

# MANAGING PERMISSIONS WITH AWS IAM



**Megha Naik**



<https://www.linkedin.com/in/naikmegha>

# WHAT IS AWS IAM?

## What it does:

- AWS IAM is a Identity and Access Management service used to manage access to your AWS resources by users/services. It restricts access to critical resources based on the principle of least privilege.

## Why it's useful:

- Helps organisations meet compliance requirements by providing detailed audit logs and enforcing access controls.

## How I'm using it in today's project:

- Setting up EC2 instance to grant access to user groups, while using tags to test permission settings.



**Megha Naik**

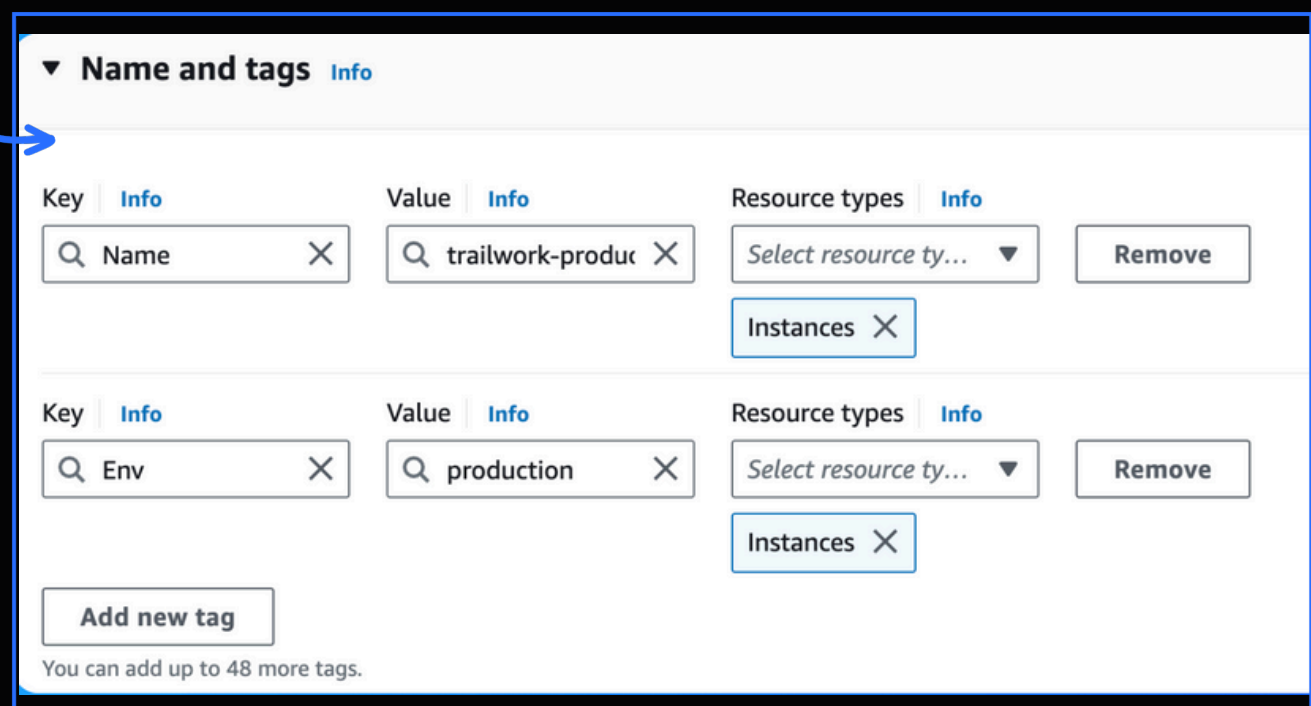


<https://www.linkedin.com/in/naikmegha>

# SETTING UP TAGS

- I have set up two EC2 instances to test the effectiveness of the permission settings set up in AWS IAM. I have used tags to label them.
- Tags are keywords/labels that you can attach to a resources like EC2 instance. It helps you to identify resources when troubleshooting.
- The tag I've used on my EC2 instances is called "Env". The value I've assigned for my instances are "production" or "development". By applying tags, I can easily locate development and production instances.

How the tags are set up for my EC2 instances



▼ Name and tags Info

Key	Value	Resource types	
Name	trailwork-produc	Select resource ty...	Remove
		Instances	
Env	production	Select resource ty...	Remove
		Instances	

Add new tag

You can add up to 48 more tags.



**Megha Naik**



<https://www.linkedin.com/in/naikmegha>



# IAM POLICIES

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

- IAM Policies are rules for those using your AWS resources. It's about giving permissions to those users, groups or roles indicating the do's and don'ts for certain resources and when the rules apply.
- For this project, I've set up a policy using JSON by switching from the default visual editor. I've created a Policy that allows actions in the Env - development tag.
- It denies the ability to create and delete tags for all instances. When writing JSON Policy statements, you have to specify the:
  - **Effect:** 2 values - Allow or Deny, indicating that the policy allows or denies an action. In the JSON script it indicates "Allow", the statement is allowing an action.
  - **Action:** a list of actions that the policy allows or denies. In the JSON script, "ec2:\*" indicates all actions possible with EC2 instances are allowed. The "\*" means all.
  - **Resource:** specifies which resources the policy applies to. "\*" means all resources.
  - **Condition Block:** specifies which policy is in action. In this case, Env-development is the tagged resource. In line 25 of the JSON script, it indicates all resources tagged with the Env-development are impacted.

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"
24      ],
25      "Resource": "*"
26    }
27  ]
28 }
```

+ Add new statement



**Megha Naik**



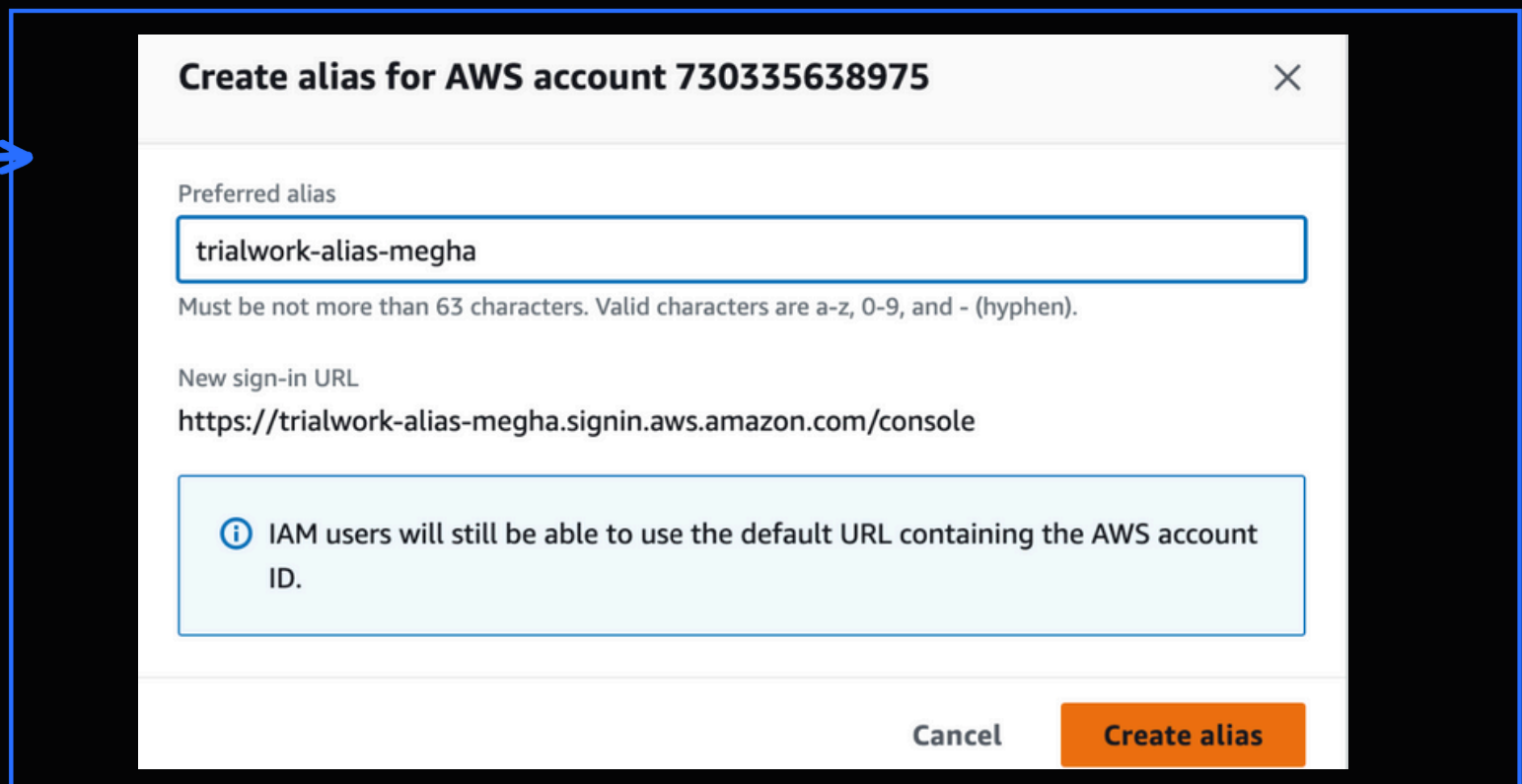
<https://www.linkedin.com/in/naikmegha>



# AWS ACCOUNT ALIAS

- When new users get onboarded onto my AWS account, they used to sign into a unique URL created for my account's long and cryptic Account ID. This can be cumbersome to remember and type in.
- An account alias is a user-friendly name that replaces your lengthy AWS account ID in the sign-in URL. It makes the sign-in process easier for new users and helps to avoid any confusion caused by the long account ID.
- Creating an account alias took me less than 10 seconds. Now, my new AWS console sign-in URL is  
**<https://trialwork-alias-megha.signin.aws.amazon.com/console>**

You get to set up  
your own account  
alias name!



**Create alias for AWS account 730335638975**

Preferred alias

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

New sign-in URL

<https://trialwork-alias-megha.signin.aws.amazon.com/console>

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

Cancel **Create alias**



**Megha Naik**

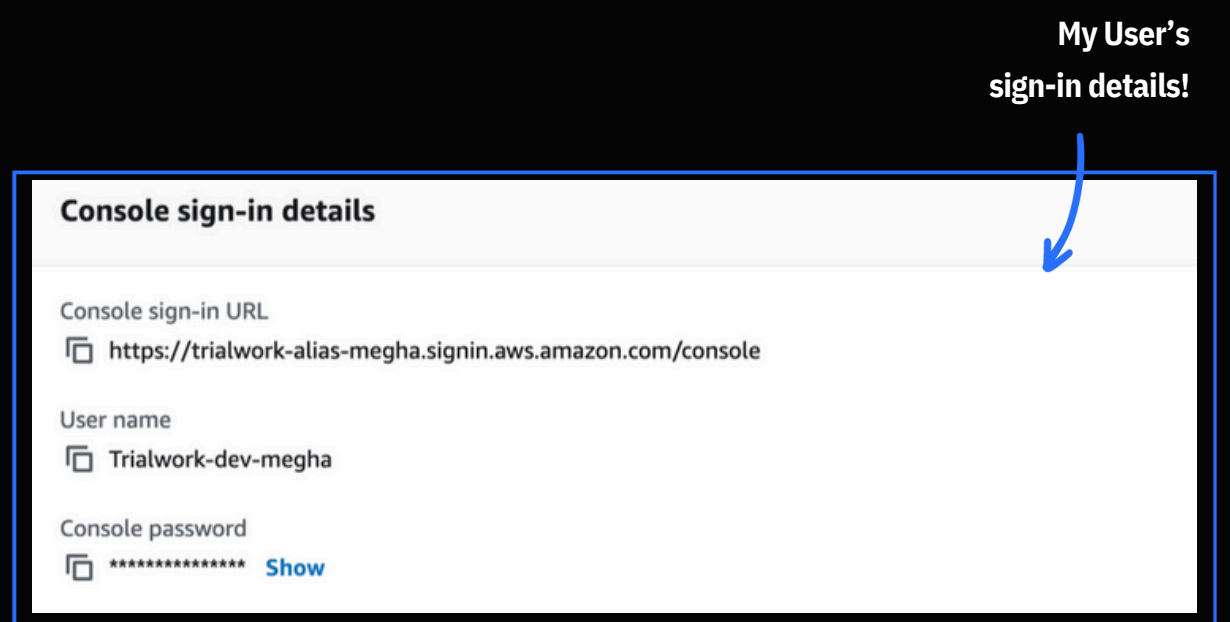
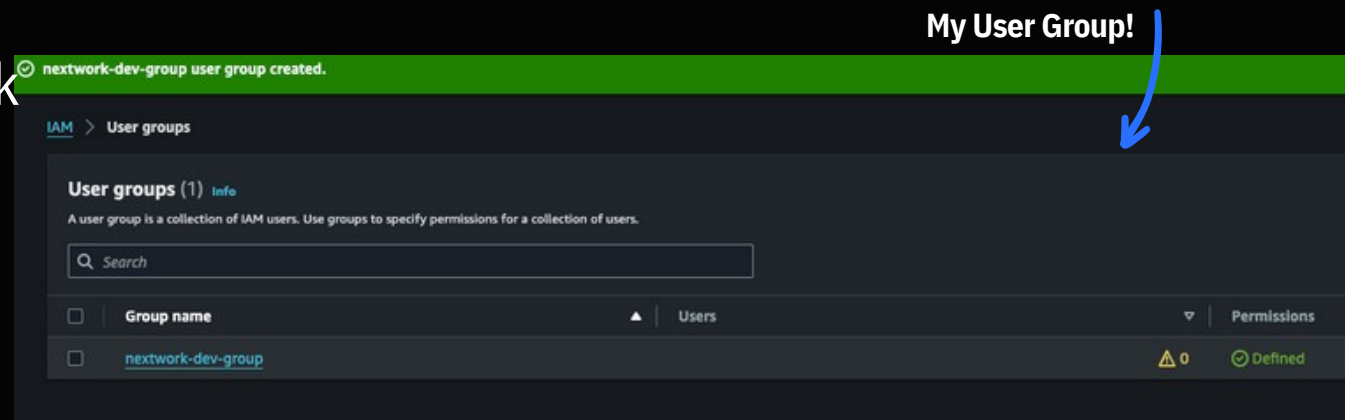


<https://www.linkedin.com/in/naikmegha>



# IAM USERS + USER GROUPS

- **IAM Users** are collections of IAM users.
- I also created a **User Group**. User Groups are useful for managing permissions for all users at the same time by making policies for the group instead of an individual user.
- My User Group is called nextwork-dev-group. I attached the Policy I created to this User Group, which means the rules created in the JSON script will apply to this group and its users. The only access this group has is to the development environment and not production.
- When I created a new User, I had to tick a checkbox that connected the group to the policy (NextWorkDevEnvironmentPolicy), to allow users in the group to gain access to the production environment.
- Once my new user was set up, there were two ways I could share its sign-in details: Email and downloading the .csv file
- My new user had a unique sign-in URL - this is my Account Alias at work!



**Megha Naik**



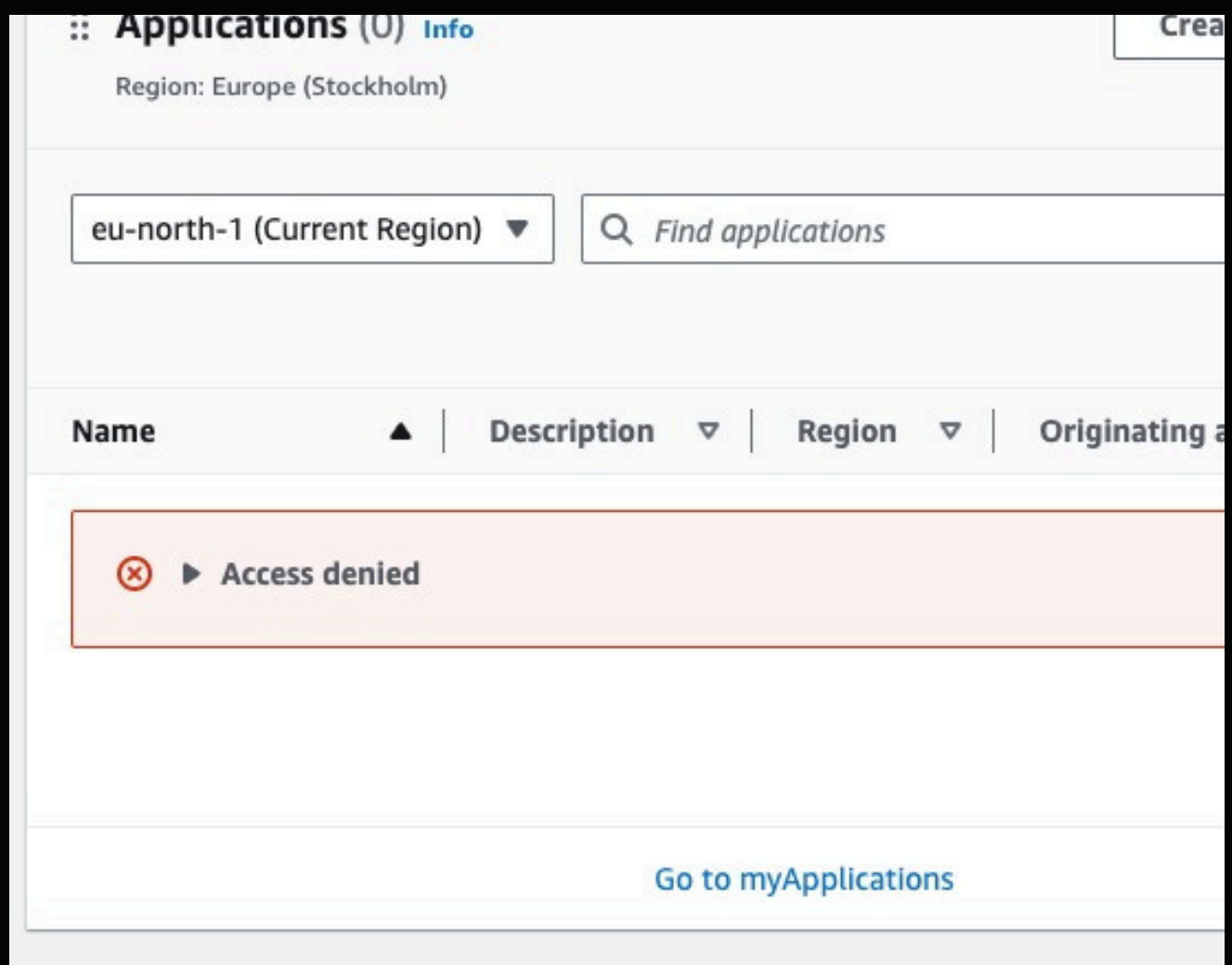
<https://www.linkedin.com/in/naikmegha>



# 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 the new user.
- To log in as my IAM User, I copied the URL into an incognito browser, used my username for the IAM and the copied the auto-generated password.
- Once I logged in, I noticed that most of the services had an access denied warning on them.

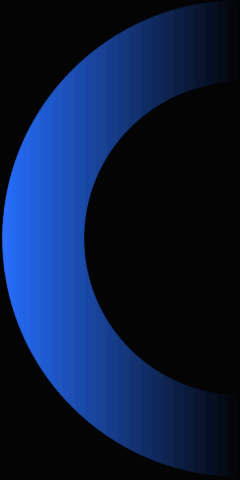
Some of my dashboard's panels showed **access denied!**



**Megha Naik**



<https://www.linkedin.com/in/naikmegha>



# IAM POLICIES IN ACTION

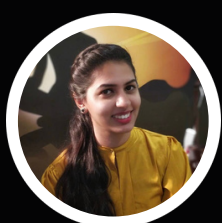
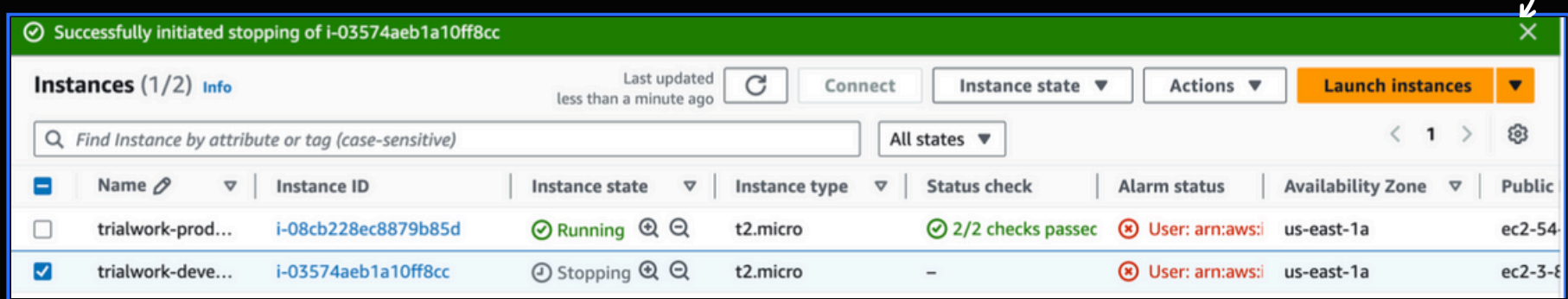
- Then, I tested the JSON IAM policy I set up by heading into the production environment and manage the instance state.
- When I tried to stop the production instance, the error message banner popped up. It said that I was not authorised to conduct that sort of operation.

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



- Next, when I tried to stop the development instance, I performed the same steps as before, once I said stop it proved to be a success. This is due to having access to the development environment.

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



**Megha Naik**



<https://www.linkedin.com/in/naikmegha>



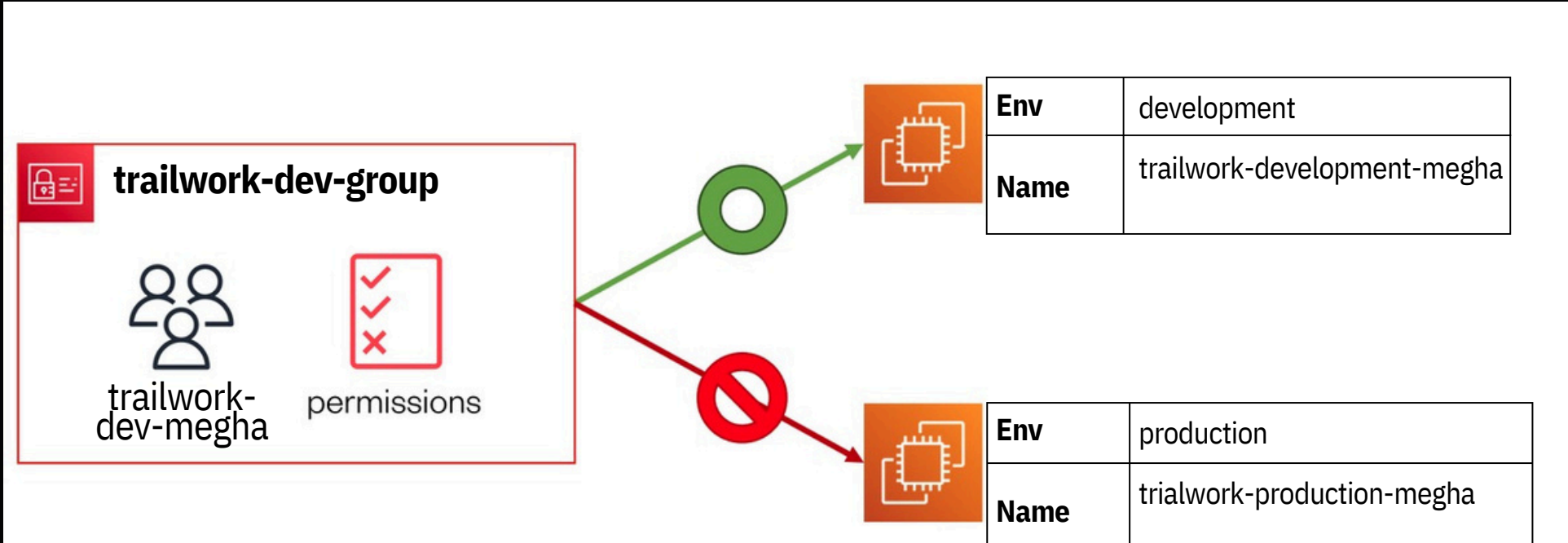




# TO SUMMARISE

I created:

- An IAM User Group called **trailwork-dev-group** with defined permissions using an IAM Policy.
- An IAM User called **trialwork-dev-megha** that is added to the user group
- An EC2 instance with the Env tag **development** and Name **trialwork-development-megha**
- An EC2 instance with the Env tag **production** and Name **trialwork-production-megha**
- In the **prodution EC2 instance** user can do stop, start, terminate. but in the **development instance** user cannot change the state of the instance



**Megha Naik**



<https://www.linkedin.com/in/naikmegha>

# My Key Learnings

**01** **What are IAM Policies?** A set rules that definewhich users or groups has authorisation and access to resources.

**02** **What are IAM User Groups? Why would you create one?**  
A person that has permission to use your AWS resources. The group where IAM Users are stored, each group is given access and authorisation to resources the group is not denied into or does not have permission to use.

**03** **What is an AWS Account Alias?**  
A user-friendly name that you can assign to your AWS account to simplify the login process.

**04** I learnt that you do not need to use the conventional method of testing instances in progress, there is a built in simulator to test user access.



**Megha Naik**



<https://www.linkedin.com/in/naikmegha>

# Final Thoughts...

- This project took me about just over a hour.
- Delete EVERYTHING at the end! Let's keep this project free.)
- One thing I didn't expect was how to create and give access to users and groups.
- Now that I know how IAM could be used to enhance security and permissions in my AWS account, some real-world use cases of what I've learnt are creating different clientele environments for starting a business in web development and cloud based solutions.



**Megha Naik**



<https://www.linkedin.com/in/naikmegha>

# Find this helpful?



Like this post



Leave a comment



Save for later



Let's connect!



**Megha Naik**



<https://www.linkedin.com/in/naikmegha>

Thanks NextWork for the  
free project guide!

 **NEXTWORK**