😊 Integrating Azure AD with AWS

Integrating Azure Active Directory (Azure AD) with AWS allows you to enable single sign-on (SSO) and federated authentication for your AWS accounts. This integration provides a seamless login experience for users and centralized management of access policies.

😄 Use Cases for Integrating Azure AD with AWS

1. **Centralized User Management**: Organizations can manage user access to AWS resources through Azure AD, leveraging existing user identities and groups. This simplifies user provisioning and de-provisioning, ensuring that only authorized users have access to AWS services.
2. **Single Sign-On (SSO)**: Users can log in to AWS using their Azure AD credentials. This provides a seamless login experience, reducing the need to remember multiple passwords and improving user productivity.
3. **Federated Authentication**: Integrating Azure AD with AWS allows for federated authentication, enabling users to authenticate against Azure AD and obtain temporary security credentials to access AWS resources.
4. **Enhanced Security**: By leveraging Azure AD's security features such as Multi-Factor Authentication (MFA), Conditional Access Policies, and Identity Protection, organizations can enhance the security of their AWS environments.
5. **Compliance and Governance**: Centralizing access management through Azure AD helps organizations maintain compliance with various regulatory requirements by providing consistent access control and detailed audit logs.
6. **Role-Based Access Control (RBAC)**: Azure AD groups can be mapped to AWS IAM roles, allowing for granular access control. This ensures users have the appropriate level of access based on their role within the organization.
7. **Improved User Experience**: By providing a unified login experience across multiple cloud services, organizations can improve the user experience and reduce login-related issues.

😄 Benefits of Integrating Azure AD with AWS

1. **Streamlined Access Management**: Centralizing access management through Azure AD simplifies the administration of user identities and access permissions, reducing the administrative overhead associated with managing multiple identity providers.
2. **Enhanced Security Posture**: Leveraging Azure AD's advanced security features, such as MFA and Conditional Access, helps to protect AWS resources from unauthorized access and potential security threats.
3. **Operational Efficiency**: Integrating Azure AD with AWS allows for the automation of user provisioning and de-provisioning, ensuring that access is granted and revoked in a

timely manner. This reduces the risk of orphaned accounts and potential security vulnerabilities.

4. **Cost Savings**: By consolidating identity management and access control into a single platform, organizations can reduce costs associated with maintaining multiple identity solutions and streamline their IT operations.
5. **Consistent User Experience**: Providing a consistent login experience across different cloud platforms improves user satisfaction and reduces the likelihood of login-related issues.
6. **Compliance and Auditability**: Centralizing access management through Azure AD ensures that access controls are consistently applied and auditable, helping organizations meet compliance requirements and maintain detailed records of user activity.
7. **Scalability**: Azure AD is designed to handle large-scale enterprise environments, making it suitable for organizations of all sizes. This scalability ensures that the identity and access management solution can grow with the organization's needs.
8. **Simplified Onboarding and Offboarding**: With Azure AD integration, onboarding new employees and offboarding departing employees becomes more efficient, as access to AWS resources can be managed through existing identity management workflows.

🤔 What are we doing in this lab?

This process outlines the steps to set up Single Sign-On (SSO) and automatic user provisioning between Azure Active Directory (Azure AD) and AWS IAM Identity Center. Here's a summary of what you're doing and the end goal:

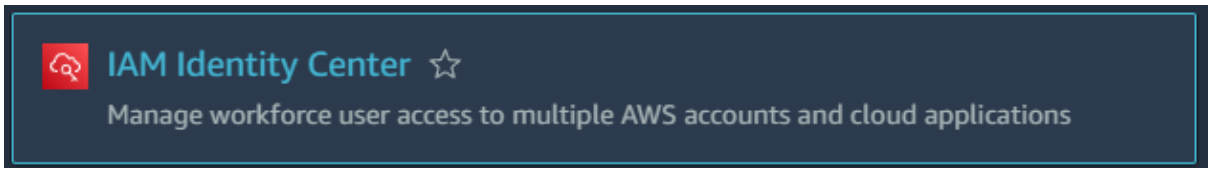
1. **Configure SSO Integration**: You're setting up SSO by integrating Azure AD with AWS IAM Identity Center. This involves enabling IAM Identity Center in AWS, configuring SAML-based authentication, and uploading necessary metadata files from both Azure and AWS.
2. **Set Up Automatic Provisioning**: You're configuring automatic user provisioning from Azure AD to AWS. This includes providing the SCIM endpoint and access token in Azure to automate the creation and management of AWS users based on Azure AD.
3. **Add and Manage Users**: You add users to Azure AD and ensure they are automatically provisioned into AWS IAM Identity Center. You then assign permissions to these users within AWS to grant appropriate access.
4. **Verify and Test**: Finally, you test the integration by logging in to the AWS Access Portal using Azure AD credentials to confirm that everything is working correctly and that users have the expected access.

End Goal: In simple terms, the end goal is to make it easier for users to log in to AWS using their existing Azure AD credentials, and to automatically manage their access and permissions in AWS based on their Azure AD account. This means users don't need to remember a new set

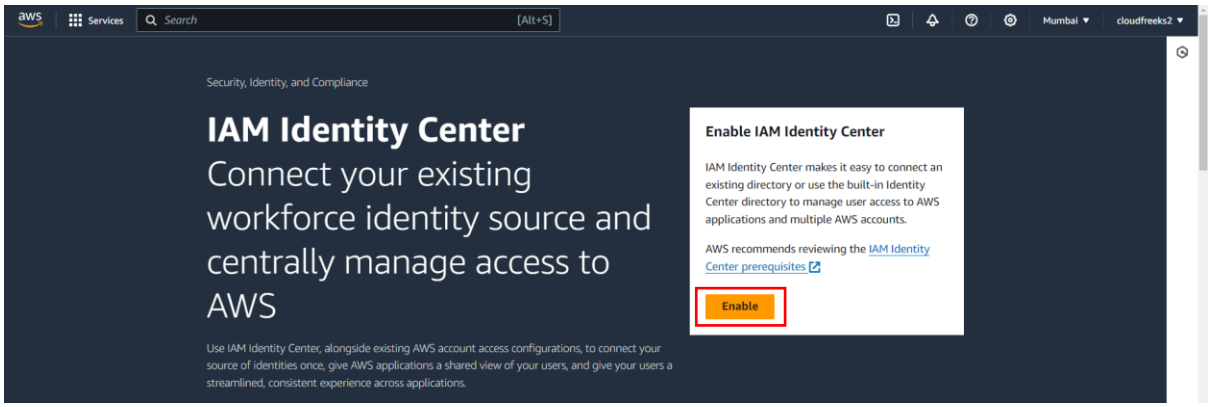
of login details, and their access rights in AWS are automatically kept up-to-date based on what's set in Azure AD.

😄 To begin with the Lab:

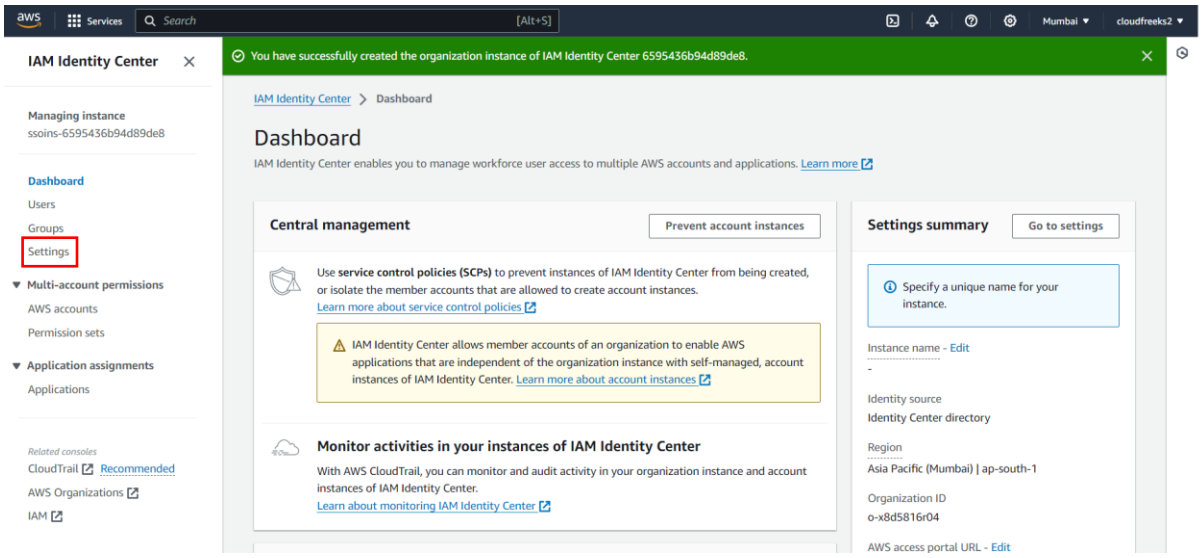
1. Login to AWS Console and navigate to IAM Identity Centre. Choose the service accordingly.



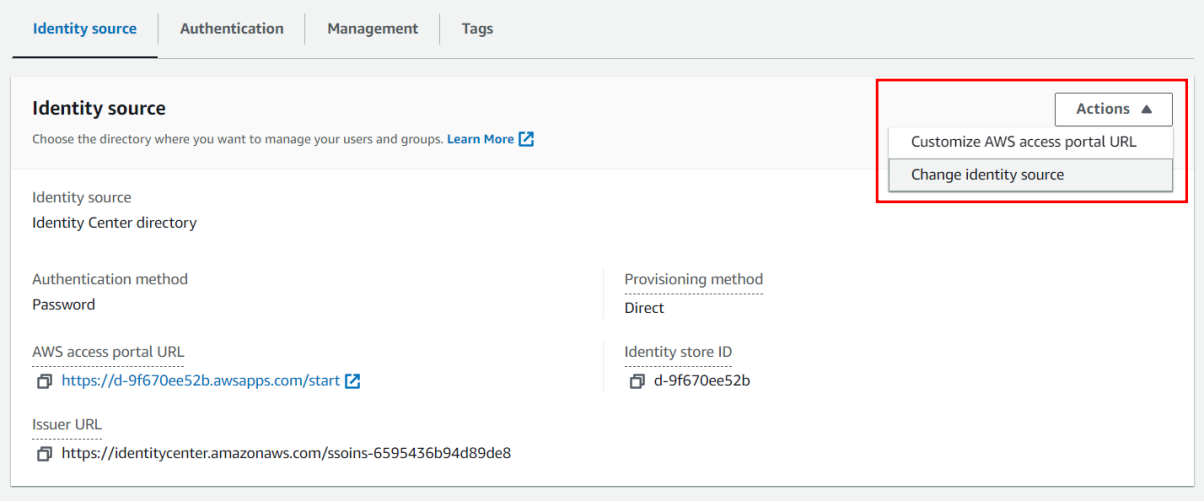
2. Now from its dashboard click on Enable.



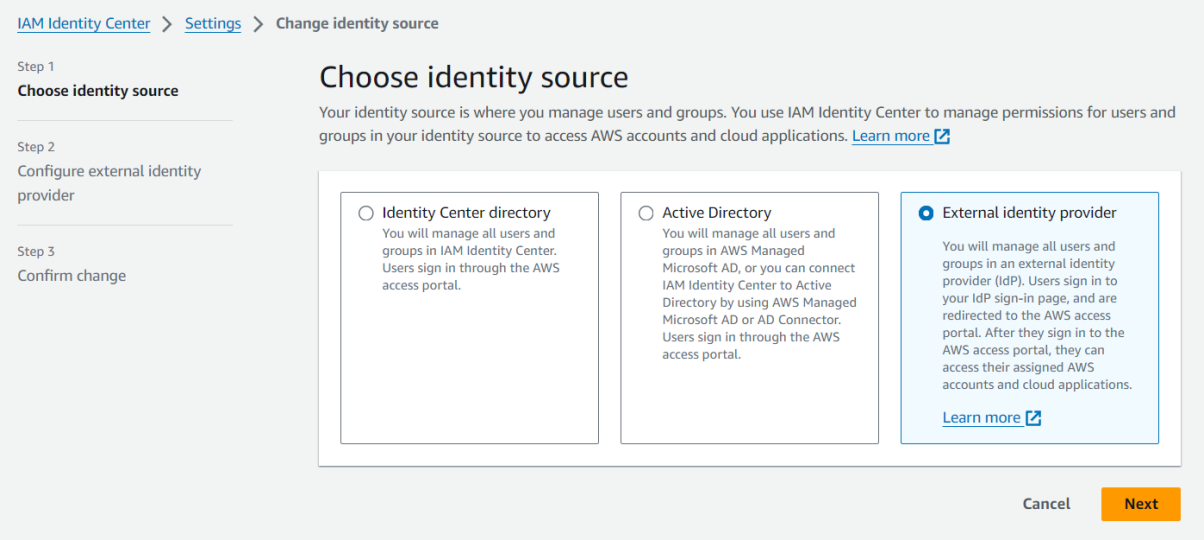
3. Once it is enabled you need to click on settings, highlighted below.



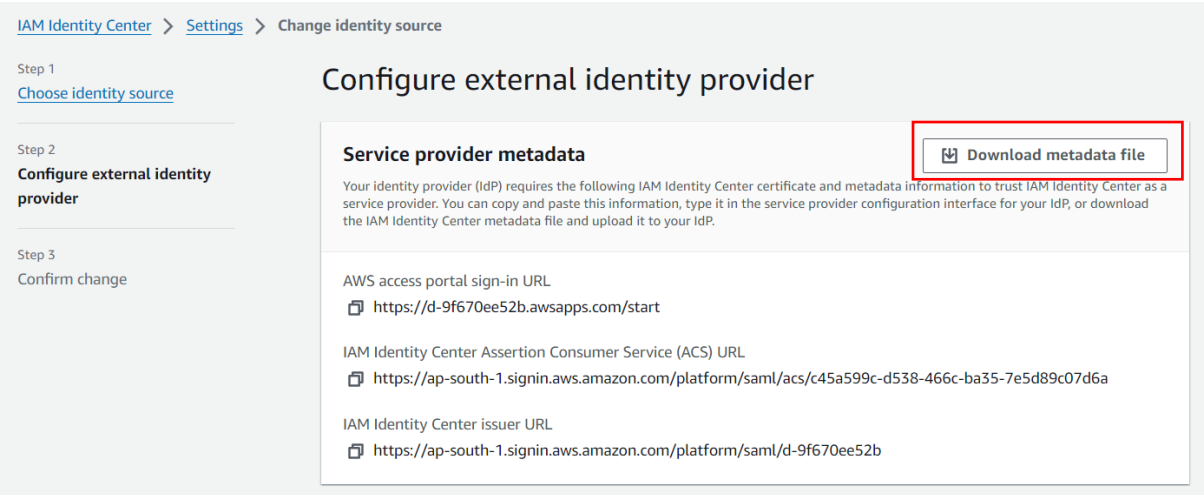
4. Then in the settings you need to scroll down and you will see the Identity source. So, click on Actions and choose change identity source.



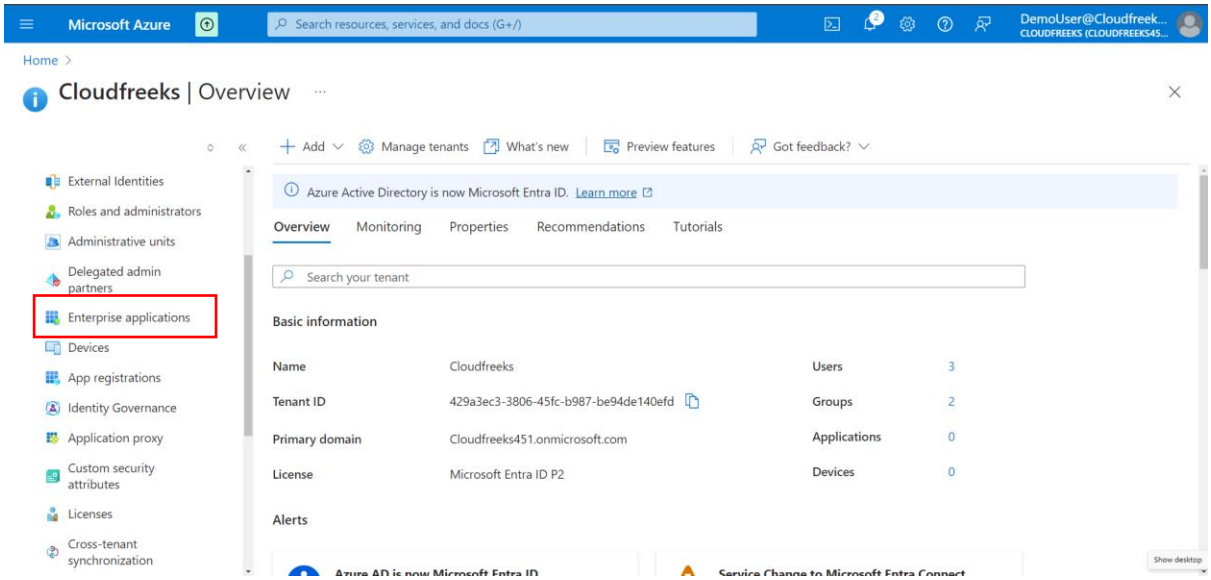
5. Now you need to choose External identity provider and click on next.



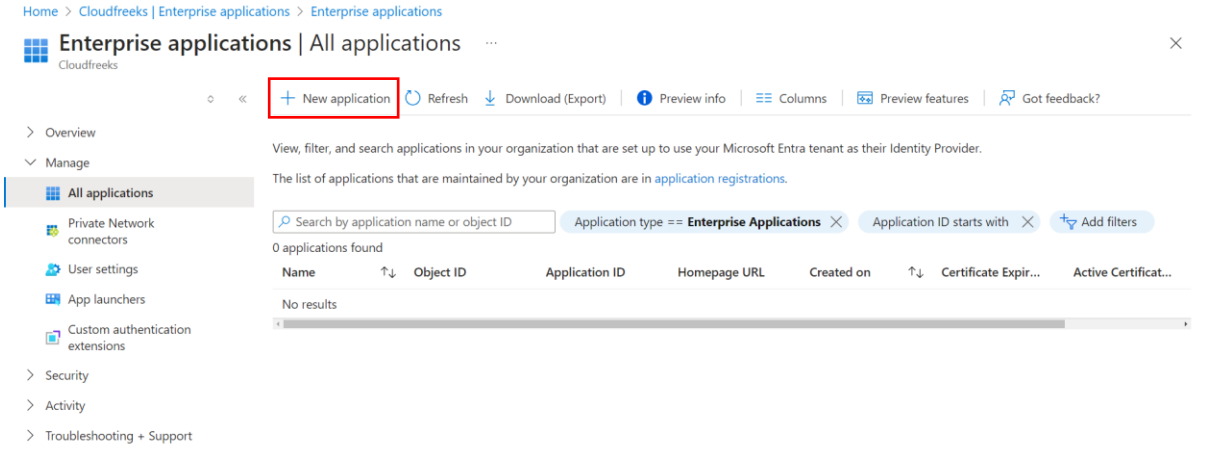
6. Now you have to download the metadata file.



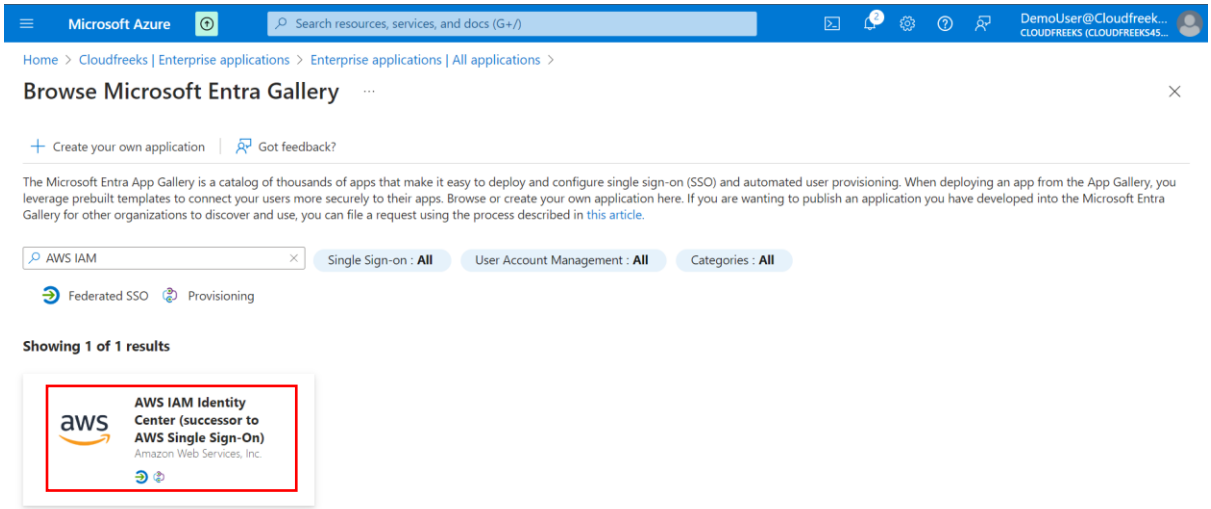
7. Now you are going to open your Azure Portal and here we are going to configure the related parameters required. In your Portal you need to go to Microsoft Entra ID.
8. So, here you need to go to Enterprise Application.



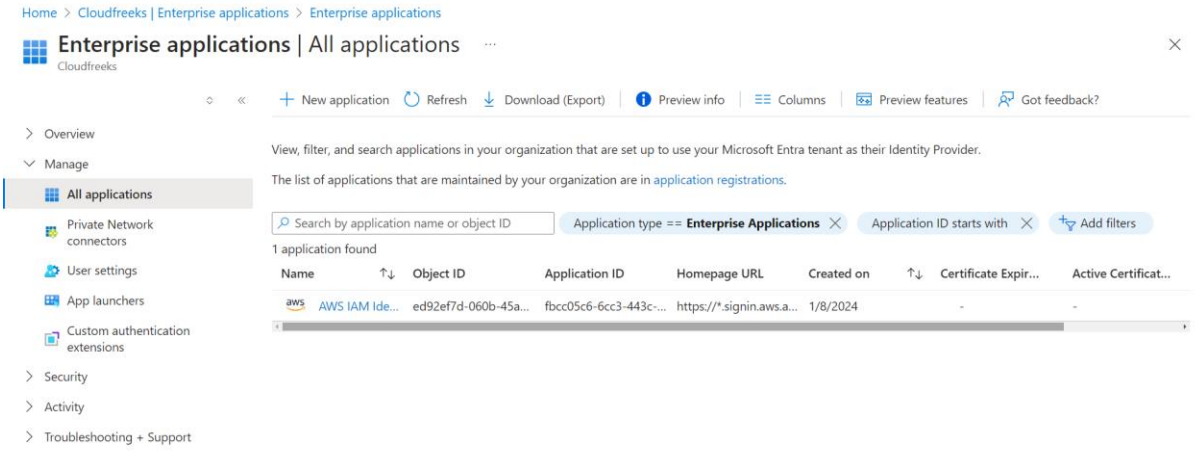
9. Then choose to add a new application.



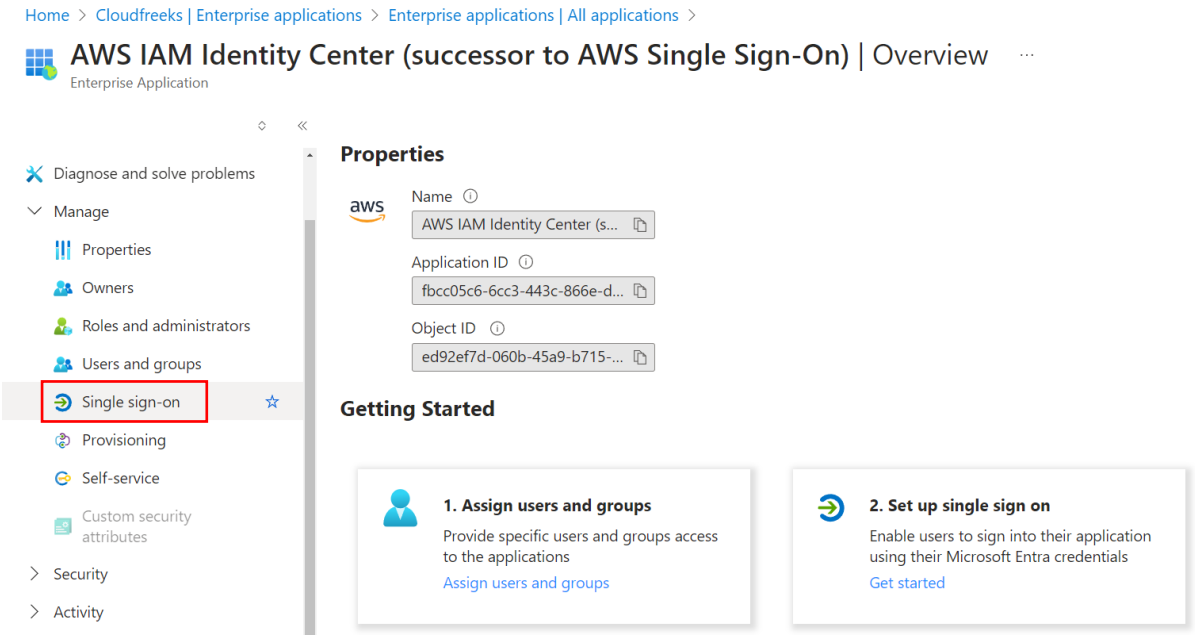
10. Now you need to search for AWS IAM and choose IAM Identity Centre as shown below, then click on create.



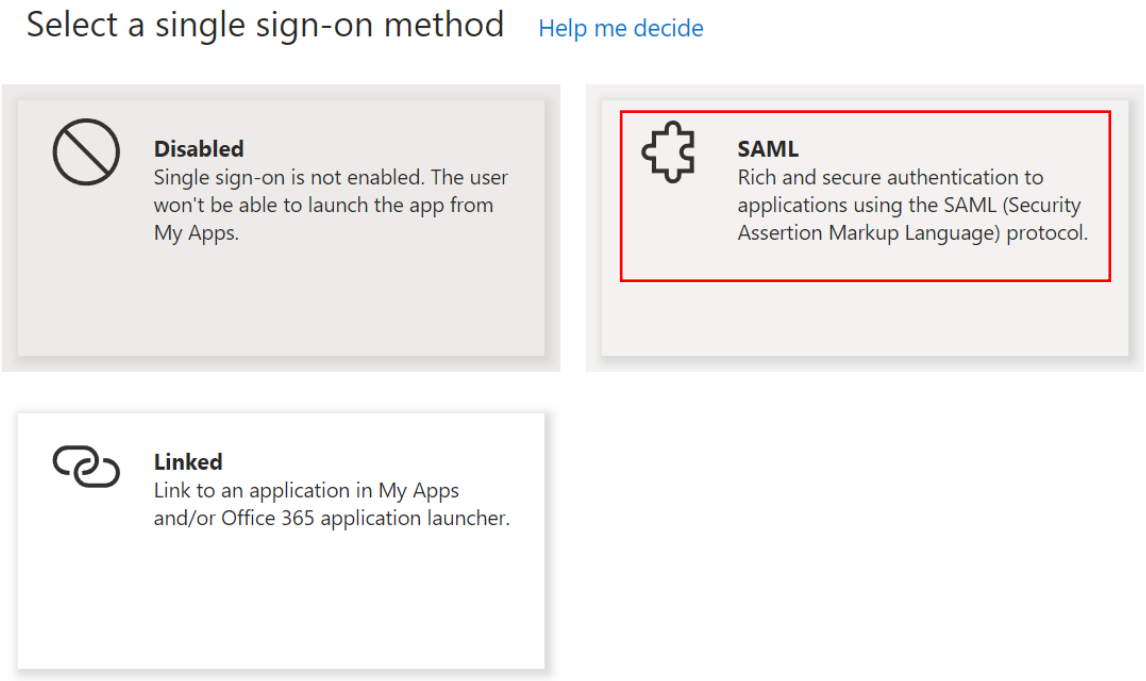
11. Here now you can see the AWS IAM Identity Centre as your enterprise application in Azure.
12. Now you need to go inside of it.



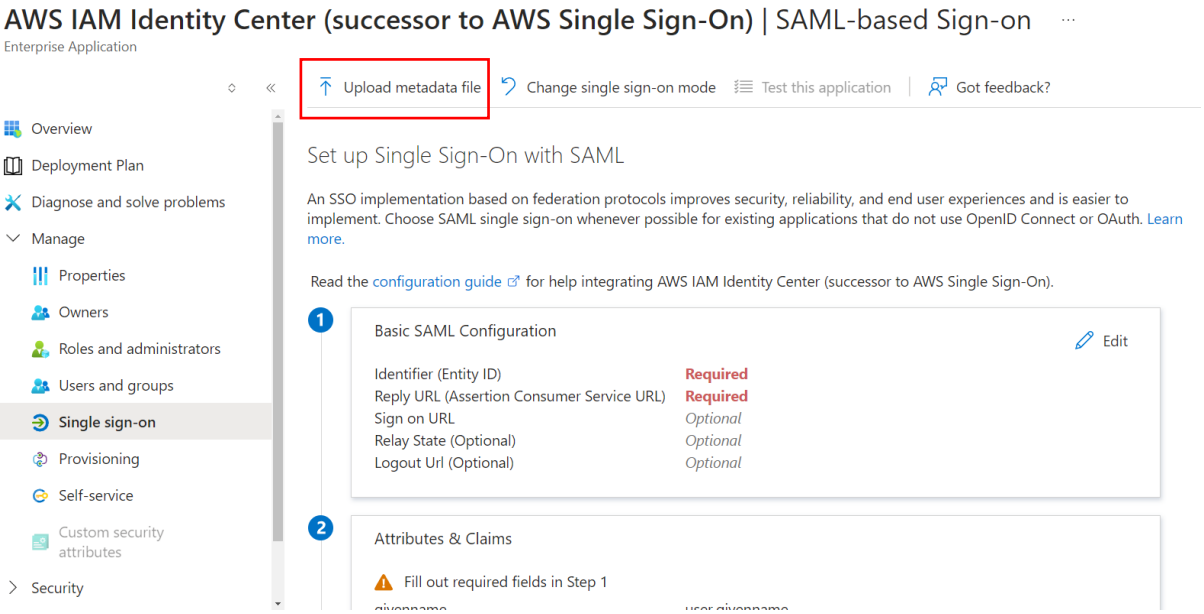
13. Here we need to select Single sign-on.



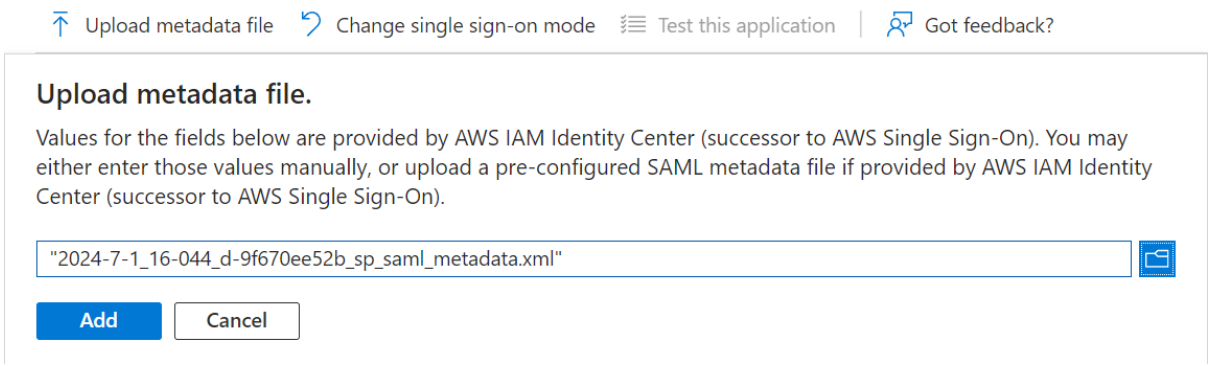
14. Now you are going to choose SAML (Security Assertion Markup Language).



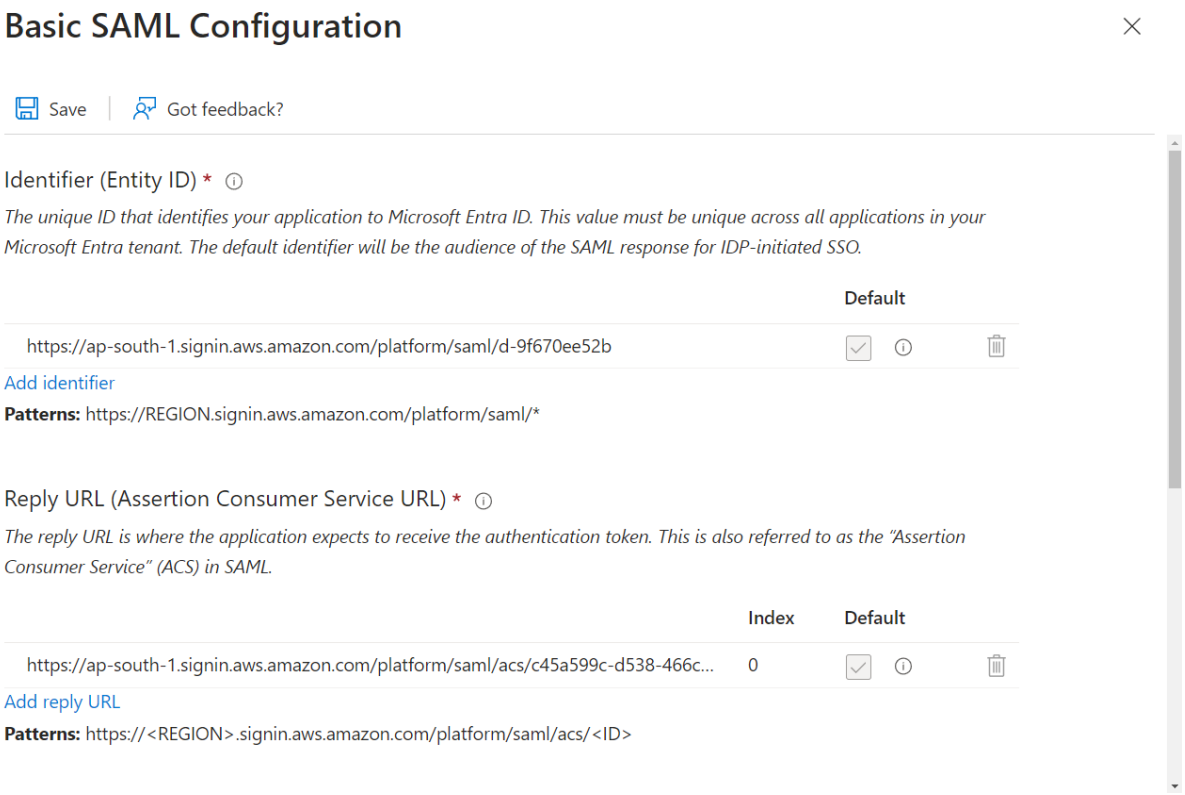
15. Here we are going to click on Upload metadata file.



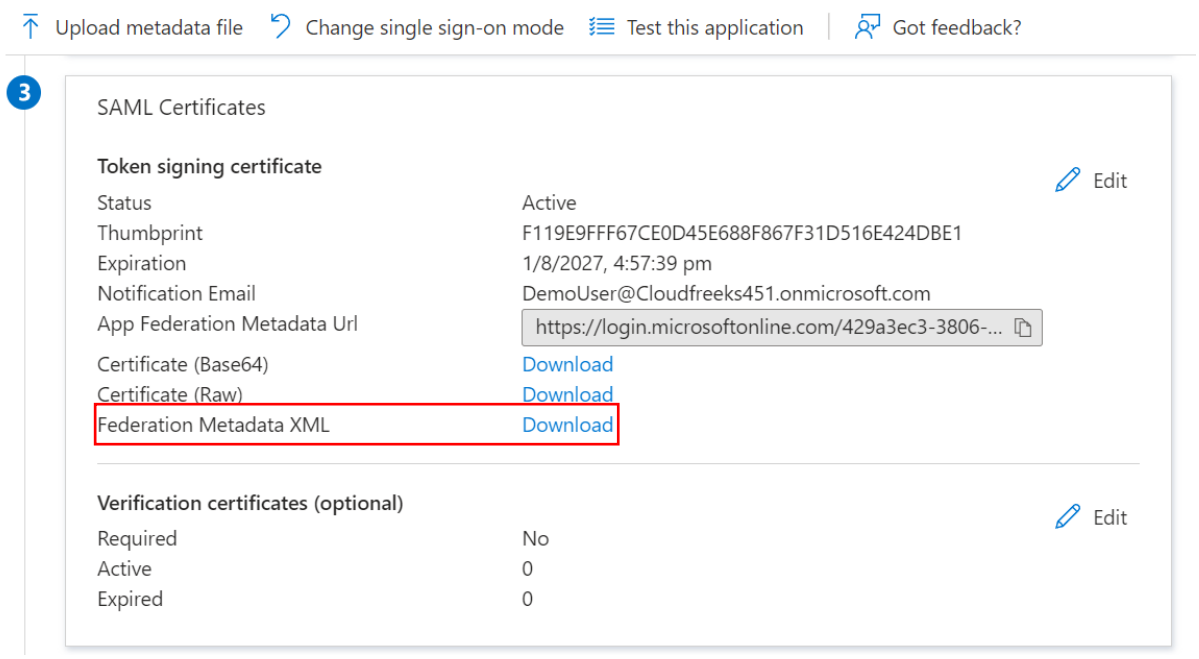
16. Then we are going to upload the metadata file which we'd downloaded from AWS and click on Add.



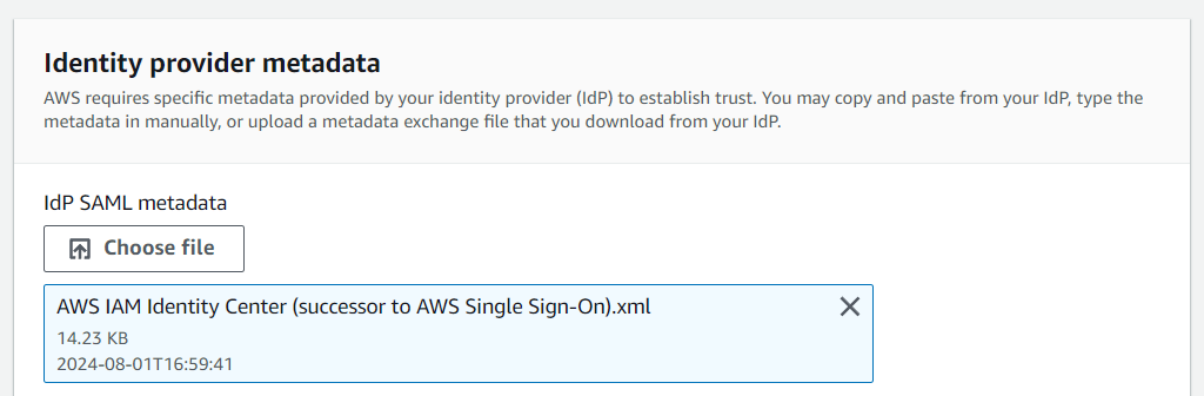
17. Now you can see the basic configuration settings and just click on save no need to add or change anything.



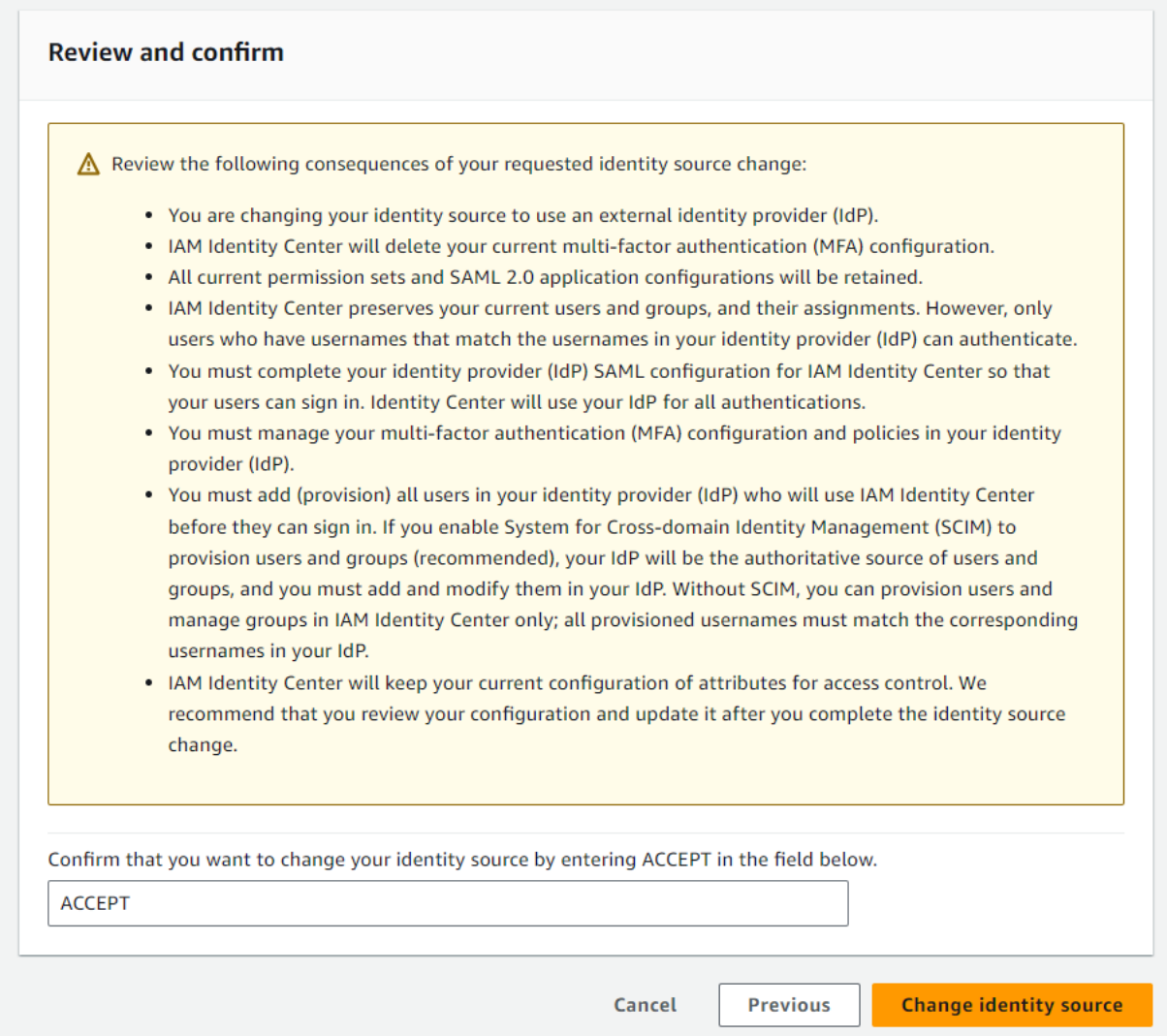
18. Once it is done now you are going to scroll down a little and from the SAML Certificates, you need to download the Federation Metadata XML. So, this file will be required in the AWS console.



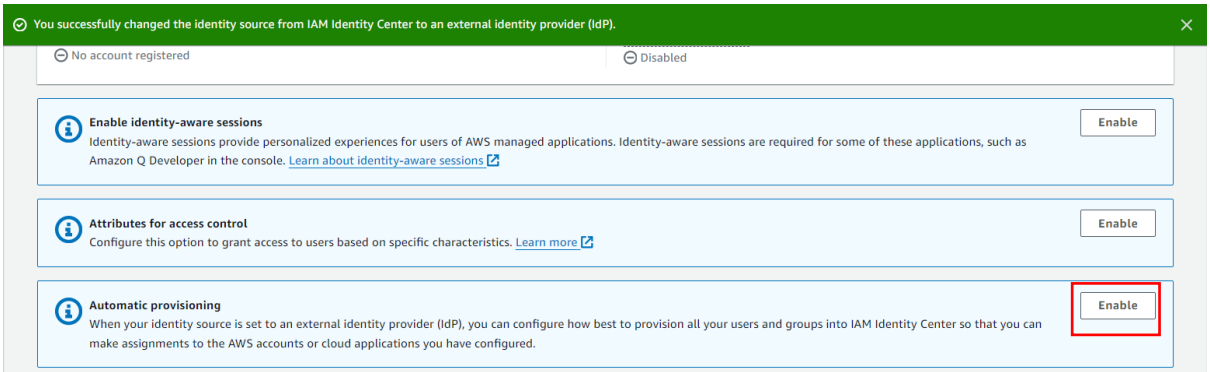
19. Now comeback to AWS Console on the same page where you have left if you scroll down, you will see this identity provider metadata option. Here in the IdP SAML metadata you need to upload the certificate which you downloaded from Azure Portal. Click on next.



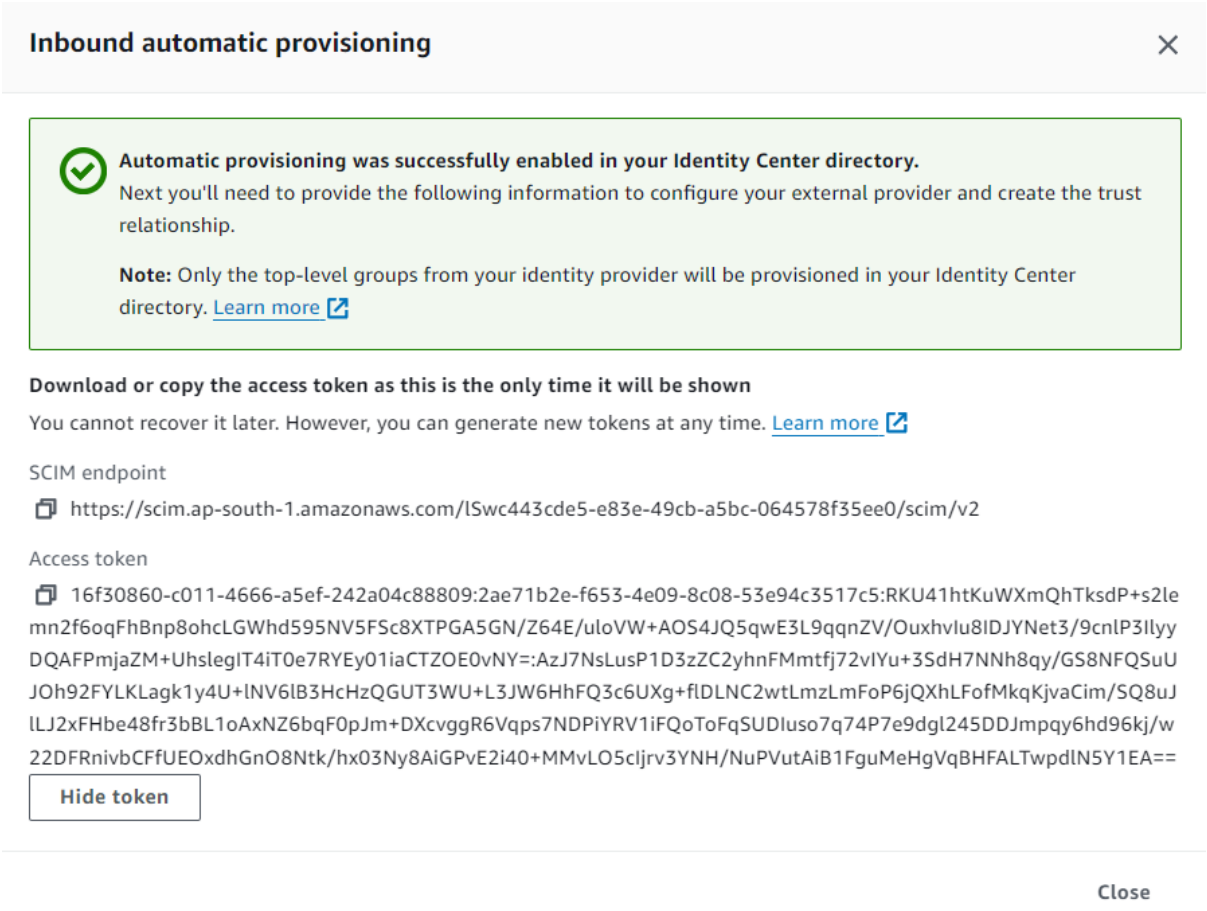
20. In the end you just need to verify everything and type ACCEPT, then just click on change identity source.



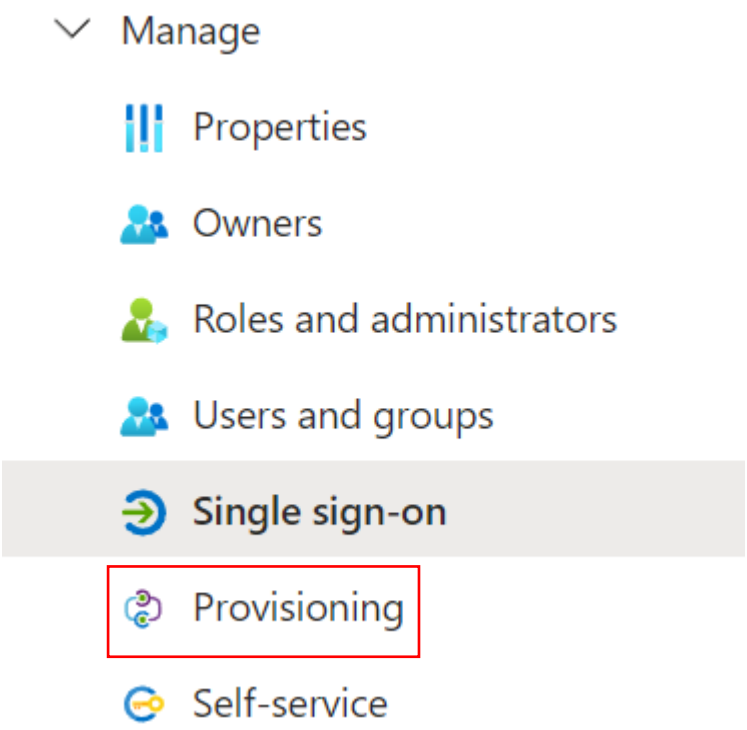
21. You also get a message that you have successfully changed the identity. Now you are going to enable automation provisioning.



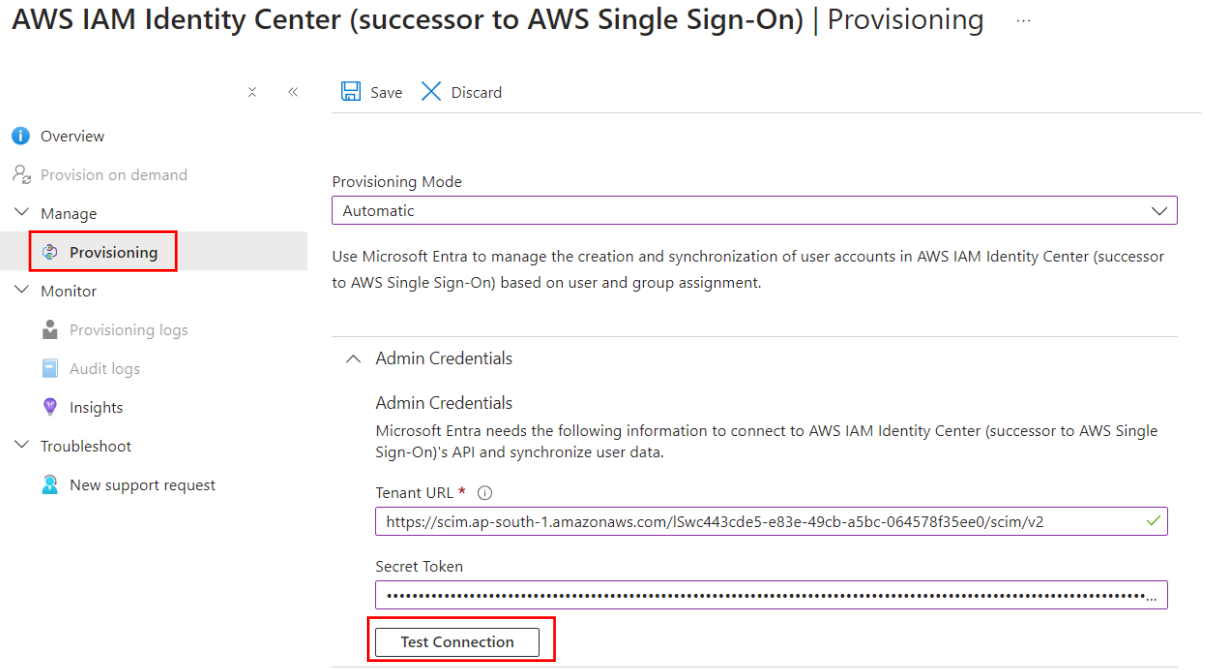
22. So, this automatic provisioning will give you an endpoint and access token. You must copy them both onto your notepad.

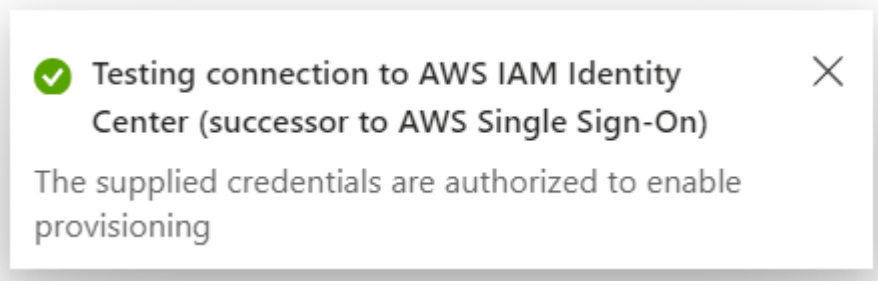


23. Once you have the endpoint and token now you are going to Azure Portal, here you need to go to Provisioning.

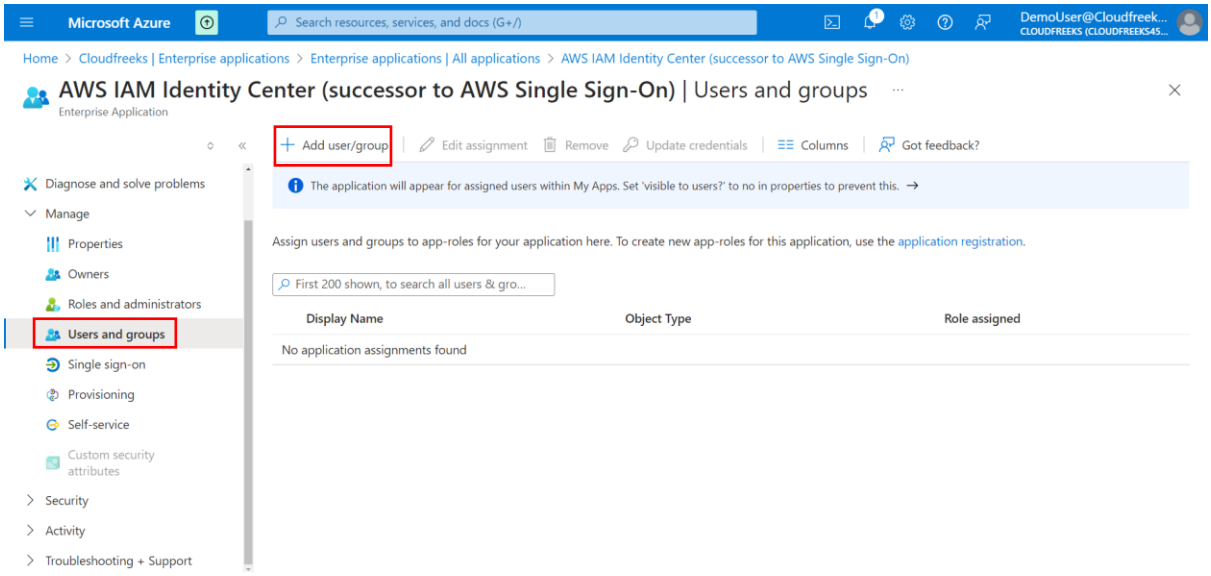


24. Now here you have to go to provisioning and choose the provisioning mode to Automatic. Then you need to paste the SCIM Endpoint and Access Token. After that click on Test connection. If your test is successful then click on save.

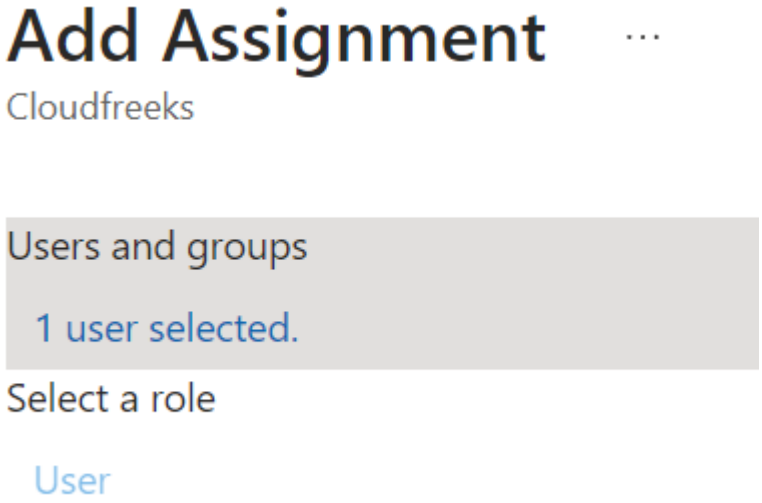


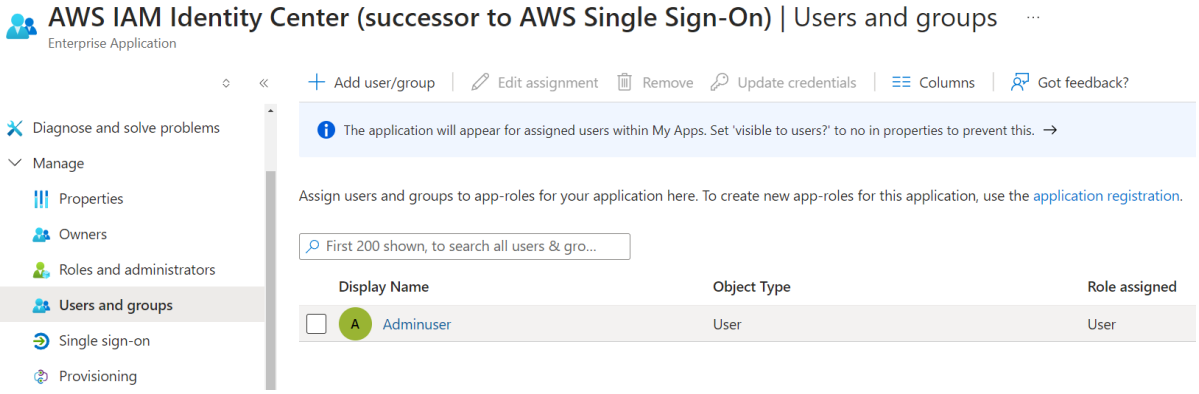


25. Now we are going to add the users from Azure to our Enterprise application. By doing so we can also see the users in our AWS Console. So, for that come back to your application in Azure Portal and choose users and groups then click on add user/group.

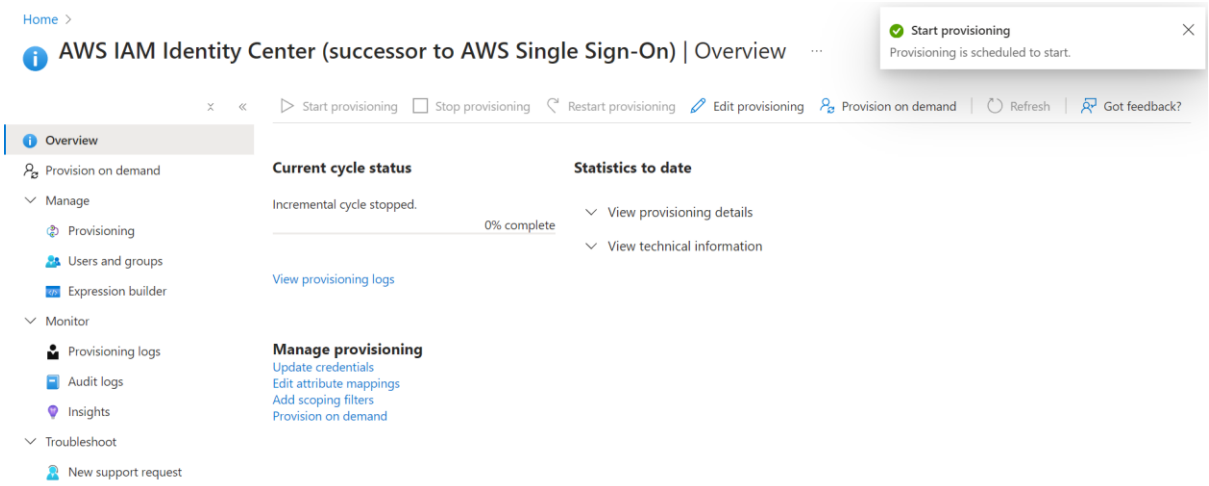


26. Then you need to add a user and click on save. Below you can see that user has been added.



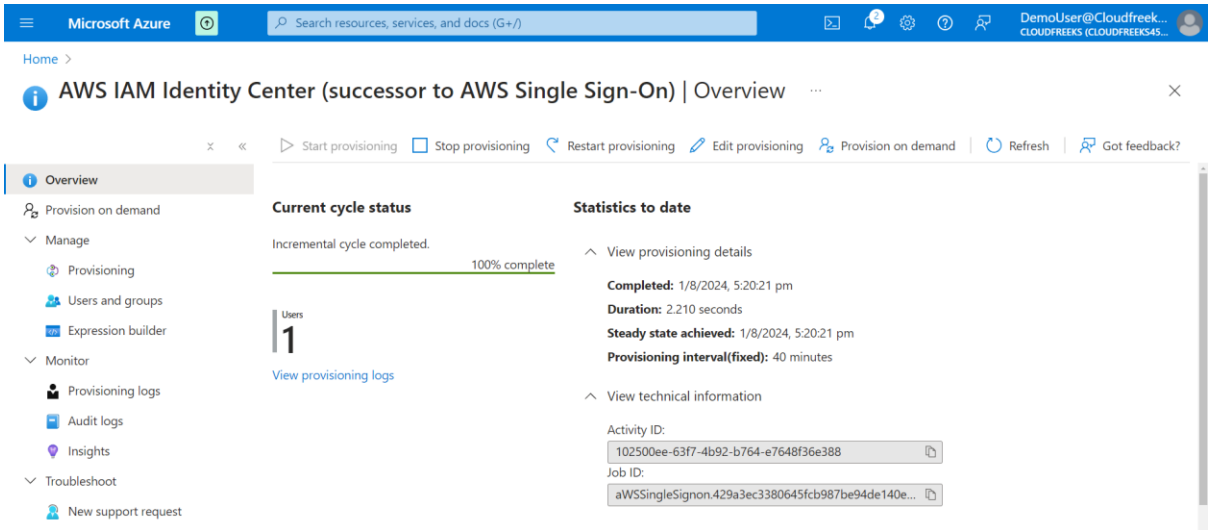


27. Once again go to provisioning and click on start provisioning.

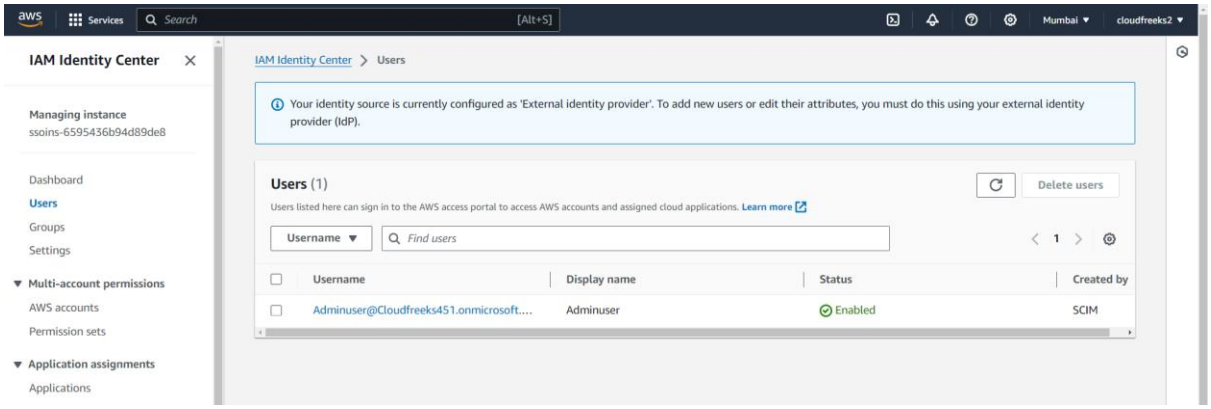


28. Now you need to restart the page and you will see that your user has been added.

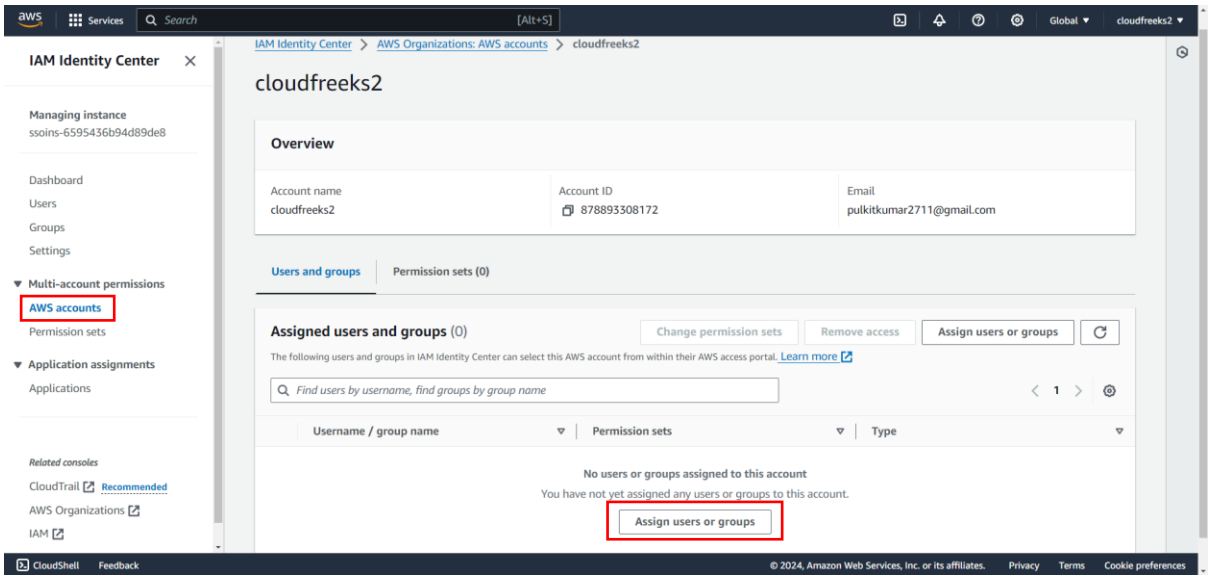
NOTE: Your Azure User should have a first and last name. It is mandatory or it will not show up.



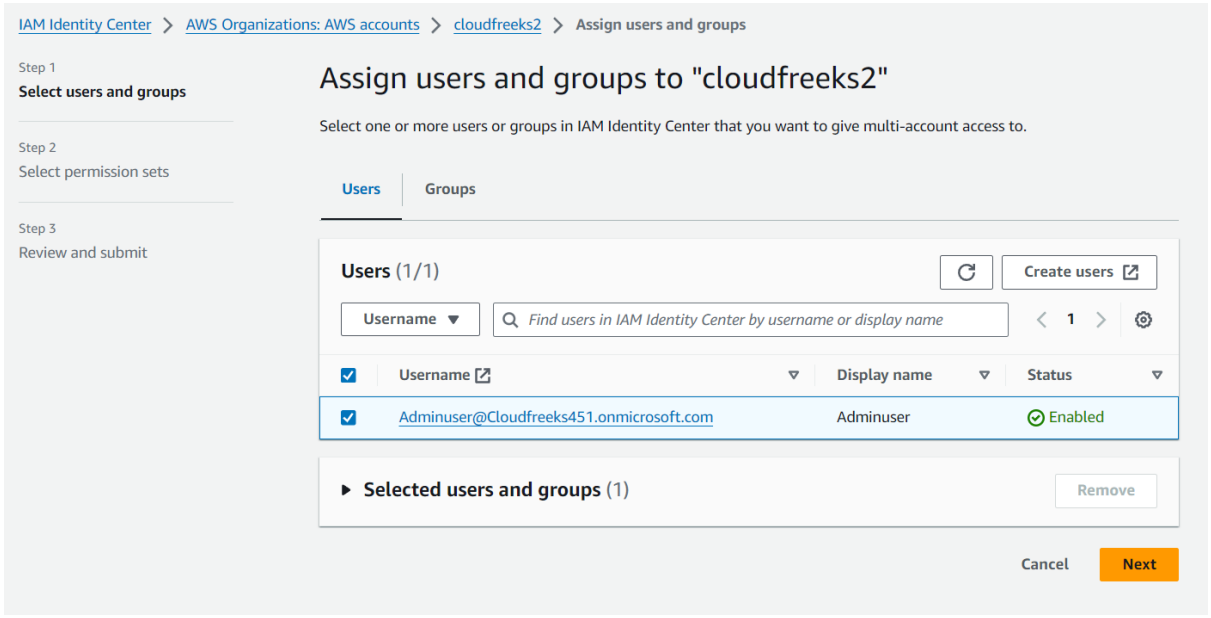
29. Then come back to AWS Console and go to user section in IAM Identity Center. Here you will see the user has been added successfully.



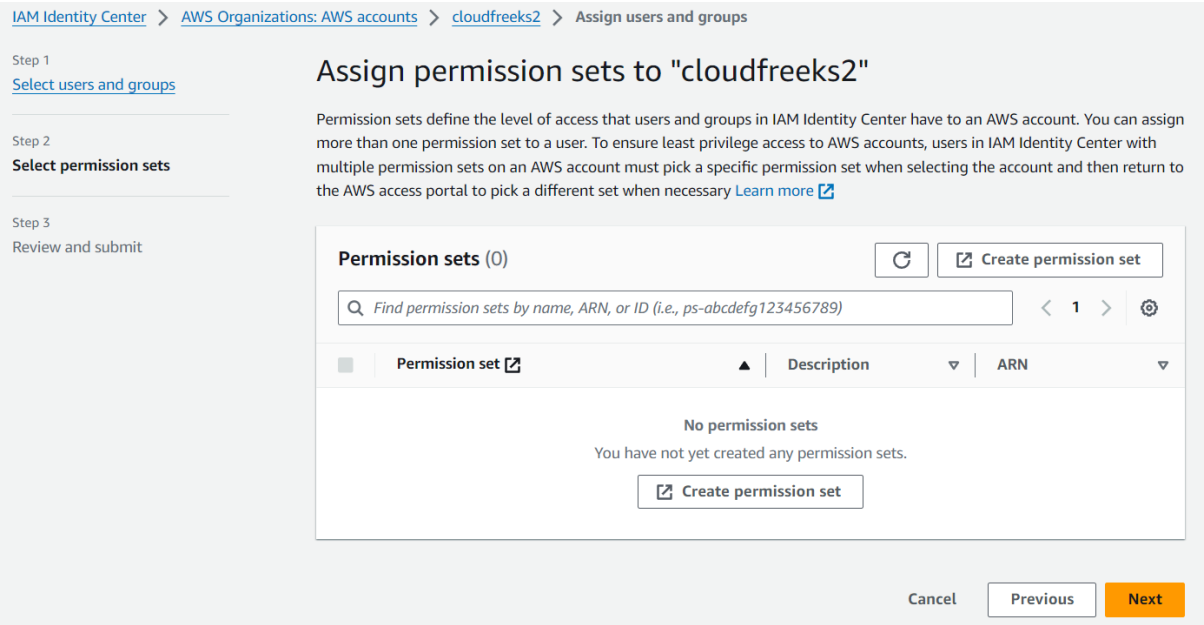
30. Now we are going to assign this user with some permissions so that it can work. For that in the IAM Identity Center we need to click on AWS Accounts, then choose your account after that you need to choose Assign Users or Groups.



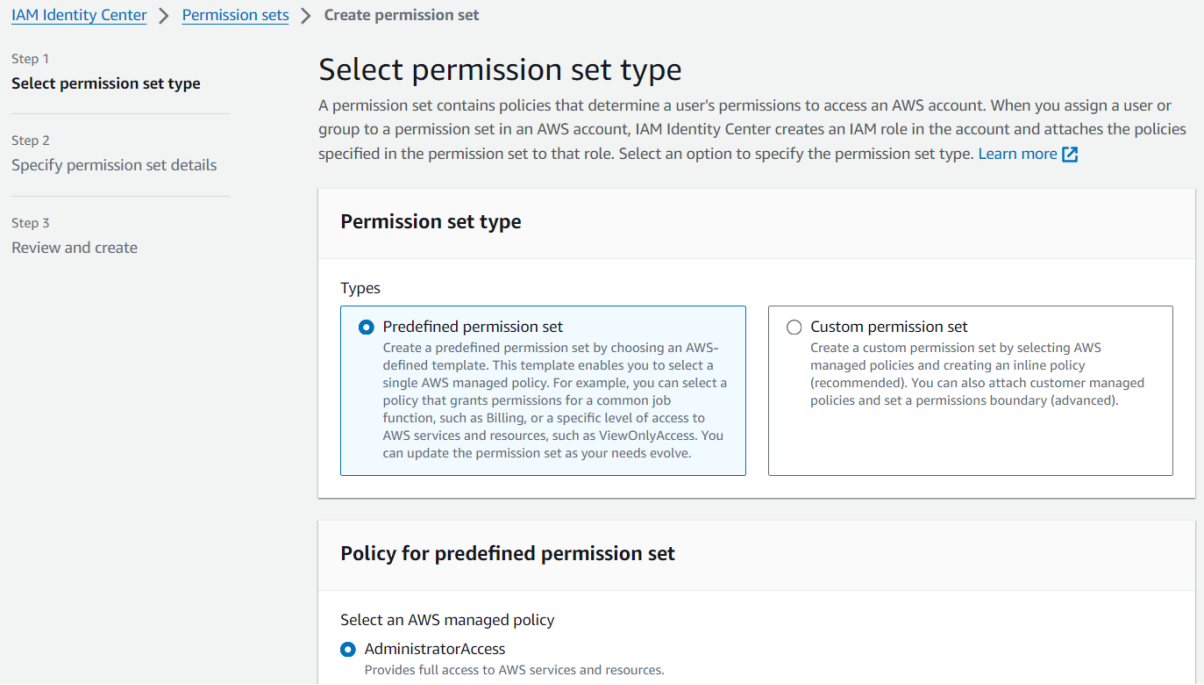
31. Now you need to choose the user and click on next.



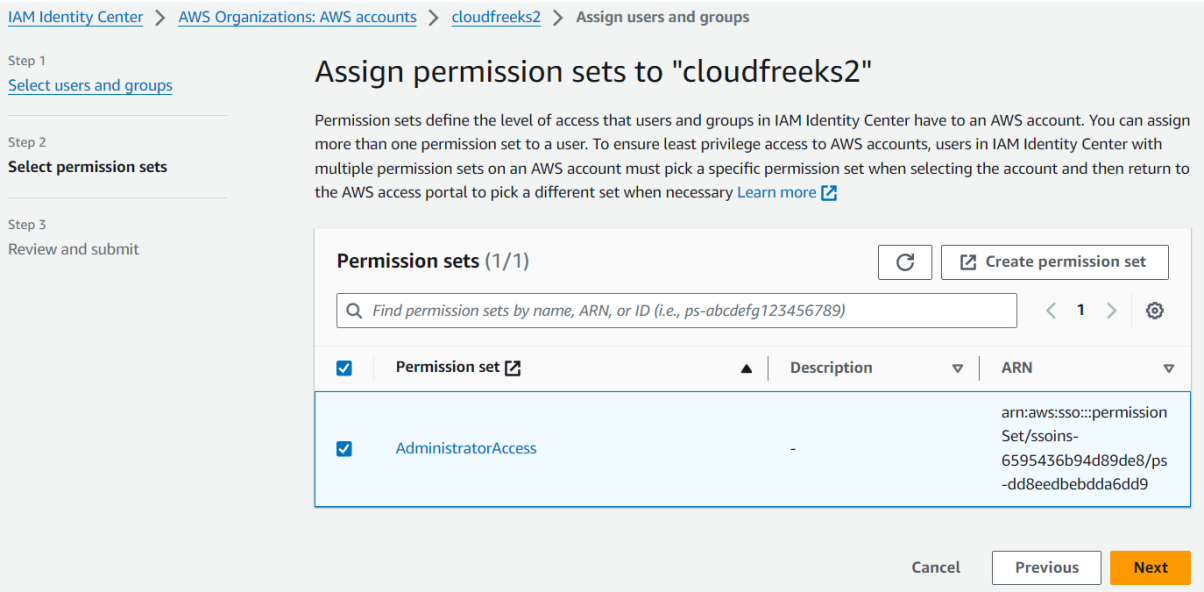
32. Then you have to click on Create Permission set.



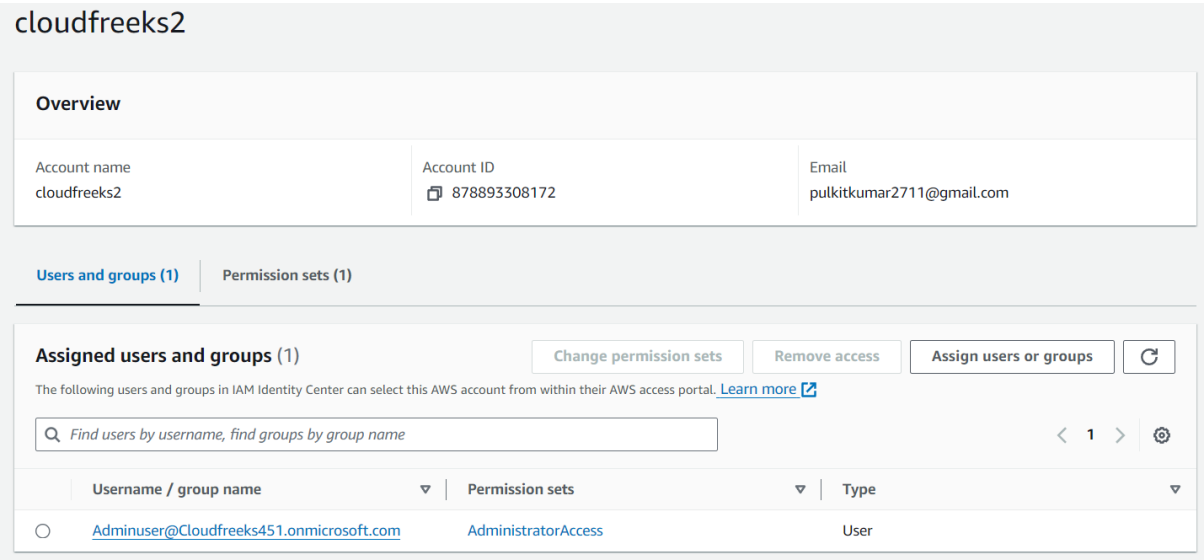
33. Then just choose the predefined permission set and in the choose administrator access and create your permission set.



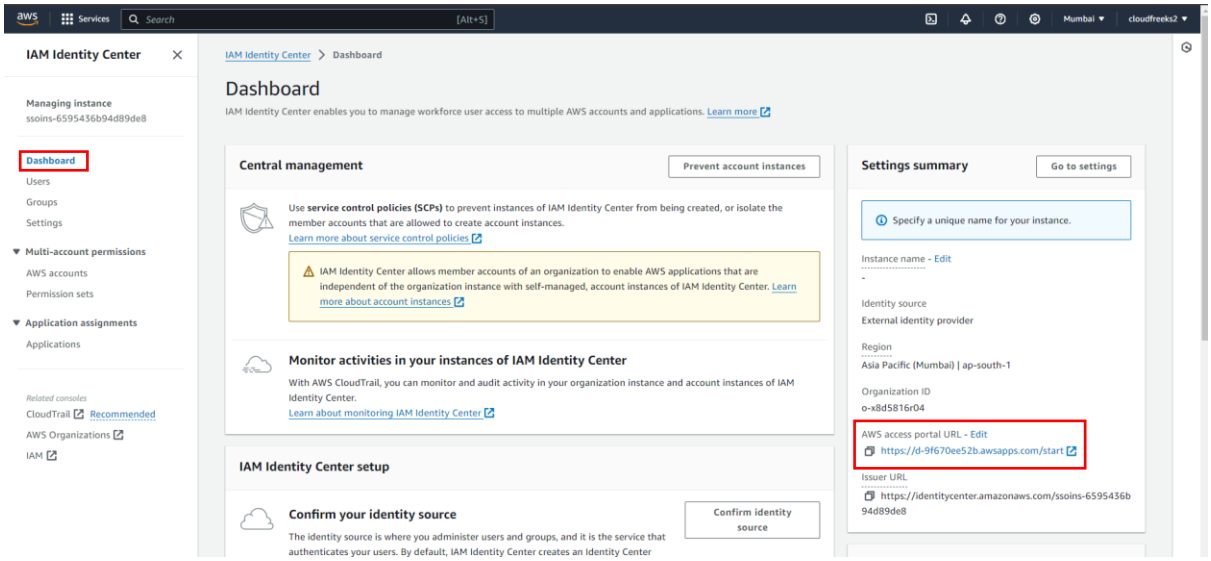
34. Once it is created come back and refresh it you will see your Administrator Access Permission set choose it and click on next to go to review page. Then just create it.



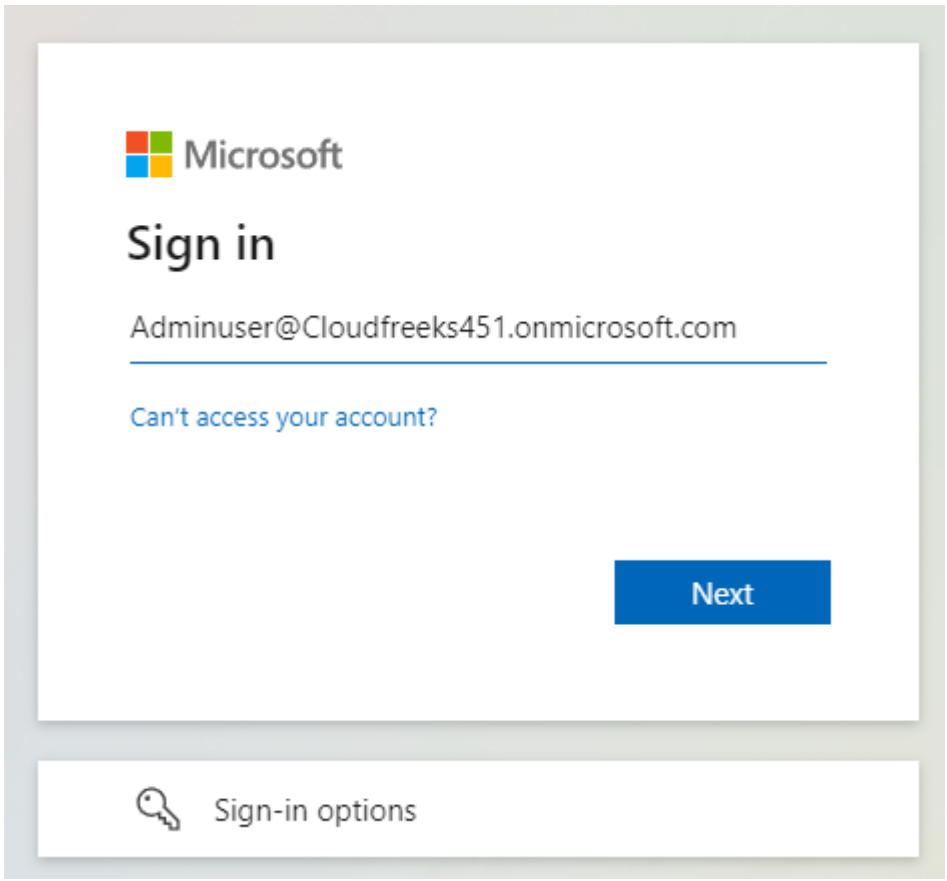
35. Below you can see that the user now has the administrator access.



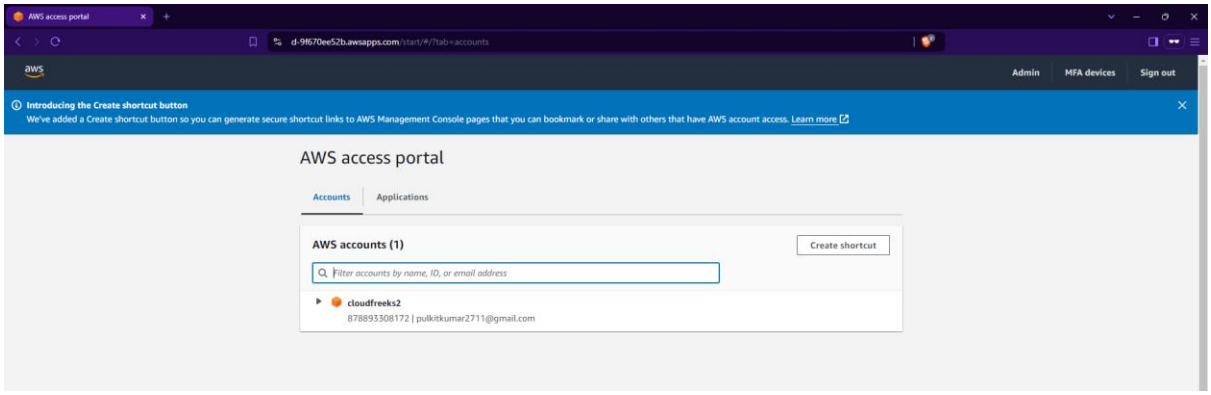
36. Then just go to the Dashboard of IAM Identity Center and copy the AWS Access Portal URL.



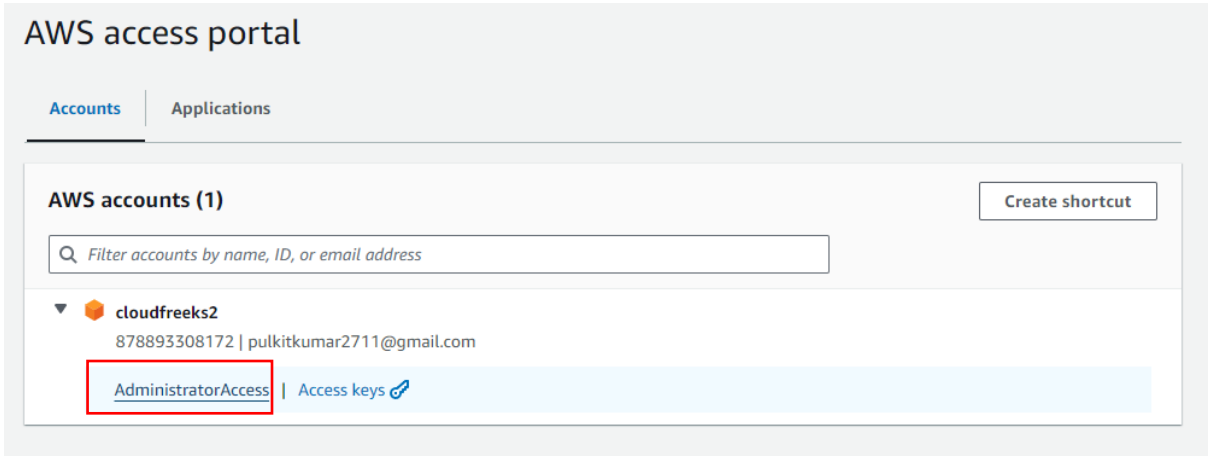
37. After that open a new browser and paste the URL and it will ask you to sign in, you can use your Azure User credentials to login. So, give the username and password.



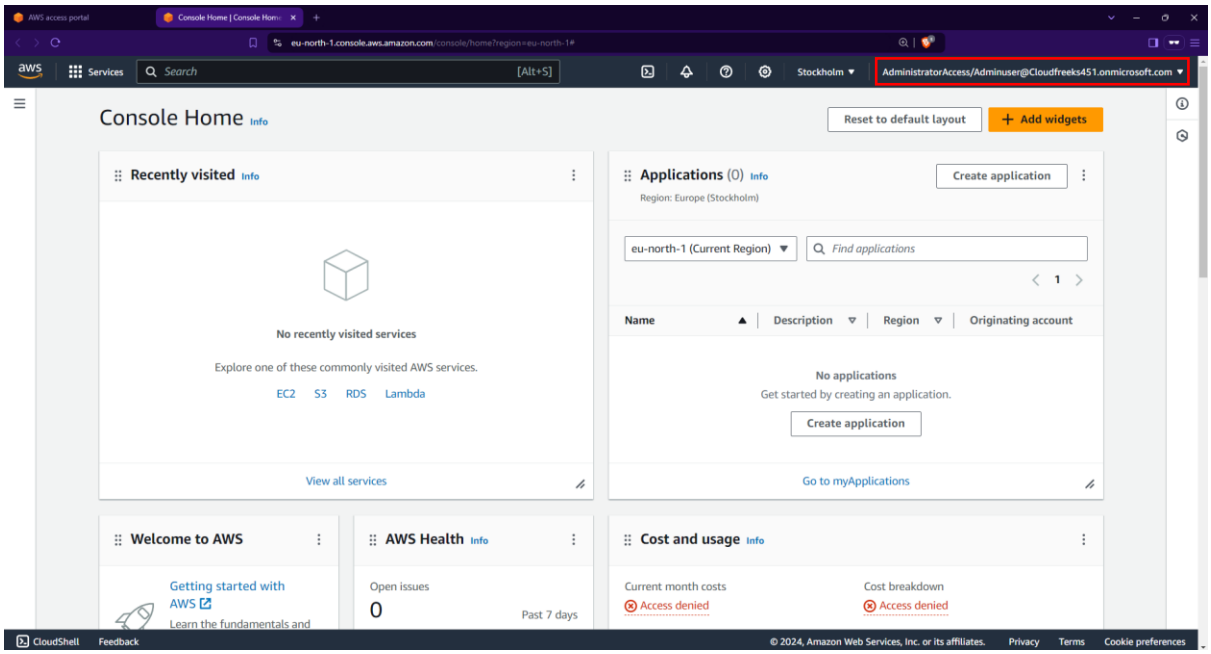
38. Below you can see that you are in the AWS Access Portal now .



39. Then expand the account name and click on Administrator Access.



40. Now you can see that you are in the AWS account using Azure User Credentials.



😊 Configuring the AWS CLI to use IAM Identity Center

🧐 What are we doing in this part?

In this process, you are setting up the AWS Command Line Interface (CLI) to use AWS IAM Identity Center (previously known as AWS Single Sign-On) for authentication. The end goal is to securely manage access to your AWS resources by using centralized authentication through Azure Active Directory.

Summary of Steps:

1. **Install AWS CLI:** Begin by installing the AWS CLI on your computer.
2. **Configure AWS CLI with Identity Center:** Use the aws configure sso command to set up Single Sign-On (SSO) authentication. You'll provide the SSO Start URL and other details from the AWS IAM Identity Center.
3. **Authentication Flow:** After entering the configuration details, you are redirected to a web browser to authenticate. You match a code displayed in the command prompt with one in the browser to confirm your identity.
4. **Granting Permissions:** You ensure that your Azure Admin user account has the necessary permissions in the AWS IAM Identity Center and Azure Portal.
5. **Assign Administrator Access:** Assign the appropriate roles and permissions to the user in AWS IAM Identity Center, ensuring they have the required access to AWS resources.
6. **Finalizing Configuration:** Complete the setup by specifying profile details like region and output format. Then, use the aws sso login --profile demo-admin command to log in, which authenticates you via the browser.
7. **Access AWS Console:** After successful authentication, you can access the AWS Management Console with the assigned administrator role.

End Goal:

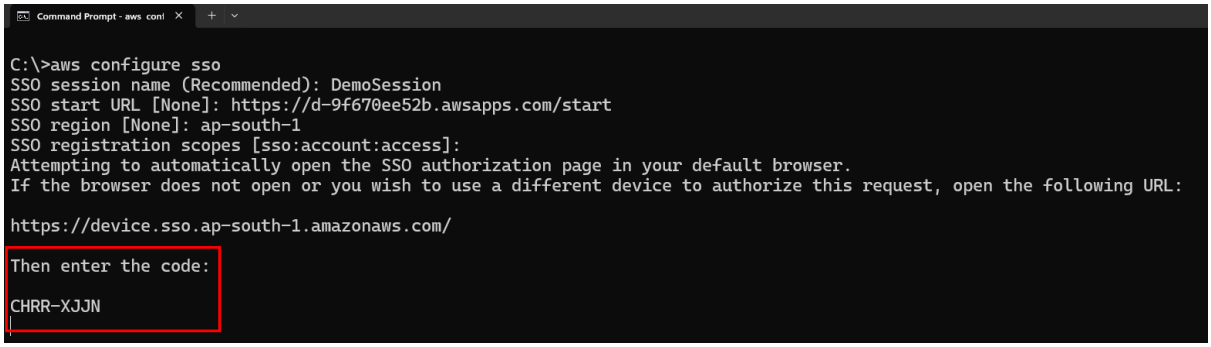
The end goal is to make it easy and secure for you to log in and manage your AWS resources using your existing Azure account. By setting this up, you can use a single sign-on process to access AWS, so you don't have to remember multiple passwords or go through complicated login steps. This setup also ensures that only authorized users can access important AWS tools and services, keeping everything secure.

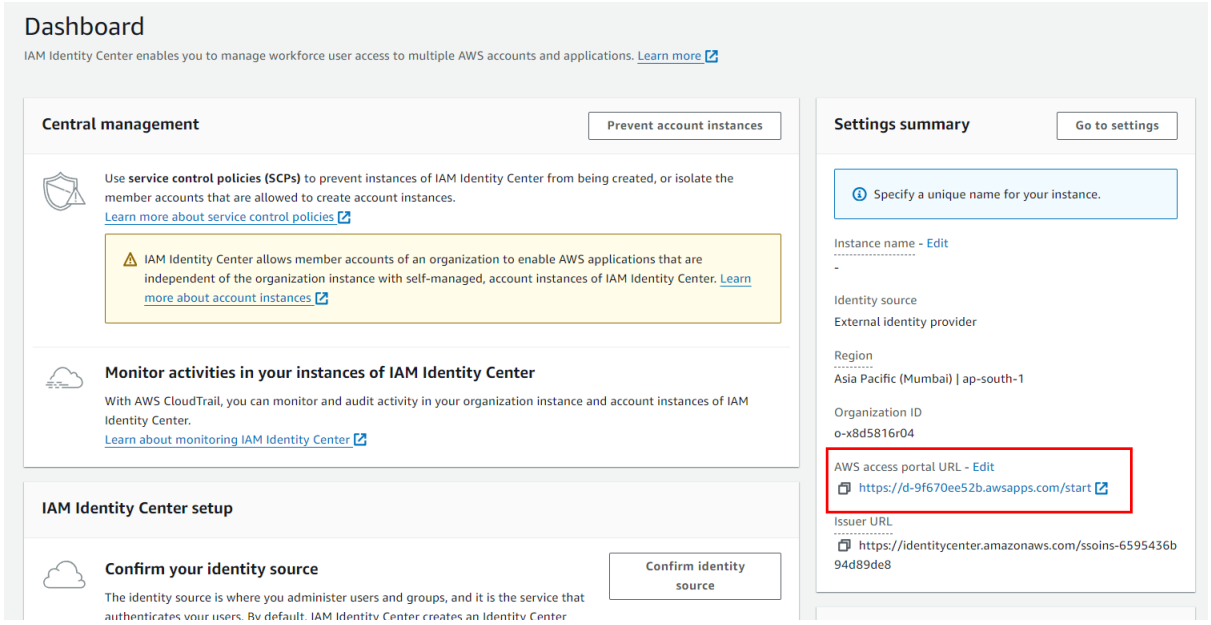
😄 To begin in this part

1. Now we are going to configure AWS CLI to use Identity Center for that first we need to install AWS CLI into our laptop.
2. Once it is done then we are going to run some commands to configure it. Now enter the command given below into your command prompt.

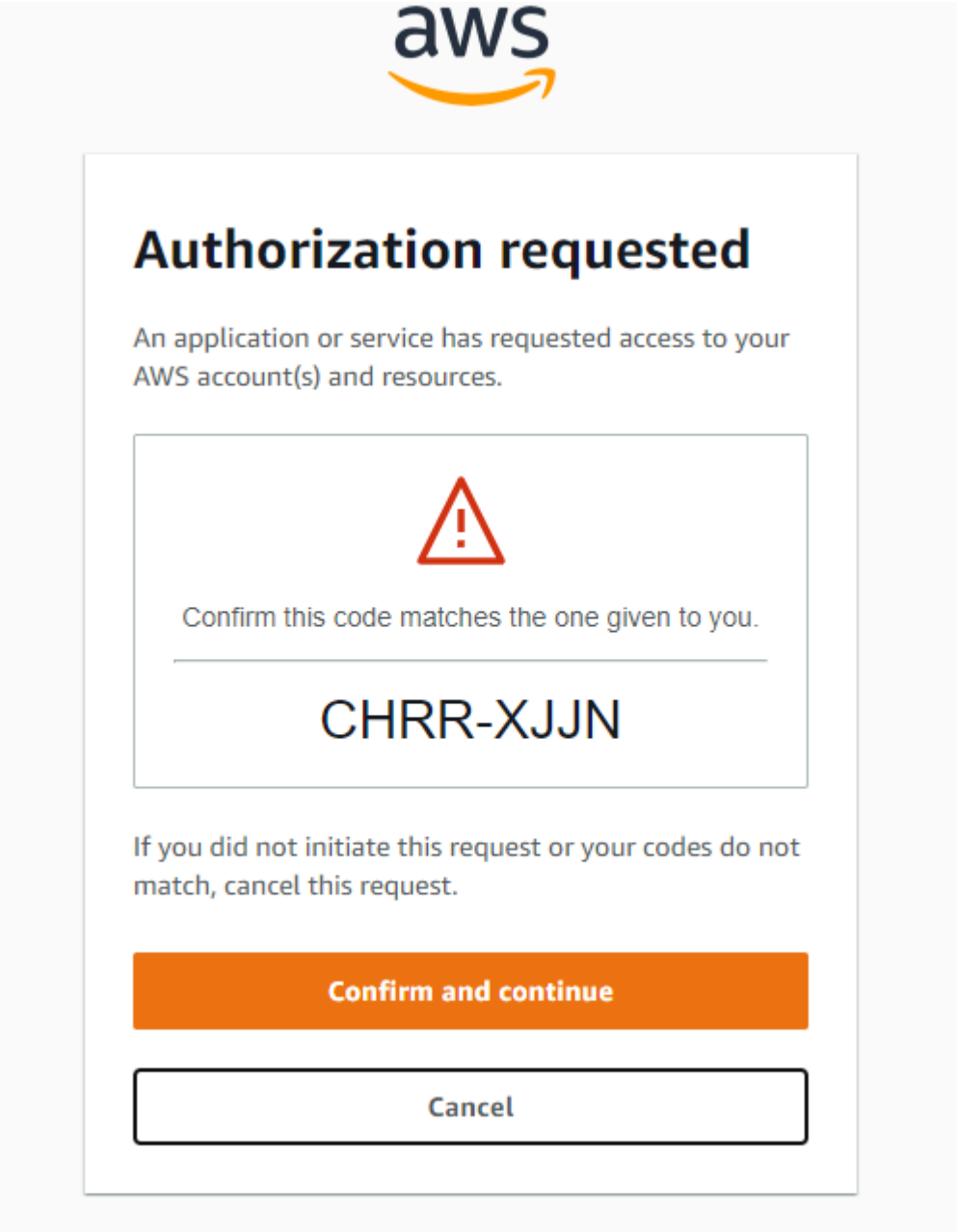
   aws configure sso

3. Then you will see that it will ask you the session name, and the SSO Start URL which you get from the AWS IAM Identity Center Dashboard. You can also see that in the snapshot below.
4. After that write your SSO region then leave the SSO registration scopes and hit on Enter.
5. Now you are directed to the browser and there you need to match the code which you can see below in the command prompt.





6. On the browser also you can see the same code. If the code matches, then you need to click on Confirm and Continue.

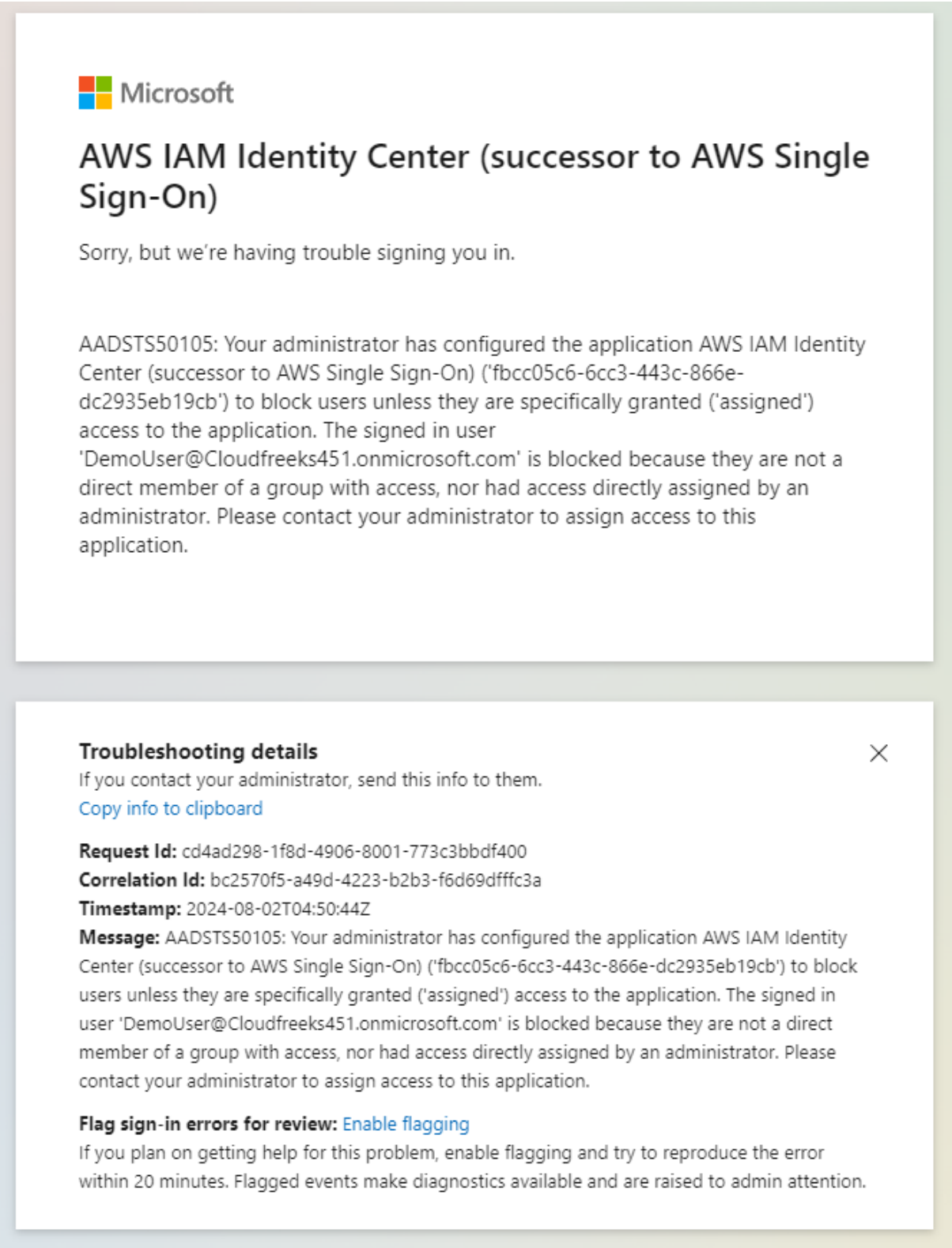


NOTE: If you are facing this error shown below after clicking on confirm and continue then it means that you need to add your Azure Admin user account into the Enterprise application which is AWS IAM Identity center application in Azure Portal.

You need to go inside of your Enterprise application and click on Users and Groups then choose to add new user then choose to add your Account Admin user and then wait for it to get provisioned.

Once it has been provisioned then you need to come back to AWS and go to users in IAM Identity Center. There you need to check whether the user has been added or not. If it has been added, then you need to go AWS Accounts and from there give your new user the Administrator access like you had given to the previous user.

Once it is done then you can go back to command prompt and run the aws sso configure command again and you will be able to perform the steps shown below.



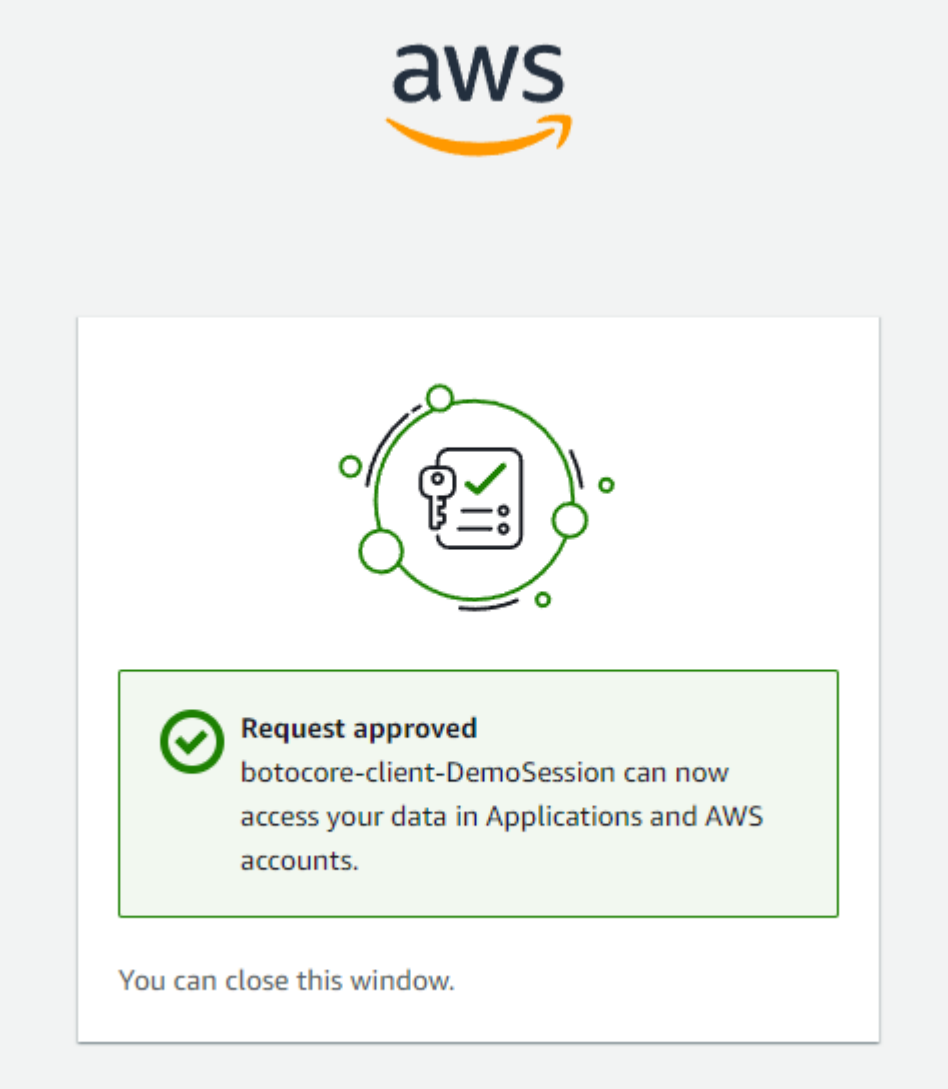
7. After that you will see this screen and you need to click on Allow Access. Then will see a message saying that your request has been approved.

Allow botocore-client-DemoSession to access your data?

By choosing **Allow access**, you agree to allow **botocore-client-DemoSession** to access the following:

Applications and AWS accounts
Show details

Deny access | Allow access

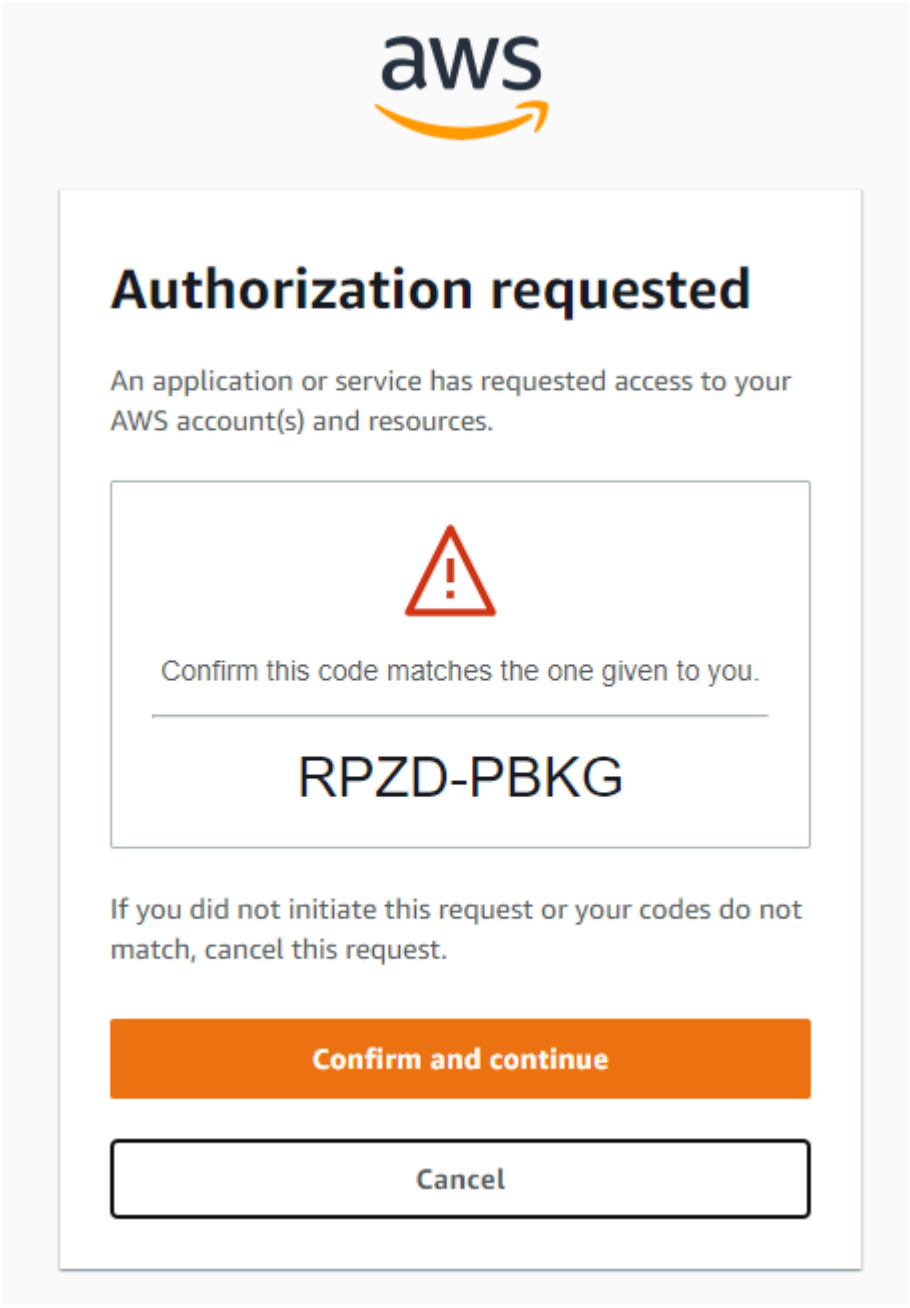


8. Now you should come back to command prompt and here you will see that it is saying that the only account available is my own account and the role name is Administrator access.
9. Then just leave the default region name and default output format. After that you need to give a profile name. If you want to use, then you can use the default name AWS is suggesting you.

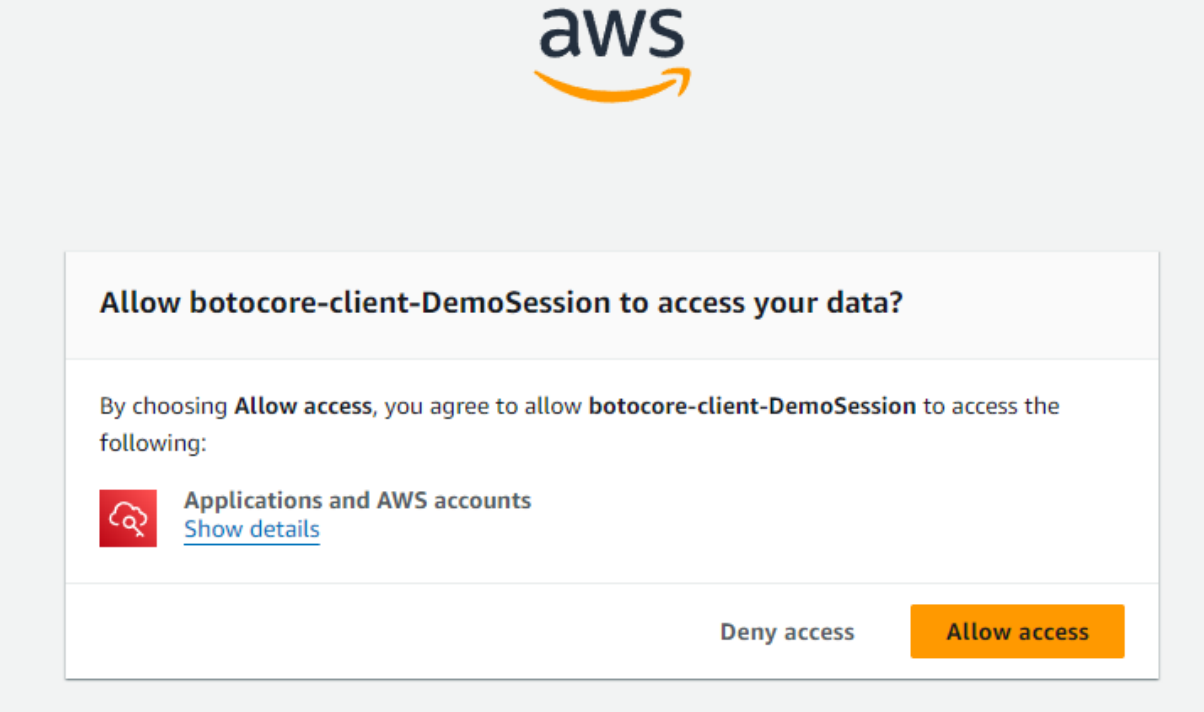
```
Then enter the code:

PTNM-HHGL
The only AWS account available to you is: 878893308172
Using the account ID 878893308172
The only role available to you is: AdministratorAccess
Using the role name "AdministratorAccess"
CLI default client Region [ap-south-1]:
CLI default output format [none]:
CLI profile name [AdministratorAccess-878893308172]: demo-admin

To use this profile, specify the profile name using --profile, as shown:

aws s3 ls --profile demo-admin
```

10. Once you profile is setup now you are going to login for that you need to write this command in your command prompt. Then it will take you to the browser again to confirm the code. Just click on confirm and continue.

    aws sso login --profile demo-admin



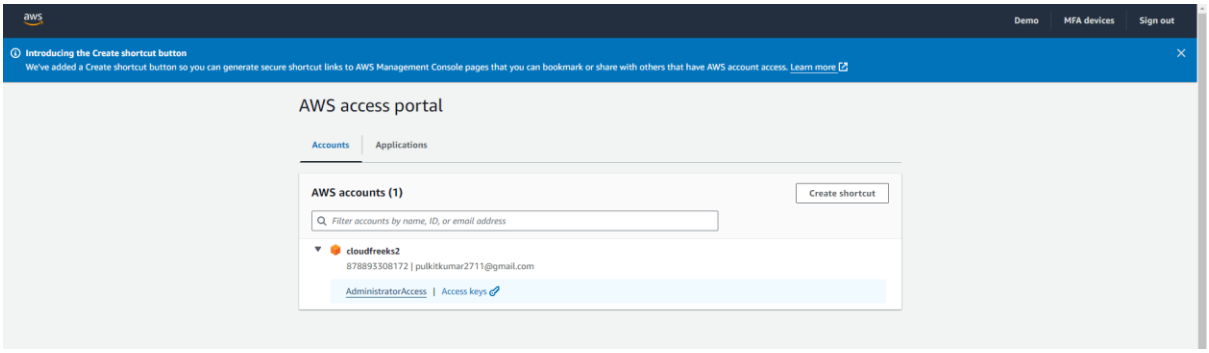
11. Now you need to click on Allow Access and then close the tab.



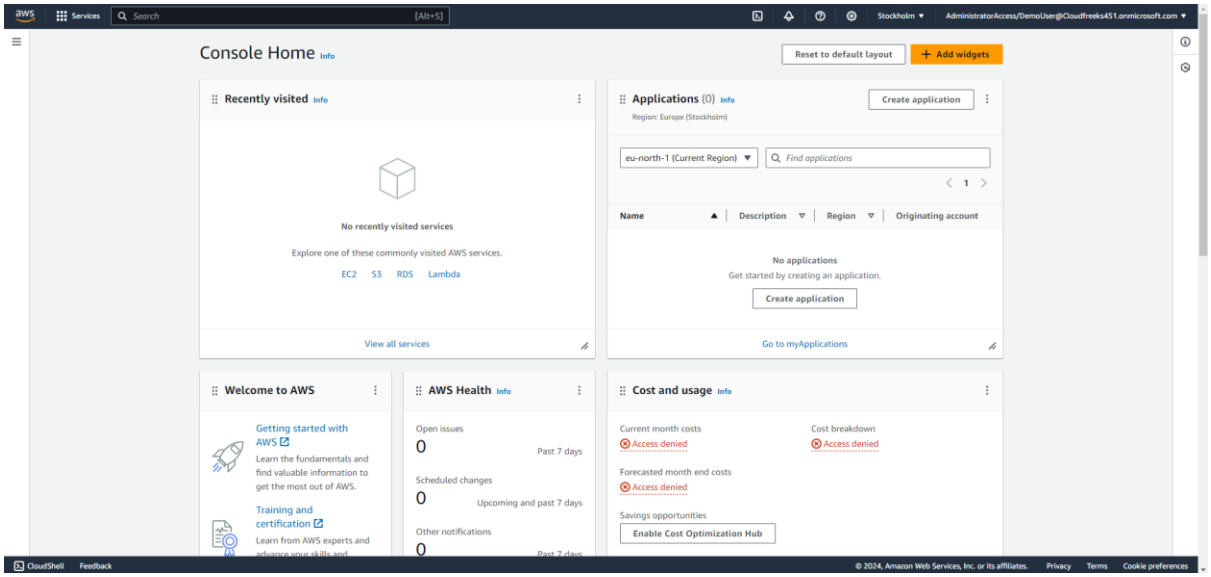
12. After that come back to the command prompt and here you will see that it has given you the URL. Copy this URL and paste it in a new tab or browser.

```
C:\>aws sso login --profile demo-admin
Attempting to automatically open the SSO authorization page in your default browser.
If the browser does not open or you wish to use a different device to authorize this request, open the following URL:

https://device.sso.ap-south-1.amazonaws.com/

Then enter the code:

RPZD-PBKG
Successfully logged into Start URL: https://d-9f670ee52b.awsapps.com/start

C:\>
```

13. Then you will be on this page and you need to got console, for that click on administrator access



14. Then you will be in the Console and you can use it.


Services
Search
[Alt+S]
Stockholm
AdministratorAccess/DemoUser@Cloudfreeks451.onmicrosoft.com
Console Home Info
Reset to default layout
+ Add widgets
Recently visited Info
No recently visited services
Explore one of these commonly visited AWS services.
EC2 S3 RDS Lambda
View all services
Applications (0) Info
Region: Europe (Stockholm)
Create application
eu-north-1 (Current Region)
Find applications
1
Name
Description
Region
Originating account
No applications
Get started by creating an application.
Create application
Go to myApplications
Welcome to AWS
Getting started with AWS
Learn the fundamentals and find valuable information to get the most out of AWS.
Training and certification
Learn from AWS experts and
AWS Health Info
Open issues
0
Past 7 days
Scheduled changes
0
Upcoming and past 7 days
Other notifications
0
Past 7 days
Cost and usage Info
Current month costs
Access denied
Cost breakdown
Access denied
Forecasted month end costs
Access denied
Savings opportunities
Enable Cost Optimization Hub
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences