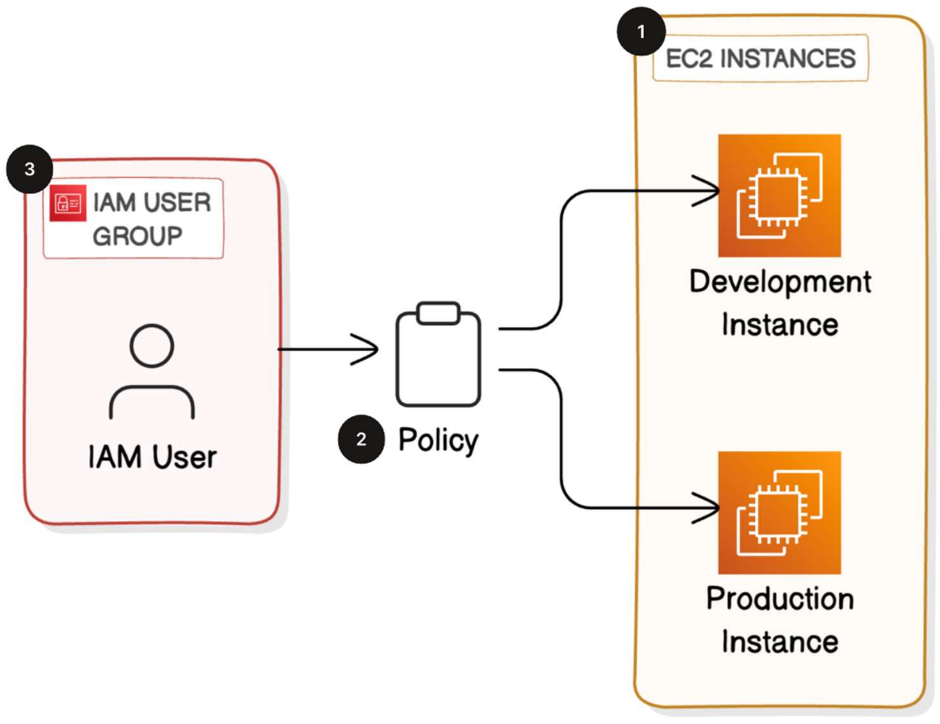
Cloud Security with AWS IAM

Using IAM to control access to AWS resources.



Introduction:

In this project, I will demonstrate how to use AWS IAM to control access and permission settings in my AWS project. I'm doing this project to learn about cloud security from the absolute foundations.

Tools and Concepts:

Services I used were Amazon EC2 and AWS IAM. Key concepts I learnt include IAM users, policies, user groups, and account alias. I also learnt how to use the Policy Simulator and how JSON policies work.

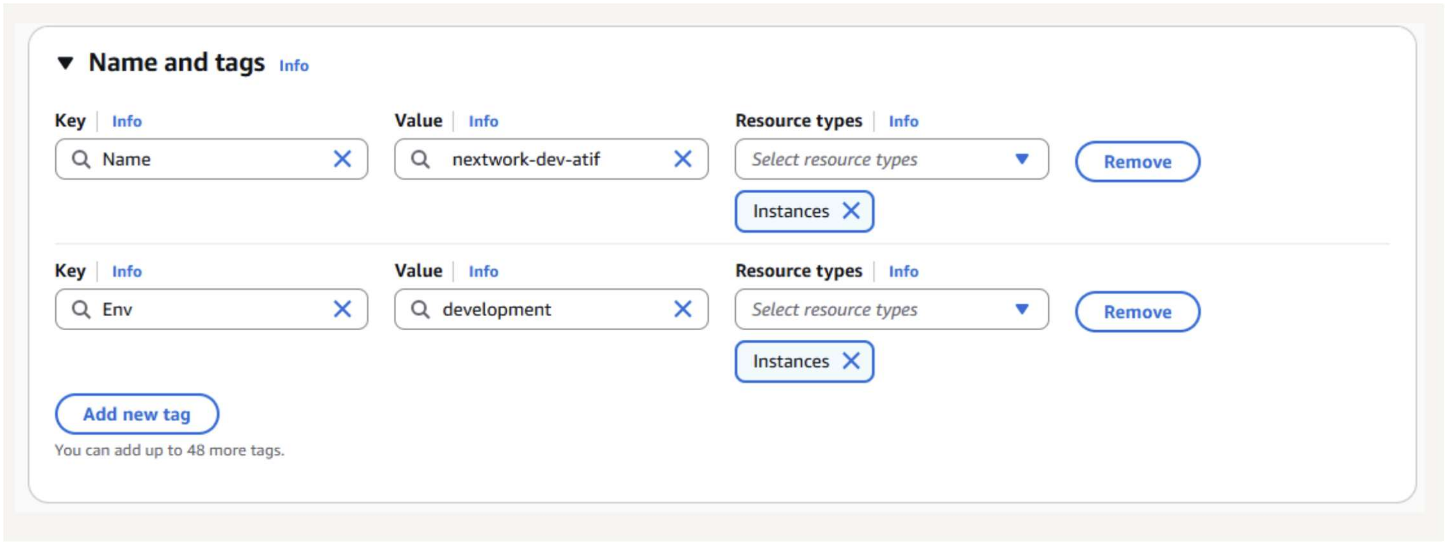
Project reflection:

This project took me approximately 1.5 hours including demo time. The most challenging part was understanding the IAM policy as it was with JSON and containing multiple statements. It was most rewarding to me to saw Permission denied for new user.

Tags:

Tags are organizational tools that let us label us 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. The value I've assigned for my instances are production and development.



Name and tags

Key	Value	Resource types
Name	nextwork-dev-atif	Instances
Env	development	Instances

You can add up to 48 more tags.

IAM Policies:

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

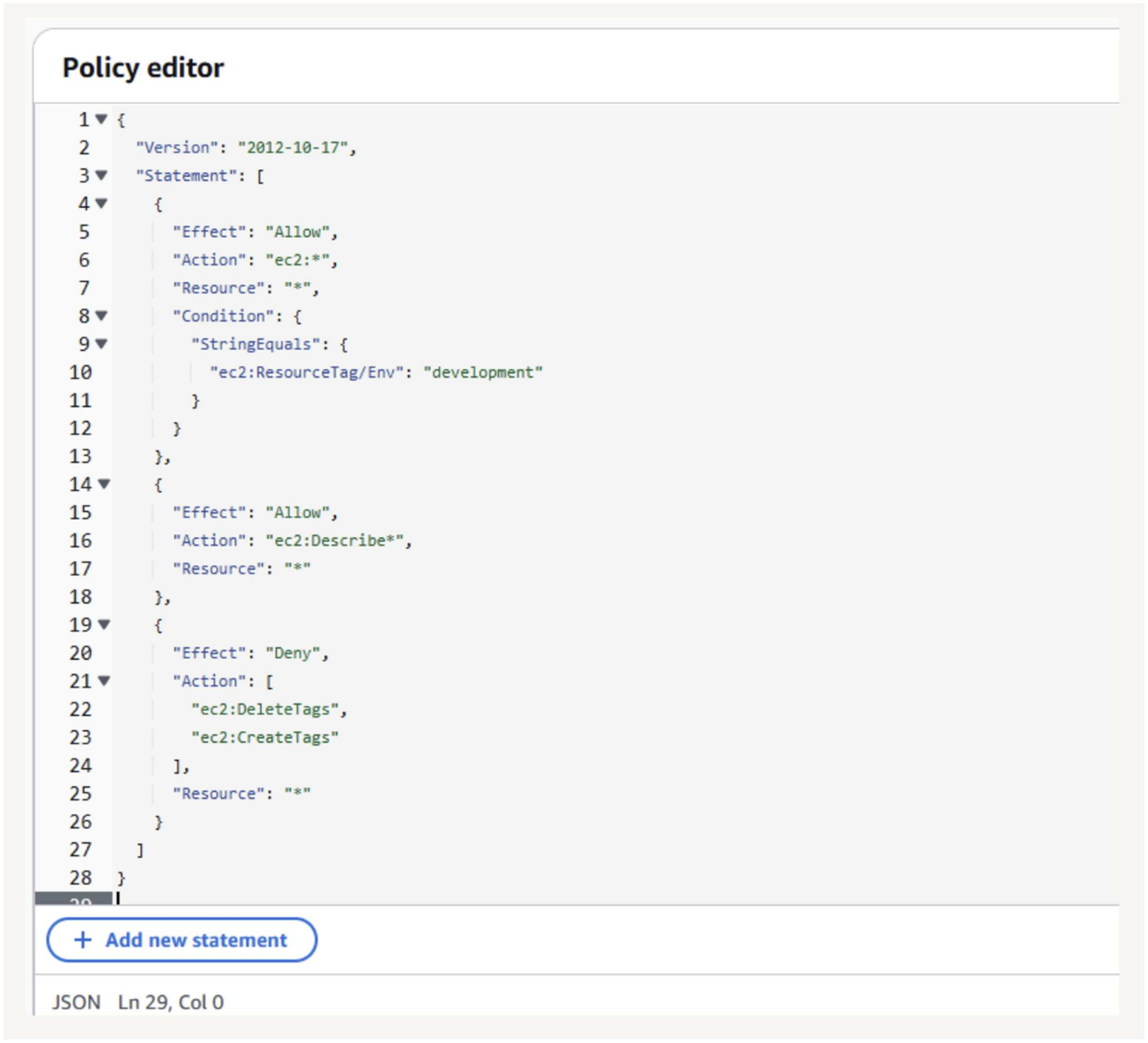
The policy I set up

For this project, I’ve set up a policy using JSON. I’ve created a policy that allows the policy holder (i.e. new user) to have permission to do anything they want to any instance tagged with development including seeing information for any instance. but are re denied access to delete/create any tags.

When creating a JSON policy, you must 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 hence Effect, what policy holder can/can't do hence Action, and specific resources that the policy relates to (i.e. Resource).

My JSON Policy

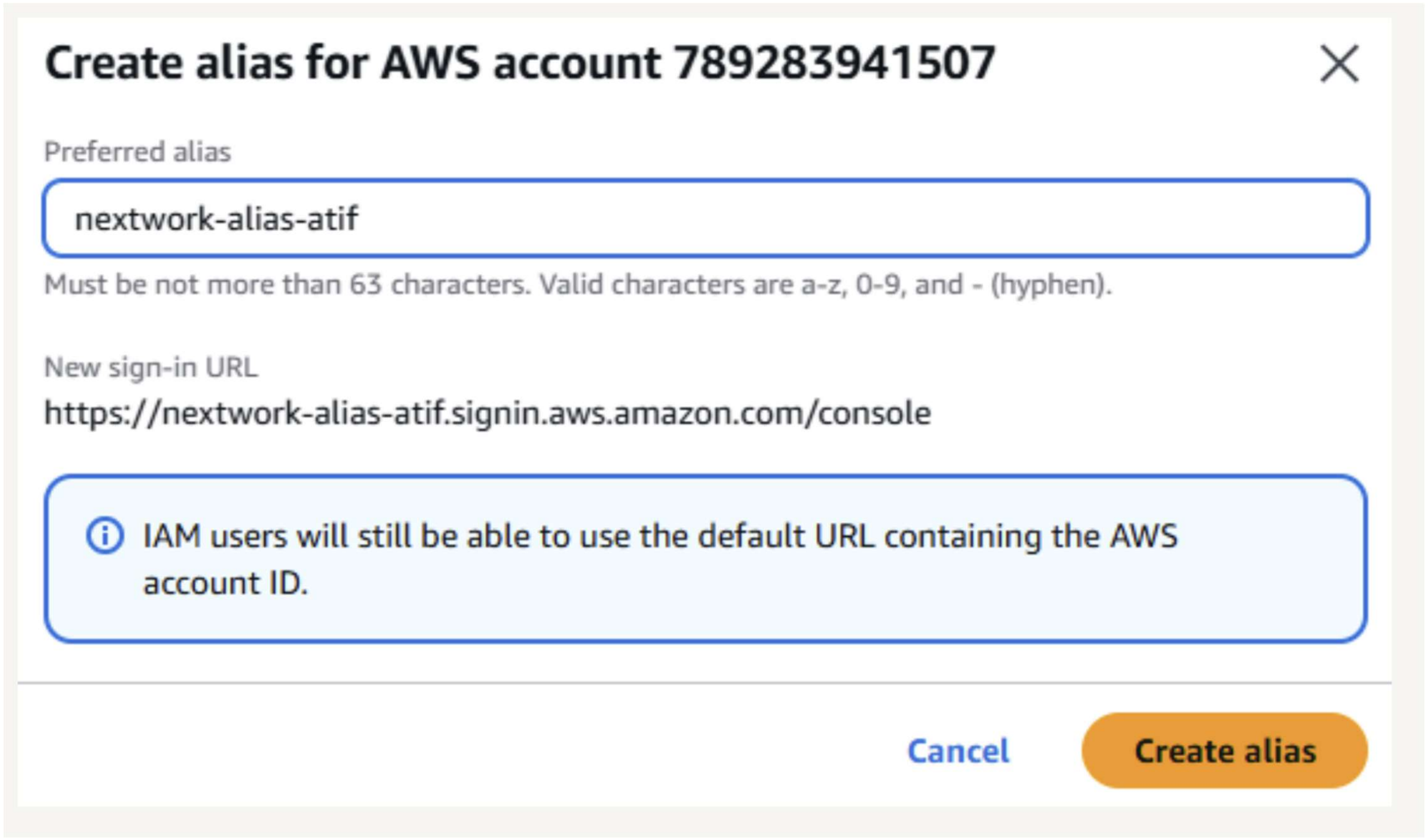


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Env": "development"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",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

Account Alias

An account alias is simply a nickname for our AWS account. Instead of a long account ID, we can now reference our account alias.

Creating an account alias took me 30 seconds - It’s a simple configuration in the IAM dashboard. Now, my new AWS console sign-in URL uses the alias instead of my account ID.



Create alias for AWS account 789283941507

Preferred alias

nextwork-alias-atif

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

New sign-in URL

https://nextwork-alias-atif.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 people/entities that have access/can login to our AWS account.

User Groups

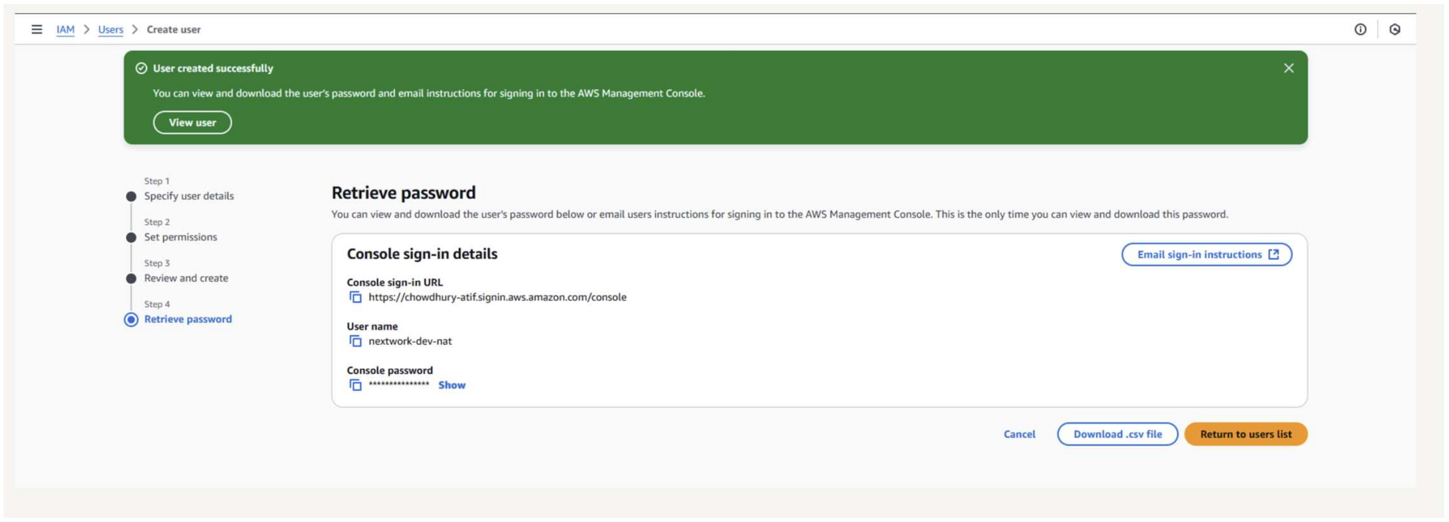
IAM user groups are like folders that collect IAM users so that we can apply permission settings at the group level.

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

Logging in as an IAM User

The first way is email sign-in instructions to the user, while the second way is to download a .csv file with the sign-in details inside.

Once I logged in as my IAM user, I noticed that our user is already denied to panels on the main AWS console dashboard. This was because we gave permission to develop EC2 instance, so our new user wouldn't have access to/even see anything else.



User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

Console sign-in URL
https://chowdhury-atif.signin.aws.amazon.com/console

User name
nextwork-dev-nat

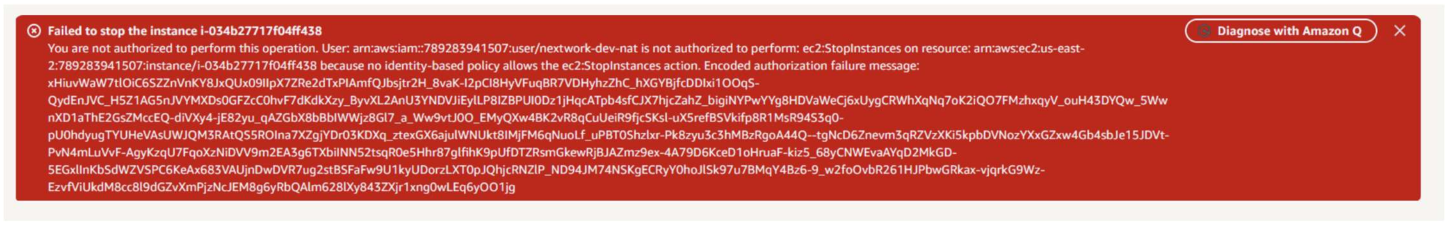
Console password
****

Testing IAM Policies

I tested my JSON IAM policy by attempting to stop both the development and the production instances.

Stopping production instant

When I tried to stop the production instance and had an error. This is because our production instance is tagged with production label, which is out of the scope of our permission policy new users are only allowed to do things to development instance



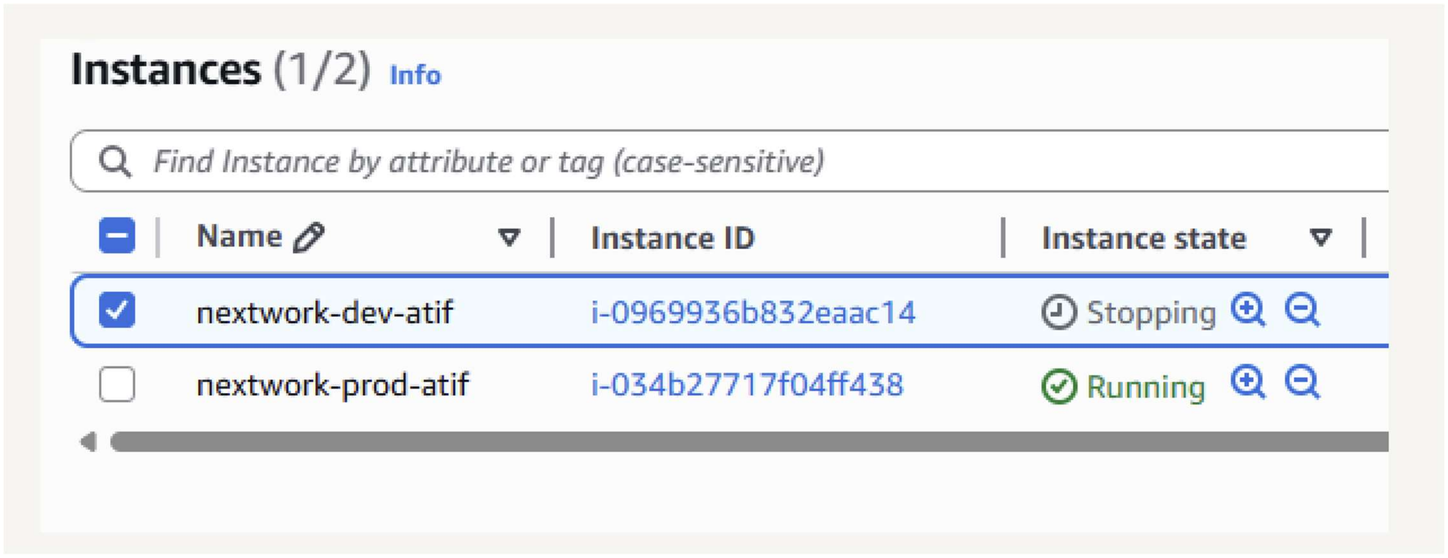
Failed to stop the instance i-034b27717f04ff438

```
You are not authorized to perform this operation. User: arn:aws:iam::789283941507:user/nextwork-dev-nat is not authorized to perform: ec2:StopInstances on resource: arn:aws:ec2:us-east-
2:789283941507:instance/i-034b27717f04ff438 because no identity-based policy allows the ec2:StopInstances action. Encoded authorization failure message:
xHiuvWaW7tlOiC6SZZnVnKY8JxQUx09IlpX7ZRe2dTxPIAmfQJbsjtr2H_8vaK-I2pCI8HyVFuqBR7VDHyhzZhC_hXGYBjfcDDlxi1OOqS-
QydEnJVC_H5Z1AG5nJVYMXDs0GFZcC0hvF7dKdkXzy_ByvXL2AnU3YNDVJiEylLP8IZBPUI0Dz1jHqcATpb4sfCJX7hjcZahZ_bigiNYPwYYg8HDVaWeCj6xUygCRWhXqNq7oK2iQO7FMzhxqyV_ouH43DYQw_5Ww
nXD1aThE2GsZMccEQ-diVXy4-jE82yu_qAZGbX8bBblWWjz8Gl7_a_Ww9vtJ0O_EMyQXw4BK2vR8qCuUeiR9fjcSKsl-uX5refBSVkifp8R1MsR94S3q0-
pU0hdyugTYUHeVAsUWJQM3RAtQS5ROIna7XZgjYDr03KDXq_ztexGX6ajulWNUkt8IMjFM6qNuoLf_uPBT0Shzlxr-Pk8zyu3c3hMBzRgoA44Q--tgNcD6Znevm3qRZVzXKi5kpbDVNozYXxGZxw4Gb4sbJe15JDVt-
PvN4mLuVvF-AgyKzqU7FqoXzNiDVV9m2EA3g6TXbiINN52tsqR0e5Hhr87glfihK9pUfDTZRsmGkewRjBJAZmz9ex-4A79D6KceD1oHruaF-kiz5_68yCNWEvaAYqD2MkGD-
5EGxlInKbSdWZVSPC6KeAx683VAUjnDwDVR7ug2stBSFaFw9U1kyUDorzLXT0pJQhjcRNZlP_ND94JM74NSKgECRyY0hoJlSk97u7BMqY4Bz6-9_w2foOvbR261HJPbwGRkax-vjqrkG9Wz-
EzvfViUkdM8cc8l9dGZvXmPjzNcJEM8g6yRbQAlm628lXy843ZXjr1xng0wLEq6yOO1jg
```

Testing IAM Policies

Stopping the development of instance

Next, when I tried to stop the development instance, I successfully changed the instance state to Stopping and then Stopped. This was because my permission policy allows the users (i.e. in the dev group) to stop instances.



Instances (1/2) Info

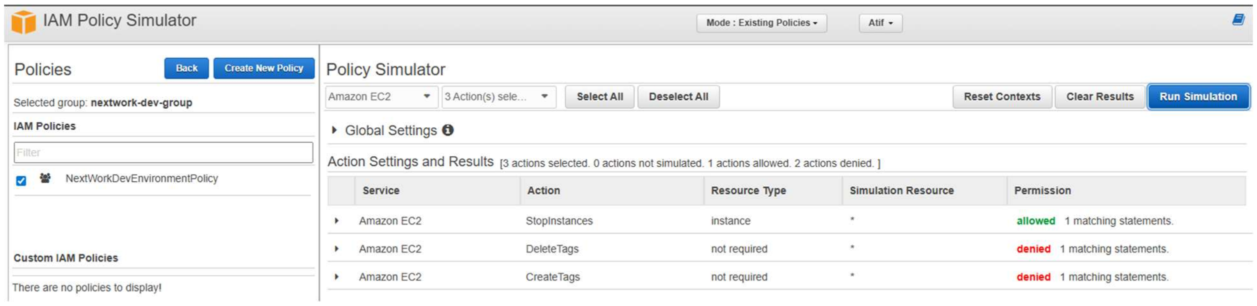
Find Instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state
✓	nextwork-dev-atif	i-0969936b832eaac14	Stopping
	nextwork-prod-atif	i-034b27717f04ff438	Running

IAM Policy Simulator

To extend my project, I am going to test my permission policies in a safer and more controlled way - a tool called the IAM Policy Simulator. I am doing this because having to stop instances and log into AWS accounts as other users is a bit disruptive.

So, the IAM Policy Simulator is a tool that let's us simulate actions and test permission settings by defining a specific user/group/role and the action we want to test for. It is useful for saving time when testing permission settings for new users.



Selected group: **nextwork-dev-group**

IAM Policies

NextWorkDevEnvironmentPolicy

Custom IAM Policies

There are no policies to display!

Amazon EC2 | 3 Action(s) sele...

Action Settings and Results [3 actions selected. 0 actions not simulated. 1 actions allowed. 2 actions denied.]

Service	Action	Resource Type	Simulation Resource	Permission
Amazon EC2	StopInstances	instance	*	allowed 1 matching statements.
Amazon EC2	DeleteTags	not required	*	denied 1 matching statements.
Amazon EC2	CreateTags	not required	*	denied 1 matching statements.

I set up the simulation for whether our dev user group has permission to stop instances and delete tags/create tags, the result was denied for both – we had to adjust the scope of the EC2 instances to ones that are tagged with "development". Once we applied that tag, permission was allowed.

Learning Outcomes:

- EC2 instances
- IAM Policies
- IAM Users and User Groups
- AWS Account Alias

Sources of Help:

Thank you learn.nextwork.org for helping with a step-by-step guide to complete this project.