

AWS Identity and Access Management

Author: Oluwaseun Osunsola

Environment: AWS

Project Link: <https://github.com/Oluwaseunoa/DevOps-Projects/tree/main/Cloud%20Computing/AWS%20Identity%20and%20Access%20Management>

Overview

We're going to learn about AWS Identity and Access Management (IAM) which helps control who can access what in Amazon Web Services. We'll cover things like users, roles, policies, and groups, and we'll also show you how to actually set them up to keep your AWS resources safe.

But before we get into all that, let's make sure you understand the basics of cloud computing. If terms like "Cloud" sound new to you, it's a good idea to go back and review some earlier materials to get a solid grasp of what it's all about.

Project Goals

- Understand AWS Identity and Access Management (IAM) principles and components.
- Learn to create and manage IAM policies for regulating access to AWS resources securely.
- Apply IAM concepts practically to control access within AWS environments.
- Explore best practices for IAM implementation and security in AWS.

Learning Outcomes

- Recognize IAM components like users, roles, policies, and groups.
- Create and manage IAM policies to define permissions for users and roles.
- Set up IAM users, groups, and roles to control access to AWS services.
- Understand IAM best practices for maintaining security and managing access to AWS resources.

Key Concepts

What is IAM?

IAM, or Identity and Access Management, is the gatekeeper for your AWS resources, deciding who gets in and what they're allowed to do once inside.

Imagine you have a big digital "house" full of all your AWS stuff—your data, your applications, the whole shebang. You don't want just anyone wandering in and messing around with your things, right? That's where IAM steps in.

It's like having your own VIP list for your digital world. IAM helps keep your AWS resources safe, ensuring only the right people get in and can only do what you allow. It's all about keeping your digital house in order and protecting your precious stuff from unwanted guests.

Note: AWS resources are the various services and tools provided by Amazon Web Services (AWS) that users can utilize to build and manage their applications and infrastructure in the cloud.

What is IAM User?

IAM users are individual accounts for different people or entities within your AWS environment.

For example, if you have a team working on a project, you can create separate IAM users for each team member. Each IAM user has a unique username and password, allowing them to access the AWS resources needed for their work.

IAM users help manage and control access to your AWS resources securely, ensuring each user only has access to the resources required for their tasks.

What is IAM Role?

An IAM role defines what someone or something (like an application or service) can do within your AWS account. Each role has a set of permissions that determine which actions it can perform and which AWS resources it can access.

For example, you might have an "admin" role with full access to all resources or a "developer" role that only allows access to certain services for building applications.

Imagine a visitor needing temporary access to your house to fix something. Instead of giving them a permanent key (IAM user), you give them a temporary key (IAM role) that works for a limited time and grants access to specific rooms (AWS resources).

IAM roles are flexible and can be assumed by users, services, or applications as needed. They are used for tasks like granting permissions to AWS services, allowing cross-account access, or providing temporary access to external users. IAM roles enhance security and efficiency by providing controlled access without permanent credentials.

What is IAM Policy?

An IAM policy is a set of rules defining what actions a role or user can take. Think of it as a rulebook outlining which actions are allowed and which are not, ensuring secure and controlled access to your AWS resources.

For example, the rulebook might say the "admin" key (IAM role or user) can open any door and perform any action within the house (AWS resources), while the "viewer" key can only open certain doors and look around without making changes.

IAM policies define permissions for IAM roles and users, specifying which AWS resources they can access and what actions they can take. They are essential for maintaining security and ensuring only authorized actions are performed.

What is IAM Group?

IAM Groups are collections of IAM users. Instead of managing permissions for each user individually, you can organize users into groups based on roles or responsibilities.

Think of IAM Groups as teams based on tasks. You might have a group for developers, another for administrators, and so on. Instead of setting permissions for each person one by one, you set them for the whole group at once.

For example, for a development team, you can create an IAM Group called "Developers" and add all developers to it. Assign permissions to the group, and all developers get the same resource access, simplifying management.

Best Practices

- **Give only the permissions needed:** Avoid granting more access than necessary.
- **Use roles instead of users:** Roles are safer and can be used when needed.
- **Review roles regularly:** Remove unused roles to keep things tidy and secure.
- **Add extra security with MFA:** Use Multi-Factor Authentication for extra protection.
- **Use ready-made policies:** They're safer and easier to use.
- **Keep policies simple:** Create separate policies for different tasks.
- **Keep track of changes:** Record who changes what.
- **Test policies before using them:** Ensure they work as intended before applying.
- **Use descriptive names:** Choose clear names for IAM groups to aid understanding and management.
- **Enforce strong password policies:** Encourage strong passwords with expiration and complexity requirements.

Note (difference between users and roles): Users are like individuals with permanent keys to access resources, tied to specific people. Roles are like special keys providing temporary access, usable by different users or programs as needed. Users are fixed to individuals, while roles are flexible for specific tasks.

For MFA, see Multi-Factor Authentication (MFA) for IAM.

Note on AWS policies:

- **Managed Policies:** Created by AWS, widely used.
- **Customer Managed Policies:** You create and manage them.
- **Inline Policies:** Made for one specific thing.

For further details, refer to [Policies and permissions in IAM](#) in the IAM documentation.

Practical Implementation

A growth marketing consultancy, GatoGrowFast.com, wants to grant access to AWS resources for employees Seun, Jack, and Ade. The project is divided into three parts:

1. Create a policy granting full EC2 access and assign it to user Seun, along with MFA setup.
2. Create a group, add users Jack and Ade, and create a policy granting full EC2 and S3 access for the group.
3. Enable MFA for user Seun to enhance security.

The following steps are ordered numerically based on the provided screenshot names, with each step linked to its corresponding screenshot. Note: Screenshots use "Seun" as the user name, consistent with the provided naming.

Part 1: Create EC2 Policy and IAM User (Seun)

1. Log In and Navigate to Your AWS Console

The screenshot shows the AWS Console Home page. On the left, under 'Recently visited', there are links to IAM, CodeBuild, Secrets Manager, S3, EC2, IAM Identity Center, and CodePipeline. Below this is a 'View all services' link. To the right, under 'Applications (0)', it says 'Region: US East (Ohio)' and has a 'Create application' button. It also includes a 'Select Region' dropdown set to 'us-east-2 (Current Region)' and a search bar for 'Find applications'. A message at the bottom says 'No applications' and 'Get started by creating an application.' with a 'Create application' button. At the bottom of the page, there are links for 'CloudShell', 'Feedback', and copyright information.

2. Use the Services Search to Search for IAM and Click to Visit the Page

The screenshot shows the AWS Console Home page with a search bar containing 'IAM'. The 'Services' section is highlighted, showing the IAM card with the text 'Manage access to AWS resources'. A red arrow points to this card. Below it are cards for IAM Identity Center and Resource Access Manager. The 'Features' section includes cards for IAM Access analyzer for S3, Groups, and Roles. At the bottom, there's a 'Were these results helpful?' section with 'Yes' and 'No' buttons, and a 'Getting started' link.

3. Click on Policies on the Menu Option

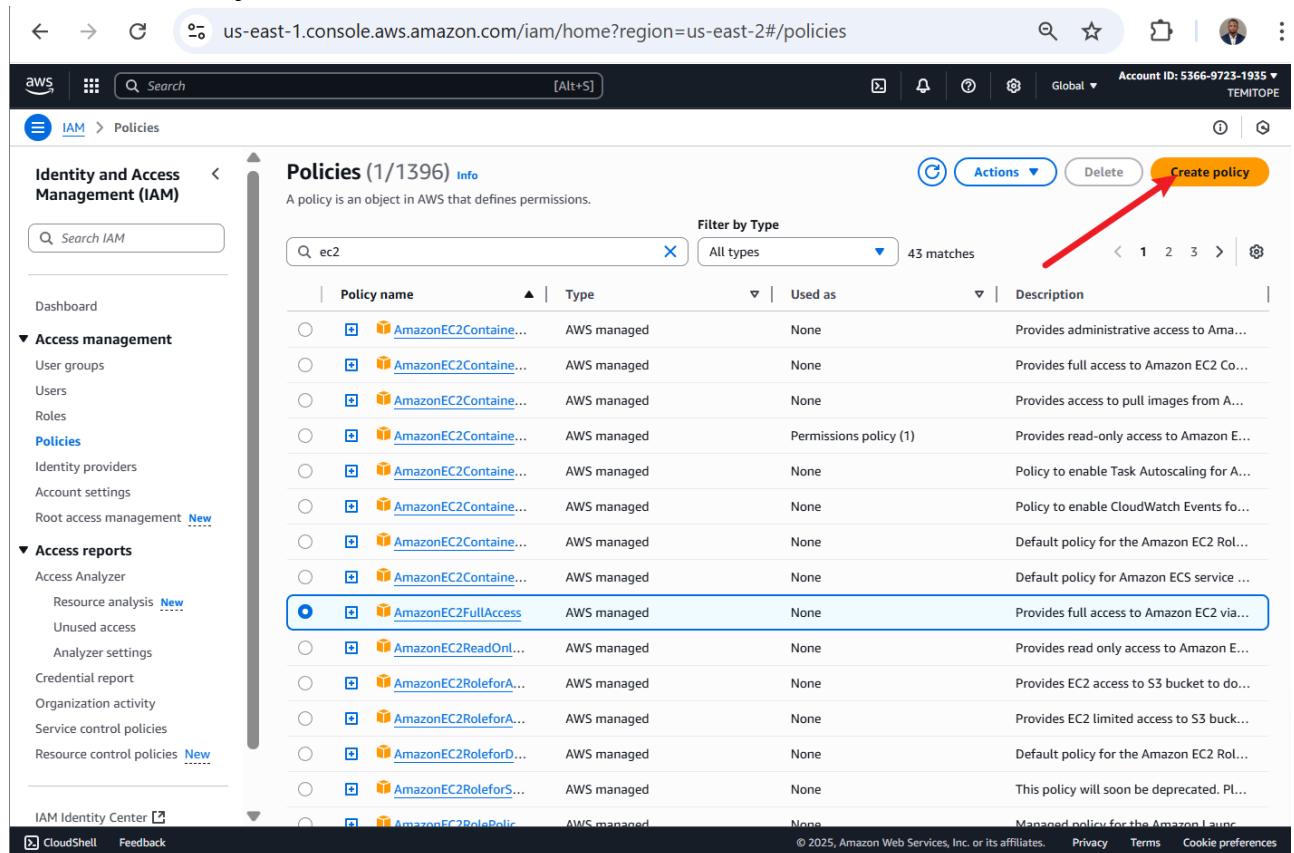
The screenshot shows the AWS IAM Dashboard. On the left sidebar, under 'Access management', the 'Policies' option is highlighted with a red arrow. The main content area displays 'IAM resources' with counts: User groups (2), Users (1), Roles (3), Policies (10), and Identity providers (0). Below this is a 'What's new' section listing recent updates. To the right are panels for 'AWS Account' (Account ID: 536697231935) and 'Quick Links'.

4. Search for EC2 and Grant by Selecting AmazonEC2FullAccess

The screenshot shows the 'Policies' page with a search bar containing 'ec2' highlighted with a red arrow. The results table lists various AWS managed policies, with 'AmazonEC2FullAccess' highlighted with a blue arrow. The table includes columns for Policy name, Type, Used as, and Description.

Policy name	Type	Used as	Description
AmazonEC2ContainerRegistryFullAccess	AWS managed	None	Provides administ
AmazonEC2ContainerRegistryPowerUser	AWS managed	None	Provides full acce
AmazonEC2ContainerRegistryPullOnly	AWS managed	None	Provides access to
AmazonEC2ContainerRegistryReadOnly	AWS managed	Permissions policy (1)	Provides read-onl
AmazonEC2ContainerServiceAutoscaleRole	AWS managed	None	Policy to enable t
AmazonEC2ContainerServiceEventsRole	AWS managed	None	Policy to enable C
AmazonEC2ContainerServiceforEC2Role	AWS managed	None	Default policy for
AmazonEC2ContainerServiceRole	AWS managed	None	Default policy for
AmazonEC2FullAccess	AWS managed	None	Provides full acce
AmazonEC2ReadOnlyAccess	AWS managed	None	Provides read onl
AmazonEC2RoleforAWSCodeDeploy	AWS managed	None	Provides EC2 acc
AmazonEC2RoleforAWSCodeDeployLimited	AWS managed	None	Provides EC2 limi
AmazonEC2RoleforDataPipelineRole	AWS managed	None	Default policy for
AmazonEC2RoleforSSM	AWS managed	None	This policy will so

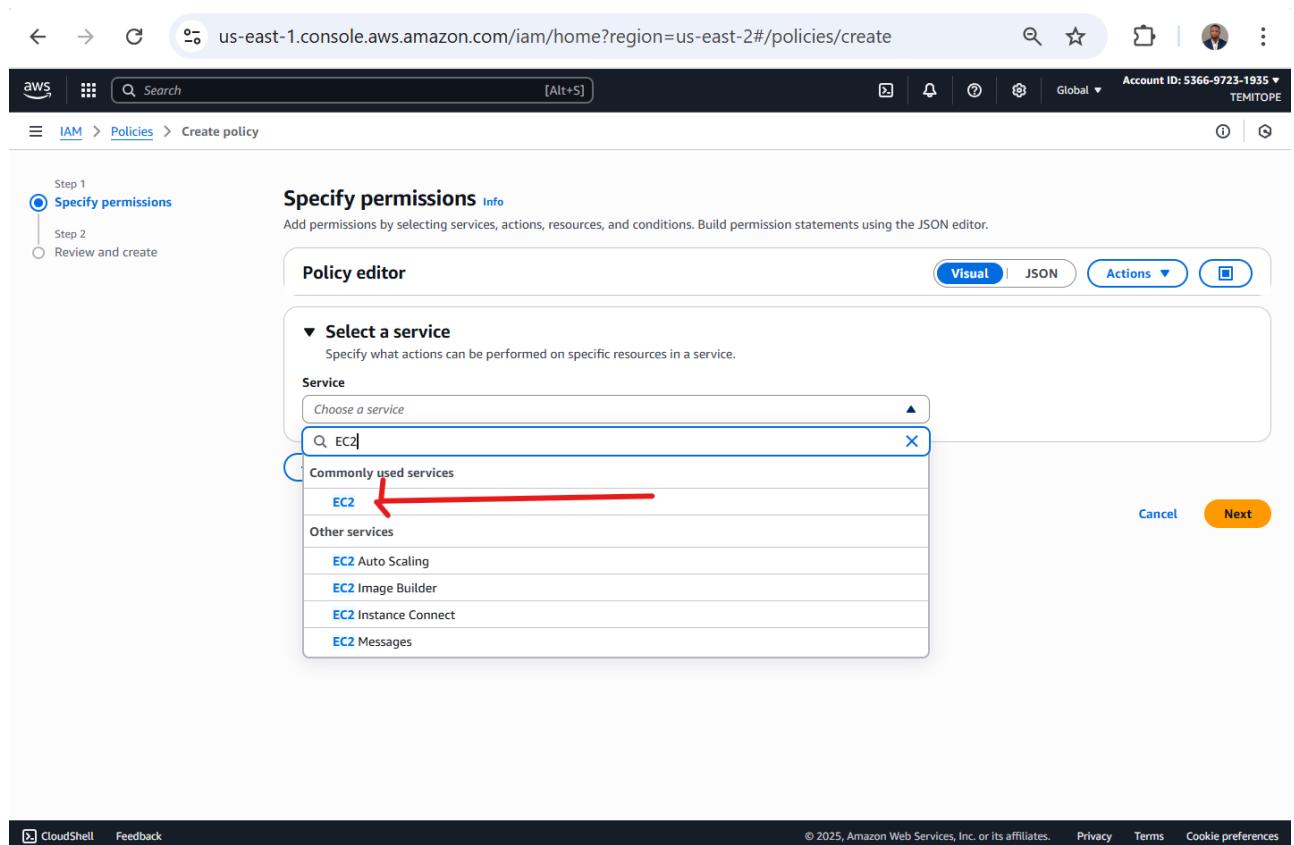
5. Click Create Policy



The screenshot shows the AWS IAM Policies page. A red arrow points to the 'Create policy' button at the top right of the table header. The table lists various AWS managed policies related to EC2.

Policy name	Type	Used as	Description
AmazonEC2Container...	AWS managed	None	Provides administrative access to Ama...
AmazonEC2Container...	AWS managed	None	Provides full access to Amazon EC2 Co...
AmazonEC2Container...	AWS managed	None	Provides access to pull images from A...
AmazonEC2Container...	AWS managed	Permissions policy (1)	Provides read-only access to Amazon E...
AmazonEC2Container...	AWS managed	None	Policy to enable Task AutoScaling for A...
AmazonEC2Container...	AWS managed	None	Policy to enable CloudWatch Events fo...
AmazonEC2Container...	AWS managed	None	Default policy for the Amazon EC2 Rol...
AmazonEC2Container...	AWS managed	None	Default policy for Amazon ECS service ...
AmazonEC2FullAccess	AWS managed	None	Provides full access to Amazon EC2 via...
AmazonEC2ReadOnl...	AWS managed	None	Provides read only access to Amazon E...
AmazonEC2RoleforA...	AWS managed	None	Provides EC2 access to S3 bucket to do...
AmazonEC2RoleforA...	AWS managed	None	Provides EC2 limited access to S3 buck...
AmazonEC2RoleforD...	AWS managed	None	Default policy for the Amazon EC2 Rol...
AmazonEC2RoleforS...	AWS managed	None	This policy will soon be deprecated. Pl...
AmazonEC2RolePolicy	AWS managed	None	Managed policy for the Amazon Launc...

6. Select EC2 Service



The screenshot shows the 'Specify permissions' step of the Create Policy wizard. A red arrow points to the 'EC2' option in the 'Select a service' dropdown menu.

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

Policy editor Visual | JSON | Actions ▾

Select a service
 Specify what actions can be performed on specific resources in a service.

Service
 Choose a service X

Commonly used services
 EC2 EC2 EC2 Auto Scaling EC2 Image Builder EC2 Instance Connect EC2 Messages

Other services

Cancel Next

7. Select All EC2 Actions Under Action and Scroll Down

The screenshot shows the AWS IAM 'Create policy' interface. The left sidebar indicates 'Step 1 Specify permissions' and 'Step 2 Review and create'. The main area is titled 'Specify permissions' with a 'Policy editor' tab selected. Under the 'EC2' service, the 'Allow' button is selected for 'All actions'. A red arrow points to the 'All EC2 actions (ec2:*)' checkbox. The 'Actions allowed' section lists several actions like 'List', 'Read', 'Write', and 'Permissions management'. A warning message states: 'Dependent permissions not selected. To grant permissions for the selected resource actions, including additional dependent actions might be required.' At the bottom, there are 'Effect' options ('Allow' is selected), 'Expand all | Collapse all' buttons, and a note about granting permissions for dependent actions.

8. Under Resources, Tick All and Click Next Button

This screenshot continues from the previous one, showing the 'Specify permissions' step. A red arrow points to the 'All' radio button under the 'Resources' section, which is highlighted with a yellow border. Another red arrow points to the 'Next' button at the bottom right of the page. The bottom of the screen shows standard navigation links like CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

9. Add Policy Names and Create Policy

Step 1
Specify permissions

Step 2
Review and create

Review and create Info

Review the permissions, specify details, and tags.

Policy details

Policy name
Enter a meaningful name to identify this policy.

Description - optional
Add a short explanation for this policy.

Permissions defined in this policy Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

Allow (1 of 450 services)

Service	Access level	Resource	Request condition
EC2	Full access	All resources	None

Add tags - optional Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Create policy

10. Policy Successfully Created

Identity and Access Management (IAM)

Policies (1/1397) Info

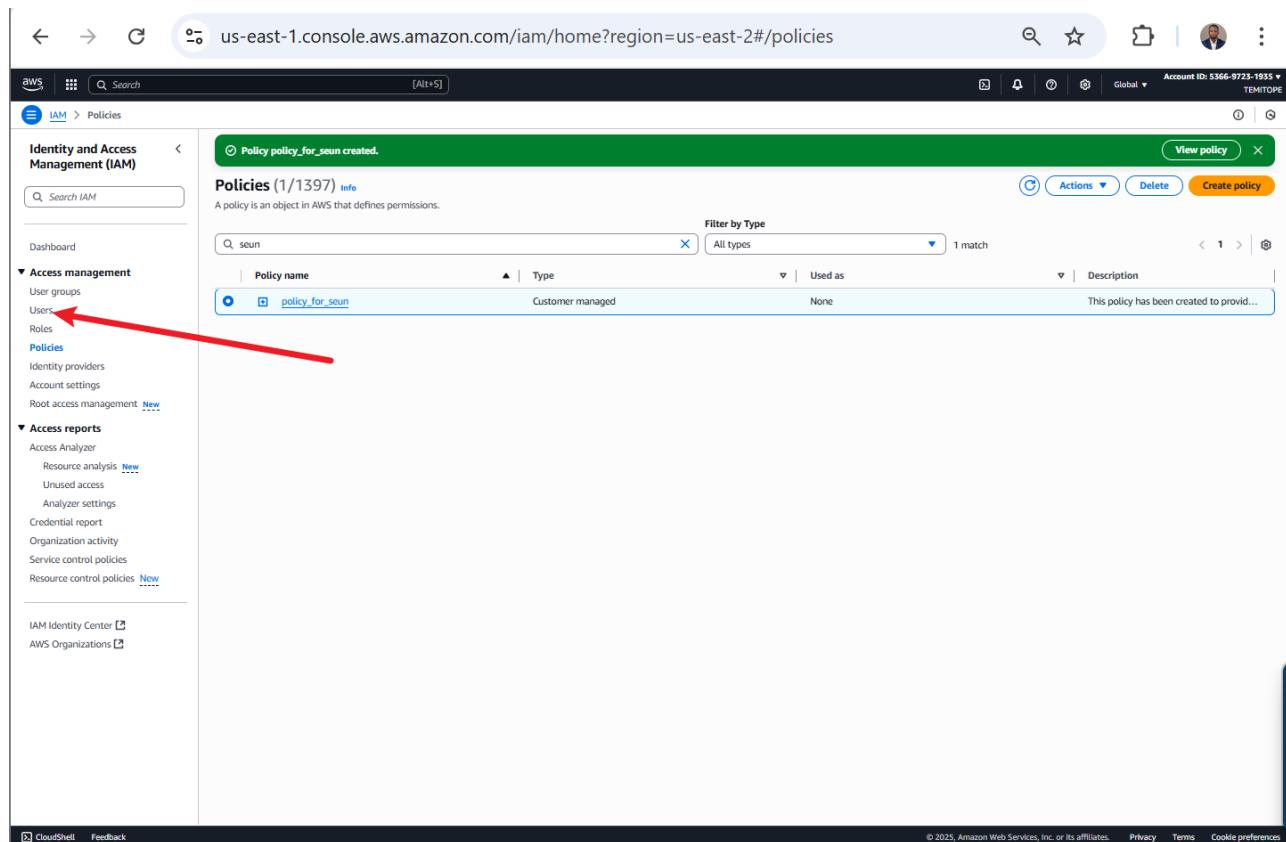
A policy is an object in AWS that defines permissions.

Filter by Type

Policy name	Type	Used as	Description
<input type="text" value="policy_for_seun"/>	Customer managed	None	This policy has been created to provid...

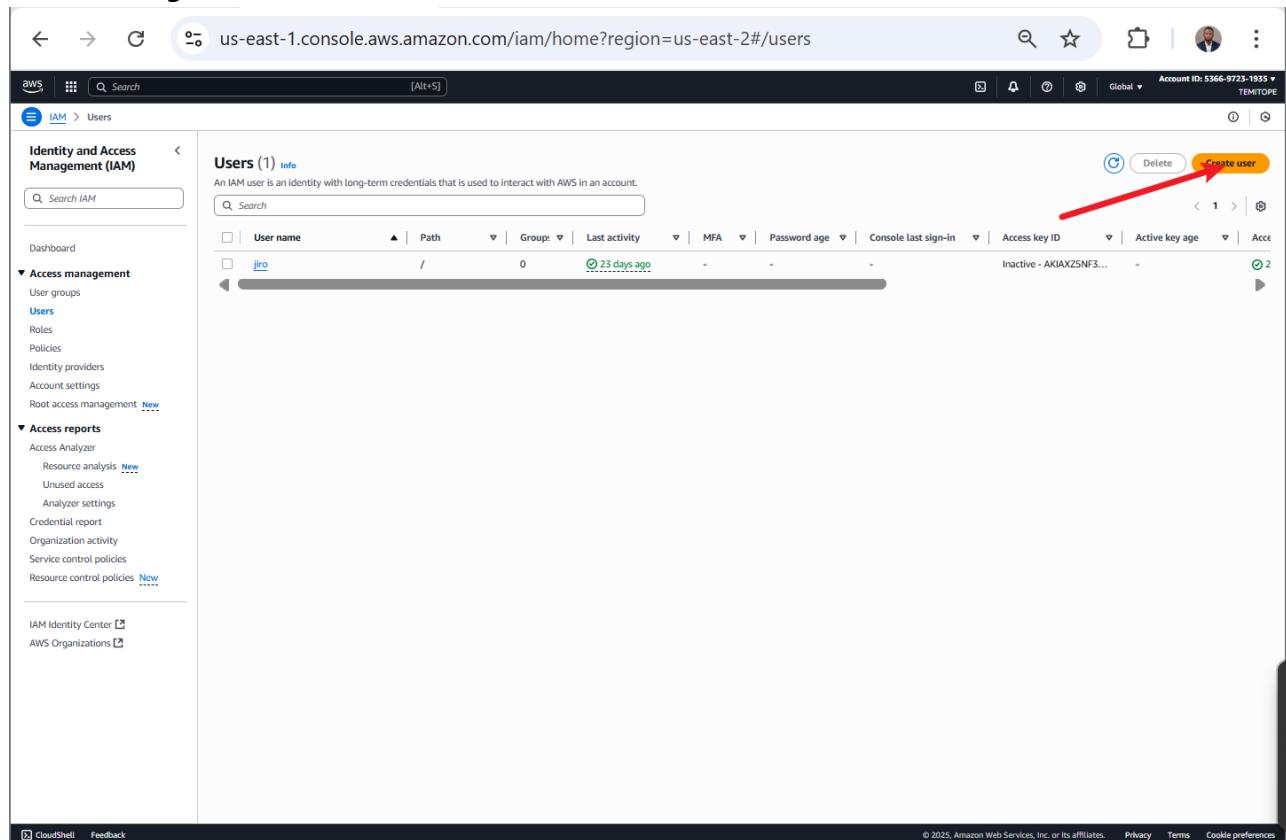
Actions

11. Click on Users, Proceed to Create User



The screenshot shows the AWS IAM Policies page. In the top right corner, there is a green banner that says "Policy policy_for_seun created." Below this, the main title is "Policies (1/1397) Info". The left sidebar has a section titled "Access management" which includes "User groups", "Users" (with a red arrow pointing to it), and "Roles". The main content area shows a table with one row for "policy_for_seun". The table columns are "Policy name", "Type", "Used as", and "Description". The "Policy name" column shows "policy_for_seun", "Type" shows "Customer managed", "Used as" shows "None", and the "Description" column shows "This policy has been created to provid...". At the bottom right of the table, there are buttons for "Actions", "Delete", and "Create policy".

12. On Users Page, Click Create User



The screenshot shows the AWS IAM Users page. In the top right corner, there is a green banner that says "Users (1) Info". Below this, the main title is "Users (1) Info". The left sidebar has a section titled "Access management" which includes "User groups", "Users" (with a red arrow pointing to it), and "Roles". The main content area shows a table with one row for "jdoe". The table columns are "User name", "Path", "Group", "Last activity", "MFA", "Password age", "Console last sign-in", "Access key ID", "Active key age", and "Access". The "User name" column shows "jdoe", "Path" shows "/", "Group" shows "0", "Last activity" shows "23 days ago", "MFA" shows "-", "Password age" shows "-", "Console last sign-in" shows "-", "Access key ID" shows "Inactive - AKIAZ2NF3...", "Active key age" shows "-", and "Access" shows "2". At the top right of the table, there are buttons for "Delete" and "Create user" (with a red arrow pointing to it).

13. Enter User Name, Give User Access to the Console, and Create User as an IAM User

Specify user details

User details

User name: seun

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.

14. Set Custom Password and Force User to Create New Password at Next Login and Click Next

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.
- Custom password
Enter a custom password for the user.

Seun@123

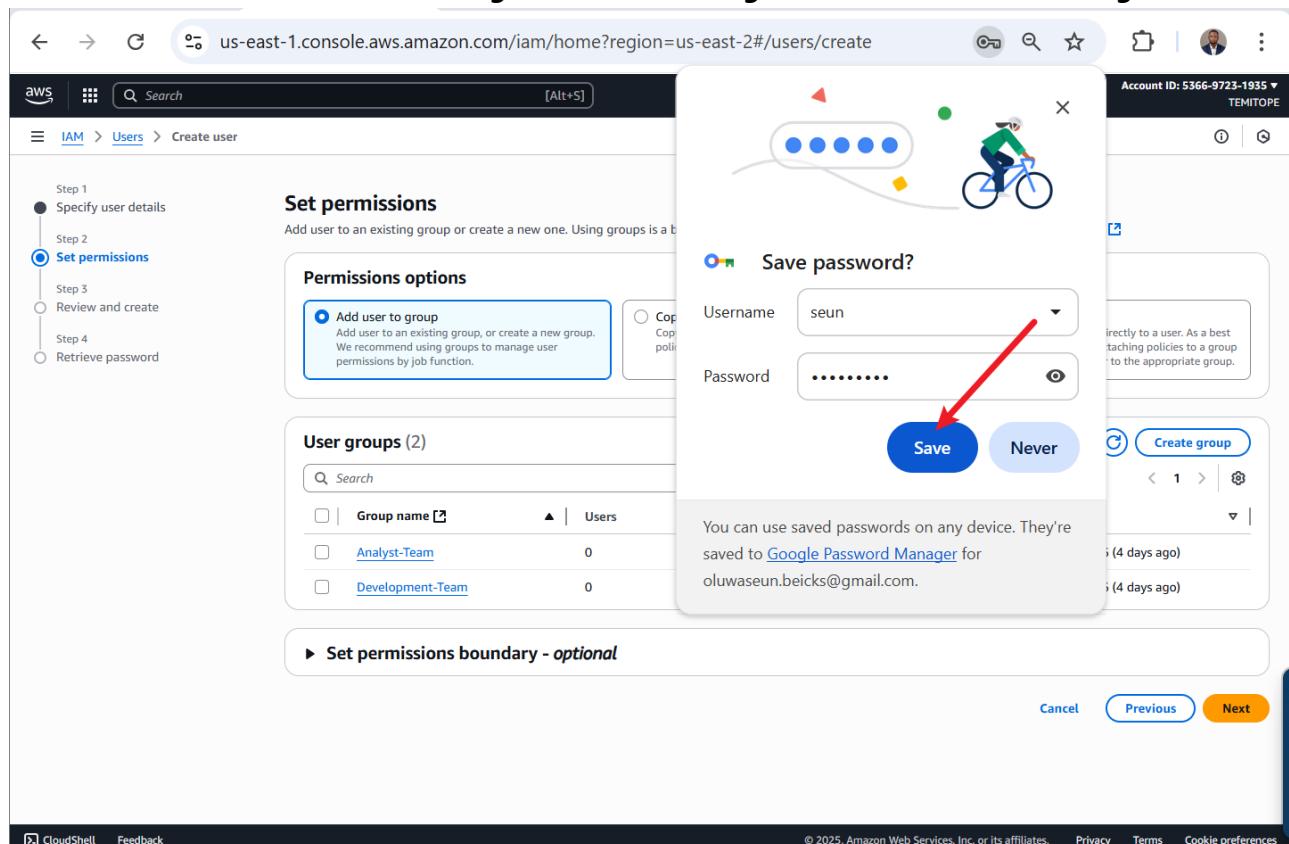
Show password

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.

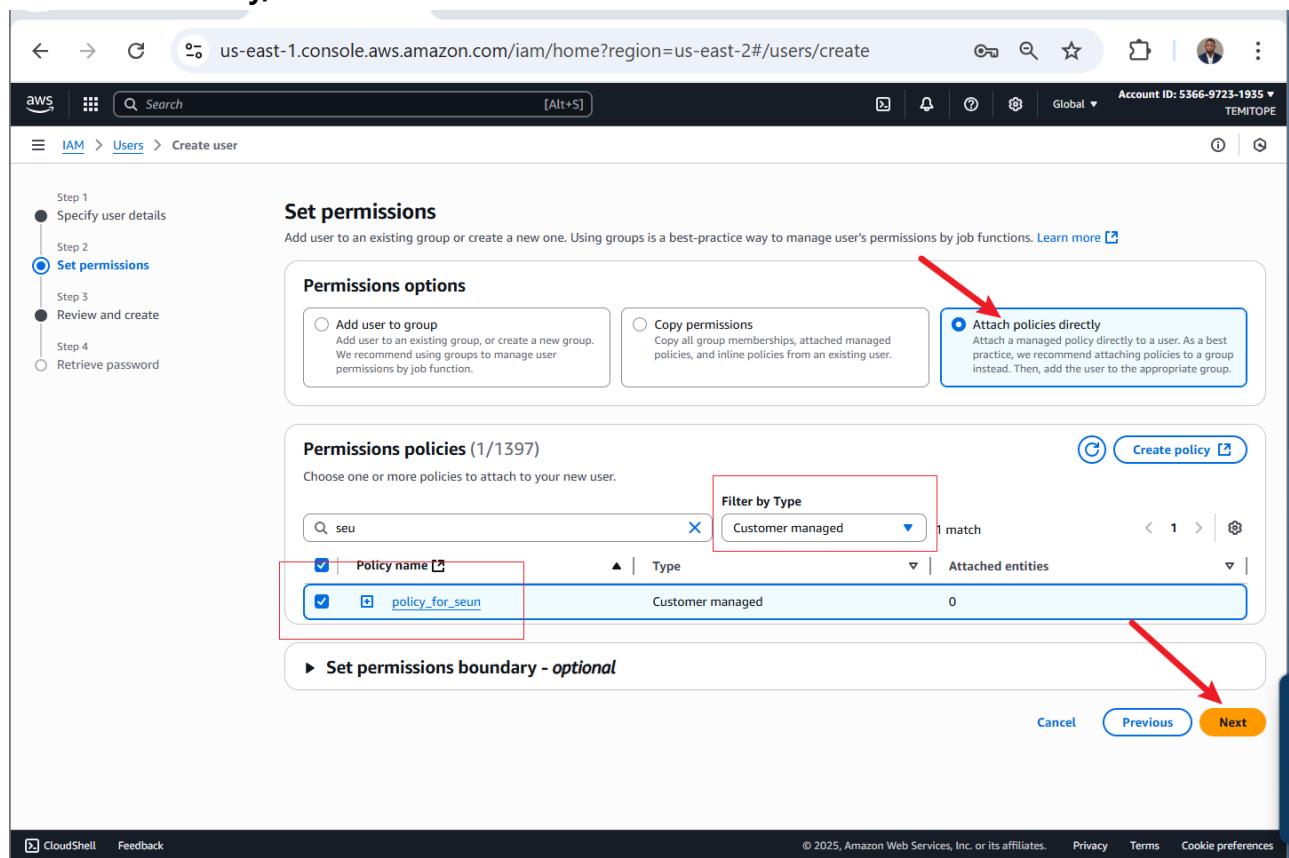
If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel **Next**

15. You Can Click Save Password on Google Password Manager If You Prefer for Next Log In



16. On Permission Page, Click Attach Policy Directly, Filter Policy by Customer Managed Policy, Select Created User Policy, and Click Next



17. Review and Create

Review and create

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

User details

User name	seun	Console password type	Custom password
		Require password reset Yes	

Permissions summary

Name	Type	Used as
IAMUserChangePassword	AWS managed	Permissions policy
policy_for_seun	Customer managed	Permissions policy

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

18. User Seun Successfully Created, Now Download CSV File to Retrieve Chosen Password

User created successfully

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

View user

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://536697231935.signin.aws.amazon.com/console

User name
seun

Console password
***** Show

Email sign-in instructions

Cancel Download .csv file Return to users list

19. Credential Successfully Downloaded, Proceed to User List Page

User created successfully

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

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

Console sign-in URL
https://536697231935.signin.aws.amazon.com/console

User name
seun

Console password
***** Show

Email sign-in instructions

Cancel Download .csv file Return to users list

Part 2: Create Group, Add Users (Jack and Ade), Create EC2/S3 Policy, and Attach to Group

20. Created User Listed, Now Click on User Group

Identity and Access Management (IAM)

Dashboard

Access management

- User groups (highlighted)
- Users** (highlighted)
- Roles
- Policies
- Identity providers
- Account settings
- Root access management New

Access reports

- Access Analyzer
- Resource analysis New
- Unused access
- Analyzer settings
- Credential report
- Organization activity
- Service control policies
- Resource control policies New

Users (2) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Group:	Last activity	MFA	Password age	Console last sign-in
jiro	/	0	23 days ago	-	-	-
seun	/	0	-	-	-	-

21. On User Group Page, Click on Create Group

The screenshot shows the AWS IAM User Groups page. The left sidebar includes sections for Identity and Access Management (IAM), Access management (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management), and Access reports (Access Analyzer, Resource analysis, Unused access, Analyzer settings, Credential report, Organization activity, Service control policies, Resource control policies). The main content area displays 'User groups (0)'. A search bar is at the top, followed by a table header with columns for Group name, Users, Permissions, and Creation time. Below the table, it says 'No resources to display'. At the bottom right of the main area, there is a 'Create group' button, which is highlighted with a red arrow.

22. Set Group Name and Scroll Down and Click Create Group

The screenshot shows the 'Create user group' page. The left sidebar is identical to the previous screenshot. The main form starts with a 'Name the group' section, which includes a 'User group name' input field containing 'Development-team'. A red arrow points to this input field. Below this is an 'Add users to the group - Optional (2)' section, showing two users: 'jiro' and 'seun'. At the bottom is an 'Attach permissions policies - Optional (1082)' section, showing a table with one policy: 'AdministratorAccess' (AWS managed - job function, Permissions policy (1), Provides full access to AWS services an...). The 'AdministratorAccess' row is highlighted with a red arrow.

23. Group Successfully Created, Click Users to Create New User

The screenshot shows the AWS IAM User Groups page. A green success message at the top states "Development-team user group created." Below it, a table titled "User groups (1) Info" lists one group: "Development-team". The table includes columns for "Group name", "Users", "Permissions", and "Creation time". A red arrow points to the "Users" link in the left sidebar under "Access management".

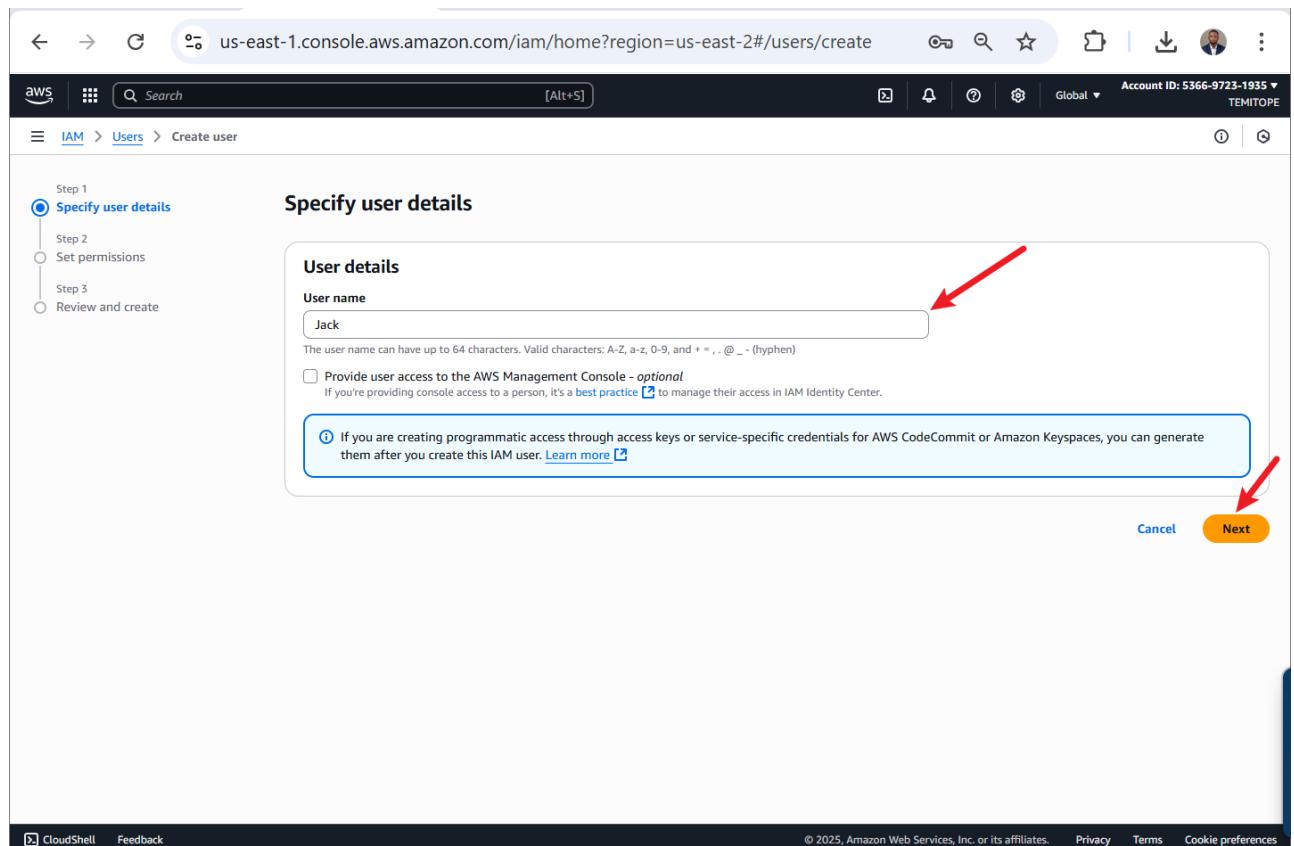
Group name	Users	Permissions	Creation time
Development-team	0	Not defined	Now

24. User List Page Appears, Click Create User

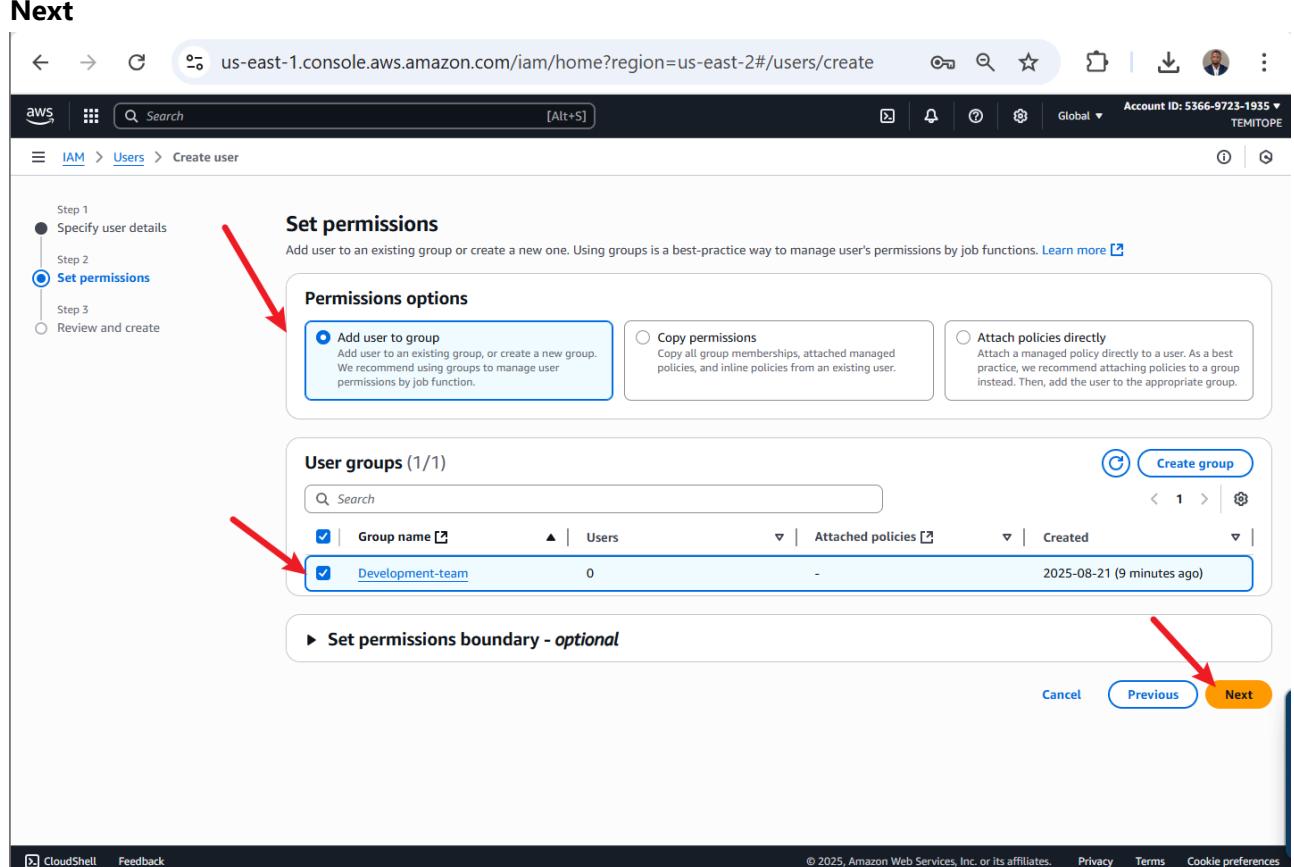
The screenshot shows the AWS IAM Users page. A table titled "Users (2) Info" lists two users: "jiro" and "seun". The table includes columns for "User name", "Path", "Groups", "Last activity", "MFA", "Password age", and "Console last sign-in". A red arrow points to the "Create user" button in the top right corner of the page.

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in
jiro	/	0	23 days ago	-	-	-
seun	/	0	-	-	-	-

25. Choose User Name and Click Next



26. In the Permission Option, Select Add User to Group and Select Development-Team and Click Next



27. Review and Click Create User

The screenshot shows the 'Create user' wizard in the AWS IAM console. The current step is 'Review and create'. The left sidebar shows steps 1 through 3: 'Specify user details', 'Set permissions', and 'Review and create' (which is selected). The main area displays 'User details' (User name: Jack, Console password type: None, Require password reset: No), 'Permissions summary' (Development-team group assigned), and 'Tags' (optional). A red arrow points to the 'Create user' button at the bottom right of the page.

28. User Jack Successfully Created

The screenshot shows the 'Users' page in the AWS IAM console. A green success message box at the top left states 'User created successfully' with a link to 'View user'. The main table lists three users: 'Jack' (created 23 days ago), 'jiro', and 'seun'. The left sidebar includes sections for 'Access management' (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management), 'Access reports' (Access Analyzer, Resource analysis, Unused access, Analyzer settings, Credential report, Organization activity, Service control policies, Resource control policies), and 'IAM Identity Center'. The 'Create user' button is located in the top right corner of the user list table.

29. Create Another User by Clicking Create User

The screenshot shows the AWS IAM console under the 'Users' section. On the left, there's a sidebar with 'Access management' and 'Access reports' sections. The main area displays a table of users with columns for User name, Path, Group, Last activity, MFA, Password age, and Console last sign-in. Three users are listed: Jack, jiro, and seun. The 'Create user' button is highlighted with a red arrow.

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in
Jack	/	0	-	-	-	-
jiro	/	0	23 days ago	-	-	-
seun	/	0	-	-	-	-

30. Choose User Name and Click Next

The screenshot shows the 'Specify user details' step of the IAM user creation wizard. It includes a sidebar with steps: Step 1 (selected), Step 2, and Step 3. The main form has a 'User details' section with a 'User name' input field containing 'Ade'. Below it are optional checkboxes for AWS Management Console access and programmatic access. A note at the bottom explains generating access keys. The 'Next Step' button is highlighted with a red arrow.

31. In the Permission Option, Select Add User to Group and Select Development-Team and Click Next

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

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 (1/1)

Group name	Users	Attached policies	Created
Development-team	0	-	2025-08-21 (24 minutes ago)

Set permissions boundary - optional

Cancel Previous Next

32. Review and Click Create User

Review and create

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

User details

User name	Console password type	Require password reset
Ade	None	No

Permissions summary

Name	Type	Used as
Development-team	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

33. User Ade Successfully Created

The screenshot shows the AWS IAM Users page. The left sidebar is titled "Identity and Access Management (IAM)" and includes sections for Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management), and Access reports (Access Analyzer, Resource analysis, Unused access, Analyzer settings, Credential report, Organization activity, Service control policies, Resource control policies). The main area is titled "Users (4) Info" and contains a table with the following data:

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in
Ade	/	1		-	-	-
Jack	/	1	-	-	-	-
jiro	/	0	23 days ago	-	-	-
seun	/	0	-	-	-	-

At the bottom of the page, there are links for IAM Identity Center, CloudShell, and Feedback, along with copyright information and links for Privacy, Terms, and Cookie preferences.

34. Click Policies to Create New Policy for the Group

The screenshot shows the same AWS IAM Users page as the previous one, but with a red arrow pointing to the "Policies" link in the left sidebar under the "Access management" section. The rest of the interface and data in the table remain the same as in the previous screenshot.

35. Click Create Policy

The screenshot shows the AWS IAM Policies page. On the left, there's a navigation sidebar with 'Identity and Access Management (IAM)' selected. The main area displays a table of policies, with one row highlighted. At the top right of the table, there's a 'Create policy' button. A red arrow points to this button.

Policy name	Type	Used as	Description
AccessAnalyzerService...	AWS managed	None	Allow Access Analyzer to analyze resou...
AdministratorAccess	AWS managed - job function	Permissions policy (1)	Provides full access to AWS services an...
AdministratorAccess-...	AWS managed	None	Grants account administrative permis...
AdministratorAccess-...	AWS managed	None	Grants account administrative permiss...
AIOpsAssistantPolicy	AWS managed	None	Provides ReadOnly permissions requir...
AIOpsConsoleAdmin...	AWS managed	None	Grants full access to Amazon AI Opera...
AIOpsOperatorAccess	AWS managed	None	Grants access to the Amazon AI Opera...
AIOpsReadOnlyAccess	AWS managed	None	Grants ReadOnly permissions to the A...
AlexaForBusinessDev...	AWS managed	None	Provide device setup access to AlexaFo...
AlexaForBusinessFull...	AWS managed	None	Grants full access to AlexaForBusiness ...
AlexaForBusinessGat...	AWS managed	None	Provide gateway execution access to A...
AlexaForBusinessLife...	AWS managed	None	Provide access to Lifesize AVS devices
AlexaForBusinessNet...	AWS managed	None	This policy enables Alexa for Business ...
AlexaForBusinessPol...	AWS managed	None	Provide access to Poly AVS devices
AlexaForBusinessPea...	AWS managed	None	Provide read only access to AlexaForB...

36. Choose EC2 and Select All Actions and Resources

The screenshot shows the 'Specify permissions' step of the 'Create policy' wizard. The left sidebar shows 'Step 1 Specify permissions' is selected. The main area has a title 'Specify permissions' and a sub-section 'Policy editor'. Under 'Actions allowed', there's a section for 'EC2' with a 'Allow' button. A red arrow points to the 'All EC2 actions (ec2:*)' checkbox. Below it, there's a list of actions like 'List', 'Read', 'Write', etc., and a note about dependent permissions.

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

Policy editor

EC2 **Allow** All actions

Specify what actions can be performed on specific resources in EC2.

Actions allowed

Specify actions from the service to be allowed.

Manual actions | Add actions

All EC2 actions (ec2:*)

Access level

- List (Selected 154/194)
- Read (Selected 44/44)
- Write (Selected 448/448)
- Permissions management (Selected 16/16)
- Tagging (Selected 2/2)

Dependent permissions not selected. To grant permissions for the selected resource actions, including additional dependent actions might be required.

- ec2:AssociateIamInstanceProfile requires 1 more action.
- ec2:CreateFlowLogs requires 6 more actions.
- ec2:CreateIam requires 1 more action.
- ec2:CreateIamResourceDiscovery requires 1 more action.
- ec2:CreateLaunchTemplate requires 1 more action.
- ec2:CreateLaunchTemplateVersion requires 1 more action.
- ec2:CreateRouteServer requires 1 more action.
- ec2>CreateVpcEndpoint requires 1 more action.
- ec2:DeleteRouteServer requires 1 more action.
- ec2:DisableIamOrganizationAdminAccount requires 1 more action.
- ec2:EnableFastLaunch requires 1 more action.
- ec2:EnableIamOrganizationAdminAccount requires 3 more actions.
- ec2:EnableRe达abilityAnalyzerOrganizationSharing requires 2 more actions.
- ec2:ReplaceIamInstanceProfileAssociation requires 1 more action.
- ec2:RequestSpotInstances requires 1 more action.
- ec2:RunInstances requires 2 more actions.

37. Select Add More Permission and Select S3 Service

The screenshot shows the 'Create policy' wizard on the AWS IAM console. The current step is 'Add more permissions'. The 'S3' service is selected, and under 'Actions allowed', several actions are selected: List (Selected 16/16), Read (Selected 6/61), Write (Selected 49/49), Permissions management (Selected 27/27), and Tagging (Selected 12/12). A red arrow points to the '+ Add more permissions' button at the bottom of the list.

38. Choose S3 and Select All Actions and All Resources and Click Next

The screenshot shows the 'Create policy' wizard on the AWS IAM console. The current step is 'Add more permissions'. The 'S3' service is selected, and under 'Actions allowed', several actions are selected: List (Selected 16/16), Read (Selected 6/61), Write (Selected 49/49), Permissions management (Selected 27/27), and Tagging (Selected 12/12). A red arrow points to the '+ Add more permissions' button at the bottom of the list.

39. Name Policy and Click Create Policy

Review and create Info

Step 1
Specify permissions

Step 2
Review and create

Policy details

Policy name
Enter a meaningful name to identify this policy.

Description - optional
Add a short explanation for this policy.

Permissions defined in this policy Info

Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, or role), attach a policy to it.

Allow (2 of 450 services)

Service	Access level	Resource	Request condition
EC2	Full access	All resources	None
S3	Full access	All resources	None

Add tags - optional Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

Add new tag
You can add up to 50 more tags.

Create policy

40. Development-Team-Policy Successfully Created, Navigate to User Group Page

Identity and Access Management (IAM)

User groups

Policies (1398) Info

A policy is an object in AWS that defines permissions.

Create policy

Policy name	Type	Used as	Description
AccessAnalyzerServiceRolePolicy	AWS managed	None	Allow Access Analyzer to analyze resou.
AdministratorAccess	AWS managed - job function	Permissions policy (1)	Provides full access to AWS services an.
AdministratorAccess-Amplify	AWS managed	None	Grants account administrative permiss.
AdministratorAccess-AWSFamiliarBeanstalk	AWS managed	None	Grants account administrative permiss.
AIOpsAssistantPolicy	AWS managed	None	Provides ReadOnly permissions require.
AIOpsConsoleAdminPolicy	AWS managed	None	Grants full access to Amazon AI Opera..
AIOpsOperatorAccess	AWS managed	None	Grants access to the Amazon AI Opera..
AIOpsReadOnlyAccess	AWS managed	None	Grants ReadOnly permissions to the A..
AlexaForBusinessDeviceSetup	AWS managed	None	Provide device setup access to AlexaFo..
AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness ..
AlexaForBusinessGatewayExecution	AWS managed	None	Provide gateway execution access to Al.
AlexaForBusinessLifesizeDelegatedAccessPolicy	AWS managed	None	Provide access to Lifesize AVS devices
AlexaForBusinessNetworkProfileServicePolicy	AWS managed	None	This policy enables Alexa for Business ..
AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	None	Provide access to Poly AVS devices
AlexaForBusinessReadOnlyAccess	AWS managed	None	Provide read only access to AlexaForBu..
AmazonAPIGatewayAdministrator	AWS managed	None	Provides full access to create/edit/delete.
AmazonAPIGatewayInvokeFullAccess	AWS managed	None	Provides full access to invoke APIs in A..
AmazonAPIGatewayPushToCloudWatchLogs	AWS managed	None	Allows API Gateway to push logs to us..
AmazonAppFlowFullAccess	AWS managed	None	Provides full access to Amazon AppFlo..
AmazonCloudWatchCloudWatchAccess	AWS managed	None	Provides read only access to Amazon CloudWatch

41. Click Development-Team

The screenshot shows the AWS IAM User groups page. A green banner at the top indicates that a policy has been created. The main table lists one user group named 'development-team'. A red arrow points to the group name 'development-team' in the table.

Group name	Users	Permissions	Creation time
development-team	2	Not defined	1 hour ago

42. Click on Permission Tab, Then Add Permission Button and Attach Policy

The screenshot shows the 'Development-team' group details page. The 'Permissions' tab is selected. A red arrow points to the 'Permissions' tab. Another red arrow points to the 'Add permissions' button in the top right corner of the permissions section.

Summary

User group name: Development-team

Creation time: August 21, 2025, 06:26 (UTC+01:00)

ARN: arn:aws:iam::536697231935:group/Development-team

Permissions (0) Info

You can attach up to 10 managed policies.

Add permissions

43. On Policy Name List, Filter by Customer Managed

Attach permission policies to Development-team

Current permissions policies (0)

Other permission policies (1083)

Filter by Type: Customer managed

Policy name	Type	Description
AdministratorAccess	AWS managed - job function	Provides full access to AWS services an...
AdministratorAccess-Amplify	AWS managed	Grants account administrative permis...
AdministratorAccess-AWSElasticBeanstalk	AWS managed	Grants account administrative permis...
AIOpsAssistantPolicy	AWS managed	Provides ReadOnly permissions require...
AIOpsConsoleAdminPolicy	AWS managed	Grants full access to Amazon AI Opera...
AIOpsOperatorAccess	AWS managed	Grants access to the Amazon AI Opera...
AIOpsReadOnlyAccess	AWS managed	Grants ReadOnly permissions to the A...
AlexaForBusinessDeviceSetup	AWS managed	Provide device setup access to AlexaFo...
AlexaForBusinessFullAccess	AWS managed	Grants full access to AlexaForBusiness ...
AlexaForBusinessGatewayExecution	AWS managed	Provide gateway execution access to AI...
AlexaForBusinessLifesizeDelegatedAccessPolicy	AWS managed	Provide access to Lifesize AVS devices
AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	Provide access to Poly AVS devices
AlexaForBusinessReadOnlyAccess	AWS managed	Provide read only access to AlexaForBu...
AmazonAPIGatewayAdministrator	AWS managed	Provides full access to create/edit/dele...
AmazonAPIGatewayInvokeFullAccess	AWS managed	Provides full access to invoke APIs in ...
AmazonAPIGatewayPushToCloudWatchLogs	AWS managed	Allows API Gateway to push logs to us...
AmazonAppFlowFullAccess	AWS managed	Provides full access to Amazon AppFlo...
AmazonAppFlowReadOnlyAccess	AWS managed	Provides read only access to Amazon A...

44. Select Development-Team-Policy and Click Attach Policy

Attach permission policies to Development-team

Current permissions policies (0)

Other permission policies (1/1083)

Filter by Type: Customer managed

Policy name	Type	Used as	Description
analyst-policy	Customer managed	None	-
CodeBuildBasePolicy-AWSBuildProject1-us-east-2	Customer managed	None	Policy used in trust relationship with C...
CodeBuildBasePolicy-CodeBuild-SonarCloud-Role...	Customer managed	None	-
CodeBuildBasePolicy-JAVA-DAST-PRO-us-east-2	Customer managed	None	Policy used in trust relationship with C...
CodeBuildBasePolicy-JAVA-SAST-us-east-2	Customer managed	None	Policy used in trust relationship with C...
CodeBuildCodeConnectionSourceCredentialsPoli...	Customer managed	None	Policy used in trust relationship with C...
CodeBuildCodeConnectionSourceCredentialsPoli...	Customer managed	None	Policy used in trust relationship with C...
CodeBuildCodeConnectionSourceCredentialsPoli...	Customer managed	None	Policy used in trust relationship with C...
developer-policy	Customer managed	None	-
development-team-policy	Customer managed	None	-
policy_for_seun	Customer managed	Permissions policy (1)	This policy has been created to provid...

Attach policies

45. Policy Successfully Attached

The screenshot shows the AWS IAM Groups page. The URL is us-east-1.console.aws.amazon.com/iam/home?region=us-east-2#/groups/details/Development-team. The page displays the 'Development-team' group details. In the 'Permissions' tab, a single policy named 'development-team-policy' is listed under 'Permissions policies'. This policy is described as 'Customer managed' and has one attached entity. The ARN of the policy is listed as `arn:aws:iam::536697231935:group/Development-team`.

Part 3: Enable MFA for User Seun

46. Click Users to Manage User

The screenshot shows the AWS IAM Groups page. The URL is us-east-1.console.aws.amazon.com/iam/home?region=us-east-2#/groups. A red arrow points to the 'Users' link in the 'Access management' section of the left sidebar. The main content area shows the 'User groups' section, which lists the 'Development-team' group.

47. Click on Seun

The screenshot shows the AWS IAM 'Users' page with four entries:

- Ade: Path /, Last activity - (never), MFA - (none), Password age - (never), Console last sign-in - (never), Access key ID - (none), Active key age - (never), Access key last used - (never), ARN: arnawsiam:536697231935:user/Ade, Creation time: 1 hour ago, Status: Disa.
- Jack: Path /, Last activity 23 days ago, MFA - (none), Password age - (never), Console last sign-in - (never), Access key ID - (none), Active key age - (never), Access key last used - (never), ARN: arnawsiam:536697231935:user/jack, Creation time: 2 hours ago, Status: Disa.
- jiro: Path /, Last activity 23 days ago, MFA - (none), Password age - (never), Console last sign-in - (never), Access key ID - (none), Active key age - (never), Access key last used - (never), ARN: arnawsiam:536697231935:user/jiro, Creation time: 23 days ago, Status: Disa.
- seun: Path /, Last activity 2 hours ago, MFA - (none), Password age - (never), Console last sign-in - (never), Access key ID - (none), Active key age - (never), Access key last used - (never), ARN: arnawsiam:536697231935:user/seun, Creation time: 2 hours ago, Status: Enal.

48. Click on Security Credentials and Assign MFA

The screenshot shows the AWS IAM 'User details' page for 'seun'. The 'Security credentials' tab is active. In the 'Multi-factor authentication (MFA)' section, there is a button labeled 'Assign MFA device' with a red arrow pointing to it.

49. Enter Device Name and Choose Authentication Option

The screenshot shows the 'Select MFA device' step of the AWS IAM process. At the top, there are two tabs: 'Select MFA device' (selected) and 'Set up device'. The main area is titled 'Select MFA device' with a 'Info' link. It has a sub-section 'MFA device name' where the text 'Seun-Auth' is entered. Below this, under 'Device options', there are three choices: 'Paskey or security key', 'Authenticator app' (which is selected and highlighted in blue), and 'Hardware TOTP token'. At the bottom right are 'Cancel' and 'Next' buttons.

50. Show QR Code, Scan QR Code, Provide Two Subsequent Codes on the Authenticator App, and Add MFA

The screenshot shows the 'Set up device' step of the AWS IAM process. At the top, there are two tabs: 'Select MFA device' (selected) and 'Set up device' (highlighted in blue). The main area is titled 'Set up device' with a 'Info' link. It has a sub-section 'Authenticator app' with instructions to install a compatible application like Google Authenticator, Duo Mobile, or Authy app on your mobile device or computer. It also shows a QR code for scanning and a secret key for typing. Step 1 shows the QR code, Step 2 shows the secret key, and Step 3 shows the two consecutive MFA codes. At the bottom right are 'Cancel', 'Previous', and 'Add MFA' buttons. A small icon bar at the bottom right contains icons for cloudShell, Feedback, and other services.

51. Successfully Assigned MFA

The screenshot shows the AWS IAM console interface. The URL is us-east-1.console.aws.amazon.com/iam/home?region=us-east-2#/users/details/seun?section=security_credentials. The sidebar on the left shows 'Identity and Access Management (IAM)'. The main content area displays the user 'seun' details. A prominent message at the top says 'MFA device assigned' with a note about registering up to 8 devices. Below this, the 'Security credentials' tab is selected, showing 'Console access Enabled with MFA' and a link to 'Create access key'. The 'Multi-factor authentication (MFA)' section shows one MFA device assigned, with options to 'Remove', 'Resync', or 'Assign MFA device'. The 'Access keys (0)' section indicates no access keys have been created. The bottom right corner features the AWS logo.

Project Reflection

Project Outcomes

The project achieved its goals:

- Understood IAM components (users, groups, policies).
- Created an EC2 policy for user Seun and an EC2/S3 policy for the Development-team group with users Jack and Ade.
- Enabled MFA for Seun, enhancing security.
- Applied best practices like least privilege, descriptive names, and MFA.

Challenges Faced

- Navigating the AWS Console was initially complex; the search bar and sidebar were key.
- Configuring policies required balancing granularity and access.
- Differentiating user vs. group permissions was initially unclear.
- Setting up MFA involved learning authenticator apps and QR code scanning.

Lessons Learned

- Granular policies enhance security.
- Groups streamline permission management for teams.
- MFA significantly boosts account security.
- Clear documentation prevents errors.
- Best practices like least privilege and MFA are practical and effective.

Potential Improvements

- Enable MFA for Jack and Ade for comprehensive security.
- Include IAM role creation (e.g., for AWS services).
- Test policies using the IAM Policy Simulator.

- Ensure consistent naming in documentation.

Conclusion

The project built practical IAM skills, including policy creation, user/group management, and MFA implementation for secure AWS resource management. Challenges like console navigation, policy setup, and MFA configuration were overcome, reinforcing best practices. Adding MFA for all users, role-based exercises, and policy testing would enhance future iterations. I'm now confident in applying IAM and MFA to real-world AWS scenarios.