Working with AWS CodeCommit

**SPL-33 Version 2.0.10**

© 2023 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Overview**

AWS CodeCommit is a highly scalable, managed source control service that hosts private Git repositories. CodeCommit stores your data in Amazon S3 and Amazon DynamoDB giving your repositories high scalability, availability, and durability. You simply create a repository to store your code. There is no hardware to provision and scale or software to install, configure, and operate.

This hands-on lab gives you practice with AWS CodeCommit, part of AWS Developer Tools. In this lab, you first create a code repository in AWS CodeCommit. Then you create a local repository on a Linux instance running in EC2. After you create the local repo, you make some changes to it. Then you synchronize (commit) your changes to the AWS CodeCommit repository.

TOPICS COVERED

This lab demonstrates how to:

* Create a code repository using AWS CodeCommit via the Amazon Management Console
* Create a local code repository on the Linux instance using git
* Synchronize a local repository with an AWS CodeCommit repository

PREREQUISITES

Students should have some development experience and understand the principles of source code repositories, and has some prior development experience. Students should be comfortable with making SSH connections to instances running in Amazon EC2, and using Linux commands and editors from the command line in Linux. Students should have taken at a minimum *Introduction to Amazon Elastic Compute Cloud (EC2)* prior to taking this lab.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* A command that you must run
* A sample output that you can use to verify the output of a command or edited file
* A hint, tip, or important guidance

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

 You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

**Task 1: Create an AWS CodeCommit repository**

In this task, you use the AWS Management Console to create an AWS CodeCommit repository.

1. At the top of the AWS Management Console, in the search bar, search for and choose

CodeCommit

.

1. On the **AWS CodeCommit** page, choose **Create repository**.
2. On the **Create repository** page:

* For **Repository name**, enter

My-Repo

.

* For **Description** enter

My first repository

.

1. Choose **Create**.

An empty repository named *My-Repo* is created.

You should now be on the **My-Repo** page, which contains the details to connect to the repository.

 Congratulations! You have successfully created a new CodeCommit repository.

**Task 2: Connect to the Amazon EC2 instance**

An Amazon EC2 instance has been created for you as part of the lab environment build process. In this task, you connect to the instance using AWS Systems Manager Session Manager.

1. Copy the **Ec2InstanceSessionUrl** value from the list to the left of these instructions, and then paste it into a new web browser tab.

A console connection is made to the instance inside your web browser window. A set of commands are run automatically when you connect to the instance that change to the user’s home directory and display the path of the working directory, similar to this:

cd HOME; pwd; bash

sh-4.2$ cd HOME; bash; pwd

/home/ec2-user

[ec2-user@ip-10-0-1-137 ~]$

 Congratulations! You have successfully connected to a terminal session on the lab EC2 instance.

**Task 3: Create a local repository using Git**

This task provides an example of how you would use AWS CodeCommit to synchronize to any local code repository that you might create in your normal production development environment.

1. In the terminal session, run the following command to install the Git client:

sudo yum install -y git

 On a Windows-based computer, you might need to use **Ctrl + Shift + V** or open the context menu (right-click) to paste text into a Session Manager console window.

1. Run the following commands to configure the Git credential helper with the AWS credential profile, and allow the Git credential helper to send the path to repositories:

git config --global credential.helper '!aws codecommit credential-helper $@'

git config --global credential.UseHttpPath true

 These commands have no output.

Next, obtain the HTTPS URL of your AWS CodeCommit repository.

1. Return to your web browser tab with the AWS CodeCommit console, which should be on the **My-Repo** page.
2. At the upper-right of the page, choose **Clone URL**, and then choose **Clone HTTPS**.

The repository URL is copied to your clipboard and should look similar to this: *https://git-codecommit.us-east-1.amazonaws.com/v1/repos/My-Repo*.

1. Return to your web browser tab with the terminal session.
2. Run the following command to clone the **My-Repo** repository to the instance:

* Replace the **CLONE\_HTTPS\_URL** placeholder value with the **Clone HTTPS** URL that you copied previously.

git clone CLONE\_HTTPS\_URL

 The output should indicate that you are cloning the *My-Repo* repository, and that the repository is empty, similar to this:

Cloning into 'My-Repo'...

warning: You appear to have cloned an empty repository.

 Congratulations! You have successfully connected to and synchronized with the AWS CodeCommit repository.

Next, you conclude with a short demonstration of making a change and synchronizing the repositories. This is a mini example of the workflow of synchronizing code changes during the development process.

**Task 4: Making a code change and first commit to the repo**

In this task, you create your first commit in your local repo. You create two example files in your local repo, use Git to stage the changes to your local repo, and then commit the changes.

1. Run the following command to change to the **My-Repo** directory:

cd ~/My-Repo

1. Run the following command to create two files in your local repo:

echo "The domestic cat (Felis catus or Felis silvestris catus) is a small, usually furry, domesticated, and carnivorous mammal." >cat.txt

echo "The domestic dog (Canis lupus familiaris) is a canid that is known as man's best friend." >dog.txt

 These commands have no output.

1. Run the following command to list the files in the current directory:

ls

 The output should show the two files you created, similar to this:

cat.txt dog.txt

1. Run the following command to stage the changes in your local repo:

git add cat.txt dog.txt

 This command has no output.

1. Run the following command to view the status of your repo:

git status

 The output should show the branch you are current working in (master) and that the two files are ready to be committed to the repository, similar to this:

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: cat.txt

new file: dog.txt

1. Run the following command to commit the changes in your local repo:

git commit -m "Added cat.txt and dog.txt"

 The output displays a message stating that the name and email address of the committer were configured automatically. In a production environment, you would use the commands listed to set your name and email address, which are then applied to each commit you do. The output also shows that two files were changed and inserted, similar to this:

Committer: EC2 Default User <ec2-user@ip-10-1-12-142.ec2.internal>

Your name and email address were configured automatically based

on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly:

git config --global user.name "Your Name"

git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

2 files changed, 2 insertions(+)

create mode 100644 cat.txt

create mode 100644 dog.txt

1. Run the following command to view details about the commit you just made:

git log

 The output shows that there is one commit to the master branch, the name of the author, the date the commit was made, and the files that were added, similar to this:

commit 772d16037b2e0d7ee7e97aa9218e571346bebe0e (HEAD -> master)

Author: EC2 Default User <ec2-user@ip-10-1-12-142.ec2.internal>

Date: Wed Jul 20 19:20:06 2022 +0000

Added cat.txt and dog.txt

Now that you have an initial commit in your local repo, you can push the commit from your local repo to your AWS CodeCommit repository.

 Congratulations! You have successfully added files to a local repository.

**Task 5: Push your first commit**

In this task, you push the commit from your local repo to your AWS CodeCommit repository.

1. Run the following command to push your commit through the default remote name Git uses for your AWS CodeCommit repository (origin), from the default branch in your local repo (master):

git push -u origin master

 The output displays the details of the process to create the branch and push the files to the remote repository, similar to this:

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 2 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 447 bytes | 447.00 KiB/s, done.

Total 4 (delta 0), reused 0 (delta 0), pack-reused 0

To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/My-Repo

\* [new branch] master -> master

Branch 'master' set up to track remote branch 'master' from 'origin'.

After you have pushed code to your AWS CodeCommit repository, you can view the contents using the AWS CodeCommit console.

1. Return to your web browser tab with the AWS CodeCommit console, which should be on the **My-Repo** page.
2. Choose your web browser’s refresh button to refresh the page.

The two files that you added to your repository should be displayed.

1. Choose the link for each file to view its contents.

 Congratulations! You have successfully pushed the changes from your local repository to the remote CodeCommit repository.

**Conclusion**

 Congratulations! You now have successfully:

* Created a code repository using the AWS CodeCommit Management Console
* Created a local code repository on your Linux instance using git
* Synchronized a local repository with an AWS CodeCommit repository

Consider how to bring AWS CodeCommit’s features and capabilities to your development workflow, including:

* **Collaboration:** AWS CodeCommit is designed for collaborative software development. CodeCommit allows you to commit, diff, and merge your code allowing you to easily maintain control of your team’s projects. You can create a repository from the AWS Management Console, AWS CLI, or AWS SDKs and start working with the repository using Git.
* **Encryption:** You can transfer your files to and from AWS CodeCommit via HTTPS and SSH. Your repositories are also automatically encrypted at rest through AWS Key Management Service using customer-specific keys.
* **Access Control:** AWS CodeCommit uses AWS Identity and Access Management to control and monitor who can access your data as well as how, when, and where they can access it.
* **High Availability and Durability:** AWS CodeCommit stores your repositories in Amazon S3 and Amazon DynamoDB. Your data is redundantly stored across multiple facilities. This architecture increases the availability and durability of your repository data.
* **Unlimited Repositories:** AWS CodeCommit allows you to create as many repositories as you need, with no size limits. You can store and version any kind of file, including application assets such as images and libraries alongside your code.
* **Easy Access and Integration:** You can use the AWS Management Console, AWS CLI, and AWS SDKs to manage your repositories. You can also use Git commands or Git graphical tools to interact with your repository source files. AWS CodeCommit supports all Git commands and works with your existing Git tools. You can integrate with your development environment plugins or continuous integration/continuous delivery systems.

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

ADDITIONAL RESOURCES

* [AWS CodeCommit](https://aws.amazon.com/codecommit/developer-resources/)
* [AWS Training and Certification](http://aws.amazon.com/training/)

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).