

Migration: GIT repository to AWS

Objective:

Migrate a sample repository to AWS Code commit repository. The following steps need to be implemented in order to complete the migration.

Step 0: Setup required for access to codecommit

Step 1: CodeCommit repository creation

Step 2: Clone the repository and push to CodeCommit Repository

Step 3: View files in CodeCommit

Step 4: Share CodeCommit repository

Step 0: Setup process requirements for AWS access to code commit

- Create and configure IAM user for access to AWS code commit.
 - Configure your local computer for access.
 - Install AWS CLI to manage code commit.
-
1. Create an IAM user. Ensure that there is an access key ID and a secret access key associated with that IAM user. In the IAM console, in the navigation pane, choose **Users**, and then choose the IAM user you want to configure for CodeCommit access.
 2. On the **Permissions** tab, select **Add Permissions**.
 3. In **Grant permissions**, select **Attach existing policies directly**.
 4. From the list of policies, select **AWSCodeCommitFullAccess** or another managed policy for CodeCommit access. After you have selected the policy you want to attach, select **Add permissions**.

Add user

1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

[+ Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

- Access type* ☒ **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- ☒ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password* ☐ Autogenerated password
☒ Custom password


☐ Show password


☒ Require password reset User must create a new password at next sign in.


* Required

[Cancel](#)

[Next: Permissions](#)









 Add user to group

 Copy permissions from existing user

 Attach existing policies directly

[Create policy](#)



Filter policies <input type="text" value="awscode"/>		Showing 20 results	
	Policy name	Type	Used as
<input type="checkbox"/>	 AmazonEC2RoleforAWSCodeDeploy	AWS managed	None
<input type="checkbox"/>	 AWSCodeBuildAdminAccess	AWS managed	None
<input type="checkbox"/>	 AWSCodeBuildDeveloperAccess	AWS managed	None
<input type="checkbox"/>	 AWSCodeBuildReadOnlyAccess	AWS managed	None
<input checked="" type="checkbox"/>	 AWSCodeCommitFullAccess	AWS managed	Permissions policy (2)
<input type="checkbox"/>	 AWSCodeCommitPowerUser	AWS managed	None
<input type="checkbox"/>	 AWSCodeCommitReadOnly	AWS managed	None
<input type="checkbox"/>	 AWSCodeDeployDeployerAccess	AWS managed	None

[Set permissions boundary](#)

[Cancel](#)

[Previous](#)

[Next: Tags](#)

Add user

1 2 3 4 5

Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
<input type="text" value="devops"/>	<input type="text"/>	<input type="button" value="x"/>
<input type="button" value="Add new key"/>	<input type="text"/>	

You can add 49 more tags.

Showing 1 result

<input type="checkbox"/>	User name	Groups	Access key age	Password age	Last activity	MFA
<input type="checkbox"/>	devops	None	None	Today	None	Not enabled

Configure and Install AWS CLI on Windows

1. On your local machine, download and install the AWS CLI(version 1.7.38 and later), as this is a prerequisite for interacting with CodeCommit from the command line.

AWS Command Line Interface v2 Setup

End-User License Agreement

Please read the following license agreement carefully



AWS Command Line Interface

Copyright 2012-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with the License. A copy of the License is located at

<http://aws.amazon.com/apache2.0/>

☒ I accept the terms in the License Agreement

Print

Back

Next

Cancel

```
: \Program Files\Amazon\AWSCLIV2>aws2 --version
aws-cli/2.0.0dev0 Python/3.7.5 Windows/10 botocore/2.0.0dev0

: \Program Files\Amazon\AWSCLIV2>
```

2. In order to verify that the CodeCommit commands for the AWS CLI are installed, run the following command.
3. `aws2 codecommit help`

```
Command Prompt - aws2 codecommit help
C:\Program Files\Amazon\AWSCLIV2>
C:\Program Files\Amazon\AWSCLIV2>aws2 codecommit help

codecommit
^^^^^^^^^^

Description
*****

This is the *AWS CodeCommit API Reference* . This reference provides
descriptions of the operations and data types for AWS CodeCommit API
along with usage examples.

You can use the AWS CodeCommit API to work with the following objects:

Repositories, by calling the following:

* BatchGetRepositories , which returns information about one or more
  repositories associated with your AWS account.

* CreateRepository , which creates an AWS CodeCommit repository.

* DeleteRepository , which deletes an AWS CodeCommit repository.

* GetRepository , which returns information about a specified
  repository.

* ListRepositories , which lists all AWS CodeCommit repositories
-- More --
```

4. A list of CodeCommit commands will be generated by the command mentioned above.
5. AWS CLI must be configured with the following configure command.
6. `aws2 configure`
7. AWS access key and AWS secret access key of the IAM user to use with CodeCommit must be specified when prompted.. Also, specify the AWS Region where the repository exists, such as `us-east-2`. When prompted for the default output format, specify `json`.

```
C:\Program Files\Amazon\AWSCLIV2>aws2 configure
AWS Access Key ID [None]: AKIAWLBU4WURMJ5N2NPW
AWS Secret Access Key [None]: Lf7WcEfDCfYcDOEoQEUIcGcVQEaUL1fORLu9m7f
Default region name [None]: us-east-2
Default output format [None]: json

C:\Program Files\Amazon\AWSCLIV2>
```

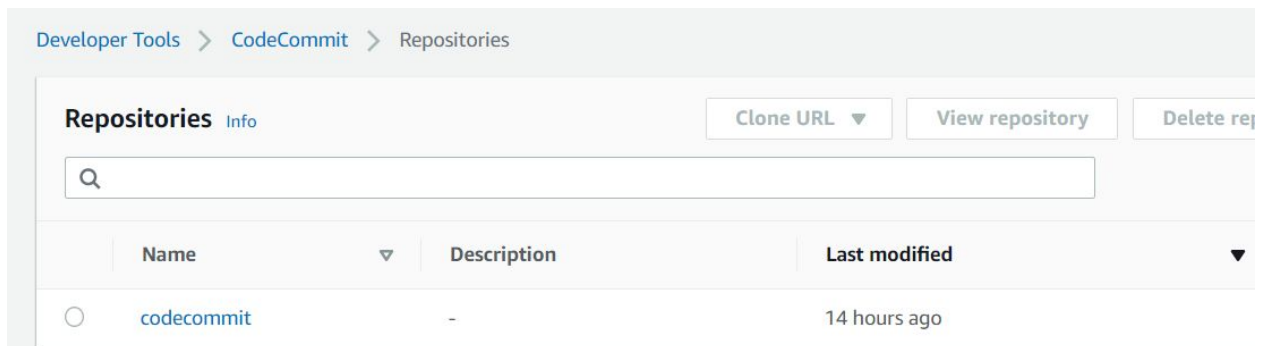
8. In order to connect to a repository in another AWS Region, AWS CLI must be reconfigured with the default Region name.

Git Installation on windows:

- In order to work with files, commits etc. in CodeCommit repositories, Git (version 1.7.9 or higher) must be installed from Git (<https://git-scm.com/download/win>) for Windows website on to the local machine.
- In order to complete setup configure Git credentials for CodeCommit (HTTPS, recommended for most users), an SSH key pair to use when accessing CodeCommit (SSH), or the credential helper included in the AWS CLI (HTTPS) as both HTTPS and SSH authentication are supported by CodeCommit
- When prompted during the **Adjusting your PATH environment** step, choose the option to use Git from the command line.
- (Optional) If HTTPS option is selected with the credential helper that is included in the AWS CLI instead of configuring Git credentials for CodeCommit, on the **Configuring extra options** page, ensure that the **Enable Git Credential Manager** option is cleared. The Git Credential Manager is only compatible with CodeCommit if IAM users configure Git credentials.

Step 1: CodeCommit repository creation

1. On the **Repositories** page of the AWS Management console, choose **Create repository**.
2. On the **Create repository** page, in **Repository name**, enter a name for the repository for example. CodeCommit.
3. Choose **Create**.



Step 2: Clone the Repository and Push to the CodeCommit Repository

Clone a Git repository to the local computer called local repo and push the contents of the local repo to CodeCommit repository.

From the command prompt run the **git clone** command with the `--mirror` option to clone a bare copy of the remote repository into a new folder named `aws-codecommit-demo`. This is meant only for migration. It is not the local repo for interacting with the migrated repository in CodeCommit. You can create that later, after the migration to CodeCommit is complete.

The following example clones a demo application hosted on GitHub (<https://github.com/aws-labs/aws-demo-php-simple-app.git>) to a local repo in a directory named `aws-codecommit-demo`.


```
mohan@LAPTOP-65TT095D MINGW64 ~  
$ git clone --mirror https://github.com/awslabs/aws-demo-php-simple-app.git aws-codecommit-demo  
Cloning into bare repository 'aws-codecommit-demo'...  
remote: Enumerating objects: 221, done.  
remote: Total 221 (delta 0), reused 0 (delta 0), pack-reused 221  
Receiving objects: 100% (221/221), 816.98 KiB | 3.25 MiB/s, done.  
Resolving deltas: 100% (95/95), done.
```

1. Change directories to the directory where you made the clone.

```
mohan@LAPTOP-65TT095D MINGW64 ~  
$ cd aws-codecommit-demo  
  
mohan@LAPTOP-65TT095D MINGW64 ~/aws-codecommit-demo (BARE:master)  
$ |
```

2. Run the **git push** command, specifying the URL(step 1)and name of the destination CodeCommit repository and the **--all** option.

```
mohan@LAPTOP-65TT095D MINGW64 ~/aws-codecommit-demo (BARE:master)  
$ git push https://git-codecommit.us-east-2.amazonaws.com/v1/repos/codecommit --all
```

In order to create a local repo with all the correct references for working with the repository in CodeCommit, run the `git clone` command without the `--mirror` option:

```
mohan@LAPTOP-65TT095D MINGW64 ~/aws-codecommit-demo (BARE:master)  
$ git clone https://git-codecommit.us-east-2.amazonaws.com/v1/repos/codecommit  
Cloning into 'codecommit'...
```

Step 3: View Files in CodeCommit

After you have pushed the contents of your directory, you can use the CodeCommit console to quickly view all of the files in that repository.

1. Open the CodeCommit console, In **Repositories**, choose the name of the repository (for example, *codecommit*).
2. View the files in the repository for the branches, the clone URLs, the settings, and more.



	Name
	appspec.yml
	duplicate.py
	get-differences.normal.json

Step 4: Share CodeCommit Repository

When a repository is created in CodeCommit, two endpoints are generated: one for HTTPS connections and one for SSH connections and both provide secure connections over a network. Users can use either protocol and both endpoints remain active no matter which protocol you recommend to your users. IAM policies that allow other users access to the repository must be created before the repository is shared along with access instructions

Customer managed policy for your repository must be created

1. Open the AWS console and in the **Dashboard** navigation area, choose **Policies**, and then choose **Create Policy**.

2. On the **Create Policy** page, next to **Copy an AWS Managed Policy**, choose **Select**.
3. On the **Copy an AWS Managed Policy** page, in **Search Policies**, enter **AWSCodeCommitPowerUser**. Choose **Select** next to the policy name.
4. On the **Review Policy** page, in **Policy Name**, enter a new name for the policy (for example, *AWSCodeCommitPowerUser-codecommit*).
In **Policy Document**, replace the "*" portion of the `Resource` line with the Amazon Resource Name (ARN) of the CodeCommit repository, as shown here:Repository ARN
5. "Resources": arn:aws:codecommit:us-east-2:436050113826:codecommit

Choose **Validate Policy**. After the policy is validated, choose **Create Policy**.

In order to manage access to the repository, create an IAM group for its users, add IAM users to that group, and then attach the customer managed policy.

Attach any other policies required for access, such as `IAMUserSSHKeys` or `IAMSelfManageServiceSpecificCredentials`.

1. In the **Dashboard** navigation area of the AWS Management console, select **Groups**, and then choose **Create New Group**.
2. On the **Set Group Name** page, in **Group Name**, enter a name for the group and then choose **Next Step**.
3. On the **Review** page, choose **Create Group**. IAM creates this group with the specified policies already attached. The group appears in the list of groups associated with your AWS account.
4. Select the group from the list.
5. On the group summary page, Select the **Users** tab on group summary page, and then choose **Add Users to Group**. Select the boxes next to the users to whom you want to allow access to the CodeCommit repository, and then choose **Add Users** and close.

The information required to connect to the repository must be shared with the IAM user.

1. On the **Repositories** page of codecommit, select the name of the repository to be shared.
2. In **Clone URL**, select HTTPS protocol and send IAM users the clone URL along with any other instructions, such as installing the AWS CLI, configuring a profile, or installing Git.

References:

Aazon aws

aws.amazon.com