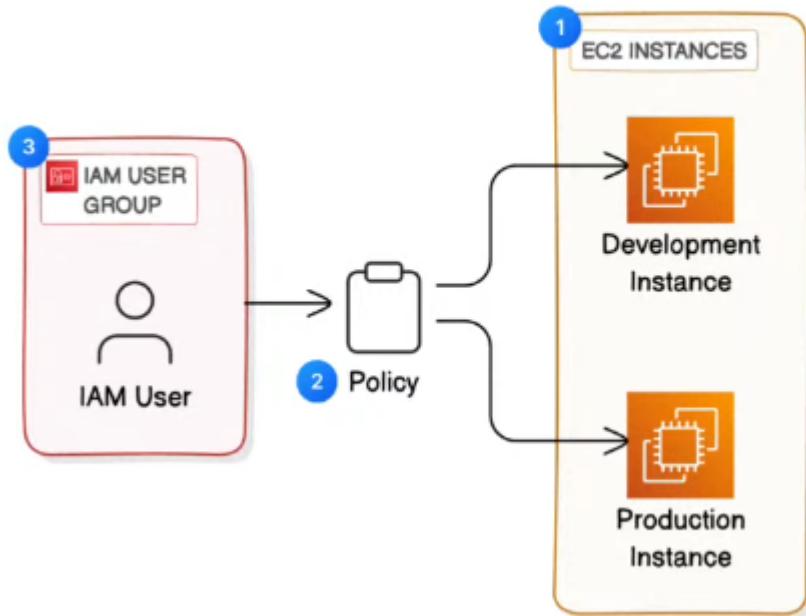
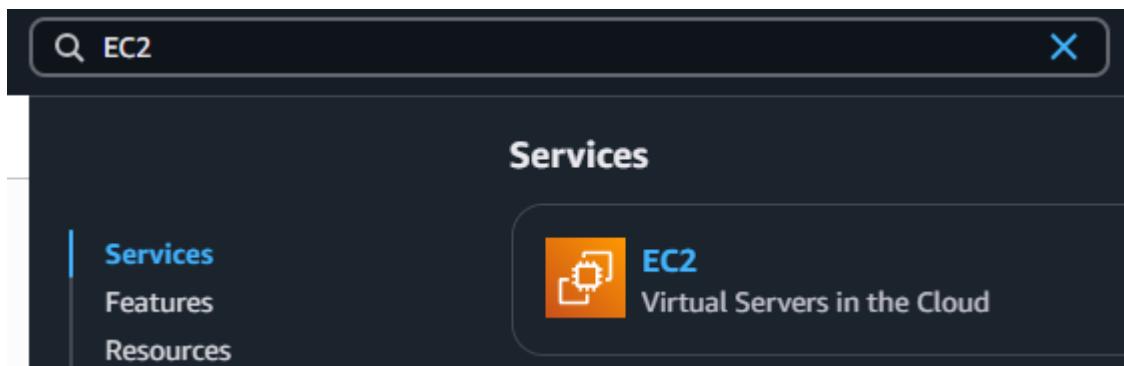


We'll launch an EC2 instance, then control who has access to it by creating some IAM policies and user groups. It will look something like this :



### Step #1: LAUNCH INTANCE EC2

- Log in to your AWS Management Console.
- Open your EC2 console - search for it at the search bar.



Let's set up your EC2 instance!

In Name, enter the value.

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

Name

 [Add additional tags](#)

Click on “Add additional tags”

For the next tag, use this information:

- Key: Env
- Value: production

The tags are like labels you can attach to AWS resources for organization that helps you to organise your resources or make search easily..

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

Key   Info	Value   Info	Resource types   Info
<input type="text" value="Name"/> <a href="#">X</a>	<input type="text" value="network-steeve"/> <a href="#">X</a>	<input type="button" value="Select resource types"/> <a href="#">▼</a> <a href="#">Remove</a>
		<input type="button" value="Instances"/> <a href="#">X</a>
Key   Info	Value   Info	Resource types   Info
<input type="text" value="Env"/> <a href="#">X</a>	<input type="text" value="production"/> <a href="#">X</a>	<input type="button" value="Select resource types"/> <a href="#">▼</a> <a href="#">Remove</a>
		<input type="button" value="Instances"/> <a href="#">X</a>

[Add new tag](#)

Head down to see your EC2 instance settings, and make sure the Amazon Machine Image using is “**Free tier eligible**”

**Quick Start**

Amazon Linux	Ubuntu	Windows	Red Hat	SUSE Linux	Debian

**Amazon Machine Image (AMI)**

Amazon Linux 2023 kernel-6.1 AMI ami-078abd88811000d7e (64-bit (x86), uefi-preferred) / ami-0387f15c965e9e817 (64-bit (Arm), uefi) Virtualization: hvm ENA enabled: true Root device type: ebs	<a href="#">Free tier eligible</a> <a href="#">▼</a>
--	--

[Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

For the **instance type** make sure you are using **free tier eligible**

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

### Instance type

t3.micro	Free tier eligible
Family: t3 2 vCPU 1 GiB Memory Current generation: true	
On-Demand Ubuntu Pro base pricing: 0.0153 USD per Hour	
On-Demand SUSE base pricing: 0.0118 USD per Hour On-Demand Linux base pricing: 0.0118 USD per Hour	
On-Demand Windows base pricing: 0.021 USD per Hour On-Demand RHEL base pricing: 0.0406 USD per Hour	

All generations

[Compare instance types](#)

**Additional costs apply for AMIs with pre-installed software**

- For **Key pair, choose processing without a key pair.** key pair is used to increase security access to your instance by allowing access just by ssh connection.

## ▼ 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

Default value ▾

you're ready! click on launch instance

[Cancel](#)

[Launch instance](#)

 [Preview code](#)

you get that message:



**Success**

Successfully initiated launch of instance ([i-0838d5bfeb54b1648](#))

## ► Launch log

- Now you're going to create one more EC2 instance for development service

## Create an IAM Policy

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/> network-devlop	i-09352e3dc9acd1a10	Running		t3.micro	3/3 checks passed <a href="#">View alarms +</a>
<input checked="" type="checkbox"/> network-steve	i-0838d5bfeb54b1648	Running		t3.micro	3/3 checks passed <a href="#">View alarms +</a>

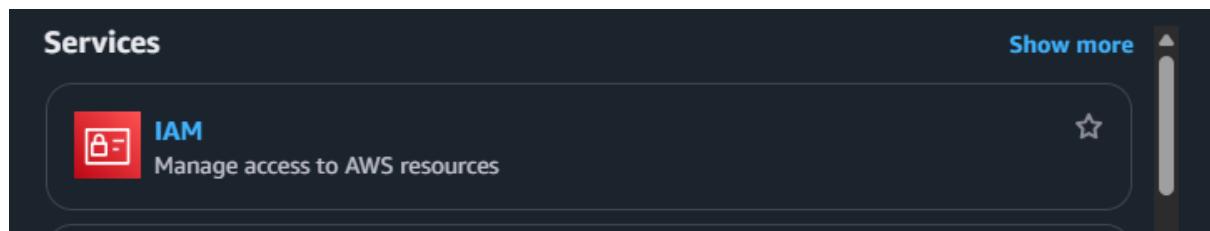
Woooo! you've deployed an EC2 instance for your production environment and development.

## Step #2: CREATE IAM POLICY

Let's move into our second task as NextWork's engineer, it's time to onboard the team's new intern and set up permission policies.

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

Go in the search console, type IAM



- Now on the left-hand navigation panel of your IAM console, choose **Policies**.
- Create policy**
- Switch on **Json**

### Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

The screenshot shows the Policy editor interface in JSON mode. The main area displays a JSON code editor with the following content:

```
1 Version: "2012-10-17",
2 Statement: [
3   {
4     Sid: "Statement1",
5     Effect: "Allow",
6     Action: [],
7     Resource: []
8   }
9 ]
10
11 }
```

To the right of the editor, there are three panels: "Edit statement", "Add actions", and "Available". The "Edit statement" panel shows "Statement1". The "Add actions" panel has a "Choose a service" dropdown with "Search serv". The "Available" panel is currently empty.

Replace de the content by this :

Select next

Fill in your policy's details:

- Name: NextWorkDevEnvironmentPolicy
- Description: IAM Policy for NextWork's development environment

## 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 Netwok development environment

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

ended click on **Create Policy**

✓ Policy NextWorkDevEnvironmentPolicy created.

## Step #3: CREATE AN AWS ACCOUNT ALIAS

In this step, get ready to simplify user login to your AWS Account using an Account Alias.

In IAM click on dashboard in left-hand side, then choose **create alias** under Account Alias



## Step #4 CREATE IAM USERS AND USERS GROUP

In this step, get ready to:

- Step up a dedicated IAM group for all NextWork interns, so you can manage all intern's permission from one place.
- Step up a dedicated IAM user for your new intern, so they have a way to log in.

So Here we go!

- choose **users group** in your left-hand navigation panel
- **Create Group**

### Name the group

#### User group name

Enter a meaningful name to identify this group.

Network-dev-group

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

### Attach permissions policies - *Optional* (1/1106) [Info](#)

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

NextWorkDev X

Policy name

Type

NextWorkDevEnvironmentPolicy

Customer managed

You can show confirmation message

 Network-dev-group user group created.

## User groups (1) Info

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

 Search

Group name

 Users

Network-dev-group

After the user group has been created you need to create a user, then add them inside the user group.

- In left-hand of panel navigation IAM choose **user**, then **Create User**

### User details

#### User name

network-dev-st

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*

In addition to console access, users with SignInLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.

#### Console password

Autogenerated password

You can view the password after you create the user.

Custom password

Enter a custom password for the user.

Show password

Users must create a new password at next sign-in - Recommended

Users automatically get the IAMUserChangePassword policy to allow them to change their own password.

 If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces [Learn more](#)

Click on **Next**, To set permissions for your user, we'll simply add it to the user group you've created. Select the checkbox **next to network-dev-group**. - Select Next. - Select **Create user!**

## 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.

Cop  
Copy  
polic

## User groups (1/1)

 Search

| Group name ↗

▲ | Users

Network-dev-group

0

## ► Set permissions boundary - *optional*

 User created successfully

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

[View user](#)

## TEST YOUR INTERN'S ACCESS

In this step we will log in to our own AWS account as the intern and test: access to production and development, because we want to make sure our intern's doesn't have availability to do anything that affects our production environment.

- So copy your **console sign-in** URL and paste it in the new window of your navigator

**IAM user sign in** ⓘ

Account ID or alias ([Don't have?](#))  
[REDACTED]

Remember this account

IAM username  
network-dev-st

Password  
[REDACTED]  
 Show Password [Having trouble?](#)

**Sign in**

[Sign in using root user email](#)

[Create a new AWS account](#)

After login you can see that the user gets just permission to something in the development instances and anything else.

## Applications (0) Informations

[Créer une application](#)

Région : Europe (Stockholm)

Sélectionner la région

eu-north-1 (Région actuelle) ▾

Rechercher des applications

< 1 >

Nom | Description | Région | Compte d

✖ ► Accès refusé à servicecatalog>ListApplications

**Example :** When we try to stop instance named **network-steeve**, we get message error because we don't get permission to perform that action

NAME	ID	STATUS	TYPE
network-steeve	i-0838d5bfeb54b1648	En cours d'... 🔍 ⏪	t3.micro

✖ Échec de Stop (Arrêter) de l'instance i-0838d5bfeb54b1648

You are not authorized to perform this operation. User: arn:aws:iam::302502112524:user/network-dev-st is not authorized to perform because no identity-based policy allows the ec2:StopInstances action. Encoded authorization failure message: WqQsLZT4Ms3xP3fhIfyblKdkqRNK1zlyPvRdwX-DTTX2M\_bQmzQTtyJSNoXP0KU0ymcuGjt7NStBrxQkcnww5\_pIXI-gN5DqPWKfL5u-ISIJjFP-XpW6PZt6ZmE9YcfCPPj6qlKccGx2lgkYMPKGNAaHYboSXrEF7SSM1kite5yueCZVQA63eR9nF2nCFqtamsZJgLGESsoeRGFmfrobr5qZmr-BtmofaSDvZH8PpjPyH9MOOLBIDMIhVHLjOEa5ZPNmiaBM7gx\_AiSnK6p4Kwb5E3b55M3SIKisG\_WMF7FtqVU9ndllygAWdpOKtEPBjHe7aNRMVEpiHoCyuUxh9QSLVd4gTdaf2Nv6AhjosQPOi8EDhNrCBMET0flzbJrUKWq6UQab3d8OkruUuvRPWHEGzBdK7vy2bLk4zRDd0s7wUVpwcoqvxViZbY7kVQCXYmgwk246iS5xjp\_rW--ilrEF3oviN19wyAbUTXO2HC5KXWJmfU9BRTmPqAlZMRwu6RdLIXRaT\_ZcjCDsW3p\_0nCga11NeMJO\_ukSeFTS\_8fzxy\_Dr1hELlyKSVnEVHmR4rbMitCiw8tM2mfgyBeXMF0mx7ZOFbBA0QAtXxoT4uaH3p8M0O4gtfHG5Mph9UCUS-N3G1fA

That's end of our project.