**General Guideline for Maintain AWS CodeCommit Repository**

This tutorial will help you to learn most of the functions and capabilities in the AWS CodeCommit and this tutorial does not based on Git commands. Therefore, no prior Git knowledge need to follow this series.

In this tutotial, you can follow,

[Step 1 : Create a CodeCommit repository 2](#_Toc95860464)

[Step 2: Add files to your repository 3](#_Toc95860465)

[Step 3 : How to maintain the repository using AWS Sagemaker 4](#_Toc95860466)

[Step 4 : Browse the contents of your repository 5](#_Toc95860467)

[Step 5: Create and collaborate on a pull request (merge request) 7](#_Toc95860468)

[Step 6: Create an approval rule 8](#_Toc95860469)

[Step 7: Monitoring 10](#_Toc95860470)

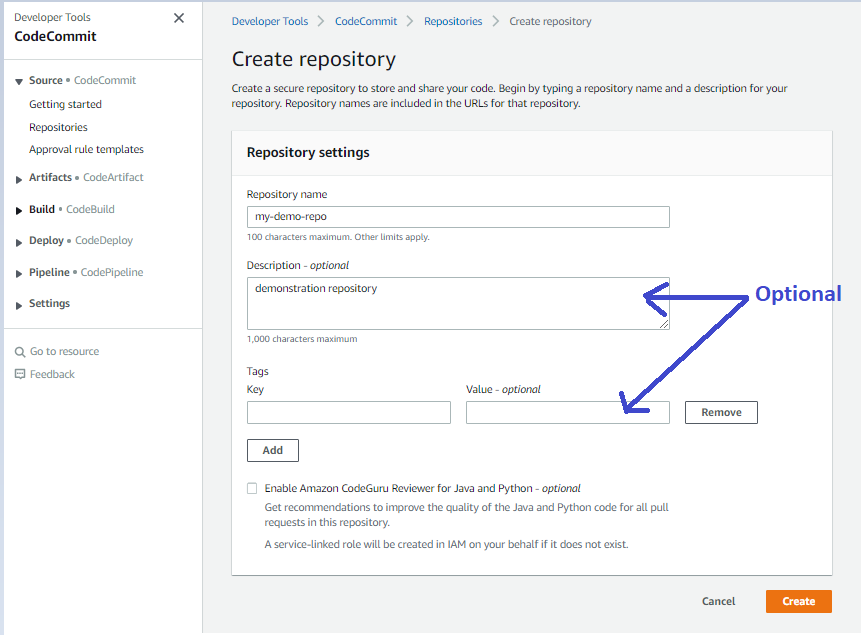
[Step 8: Manage Branches 10](#_Toc95860471)

[Step 9: Clean Up 11](#_Toc95860472)

# Step 1 : Create a CodeCommit repository

If you do not have a CodeCommit repository yet, please follow the bellow steps to create a repository.

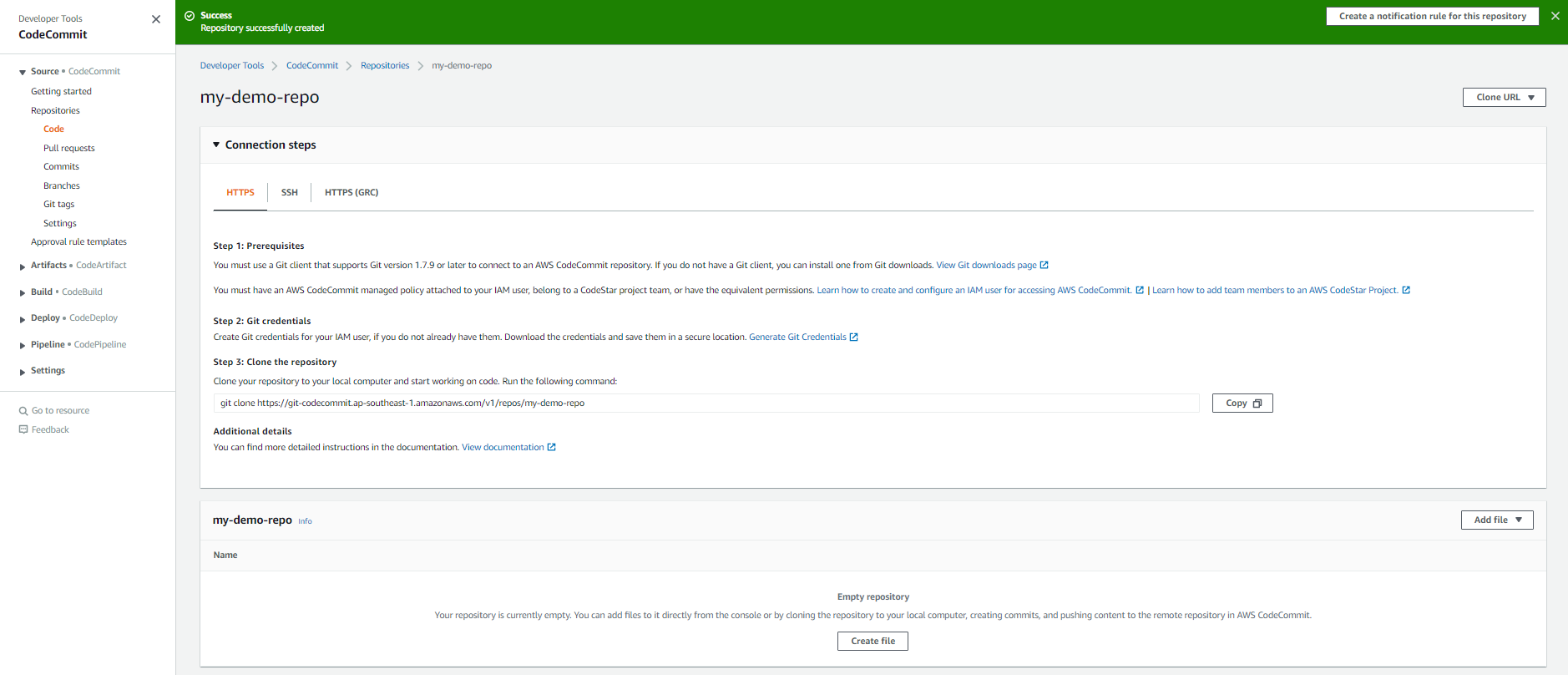
1. Open the CodeCommit console using <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. If you need to change your region use regional selector at the top right conner. Otherwise do not change or do anything and it will automatically select the correct region related to you.
3. Choose **Create repository** at the top right conner.
4. On the **Create repository** page, you have to fill bellow details.



1. **Repository name**: Enter a suitable name for your repository (for example my-demo-repo). We highly recommend you to follow a standard way for this naming convention.
2. **Description** (optional): Enter a short description about your project (for example demostration repository)
3. **Tags** (optional): Add one or more tags (this will help your organization or the compnay to manage your AWS services)
4. **Enable Amazon CodeGuru Reviewer for Java and Python** (optional): If you enable this option, CodeGuru will analyze your Java or Python code(if you have). It will automatically suggests imporvements.
5. Choose **create** option

# Step 2: Add files to your repository

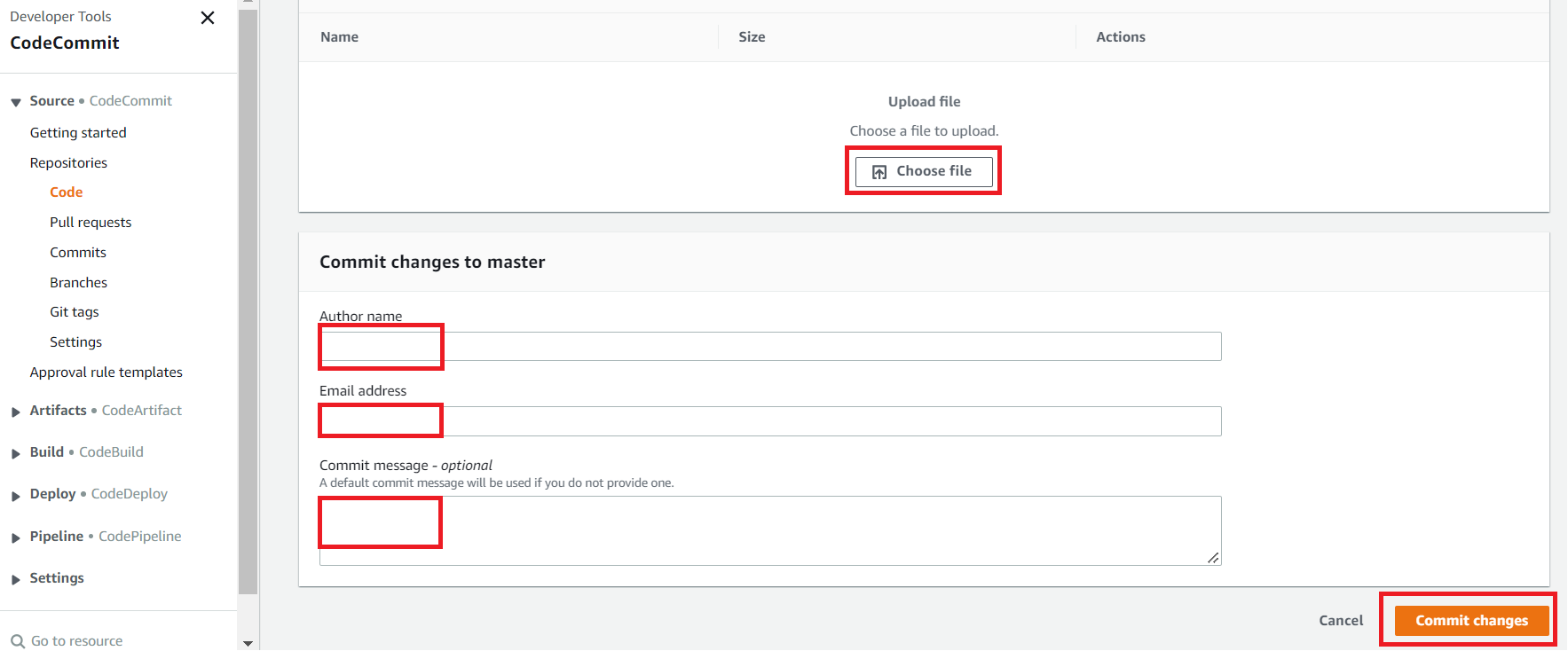
After doing the above step 1, you can now see the below interface.



* You can add your local file into this repository or you can create a new file by clicking a **Create file** option. If you create/upload the first file to the repo, a branch will automatically generated and it’s name called ‘**main’.**
* Using git client to clone the CodeCommit repository to your local computer, and then you can easily do commiting, pushing and merging your files.

Usually we go with the **upload file** option to upload a local computer file to the repository and you should follow the below steps.

1. Choose **Add file**, and then choose **upload a file** option to upload a file in your local computer.
2. After that upload file window, you have to fill the below details.
3. **Author name** : Enter the name you want displayed to other repository users.
4. **Email address**: Enter your/organization email address.
5. **Commit message** (optional) : Enter a brief message and we higly recommend that you add a commit message to help your teammates to understand why you added this file.
6. Select **Commit changes**.



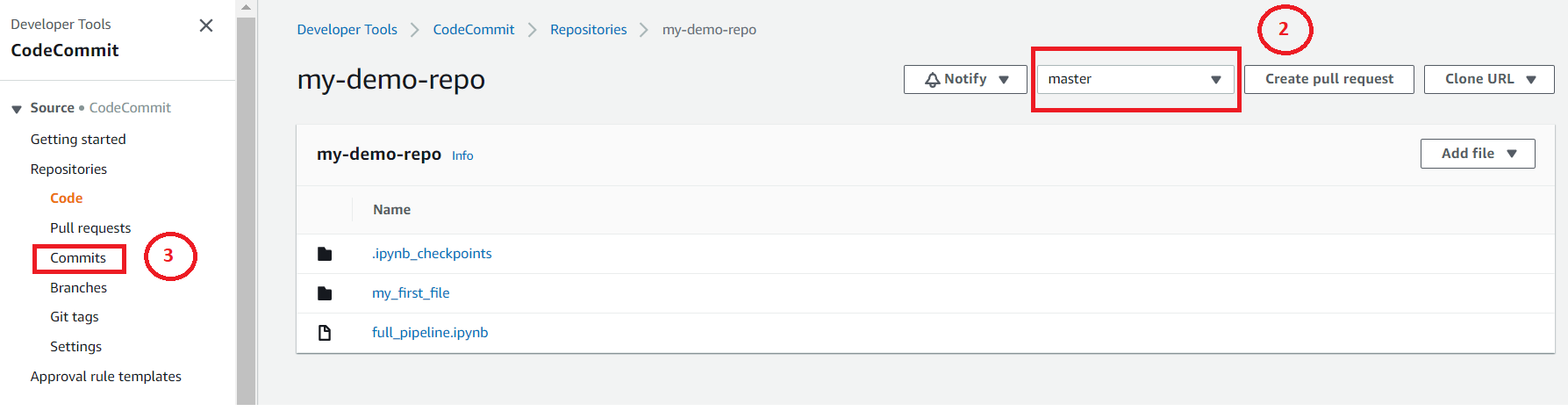
# Step 3 : How to maintain the repository using AWS Sagemaker

Please follow my Github repository

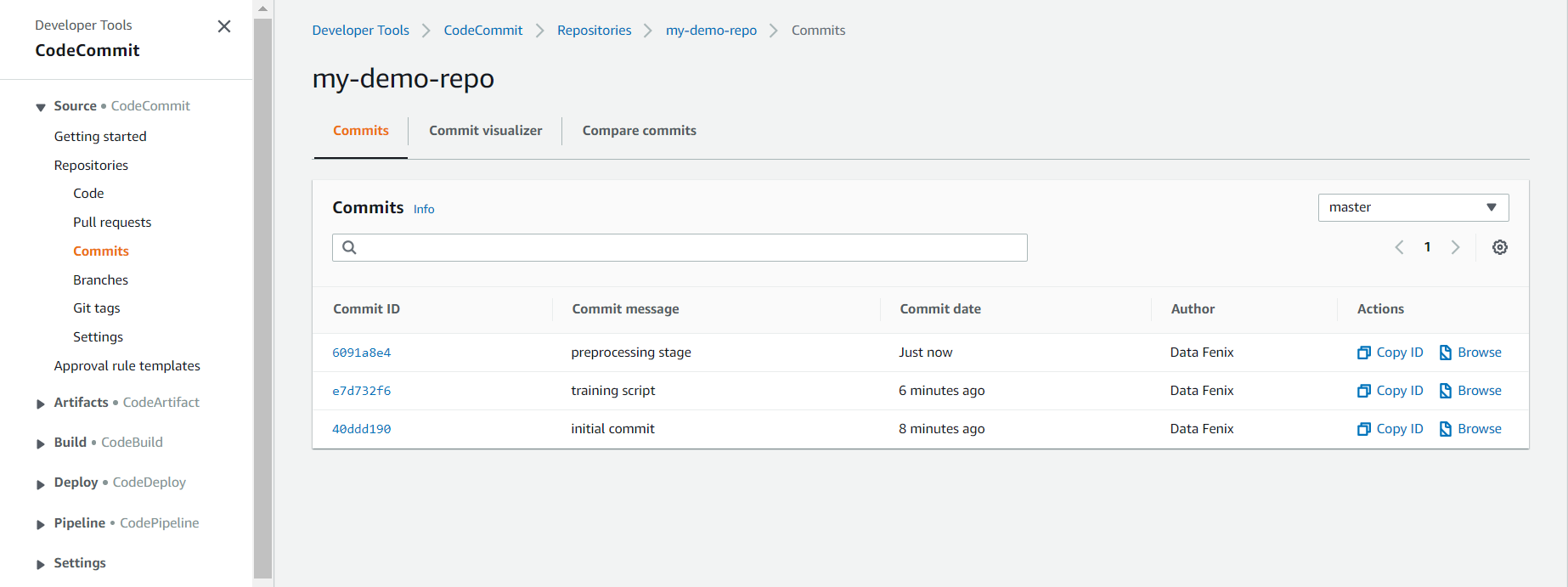
<https://github.com/Data-Fenix/maintain-codecommit-using-aws-sagemaker>

# Step 4 : Browse the contents of your repository

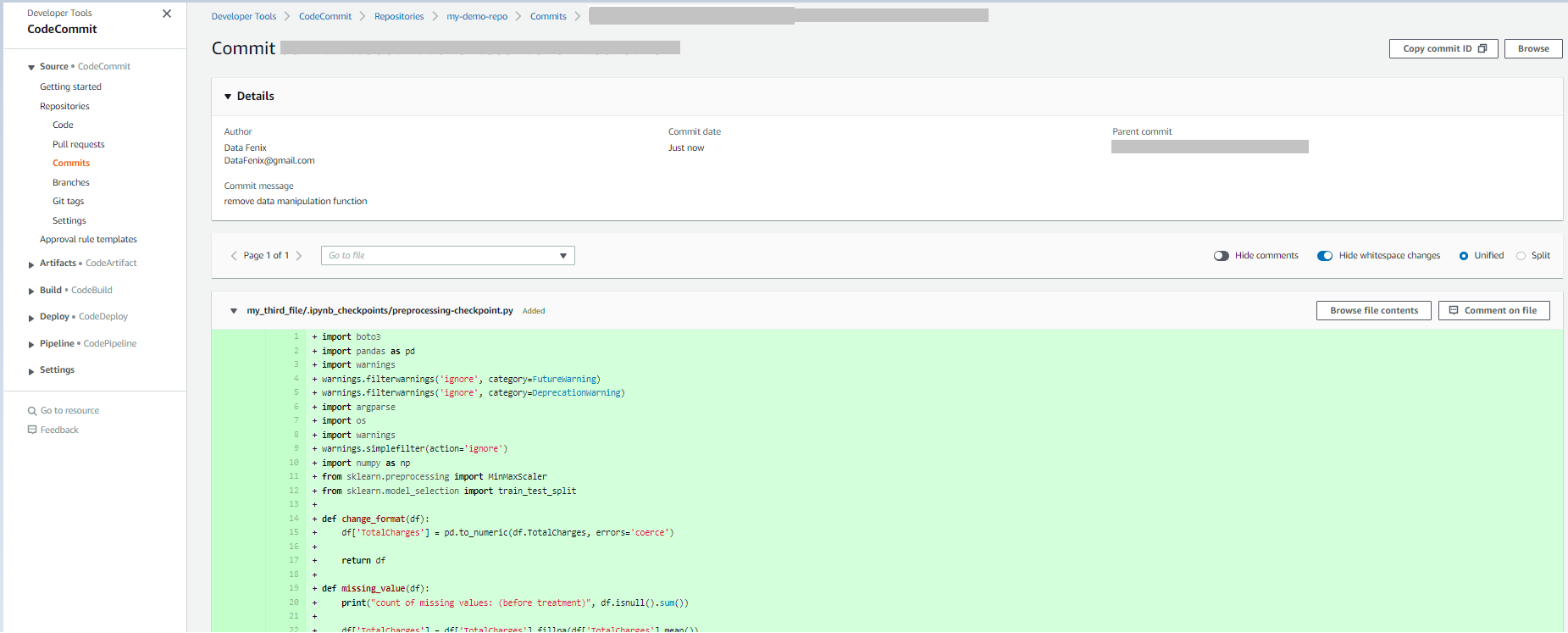
1. From repositories, click my-demo-repo.
2. The page shows the folders and files in the default branch of my-demo-repo. To change the branch, click the arrow down button in the **main** option (Next to **Notify** button)



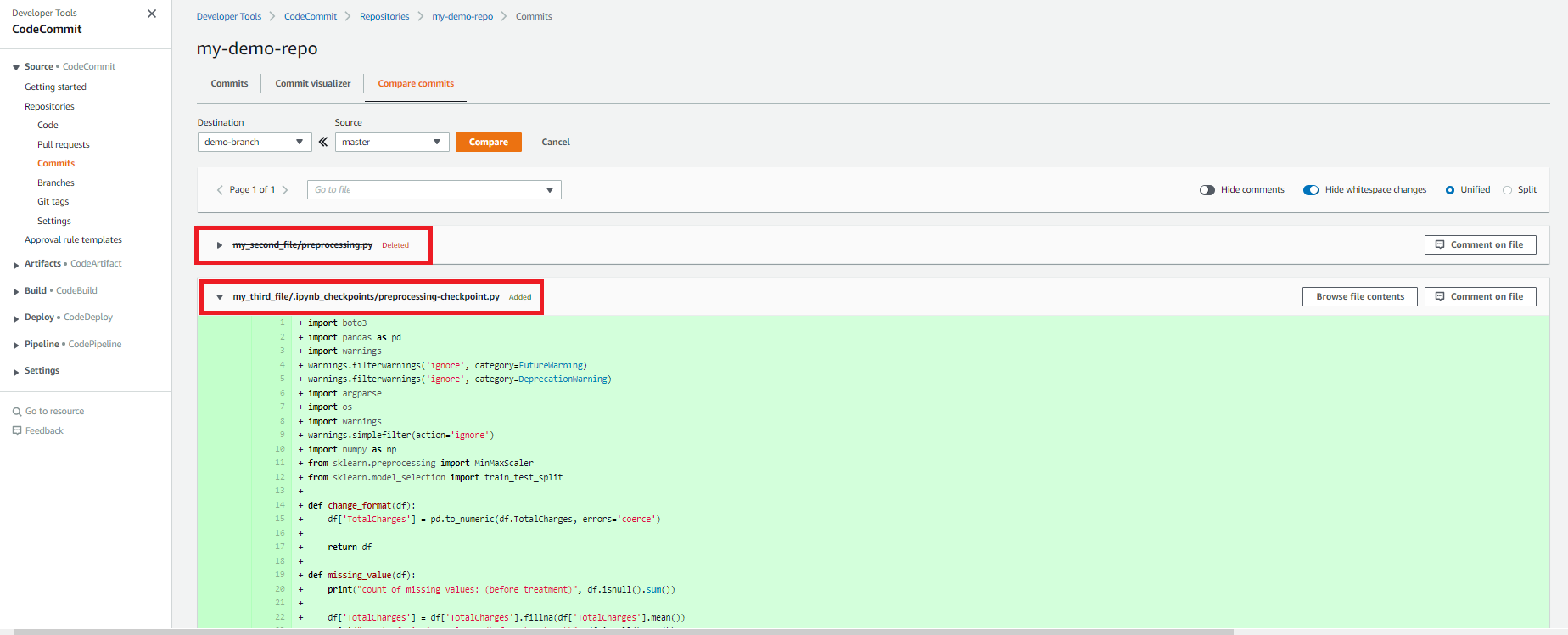
1. To browse the commit history of the repository, choose **Commits**(in the right hand side). The console displays the commit history for the default branch(main). If you need to change the branch once again click on the arrow down in the main button.



1. Let’s say you need to see more details about the commit, you have to click on the **Commit ID (relavent**). After that you can see the more details about that commit and specially differences.

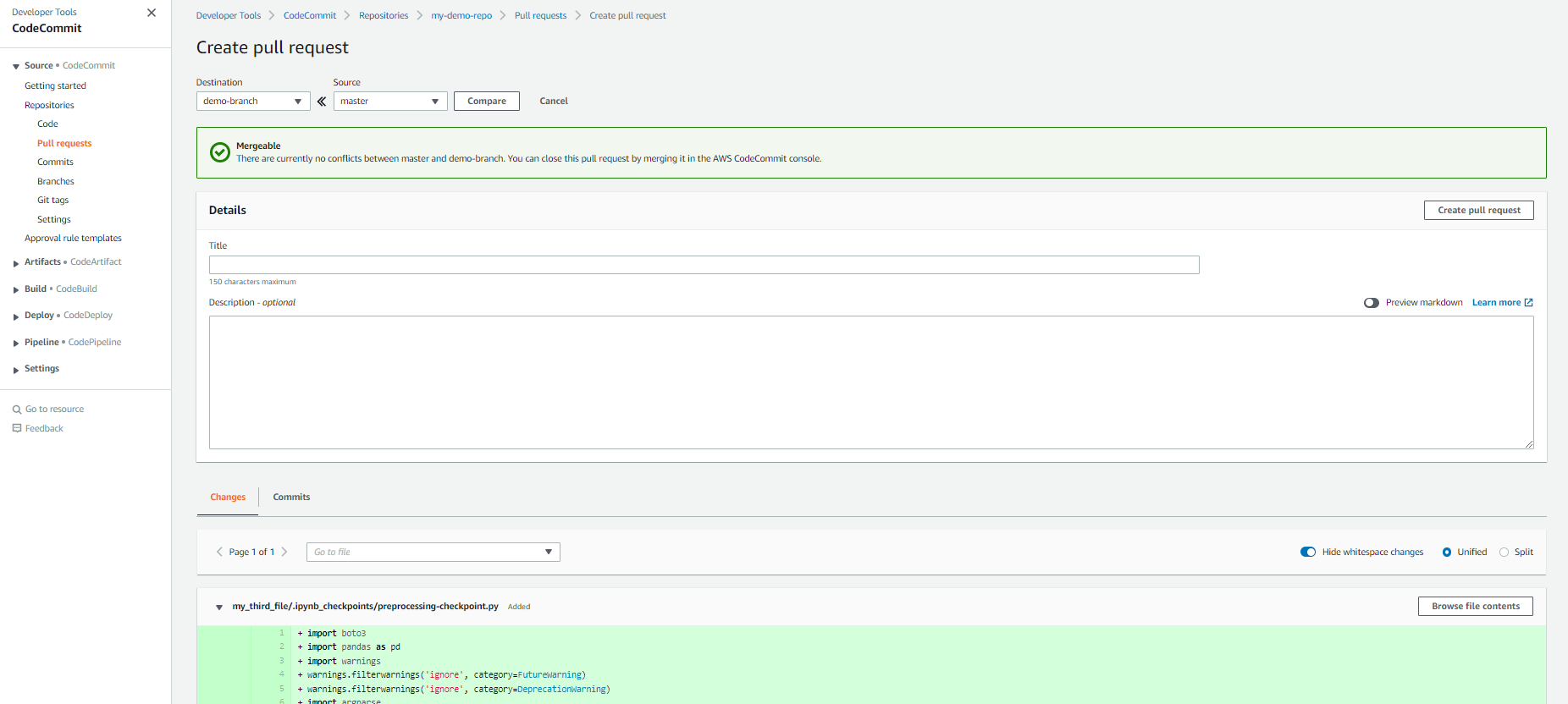


1. In **Commits**, choose the **Commit visualizer** tab.
2. In **Commits**, click Compare commits. In here you can compare two branches easily.( you can compare the difference between the previous version of the file with the latest version)



# Step 5: Create and collaborate on a pull request (merge request)

Let’s say you newly join to the current ongoing project and your supervisor said you do not have to start from the begging and continue this workload. So you are new to this project and you do not need to mess up this exiting improvement. Therefore you can copy all the files into new baranch and using that new branch you can upgrade this project. So now you have a backup file as well. This method called pull request.



Please follow the bellow steps to create a pull request.

1. In the navigation pane(Right hand side), choose **Pull requests.**
2. In **Pull request**, choose **Create pull request**.
3. Enter the Destination branch and the Source branch according to your requirement.

**Source**: Branch that contains the changes you want reviewed

**Destination:** Branch where you want the reviewed code to be merged when the pull request is closed.

1. Choose **Compare**. (Ideaily it has to pop up Mergeable message)
2. Review and compare the merge details and enter a suitable **Title** and **Description** (Optional) in the given space.
3. Press **Create Pull Request** button.
4. Your pull request appears in the list of pull requests for the repository and if there’s no approval rules, you can choose the **merge** option in the top right conner. If you need to add an approval rule, please follow the next section (Step 6)
5. After that it will appear the below window and choose a suitable merge strategy. If you want to delete the source branch after the merge, tick the bottom left square box.

More details about merge strategies:

<https://www.geeksforgeeks.org/merge-strategies-in-git/>

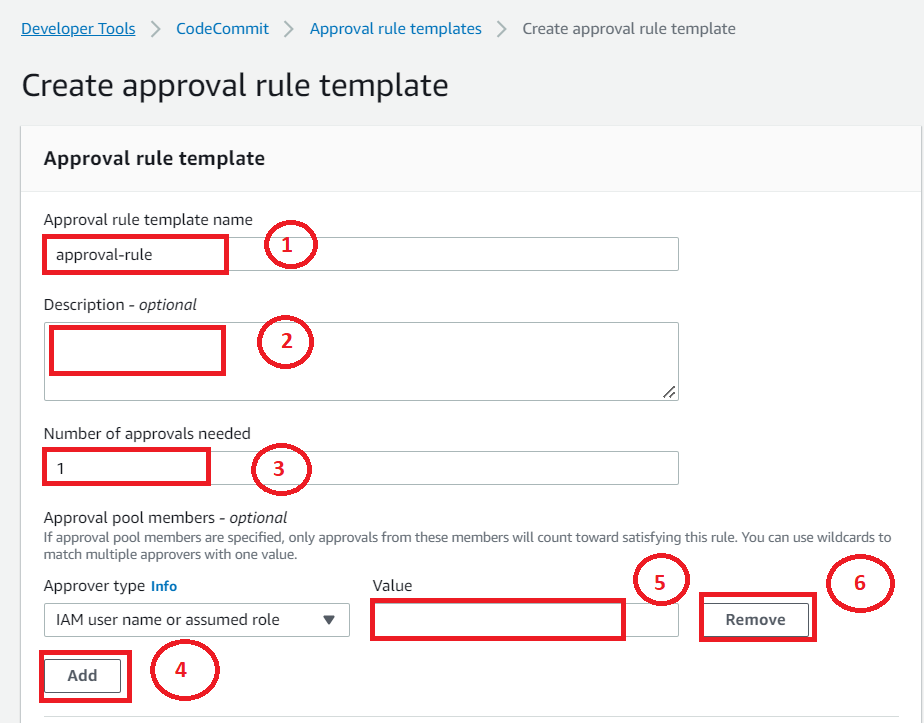
1. Finally press the Merge pull request button in the bottom right conner.

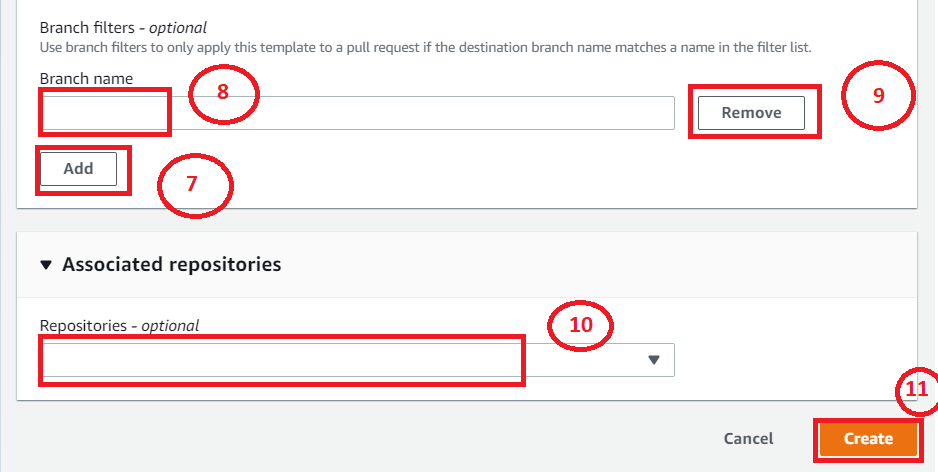
# Step 6: Create an approval rule

If you need to apply some approval stage before merging, you shuold create an approval rule for that. Let’s say you are a member of team and you need to merge two branches. But your team leader need some governance(he need some tracking system) or he need to see the changes before merge branches. So in this case, better option is to create an approval rule in CodeCommit.

If you need to create an approval rule, follow the bellow process.

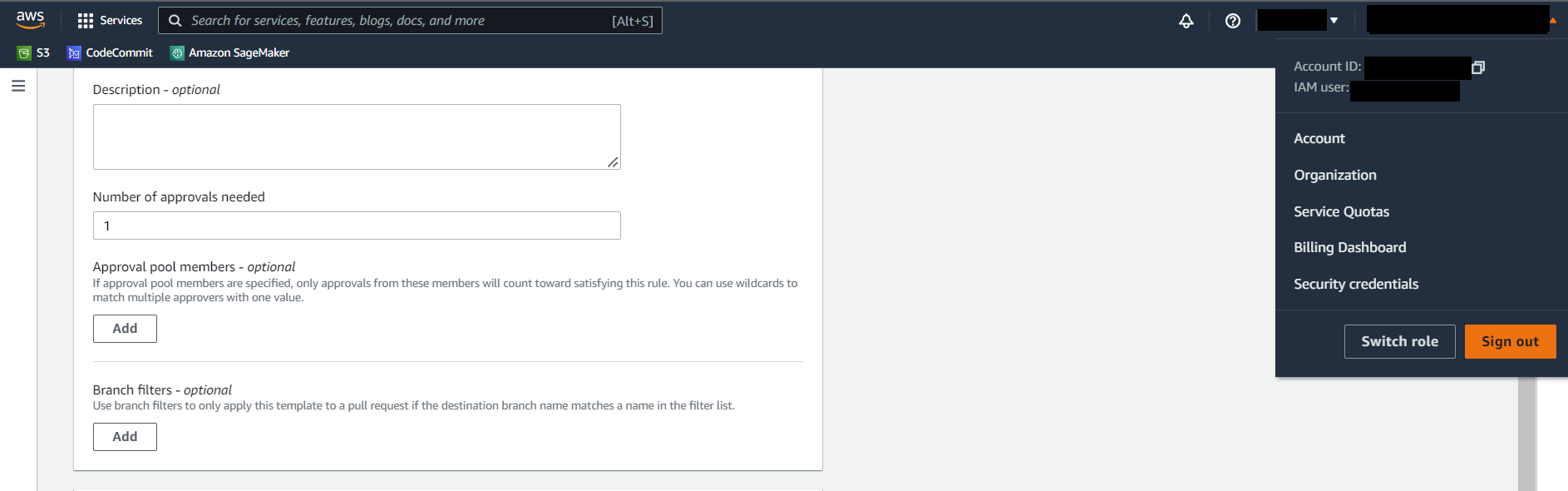
1. In the navigation pane(Right hand side), choose **Approval rule template.**
2. Click the **Create template** buttonand after that you can see the bellow interface.





Please follow the order in the above image and you can fill the gaps in the above image using the following description.

1. Choose an appropriate name for your approval rule.
2. If you need you can add some short description about the rule.
3. Enter the number approval need for the conformation.
4. Press Add button to enter the responsbile persons. It’s better to use IAM user and it means you have to enter their IAM user name.
5. Add the relevant parties’ IAM user names.You can find the IAM user by refering this image.



1. If you need to remove some members, please click the remove button.
2. If you nedd to add this approval template only for some specific branches, please add the necessary branches using add.
3. Specify the branch name in the given space.
4. If you need to remove some branches, please click the remove button.
5. Specify the relavant code repository by clicking the drop down button.
6. Finally, review all the details in the space and if you are happy with the filled details, please click the **create** button to create the approval request.

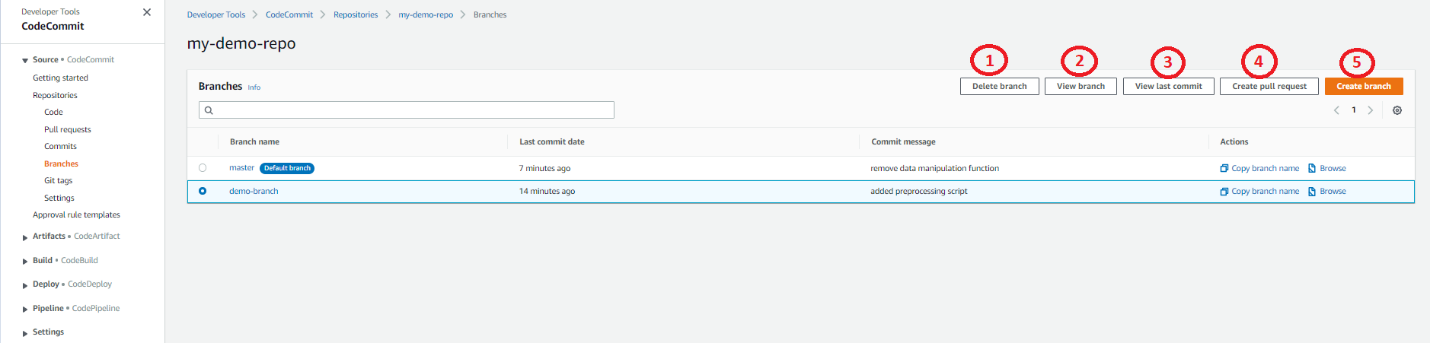
# Step 7: Monitoring

Please follow my Github repository

<https://github.com/Data-Fenix/aws-codecommit-alerting-system>

# Step 8: Manage Branches

Let’s say you need to create,view,delete or view last commits of the branch, this process will help you.

In the navigation pane(Right hand side), choose **Branches.**

By looking at the above image and it has highlighted few buttons.

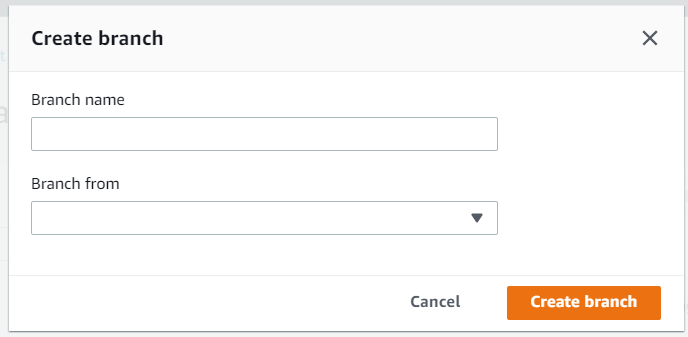
**Delete Branch** : Delete the selected baranch form your repository.

**View Branch**: Select the selected baranch form your repository

**View last commit**: To see the commit history of the selected baranch

**View last commit**: Another option to create a pull request.

**Create branch**: To create a new branch and after clicking this option it will appear bellow window.

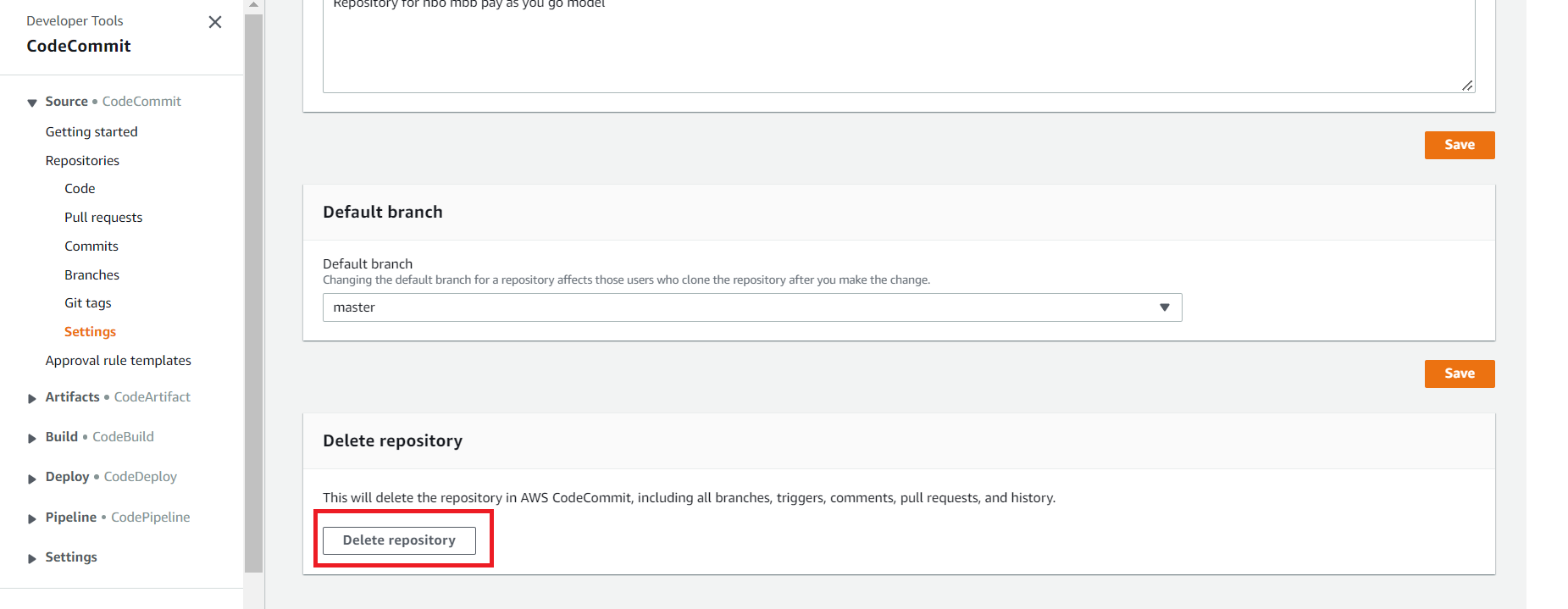


You can create a new branch after enter a suitable barnch name and reference branch in the empty boxes.

# Step 9: Clean Up

If you need to delete this repository, follow the bellow steps:

1. Open the CodeCommit console using  <https://console.aws.amazon.com/codesuite/codecommit/home>.
2. In **Repositories**, choose the relavant repository and in this case my-demo-repo.
3. In the navigation pane, choose **Settings**.
4. In the **Settings** page, in **Delete repository**, choose **Delete repository**.
5. Type **delete**, and then choose **Delete**. This will permanently delete your repository from the CodeCommit.



Now you are faimiliarize with the CodeCommit and you can start store your files, codes and other documents in the CodeCommit.