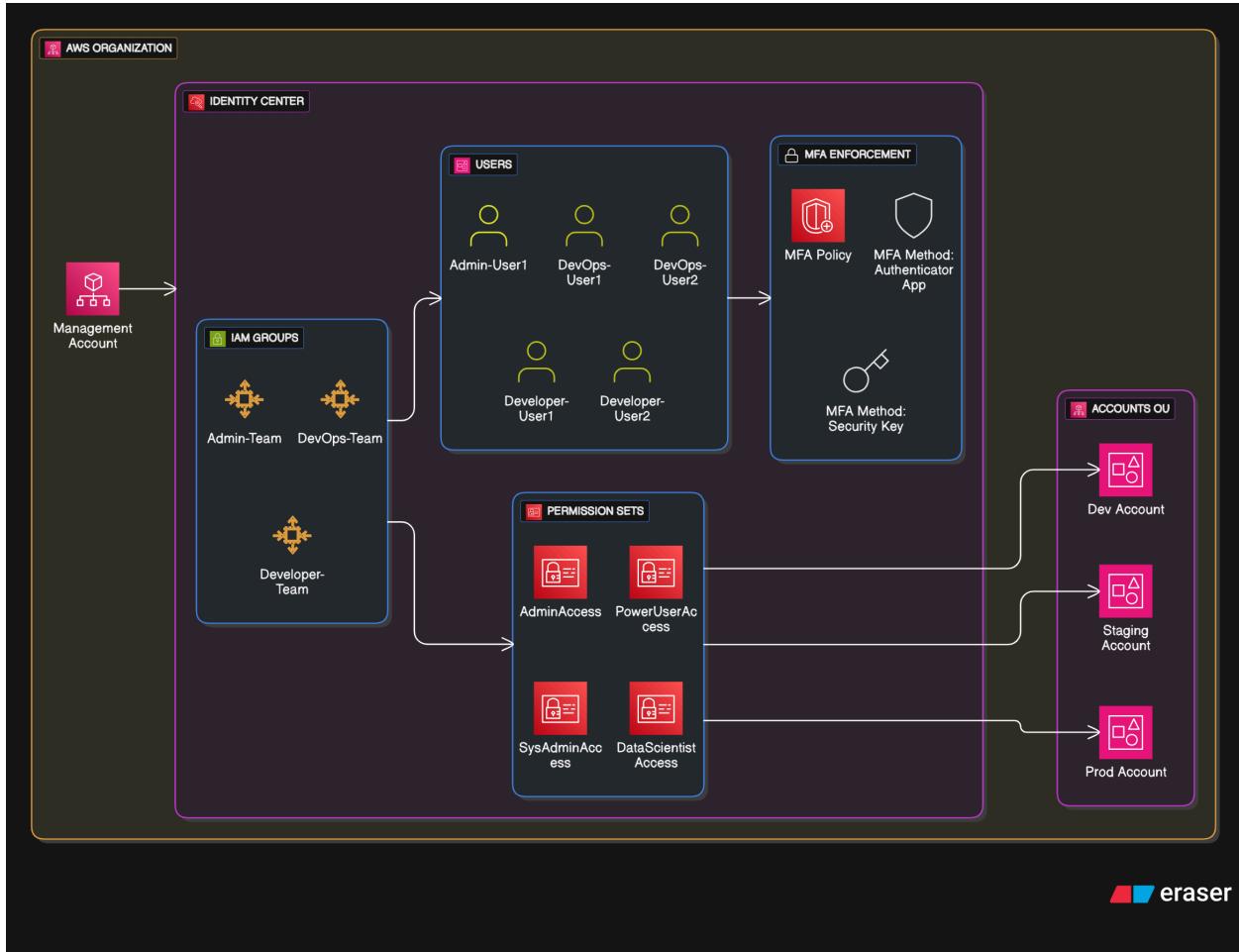


# PROJECT 1

## AWS IAM, Identity Center & Organization Project



**Team Leader: Chinazor Nwode**

**Co-Leader: Ifunanya Benedicta**

This document provides a complete overview of our AWS Organization setup project. The objective was to build a secure, multi-account AWS environment with centralized billing, role-based access control, and comprehensive identity management using AWS Identity Center.

- **Project Date:** 11-07-2025
- **Team Members:** POD 15
- **Cohort:** 15

- **Key Achievements:**

- 1 Management account + 3 Member accounts (Dev, Staging, Production)
- AWS Identity Center configured with 5 users in 3 role-based groups
- 4 permission sets created and assigned across accounts
- MFA enforced for all users
- Complete cross-account access validation

## Organization Setup

- 1 Management Account (Root)
- 3 Member Accounts (Dev, Staging, Production)
- Centralized billing & governance
- Account isolation & security

## Identity Center (SSO)

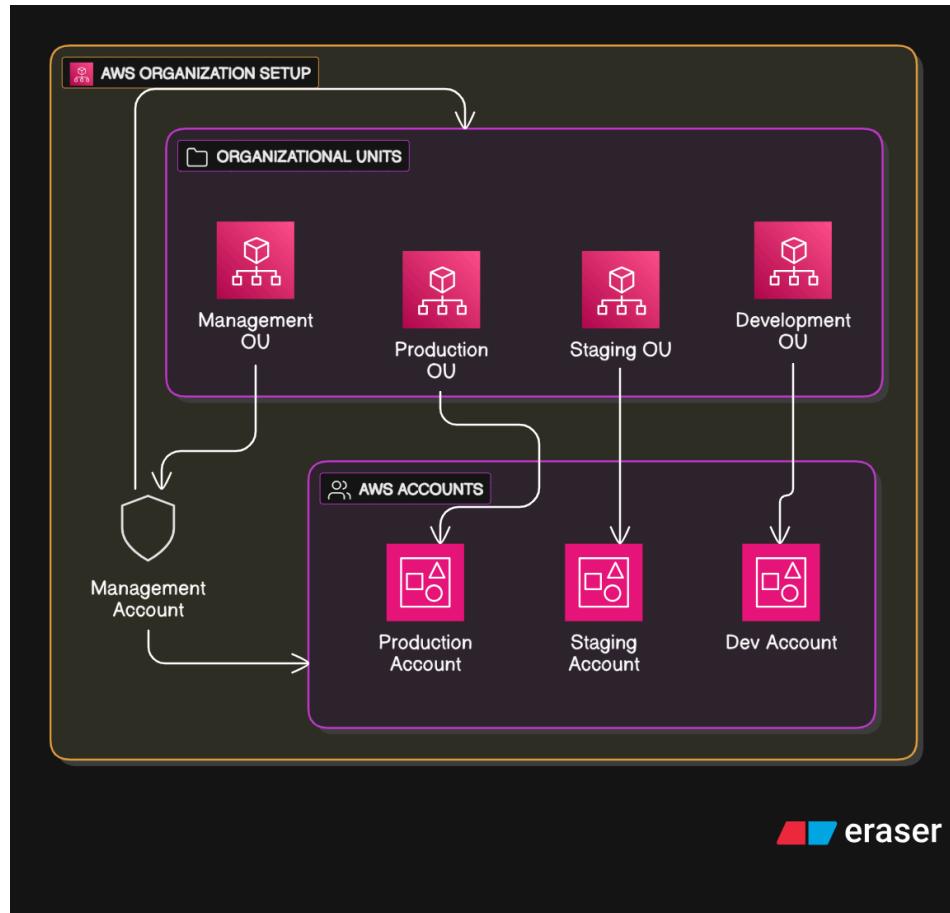
- 5 Users in 3 Groups
- 4 Permission Sets
- Single Sign-On Portal
- MFA for all users

## Security & Testing

- Multi-Factor Authentication
- Least Privilege Access
- Login Testing & Verification
- Complete Documentation

---

# TASK 1: Set up AWS Organization



## 1.1 Organization Setup

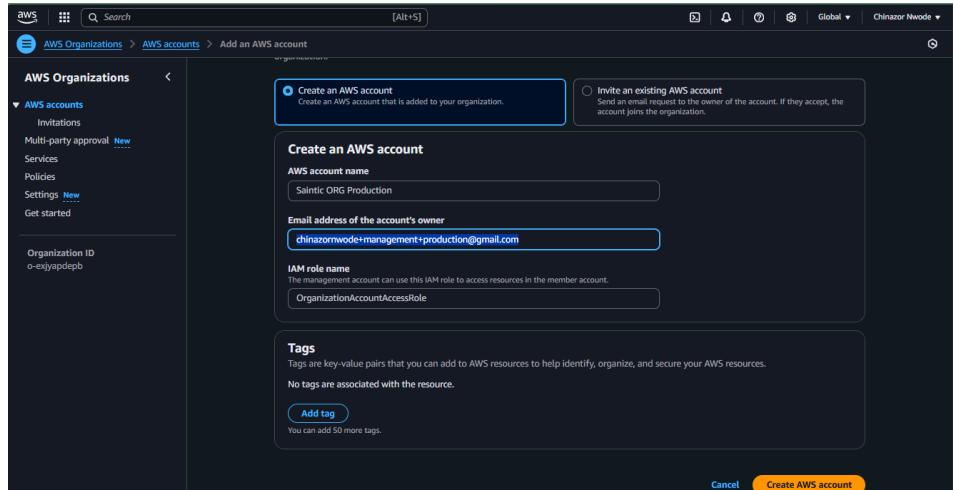
**Navigate to AWS Organizations:**

- Search for "Organizations" in the services search bar
- Click on "AWS Organizations"

The screenshot shows the AWS Organizations service page. The search bar at the top contains "aws organizations". The main area displays the "Services" section with "AWS Organizations" listed first, described as "Central governance and management across AWS accounts". Below it are "Resource Access Manager" and "AWS Private Certificate Authority". The "Features" section includes "Delegated administrator for AWS Organizations" and "AI services opt-out policies". A sidebar on the left lists "Services", "Features", "Resources", "Documentation", "Knowledge articles", "Marketplace", "Blog posts", "Tutorials", and "Events". A feedback section at the bottom asks "Were these results helpful?" with "Yes" and "No" buttons. The right side of the screen shows a "Create application" interface with tabs for "Applications" and "Regions".

### Create Organization:

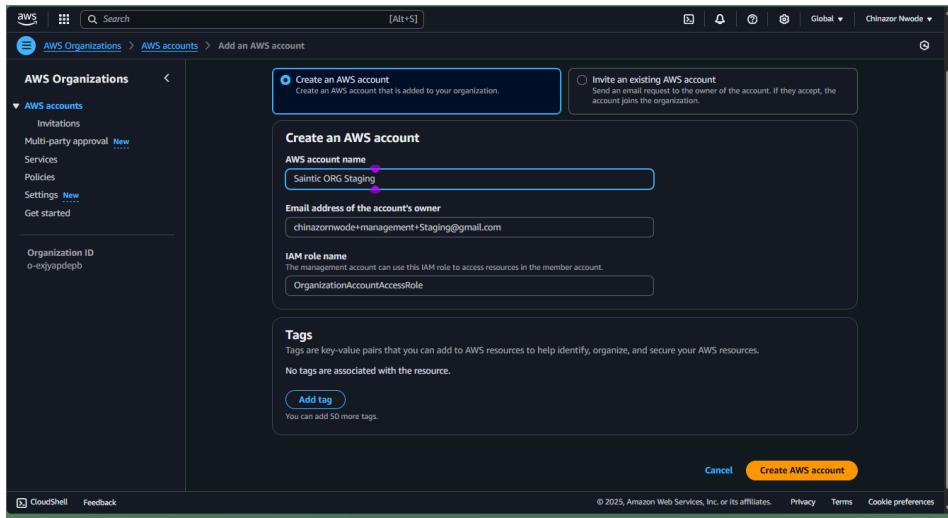
1. Click "Create organization" button
2. Choose "Enable all features" (recommended)
3. Click "Create organization"



### Create Member Accounts:

#### Development Account:

- Click "Add an AWS account"
- Select "Create an AWS account"
- Account name: "Saintic ORG Development"
- Email: chinazornwode+SainticOrgDevelopment@gmail.com
- IAM role name: OrganizationAccountAccessRole
- Click "Create AWS account"



### Staging Account:

- Repeat the process with:
- Account name: **Saintic ORG Staging**
- Email: chinazornwode+SainticOrgStaging@gmail.com

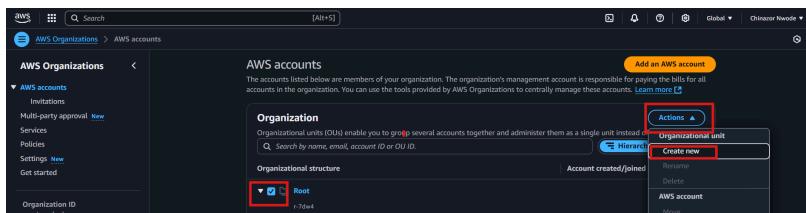
### Production Account:

- Repeat the process with:
- Account name: **Saintic ORG Production**
- Email: chinazornwode+SainticOrgProduction@gmail.com

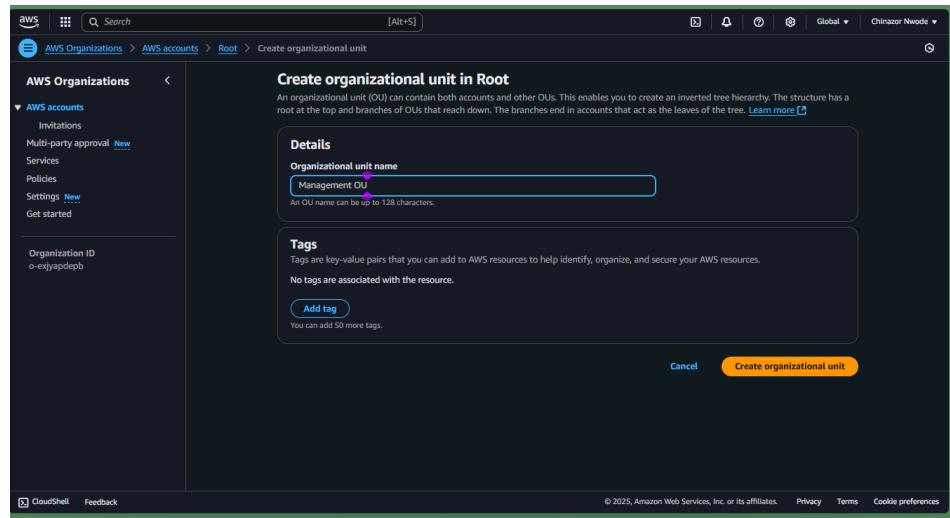
## 1.2 Create Organizational Units (OUs)

### Management OU:

- On the organization tab, click on the checkbox before the root
- click on "Action" Dropdown button
- Then clicked "Create New"

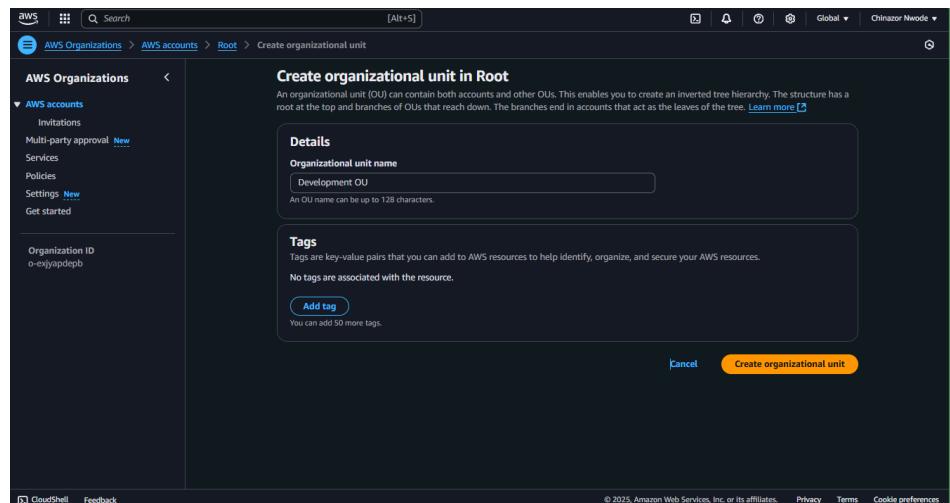


- Click "Create organizational unit"
- Name: **Management OU**



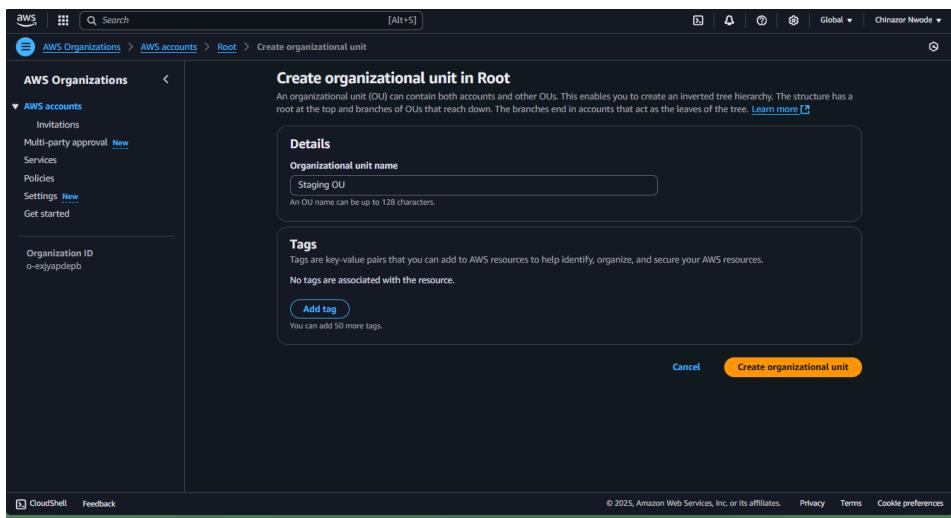
### Development OU:

- Create OU with name: Development OU



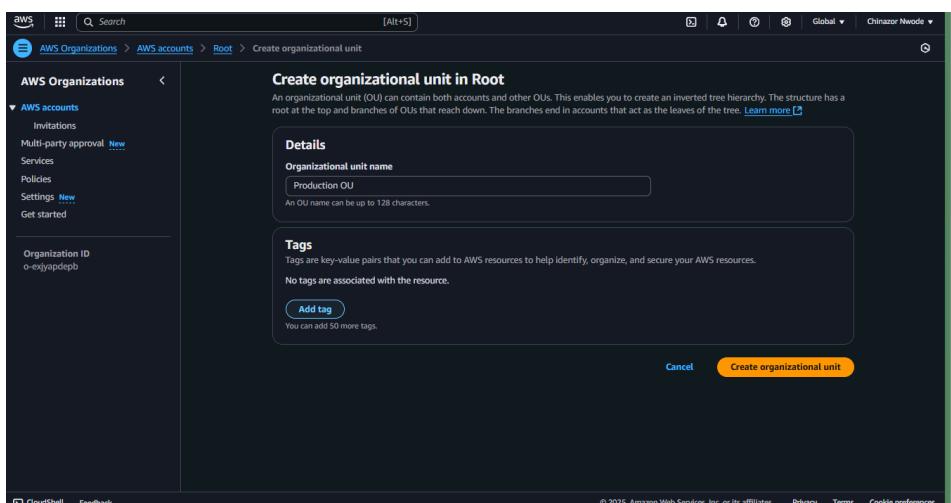
### Staging OU:

- Create OU with name: Staging OU



### Production OU:

- Create OU with name: `Production OU`



### 1.3 Move Accounts to their OUs:

- Moved each account to its corresponding OU
- Verify organizational structure

The screenshot shows the AWS Organizations console with the following details:

- Root OU:** Contains four child OUs: Development OU, Management OU, Production OU, and Staging OU.
- Development OU:** Contains one member account: Salinic ORG Development (Created 2025/07/09).
- Management OU:** Contains one member account: Chinazor Nwode (management account, Joined 2025/05/09).
- Production OU:** Contains one member account: Salinic ORG Production (Created 2025/07/09).
- Staging OU:** Contains one member account: Salinic ORG Staging (Created 2025/07/10).

In Task 1, the team successfully established a comprehensive AWS Organization structure that serves as the foundation for the entire multi-account environment.

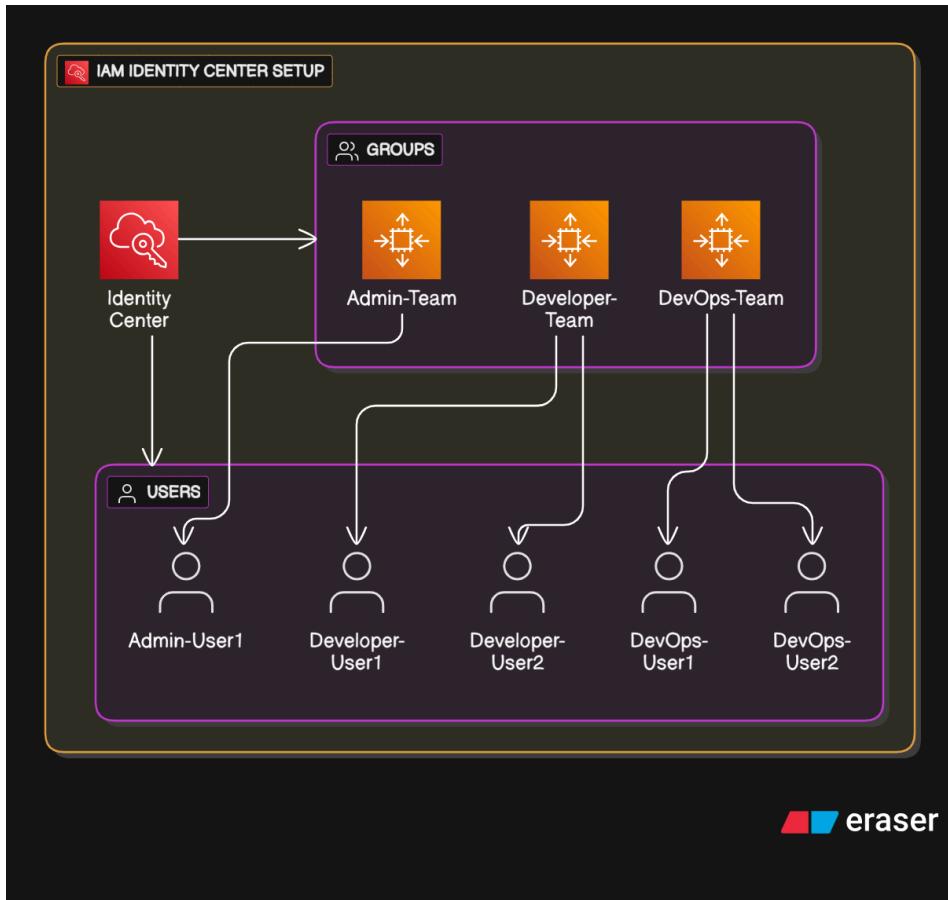
- Created one management account and three specialized member accounts (Development, Production, and Staging)[1]
- Organized these accounts into logical Organizational Units (OUs) based on function (Management, Development, Production, and Staging)[2].
- Implemented centralized billing and governance mechanisms[3]
- Established clear account isolation and security boundaries[4].

This organizational structure provides the architectural foundation for implementing role-based access control, centralized security policies, and effective resource management across all AWS accounts. The hierarchical OU structure also enables more granular policy application and ensures proper separation between development, staging, and production environments.[5].

---

## TASK 2: Create Users and Groups in Identity Center

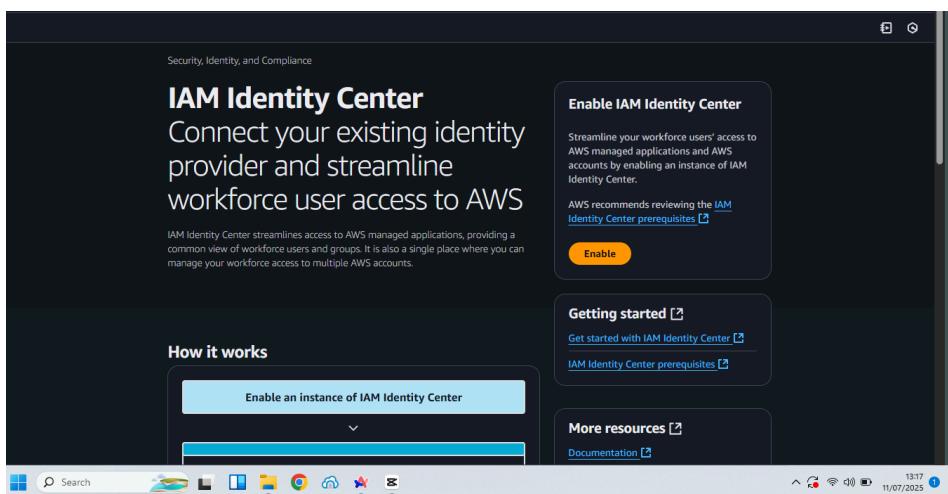
### Overview Flowchart



## 2.1 Enable IAM Identity Center

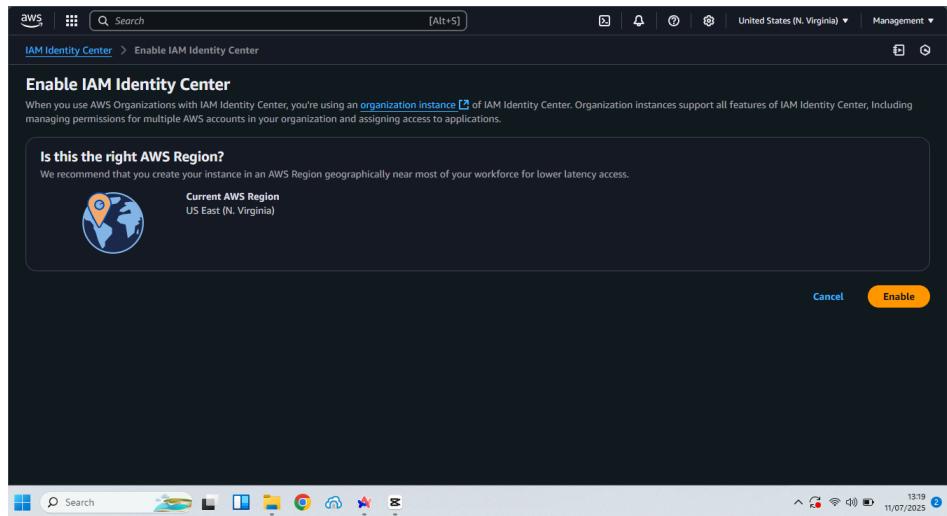
**Navigate to Identity Center:**

- Search for "IAM Identity Center" in the AWS Console
- Click "Enable IAM Identity Center"



**Choose Region:**

- Select your preferred region for Identity Center
- Click "Enable"



### Identity Center Dashboard:

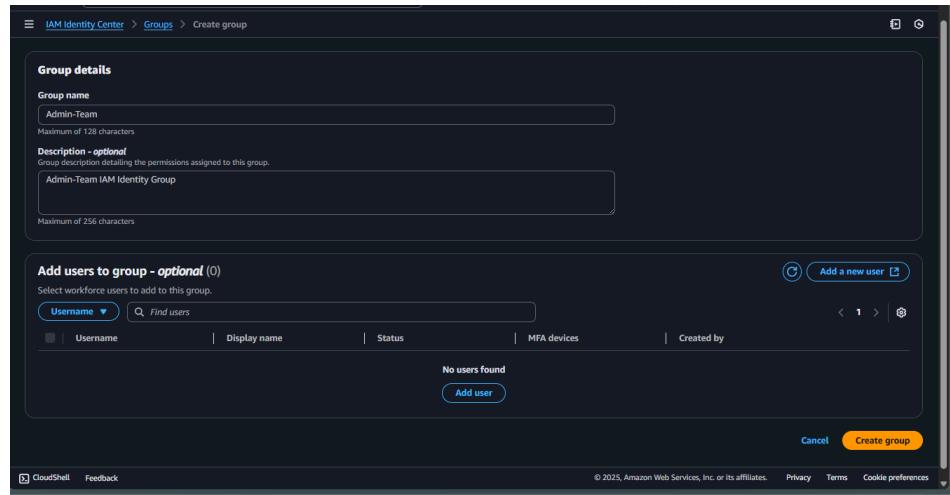
- Verified successful enablement
- Accessed the Identity Center dashboard

The screenshot shows the IAM Identity Center dashboard. On the left, there's a sidebar with 'Managing instance ssoins-7223eac6faf85e80' and links for 'Dashboard', 'Multi-account permissions', 'Application assignments', and 'Related consoles' (CloudTrail, AWS Organizations, IAM). The main content area has several sections: 'Monitor activities in your instances of IAM Identity Center' (with a note about CloudTrail), 'IAM Identity Center setup' (with a 'Confirm your identity source' button), and 'What's new' (with a note about upcoming CloudTrail changes). The 'Confirm your identity source' section is highlighted with a yellow box.

## 2.2 Create Team Groups

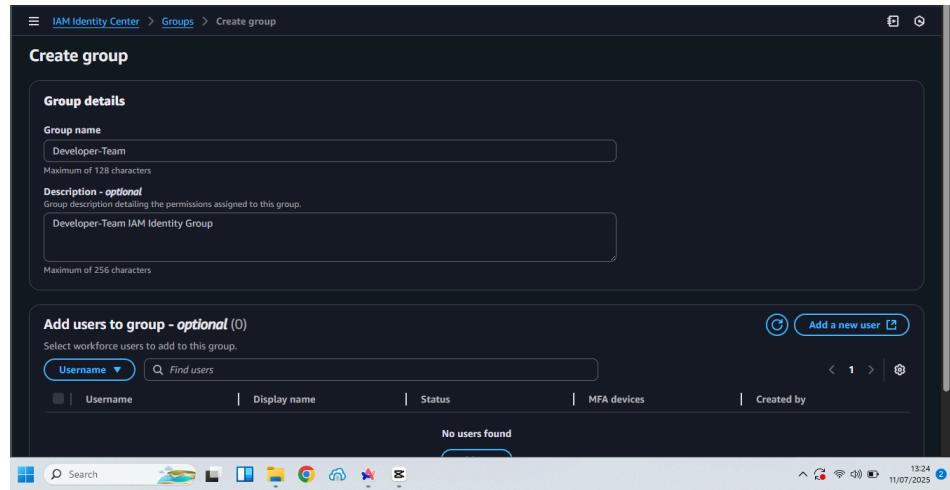
### Admin Team Group:

- Navigated to "Groups" in Identity Center
- Click "Create group"
- Group name: `Admin-Team`
- Description: `Admin-Team IAM Identity Group`
- Click "Create group"



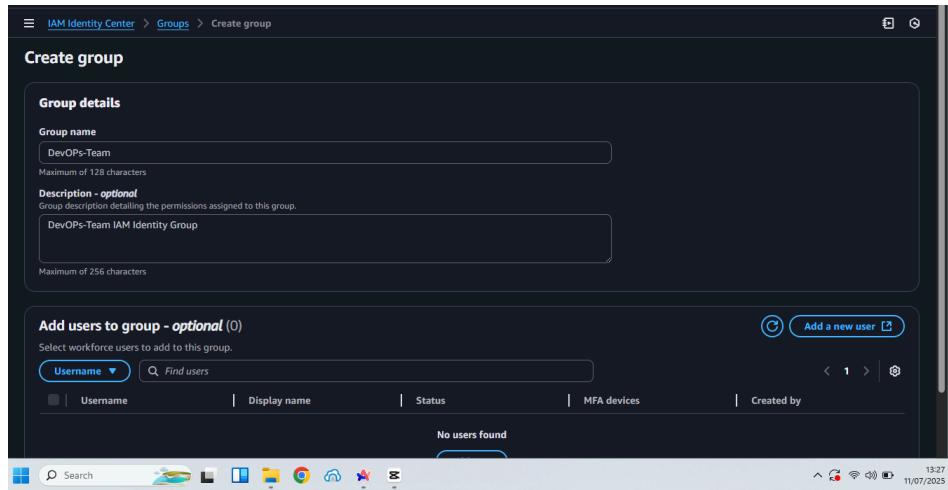
### Developer Team Group:

- Created group with name: **Developer-Team**
- Description: **Developer-Team IAM Identity Group**



### DevOps Team Group:

- Create group with name: **DevOps-Team**
- Description: **DevOps-Team IAM Identity Group**



**Screenshot Overview of the Group Created for the IAM Identity Groups**

Group name	Description	Created by
Developer-Team	-	Manual
DevOps-Team	-	Manual
Root Users	-	Manual
Admin-Team	-	Manual

## 2.3 Create Users and Assign to Groups

### Admin User Creation:

- Navigate to "Users" in Identity Center
- Click "Add user"
- Username: `Admin-User1`
- Email: `Chinazornwode+admin-user1@gmail.com`
- First name: `Admin`
- Last name: `User 1`
- Display name: `Admin User 1`

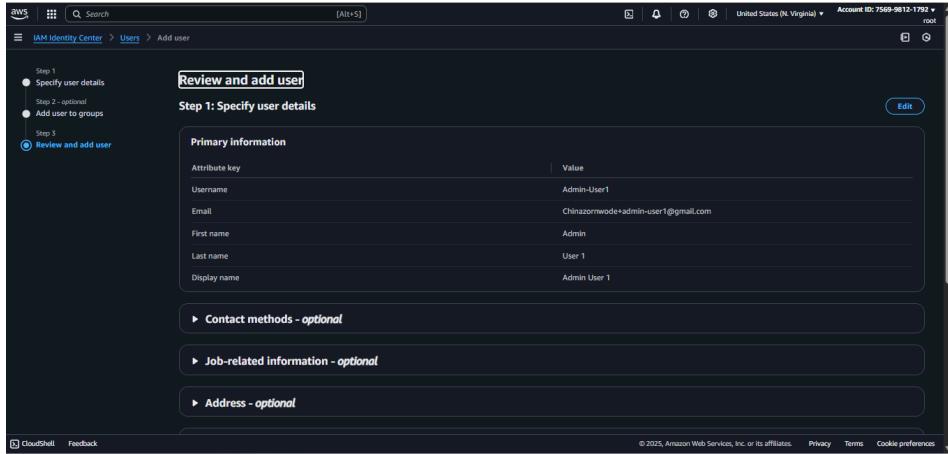
The screenshot shows the 'Specify user details' step in the AWS IAM Identity Center. The 'Username' field contains 'Admin-User1'. Under 'Password', the 'Send an email to this user with password setup instructions' option is selected. The 'Email address' and 'Confirm email address' fields both contain 'Chinazormode-admin-user1@gmail.com'. The 'First name' field is 'Admin', 'Last name' is 'User 1', and 'Display name' is 'Admin User 1'.

### Assign Admin User to Group:

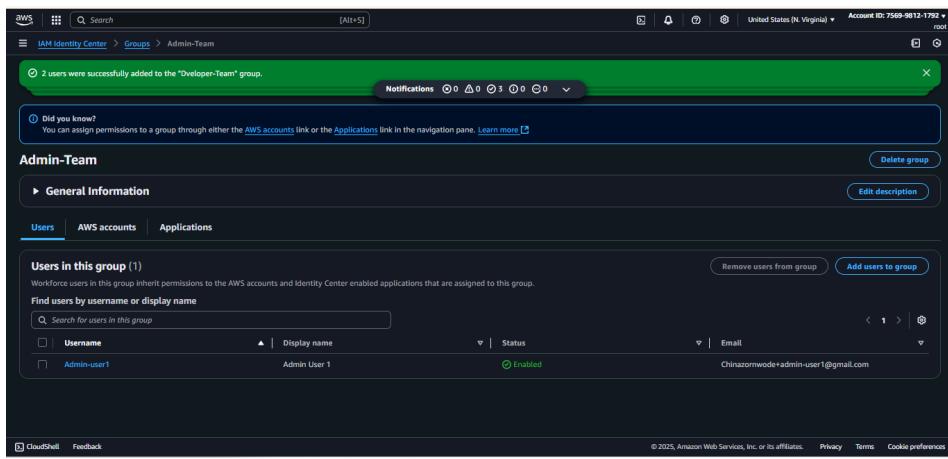
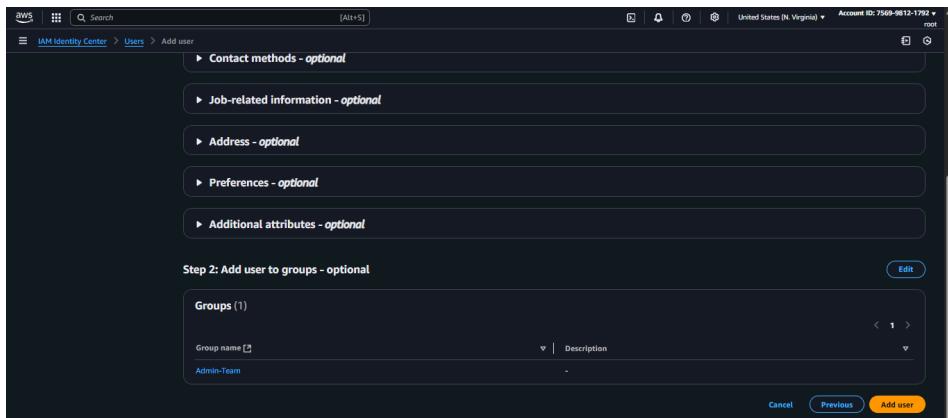
- Select "Admin-Team" group
- Click "Add user"
- Choose "Send an email to the user with password setup instructions"

The screenshot shows the 'Add user to groups - optional' step. A table lists four groups: 'Developer-Team', 'DevOps-Team', 'Root Users', and 'Admin-Team'. The 'Admin-Team' row is highlighted with a blue background, indicating it is selected. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

- Reviewed our settings



- Then create



we've created additional user accounts and assigned them to their respective groups for my project, following the same process as before

### DevOps Users Creation:

- Create DevOps-User1 with email Chinazornwode+DevOPs-user1@gmail.com

- Create **DevOps-User2** with email `Chinazornwode+DevOPs-user2@gmail.com`
- Assign both to **DevOps-Team** group

### Developer Users Creation:

- Create **Developer-User1** with email `Chinazornwode+Developer-user1@gmail.com`
- Create **Developer-User2** with email `Chinazornwode+Developer-user2@gmail.com`
- Assign both to **Developer-Team** group

### Screenshot Overview of the Users Assigned to Group Created for the IAM Identity Users

The image contains two screenshots of the AWS IAM Identity Center Groups page.

**Screenshot 1: Admin-Team Group**

This screenshot shows the "Admin-Team" group details. It lists one user, "Admin-user1", who is an "Admin User". The user's display name is "Admin User 1", status is "Enabled", and email is "Chinazornwode+admin-user1@gmail.com".

Username	Display name	Status	Email
Admin-user1	Admin User 1	Enabled	Chinazornwode+admin-user1@gmail.com

**Screenshot 2: Developer-Team Group**

This screenshot shows the "Developer-Team" group details. It lists two users, "Developer-user1" and "Developer-user2", both of whom are "Developer Users". The users' display names are "Developer User 1" and "Developer User 2", both statuses are "Enabled", and their emails are "Chinazornwode+Developer-user1@gmail.com" and "Chinazornwode+Developer-user2@gmail.com" respectively.

Username	Display name	Status	Email
Developer-user1	Developer User 1	Enabled	Chinazornwode+Developer-user1@gmail.com
Developer-user2	Developer User 2	Enabled	Chinazornwode+Developer-user2@gmail.com

The screenshot shows the AWS IAM Identity Center Groups page. The navigation bar at the top includes 'AWS', 'Search' (with a placeholder '(Alt+S)'), and account information ('United States (N. Virginia) Account ID: 7569-9812-1792 root'). Below the navigation is a green notification bar stating '2 users were successfully added to the "Developer-Team" group.' A 'Notifications' section shows 0 alerts. A 'Did you know?' box provides information about assigning permissions to groups through AWS accounts or Applications.

The main content area is titled 'DevOps-Team' and shows 'General Information'. Below this are tabs for 'Users' (selected), 'AWS accounts', and 'Applications'. The 'Users' tab displays 'Users in this group (2)'. A search bar allows finding users by username or display name. Two users are listed: 'DevOps-User1' (Enabled, Email: Chinazormwode+DevOps-user1@gmail.com) and 'DevOps-User2' (Enabled, Email: Chinazormwode+DevOps-user2@gmail.com). Buttons for 'Remove users from group' and 'Add users to group' are visible.

In Task 2, the team successfully implemented AWS IAM Identity Center (formerly SSO) as the central authentication and identity management system.

- Enabled IAM Identity Center in their chosen AWS region
- Created three functional groups (Admin-Team, Developer-Team, DevOps-Team) to reflect organizational roles
- Established five user accounts and assigned them to appropriate groups based on their responsibilities

This identity management foundation enables centralized user administration, simplifies access management across accounts, and establishes the groundwork for implementing the principle of least privilege through role-based access control.

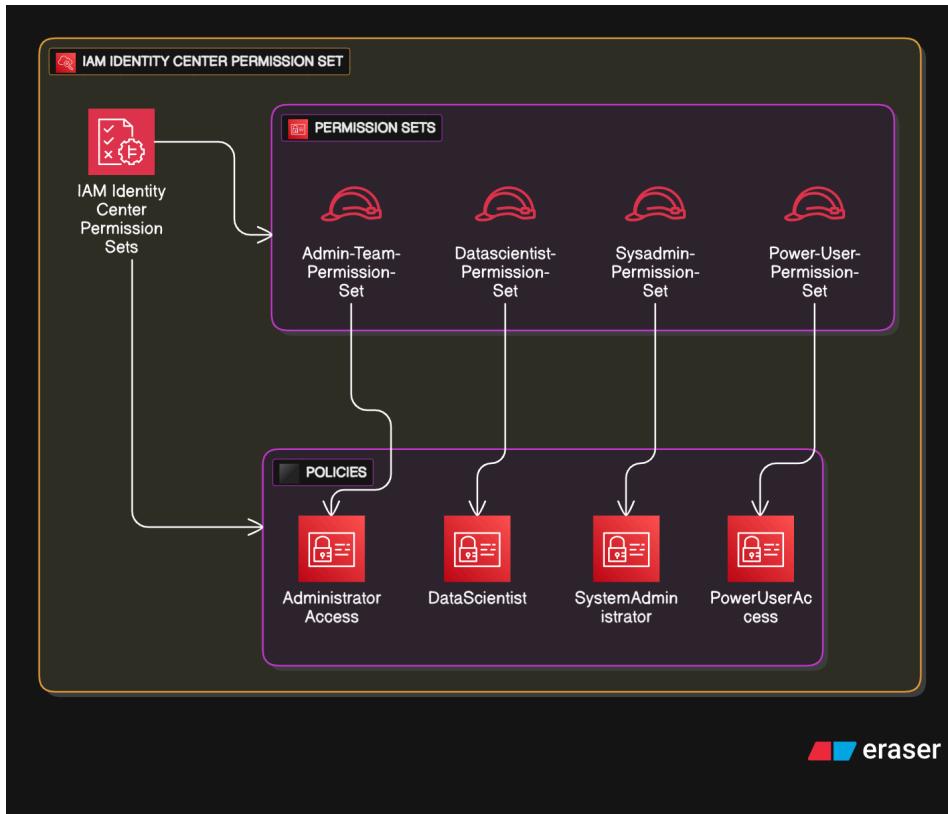
---

=====

=====

## TASK 3: Create Permission Sets

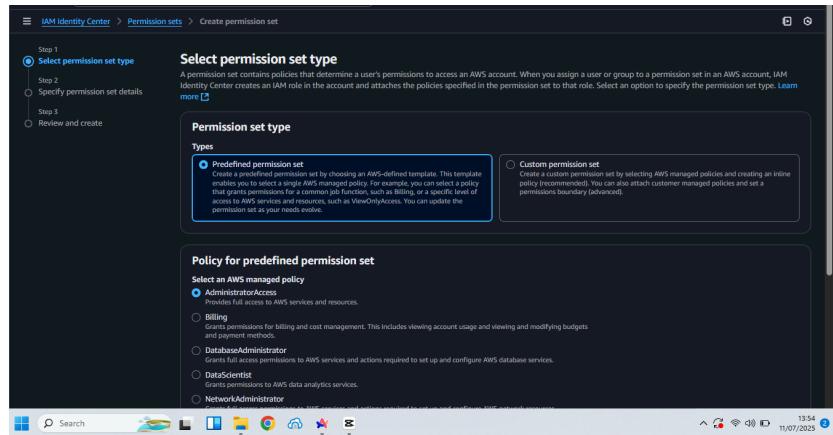
### Overview Flowchart



### 3.1 Navigate to Permission Sets

#### Access Permission Sets:

- In Identity Center, navigate to "Permission sets"
- Click "Create permission set"

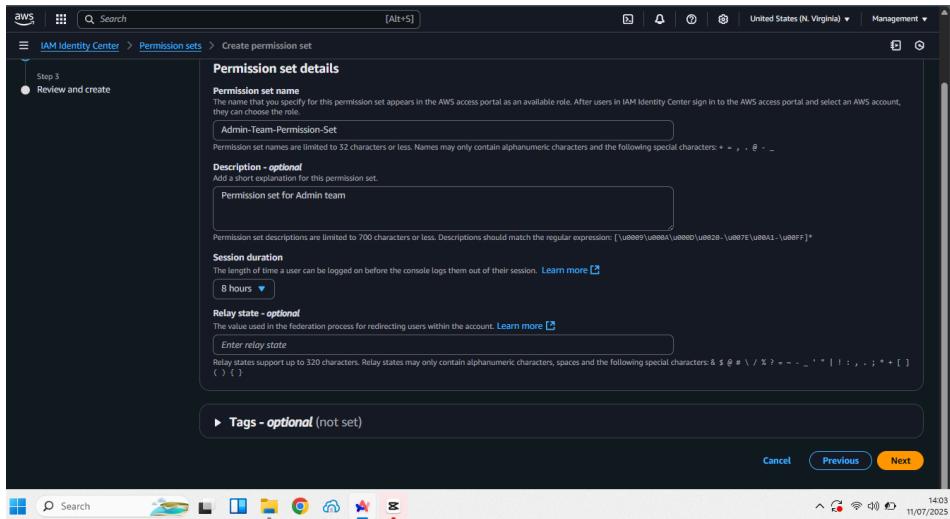


### 3.2 Create Admin Permission Set

#### Admin-Team Permission Set:

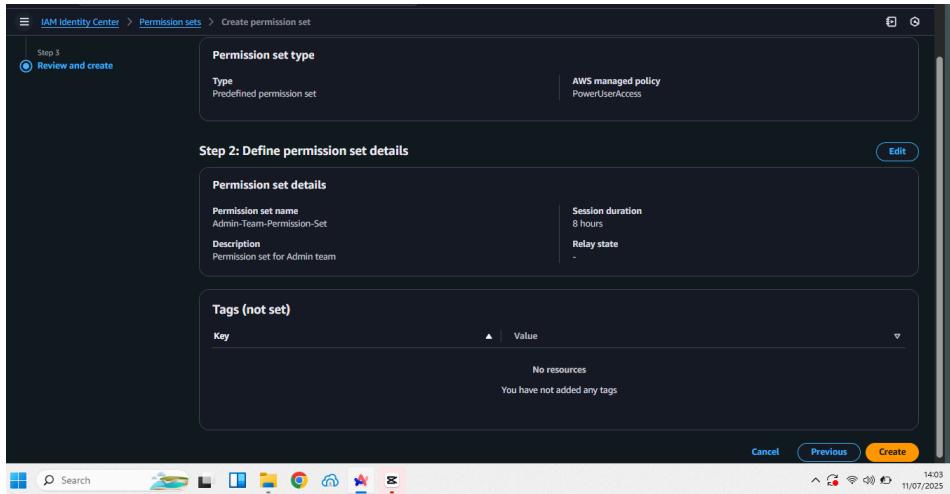
- Name: Admin-Team-Permission-Set

- **Description:** Full administrative access for Admin team members
- **Session duration:** 8 hours
- Click "Next"



#### Assign Policies:

- Select "AWS managed policies"
- Search and add: AdministratorAccess
- Click "Next" → "Create"



- I repeated the same steps to create additional permission sets for the other teams, similar to the screenshot for the Admin-Team-Permission-Set.

### 3.3 Create Additional Permission Sets

#### PowerUser Permission Set:

- Name: Power-User-Permission-Set

- Description: [Permission set for Power user Team](#)
- Session duration: [8 hours](#)
- Add policies: [PowerUserAccess](#)

### SystemAdmin Permission Set:

- Name: [Sysadmin-Permission-Set](#)
- Description: [Permission set for Sysadmin team](#)
- Session duration: [8 hours](#)
- Add policies: [SystemAdministrator](#)

### DataScientist Permission Set:

- Name: [Datascientist-Permission-Set](#)
- Description: [Permission set for data scientist team](#)
- Session duration: [8 hours](#)
- Add policies: [DataScientist](#)

## SCREEN SHOT OVERVIEW OF THE PERMISSION SET

Permission set	Description	ARN	Provisioned status
<a href="#">AdministratorAccess</a>	-	arnaws:permissionSet:ssoline-7223e8f00e7a686c/p-05e182c34db...	Provisioned
<a href="#">Sysadmin-Permission-Set</a>	Permission set for Sysadmin t...	arnaws:permissionSet:ssoline-7223e8f00e7a686c/p-874b78a96f...	Provisioned
<a href="#">Power-User-Permission-set</a>	Permission set for Power user...	arnaws:permissionSet:ssoline-7223e8f00e7a686c/p-6d003a4cfb5...	Provisioned
<a href="#">Admin-Team-Permission-Set</a>	Permission set for admin team	arnaws:permissionSet:ssoline-7223e8f00e7a686c/p-8d4b037e1dd...	Provisioned
<a href="#">Datascientist-Permission-Set</a>	Permission set for data scienti...	arnaws:permissionSet:ssoline-7223e8f00e7a686c/p-fcfc1cb7ad3dc...	Provisioned

In Task 3, the team defined standardized permission sets that establish the access boundaries for different user roles.

- Created four distinct permission sets (Admin-Team-Permission-Set, Power-User-Permission-set, Sysadmin-Permission-Set, and Datascientist-Permission-Set)[1]
- Configured appropriate session durations (8 hours) to balance security with user convenience[2]
- Established clear permission boundaries aligned with job functions across the organization[3]

These standardized permission sets create reusable access policies that can be consistently applied across accounts, ensuring appropriate access levels while maintaining security and compliance requirements.

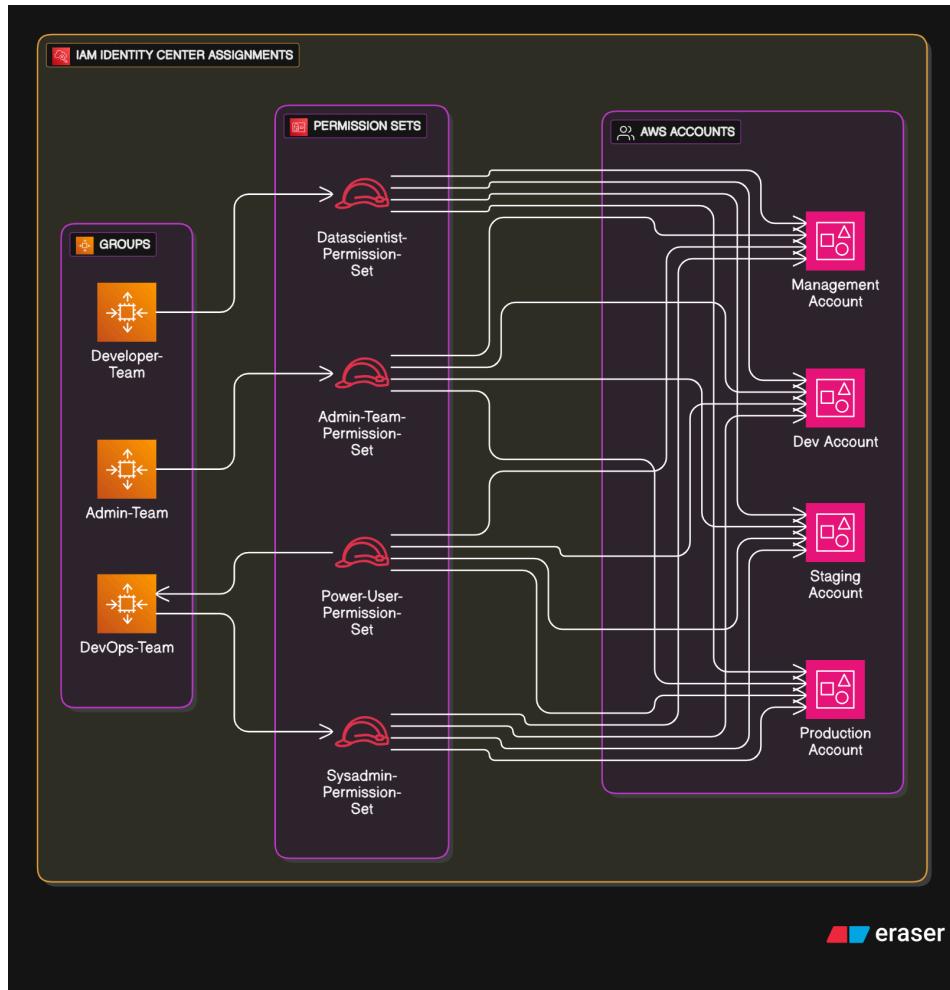
---



---

# TASK 4: Assign Permissions to Groups

## Overview Flowchart



### 4.1 Assigning appropriate permissions to each group using Permission-set policies to enable multi Account Role Switch

#### Navigate to AWS Accounts:

- In Identity Center, go to "AWS accounts"
- Selected my first account in the Development OU [Development OU](#)
- Click "Assign users or groups"

### Assign Admin-Team to Development Account:

- Click "Assign users or groups"
- Select "Groups" tab
- Choose "Admin-Team"
- Click "Next"

### Select Permission Set:

- Choose "Admin-Team-Permission-Set"
- Click "Next"
- Review and click "Submit"

**Select permission sets**

**Assign permission sets to "Saintic ORG Development"**

Permission sets define the level of access that users and groups in IAM Identity Center have to an AWS account. You can assign more than one permission set to a user. To ensure least privilege access to AWS accounts, users in IAM Identity Center with multiple permission sets on an AWS account must pick a specific permission set when selecting the account and then return to the AWS access portal to pick a different set when necessary. Learn more.

Permission set	Description	ARN
AdministratorAccess	-	arn:aws:sso::permissionSet/ssoins-7223e8f00e7a686c/ps-05e182c34dcb1fb69
Sysadmin-Permission-Set	Permission set for Sysadmin team	arn:aws:sso::permissionSet/ssoins-7223e8f00e7a686c/ps-074b78a9fb5bf728
Power-User-Permission-set	Permission set for Power user Team	arn:aws:sso::permissionSet/ssoins-7223e8f00e7a686c/ps-6d03a3acefc59491
<b>Admin-Team-Permission-Set</b>	Permission set for admin team	arn:aws:sso::permissionSet/ssoins-7223e8f00e7a686c/ps-8d4b037e1dddc3dc
DataScientist-Permission-Set	Permission set for data scientist team	arn:aws:sso::permissionSet/ssoins-7223e8f00e7a686c/ps-fce1cb7ad5dca4e5

- Then we review our choice and Clicked submit

**Review and submit**

**Review and submit assignments to "Saintic ORG Development"**

**Step 1: Select users and groups**

**Users and groups (1)**

Display name / group name	Type
Admin-Team	Group

**Step 2: Select permission sets**

**Permission sets (1)**

Permission set	Description	ARN	Creation time
Admin-Team-Permission-Set	Permission set for admin team	arn:aws:sso::permissionSet/ssoins-7223e8f00e7a686c/ps-8d4b037e1dddc3dc	18 hours ago

### Repeat for All Accounts:

- Assign Admin-Team to Management, Staging, and Production accounts
- Use the same Admin-Team-Permission-Set for all

### Assign DevOps-Team Permissions:

- For each account, assign DevOps-Team to:
  - Power-User-Permission-Set
  - Sysadmin-Permission-Set
- Repeat for all four accounts

### Assign Developer-Team Permissions:

- For each account, assign Developer-Team to:
  - Datascientist-Permission-Set
- Repeat for all four accounts

## Screenshot Overview of the Multi Account Permission

- Screen shot of the All team assigned to the development team with appropriate permission set

The screenshot shows the IAM Identity Center interface. The left sidebar is collapsed. The main area displays a table titled "Assigned users and groups (3)". The table has columns for "Username / group name", "Permissions sets", and "Type". Three groups are listed: "Admin-Team" (with permissions Admin-Team-Permission-Set, DataScientist-Permission-Set, Power-User-Permission-set), "DevOps-Team" (with the same permissions), and "Developer-Team" (with the same permissions). A "Change permission sets" button is at the top right of the table.

- Screen Shot of the all teams being assigned to all the accounts in the organization

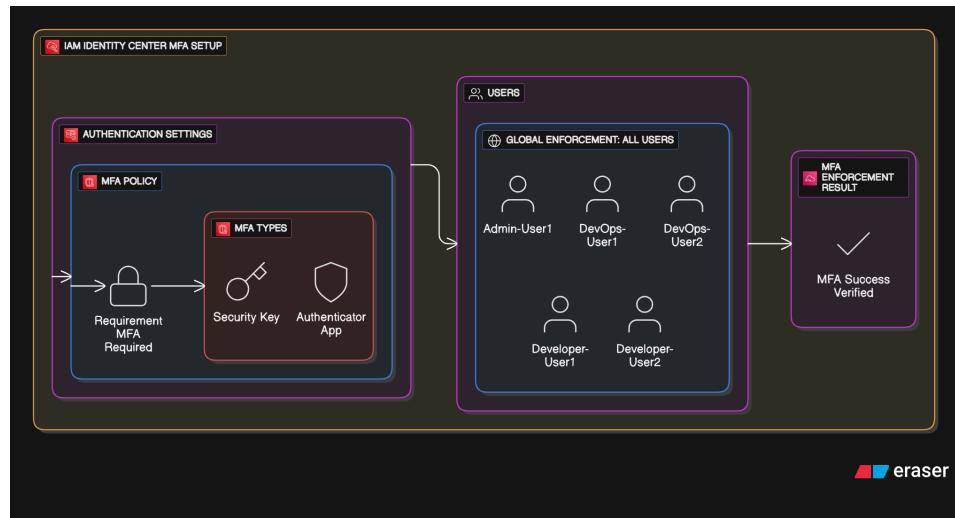
The screenshot shows the IAM Identity Center interface with the left sidebar expanded. The "Organizational structure" section on the left shows a hierarchy: Root > Development OU > Saintic ORG Development. The "Permission sets" section on the right shows three accounts under "Saintic ORG Development": "Chinazor Nwode" (management account) and "Saintic ORG Production", both with the same permission set (Admin-Team-Permission-Set, DataScientist-Permission-Set, Power-User-Permission-set). Other OUs like "Management OU" and "Production OU" also have their respective accounts assigned the same permission set.

**Task 4 Achievement:** In Task 4, the team implemented and validated the cross-account access strategy. Assigned team groups to appropriate AWS accounts with their corresponding permission sets, established access pathways across all organizational accounts for each team, tested the entire implementation through user login validation, verified MFA enforcement for enhanced security, and confirmed appropriate access levels and permission set functionality.

---

## TASK 5: Multi-Factor Authentication Setup

### Overview Flowchart



### 5.1 Configure MFA Settings

#### Navigate to Authentication Settings:

- In Identity Center, go to "Settings"
- Click "Authentication" tab

## Configure MFA Policy:

- Under "Multi-factor authentication"

- Select "Users must provide a second factor to sign in"
- Choose "Authenticator apps" and "Security keys"

## Apply Settings:

- Click "Save changes"
- Verify MFA enforcement is active

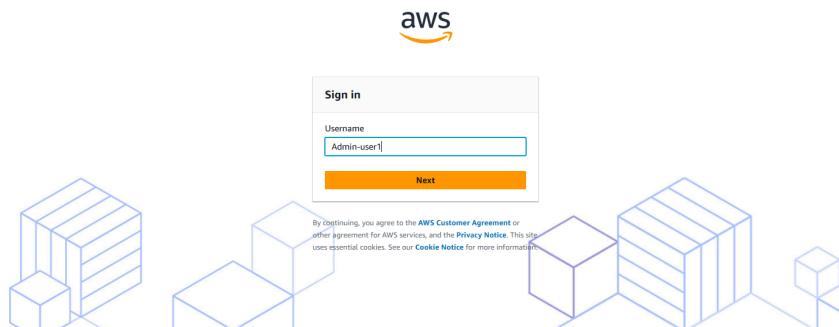
Users (6)					
	Username	Display name	Status	MFA devices	Created by
<input type="checkbox"/>	Developer-user2	Developer User 2	<span>Enabled</span>	None	Manual
<input type="checkbox"/>	Developer-user1	Developer User 1	<span>Enabled</span>	None	Manual
<input type="checkbox"/>	admin	Chinazor Nwode	<span>Enabled</span>	None	Manual
<input type="checkbox"/>	DevOps-user1	DevOps user 1	<span>Enabled</span>	None	Manual
<input type="checkbox"/>	Admin-user1	Admin User 1	<span>Enabled</span>	None	Manual
<input type="checkbox"/>	DevOps-user2	DevOps User 2	<span>Enabled</span>	None	Manual

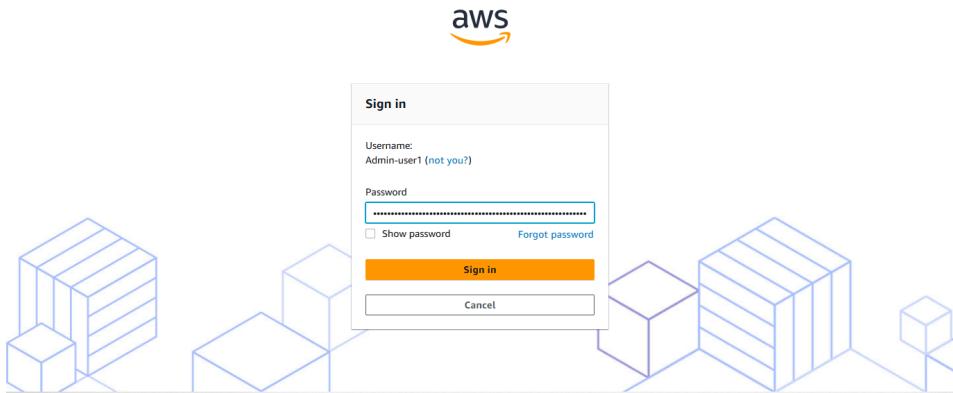
Users (6)					
	Username	Display name	Status	MFA devices	Created by
<input type="checkbox"/>	Developer-user2	Developer User 2	<span>Enabled</span>	None	Manual
<input type="checkbox"/>	Developer-user1	Developer User 1	<span>Enabled</span>	None	Manual
<input type="checkbox"/>	admin	Chinazor Nwode	<span>Enabled</span>	1 device	Manual
<input type="checkbox"/>	DevOps-user1	DevOps user 1	<span>Enabled</span>	1 device	Manual
<input type="checkbox"/>	Admin-user1	Admin User 1	<span>Enabled</span>	1 device	Manual
<input type="checkbox"/>	DevOps-user2	DevOps User 2	<span>Enabled</span>	None	Manual

## 5.2 Test MFA with All User Types

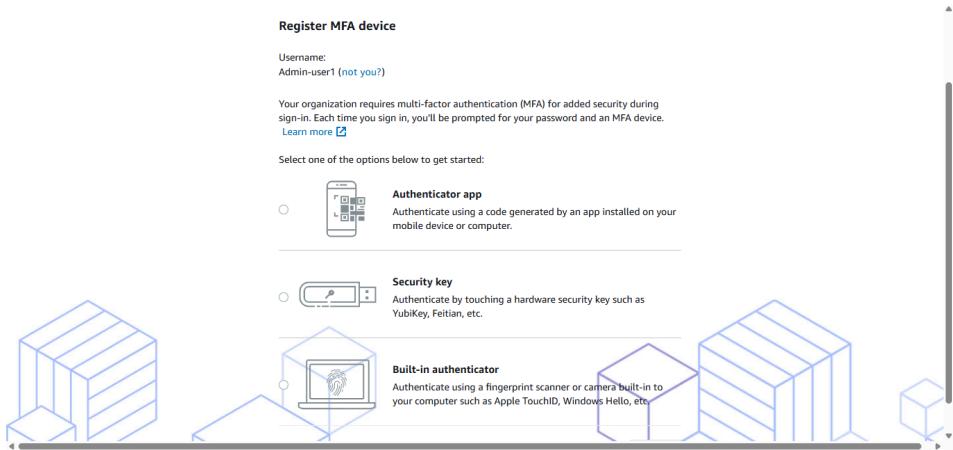
### Test Admin-User1 MFA:

- Login with **Admin-User1** credentials

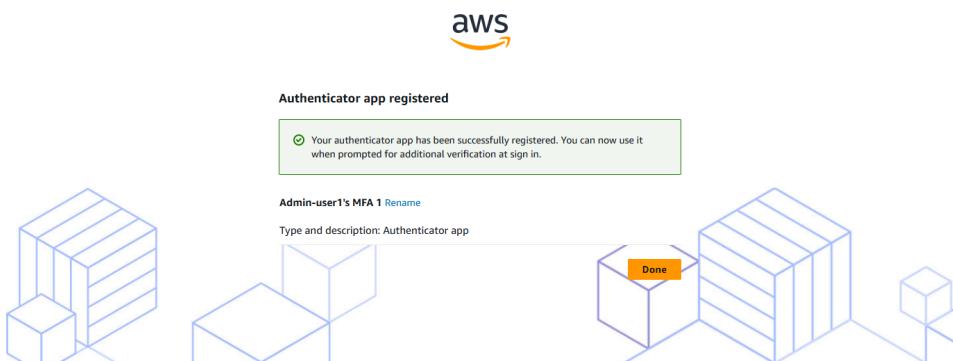




- Complete MFA setup process



- Verify successful authentication



- We were logged into the dashboard, and saw the accounts which we were assigned permission to switch role on.

AWS access portal

Accounts Applications

AWS accounts (4)

Filter accounts by name, ID, or email address

Create shortcut

Feedback ©2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Terms Cookie Preferences

Username	Display name	Status	MFA devices	Created by
Developer-user2	Developer User 2	Enabled	None	Manual
Developer-user1	Developer User 1	Enabled	None	Manual
admin	Chinazor Nwode	Enabled	1 device	Manual
DevOps-user1	DevOps user 1	Enabled	1 device	Manual
Admin-user1	Admin User 1	Enabled	1 device	Manual
DevOps-user2	DevOps User 2	Enabled	None	Manual

**Task 5 Achievement:** Successfully configured and enforced Multi-Factor Authentication across all user accounts, ensuring enhanced security posture for the organization. All users now require both password and MFA token for authentication, significantly reducing the risk of unauthorized access.

IAM Identity Center

Managing instance

Users Groups Settings

Multi-account permissions AWS accounts Permission sets Application assignments Applications

Related consoles CloudTrail Recommended AWS Organizations IAM

CloudShell Feedback

Users (6)

Users listed here can sign in to the AWS access portal to access AWS accounts and assigned cloud applications. Learn more

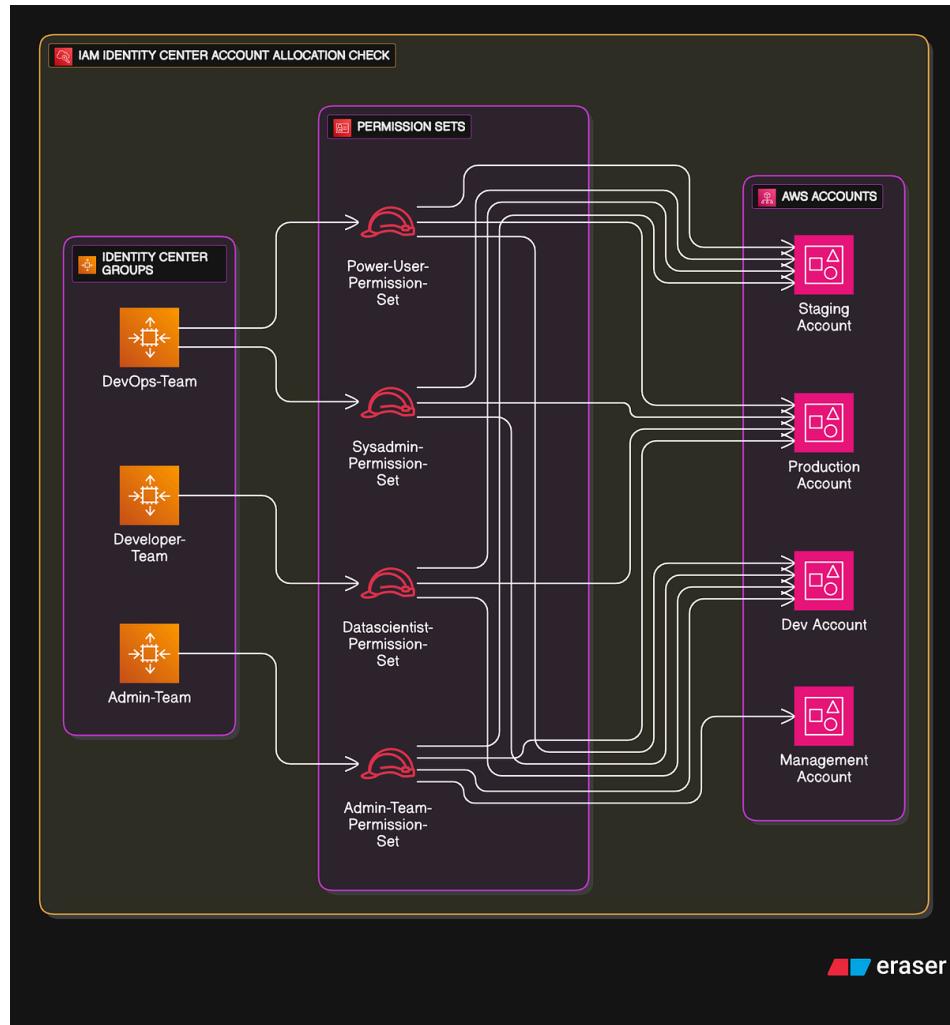
Username Find users

Username	Display name	Status	MFA devices	Created by
Developer-user2	Developer User 2	Enabled	None	Manual
Developer-user1	Developer User 1	Enabled	None	Manual
admin	Chinazor Nwode	Enabled	1 device	Manual
DevOps-user1	DevOps user 1	Enabled	1 device	Manual
Admin-user1	Admin User 1	Enabled	1 device	Manual
DevOps-user2	DevOps User 2	Enabled	None	Manual

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## TASK 6 : Account Allocation Check

### Overview Flowchart



## 6.1 Check Account Allocations

- Admin-Team Allocation

The screenshot shows the AWS IAM Identity Center Groups page for the 'Admin-Team' group. The 'AWS accounts' tab is selected under the 'AWS account access' section. A red box highlights the list of AWS accounts assigned to the group:

- Saintic ORG Production (78499176155) chinazormwde-SainticOrgProduction@gmail.com
- Saintic ORG Staging (50403590568) chinazormwde-SainticOrgStaging@gmail.com
- Saintic ORG Development (808872802004) chinazormwde-SainticOrgDevelopment@gmail.com
- Chinazor Nwode (management account) (756998121792) chinazormwde@gmail.com

A note at the bottom right says: "Choose an account to view permission sets and policies granting Admin-Team access to that account."

- DevOPs-Team Allocation

The screenshot shows the AWS IAM Identity Center interface. The left sidebar shows 'Managing instance ssolms-7225ebff00e7a68fc'. The main area is titled 'General Information' for the 'DevOPs-Team' group. Under 'AWS account access (4)', there is a search bar and a list of four AWS accounts, each with a radio button and an email address. A red box highlights this list.

- Developer-Team Allocation

The screenshot shows the AWS IAM Identity Center interface. The left sidebar shows 'Managing instance ssolms-7225ebff00e7a68fc'. The main area is titled 'General Information' for the 'Developer-Team' group. Under 'AWS account access (4)', there is a search bar and a list of four AWS accounts, each with a radio button and an email address. A red box highlights this list.

## 6.2 Check Account Permission Set

- OVERVIEW OF THE PERMISSION SET

The screenshot shows the AWS IAM Identity Center interface. The left sidebar shows 'Managing instance ssolms-7225ebff00e7a68fc'. The main area lists three groups: 'Developer-Team', 'DevOps-Team', and 'Admin-Team'. Each group has a list of permission sets assigned to it, indicated by a small circle icon. The permission sets listed are: Admin-Team-Permission-Set, DataScientist-Permission-Set, Power-User-Permission-set, and 1 more for Developer-Team; Admin-Team-Permission-Set, DataScientist-Permission-Set, Power-User-Permission-set, and 1 more for DevOps-Team; and Admin-Team-Permission-Set for Admin-Team.

**Assigned users and groups (4)**

The following users and groups in IAM Identity Center can select this AWS account from within their AWS access portal. [Learn more](#)

Username / group name	Permission sets	Type
DevOps-Team	• Power-User-Permission-set • Sysadmin-Permission-Set	Group
Admin-Team	• Admin-Team-Permission-Set	Group
Developer-Team	• DataScientist-Permission-Set	Group

**Assigned users and groups (5)**

The following users and groups in IAM Identity Center can select this AWS account from within their AWS access portal. [Learn more](#)

Username / group name	Permission sets	Type
DevOps-Team	• Power-User-Permission-set • Sysadmin-Permission-Set	Group
Admin-Team	• AdministratorAccess	Group
Developer-Team	• DataScientist-Permission-Set	Group

**Assigned users and groups (3)**

The following users and groups in IAM Identity Center can select this AWS account from within their AWS access portal. [Learn more](#)

Username / group name	Permission sets	Type
Admin-Team	• Admin-Team-Permission-Set	Group
DevOps-Team	• Power-User-Permission-set • Sysadmin-Permission-Set	Group
Developer-Team	• DataScientist-Permission-Set	Group

**AWS accounts**

**Organization o-exjyapdepb**

Select one or more AWS accounts in your organization to provide multi-account access to users and groups in IAM Identity Center. [Learn more](#)

Search by name, email, account ID or OU ID.	Hierarchy	List
Organizational structure	Permission sets	
Root		
Development OU	• Admin-Team-Permission-Set • DataScientist-Permission-Set • Power-User-Permission-set • 1 more	
Saintic ORG Development		
Management OU	• Admin-Team-Permission-Set • AdministratorAccess • DataScientist-Permission-Set	
Chinazor Nwode		

The screenshot shows the AWS Organizations console interface. It displays three Organizational Units (OU) under the root account:

- Management OU**: Contains a user account named "Chinazor Nwode" (management account). This account has the following permission sets assigned:
  - Admin-Team-Permission-Set
  - Administrator/Access
  - DataScientist-Permission-Set
  - Power-User-Permission-Set
  - 2 more
- Production OU**: Contains a user account named "Saintic ORG Production". This account has the following permission sets assigned:
  - Administrator/Access
  - DataScientist-Permission-Set
  - Power-User-Permission-Set
  - 1 more
- Staging OU**: Contains a user account named "Saintic ORG Staging". This account has the following permission sets assigned:
  - Admin-Team-Permission-Set
  - DataScientist-Permission-Set
  - Power-User-Permission-Set
  - 1 more

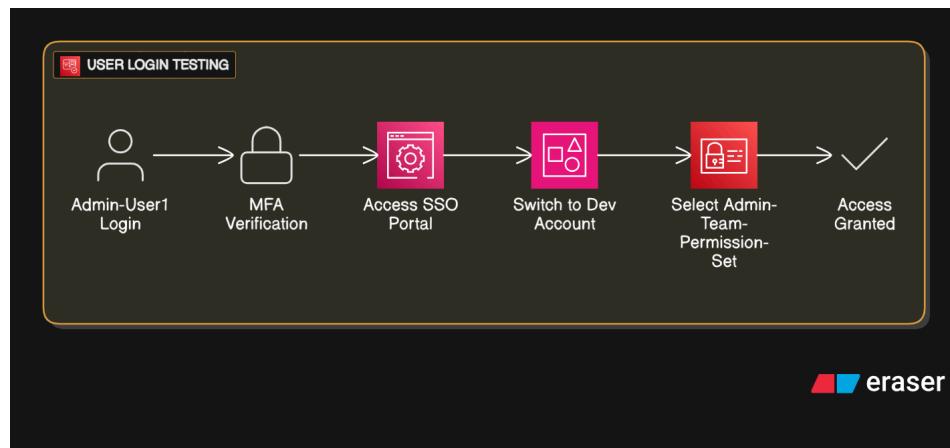
A notification badge in the bottom right corner indicates 151 unread items.

**Task 6 Achievement:** Successfully checked and validated the complete user Permission set for all user Teams Group in their various AWS Account in the organization, confirming that all permissions needed for each of the groups are assigned.

---

## TASK 7: User Login Testing and Account Switching

### Overview Flowchart



### 7.1 Complete User Login Testing

#### Test Admin-User1 Complete Flow:

- Navigate to SSO portal
- Enter username and password (first authentication)

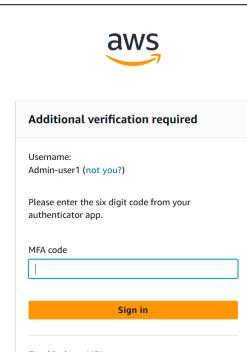


The screenshot shows the first step of the AWS sign-in process. It features the AWS logo at the top. Below it is a "Sign in" form with a "Username" field containing "Admin-user1". A "Next" button is at the bottom of the form. A small note at the bottom states: "By continuing, you agree to the [AWS Customer Agreement](#) or other agreement for AWS services, and the [Privacy Notice](#). This site uses essential cookies. See our [Cookie Notice](#) for more information."



The screenshot shows the second step of the AWS sign-in process. It features the AWS logo at the top. Below it is a "Sign in" form with a "Username" field containing "Admin-user1 (not you?)". It includes a "Password" field with a masked password, a "Show password" link, a "Forgot password" link, a "Sign in" button, and a "Cancel" button.

- Enter MFA code (second authentication)



The screenshot shows the third step of the AWS sign-in process. It features the AWS logo at the top. Below it is an "Additional verification required" form. It displays the same "Username" field as the previous step. It includes a note: "Please enter the six digit code from your authenticator app." A "MFA code" input field contains a single character, and a "Sign in" button is at the bottom.

- Access dashboard and switch accounts
- So we click on the drop down button on the "Saintic ORG Development"

The screenshot shows the AWS Access Portal interface. At the top, there's a navigation bar with the AWS logo and a search bar. Below it, the main title is "AWS access portal". Underneath, there are tabs for "Accounts" and "Applications", with "Accounts" being the active tab. A sub-section titled "AWS accounts (4)" is displayed, showing a list of accounts with their names, IDs, and email addresses. The "Admin" button is located in the top right corner of this section. At the bottom of the page, there are links for "Feedback", "©2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.", "Privacy", "Terms", and "Cookie Preferences".

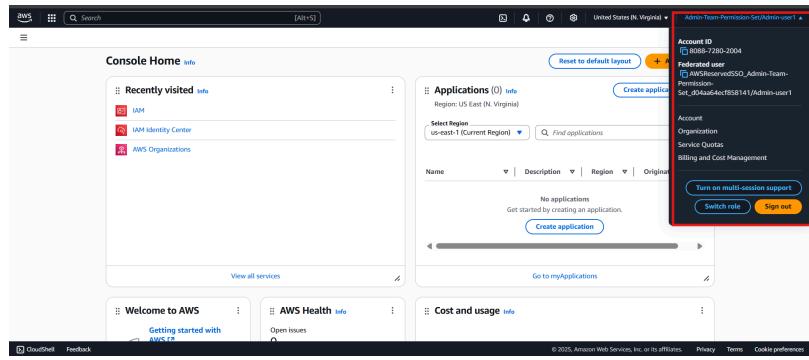
- So we click the "**Admin-Team-Permission-Set**" button on the "**Saintic ORG Development**"

This screenshot is similar to the previous one, showing the AWS Access Portal's "Accounts" section. The "Admin-Team-Permission-Set" button is now highlighted with a red box around the "Saintic ORG Development" account entry. The other accounts (Chinazor Nwode, Saintic ORG Production, and Saintic ORG Staging) are also listed below it. The "Admin" button remains in the top right corner.

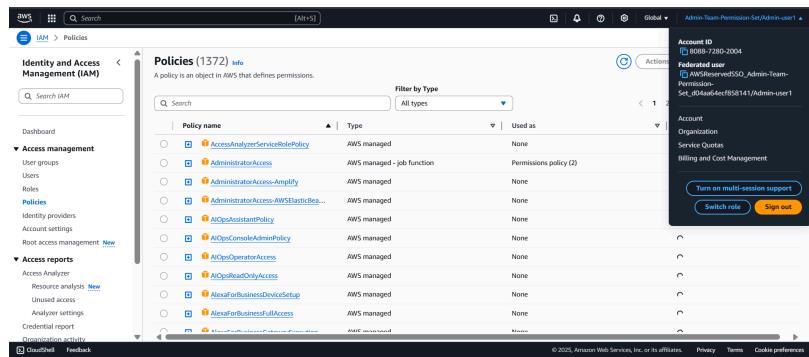
- The Link took us to a new page, which means we now logged as admin in the development account

This screenshot shows the AWS Console Home page. The top navigation bar includes the AWS logo, a search bar, and the URL "Europe (Stockholm)". The title "Console Home" is at the top left, and the user "Admin-Team-Permission-Set/Admin-user1" is shown at the top right. On the left, there's a "Recently visited" section with a message "No recently visited services" and links to EC2, S3, Aurora, RDS, and Lambda. On the right, there's an "Applications" section with a message "No applications. Get started by creating an application." and a "Create application" button. The bottom of the page includes links for "CloudShell", "Feedback", and the standard copyright and privacy information.

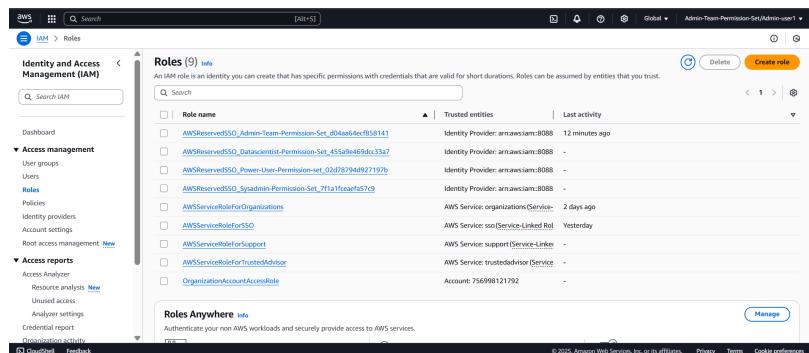
- Lets Verify this



- Checked our Account Policies



- Checked our Account Role



## Question: why is there this permissions in the admin account?

ANSWER: We made a mistake in permission set earlier by assigning permission that was not meant for the admin which is unnecessary needed in this context so removed it but was still showing due AWS is kind of keeping the logs or history of whatsoever happened so that when there is an Audit. They can know what happened and when it happened

<input type="checkbox"/> <a href="#">AWSReservedSSO_DataScientist-Permission-Set_455a9e469dccc53a7</a>	Identity Provider: arn:aws:iam:8088 -
<input type="checkbox"/> <a href="#">AWSReservedSSO_Power-User-Permission-set_02d78794d927197b</a>	Identity Provider: arn:aws:iam:8088 -
<input type="checkbox"/> <a href="#">AWSReservedSSO_Sysadmin-Permission-Set_f71a1fceaf857c9</a>	Identity Provider: arn:aws:iam:8088 -

## 7.2 Account Switching Documentation

### SSO Portal Interface:

- Document the single sign-on experience
- Show account switching capabilities
- Demonstrate permission set selection

The image contains three screenshots of the AWS SSO portal interface, illustrating the account switching process and permission set selection.

- Screenshot 1: AWS accounts page**  
Shows the "AWS accounts" section with four accounts listed:
  - Chinazor Nwode (selected, highlighted with a red box)
  - Saintic ORG Development
  - Saintic ORG Production
  - Saintic ORG StagingEach account entry includes a "Admin-Team-Permission-Set" link, which is also highlighted with a red box.
- Screenshot 2: Console Home - Europe (Stockholm)**  
Shows the "Applications" section with zero applications listed. A modal window is open over the applications list, titled "Admin-Team-Permission-Set/Admin-user1". It displays the following information:
  - Account ID: 08088-7280-2004
  - Federated user: AWSIdentitySSO\_Admin-Team-Permission-Set\_ufb4a4decf858141/Admin-user1
  - Account: United States (N. Virginia)
  - Organization: Organization
  - Service Quotas: Billing and Cost ManagementButtons at the bottom of the modal include "Turn on multi-session support", "Switch role", and "Sign out".
- Screenshot 3: Console Home - United States (N. Virginia)**  
Shows the "Applications" section with zero applications listed. A modal window is open over the applications list, titled "Admin-Team-Permission-Set/Admin-user1". It displays the following information:
  - Account ID: 08088-7280-2004
  - Federated user: AWSIdentitySSO\_Admin-Team-Permission-Set\_ufb4a4decf858141/Admin-user1
  - Account: United States (N. Virginia)
  - Organization: Organization
  - Service Quotas: Billing and Cost ManagementButtons at the bottom of the modal include "Turn on multi-session support", "Switch role", and "Sign out".

The screenshot shows the AWS IAM Policies list page. The left sidebar includes links for Identity and Access Management (IAM), Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management), and Access reports (Resource analysis, Unused access, Analyzer settings, Credential report, Organization activity). The main content area displays 137 policies, with a search bar and filter options for Policy name, Type (AWS managed, Customer managed), and Used as (None, Permissions policy). A modal window on the right provides details for a selected policy, including its ARN, creation date, and associated users.

Policy name	Type	Used as
AccessAnalyzerServiceRolePolicy	AWS managed	None
AdministratorAccess	AWS managed - job function	Permissions policy (2)
AdministratorAccess-Amplify	AWS managed	None
AdministratorAccess-AWSElasticBea...	AWS managed	None
AdministratorAccessPolicy	AWS managed	None
AIOpsConsoleAdminPolicy	AWS managed	None
AIOpsOperatorAccess	AWS managed	None
AIOpsReadOnlyAccess	AWS managed	None
AlexaForBusinessDeviceSetup	AWS managed	None
AlexaForBusinessFullAccess	AWS managed	None

The screenshot shows the AWS IAM Roles page. At the top, there's a search bar and a global navigation bar with the text "Admin-Team-Permission-Set/Admin-control". On the left, there's a sidebar with "Identity and Access Management (IAM)" selected. The main content area has a header "Roles (9) Info" and a sub-header "An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust." Below this is a search bar and a table with the following data:

Role name	Trusted entities	Last activity
AWSReservedSSO_Admin-Team-Permission_Set_d04aa54ecf83b141	Identity Provider: arnawsiam:arn:awsiam:8088	12 minutes ago
AWSReservedSSO_DataScientist-Permission_Set_d55a646dc53a7	Identity Provider: arnawsiam:arn:awsiam:8088	-
AWSReservedSSO_Power-User-Permission_Set_02d78794ad27197b	Identity Provider: arnawsiam:arn:awsiam:8088	-
AWSReservedSSO_Syndicate-Permission_Set_7f11a7f5eaf5f529	Identity Provider: arnawsiam:arn:awsiam:8088	-
AWSServiceRoleForOrganizations	AWS Service: organizations/Service-Linked Role	2 days ago
AWSServiceRoleForSSO	AWS Service: sso/Service-Linked Role	Yesterday
AWSServiceRoleForSupport	AWS Service: support/Service-Linked Role	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor/Service-Linked Role	-
OrganizationAccountAccessRole	Account: 756098121792	-

At the bottom, there's a "Roles Anywhere" section with a "Manage" button.

**Task 7 Achievement:** Successfully tested and validated the complete user experience for all user types, confirming that authentication flows, MFA enforcement, and account switching work seamlessly across the organization.

## **TASK 8: Documentation and Final Validation**

## Overview Flowchart

flowchart TD  
A[Compile Documentation] --> B[Organize Screenshots]  
B --> C[Create Final Report]  
C --> D[Validate All Requirements]  
D --> E[Prepare Submission]  
E --> F[Project Complete]

style A fill:#e1f5fe  
style F fill:#c8e6c9

## 8.1 Documentation Checklist

## Screenshots Completed:

- ✓ AWS Organizations dashboard with all accounts

- Organizational Units structure
- Identity Center users and groups
- All permission sets created
- Permission assignments for each account
- MFA configuration and testing
- Complete login flows for all user types
- Account switching demonstrations
- Cross-account access verification

## 8.2 Final Validation

### Requirements Verification:

- Organization Setup:** 4 AWS accounts (Management, Dev, Staging, Production)
- User Management:** 5 users distributed across 3 groups
- Permission Management:** 4 permission sets properly configured
- Access Control:** All groups have appropriate permissions across all accounts
- Security:** MFA enabled and working for all users
- Functionality:** Account switching works seamlessly
- Documentation:** Complete screenshot documentation of all processes

## Lessons Learned

- **Strategic Value of Multi-Account Architecture:** The team discovered that a well-structured multi-account strategy doesn't just improve security, but also creates clearer ownership boundaries and simplifies cost allocation across development, staging, and production environments.
- **Proactive Security Through Identity Federation:** Beyond simply implementing MFA, the team learned that AWS Identity Center creates a centralized authentication point that significantly reduces credential management overhead and security risks compared to managing multiple IAM users across accounts.
- **Permission Templating for Scalability:** The creation of standardized permission sets revealed that defining access patterns once and deploying them consistently across accounts dramatically improves governance and reduces the risk of permission drift over time.
- **Cross-Account Access Workflow Optimization:** The team gained practical experience in balancing security with usability by establishing seamless role-switching capabilities that maintain strict security boundaries while providing a friction-free user experience.
- **Documentation as Risk Mitigation:** The detailed documentation process wasn't just for knowledge transfer - it created an auditable trail that reduces organizational risk by ensuring configurations can be replicated, troubleshooted, or validated against compliance requirements.
- **Naming Convention Discipline:** The team discovered that consistent naming conventions across accounts, groups, and permission sets significantly reduced operational complexity and created a more intuitive user experience for both administrators and end users.
- **IAM Trust Relationship Criticality:** The team gained deeper appreciation for how IAM trust relationships form the foundation of cross-account access, and how small misconfiguration details can completely break otherwise well-designed security architectures.

This project was completed by the dedicated team members who demonstrated exceptional technical skills and collaborative spirit throughout the implementation.

**Thank you for your attention to this comprehensive AWS Organization project.**

| Signed by *SainT*

