

Cross-Account IAM Roles

1. So basically, you have two sets of accounts. First one is referred as the identity account. So, an identity account is where the identities are stored and what are identities. Identities are IAM users and then you have various other accounts where the actual resources are created. It can be S3 bucket, it can be EC2 instances and so on.
2. So, the architecture is the IAM user will log in to this identity account and then depending upon the permission of the user, he can log in to account B account C or whatever AWS accounts are present through the assume role functionality.
3. Now the overall steps that are involved are three.
4. First one is you have to create a user in account A. So, the account A is nothing but the identity account.
5. Then you need to create a cross-account role in account B. So, cross-account role is where you go ahead and create an IAM role and this IAM role is what the IAM users from the identity account will be assuming.
6. And then the third step is allowing user to switch to account B role. So, for the IAM user that has stored in the identity account, that user also needs permission to assume this role so that permission you'll have to give in the identity account.

What are we doing in this Lab?

In this scenario, you are setting up cross-account IAM roles in AWS to securely manage access between different AWS accounts. The process involves:

1. **Creating an IAM User in Account A (Identity Account):** This is where the identities, like IAM users, are stored. A user, such as Alex, is created with AWS console access but no initial permissions.
2. **Creating a Cross-Account Role in Account B (Resource Account):** A role is created in another AWS account (Account B) that defines what permissions are granted when users from Account A assume this role. For example, giving read-only access to S3 buckets.
3. **Granting Permission to Assume the Role:** The IAM user in Account A needs permission to assume the role in Account B. This is done by creating an inline policy in Account A that allows the user to assume the specific role created in Account B.
4. **Switching Roles and Accessing Resources:** The IAM user (Alex) from Account A can switch to the role in Account B, gaining the specified permissions (like read-only access to S3). The user can switch back to their original account when needed.

End Goal: The end goal is to let a user from one AWS account (where their identity is managed) securely access resources in another AWS account (like files or servers) without having to create separate user accounts in each place. This setup makes managing access easier and more secure, as permissions can be controlled centrally and users only need to remember one set of credentials.

Step 1: Create User in Account A

1. Now for this lab you should be having two accounts and you need to login in both of these accounts.
2. Now go whichever you have considered account A, navigate to it. Then go to IAM and create a user.
3. Give your user a name and then provide it with AWS Console access.
4. Then type in your custom password and uncheck for new password at next sign up.
5. Then create your user without attaching any policies.

User details

User name

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 Keypairs, 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.

- Must be at least 8 characters long
- Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (hyphen) = [] { } | `

Show password

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

6. Once your user is created then move to step 2.

😊 Step 2: Create Cross Account Role in Account B

1. Now you need to navigate to your account B and go to IAM.
2. There you are going to create a new role. Now while creating your role, you need to click on AWS account and in that you have to select Another AWS Account.
3. Afterwards go back to your Main account and copy your Account ID. Then click on next.

Select trusted entity [Info](#)

Trusted entity type

- AWS service Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy Create a custom trust policy to enable others to perform actions in this account.

An AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

- This account (463646775279)
- Another AWS account

Account ID
Identifier of the account that can use this role

Account ID is a 12-digit number.

Options

- Require external ID (Best practice when a third party will assume this role)
- Require MFA Requires that the assuming entity use multi-factor authentication.

[Cancel](#) [Next](#)

4. On the next page it will ask you for permission.
5. So, what this basically means is that once the IAM user from the identity account assumes this role and connects to the account, B what are the permissions that this user must have, whether he must have access to S3, access to EC2 and so on.
6. Now you can give permission whether to give access to S3 Read only policy or EC2 read only policy. It depends on you. I have given permission for S3 read only access.

Step 1: Select trusted entities [Edit](#)

Trust policy

```

1> [{
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "sts:AssumeRole",
7       "Principal": {
8         "AWS": "878893308172"
9       },
10      "Condition": {}
11    }
12  ]
13 }]

```

Step 2: Add permissions [Edit](#)

Permissions policy summary

Policy name	Type	Attached as
AmazonS3ReadOnlyAccess	AWS managed	Permissions policy

7. Now you see for the cross account, IAM role, you basically have one more option which states, give this link to user who can switch the role in the console. So, if a user from the identity account wants to connect to the account B then he has to go ahead and click on this specific link.

IAM > Roles > CrossAccount-B

CrossAccount-B [Info](#)

Summary		Edit
Creation date	ARN	Link to switch roles in console https://signin.aws.amazon.com/switchrole?roleName=CrossAccount-B&account=463646775279
February 25, 2024, 23:05 (UTC+05:30)	arn:aws:iam::463646775279:role/CrossAccount-B	
Last activity	Maximum session duration	1 hour
-	-	-

😊 Step 3: Allow user to switch to Account B

1. So, currently what has happened is our Alex user is created in identity account and that user does not have any permission. So that means that the Alex user will not even be able to go ahead and connect to the cross account IAM role in account B. So, we have to give minimal permission to Alex user so that he can at least connect to the IAM role of different AWS accounts.
2. Now quickly go back to your main account A and then open up your user and then click on add permission to it.
3. Then select create inline policy.
4. Here you need to switch to JSON and paste this policy over here. In this policy role is to just assume role and you need to paste the ARN of the Role that you have created in your account B.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::463646775279:role/CrossAccount-B"
  }
}
```

Specify permissions [Info](#)

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

Policy editor

```

1 ▼ {
2   "Version": "2012-10-17",
3   "Statement": {
4     "Effect": "Allow",
5     "Action": "sts:AssumeRole",
6     "Resource": "arn:aws:iam::463646775279:role/CrossAccount-B"
7   }
8 }
```

5. Now go to the review page and create your policy.
6. Now our all steps are completed. We just need to login with our Alex user account.

😊 Step 4: Trying out the Cross Account.

1. Now go back to your main account and login with your Alex user.
2. Once you are logged in then quickly go to your account B and from the roles copy the link to switch roles and paste in the Alex user.

The screenshot shows the AWS Console Home page. On the left, there's a sidebar with 'Recently visited' services (EC2, S3, RDS, Lambda) and a 'View all services' link. Below that is a 'Welcome to AWS' section with links for 'Getting started with AWS' and 'Training and certification'. On the right, there are two main panels: 'Applications (0)' and 'Cost and usage'. Both panels display a red box with the message 'Access denied'. The top right corner shows the region as 'Stockholm' and the user as 'CrossAccount-Alex @ 8788-9330-8172'.

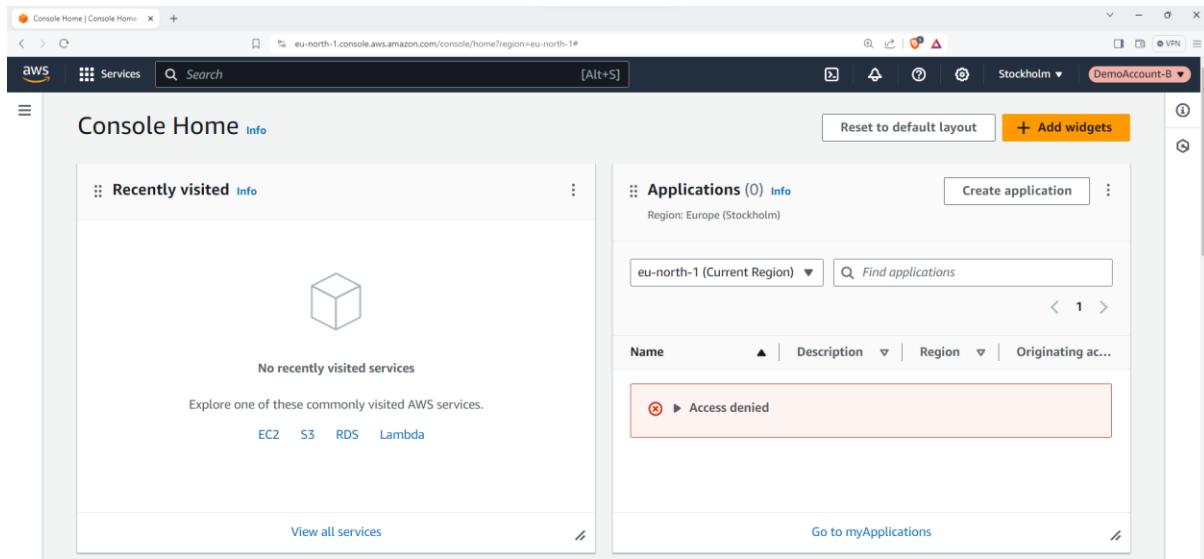
3. Once you have pasted it and press enter you will see a different tab.

The screenshot shows a browser window with three tabs open. The active tab is the middle one, which contains the URL <https://signin.aws.amazon.com/switchrole?roleName=CrossAccount-B&account=463646775279>. The other two tabs are blurred. The top right corner of the browser window shows the region as 'Stockholm' and the user as 'CrossAccount-Alex @ 8788-9330-8172'.

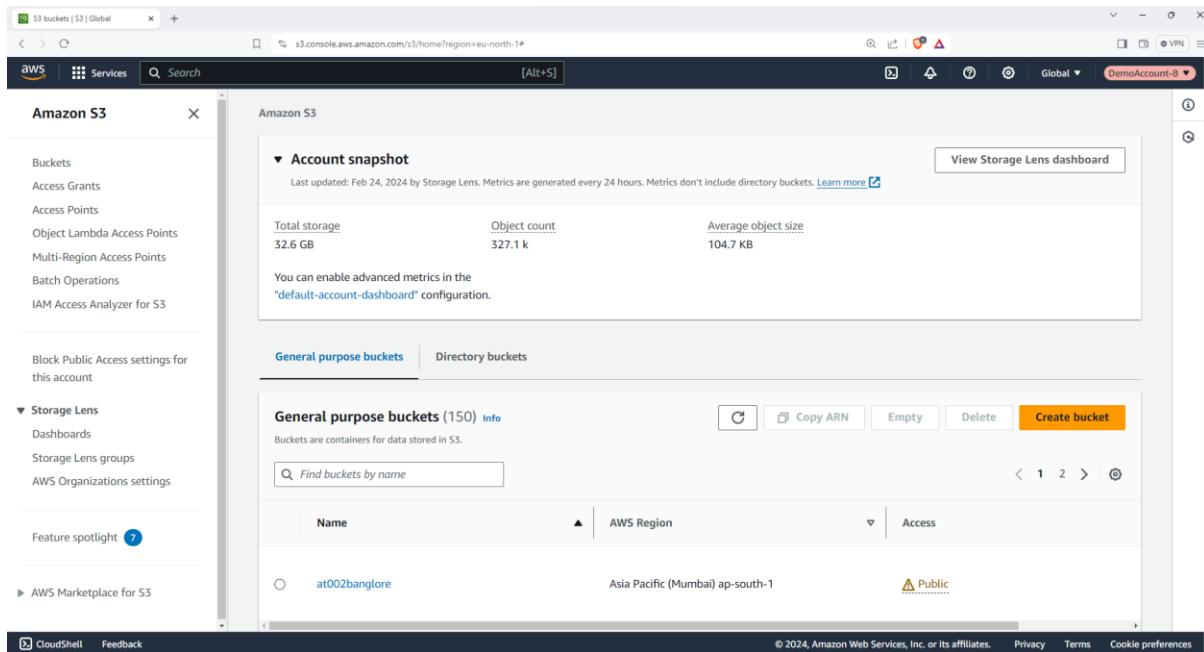
4. Here you are going to give it a display name click on Switch role.

The screenshot shows the 'Switch Role' page from the AWS website. It features a form with fields for 'Account' (463646775279), 'Role' (CrossAccount-B), and 'Display Name' (DemoAccount-B). Below the form is a color picker with several colored squares. At the bottom, there are buttons for 'Required', 'Cancel', and a blue 'Switch Role' button. The URL in the address bar is <https://sign-in.aws.amazon.com/signin?account=463646775279>.

5. Now you can see that Alex user is signed in to Account B. As of now I have provided it with S3 Read only access.
6. So, I will navigate to S3.



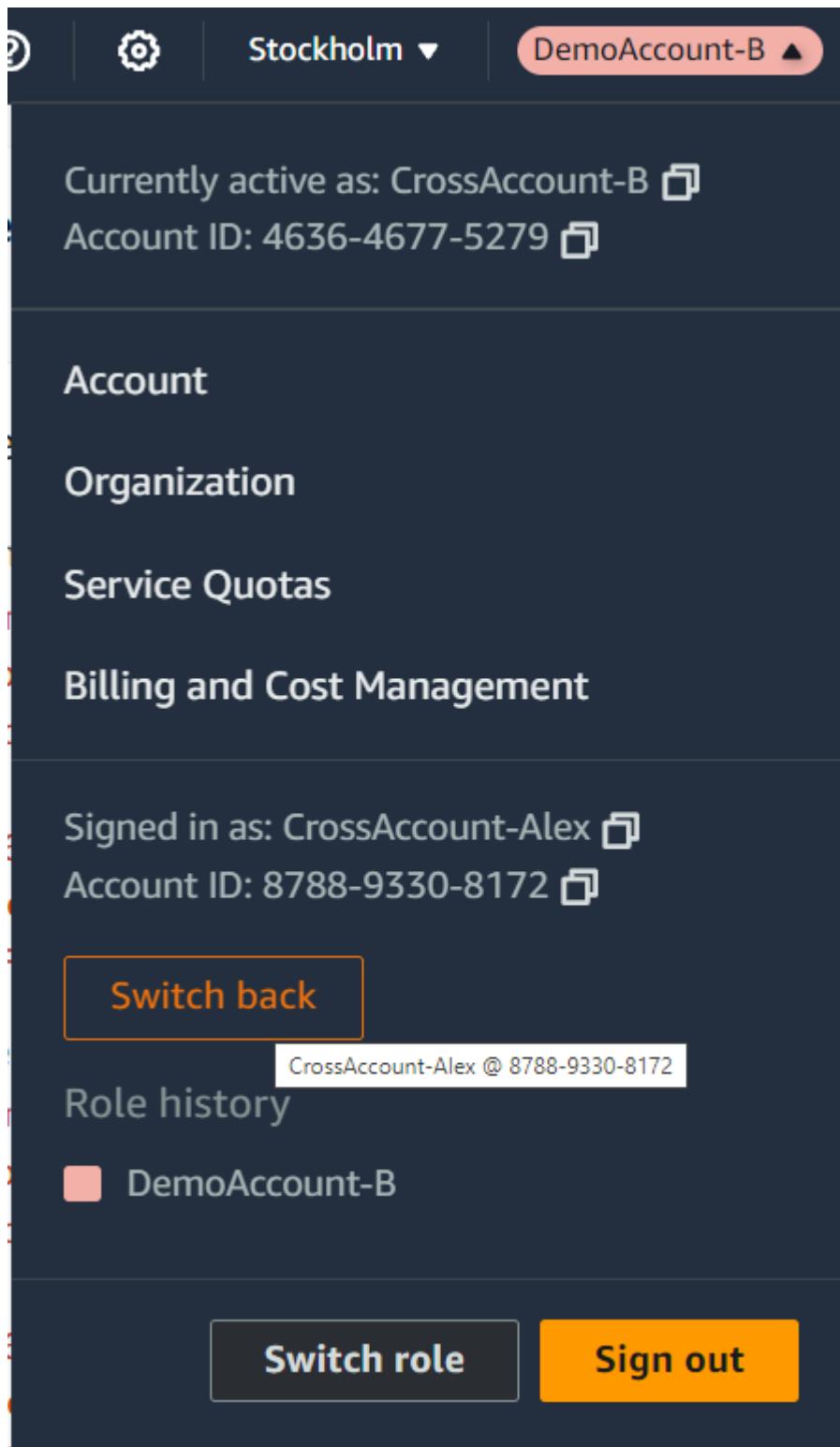
7. I can see all the buckets available as in read only mode using Alex user account.



8. But if this user Alex tries to open EC2 then he will get access denied because there is no permission for it.

The screenshot shows the AWS EC2 Dashboard for the Europe (Stockholm) Region. The left sidebar lists various EC2 management options like Instances, Images, and Elastic Block Store. The main area displays a grid of resources with counts and status indicators (e.g., 0 instances running, 0 auto scaling groups, etc.). A 'Launch instance' button is prominent. On the right, there's a 'Service health' section showing the AWS Health Dashboard and the current region as Europe (Stockholm). A sidebar on the right also shows EC2 Free Tier offers.

9. Now let suppose this user work has been done in this account now he can switch back to his account by clicking on switch back.



10. Now if you will just visit to S3 or EC2 in this account you will get access denied as well because this user Alex does not have sufficient permissions.