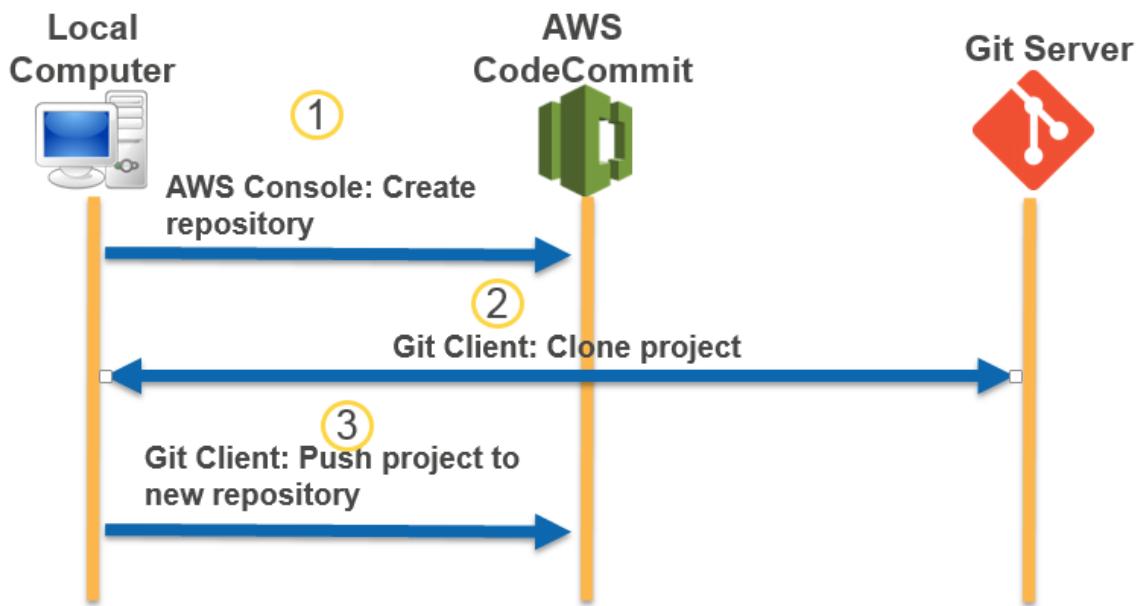


# Project



## Migrate a Git repository to AWS CodeCommit

24.10.2023

Himanshu Nimje  
Nagpur-440017

## Overview

AWS CodeCommit is a fully managed source control service that makes it easy for teams to host secure and scalable Git repositories. It is part of the AWS DevOps tool suite, providing a secure and highly available platform for storing and managing source code.

## Benefits

AWS CodeCommit is a fully-managed source control service that hosts secure Git-based repositories. CodeCommit eliminates the need to operate your own source control system or worry about scaling its infrastructure

AWS CodeCommit Fully-managed Secure and automatically encrypts your files in transit and at rest. CodeCommit is integrated with AWS identity and Access Management (IAM) allowing you to customize user-specific access to your repositories.

AWS CodeCommit supports all Git commands and works with your existing Git tools. You can keep using your preferred development environment plugins, continuous integration/continuous delivery systems, and graphical clients with CodeCommit

## Service Used

AWS CodeCommit

AWS IAM

## Steps

1. Set up access to CodeCommit. This includes creating an IAM user for CodeCommit and configuring your computer for access.
2. Create a CodeCommit repository.
3. Clone the repository to your local machine.
4. Push the cloned repository to CodeCommit.
5. View the files in the CodeCommit repository.

## Implementation

## Go to CodeCommit

The screenshot shows the AWS Console search results for 'code commit'. The 'CodeCommit' service card, which describes it as 'Store Code in Private Git Repositories', is highlighted with an orange box. Other services listed include Amazon CodeGuru, Amazon CodeWhisperer, and CodePipeline.

## Create Repository

The screenshot shows the AWS CodeCommit 'Repositories' page. The 'Create repository' button is highlighted with an orange box. The left sidebar shows navigation options like 'Source', 'Artifacts', 'Build', 'Deploy', 'Pipeline', and 'Settings'. The main area displays a table of repositories with columns for Name, Description, Last modified, and Clone URL.

Assigned Name to the Repository (firstrepo) and click on the Create Option

Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

**Repository settings**

Repository name  
firstrepo

Description - optional  
firstrepo on AWS

Tags  
Add tag

Cancel **Create**

Repository is created successfully, go inside to the repository

Success  
Repository successfully created

Create a notification rule for this repository

Developer Tools > CodeCommit > Repositories > firstrepo

firstrepo

Clone URL

CodeCommit

Code

Pull requests

Commits

Branches

Git tags

Settings

Approval rule templates

Artifacts • CodeArtifact

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Settings

Go to resource

CloudShell Feedback

HTTPS SSH HTTPS (GRC)

⚠ You are signed in using a root account. You cannot configure SSH connections for a root account, and HTTPS connections for a root account are not recommended. Consider signing in as an IAM user and then setting up your connection.

**Step 1: Prerequisites**

You must use a Git client that supports Git version 1.7.9 or later to connect to an AWS CodeCommit repository. If you do not have a Git client, you can install one from Git downloads. [View Git downloads page](#)

You must have an AWS CodeCommit managed policy attached to your IAM user, belong to a CodeStar project team, or have the equivalent permissions. [Learn how to create and configure an IAM user for accessing AWS CodeCommit](#) | [Learn how to add team members to an AWS CodeStar Project](#)

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Create file

You must have an AWS CodeCommit managed policy attached to your IAM user, belong to a CodeStar project team, or have the equivalent permissions. [Learn how to create and configure an IAM user for accessing AWS CodeCommit.](#) [Learn how to add team members to an AWS CodeStar Project.](#)

**Step 2: Set up the AWS CLI Credential Helper**

Set up your connection to AWS CodeCommit repositories using the credential helper included in the AWS CLI. This is the only connection method for AWS CodeCommit repositories that does not require an IAM user, so it is the only method that supports root access, federated access, and temporary credentials. [Learn more](#)

**Additional details**

You can find more detailed instructions in the documentation. [View documentation](#)

**firstrepo Info**

Add file ▾

Name

Empty repository

Your repository is currently empty. You can add files to it directly from the console or by cloning the repository to your local computer, creating commits, and pushing content to the remote repository in AWS CodeCommit.

**Create file**

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Developer Tools > CodeCommit > Repositories > firstrepo > File

## Create a file

**firstrepo Info**

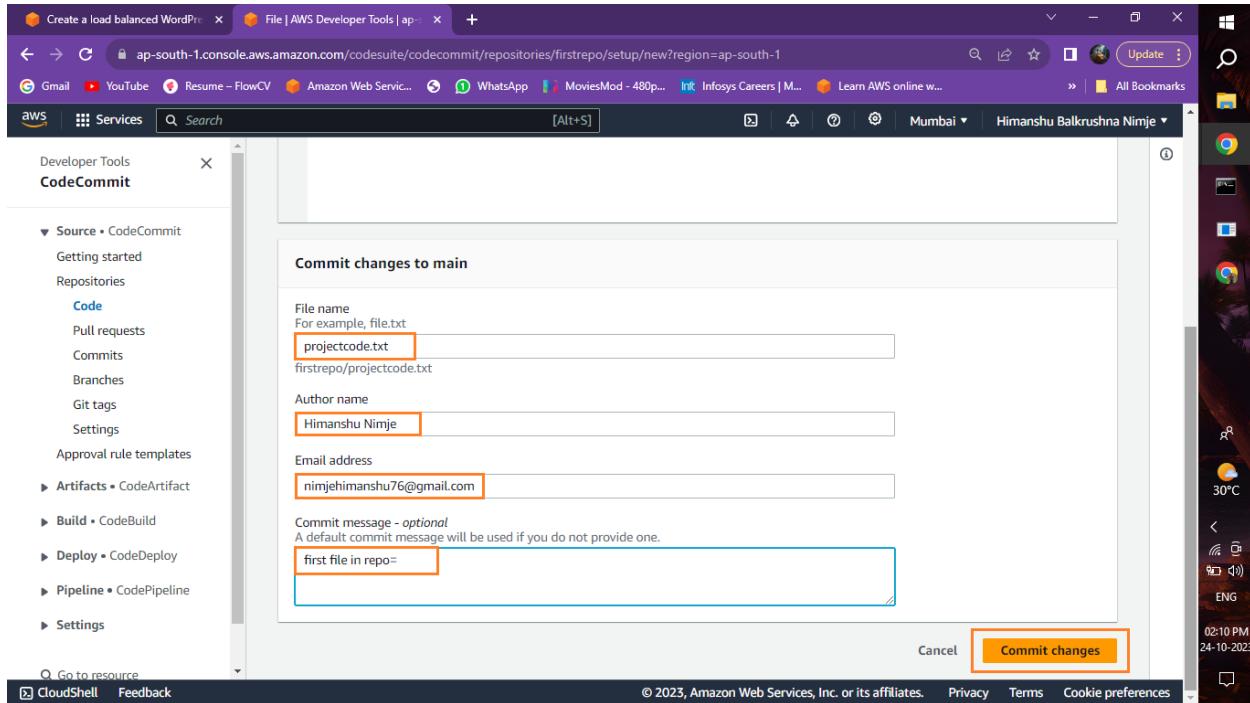
1 Welcome to my world.

**Commit changes to main**

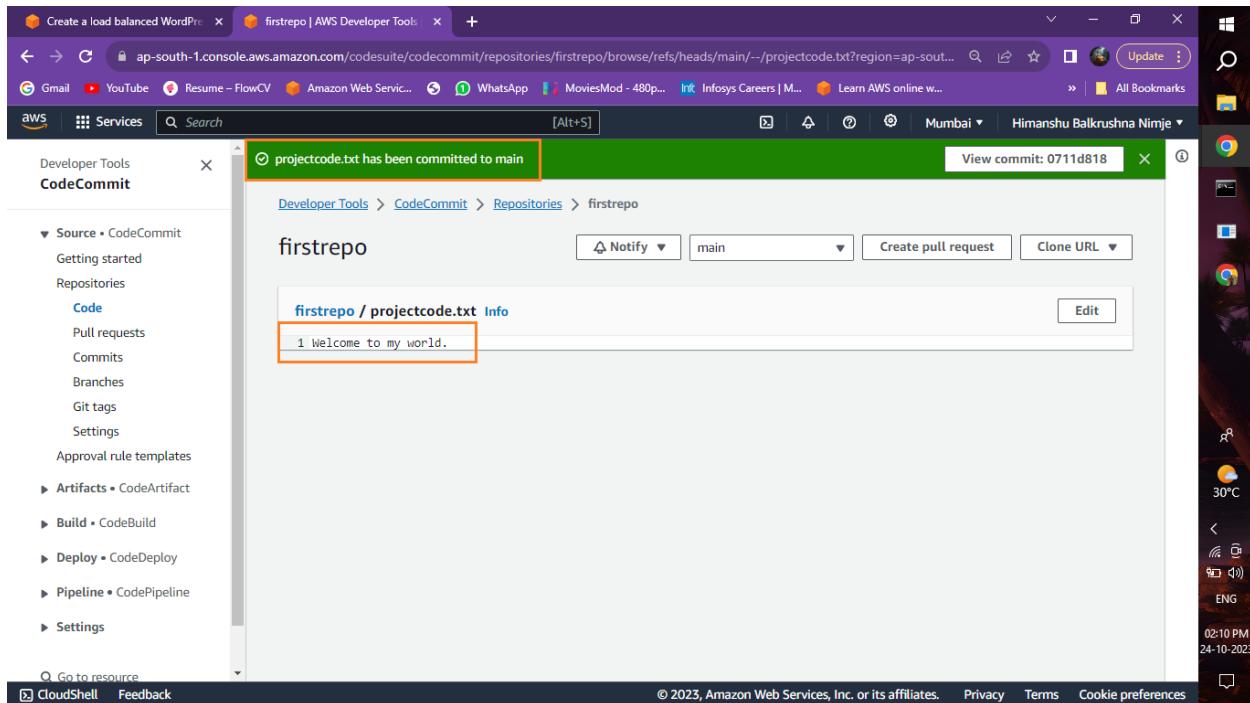
File name  
For example, file.txt

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Assign name to the File (projectcode.txt) and Click on the Commit Changes



File has been Committed Successfully



## Open new tab for IAM

The screenshot shows the AWS Management Console search results for 'iam'. The 'IAM' service card is highlighted with an orange box. The search bar at the top contains 'iam'. The left sidebar shows 'Services (10)' with various options like Lightsail, EC2, Billing, API Gateway, DynamoDB, Lambda, and AWS Amplify. The right sidebar displays a 'Health' dashboard with various metrics and a link to 'AWS Health'.

## Create user

The screenshot shows the AWS Management Console IAM Users page. The 'Create user' button is highlighted with an orange box. The left sidebar shows 'Identity and Access Management (IAM)' with 'Users' selected. The main area displays a table with columns: User name, Path, Group, Last activity, MFA, Password age, and Consolidate. A message states 'No resources to display'.

Click on Next

User details

User name  
Himanshu

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 Keyspaces, or a backup credential for emergency account access.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel **Next**

Select→Attach Policies directly

Step 1 [Specify user details](#)

Step 2 **Set permissions**

Step 3 [Review and create](#)

**Set permissions**

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Add user to group  
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions  
Copy all group memberships, attached managed policies, and inline policies from an existing user.

Attach policies directly  
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

**Permissions policies (1138)**

Choose one or more policies to attach to your new user.

Policy name	Type	Attached entities
AccessAnalyzerServiceRolePolicy	AWS managed	0

Filter by Type

Search All types

Create policy

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Select required Policies and click to create user

The screenshot shows the 'Create user' step in the AWS IAM console. The 'User details' section is filled with a user named 'Himanshu'. In the 'Permissions summary' section, several AWS managed policies are listed:

Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy
AWSCodeCommitFullAccess	AWS managed	Permissions policy
AWSCodeCommitPowerUser	AWS managed	Permissions policy
AWSCodeCommitReadOnly	AWS managed	Permissions policy
IAMUserChangePassword	AWS managed	Permissions policy

The 'Create user' button at the bottom right is highlighted with an orange box.

User has been successfully Created

The screenshot shows the 'Users' page in the AWS IAM console. A green banner at the top states 'User created successfully'. Below it, the 'Users (1) Info' section describes an IAM user. The user list table shows one entry:

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Access keys
Himanshu	/	0	C	-	-	-	-

The 'Create user' button at the top right of the user list is highlighted with an orange box.

click on the Security Credentials option

The screenshot shows the AWS IAM user details page for 'Himanshu'. The 'Security credentials' tab is highlighted with a red box. The page displays basic user information like ARN, console access status, and creation date. Below this, a table lists attached permissions policies. At the bottom, there are sections for 'Console sign-in' and 'Multi-factor authentication (MFA)'. A navigation sidebar on the left provides links to various IAM management options.

Scroll down

This screenshot shows the same AWS IAM user details page as the previous one, but the content has been scrolled down. The 'Security credentials' tab is still selected. The 'Console sign-in' section is now visible, showing the console sign-in link and password status. The 'Multi-factor authentication (MFA)' and 'Access keys' sections are also visible below. The rest of the interface remains the same, including the sidebar and navigation bar.

Here is an (HTTPS Git Credentials for AWS CodeCommit) click on the Generate Credentials

The screenshot shows the AWS IAM console with the 'Identity and Access Management (IAM)' sidebar open. In the main content area, there are three sections: 'SSH public keys for AWS CodeCommit', 'HTTPS Git credentials for AWS CodeCommit', and 'Credentials for Amazon Keypaces (for Apache Cassandra)'. The 'HTTPS Git credentials for AWS CodeCommit' section is active, and its 'Actions' dropdown menu has a 'Generate credentials' button highlighted with an orange border.

Here username and Password are generated, download credentials

The screenshot shows the 'Generate credentials' modal dialog from the previous step. The 'Generate credentials' button has been clicked, and a green message box appears stating 'Your new credentials are available.' Below this, a section titled 'Save your user name and password or download the credentials file.' contains a note about the password being viewable only once. It also provides instructions for using the credentials with static tools. A red box highlights the 'Password' field, which shows a masked password '\*\*\*\*\*' followed by a 'Show' link. At the bottom of the dialog are 'Download credentials' and 'Close' buttons.

## Generate a credentials and status showing is Active

The screenshot shows the AWS IAM console with the URL [https://us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users/details/Himanshu?section=security\\_credentials](https://us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/users/details/Himanshu?section=security_credentials). The left sidebar shows the IAM navigation menu. The main content area displays two credential sections: 'HTTPS Git credentials for AWS CodeCommit' and 'Credentials for Amazon Keyspaces (for Apache Cassandra)'. The first section shows one credential named 'Himanshu-at-766811045096' which is active. The second section shows no credentials.

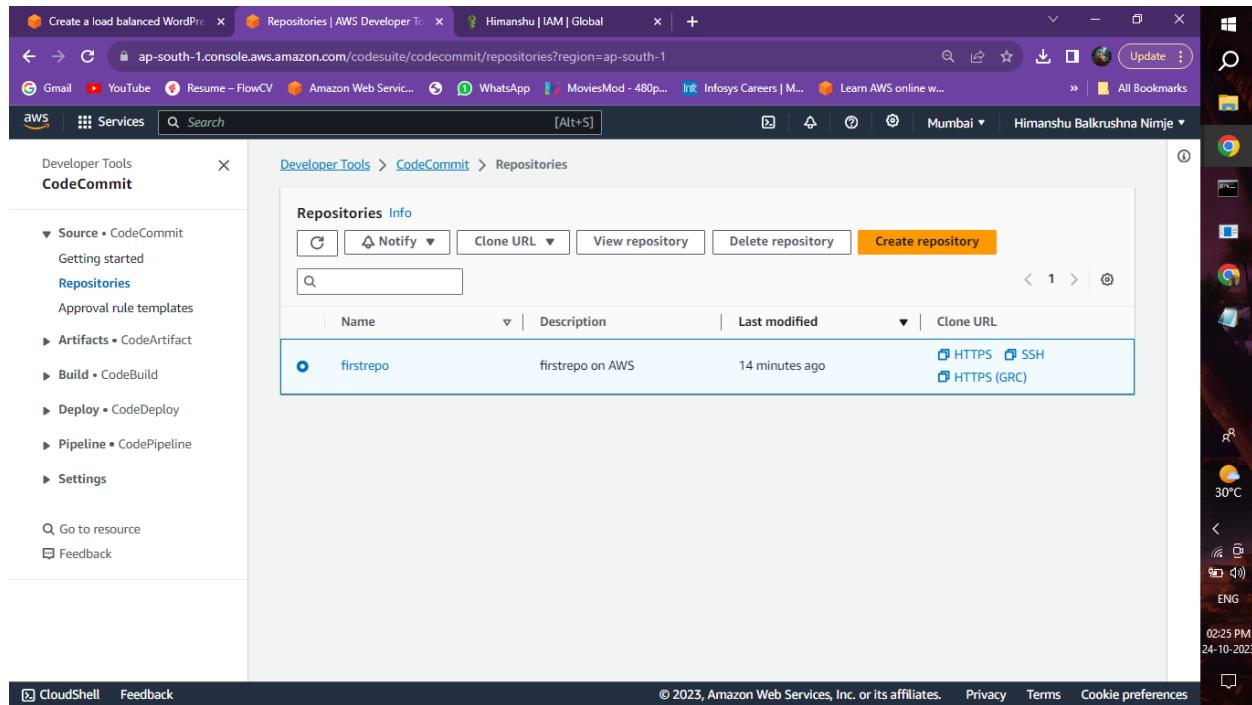
User name	Created	Status
Himanshu-at-766811045096	Now	Active

## Go to CodeCommit Dashboard

The screenshot shows the AWS CodeCommit dashboard with the URL <https://ap-south-1.console.aws.amazon.com/codesuite/codecommit/repositories?region=ap-south-1>. The left sidebar shows the CodeCommit navigation menu. The main content area displays the 'Repositories' list. A single repository named 'firstrepo' is listed, showing its details and clone URLs (HTTPS, SSH, and HTTPS (GRC)).

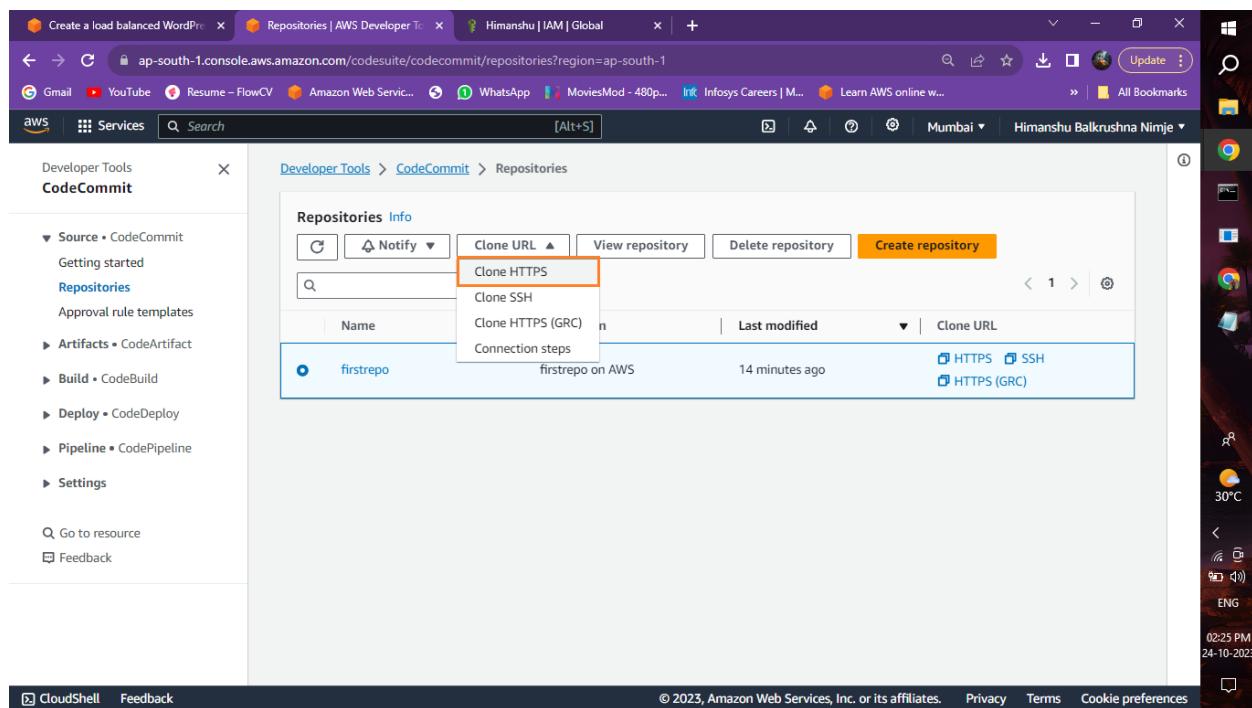
Name	Description	Last modified	Clone URL
firstrepo	firstrepo on AWS	14 minutes ago	<a href="#">HTTPS</a> <a href="#">SSH</a> <a href="#">HTTPS (GRC)</a>

Go to Repositories option→click on the firstrepo



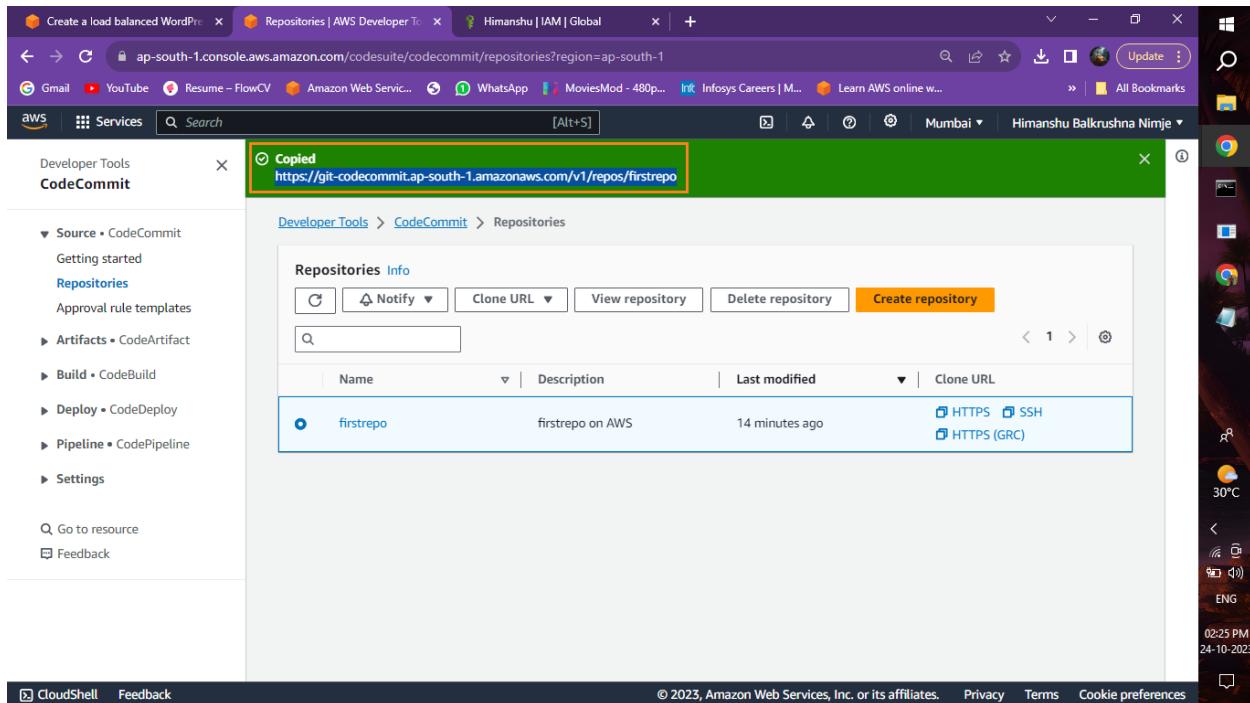
The screenshot shows the AWS Developer Tools interface for CodeCommit. On the left, a sidebar menu includes options like Source (CodeCommit), Getting started, Repositories, Approval rule templates, Artifacts (CodeArtifact), Build (CodeBuild), Deploy (CodeDeploy), Pipeline (CodePipeline), and Settings. The main content area displays a table of repositories. The first repository listed is 'firstrepo', which is highlighted with a blue border. The table columns include Name, Description, Last modified, and Clone URL (with options for HTTPS, SSH, and HTTPS (GRC)). At the top of the main area, there are buttons for Create repository, Notify, Clone URL, View repository, and Delete repository.

click clone URL→Clone HTTPS option

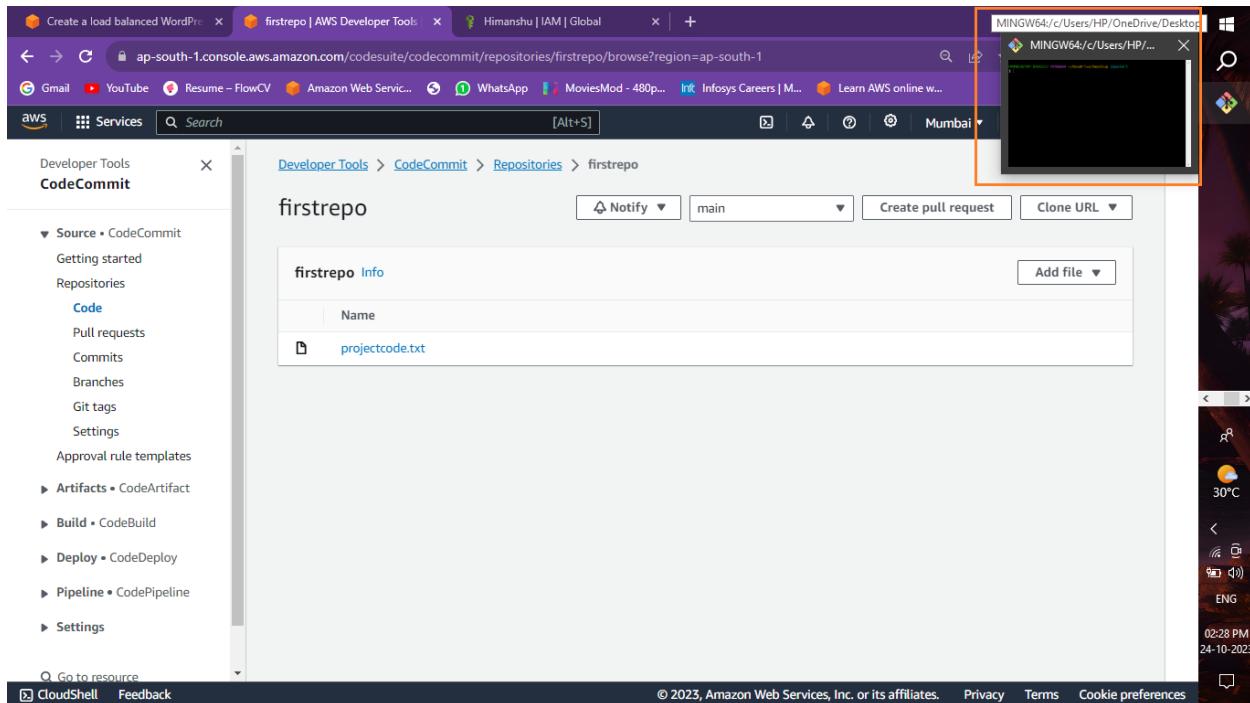


This screenshot is similar to the previous one but focuses on the 'Clone URL' dropdown menu. The 'Clone HTTPS' option is highlighted with a red box. The rest of the interface is identical to the first screenshot, showing the 'firstrepo' repository in the list and the same navigation and configuration options.

## Copy the URL



## Open Git-bash terminal on Desktop



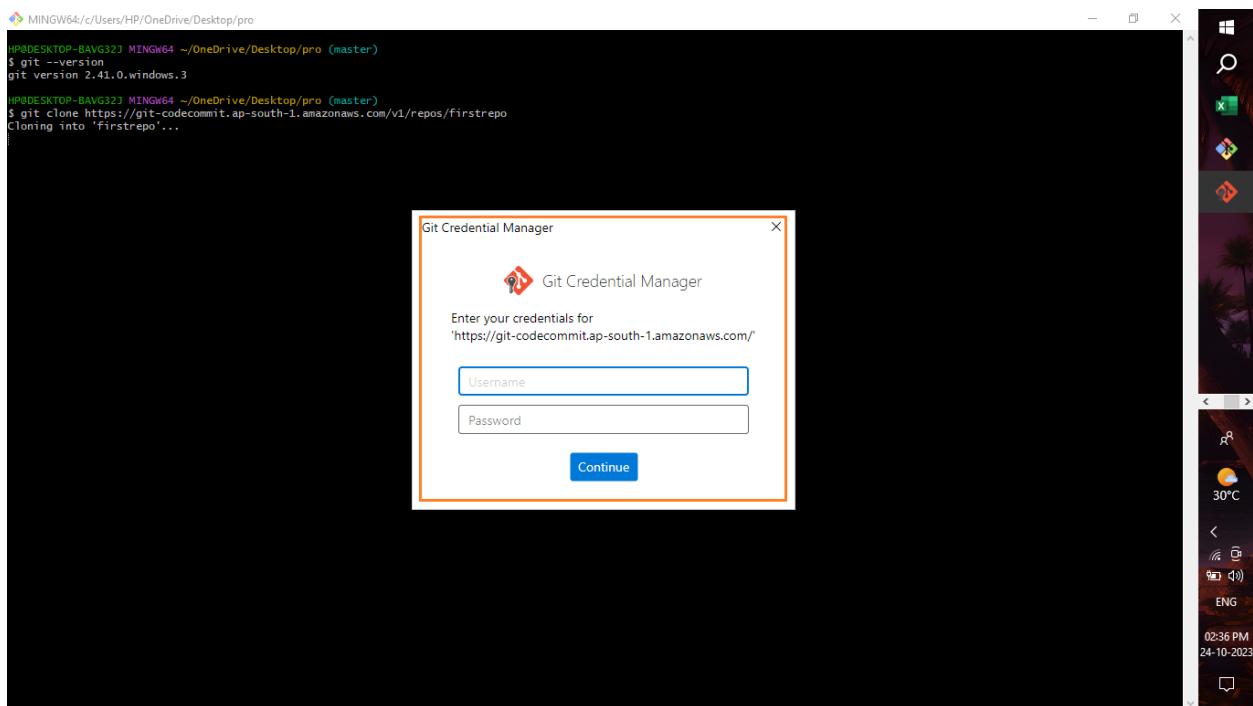
Now paste the URL in terminal



A screenshot of a Windows desktop environment. On the left, a terminal window titled 'MINGW64' shows the command \$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo being entered. On the right, a taskbar displays various pinned icons and the system tray showing the date and time as 02:35 PM 24-10-2023.

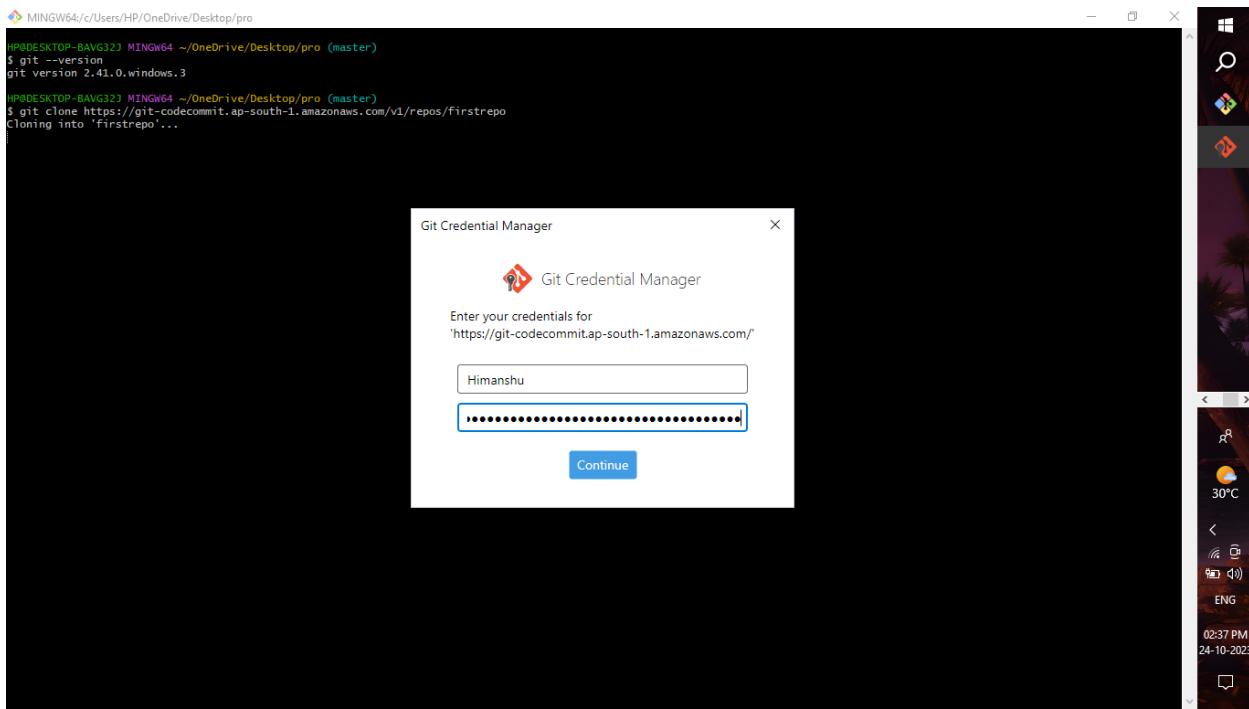
```
MINGW64/c/Users/HP/OneDrive/Desktop/pro
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git --version
git version 2.41.0.windows.3
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo
```

Enter Username and Password

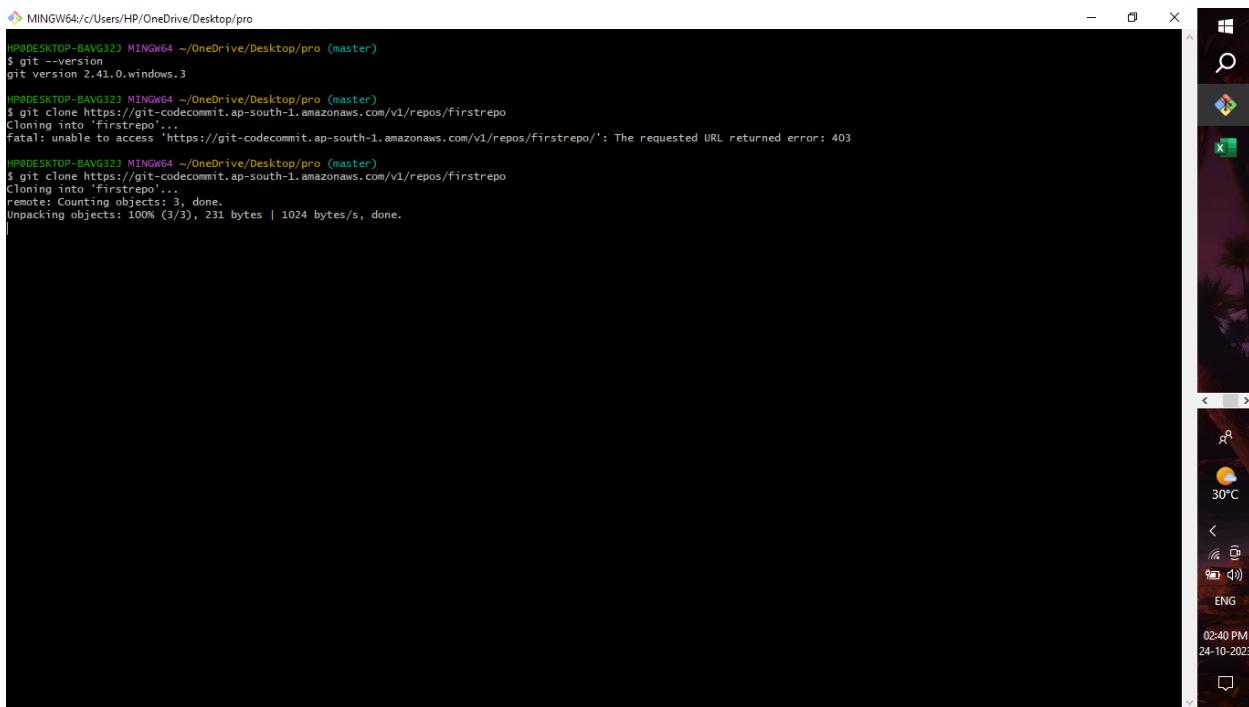


A screenshot of a Windows desktop environment. On the left, a terminal window titled 'MINGW64' shows the command \$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo being entered. A 'Git Credential Manager' dialog box is overlaid on the terminal, prompting for a 'Username' and 'Password'. On the right, a taskbar displays various pinned icons and the system tray showing the date and time as 02:36 PM 24-10-2023.

```
MINGW64/c/Users/HP/OneDrive/Desktop/pro
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git --version
git version 2.41.0.windows.3
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo
Cloning into 'firstrepo'...
```



Now the AWS Code-Commit Service cloned into Git-bash terminal



Go to /OneDrive/Desktop/pro (master)/firstrepo/

```

MINGW64:/c/Users/HP/OneDrive/Desktop/pro/firstrepo
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git --version
git version 2.41.0.windows.3

HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo
Cloning into 'firstrepo'...
fatal: unable to access 'https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo/': The requested URL returned error: 403

HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo
Cloning into 'firstrepo'...
remote: Counting objects: 3, done.
Unpacking objects: 100% (3/3), 231 bytes | 1024 bytes/s, done.

HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ cd firstrepo/
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/firstrepo (main)
$ ls -lrt

```

```

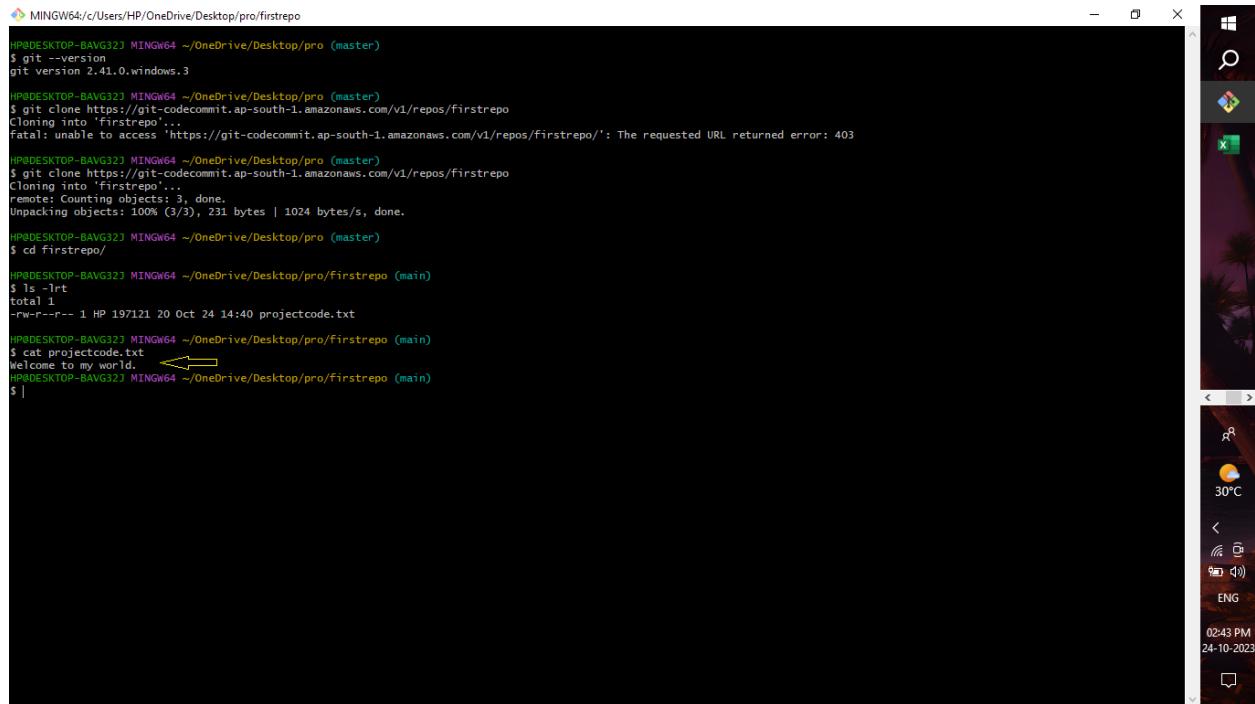
MINGW64:/c/Users/HP/OneDrive/Desktop/pro/firstrepo
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git --version
git version 2.41.0.windows.3

HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo
Cloning into 'firstrepo'...
fatal: unable to access 'https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo/': The requested URL returned error: 403

HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo
Cloning into 'firstrepo'...
remote: Counting objects: 3, done.
Unpacking objects: 100% (3/3), 231 bytes | 1024 bytes/s, done.

HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ cd firstrepo/
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/firstrepo (main)
$ ls
total 1
-rw-r--r-- 1 HP 197121 20 Oct 24 14:40 projectcode.txt
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/firstrepo (main)
$ 
```

## Cat projectcode.txt



A screenshot of a Windows desktop environment. On the left, a terminal window titled 'MINGW64' shows the command 'cat projectcode.txt' being run. The output of the command is 'Welcome to my world.' On the right, a taskbar displays various icons, including a weather widget showing '30°C' and a date/time indicator '02:43 PM 24-10-2023'.

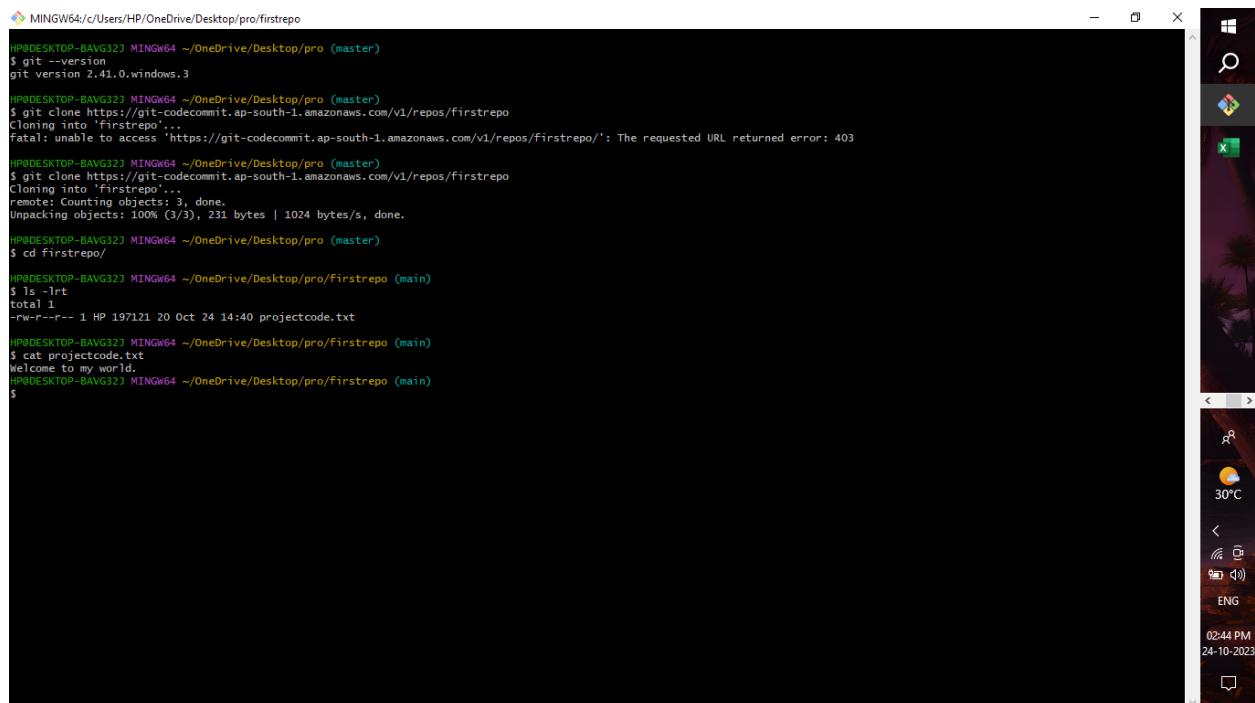
```
MINGW64:/c/Users/HP/OneDrive/Desktop/pro/firstrepo
$ git --version
git version 2.41.0.windows.3

MINGW64 ~/OneDrive/Desktop/pro (master)
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo
Cloning into 'firstrepo'...
fatal: unable to access 'https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo/': The requested URL returned error: 403

MINGW64 ~/OneDrive/Desktop/pro (master)
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo
Cloning into 'firstrepo'...
remote: Counting objects: 3, done.
Unpacking objects: 100% (3/3), 231 bytes | 1024 bytes/s, done.

MINGW64 ~/OneDrive/Desktop/pro (master)
$ cd firstrepo/
$ ls -lrt
total 1
-rw-r--r-- 1 HP 197121 20 Oct 24 14:40 projectcode.txt

MINGW64 ~/OneDrive/Desktop/pro/firstrepo (main)
$ cat projectcode.txt
Welcome to my world.
MINGW64 ~/OneDrive/Desktop/pro/firstrepo (main)
$
```



A screenshot of a Windows desktop environment, similar to the one above. On the left, a terminal window titled 'MINGW64' shows the command 'cat projectcode.txt' being run. The output of the command is 'Welcome to my world.' On the right, a taskbar displays various icons, including a weather widget showing '30°C' and a date/time indicator '02:44 PM 24-10-2023'.

```
MINGW64:/c/Users/HP/OneDrive/Desktop/pro/firstrepo
$ git --version
git version 2.41.0.windows.3

MINGW64 ~/OneDrive/Desktop/pro (master)
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo
Cloning into 'firstrepo'...
fatal: unable to access 'https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo/': The requested URL returned error: 403

MINGW64 ~/OneDrive/Desktop/pro (master)
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo
Cloning into 'firstrepo'...
remote: Counting objects: 3, done.
Unpacking objects: 100% (3/3), 231 bytes | 1024 bytes/s, done.

MINGW64 ~/OneDrive/Desktop/pro (master)
$ cd firstrepo/
$ ls -lrt
total 1
-rw-r--r-- 1 HP 197121 20 Oct 24 14:40 projectcode.txt

MINGW64 ~/OneDrive/Desktop/pro/firstrepo (main)
$ cat projectcode.txt
Welcome to my world.
MINGW64 ~/OneDrive/Desktop/pro/firstrepo (main)
$
```

Create New file that is going to pushed into CodeCommit

`nano himanshu.txt`

MINGW64/c/Users/HP/OneDrive/Desktop/pro/firstrepo

```

HP0DESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git --version
git version 2.41.0.windows.3

HP0DESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo
Cloning into 'firstrepo'...
fatal: unable to access 'https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/firstrepo/': The requested URL returned error: 403

HP0DESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ cd firstrepo/
Cloning into 'firstrepo'...
remote: Counting objects: 3, done.
Unpacking objects: 100% (3/3), 231 bytes | 1024 bytes/s, done.
HP0DESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/firstrepo (main)
$ ls -lrt
total 1
-rw-r--r-- 1 HP 197121 20 Oct 24 14:40 projectcode.txt

HP0DESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/firstrepo (main)
$ cat projectcode.txt
Welcome to my world.
HP0DESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/firstrepo (main)
$ touch himanshu.txt

HP0DESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/firstrepo (main)
$ nano himanshu.txt

```

Enter Content and save the file

MINGW64/c/Users/HP/OneDrive/Desktop/pro/firstrepo

GNU nano 7.2

himanshu.txt

Modified

welcome

Save modified buffer?

Y Yes      N No      A Cancel

```

MINGW64:/c/Users/HP/OneDrive/Desktop/pro/firstrepo
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git --version
git version 2.41.0.windows.3
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/Firstrepo
Cloning into 'Firstrepo'...
fatal: unable to access 'https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/Firstrepo/': The requested URL returned error: 403
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/Firstrepo
Cloning into 'Firstrepo'...
remote: Enumerating objects: 3, done.
Unpacking objects: 100% (3/3), 231 bytes | 1024 bytes/s, done.
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro (master)
$ cd Firstrepo/
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/Firstrepo (main)
$ ls -lrt
total 1
-rw-r--r-- 1 HP 197121 20 Oct 24 14:40 projectcode.txt
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/Firstrepo (main)
$ cat projectcode.txt
Welcome to my world.
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/Firstrepo (main)
$ touch himanshu.txt
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/Firstrepo (main)
$ cat himanshu.txt
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/Firstrepo (main)
$ cat himanshu.txt
welcome
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/Firstrepo (main)
$ |

```

`git commit -m himanshu.txt`

=> Commit changes

`git push origin main`

=> To push the changes in Remote repository

`git log`

=> To see the all the commits in history

```

MINGW64:/c/Users/HP/OneDrive/Desktop/pro/firstrepo
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/Firstrepo (main)
$ git commit -m himanshu.txt
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
nothing to commit, working tree clean
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/Firstrepo (main)
$ ls
himanshu.txt projectcode.txt
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/Firstrepo (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 296 bytes | 98.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/Firstrepo
  0711d81..46a847 main > main
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/Firstrepo (main)
$ git checkout
Your branch is up to date with 'origin/main'.
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/Firstrepo (main)
$ git log
commit 46a847d6ea2a582f5318133e31b622de11dae8 (HEAD -> main, origin/main, origin/HEAD)
Author: Himanshu Nimb 
Date: Tue Oct 24 14:56:46 2023 +0530
  Adding new file
commit 0711d818370fb9162ead14e4017e9053b971d5d
Author: Himanshu Nimb <nimbhimanshu76@gmail.com>
Date: Tue Oct 24 08:40:45 2023 +0000
  first file in repos
HPDESKTOP-BAVG323 MINGW64 ~/OneDrive/Desktop/pro/Firstrepo (main)
$ |

```

Now go to code commit dashboard and go to repositories option and click on the repository name (firstrepo) files are showing himanshu.txt

The screenshot shows the AWS CodeCommit interface. On the left, a sidebar menu for 'CodeCommit' is visible with options like 'Source', 'Artifacts', 'Build', 'Deploy', 'Pipeline', and 'Settings'. The main content area displays the 'Repositories' section under 'CodeCommit'. A table lists a single repository named 'firstrepo'. The table columns include 'Name', 'Description', 'Last modified', and 'Clone URL'. Below the table, there are links for 'HTTPS', 'SSH', and 'HTTPS (GRC)'. At the bottom of the page, there are links for 'CloudShell' and 'Feedback'.

This screenshot is similar to the previous one but shows a different view within the same repository. It displays the 'firstrepo' repository details. In the 'Info' section, there is a list of files: 'himanshu.txt' and 'projectcode.txt'. A message below the files states 'Now here is our file Push Successfully'. The sidebar and overall layout are identical to the first screenshot.

## What I Learned

I learn several key concepts and gain practical experience with AWS CodeCommit and Git.

1. I understand how to create a new CodeCommit repository using either the AWS Management Console or the AWS CLI.
2. I learned how to clone an existing Git repository locally using the `git clone` command.
3. I gain experience in pushing the content of your local repository to AWS CodeCommit using the `git push` command.
4. I configure AWS CLI with the necessary IAM credentials, ensuring secure communication with AWS CodeCommit.
5. I learned how to verify the migration by checking the CodeCommit repository's existence and details using the AWS CLI.
6. I learned that authentication with CodeCommit can be done using either HTTPS or SSH, and you've used the HTTPS method in the demo.
7. Depending on My setup, you might need to configure IAM roles and permissions for the IAM user interacting with CodeCommit.

By performing this demo, I get hands-on experience with the process of migrating a Git repository to AWS CodeCommit. This experience is valuable for developers and administrators who are working with AWS services and Git repositories, especially in scenarios where secure and managed source code hosting is essential. It also serves as a practical introduction to working with CodeCommit in a real-world scenario.