Cross Account Repository Access: Steps for the IAM user in Persistent Account

Production Account : 931584077804

Developer Account ph 1: 871842996425

Developer Account ph 2 : 070874207938

**Follow these steps to configure the AWS CLI and Git for cross-account access**

1. Install python and pip, <https://www.python.org/downloads/>

When installing python use the **customize** installation

Check **add python to your PATH checkbox**

Optional features select everything

In Advanced options check **Install for all users** option.

1. If you have not already installed **git-remote-codecommit**, download it from [**git-remote-codecommit**](https://pypi.org/project/git-remote-codecommit/)on the Python Package Index website. You can run below command to install **git-remote-codecommit** from terminal after installing python. This is for amazon Webservice

**pip install git-remote-codecommit**

1. Install the AWS CLI on the local computer.

Download and run the AWS CLI MSI installer for Windows (64-bit):

<https://awscli.amazonaws.com/AWSCLIV2.msi>

See instructions for your operating system in [Installing the AWS CLI](https://docs.aws.amazon.com/cli/latest/userguide/installing.html).

1. Install Git on the local computer. To install Git, we recommend websites such as [Git Downloads](http://git-scm.com/downloads) or [Git for Windows](http://msysgit.github.io/).
2. After installing set username and email for the Git

git config --global user.name "*firstName LastName*"

git config --global user.email [*email@persistent.com*](mailto:email@persistent.com)

1. Create Group and Assign inline policy

{

"Version": "2012-10-17",

"Statement": [{

"Effect": "Allow",

"Action": ["sts:AssumeRole"],

"Resource": "arn:aws:iam::931584077804:role/PersistentCrossAccount"

}]

}

1. Create IAM user in the above group in Persistent AWS account, and get the Access keys

* Go to AWS console -> IAM -> user -> click on your username
* Then go to Security Credentials Tab
* Create or get the access key and secret access keys for your IAM user.

1. To set the AWS profiles

* Open terminal and run “**aws configure”** command
* When prompted, provide the following information:

AWS Access Key ID [None]: Your-IAM-User-Access-Key

AWS Secret Access Key ID [None]: Your-IAM-User-Secret-Access-Key

Default region name ID [None]: us-west-2

Default output format [None]: json

1. In a plain-text editor, open the config file, also known as the AWS CLI configuration file. Depending on your operating system, this file might be located at ~/.aws/config on Linux, macOS, or Unix, or at drive:\Users\USERNAME\.aws\config on Windows.

Add account ID to the profile configuration.

[default]

account=070874207938

region = us-west-2

output = json

Add account, role\_arn and source\_profile to the PersistentCrossAccount profile configuration.

[profile PersistentCrossAccount]

region = us-west-2

account = 931584077804

role\_arn = arn:aws:iam::931584077804:role/PersistentCrossAccount

source\_profile = default

output = json

1. Clone and access the CodeCommit repository in aws codecommit account

Follow similar pattern to clone the projects in codecommit

**git clone codecommit://PersistentCrossAccount@FileSystemConnector-prod**

**git clone codecommit://PersistentCrossAccount@GithubConnector-prod**

**git clone codecommit://PersistentCrossAccount@SlackConnector-prod**

**git clone codecommit://PersistentCrossAccount@WorkDocConnector-prod**

git clone codecommit://PersistentCrossAccount@[BoxConnector-prod](https://us-west-2.console.aws.amazon.com/codesuite/codecommit/repositories/BoxConnector-prod/browse?region=us-west-2) -b [dev\_perf\_test\_allChange](https://us-west-2.console.aws.amazon.com/codesuite/codecommit/repositories/BoxConnector-prod/browse/refs/heads/dev_perf_test_allChange?region=us-west-2)

Now at this point you are able to clone the projects and do any git operation on AWS code commit account.

**In order to get access to the AWS codecommit UI, do the below steps.**

1. Open terminal and run these commands first,

**pip install requests**

**pip install boto**

1. Run the attached python script using this command.

**python getAceess.py 931584077804 PersistentCrossAccount**

<https://git-codecommit.us-west-2.amazonaws.com/v1/repos/BoxConnector-prod>

**This will open the web browser and it will direct you to the aws codecommit account [931584077804]**

**Go to Codecommit in the aws console and there you must be able to view all the repositories listed there**.

**To build and run the persistent AWS kendra connectors.**

1. Pre-requisites:
   * AWS CLI (follow https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html)
   * npm (node package manager), install from https://www.npmjs.com/get-npm
   * cdk version 1.90 (just do npm i -g aws-cdk@1.90.0)
   * Amazon Corretto 11 (Open JDK 11) from https://docs.aws.amazon.com/corretto/latest/corretto-11-ug/what-is-corretto-11.html
   * maven installed to dev machine

npm i -g aws-cdk@1.90.0

* Run below command and give the credentials when prompt.

**aws configure --profile connector-sdk**

Access key ID: AKIARBADOS3BA3UUVWGW

Secret access key: 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

* In order to compile connector code, you must have credentials to download core-sdk code hosted in Kendra’s private maven repository.
* To use it run:

**set CODEARTIFACT\_TOKEN=$(aws codeartifact get-authorization-token --domain sdkcore-alpha --domain-owner 475788766865 --query authorizationToken --output text --profile PersistentCrossAccount)**

**Note🡺need to run project folder every 12 hours generate token then past in setting=>inside password**

**${env.CODEARTIFACT\_TOKEN}**

* Connector SDK connectors requires JAVA\_HOME to be set up correctly.
* Cd to cd connector-prod/aws-kendra-connector

Now run **mvn -s settings.xml clean install** to compile code

You might face issues in referring to core-sdk code from your IDE. To fix that, add following to ~/.m2/settings.xml:

<servers>

<server>

<id>codeartifact</id>

<username>aws</username>

<password>${env.CODEARTIFACT\_TOKEN}</password>

</server>

</servers>

And replace ${env.CODEARTIFACT\_TOKEN} with actual value of that variable.

To get the actual value of CODEARTIFACT\_TOKEN run this command.

**aws codeartifact get-authorization-token --domain sdkcore-alpha --domain-owner 475788766865 --query authorizationToken --output text --profile PersistentCrossAccount**

Remember, this ${env.CODEARTIFACT\_TOKEN} needs to be refreshed every 12 hours.

Above step-by-step instructions you can find from AWS documentation.

<https://docs.aws.amazon.com/codecommit/latest/userguide/cross-account-user-b.html>

<https://aws.amazon.com/blogs/security/how-to-enable-cross-account-access-to-the-aws-management-console/>

1. Import pom.xml in intellij

Wait for the dependencies to load

1. Check that you are not on the main or master branch
2. mvn -s settings.xml -Dcheckstyle.skip -DskipTests=true -Dspotbugs.skip=true clean install

or mvn clean install -s settings.xml -Dcheckstyle.skip=true

1. java -Xdebug -Xrunjdwp:transport=dt\_socket,server=y,suspend=n,address=8000 -Dserver.port=8084 -jar -DRegion=us-west-2 -DIsHostedSetup=true -DRegionName=us-west-2 target/aws-kendra-github-connector-1.0-SNAPSHOT-exec.jar

== this is for connector running cmd

or

java -jar -DRegion=us-west-2 -DRegionName=us-west-2 -DIsHostedSetup=true -Dserver.port=8084 aws-kendra-github-connector-1.0-SNAPSHOT-exec.jar

Graphical user interface, text

Description automatically generated

1. on Intellij run-> edit config-> template ->select remote jvm debug ->create configuration ,

name the config

set the port eg 8084

1. open Postman
2. Check Config Api for the following

* Repository id should be same as the datasource id in ur kendra
* U should have a proper iam role data source iam role arn

eg: arn:aws:iam::871842996425:role/KendraAccessRoleKamlesh

* Check for the correct Host name and Organisation name
* Check for git hub token
* Index id should be checked from the aws index eg GithubIndex
* Check for the proper inclusion and exclusion patterns
* Check the port on which u send the request and the url eg:localhost:8084/configure
* Hit send
* Graphical user interface, text, application, email

  Description automatically generated

Graphical user interface, text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

1. Check Start sync job for proper

* "awsAccountId": "871842996425",
* "indexId": "same as in configure api above specified",
* "dataSourceId": " same as in configure api above specified "
* Send request
* Graphical user interface, text, application, email

  Description automatically generated

1. Get Status –

Check parameters or url for correct indexid, and datasource id

Hit send

In response body check for succeeded message

* Eg: "connectorSyncStatus": "SUCCEEDED",
* "errorCode": null,

**FAQ:** 😉

1. I see an error: git: 'remote-codecommit' is not a git command

You might need to add **git-remote-codecommit** executable path to the PATH variable.

C:\Users\bhagya\_peiris\AppData\Roaming\Python\Python310\Scripts

Graphical user interface, text, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated