**Documentation of Cross-Account Access**

In this setup, we’ll enable **cross-account access** from a **source account** to a **target account** using:

1. **Switch Role** feature in the AWS Management Console.
2. **AWS Security Token Service (STS)** for temporary security credentials.

**Terminology**

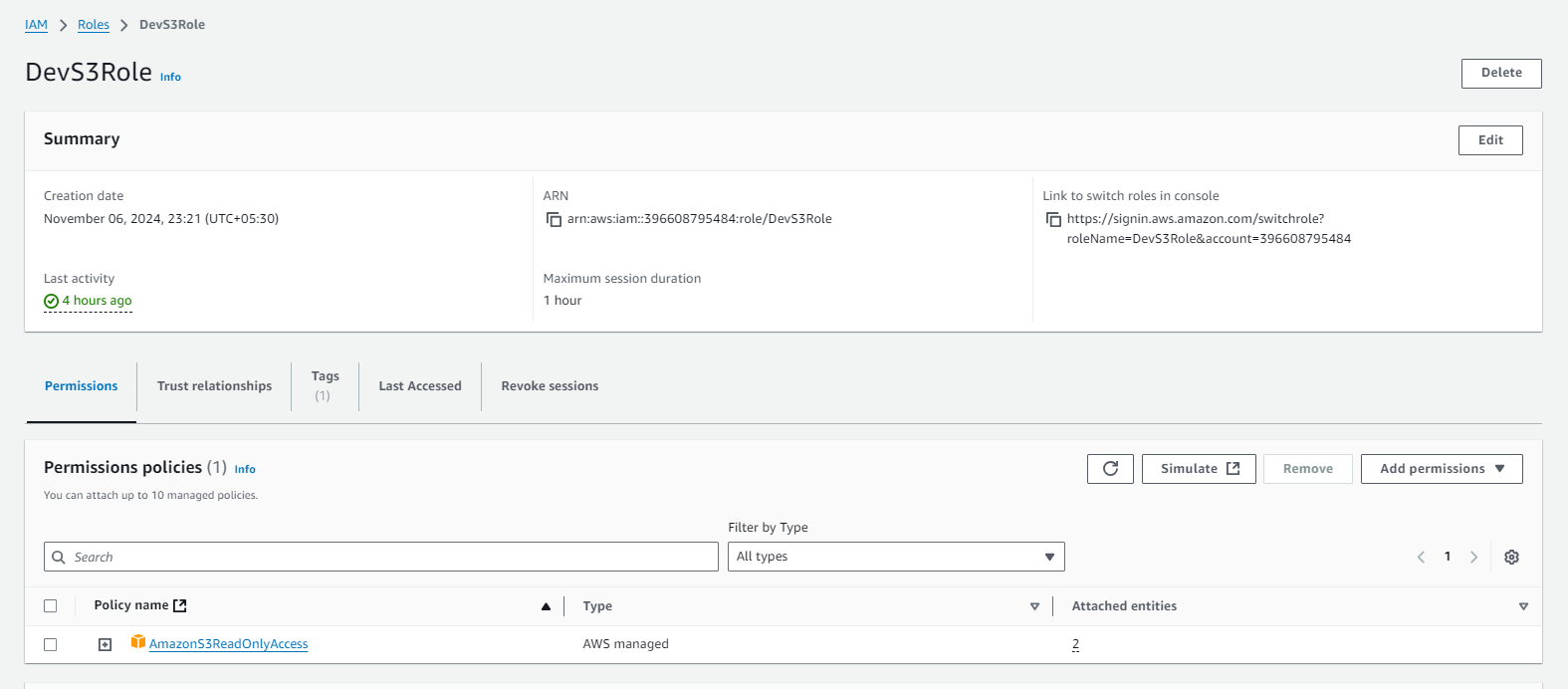
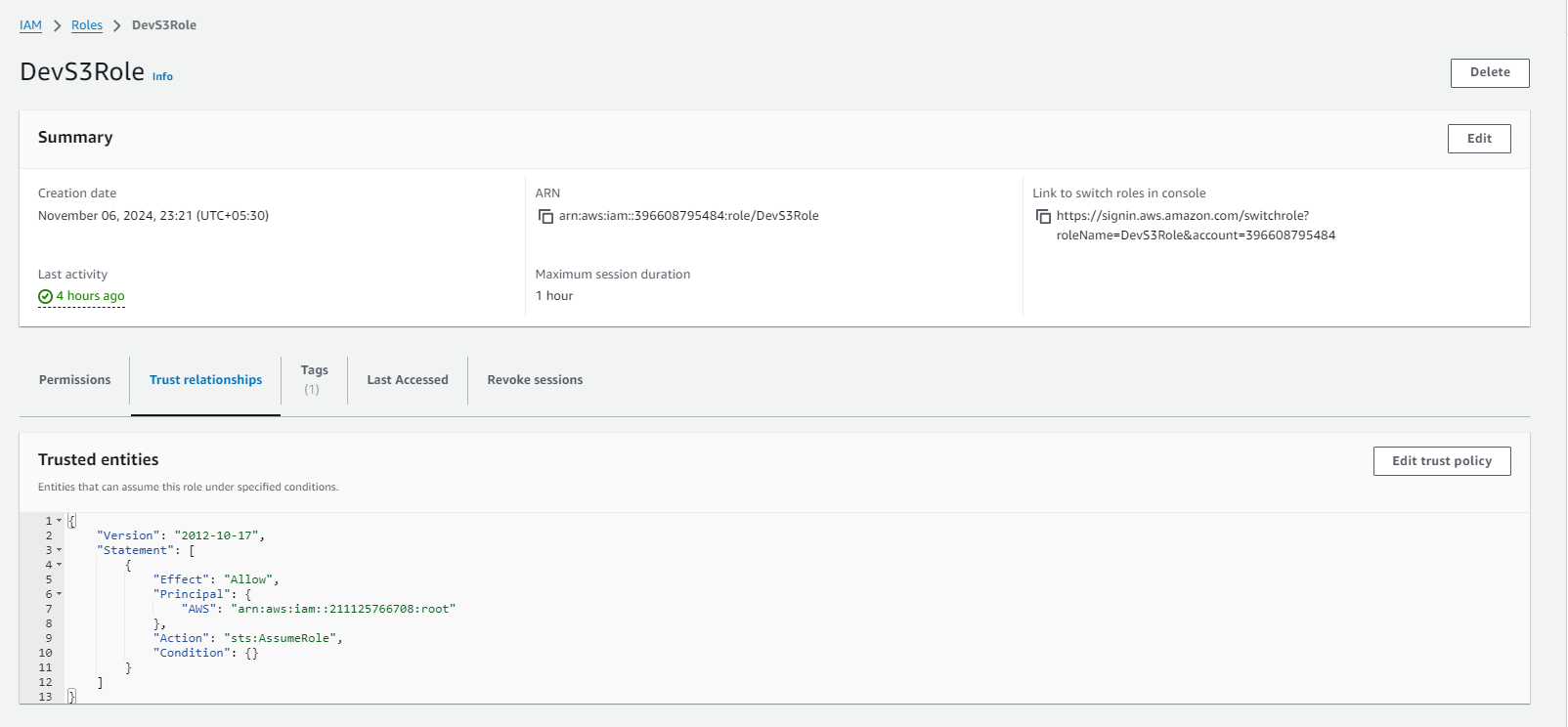
* **Source Account** (or **Trusted Account**): The account where the user (e.g., "Aayush") resides, who wants access to the target account's resources.
* **Target Account** (or **Trusting Account**): The account that contains the resources to which access is being granted.

**Exercise Objectives**

1. Enable cross-account access using AWS Management Console.
2. Enable cross-account access using AWS STS.
3. Access S3 buckets in the target account using both the Switch Role feature and the AWS CLI.

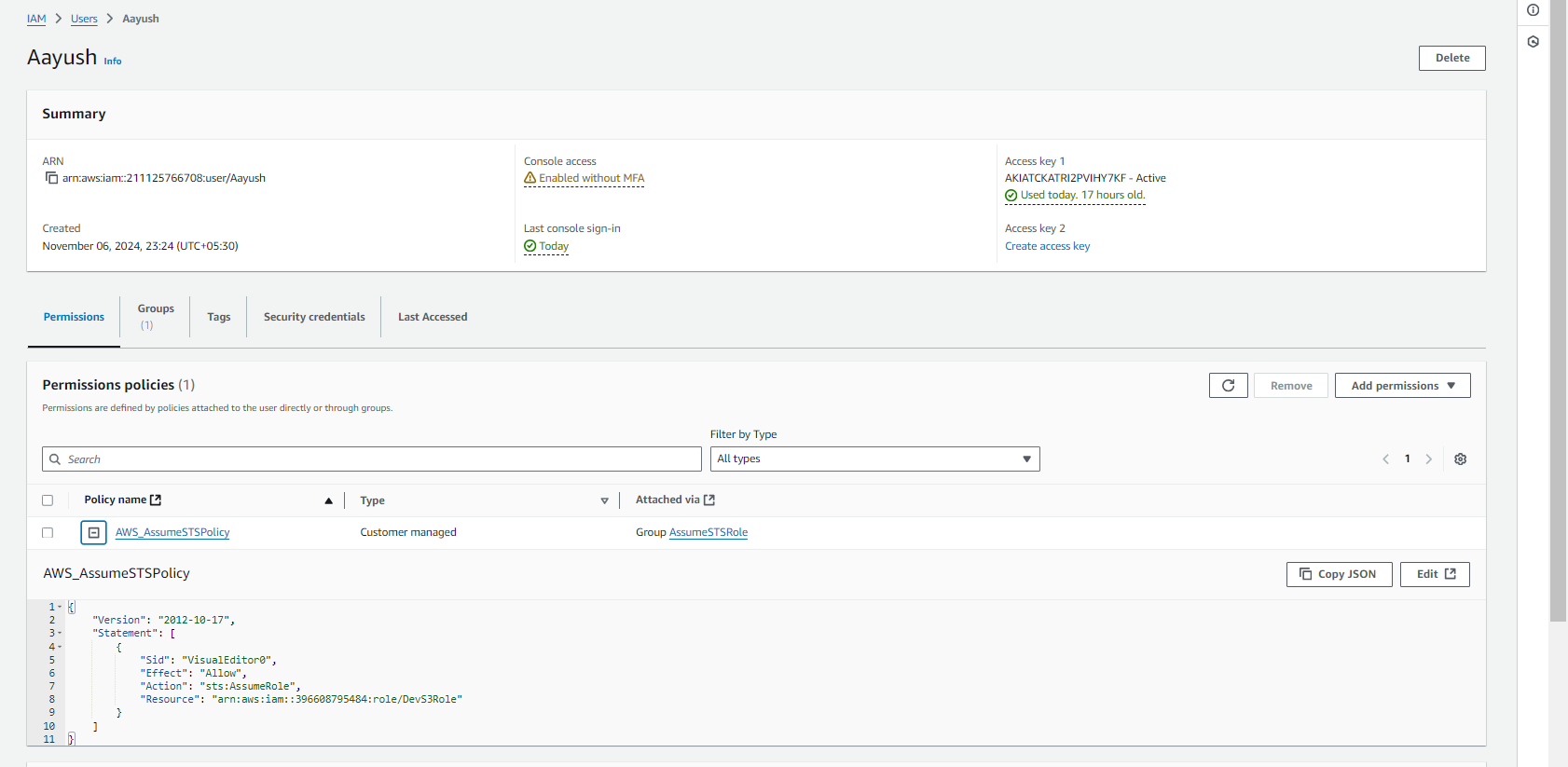
**Part 1: Configure IAM Role in Target Account**

1. **Login to Target Account**
   * Sign in to the AWS Management Console with permissions to create IAM roles in the target account.
2. **Create IAM Role in Target Account**
   * Go to the **IAM Console** and select **Roles** in the left navigation.
   * Click on **Create role**.
   * Select **AWS account** as the trusted entity type.
   * Choose **Another AWS account** and enter the **Account ID** of the source account.
3. **Assign Permissions to the Role**
   * For this exercise, we will allow read-only access to S3 resources in the target account.
   * Search for and select **AmazonS3ReadOnlyAccess**.
   * Click **Next**.
4. **Name and Tag the Role**
   * Enter a name for the role, e.g., DevS3Role.
   * Optionally, add a tag (e.g., Key: UsedBy, Value: DevTeam).
   * Click on **Create role**.
5. **Verify Role Creation**
   * In the target account, confirm the role DevS3Role has been created with the necessary permissions and trust policy for the source account.

  
  
  
Trust Policy  


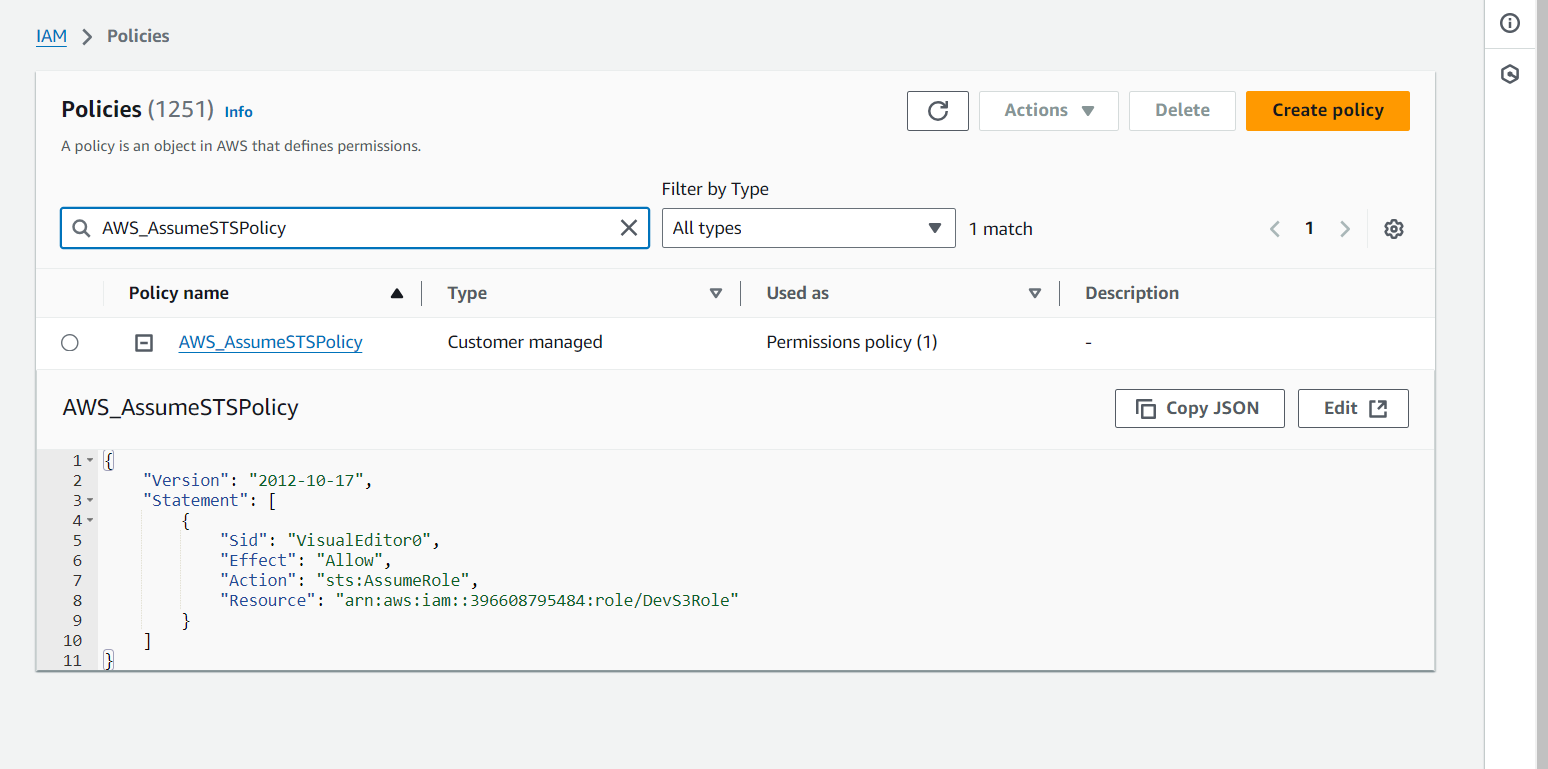
**Part 2: Set Up IAM User and Group in Source Account**

1. **Login to Source Account**
   * Sign in to the AWS Management Console in the source account.
2. **Create IAM User in Source Account**
   * Go to **IAM Console** and select **Users** from the left menu.
   * Click on **Create user**.
   * Enter a username, e.g., Aayush
   * Provide **AWS Management Console Access**.
   * Set a custom password and **uncheck** "User must create a new password at next sign-in" (for this exercise).
   * Click **Next: Permissions**.
3. **Create IAM Group and Policy**
   * Go to **IAM Groups** in the source account.
   * Create a new group, e.g., AssumeSTSRoleGroup.



* + Attach a policy allowing the user to assume the role created in the target account.

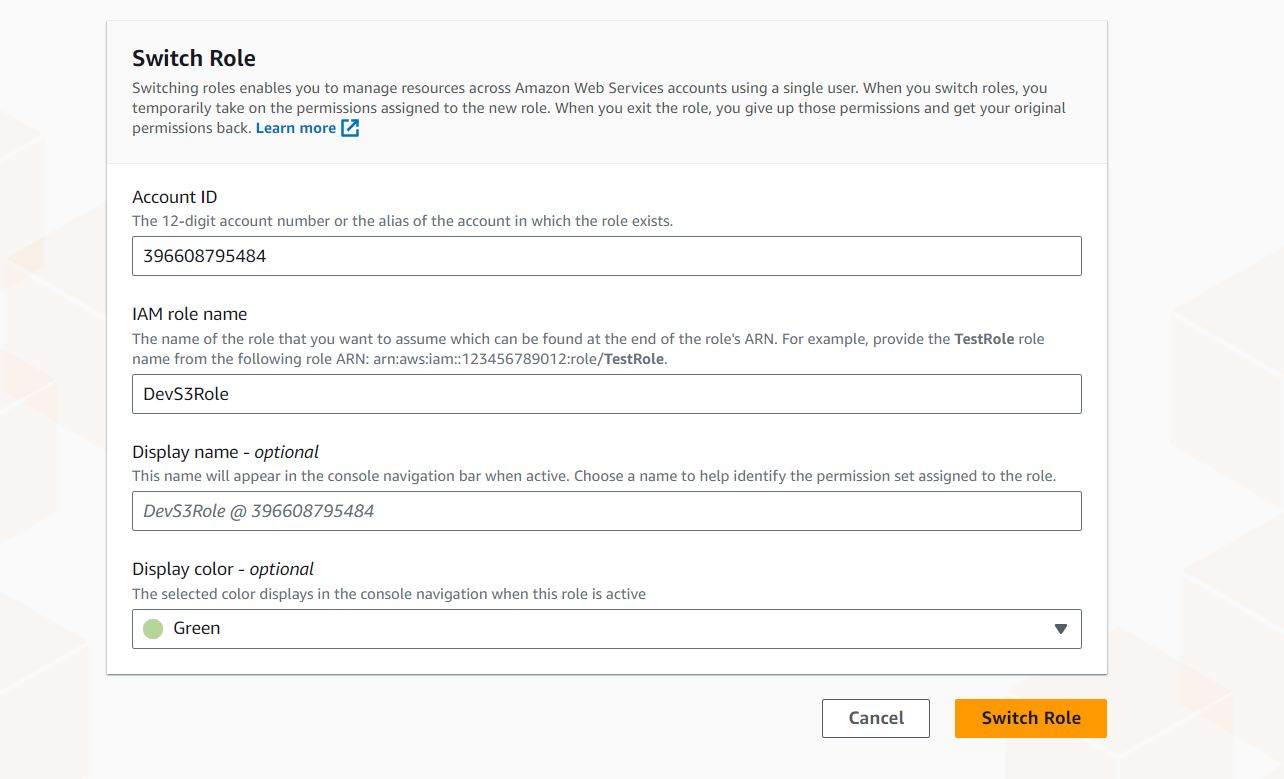
1. **Create Custom Policy for Role Assumption**
   * Go to **Policies** and click on **Create policy**.
   * Select **JSON** editor and paste the following policy:



1. **Attach Policy to Group**
   * Attach the newly created policy (AWSAssumeSTSRolePolicy) to the AssumeSTSRoleGroup.
2. **Add User to Group**
   * Go to **Users**, select Aayush, and add her to the group AssumeSTSRoleGroup.

**Part 3: Test Cross-Account Access Using Switch Role**

1. **Sign in as User in Source Account , in my case I have created Aayush user in my source account who will try to access the resources in my target account.**
   * Log out of the console, then log back in as Alice in the source account.
2. **Switch Role**
   * In the top-right corner of the console, select **Switch Role**.
   * Enter:
     + **Account ID** of the target account.
     + **Role name** as DevS3Role.
   * Click on **Switch Role**.



1. **Access S3 Buckets in Target Account**
   * After switching roles, go to **S3** and verify that you have read-only access to the S3 buckets in the target account.

**Part 4: Test Cross-Account Access Using AWS CLI and STS Token**

1. Create AWS Credentials Directory  
     
   Start by creating the AWS credentials directory on your local machine (if it doesn’t already exist):

**mkdir -p ~/.aws**

1. Edit AWS Credentials File  
     
   Open the credentials file to configure AWS CLI credentials for the IAM user in the source account.

**sudo nano ~/.aws/credentials**

* + Add your IAM user credentials in the following format:

[default]

aws\_access\_key\_id = <source\_account\_access\_key>

aws\_secret\_access\_key = <source\_account\_secret\_access\_key>

1. Assume Role Using AWS STS  
     
   Use the aws sts assume-role command to assume the IAM role in the target account and obtain temporary security credentials. Run:

aws sts assume-role --role-arn arn:aws:iam::396608795484:role/DevS3Role --role-session-name MySession

* + Replace arn:aws:iam::396608795484:role/DevS3Role with your specific Role ARN if different.
  + Make note of the output, which will include:
    - AccessKeyId
    - SecretAccessKey
    - SessionToken
  + Example output:

{

"Credentials": {

"AccessKeyId": "ASIAVYV5Z6NOFZ5V77CD",

"SecretAccessKey": "Mce2WQZrCecwRnPr6kve9rIbUDeq7cKa2QZTpdMj",

"SessionToken": "IQoJb3JpZ2luX2VjEKv//////////wEaCmV1LW5vcnRoLTEiSDBGAiEAnnmFYfkihYURRF4tcVwmouJzZI/m8EFAKaDVZXHdvJoCIQDui/HQunAKnGGohb6/EA4XPXz3PcT4V5y/E8iDUSgXPCqWAgg0EAAaDDM5NjYwODc5NTQ4NCIM3dZ5EBlF5xOB12+fKvMBzn6vTh76Al6Z5pwNDSbOw2DaXt5SRL4drGe5YcWyJaZqVlnakkHFeAmTMM",

"Expiration": "2024-11-07T12:00:00Z"

}

}

1. Set Environment Variables  
   Use the temporary security credentials to set up an environment with access to the target account resources:

export AWS\_ACCESS\_KEY\_ID="ASIAVYV5Z6NOFZ5V77CD"

export AWS\_SECRET\_ACCESS\_KEY="Mce2WQZrCecwRnPr6kve9rIbUDeq7cKa2QZTpdMj"

export AWS\_SESSION\_TOKEN="IQoJb3JpZ2luX2VjEKv//////////wEaCmV1LW5vcnRoLTEiSDBGAiEAnnmFYfkihYURRF4tcVwmouJzZI/m8EFAKaDVZXHdvJoCIQDui/HQunAKnGGohb6/EA4XPXz3PcT4V5y/E8iDUSgXPCqWAgg0EAAaDDM5NjYwODc5NTQ4NCIM3dZ5EBlF5xOB12+fKvMBzn6vTh76Al6Z5pwNDSbOw2DaXt5SRL4drGe5YcWyJaZqVlnakkHFeAmTMM"

* + Replace the values for AWS\_ACCESS\_KEY\_ID, AWS\_SECRET\_ACCESS\_KEY, and AWS\_SESSION\_TOKEN with those obtained in the previous step.

1. Access Target Account’s S3 Buckets  
   Test your temporary credentials by listing the S3 buckets in the target account:

aws s3 ls

* + This command should display the S3 buckets from the target account if the permissions are correctly configured.

1. Unset Environment Variables  
   Once you’re done, clear the temporary environment variables to avoid accidental use:

unset AWS\_ACCESS\_KEY\_ID

unset AWS\_SECRET\_ACCESS\_KEY

unset AWS\_SESSION\_TOKEN

Summary

By following these steps, you assume a role in the target account using AWS CLI and temporary security credentials (STS token). This approach allows controlled access to the target account's resources in a secure and temporary way.

**Summary of Steps**

1. **Create IAM Role in Target Account** with trust policy for the source account and S3 read-only permissions.
2. **Create IAM User and Group in Source Account** with permissions to assume the role in the target account.
3. **Test Access via Switch Role in Console** by logging in as the source account user and switching roles.
4. **Test Access via AWS CLI and STS** by assuming the role using AWS CLI, generating temporary credentials, and accessing resources in the target account.

This process allows users in the source account to access resources securely and temporarily in the target account, using both the **Switch Role** feature and **STS tokens** for CLI access.