

# ***Cloud Security with AWS IAM***

# Cloud Security with AWS IAM

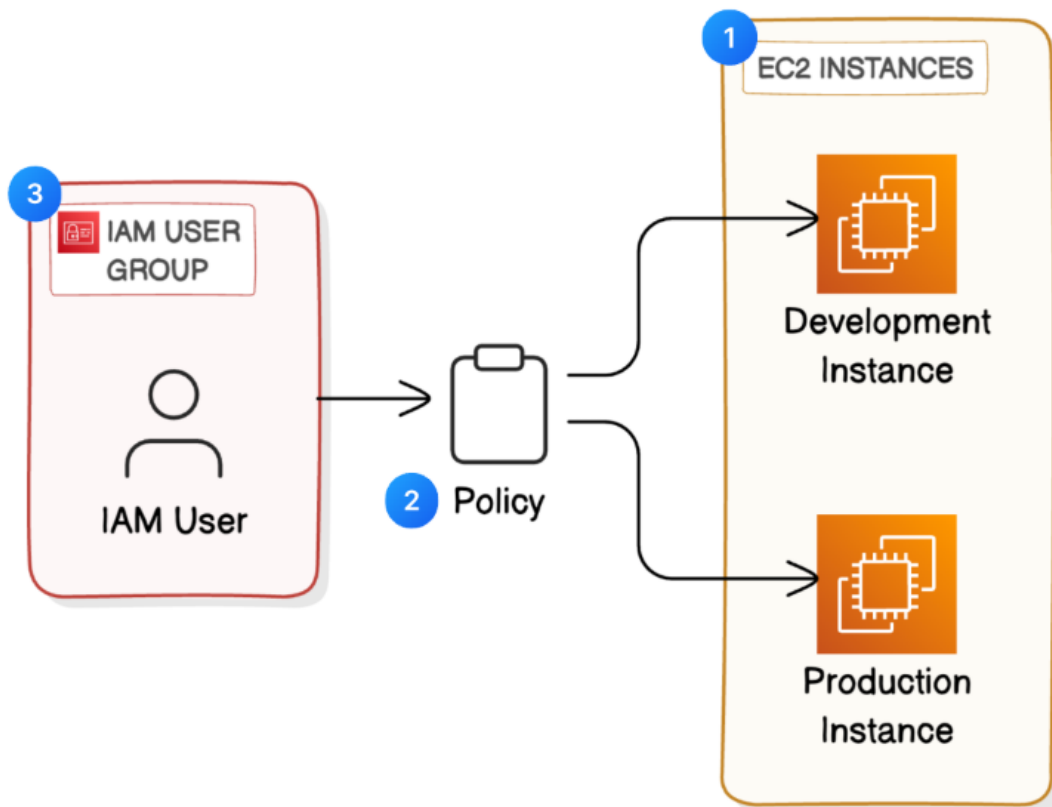
## 30 second Summary

*In AWS, a user is a person or a computer that can do things on the cloud - just like you right now!*

*Today, in this Project I will be using the AWS Identity and Access Management (IAM) service to control who is authenticated (signed in) and authorized (has permissions) in your AWS console.*

*I will launch an EC2 instance, then control who has access to it by creating some IAM policies and user groups. It will look something like this...*

*In this project, I will demonstrate how to use AWS IAM to control access and permission settings in my AWS account. I am doing this project to learn about cloud security from the absolute foundations- every company thinks about access permissions, and there are even entire jobs called 'IAM Engineers' focused on the skills we're about to build today*



### Core concept of this project

1. **EC2 instances**
2. **IAM Policies**
3. **IAM Users and User Groups**
4. **AWS Account Alias**

# Launch EC2 Instances

In this step, we will launch two EC2 instances because we need to boost NextWork’s computing power we're expecting more users and traffic into our website over the summer break!

## Creation of EC2 Instance

▼ Network settings Info

Edit

Network Info  
vpc-00eabebc9961f3c47

Subnet Info  
No preference (Default subnet in any availability zone)

Auto-assign public IP Info  
Enable  
Additional charges apply when outside of free tier allowance

Firewall (security groups) Info  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.  

Create security group

Select existing security group

We'll create a new security group called 'launch-wizard-2' with the following rules:  

☒ Allow SSH traffic from  
Helps you connect to your instance  
Anywhere  
0.0.0.0/0

☐ Allow HTTPS traffic from the internet  
To set up an endpoint, for example when creating a web server

☐ Allow HTTP traffic from the internet  
To set up an endpoint, for example when creating a web server

☰

EC2 > Instances > Launch an instance

## Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

▼ Name and tags Info

Key Info	Value Info	Resource types Info	
<div>Q Name X</div>	<div>Q nextwork-prod-rony X</div>	<div>Select resource types ▼</div>	<div>Remove</div>
		<div>Instances X</div>	

Key Info	Value Info	Resource types Info	
<div>Q Env X</div>	<div>Q production X</div>	<div>Select resource types ▼</div>	<div>Remove</div>
		<div>Instances X</div>	

Add new tag

You can add up to 48 more tags.

### ▼ Instance type [Info](#) | [Get advice](#)

#### Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Linux base pricing: 0.0124 USD per Hour  
On-Demand Windows base pricing: 0.017 USD per Hour  
On-Demand RHEL base pricing: 0.0268 USD per Hour  
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour  
On-Demand SUSE base pricing: 0.0124 USD per Hour

Free tier eligible

☐ All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

### ▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

#### Key pair name - required

Proceed without a key pair (Not recommended)

Default value ▼

[Create new key pair](#)

### ▼ Network settings [Info](#)

[Edit](#)

#### Network [Info](#)

vpc-00eabebc9961f3c47

#### Subnet [Info](#)

No preference (Default subnet in any availability zone)

#### Auto-assign public IP [Info](#)

Enable

[Additional charges apply](#) when outside of [free tier allowance](#)

#### Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

We'll create a new security group called 'launch-wizard-2' with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

Anywhere

0.0.0.0/0

☐ Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

☐ Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

## Created EC2 Instance

[EC2](#) > Instances

[Instances](#) [Refresh](#) [Help](#)

Instances (1) [Info](#)

Last updated  
less than a minute ago

[Connect](#)

[Instance state](#) ▼

[Actions](#) ▼

[Launch instances](#) ▼

All states ▼

< 1 > [Settings](#)

<input type="checkbox"/>	Name <a href="#">↗</a> ▼	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status	Availability Zone ▼	Public IPv4 DNS ▼	Public IPv4 ...
<input type="checkbox"/>	network-pro...	i-0c005bd2cae1ca39f	Running <a href="#">↗</a> <a href="#">↗</a>	t2.micro	Initializing	<a href="#">View alarms +</a>	ap-south-1b	ec2-65-0-124-241.ap-s...	65.0.124.241

Now let's create one more EC2 instance for the **development environment**. To create the development environment the above same procedure

Steps to create EC2 instance:

- Network settings – create security group (default)
- Name and tags – Choose **Add additional tags**, which is right next to your **Name** field
- Add new tag – key: Env, Value: production
- Head on down to see your EC2 settings and make sure the **Amazon Machine Image (AMI)** is using a **Free tier eligible** option.
- For the instance type, also make sure you're using a **Free tier eligible** option!
- For **Key pair (login)**, select **Proceed without a key pair**.
- Click **Launch instance**.

## Second instance

### Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags** [Info](#)

Key [Info](#)

Q Name X

Value [Info](#)

Q nextwork-dev-rony X

Resource types [Info](#)

Select resource types

Instances X

Remove

Key [Info](#)

Q Env X

Value [Info](#)

Q development X

Use: development

Resource types [Info](#)

Select resource types

Instances X

Remove

Add new tag

You can add up to 48 more tags.

### What are tags? What are they useful for?

Tags are organisational tools that lets us label our resources. They are helpful for grouping resources, cost allocation and applying policies for all resources with the same tag

### What are the tags and values you've assigned to your two EC2 instances?

The tag I've used on my EC2 instances is called Env, which stands for environment. The value we've assigned for our instances are production and development.

## The 2 created instance

Instances (2) Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

< 1 >

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
<input type="checkbox"/>	nextwork-pro...	i-0c005bd2cae1ca39f	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-65-0-124-241.ap-s...	65.0.124.241
<input type="checkbox"/>	nextwork-dev-...	i-06d100c21c5ca950f	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-52-66-241-156.ap-...	52.66.241.156

## Creation of IAM Policy

My intern should have permission to the development EC2 instance but not the production instance. I don't want them to accidentally shut down the platform or push their changes to the production environment while they're just testing things!

To start this task, I will use AWS IAM to give my intern access to the development instance first.

### What is IAM?

IAM stands for Identity and Access Management. You'll use AWS IAM to manage the access level that other users and services have to your resources.

### What do IAM Policies do?

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

## 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  ]
}
```

### Extra for Experts: how are JSON policies structured?

#### **Version**

This means 2012-10-17 is the date of the latest policy version. This tells you whether the policy is up to date with the latest standards and practices.

#### **Statement**

The main part of the policy structure and defines a list of permissions.

#### **Effect**

This can have two values - either **Allow** or **Deny** - to indicate whether the policy allows or denies a certain action. **Deny** has priority. Looking at the first statement, "Effect": "Allow" means this statement is trying to allow for an action.

#### **Action**

A list of the actions that the policy allows or denies. In this case, "Action": "ec2:\*" means all actions that you could possibly take on EC2 instances are allowed. Woohoo!

#### **Resource**

Which resources does this policy apply to? Specifying "\*" means all resources within the defined scope (see the next point).

#### **Condition Block** (optional)

The circumstances under which the policy is in action. In this case, the condition is that the resource is tagged **Env - development**. This means specifying "Resource": "\*" in the line above means all resources with the **Env - development** tag is impacted by your statement.

## Creation of policies

### Policy details

#### Policy name

Enter a meaningful name to identify this policy.

NextWorkDevEnvironmentPolicy

Maximum 128 characters. Use alphanumeric and '+=, @-\_' characters.

#### Description - optional

Add a short explanation for this policy.

IAM Policy for NextWork's development environment

Maximum 1,000 characters. Use alphanumeric and '+=, @-\_' characters.

## Create an AWS Account Alias

In this step, I will set up an Account Alias, which is like a nickname for i AWS account's console login. This is because an account alias makes it simpler for our users to login!

### Create alias for AWS account 863518452934

Preferred alias

nextwork-alias-rony

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

New sign-in URL

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

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

[Cancel](#) [Create alias](#)

### What does an Account Alias mean?

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

## Create IAM Users and User Groups

[IAM](#) > [User groups](#) > Create user group

Identity and Access Management (IAM)

Dashboard

Access management

User groups

Users

Roles

Policies

Identity providers

Account settings

Root access management

Access reports

Access Analyzer

Archive rules

Analizers

Add users to the group - Optional (1) Info

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

<input type="checkbox"/>	User name	Groups	Last activity	Creation time
<input type="checkbox"/>	Rony	1	1 hour ago	4 months ago

Attach permissions policies - Optional (1/1042) Info

You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.

Filter by Type

All types

1 match

<input checked="" type="checkbox"/>	Policy name	Type	Use...	Description
<input checked="" type="checkbox"/>	NextWorkDevEnviron...	Customer man...	None	IAM Policy for NextWorks development environment

[Cancel](#) [Create user group](#)

### What are IAM user groups?

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

What is the effect of attaching policy to user group?

I attached the policy we created to this user group, which means. any user created inside this group will 874 automatically get the permissions attached to our NextWork Dev Environment Policy IAM policy.

Step 1

Specify user details

Step 2

Set permissions

Step 3

Review and create

Step 4

Retrieve password

Specify user details

User details

User name

nextwork-dev-rony

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ \_ - (hyphen)

☒ Provide user access to the AWS Management Console - optional

If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Are you providing console access to a person?

User type

☐ Specify a user in Identity Center - Recommended

We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

☒ I want to create an IAM user

We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Console password

☒ Autogenerated password

You can view the password after you create the user.

Step 2

Set permissions

Step 3

Review and create

Step 4

Retrieve password

Permissions options

☒ Add user to group

Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions

Copy all group memberships, attached managed policies, and inline policies from an existing user.

☐ Attach policies directly

Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

User groups (1/2)

Search

< 1 >

Group name

Users

Atta...

Created

☐

admin

1

Admini...

2024-11-29 (4 months ago)

☒

nextwork-dev-group

0

NextW...

2025-04-17 (9 minutes ago)

Set permissions boundary - optional

Cancel

Previous

Next

review your choices. After you create the user, you can view and download the autogenerated password, if enabled.

User details

User name

nextwork-dev-rony

Console password type

Autogenerated

Require password reset

No

Permissions summary

Name

Type

Used as

nextwork-dev-group

Group

Permissions group

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Previous

Create user



- Step 1  
Specify user details
- Step 2  
Set permissions
- Step 3  
Review and create
- Step 4  
Retrieve password

## 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

[Email sign-in instructions](#)

#### Console sign-in URL

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

#### User name

[nextwork-dev-rony](#)

#### Console password

[\\*\\*\\*\\*\\* Show](#)[Cancel](#)[Download .csv file](#)[Return to users list](#)

## What are IAM users?

IAM users are people or entities that have access/can login to our AWS account.

## What are the two ways i could share a new User's sign-in details?

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

## What did i observe in new IAM user's AWS dashboard?

Once i logged in as our IAM user, i notice that my user is already denied access to panels on the main AWS console dashboard. This was because my only set up permissions to our development EC2 instance, so my intern wouldn't have access to even see anything else

## When I tried to stop the production instance

**Instances (2/2)** Info

Last updated less than a minute ago

[Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Find Instance by attribute or tag (case-sensitive) All states

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm s
<input checked="" type="checkbox"/>	nextwork-pro...	i-0c005bd2cae1ca39f	Running	t2.micro	2/2 checks passec	User
<input checked="" type="checkbox"/>	nextwork-dev...	i-06d100c21c5ca950f	Running	t2.micro	2/2 checks passec	User

2 instances selected

**Instance state settings**

☐ Start  
Available when the instance is stopped

☒ Stop

☐ Hibernate  
This instance did not have Stop - Hibernate enabled at launch

☐ Reboot

☐ Terminate

**Note that when your instances are stopped:**  
Any data on the ephemeral storage of your instances will be lost.


[Cancel](#) [Change state](#)


## Stop instances

Stopping your instance allows you to reduce costs, modify settings, and troubleshoot problems.

Instance ID

Stop protection

 [i-0c005bd2cae1ca39f \(nextwork-prod-rony\)](#) 

 Off (Can stop instance)

 [i-06d100c21c5ca950f \(nextwork-dev-rony\)](#) 

 Off (Can stop instance)



### You will be billed for associated resources

After you stop the instance, you are no longer charged usage or data transfer fees for it. However, you will still be billed for associated Elastic IP addresses and EBS volumes.

### Associated resources

You will continue to incur charges for these resources while the instance is stopped

Cancel

Stop

ⓘ Failed to stop the instances: You are not authorized to perform this operation. User: arn:aws:iam::863518452934:user/nextwork-dev-rony is not authorized to perform: ec2:StopInstances on resource: arn:aws:ec2:ap-south-1:863518452934:instance/i-0c005bd2cae1ca39f because no identity-based policy allows the ec2:StopInstances action. Encoded authorization failure message: C0dkAnmWF60EAa4kZwcKGTnXCOiXSmWul5oBbFIDbEVg1rN\_P5UI-gmNgJg\_-IDrCspeu0haTxlvsf9lg6lCS0fs7eG6RoCm-QtcwuhE3gWeiT6EkyL9HzZ-QRQ2y7Q0RLxLdhGf\_bxQydkjIw2hy8JRZJ76K1HZHCwidFZ8mqxLtkAeqGF0X\_UBbJUwNwBrBQhcJKZ6huQz\_M66-iyGoZOo59XoT\_pQ35OWpxlVMOgC3VjethXnnipisKFcY9UmT\_mAuT7hpgnpLio9igQ8l8jm9\_g9WH-\_rV8C2KXwz8oFtpur2FNJD0xKd5Fw1wTp6yXj\_XsZw1kdnT5ClkA7bNGw8Wc0akArBiYEdPd\_svXkkHcWm-z0BJSJeFFhYq5p-ehBWajRh992ex98A4VQ\_ekbnmyWJ7DEDzo-UJLEdZU4y6DPF3PKUc5ldn9OOH5oIM45EmhyL4ErkZcxSKZAR0atLRjcZz69X\_4l8cnlNhqthEM0hywY50aNBbkIFtc\_EaFilzXeSmdLylhjtiv9x6Q40WdvM9vLotJaNVVnKOLsLOt8BLqOyfq19ApeSHHUrQ5brh6jJrG6Pac9aZbArDp03ZFEgqbHZsyBMLgk3t4Rp9DMnp7CciwmLy9CkMrtrgMucsr1yZk8KLf9tqjZmHHL-XZ5-40P5zsTXhdrrwJaz6aUP31wApeoQeRZafefl4H2nllygJE6uv85doHOyEBgFa6SeFahRens9uScd4ACmbU8gl7a3loHSd8SNvKOb5Y1gvzpWmtLxHHbdqso02ZxrEBUsAUG6P1GjF9bC3acLTBaLTYbS0ksf-UmoMfxB4oonQyA2RFGGiWYZTOgxtgKzLw5vtp22PZ5Xz9GrmslSvfdnYQe51J4Afkg

▶ Failed to stop the following instances :

## Manage instance state

### Instance details

[i-0c005bd2cae1ca39f \(nextwork-prod-rony\)](#)

running

[i-06d100c21c5ca950f \(nextwork-dev-rony\)](#)

running

## What happened when I tried to stop the production instance?

When i tried to stop the production instance, i were met with an error! This was because our production instance is tagged with the 'production' label, which is outside of the scope of my permission policy - interns are only allowed to do things to development instances.

Instances (1/2) Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

< 1 >

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	nextwork-pro...	i-0c005bd2cae1ca39f	Running	t2.micro	2/2 checks passec	User: arn:aws:	ap-south-1b	ec2-65-0-
<input checked="" type="checkbox"/>	nextwork-dev...	i-06d100c21c5ca950f	Stopping	t2.micro	2/2 checks passec	User: arn:aws:	ap-south-1b	ec2-52-6f

## What was the action i performed on two EC2 instances?

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

## What happened when i tried to stop the development instance?

when i tried to stop the development instance, i successfully saw the instance state change to Stopping and then Stopped. This was because my permission policy allows intern. users next work.

# IAM Policy Simulator

In the last step, you ended up shutting down the development instance to test your intern's access. Shutting down EC2 instances could get pretty disruptive for your other engineers and your users, so it's best practice to run these tests in another way.

Policy Simulator

Amazon EC2

2 Action(s) sele...

Select All

Deselect All

Reset Contexts

Clear Results

Run Simulation

▶ Global Settings ⓘ

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

Service	Action	Resource Type	Simulation Resource	Permission
▶ Amazon EC2	StopInstances	instance	*	<b>allowed</b> 1 matching statements.
▶ Amazon EC2	DeleteTags	not required	*	<b>denied</b> 1 matching statements.

## Why would i use the IAM Policy Simulator?

The IAM Policy Simulator is a tool that lets us simulate actions and test permission settings by defining a specific user/group/role and the action i want to test for. It's useful for saving time when testing permission settings! No more logging into another user or actually stopping resources

## What were the simulation results for the development instance?

I set up a simulation for whether my dev group has permission to Stop instances or Delete Tags. The results were denied for both we had to adjust the scope of the EC2 Instances to ones that are tagged with "development". Once I applied that tag, permission was allowed.

## What were the key services and concepts i learnt in this project?

Services I used today were Amazon EC2 and AWS IAM!

Key concepts I learnt include IAM users, policies, user groups and account aliases.

I also learn how to use the Policy Simulator and how JSON policies work. How to launch an instance, how to tag an instance, how to log in as another user.

## How long did I take to complete this project?

This project took me approximately 1.5eurs today including project demo time! The most challenging part was understanding the IAM policy since it was written in JSON and it contained multiple statements. It was most rewarding to see permission denied when our intern tried to delete our production instance my IAM access management worked!

Thanks

**Rony Joseph Thomas**