



User Permissions with AWS IAM



Sadeesha Perera



Identity and Access Management (IAM)



 *Search IAM*

Dashboard

▼ **Access management**

User groups

Users

IAM

IAM

Se





Sadeesha Perera
[linkedin.com/sadeesha-perera](https://www.linkedin.com/sadeesha-perera)

NextWork.org

Introducing AWS IAM!

What it does & how it's useful

AWS IAM is like a manager. It's the one responsible for providing access to users/services with your data.

How I'm using it in today's project

It's useful when you want to manage a large group of users and allow others to do certain tasks within the specific range. Also, this adds another level of security to your resources.

This project took me...

Today, I'm using AWS IAM to grant certain privilege to a dev user from a dev group to control a particular instance.



Setting up tags

- I've set up two EC2 instances to test the effectiveness of the permission settings I'll set up in AWS IAM. I've used tags to label them.
- Tags are basically labels. A simple way to assign, manage and identify resources.
- The tag I've used on my EC2 instances is called "Name " and "Env ". The values I've assigned for my instances are " nextwork-production-sadeesha" for production and " nextworkdevelopment-sadeesha" for development .

How **tags** are set up for my EC2 instances

▼ **Name and tags** [Info](#)

Key	Info	Value	Info	Resource types	Info	
<input type="text" value="Name"/>	✕	<input type="text" value="nextwork-develc"/>	✕	<input type="text" value="Select resource ty..."/>	▼	<input type="button" value="Remove"/>
				<input type="text" value="Instances"/>	✕	

Key	Info	Value	Info	Resource types	Info	
<input type="text" value="Env"/>	✕	<input type="text" value="development"/>	✕	<input type="text" value="Select resource ty..."/>	▼	<input type="button" value="Remove"/>
				<input type="text" value="Instances"/>	✕	

You can add up to 48 more tags.



IAM Policies

- IAM Policies are simply just rules that define what actions are allowed or denied.
- For this project, I've set up a policy using the JSON method.
- I've created a Policy that allows a certain action to be done in all instances, and within all the scope of resources that is defined.
- When writing JSON Policy statements, I have to specify the:
 - Effect: can either allow or deny but deny has the priority.
 - Action: allows/denies action within the policy.
 - Resource: to select which resources a policy applies to.



The **policy** I've set up in the IAM Policies page!

The screenshot shows the AWS IAM 'Specify permissions' page. At the top, there is a dark header bar with a search icon and the text '[Alt+S]'. Below this, the main heading is 'Specify permissions' with an 'Info' link. A subtitle reads: 'Add permissions by selecting services, actions, resources, and conditions. Build permission'. The 'Policy editor' section contains a JSON policy document with three statements. A blue arrow points from the text above to the top of the 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"
```



Sadeesha Perera
linkedin.com/sadeesha-perera

NextWork.org

AWS Account Alias

- New users can get access to my AWS Account through a unique URL created for my account's Account ID.
- An account alias is a unique name given to the user.
- Creating an account alias took me around 10 minutes.
- Now, my new AWS console sign-in URL is [].

Create alias for AWS account 992382852391


Preferred alias

nextwork-alias-sadeesha

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

New sign-in URL

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

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

Cancel

Create alias



IAM Users + User Groups

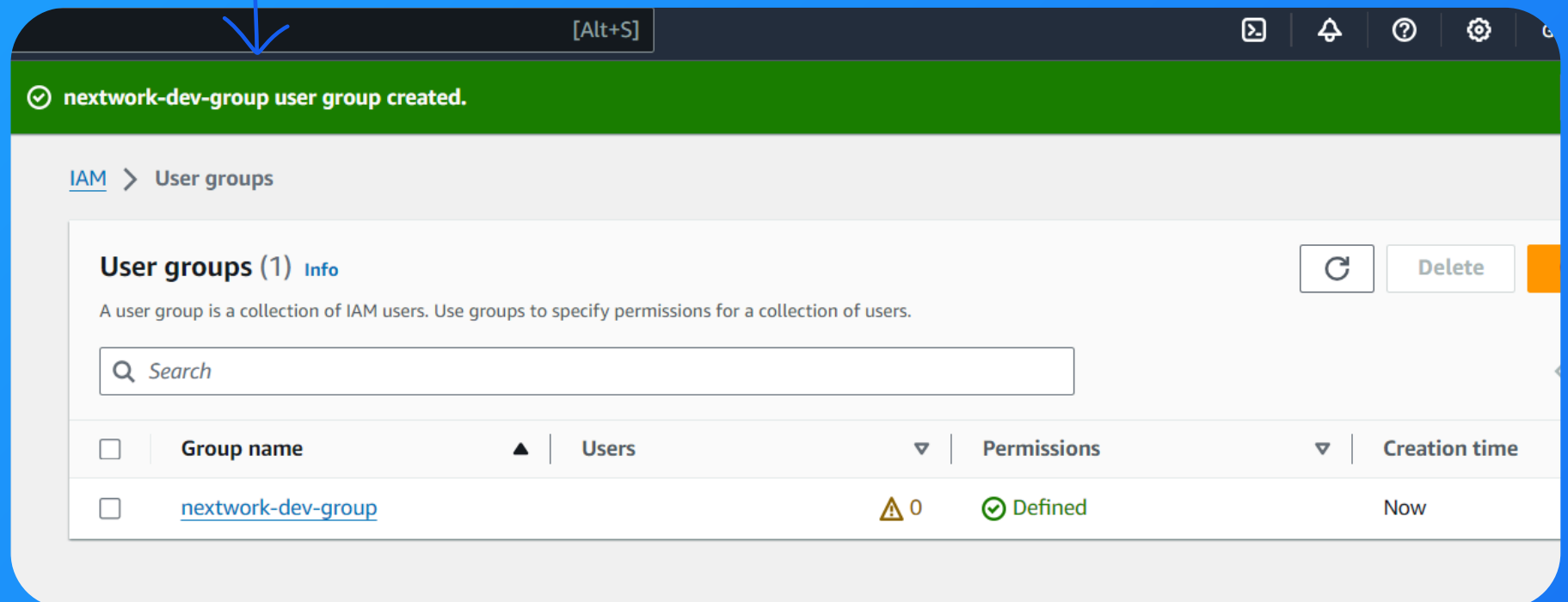
- IAM Users are users that fall under a specific group. These users are the people who will get access to my resources from my AWS main account.
- I also created a User Group. User Groups are useful for managing permissions and making sure that there are consistent privileges granted to the users within the group.
- My User Group is called “nextwork-dev-group”. I attached the Policy I created to this User Group, which means the allowed actions within the policy are carried over to the group, allowing them to gain access to resources.
- When I created a new User, I had to tick a checkbox that provides users access to the AWS management console, because without that access, they'll have to go through other methods to gain that access.
- Once my new user was set up, there were two ways I could share its sign-in details: either copy and paste it or download the CSV file.
- My new user had a unique sign-in URL nextwork-dev-sadeesha



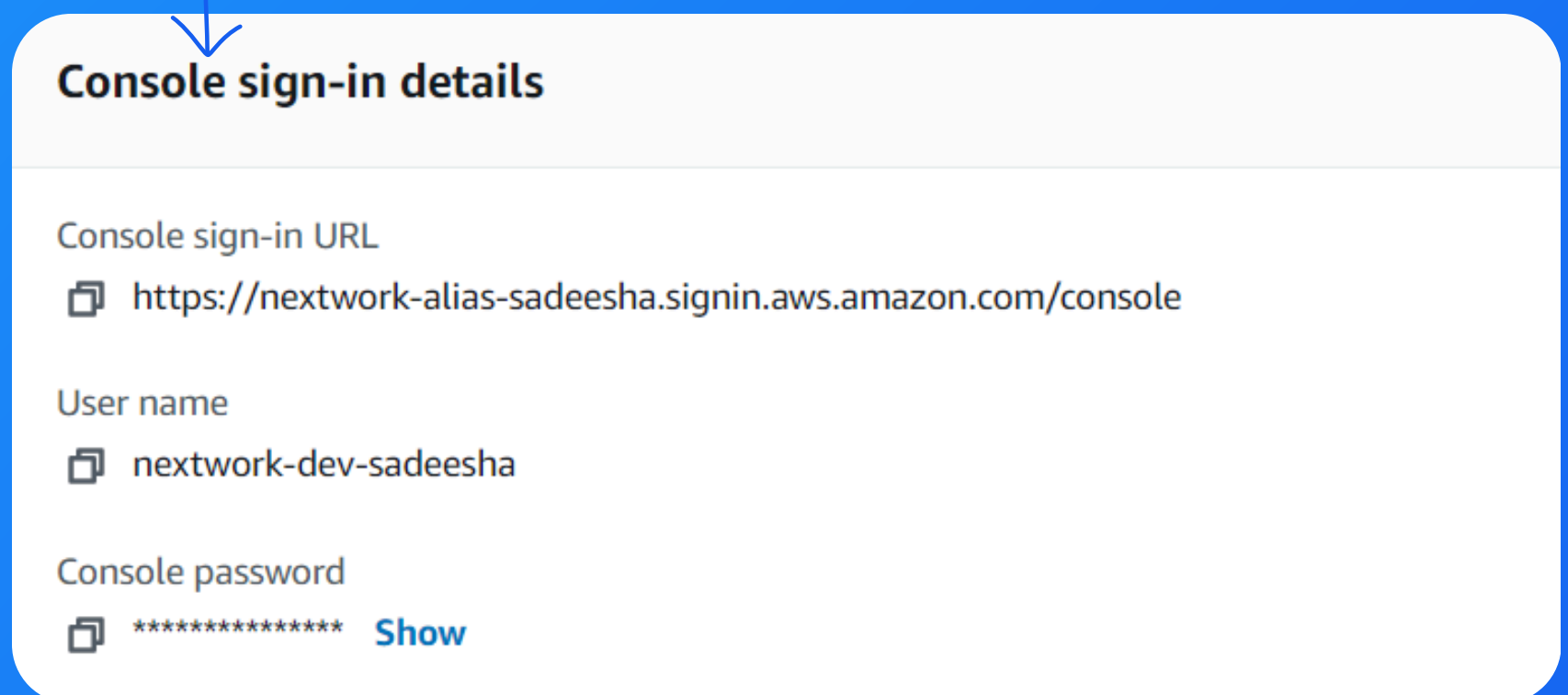
Sadeesha Perera
linkedin.com/sadeesha-perera

NextWork.org

My User Group!



My User's sign-in details!





IAM User in action

- Now with my IAM Policy, IAM User Group and IAM User all set up, let's put it all together! To do this, I logged into my AWS account as a new user.
- To log in as my IAM User, I copied and pasted the URL provided and used my provided credentials to log in.
- Once I logged in, I noticed that a lot of things were access denied for this new user.

Sign in as IAM user

Account ID (12 digits) or account alias

nextwork-alias-sadeesha

IAM user name

nextwork-dev-sadeesha



Sadeesha Perera
[linkedin.com/sadeesha-perera](https://www.linkedin.com/sadeesha-perera)

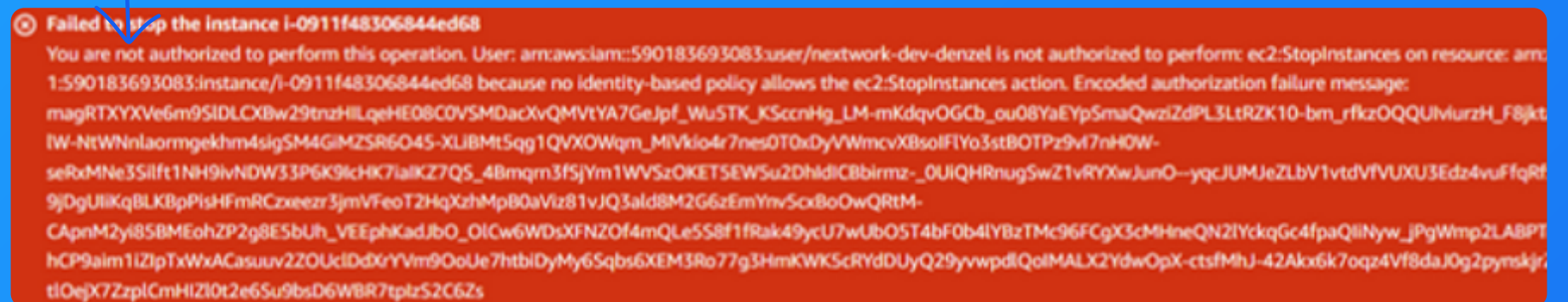
NextWork.org

IAM Policies in action

- Then, I tested the JSON IAM policy I set up by trying to stop the instances.
- When I tried to stop the production instance, it didn't work. I keep getting an error that says failed to stop the instances, and that I don't have the authority to do so.



Woah! A **red fail banner** pops up if I stop the production instance



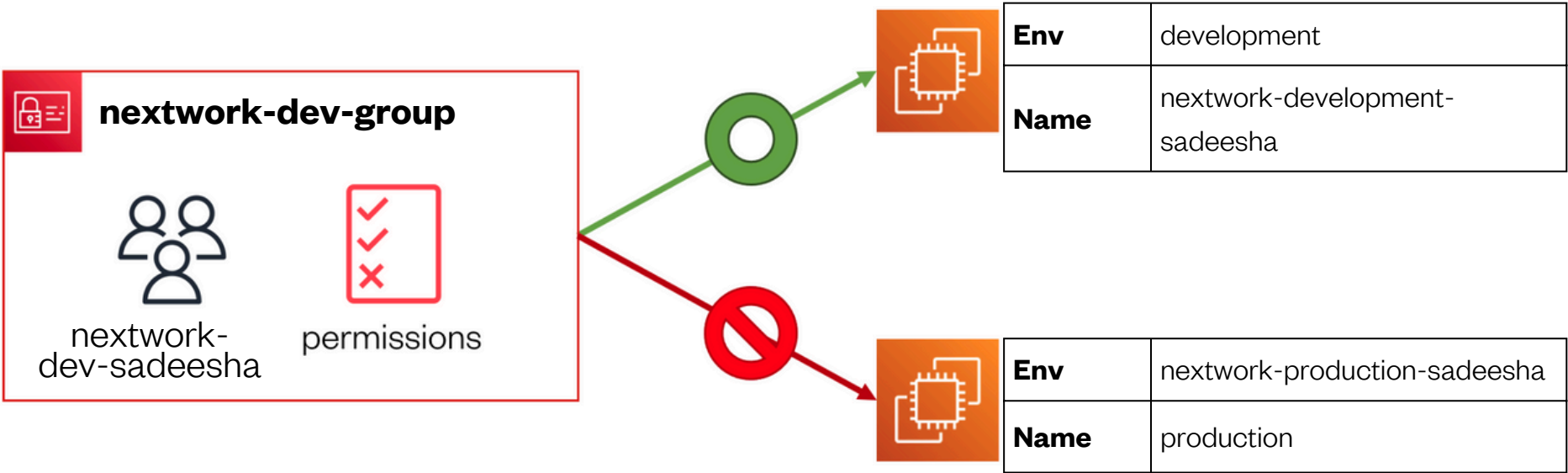
Phew! A **green success** banner pops up if I stop the development instance





To Summarise

- I created:
 - An IAM User Group called AAA with defined permissions using an IAM Policy
 - An IAM User called BBB that is added to the user group
 - An EC2 instance with the Env tag WWW and Name XXX
 - An EC2 instance with the Env tag YYY and Name ZZZ
- The users are allowed to stop the development instance, but the user cannot control the state of the production instance.





My key learnings

- 1 IAM Policies are basically just rules set to provide privileges to users and perform a certain action. It also secures the resources in your main AWS account by limiting the actions a certain user in a group can perform.
- 2 IAM users are the people that are working with the main user that are granted access to the resources.
- 3 IAM group is like a folder or organizer of users for specific tasks. They are grouped based on the policy provided by the main user.
- 4 an AWS Alias is just a username given to the users under a group. It is a unique identifier from which group the user belongs to.



Everyone should be in a job they love. *yes!*

Check out community.nextwork.org for more free projects

