**Getting started with AWS CodeCommit**

In this lab you would be working on key CodeCommit features. First, you create a repository and commit some changes to it. Then, you browse the files and view the changes. You can also create a pull request so other users can review and comment on changes to your code.

**Prerequisites**

**Setup for HTTPS users using Git credentials**

The simplest way to set up connections to AWS CodeCommit repositories is to configure Git credentials for CodeCommit in the IAM console, and then use those credentials for HTTPS connections. You can also use these same credentials with any third-party tool or individual development environment (IDE) that supports HTTPS authentication using a static user name and password.

**Step 1: Initial configuration for CodeCommit**

* Create an IAM user, or use an existing one, in your AWS account. Make sure you have an access key ID and a secret access key associated with that IAM user. For more information, see Creating an IAM User in Your AWS Account.
* Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
* In the IAM console, in the navigation pane, choose Users, and then choose the IAM user you want to configure for CodeCommit access.
* On the Permissions tab, choose Add Permissions.
* In Grant permissions, choose Attach existing policies directly.
* From the list of policies, select AWSCodeCommitPowerUser or another managed policy for CodeCommit access. For more information, see AWS managed (predefined) policies for CodeCommit.
* After you have selected the policy you want to attach, choose Next: Review to review the list of policies to attach to the IAM user. If the list is correct, choose Add permissions.

**Step 2: Install Git**

To work with files, commits, and other information in CodeCommit repositories, you must install Git on your local machine. CodeCommit supports Git versions 1.7.9 and later. We recommend using a recent version of Git.

**Step 3: Create Git credentials for HTTPS connections to CodeCommit**

After you have installed Git, create Git credentials for your IAM user in IAM

**To set up HTTPS Git credentials for CodeCommit**

* Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
* Make sure to sign in as the IAM user who will create and use the Git credentials for connections to CodeCommit.
* In the IAM console, in the navigation pane, choose Users, and from the list of users, choose your IAM user.
* On the user details page, choose the Security Credentials tab, and in HTTPS Git credentials for AWS CodeCommit, choose Generate.


                Generating Git credentials in the IAM console
            

* Copy the user name and password that IAM generated for you, either by showing, copying, and then pasting this information into a secure file on your local computer, or by choosing **Download credentials** to download this information as a .CSV file. You need this information to connect to CodeCommit.


                Downloading Git credentials from the IAM console
            

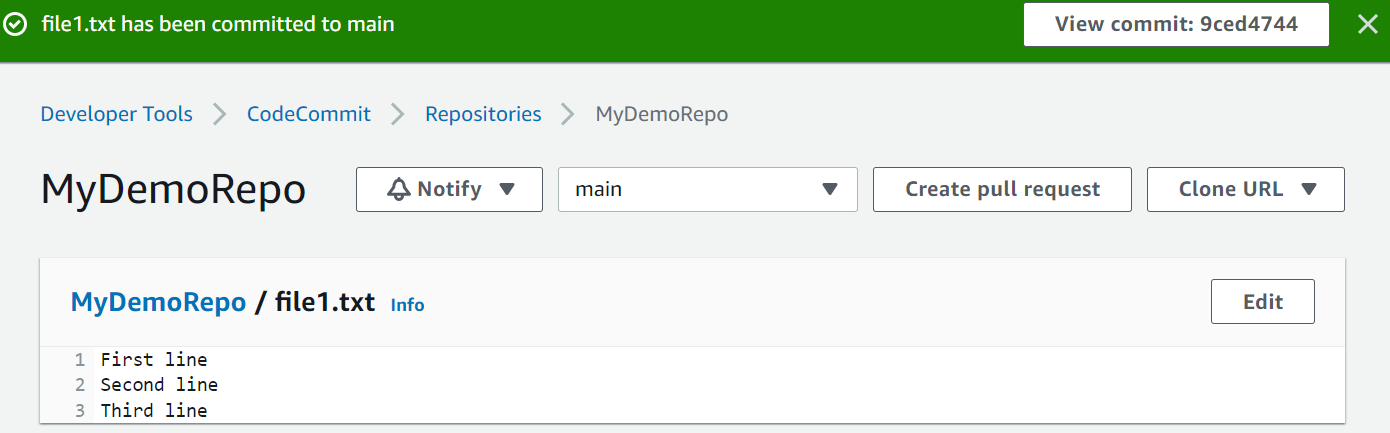
**Step 4: Create a CodeCommit repository**

* Open the CodeCommit console at https://console.aws.amazon.com/codesuite/codecommit/home.
* Use the region selector to choose the AWS Region where you want to create the repository. For more information, see Regions and Git connection endpoints.
* On the Repositories page, choose Create repository.
* On the Create repository page, in Repository name, enter a name for your repository (for example, MyDemoRepo).
* (Optional) In Description, enter a description (for example, My demonstration repository). This can help you and other users identify the purpose of the repository.
* (Optional) Choose Add tag to add one or more repository tags (a custom attribute label that helps you organize and manage your AWS resources) to your repository. For more information, see Tagging repositories in AWS CodeCommit.
* (Optional) Select Enable Amazon CodeGuru Reviewer for Java if this repository will contain Java code, and you want to have CodeGuru Reviewer analyze that Java code. CodeGuru Reviewer uses multiple machine learning models to find Java code defects and to automatically suggest improvements and fixes in pull requests. For more information, see the Amazon CodeGuru Reviewer User Guide.
* Choose Create.


                Creating a repository from the console
            

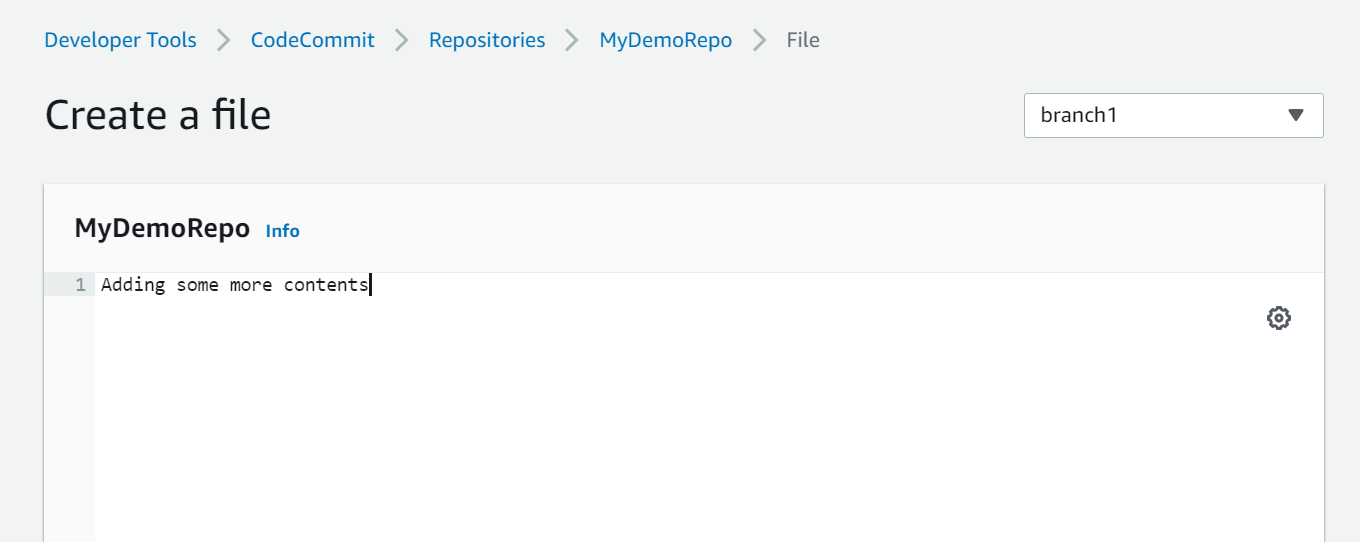
**Step 5: Add files to your repository**

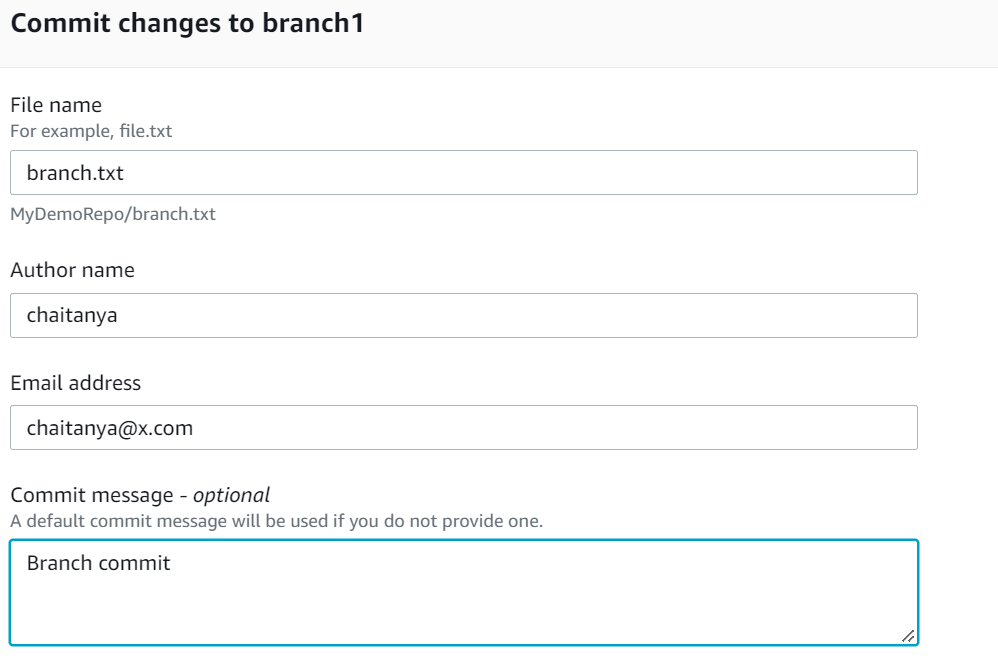
* In the navigation bar for the repository, choose Code.
* Choose Add file, and then choose whether to create a file or upload a file from your computer.
* In the drop-down list of branches, choose the branch where you want to add the file. The default branch is selected automatically , which is **main**
* In File name, enter a name for the file. In the code editor, enter the code for the file.
* In Author name, enter the name you want displayed to other repository users.
* In Email address, enter an email address.
* (Optional) In Commit message, enter a brief message. Although this is optional, we recommend that you add a commit message to help your team members understand why you added this file. If you do not enter a commit message, a default message is used.
* Choose Commit changes.

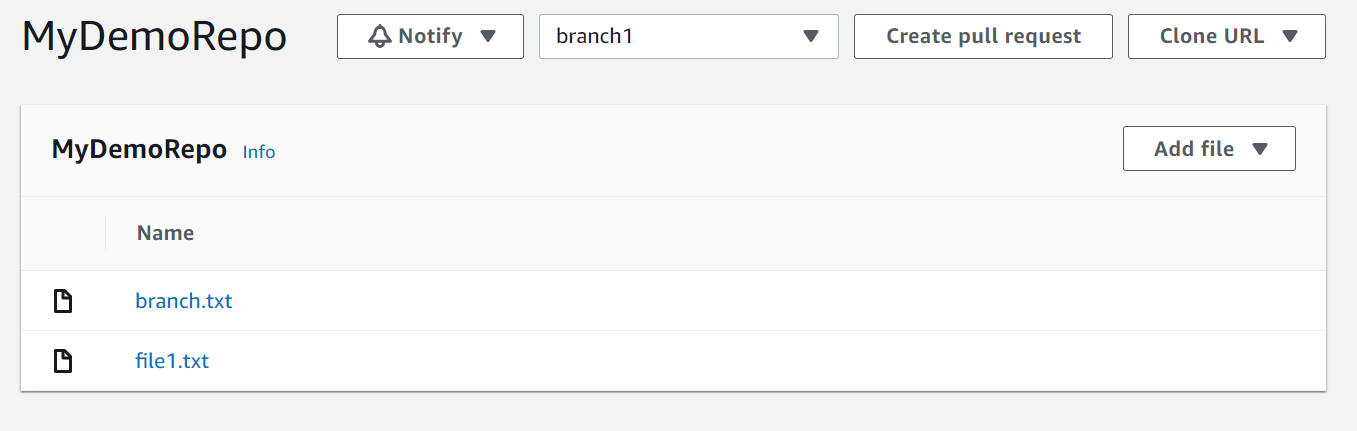


**Step 6: Creating branches:**

* Create a branch named branch 1
* Add a file in branch named as branch.txt
* Commit the file in the branch
* Check if the file has been created

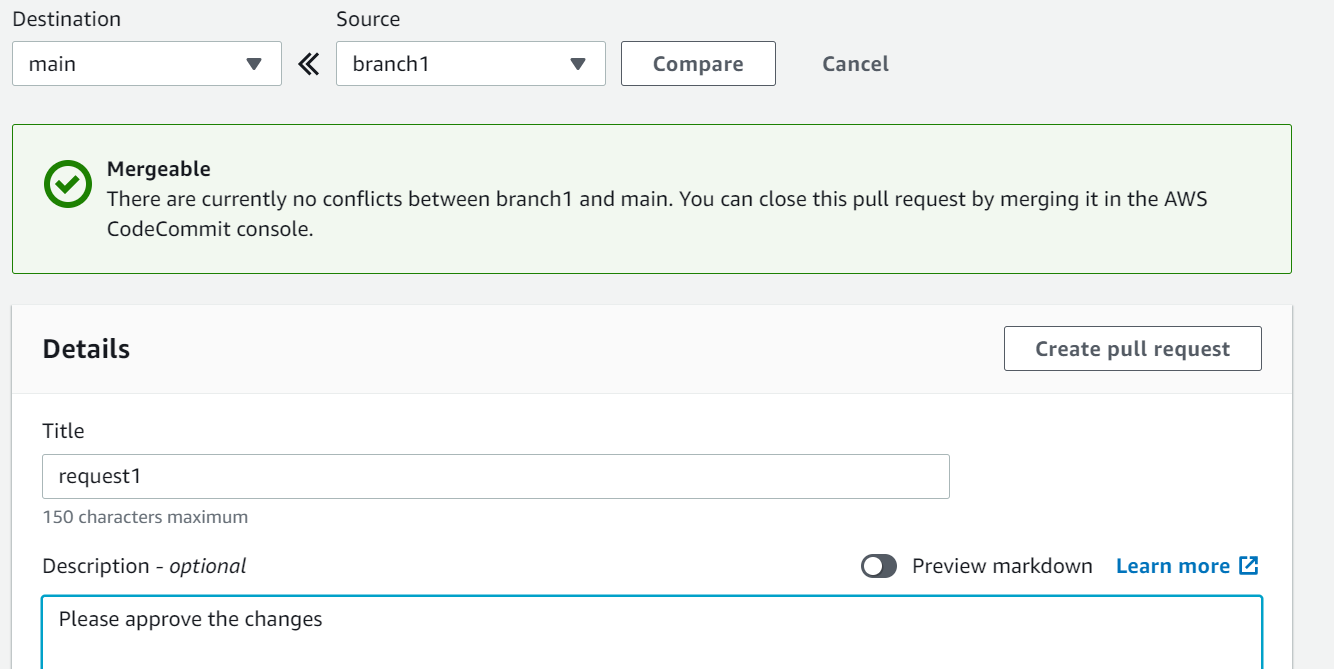






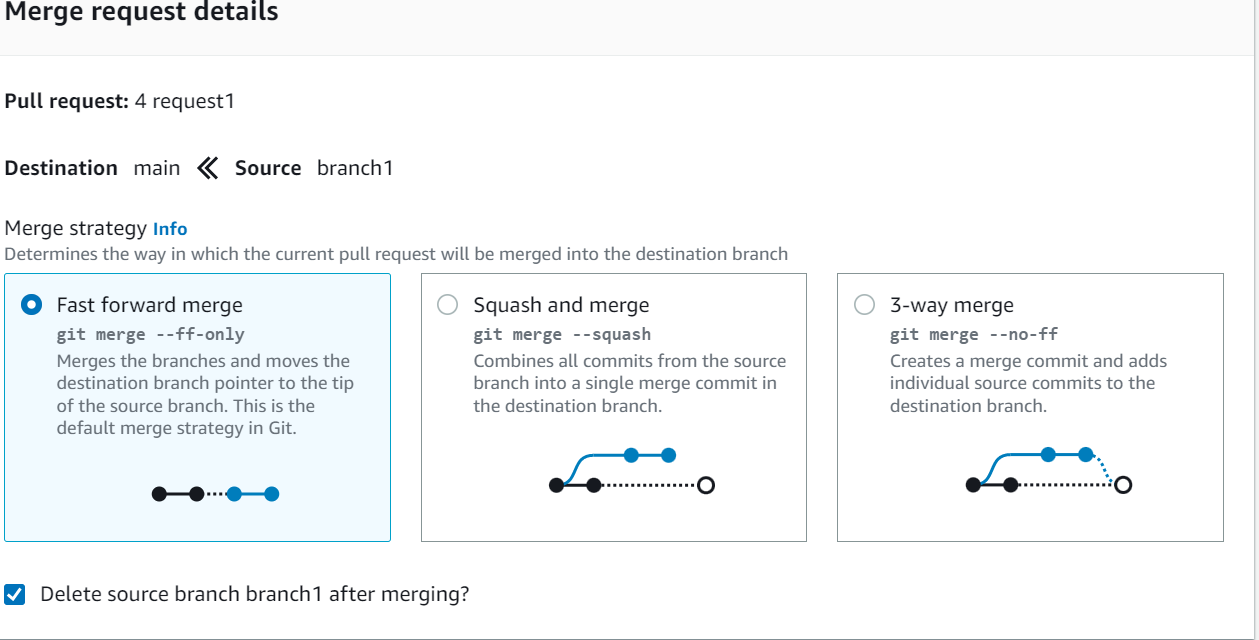
**Step 7: Create a Pull Request for merging branch1 contents to be compared and merged with main**

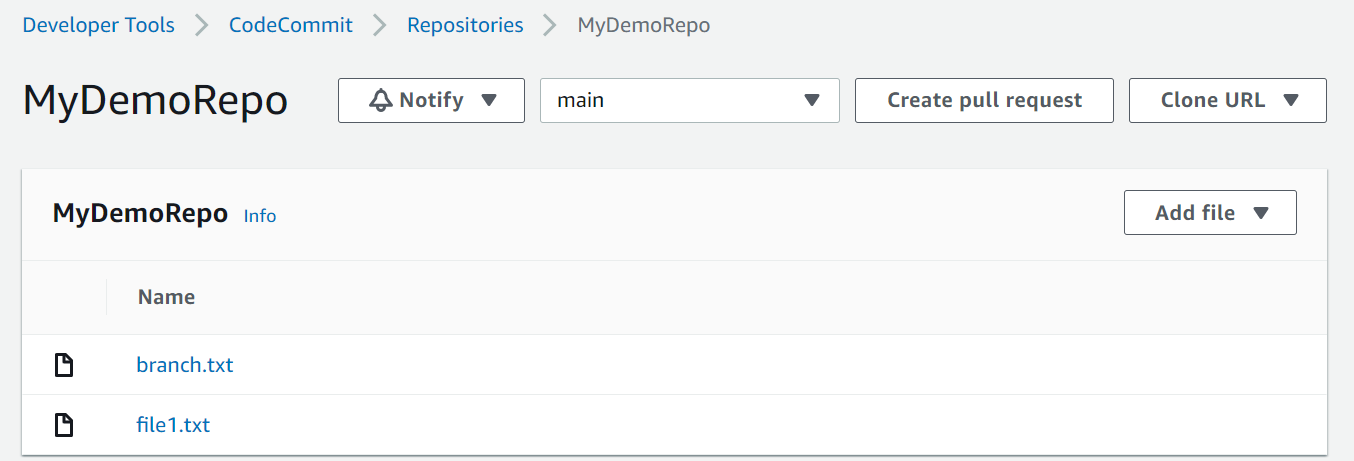
* Click on branch named branch1 and Click **on Create Pull request** button
* Compare the source and destination by clicking on Compare button
* Give a name for pull request and Click on Create Pull request button in the bottom



**Step 8: Merge the branches**

* Merge the branch files with the main branch by clicking the Merge button
* Choose the appropriate Merge Strategy
* Click on Merge Pull Request in the bottom
* Check if the files have been updated in the main branch





**Ref:**

**Connect to the CodeCommit console and clone the repository from local Git**

* Open the CodeCommit console at https://console.aws.amazon.com/codesuite/codecommit/home.
* In the region selector, choose the AWS Region where the repository was created. Repositories are specific to an AWS Region. For more information, see Regions and Git connection endpoints.
* Find the repository you want to connect to from the list and choose it. Choose Clone URL, and then choose the protocol you want to use when cloning or connecting to the repository. This copies the clone URL.
* Copy the HTTPS URL if you are using either Git credentials with your IAM user or the credential helper included with the AWS CLI.
* Open a terminal, command line, or Git shell. Run the git clone command with the HTTPS clone URL you copied to clone the repository. For example, to clone a repository named MyDemoRepo to a local repo named my-demo-repo in the US East (Ohio) Region

**git clone** [**https://git-codecommit.us-east-2.amazonaws.com/v1/repo**](https://git-codecommit.us-east-2.amazonaws.com/v1/repo)