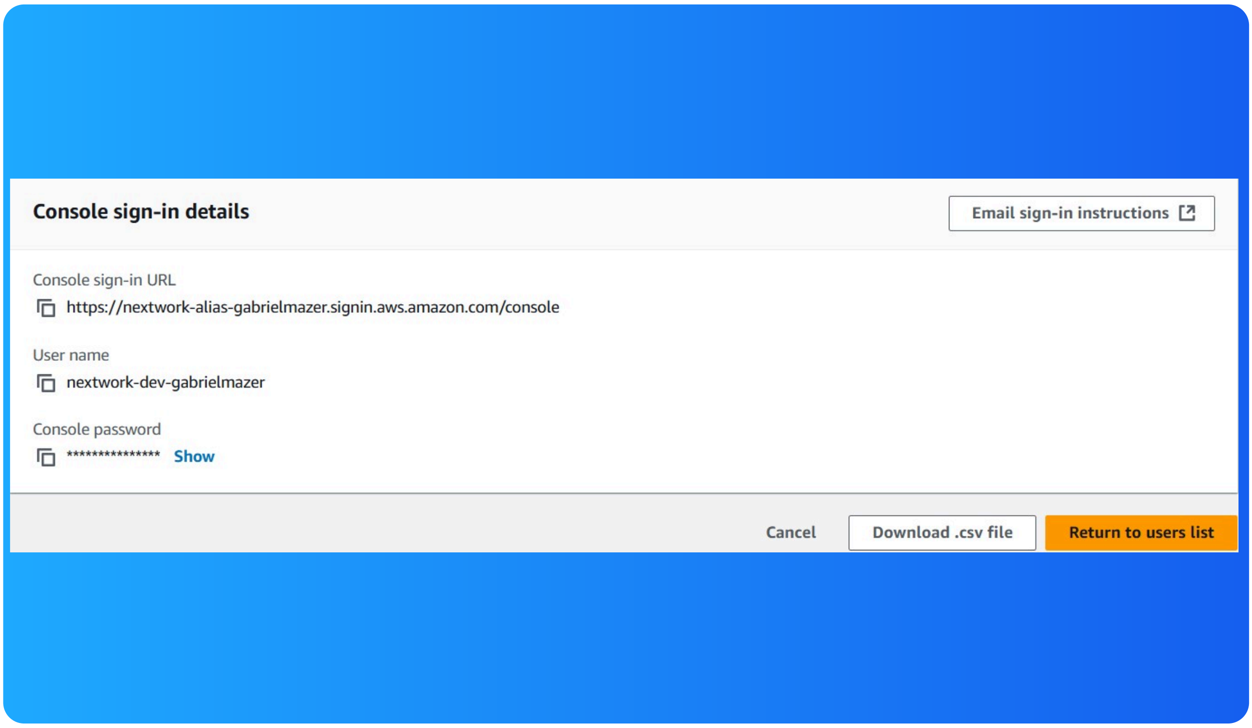
IAM user groups are collections of IAM users, making it easier to manage permissions by attaching policies to the entire group.

I attached the policy I created to this user group, which means all users in the group inherit the permissions specified by the policy.

Logging in as an IAM User

The first way is by emailing the login credentials, and the second is by providing the console sign-in URL and credentials directly.

Once I logged in as my IAM user, I noticed certain actions were restricted, with "Access Denied" messages for various AWS services, reflecting limited permissions.



Console sign-in details

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

User name
nextwork-dev-gabrielmazer

Console password
****

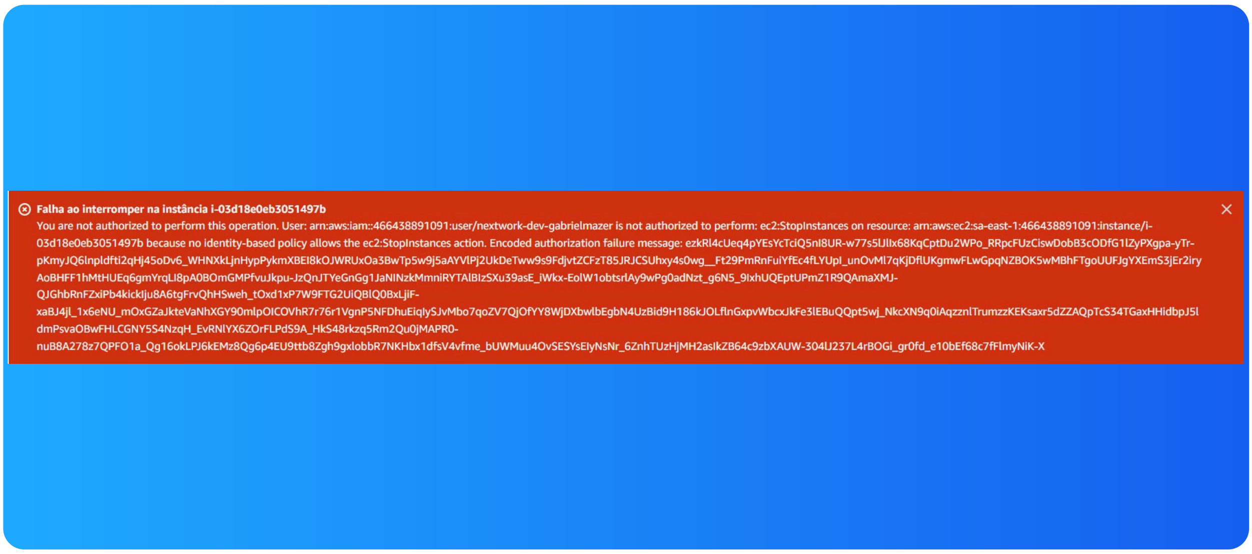
Gabriel Mazer

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 due to the policy restriction applied to resources tagged as "production."



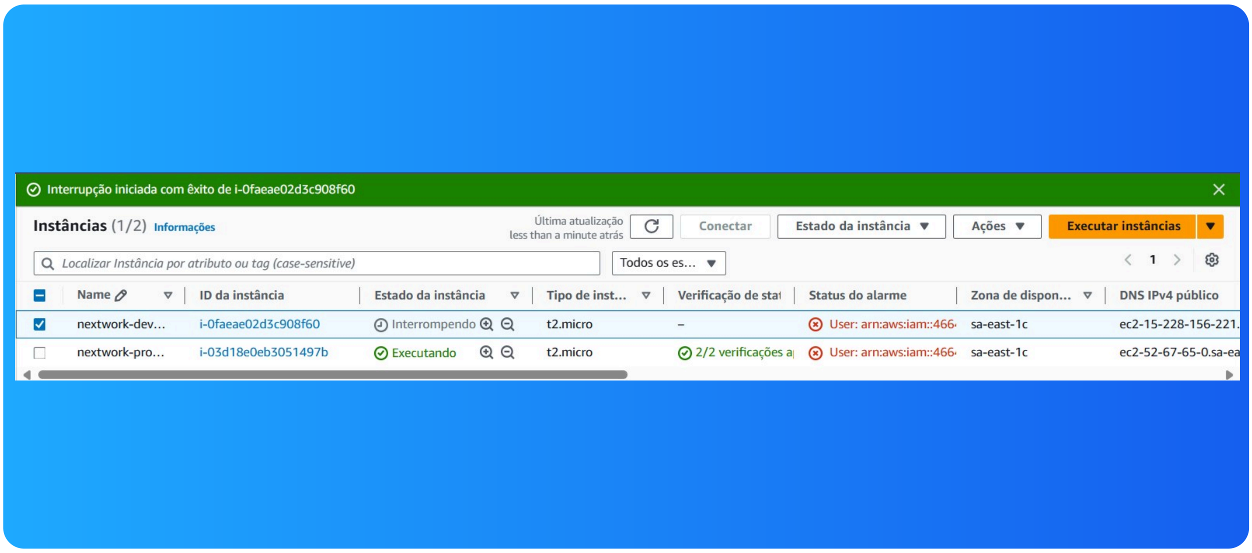
```
Falha ao interromper na instância i-03d18e0eb3051497b
You are not authorized to perform this operation. User: arn:aws:iam::466438891091:user/nextwork-dev-gabrielmazer is not authorized to perform: ec2:StopInstances on resource: arn:aws:ec2:sa-east-1:466438891091:instance/i-
03d18e0eb3051497b because no identity-based policy allows the ec2:StopInstances action. Encoded authorization failure message: ezkRl4cUeq4pYEsYcTciQ5nI8UR-w77s5lJlIx68KqCptDu2WPo_RRpcFUzCiswDobB3cODfG1lZyPXgpa-yTr-
pKmyJQ6lnpldfti2qHj45oDv6_WHNXkLjnHypPykmXBEI8kOJWRUxOa3BwTp5w9j5aAYVlPj2UkDeTww9s9FdjvtZCFzT85JRJCSUhxy4s0wg__Ft29PmRnFuiYfEc4fLYUpI_unOvMl7qKjDflUKgmwFLwGpqNZBOK5wMBhFTgoUUFJgYXEmS3jEr2iry
AoBHFF1hMtHUEq6gmYrqLI8pA0BOmGMPfvuJkpu-JzQnJTYeGnGg1JaNINzkMmniRYTAlBIzSXu39asE_iWkx-EolW1obtsrlAy9wPg0adNzt_g6N5_9IxhUQEptUPmZ1R9QAmaXMJ-
QJGhbRnFZxiPb4kickIju8A6tgFrvQhHSweh_tOxd1xP7W9FTG2UiQBlQ0BxLjiF-
xaBJ4jl_1x6eNU_mOxGZaJkteVaNhXGY90mlpOICOVhR7r76r1VgnP5NFDhuEiqIySJvMbo7qoZV7QjOfYY8WjDXbwlbEgbN4UzBid9H186kJOLflnGxpvWbcxJkFe3lEBuQQpt5wj_NkcXN9q0iAqzznlTrumzzKEKsaxr5dZZAQpTcS34TGaxHHidbpJ5l
dmPsvaOBwFHLCGNY5S4NzqH_EvRNlYX6ZOrFLPdS9A_HkS48rkzq5Rm2Qu0jMAPR0-
nuB8A278z7QPFO1a_Qg16okLPJ6kEMz8Qg6p4EU9ttb8Zgh9gxlobbR7NKHbx1dfsV4vfme_bUWMuu4OvSESYsEIyNsNr_6ZnhTUzHjMH2asIkZB64c9zbXAUW-304lJ237L4rBOGi_gr0fd_e10bEf68c7fFlmyNiK-X
```

Gabriel Mazer

Testing IAM Policies

Stopping the development instance

Next, when I tried to stop the development instance, the action succeeded, as the IAM policy allowed control over resources tagged as "development."



✓ Interrupção iniciada com êxito de i-0faeae02d3c908f60

Instâncias (1/2) Informações

Última atualização less than a minute atrás

☐	Name	ID da instância	Estado da instância	Tipo de inst...	Verificação de stat	Status do alarme	Zona de dispon...	DNS IPv4 público
☑	nextwork-dev...	i-0faeae02d3c908f60	Interrompendo	t2.micro	–	☒ User: arn:aws:iam::466	sa-east-1c	ec2-15-228-156-221.
☐	nextwork-pro...	i-03d18e0eb3051497b	✓ Executando	t2.micro	✓ 2/2 verificações a	☒ User: arn:aws:iam::466	sa-east-1c	ec2-52-67-65-0.sa-ea