# Continuous Integration Pipeline Using AWS CodeCommit, CodeBuild and CodePipeline

Created by[Zeal Vora](https://deloittedevelopment.udemy.com/user/cybercorp/)

**AWS Workspace**

**60-120 minutes**

In this lab, you will be a consultant at an E-Commerce based startup that wants to create a simple solution that can quickly build the application code whenever the source code is committed to the Git repository. The architecture group thinks that using the AWS CodeBuild service will allow developers to easily build and test their code with continuous scaling since it is a managed continuous integration service offering from AWS. In addition, making use of the AWS CodePipeline service will allow automating the entire process. Your boss wants you to build a proof-of-concept.

The Engineering team at a company has raised an issue. Every time a developer commits their code to the repository, they have to manually trigger a single EC2 instance-based Jenkins pipeline to build and test the code. Often, the Jenkins server slows down and stops responding, resulting in a lot of time wastage. This approach is a time-consuming task for the developers. They would just like to focus on pushing their application code to the Git repository, and the rest should be taken care of automatically by a more stable pipeline. I want you to create an automated pipeline using AWS services that can achieve this. In addition, developers should be notified via email if their build fails.

The environment should be based on the AWS CodeBuild service. Developers make use of AWS CodeCommit for storing their application code. All that the developers must be required to do is to commit the application code in the CodeCommit repository, and the rest steps of building code and reporting the failure status must be taken care of by the pipeline service automatically. Developers have provided you with a sample application code to build a POC pipeline. The application code is stored in demo.cpp file. This code is based on C++. Developers have also provided you with the necessary commands required to compile the code in a Linux environment, these instructions can be found in the file base-prerequisite.txt

**How you'll work**

Your project has been broken into a set of tasks. To complete these tasks, use the provided workspace. You can launch your workspace by clicking below or using the button in the top right of the screen.

Launch workspace

Tasks

1-Set up AWS Codecommit repository to store the application

2-Add sample code to the Codecommit repository

3-Create buildspec.yml file for Code Project

4-Create CodeBuit Project

5-Create SNS Topic

6-enable Codebuil Notification

7-Create CodePipeline

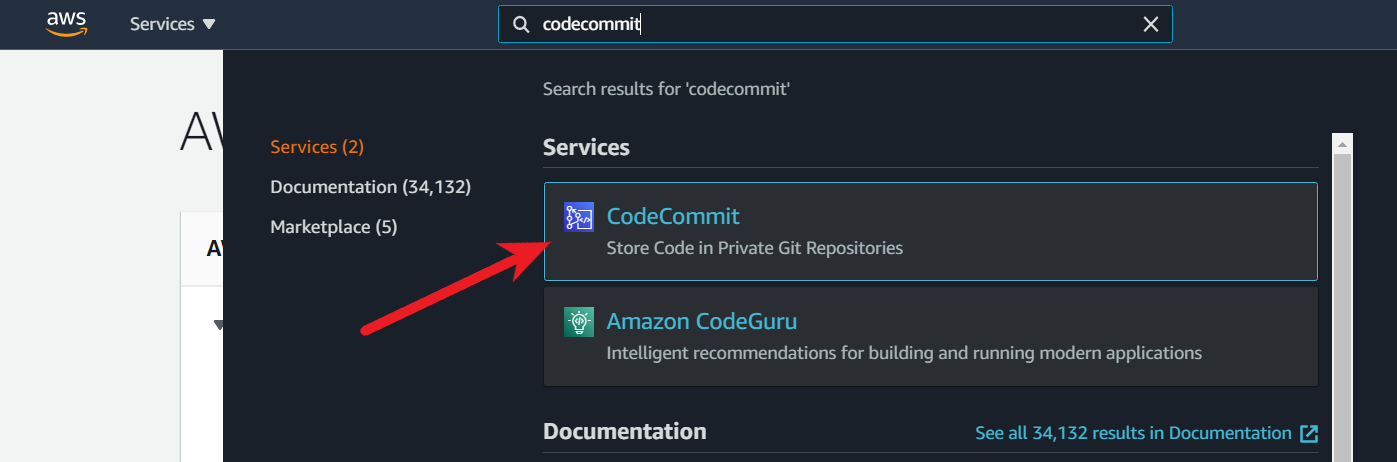
8-Verify Fail Notification

9-Cleanup

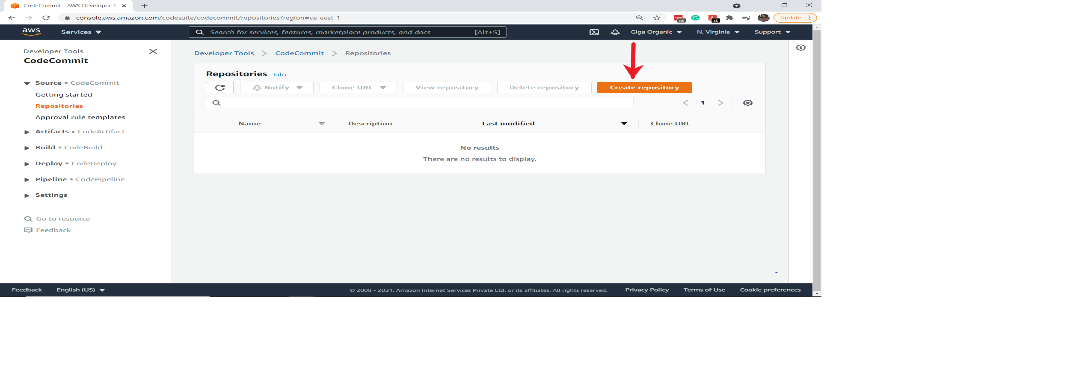
Set up AWS Codecommit repository to store the application

You need to create a new repository in AWS CodeCommit where the application code can be stored. Eventually, you will have to connect this repository to the pipeline, so any new update pushed to this repository is automatically deployed to a set of EC2 instances.

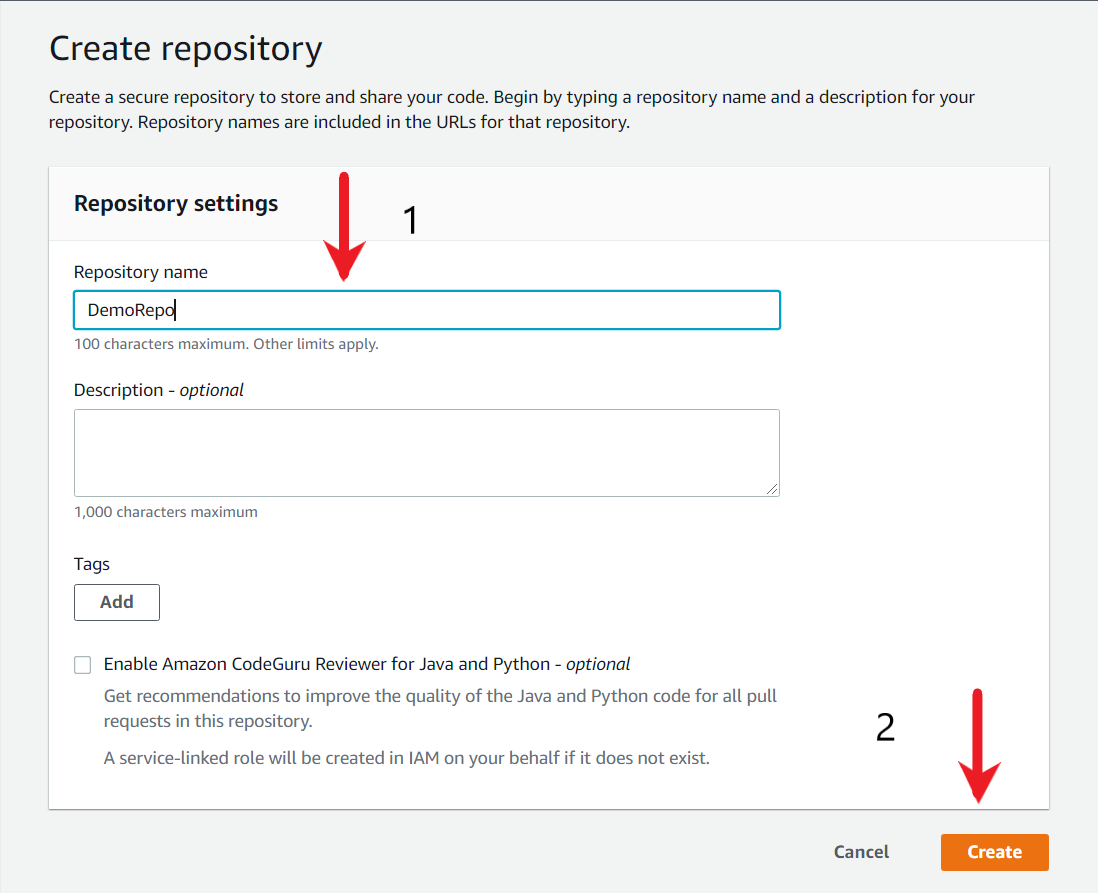
1. From the AWS console, search for the CodeCommit service and click on the available option.



1. On the Repositories page, choose Create repository.



1. On the Create repository page, in Repository name, enter a name for your repository (for example, DemoRepo) and click on Create.

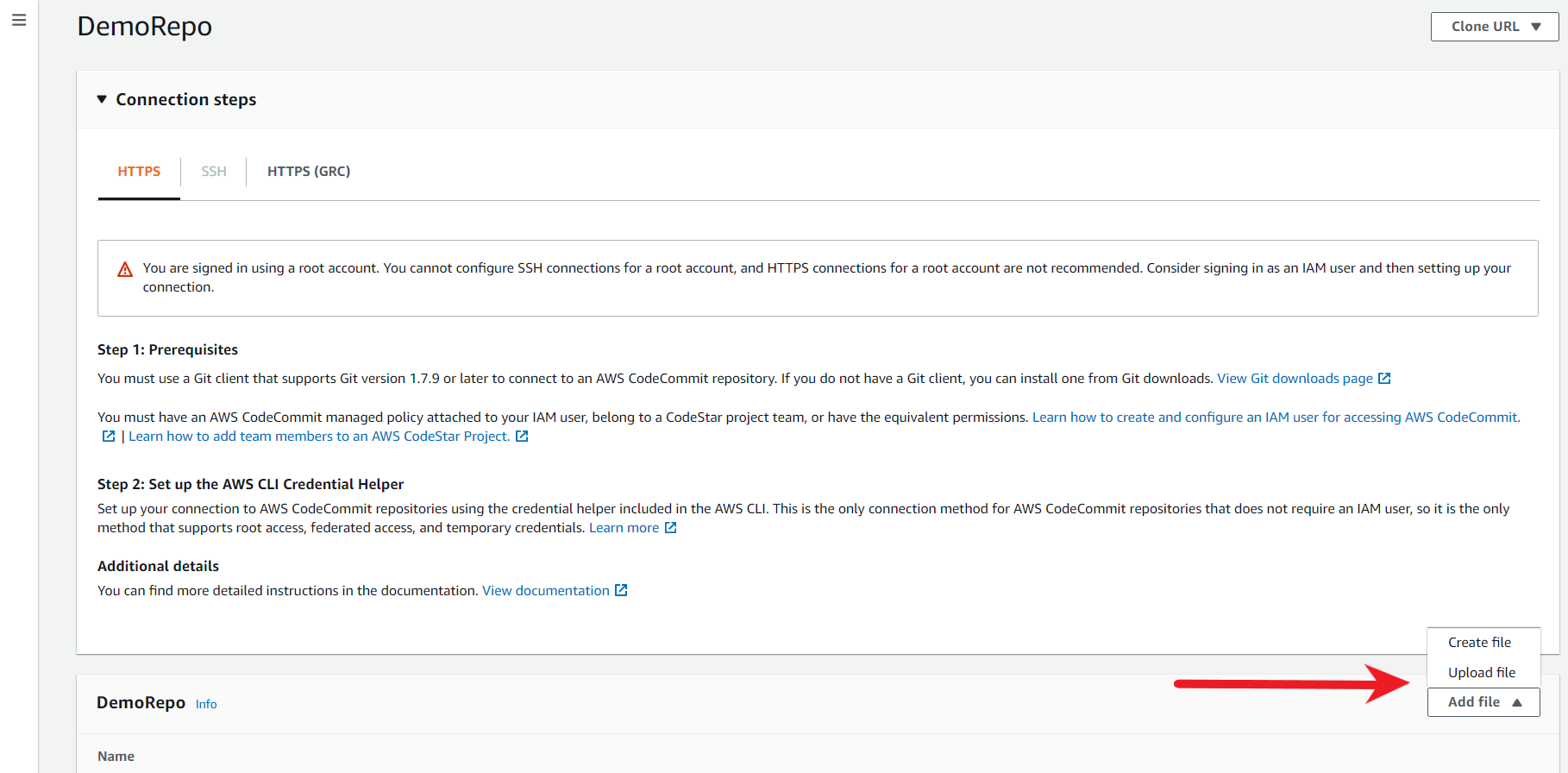


<https://docs.aws.amazon.com/codecommit/latest/userguide/getting-started-cc.html>

Add sample code to the Codecommit repository

The sample C++ code that the developers have provided to you needs to be committed as part of the CodeCommit repository. Commit the contents of the demo.cpp file within your existing repository.

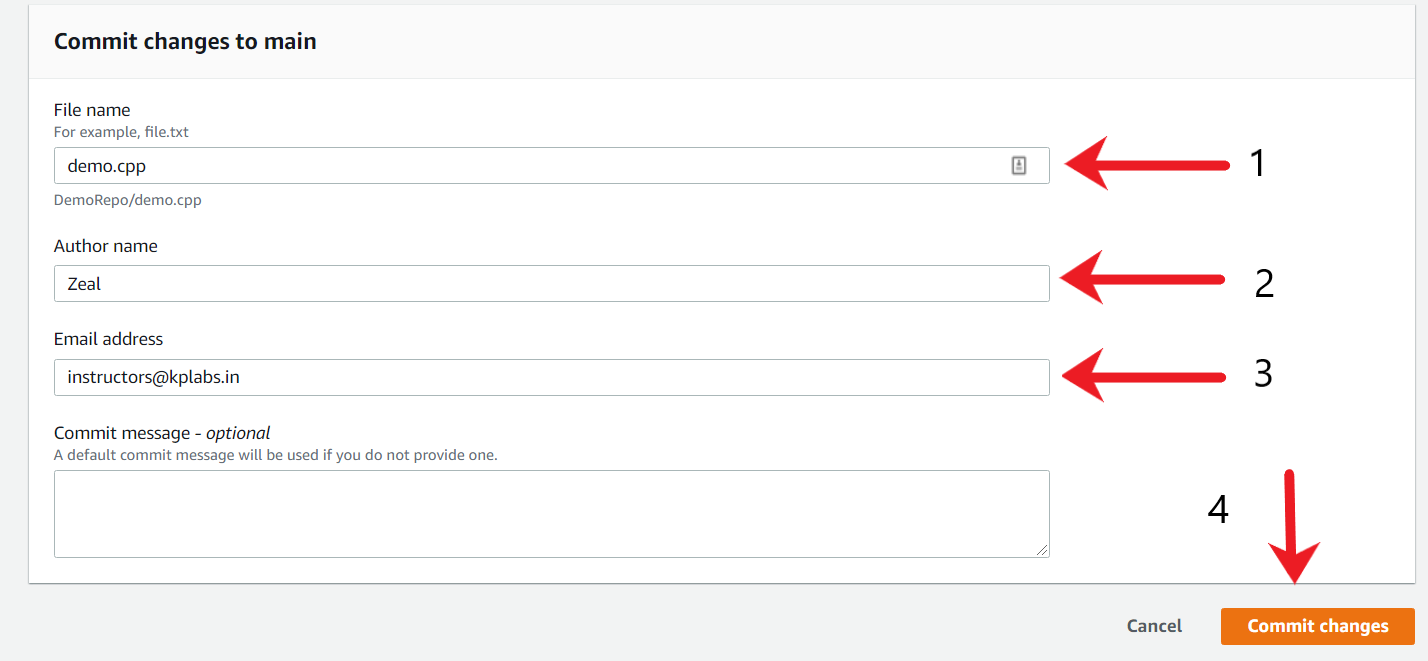
1. Open the CodeCommit console, and choose your repository from the Repositories list.
2. Click on the Add file button, and then choose Create a file. You can easily open the demo.cpp file with Notepad to see its contents. Alternatively, you can also directly upload the file.



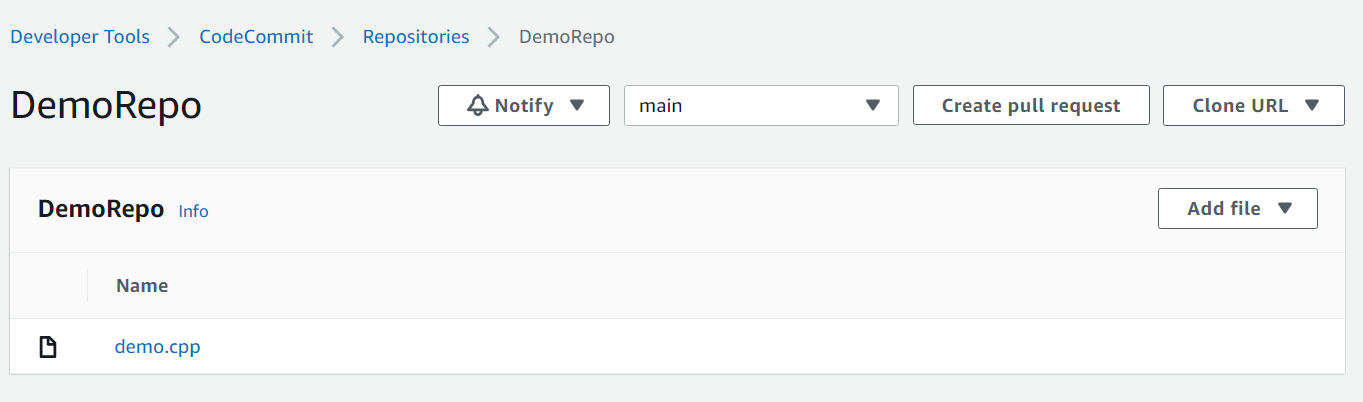
1. Copy the contents of the demo.cpp file from your workstation to CodeCommit code-editor.



Scroll to the bottom page and add the File Name as demo.cpp followed by Author name and email address post which click on the "Commit changes" button. This will add the demo.cpp file inside the repository.



1. After the file has been committed, go back to the DemoRepo repository and the demo.cpp file should be visible on the home page.

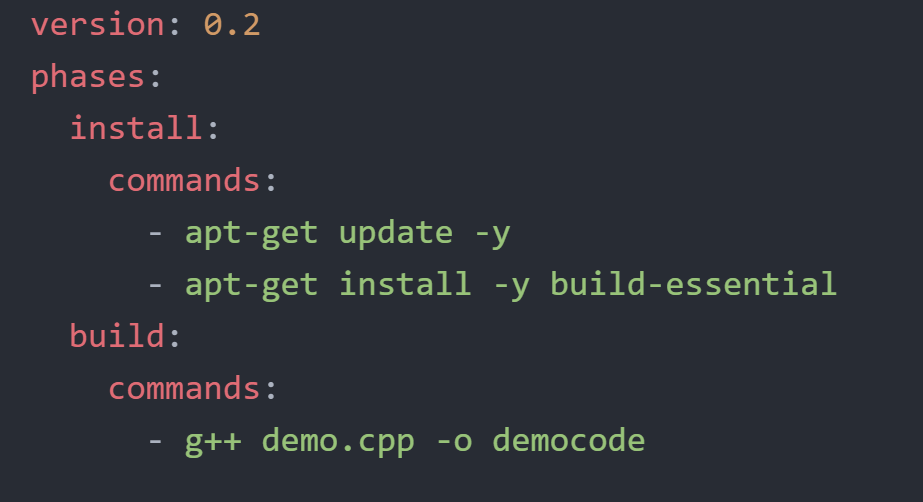


Ref demo.cpp

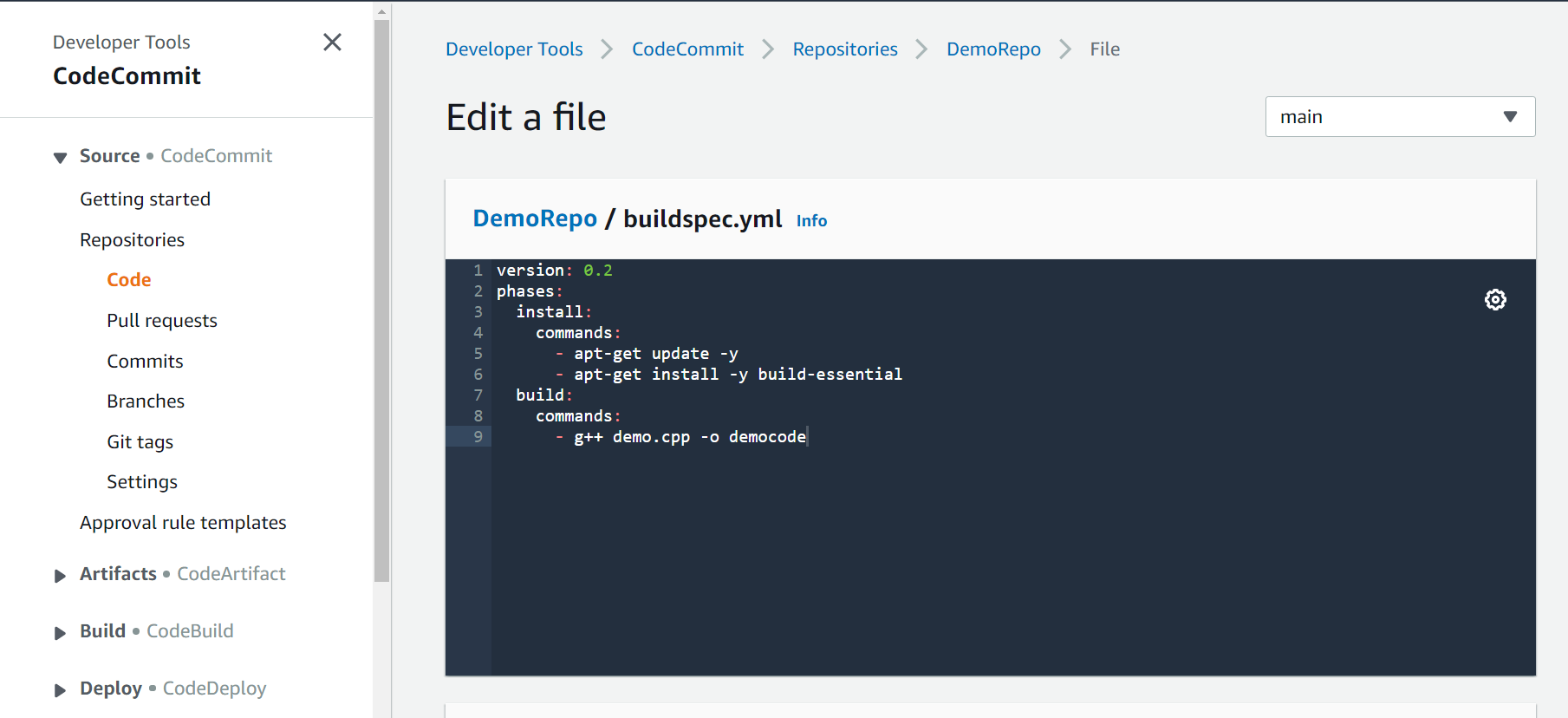
Create buildspec.yml file for Code Project

CodeBuild service needs to be provided with instructions required to build the application code. Create a custom buildspec.yml file based on the build commands that the developer has provided. Using the buildspec.yml, the CodeBuild service should be able to build the application code.

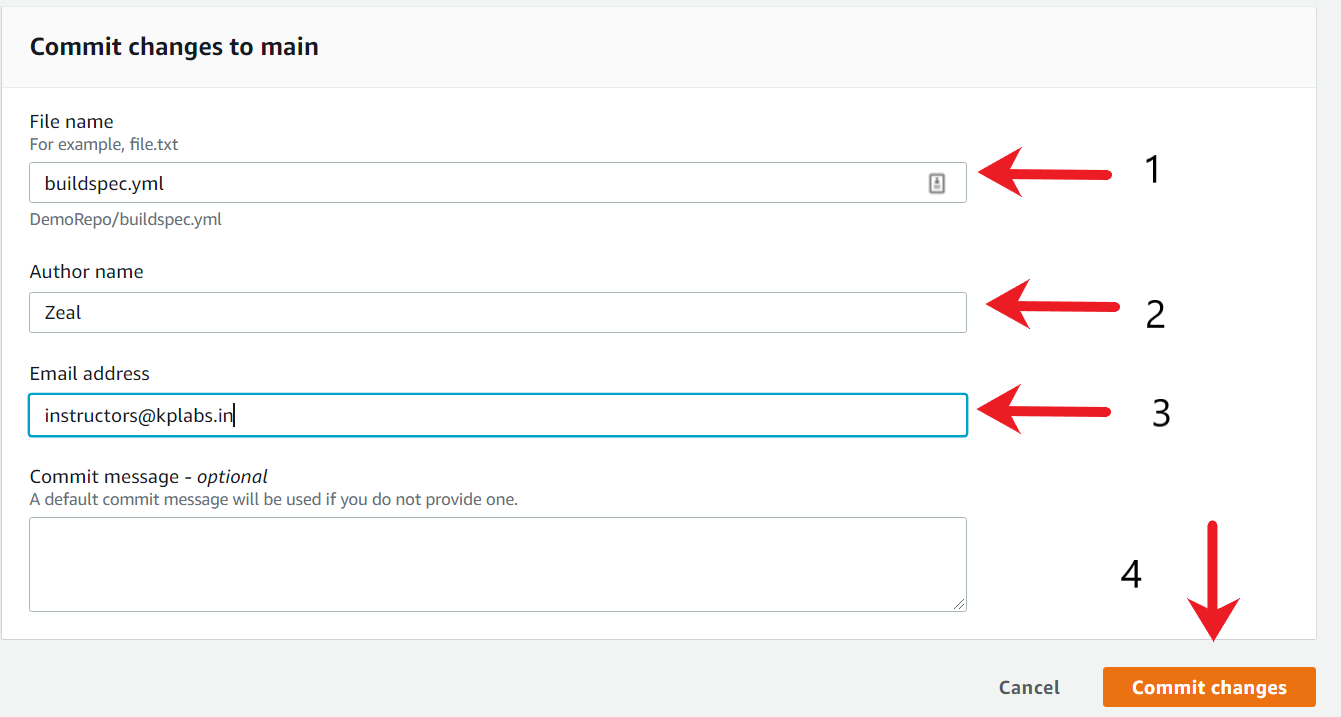
1. Based on the requirements given by the developer, here is a translation of it into buildspec.yml file.



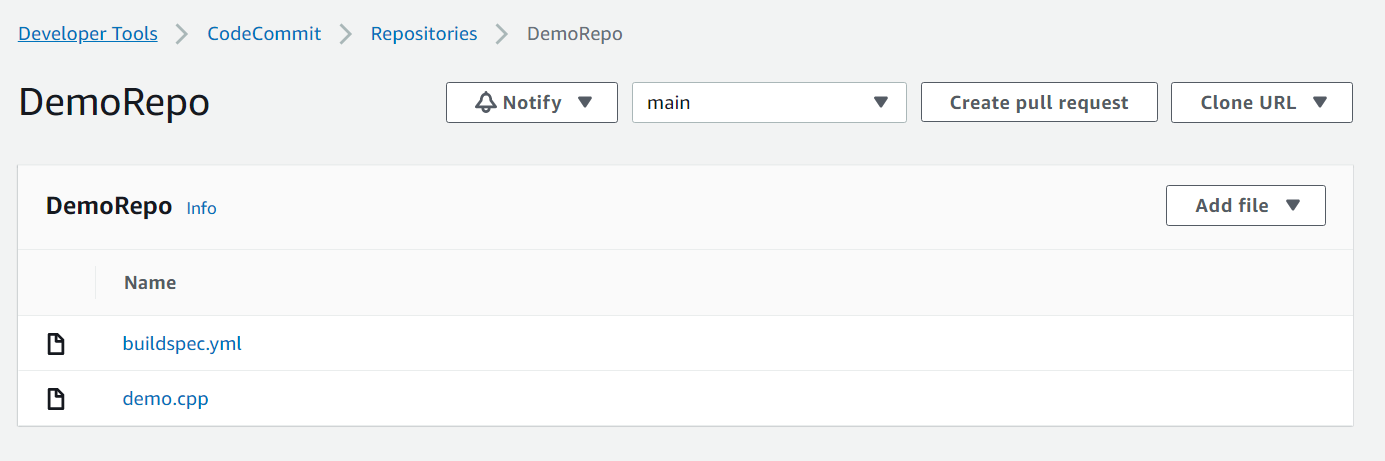
1. Once the buildspec.yml file is created, you have to add it as part of the CodeCommit Repository. CodeBuild will refer to the instructions in this file for building the code. To add the file, go to your CodeCommit repository, click on the Add file button, and then choose Create a file. Copy the contents of the buildspec.yml file from your workstation to CodeCommit code-editor.



1. Scroll to the bottom page and add the File Name as buildspec.yml followed by Author name and email address post which click on the "Commit changes" button. This will add the buildspec.yml file inside the repository.



1. After the file is committed, go back to the DemoRepo repository, and the buildspec.yml file should be visible on the home page.



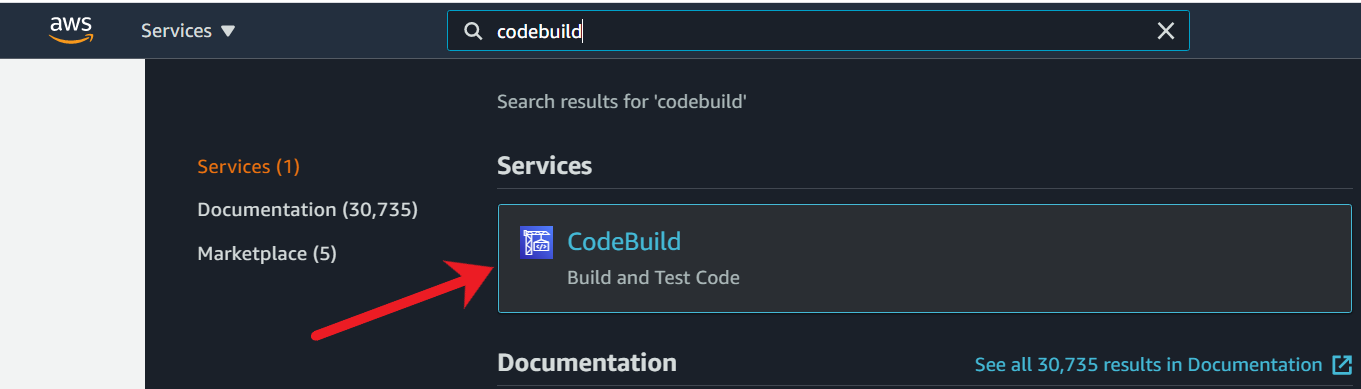
Ref build-prerequisite.txt

<https://docs.aws.amazon.com/codebuild/latest/userguide/build-spec-ref.html>

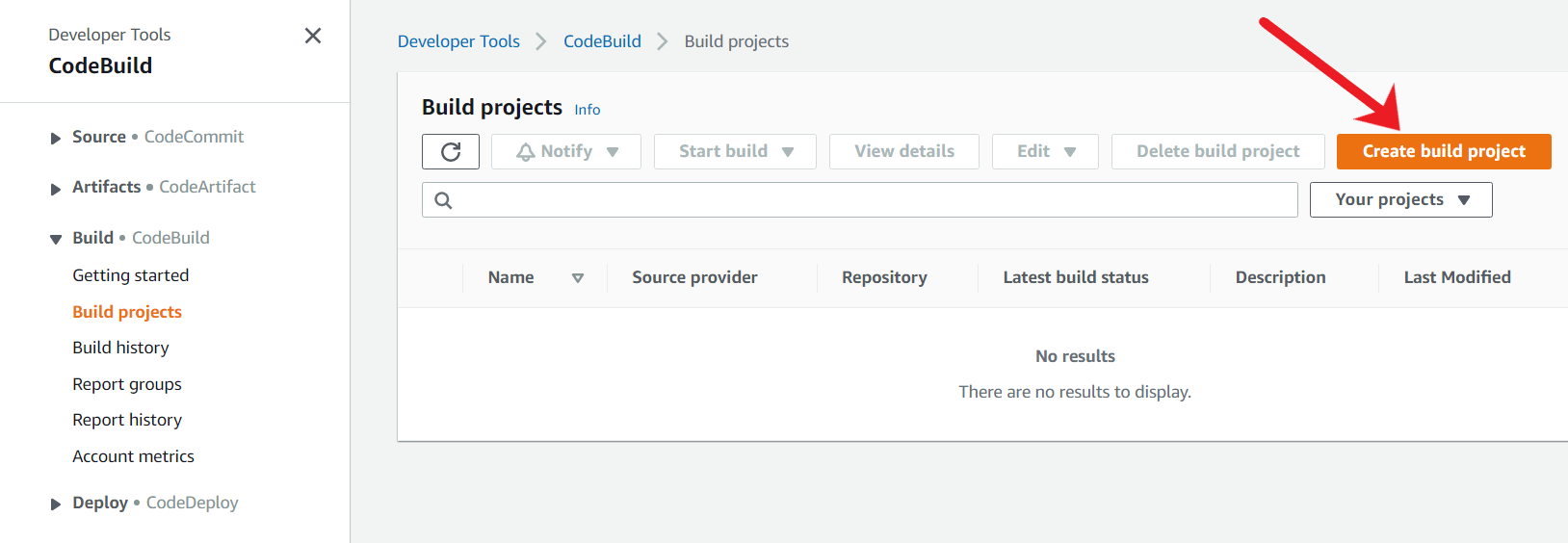
Create CodeBuit Project

You have to create a CodeBuild project that integrates with the CodeCommit repository. The CodeBuild project should be able to fetch the application and perform the build process successfully.

1. From the AWS console, search for the CodeBuild service and click on the available option.



1. If a CodeBuild information page is displayed, choose **Create build project**. Otherwise, on the navigation pane, expand Build, choose Build projects, and then choose Create build project.

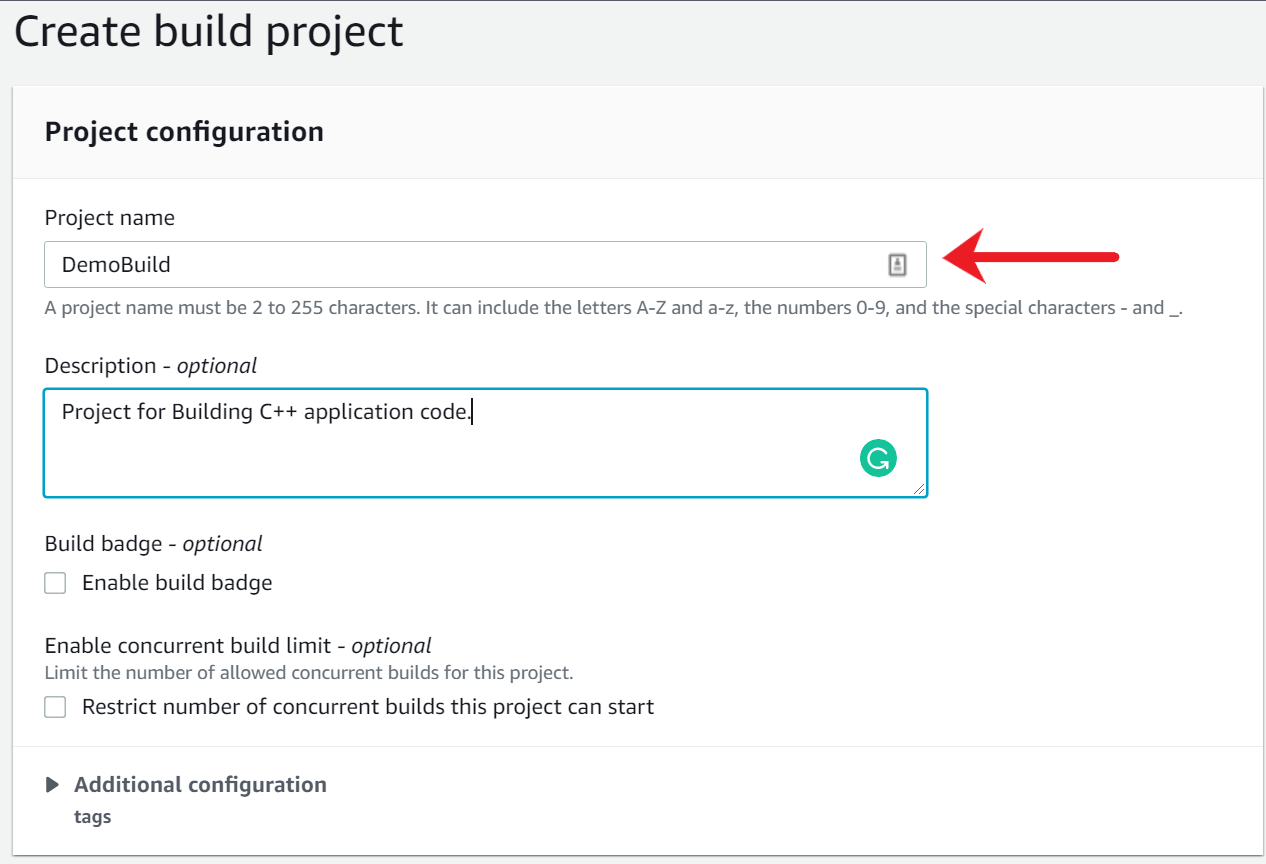


1. Fill in the following sections. Once complete, choose Create build project at the bottom of the page.
   * Project configuration
   * Source
   * Environment
   * Buildspec
   * Batch configuration
   * Artefacts
   * Logs

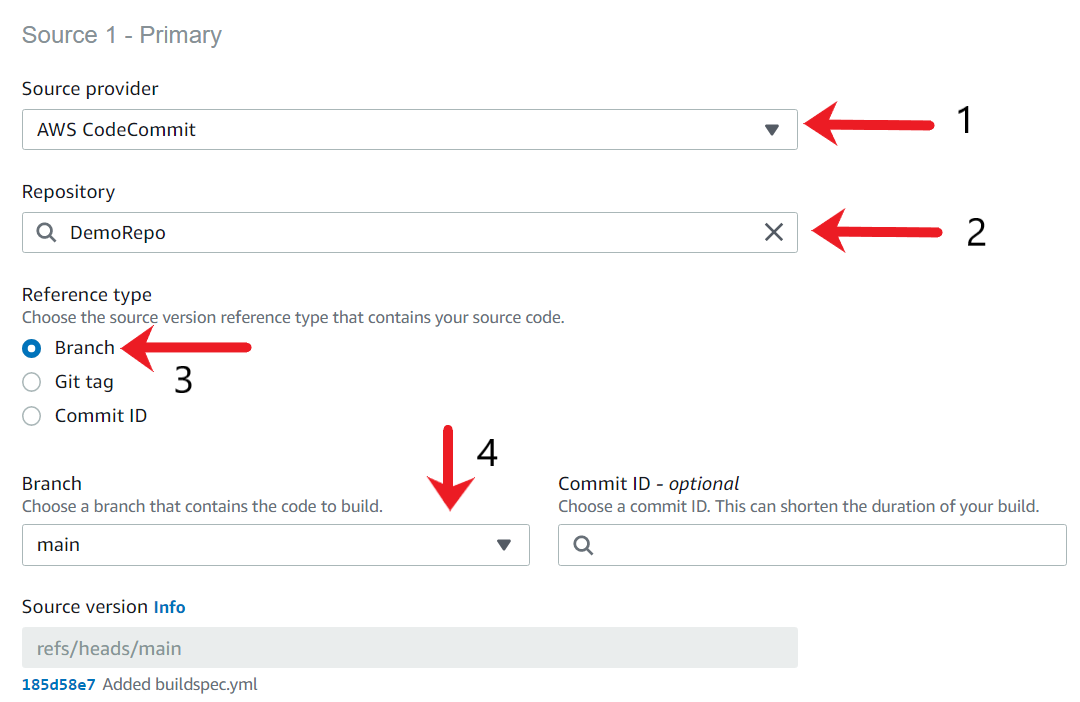
Under the Project Configuration section, add the following data:

* + Project name: DemoBuild

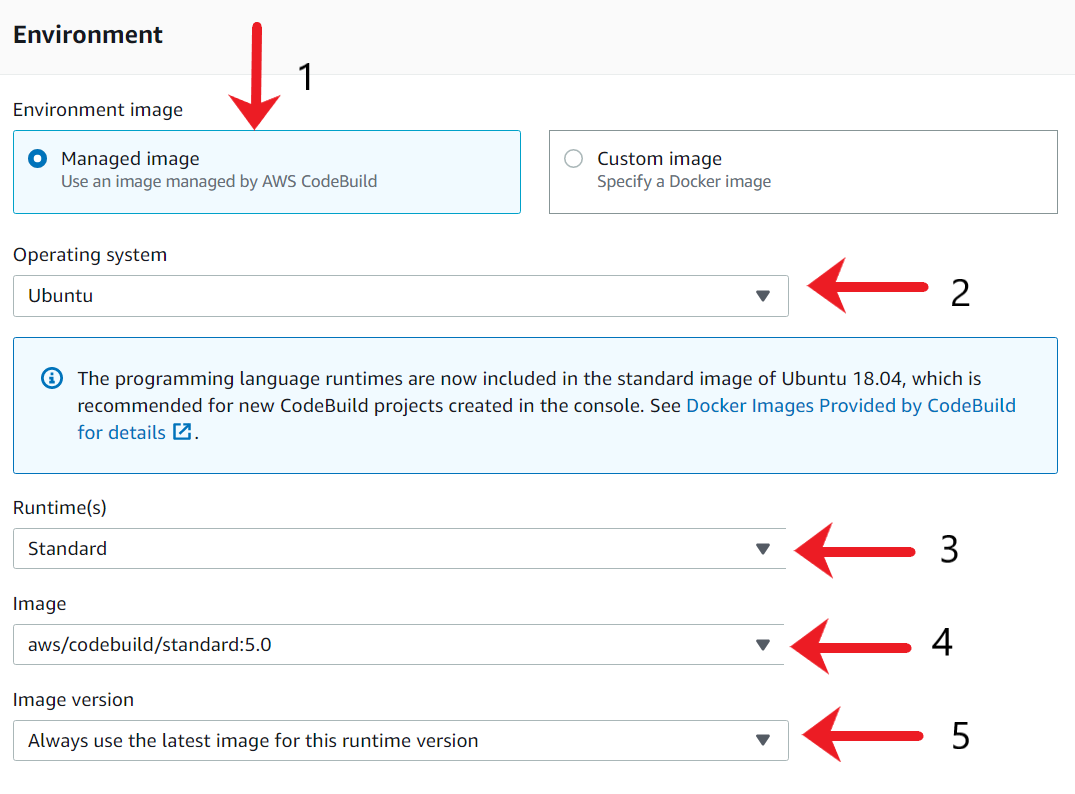
You can also add an optional description if required.

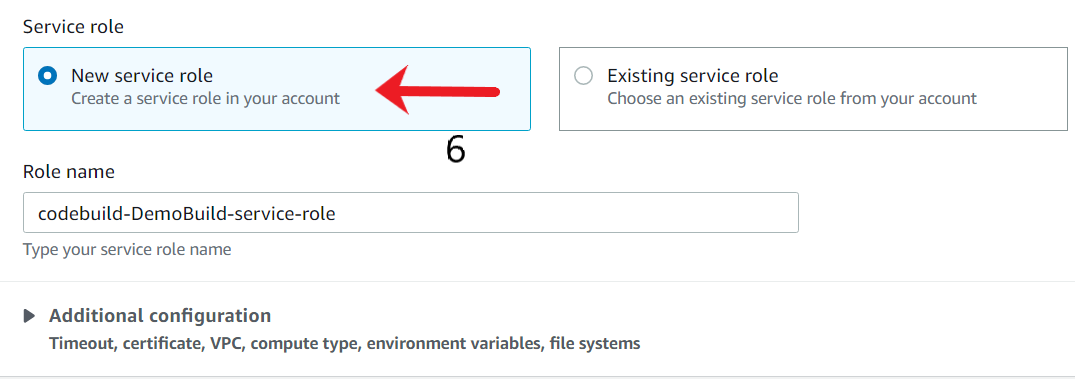


1. Under the Source section, add the following details:
   * Source Provider: AWS Code Commit
   * Repository: DemoRepo
   * Reference type: Branch
   * Branch: main

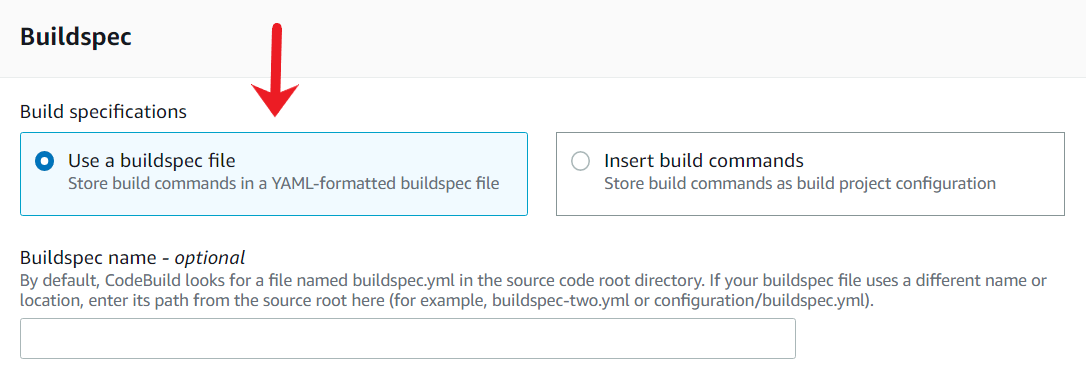


1. Under the environment section, add the following data
   * Environment image: Managed image
   * Operating system: Ubuntu
   * Runtimes: Standard
   * Image: aws/codebuild/standard:5.0
   * Image Version: Always use the latest image for this runtime version
   * Environment Type: Linux
   * Service Role: New Service Role

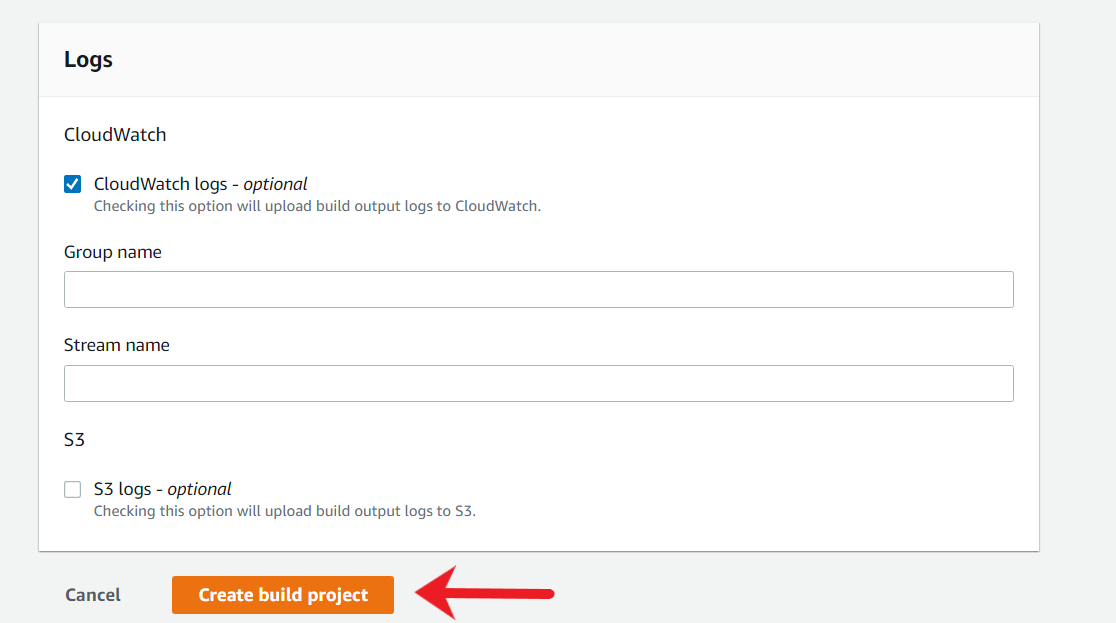




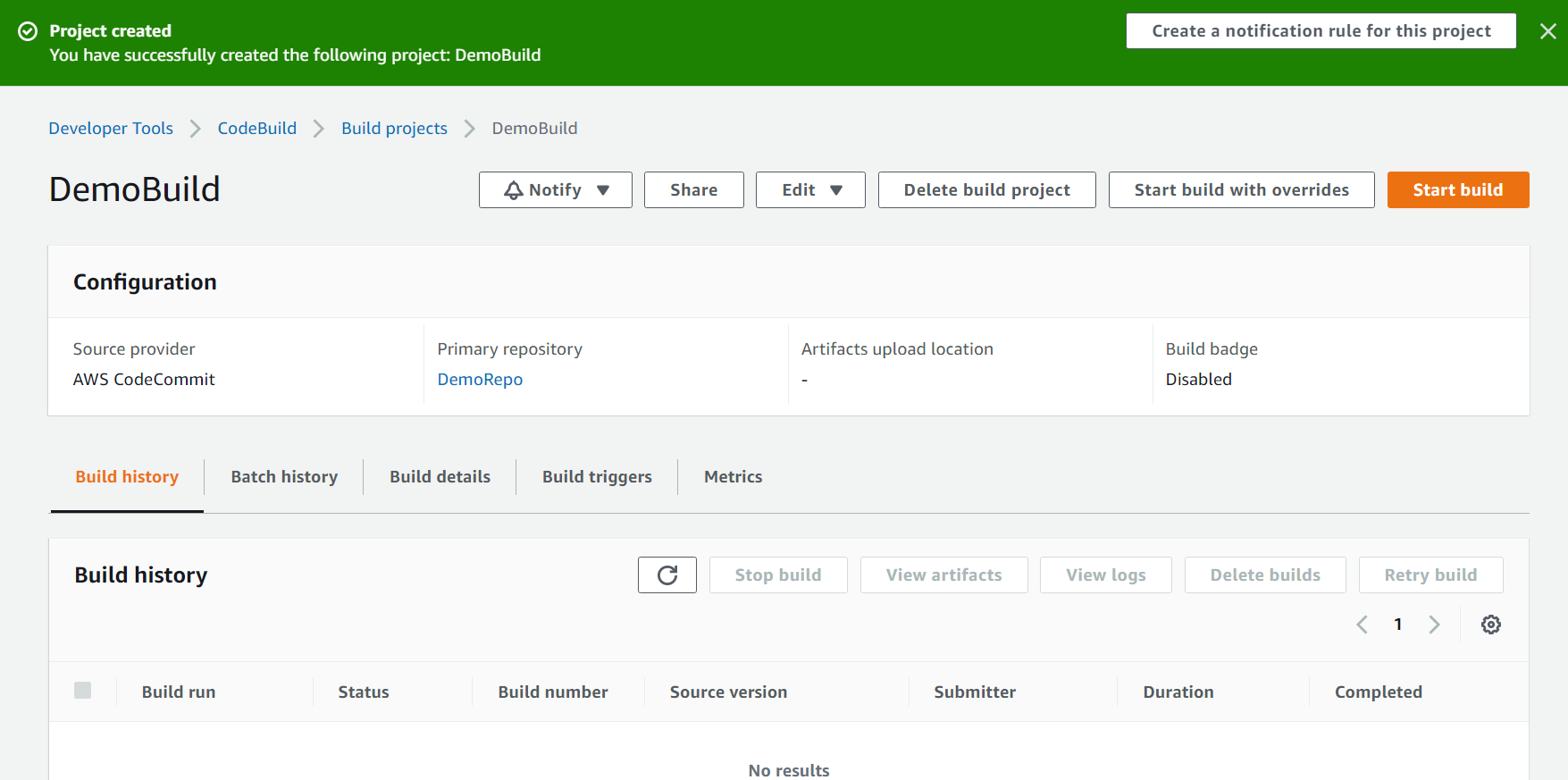
1. Under the Buldspec section, ensure the "Use a buildspec file" option is selected.



1. Once the configurations is completed, go to the bottom of the page and click on the "Create build project" button.



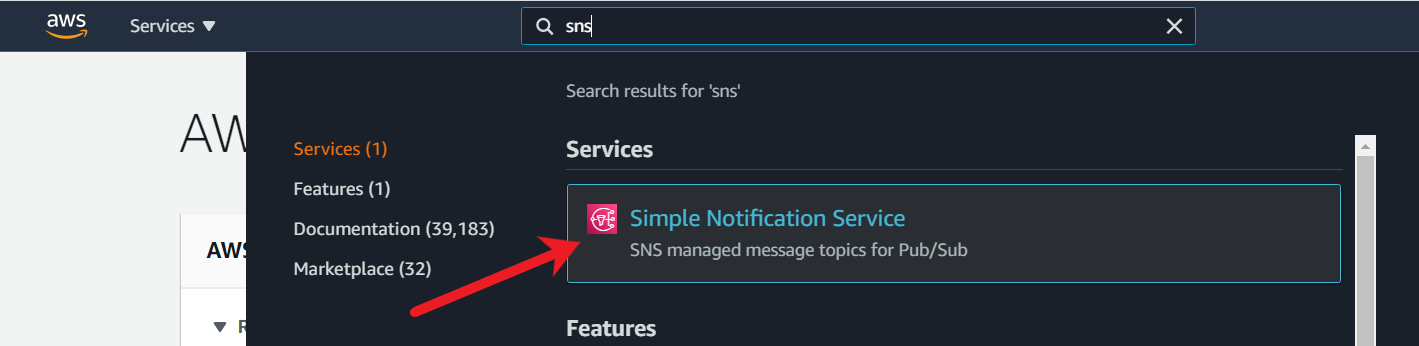
1. After the build project is created successfully, you will be redirected to the console of your CodeBuild project.



Create SNS Topic

You need to create an SNS topic that will allow CodeBuild to send email notifications to the developers in case if there is a build failure. Create an SNS topic and provide your email for testing this use case.

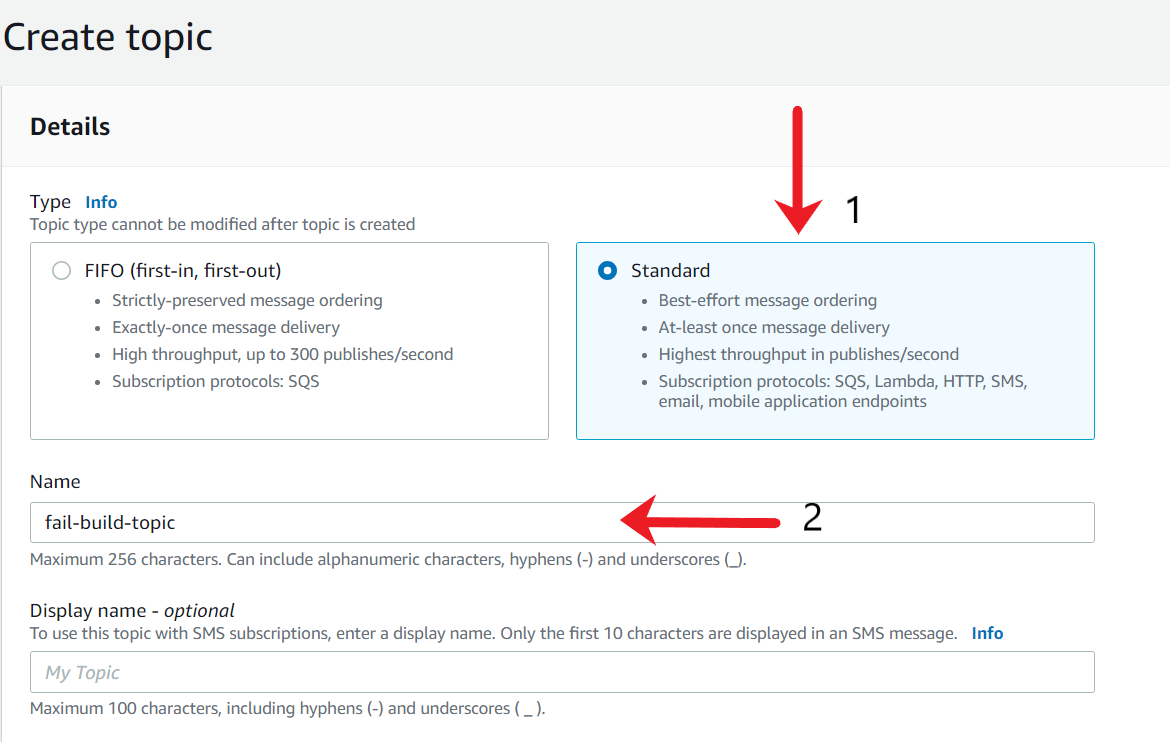
1. From the AWS console, search for the SNS service and click on the available option.



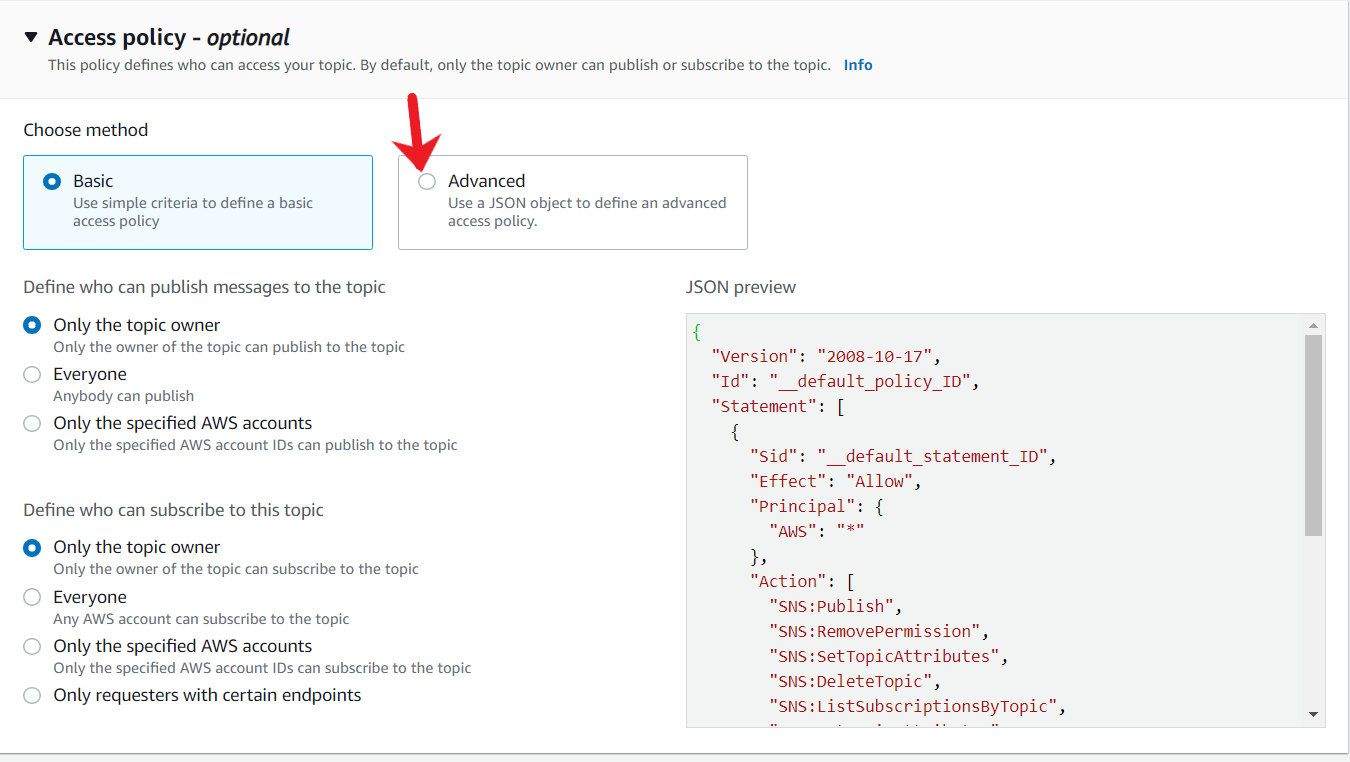
1. Do one of the following:
   * If no topics have ever been created, type the topic name on the main page and click on the Next step. For this example, I have give topic name as fail-build-topic.
   * If topics have been created under your AWS account before, on the navigation panel, choose Topics. On the Topics page, choose Create topic.



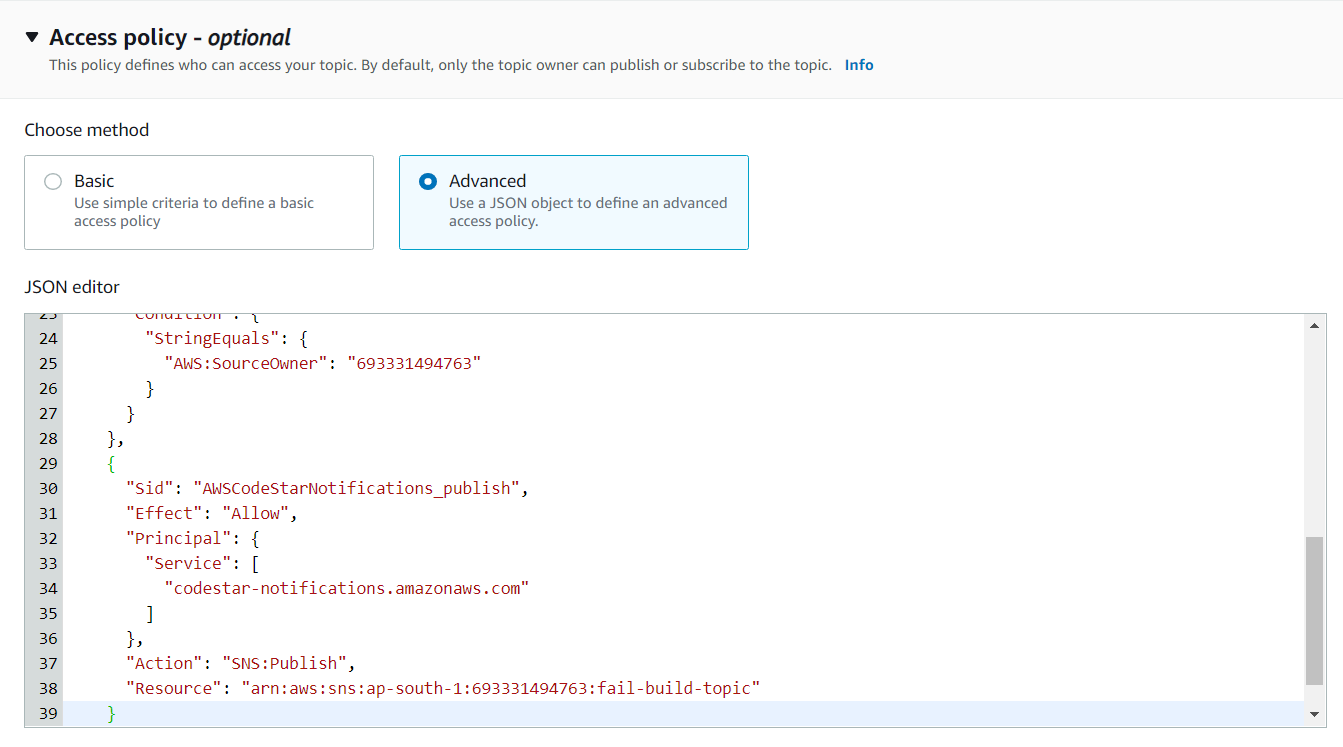
1. On the Create topic page, in the Details section, do the following:
   * Type: Standard
   * Name: fail-build-topic



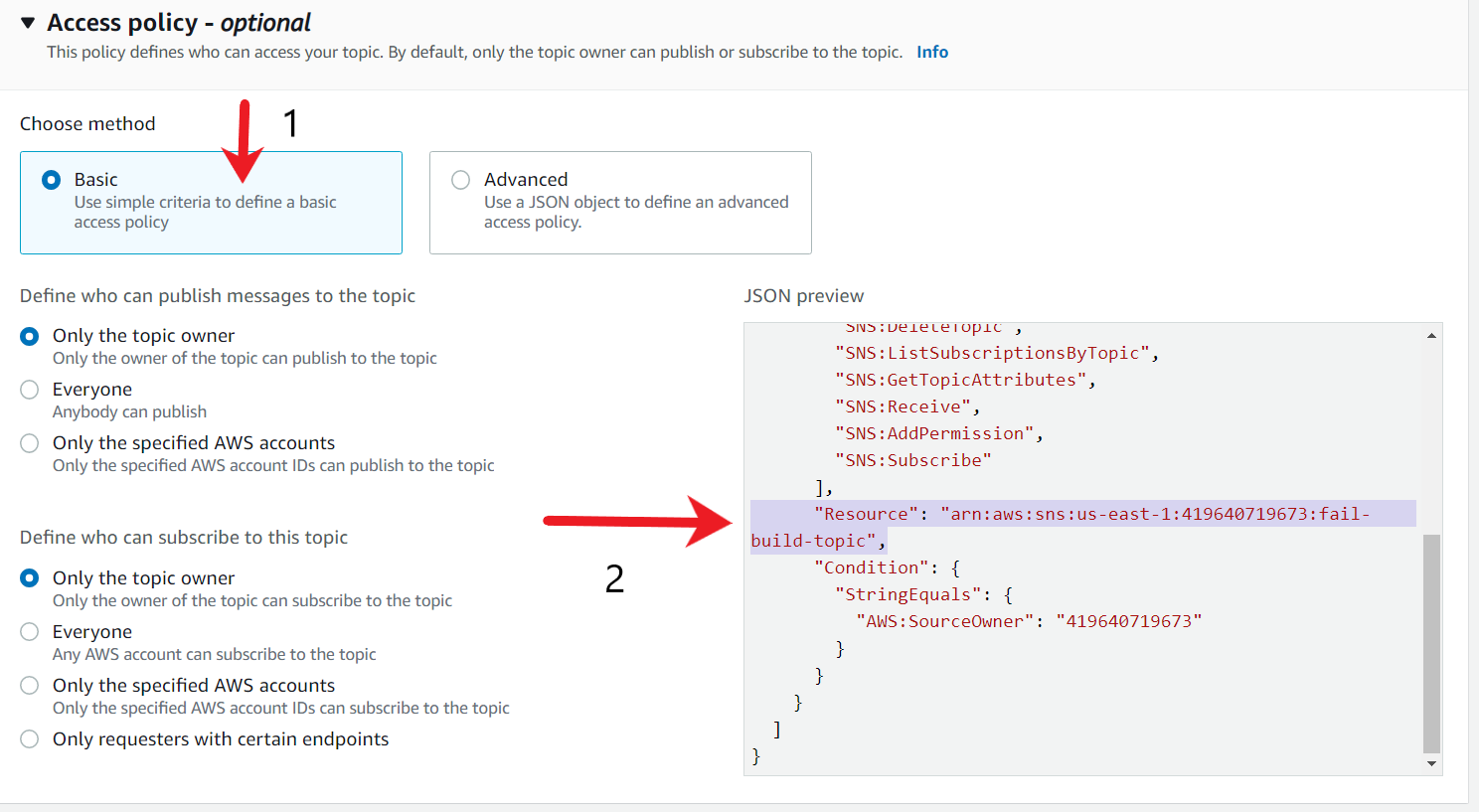
1. Scroll down to the "Access Policy - optional" section and expand it. Under the Choose method, click on "Advanced".



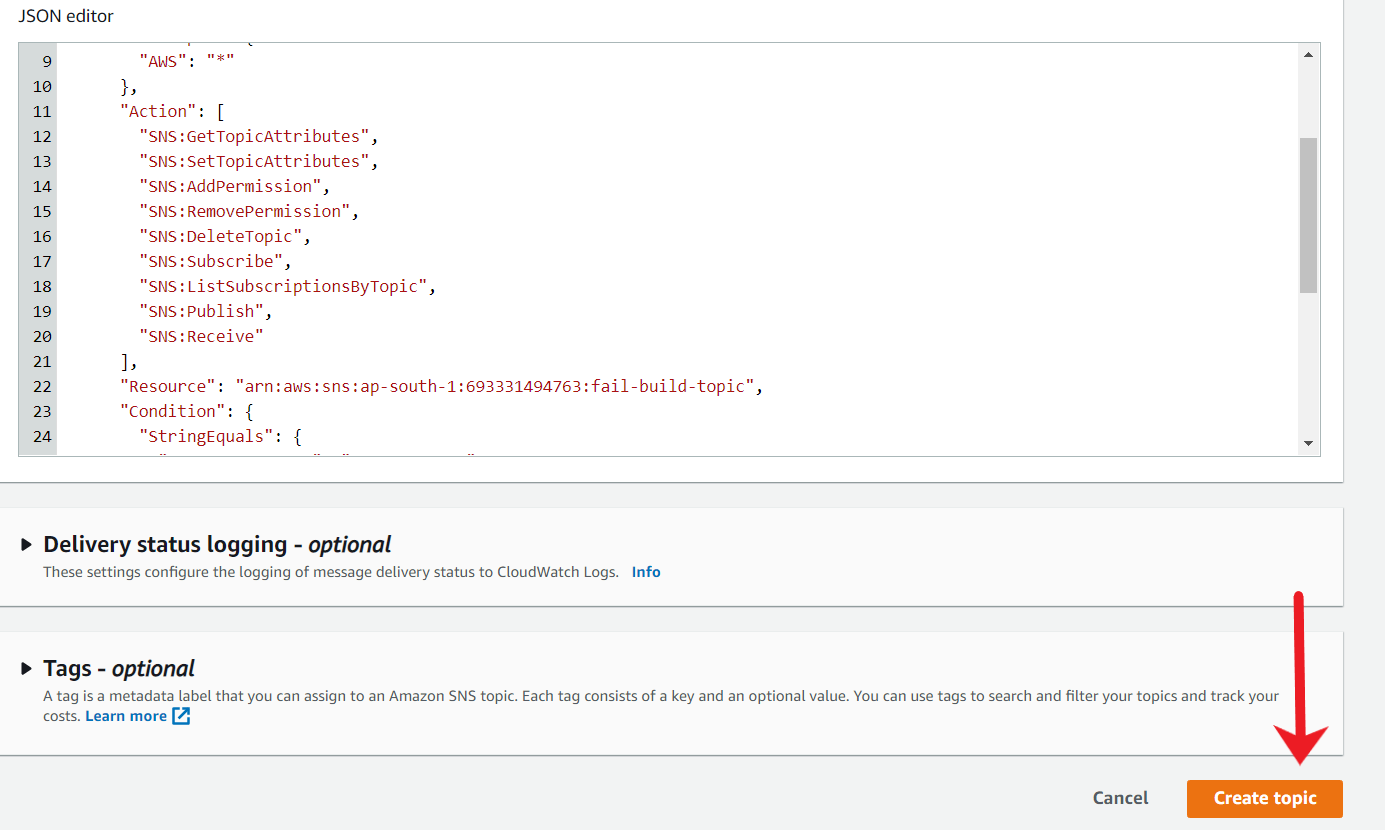
1. In the Advanced tab, create a new policy in JSON editor. The policy should allow CodeBuild to publish messages to the SNS topic. Ensure that the Resource element has values of "Region", "ARN", "Account ID" and "Topic name". Also, ensure the Account owner is correct in the policy. The base template of this policy is part of the Assets file named sns-access-policy.json.



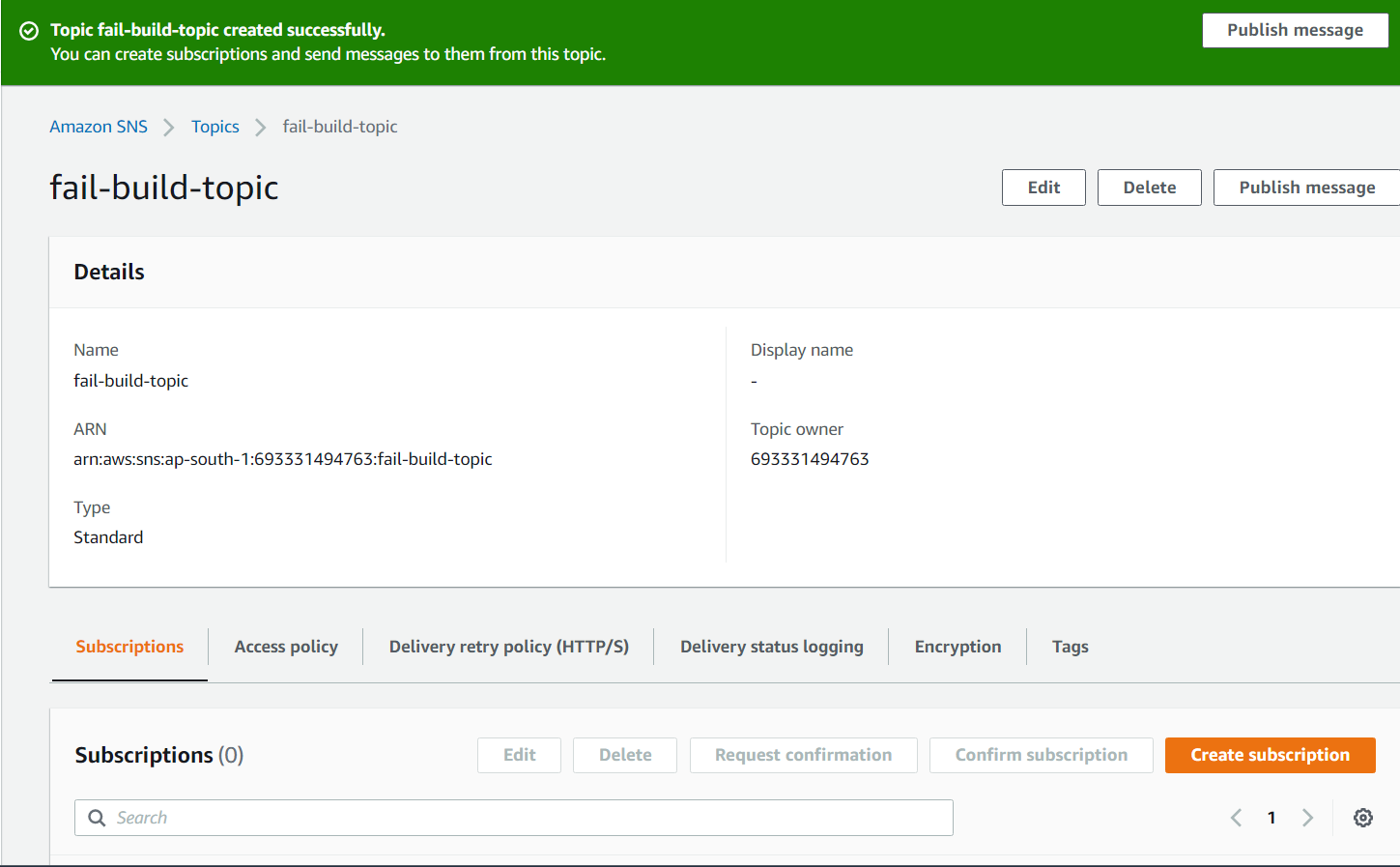
If you want to quickly get the ARN and Account Number required for the custom SNS Access policy, you can refer to the sample policy provided under the "Basic" method.



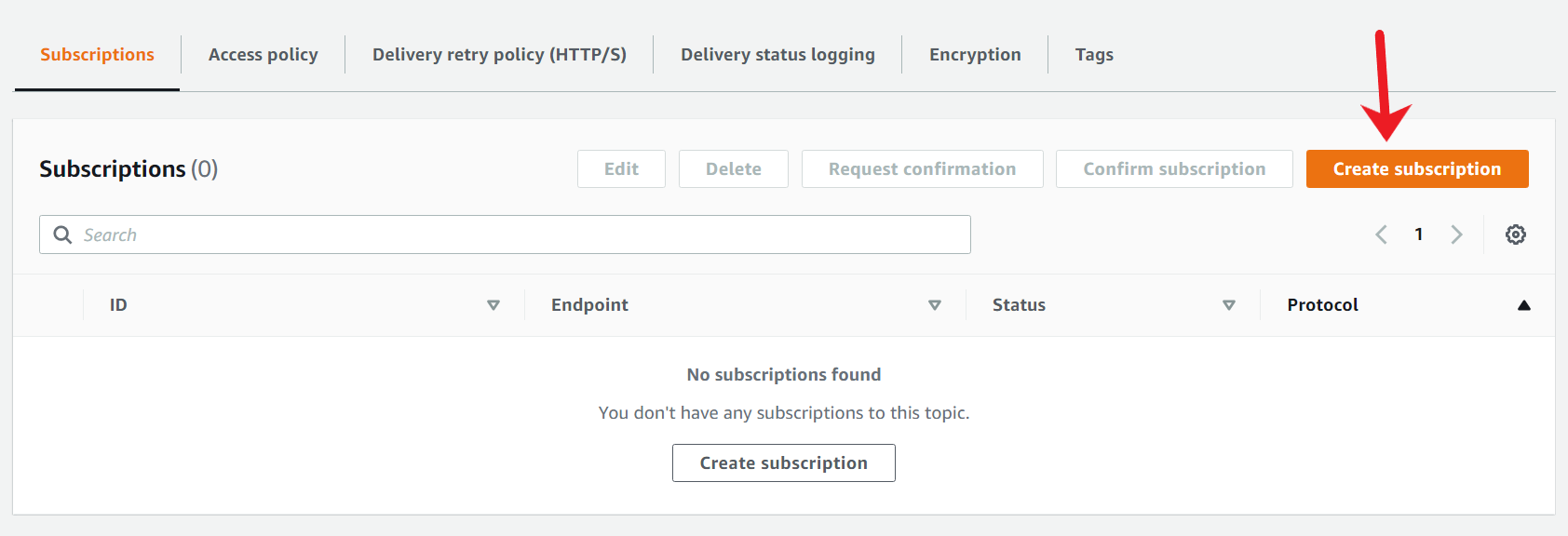
1. Once appropriate Access Policy is added, scroll to the bottom of the page and click on **Create topic**.



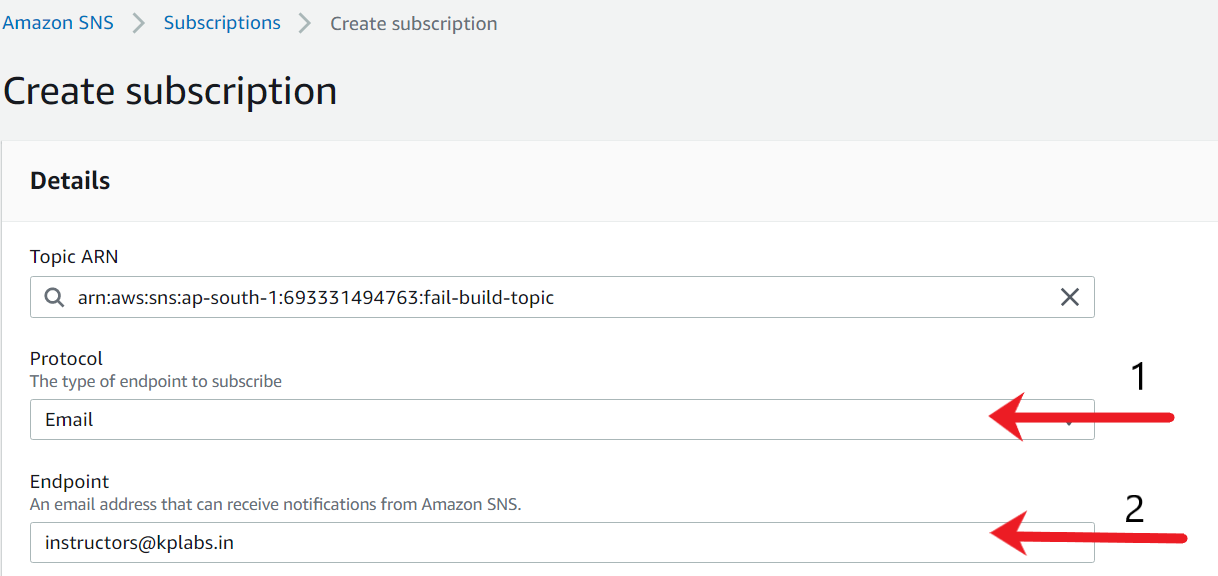
1. After the topic is created, you will see a success message and will be re-directed to the console page of the Topic.



1. Scroll to the bottom of the page, under Subscriptions tab, click on **Create subscription**.



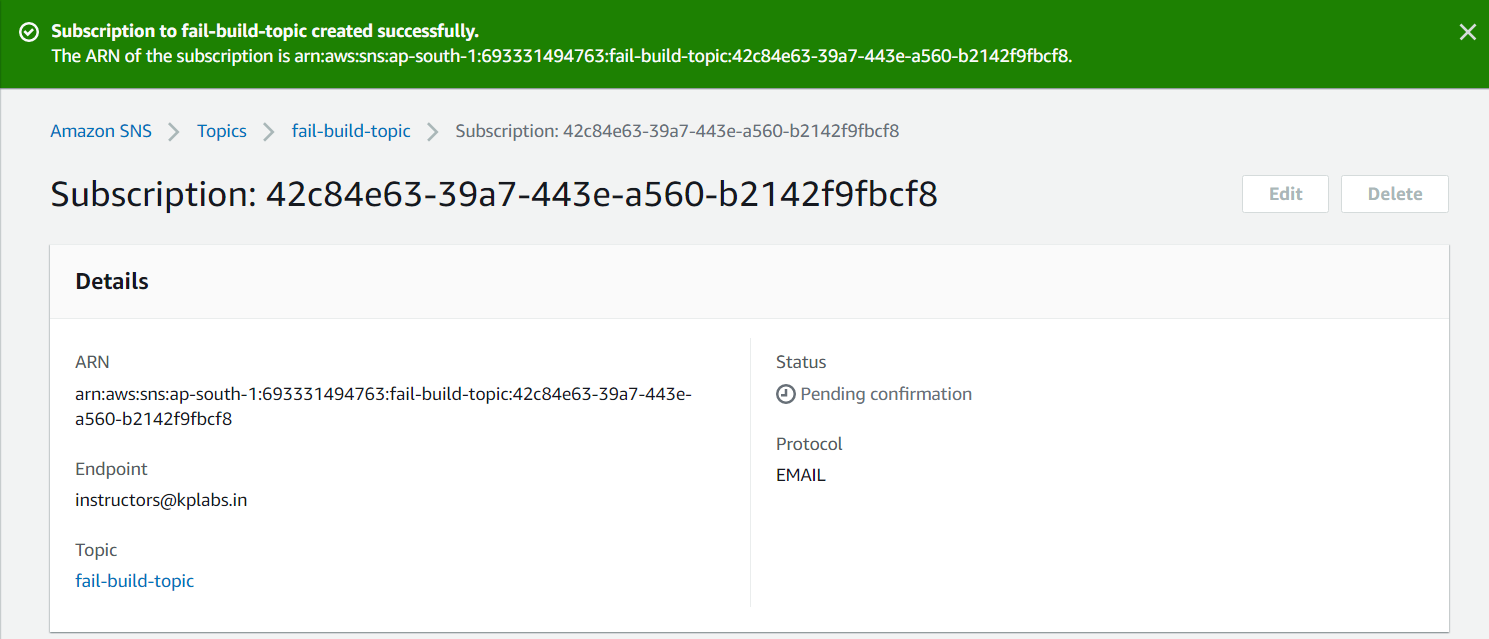
1. For Protocol, choose an endpoint type as Email. For Endpoint, enter the endpoint value, in this case, your email address.



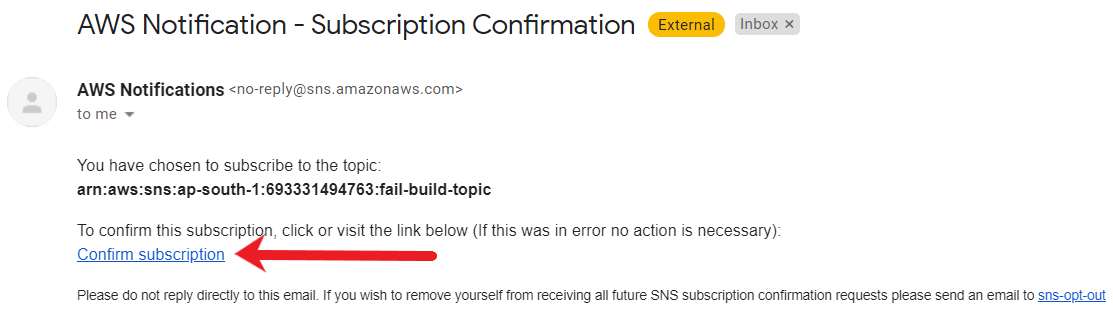
1. After the protocol and endpoint have been added, scroll to the bottom of the page and click on **Create subscription**.



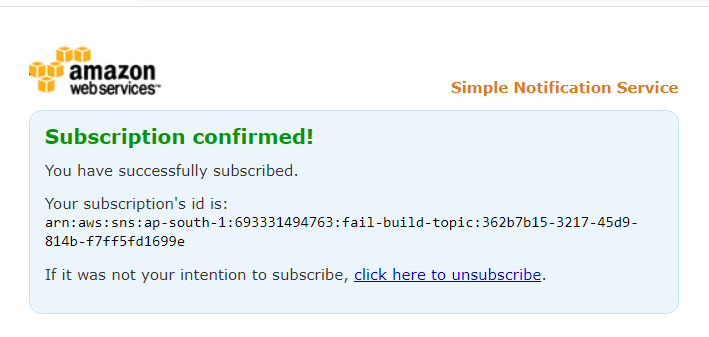
1. After a subscription is created, you will get a success message. A verification email to confirm the subscription will also be sent to your email address.



1. Login to your email and find the email from AWS Notifications. Click on "Confirm subscription" hyperlink.



1. Once the subscription is confirmed, you will get a success message.



Ref sns-access-policy.json

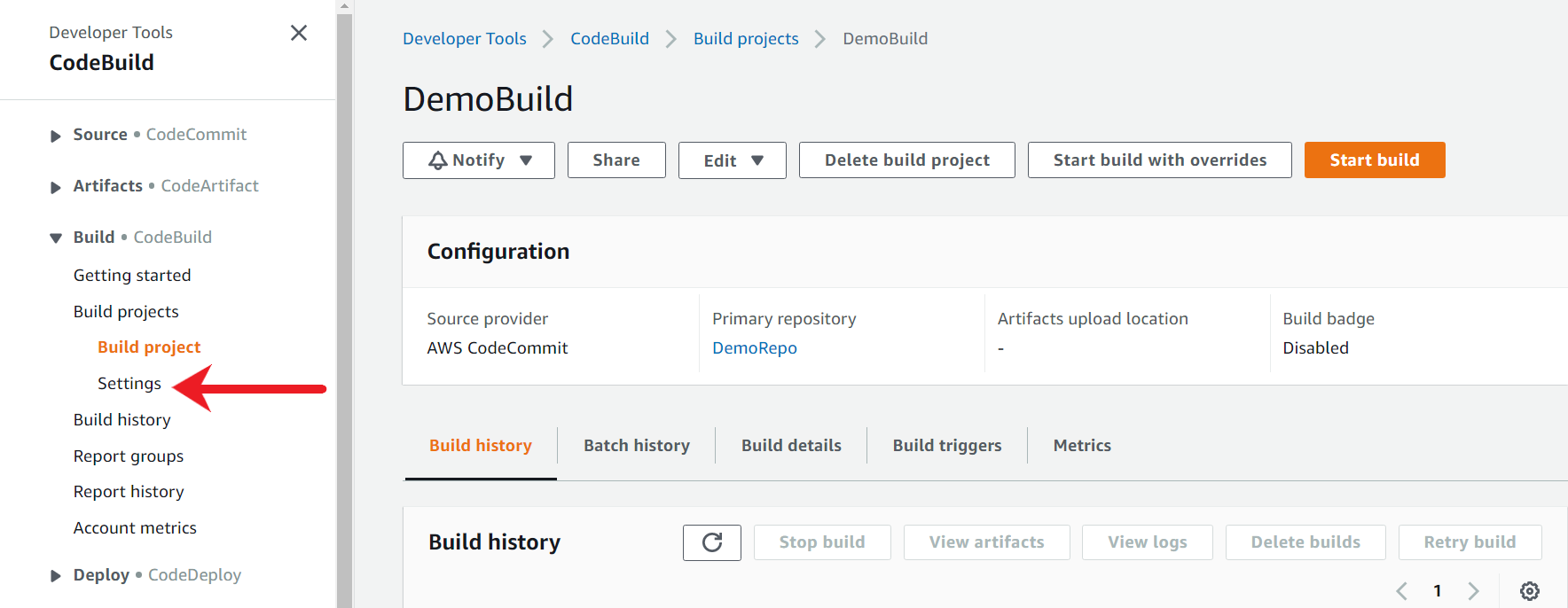
<https://docs.aws.amazon.com/sns/latest/dg/sns-access-policy-language-architectural-overview.html>

<https://docs.aws.amazon.com/dtconsole/latest/userguide/set-up-sns.html>

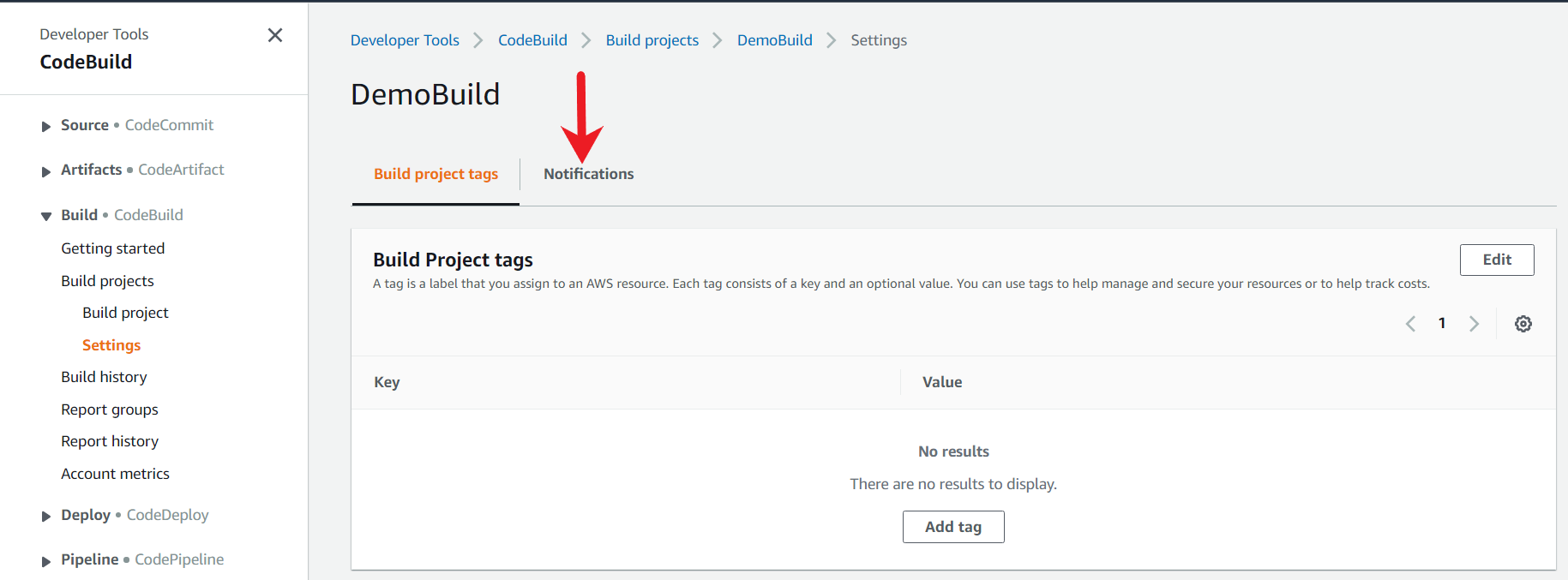
Enable Codebuil Notification

For a fail build notification to be sent to the email, you need to integrate the CodeBuild with SNS. Create a CodeBuild Notification with the necessary integration.

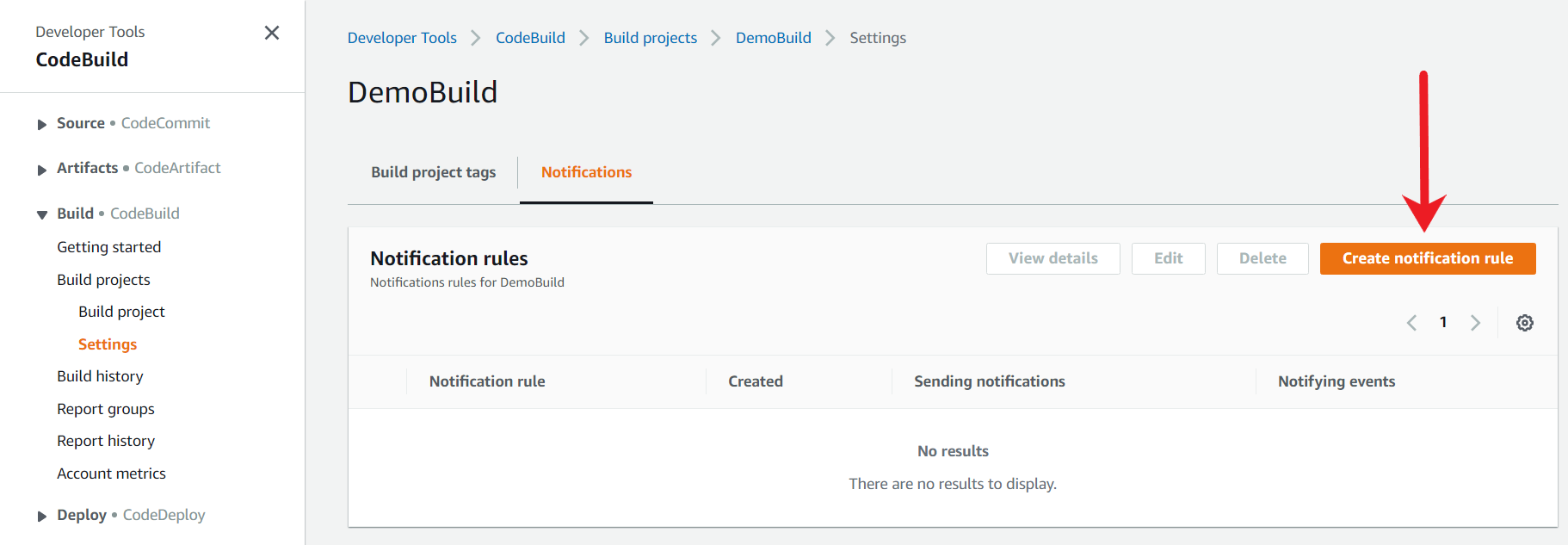
1. Under your CodeBuild project, click on **Settings** from the left-hand tab.



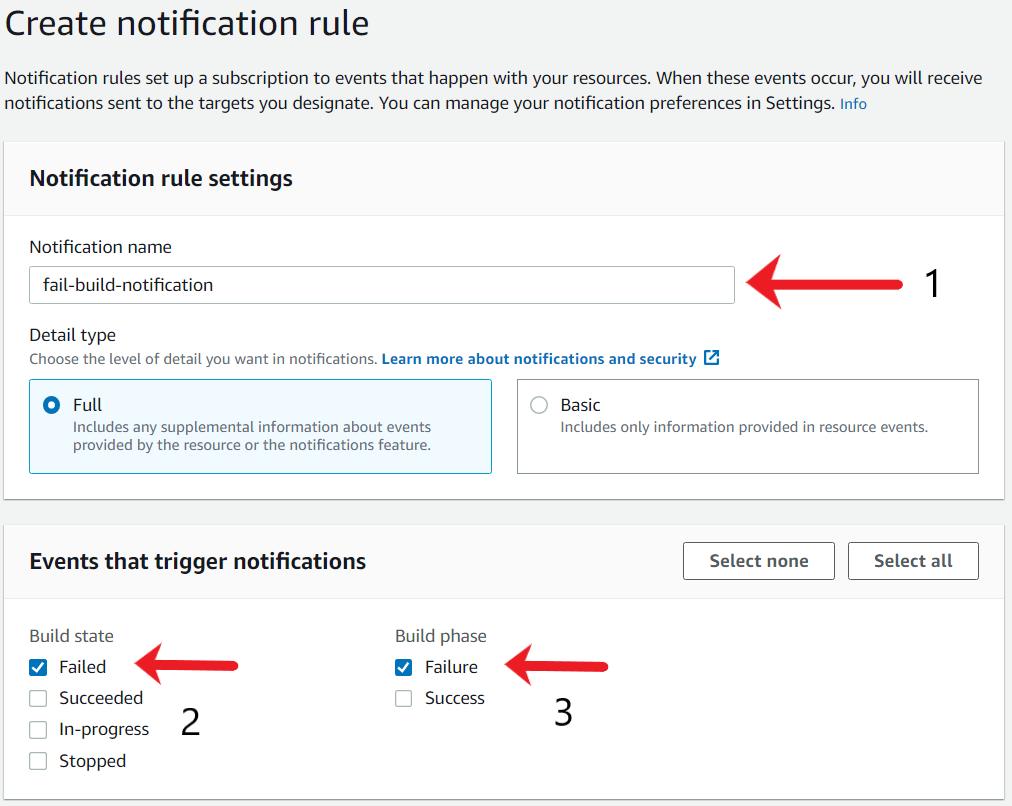
1. Under settings, click on the **Notifications** tab.



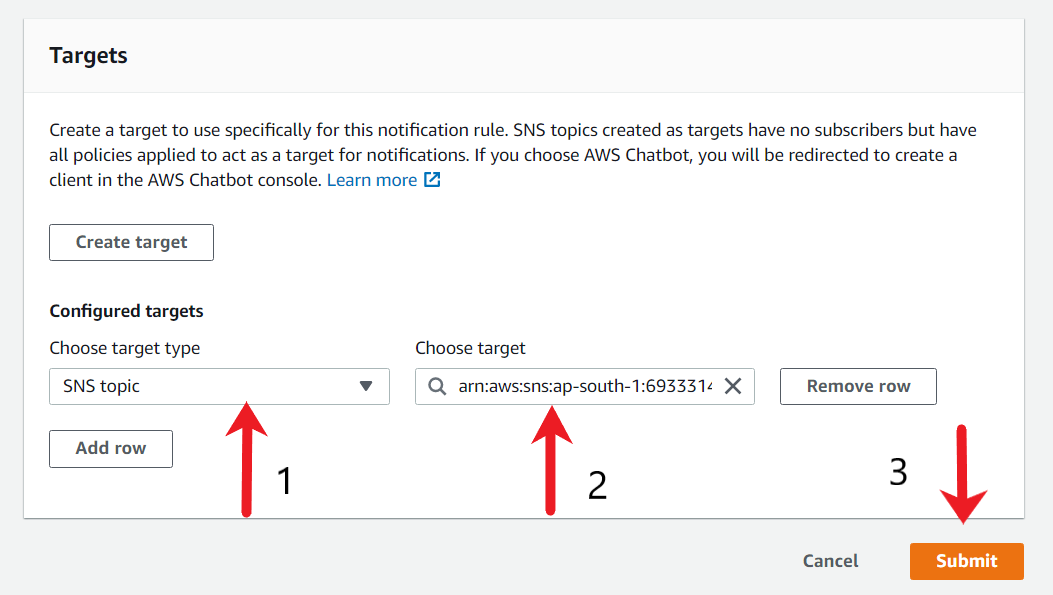
1. Under the Notifications tab, choose **Create notification rule**.



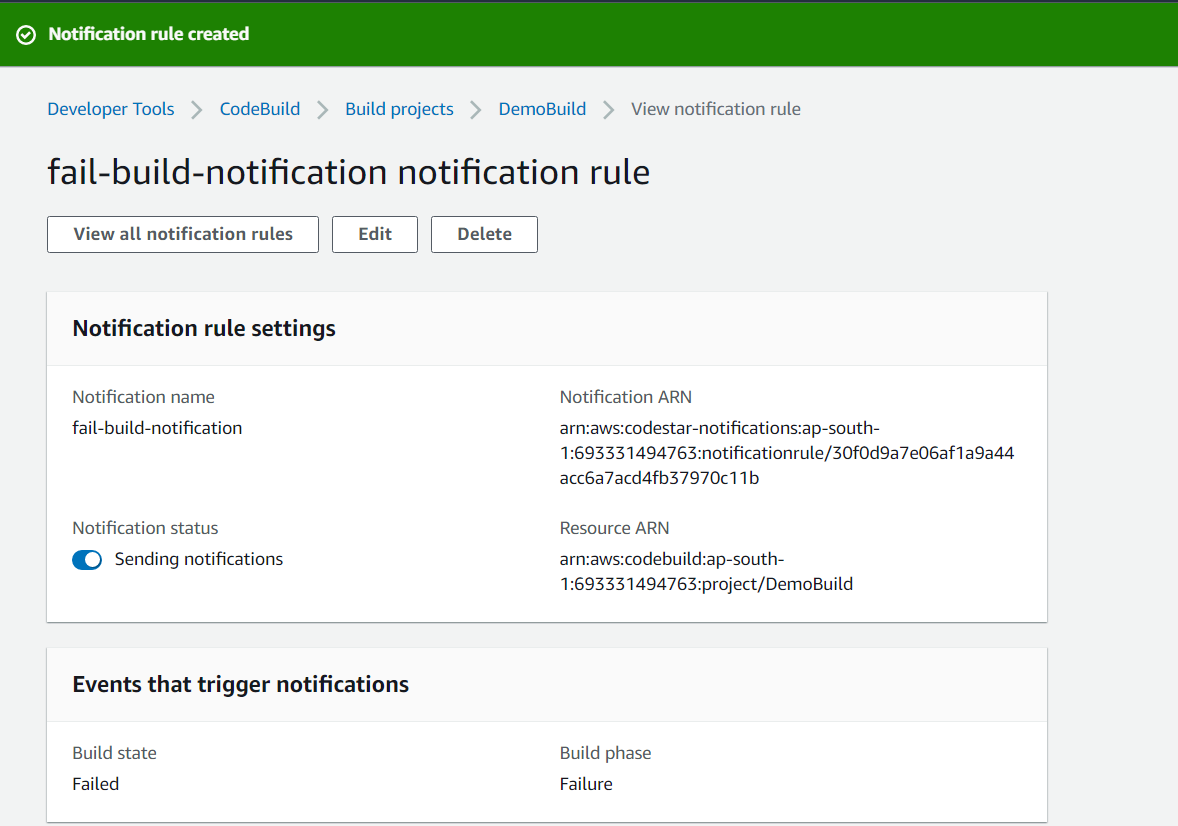
1. In Notification name, enter a name for the rule. For this example, we have added it as fail-build-notification. In Detail type, choose Full. In Events that trigger notifications, select the events for which you want to send notifications, for our case, we need to select "Failed" under Build state and "Failure" under Build phase.



1. In Targets, under Configured targets, select SNS topic and select the SNS topic under the Choose Target. Once done click on Submit.



1. After the notification rule is created, you will get a success message and all the details related to the rule. Ensure notification status is "Sending notifications".



1. Scroll the bottom of the notification rule page and ensure that the "Notification target status" is set to Active. If it is "unreachable" then there is some issue related to the SNS Access policy.

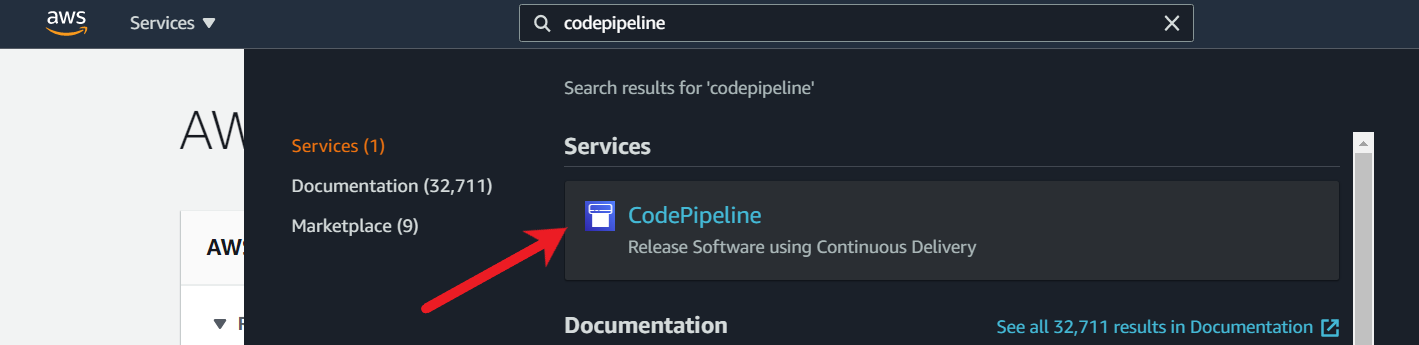


Create CodePipeline

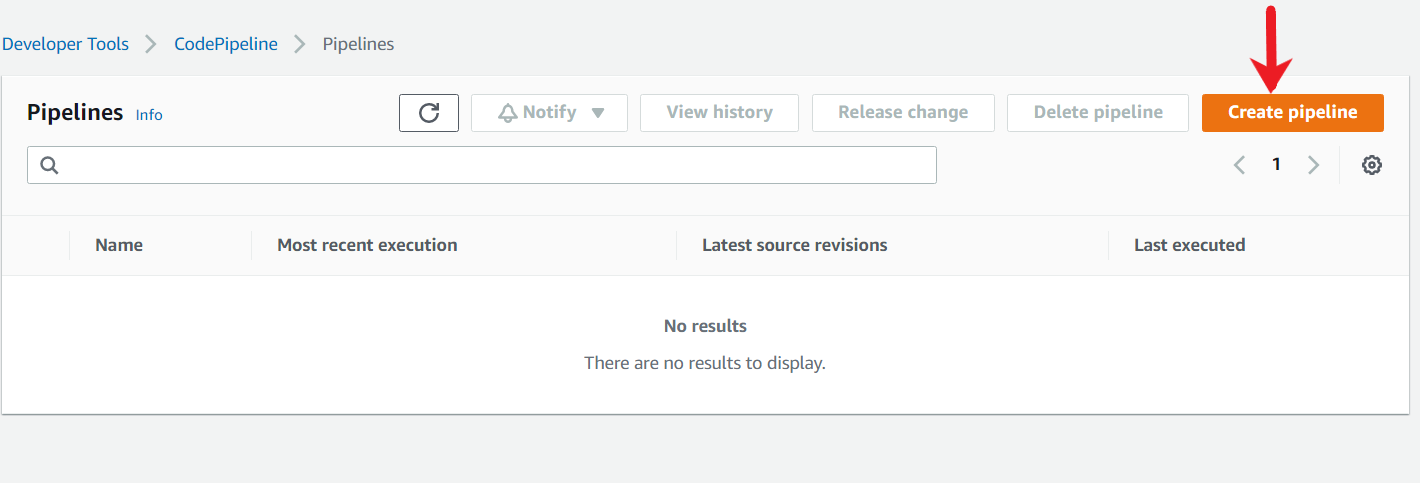
You have to use the AWS CodePipeline service to create a continuous integration pipeline with an appropriate source and build stages.

The pipeline will detect changes in the code stored in your CodeCommit repository and then build the application code.

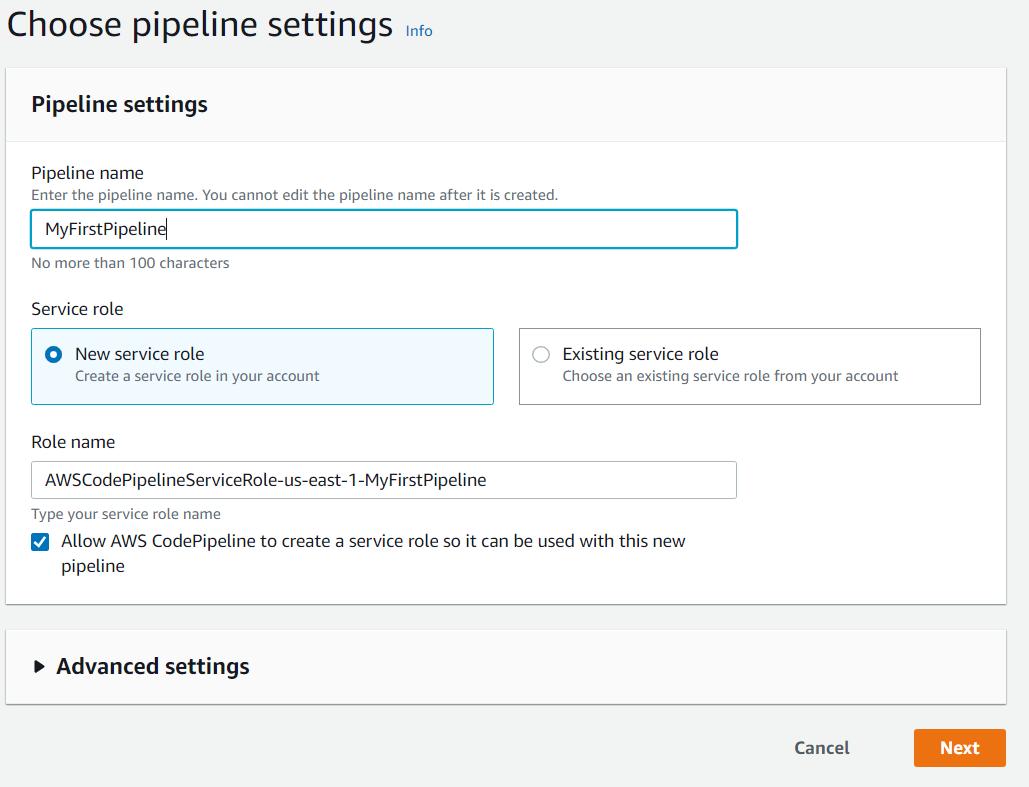
1. To create a CodePipeline pipeline, search for the CodePipeline service and click on the available option.



1. Click on the Create pipeline button.



1. Under the pipeline settings, in the Pipeline name, enter MyFirstPipeline. In the Service role, choose New service role to allow CodePipeline to create a service role in IAM.

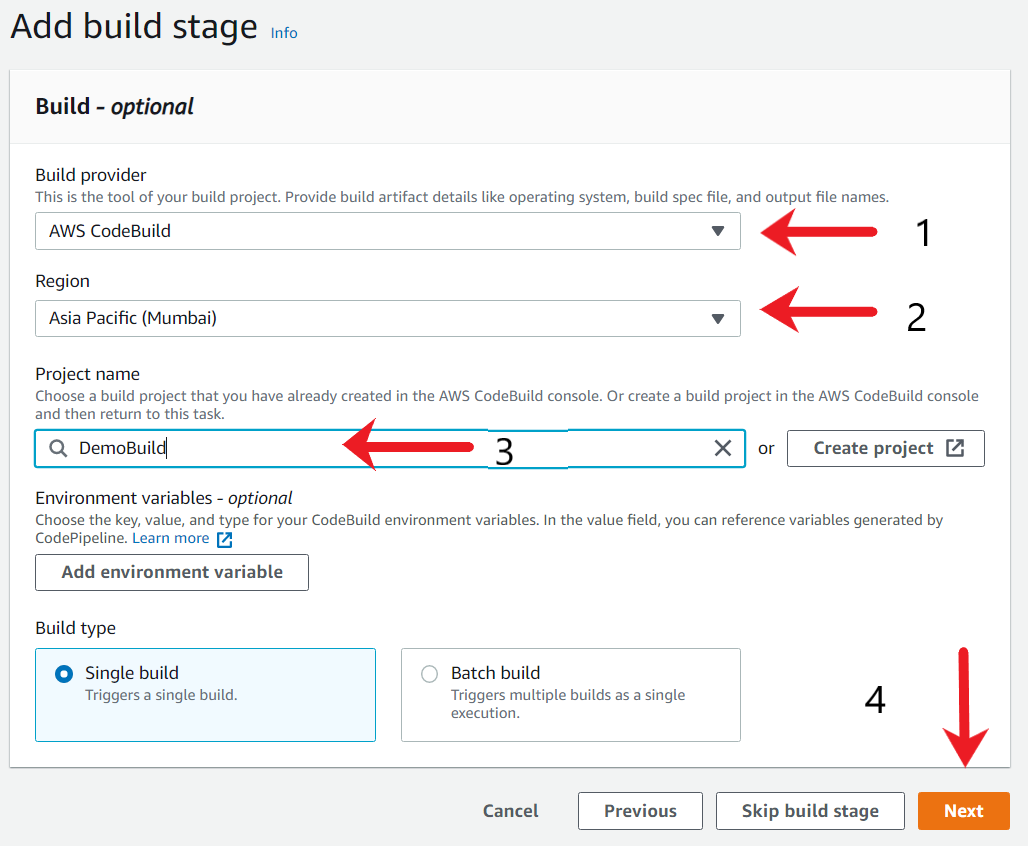


1. Leave the settings under Advanced settings at their defaults, and then choose Next.
2. Add source stage, in Source provider, choose AWS CodeCommit. In the Repository name, choose the name of the CodeCommit repository you created (DemoRepo, for this example). In Branch name, choose main.

After you select the repository name and branch, a message displays the Amazon CloudWatch Events rule to be created for this pipeline. Under Change detection options, leave the defaults. This allows CodePipeline to use Amazon CloudWatch Events to detect changes in your source repository. Choose Next.



1. In Step of **Add build stage**, choose Build Provider as AWS CodeBuild. Ensure correct region is selected. Under the Project name, select the CodeBuild project name, in our case it is DemoBuild. Once done, click on Next.



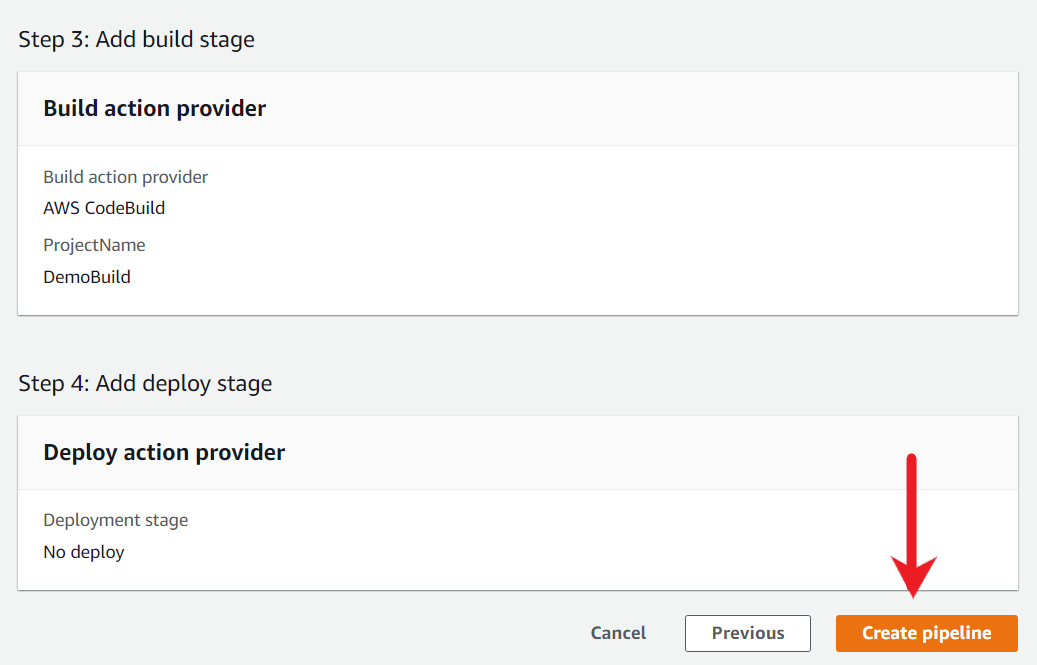
1. In the Step of **Add deploy stage,**select "Skip deploy stage" option.



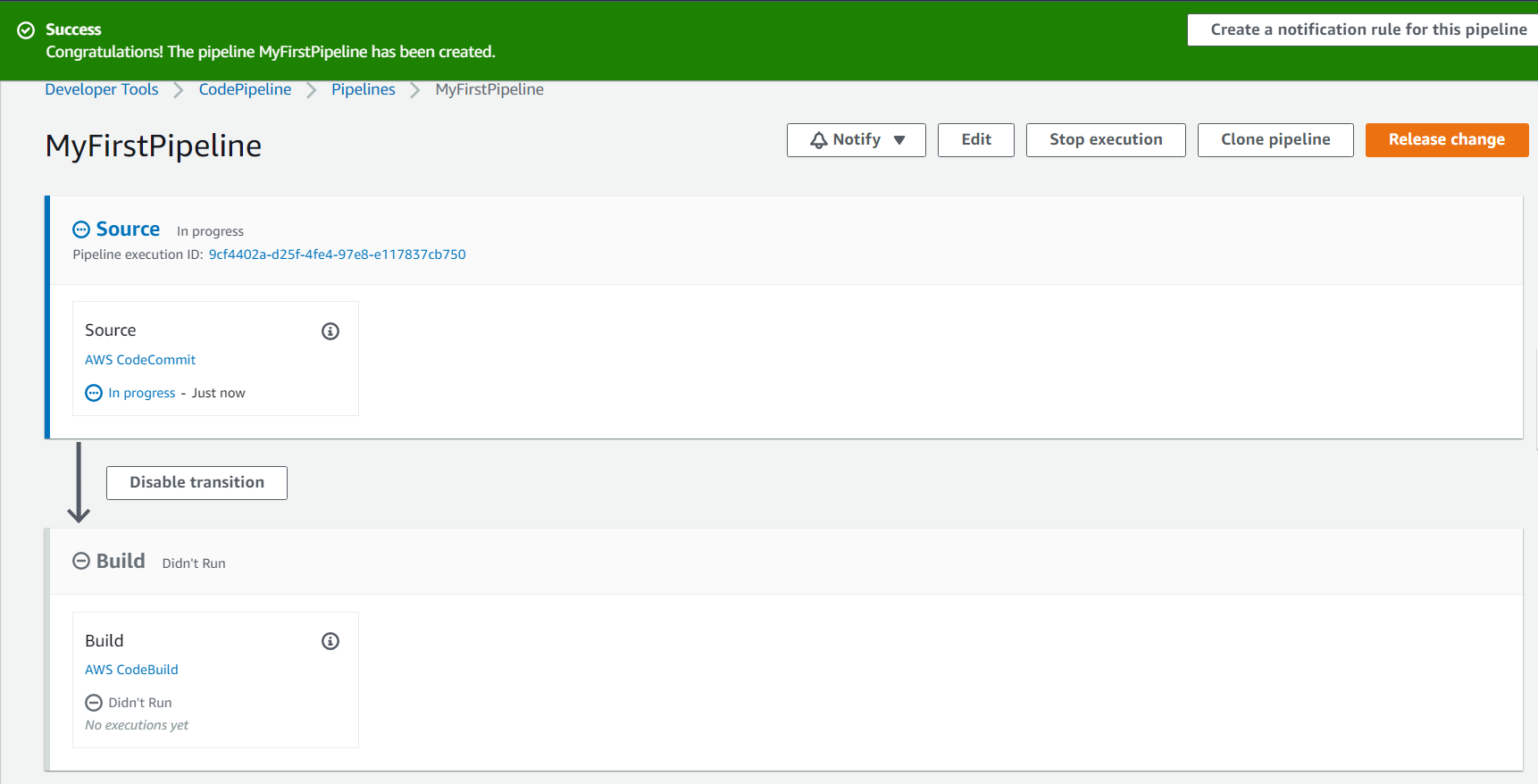
A new popup will appear for confirmation. Select Skip.



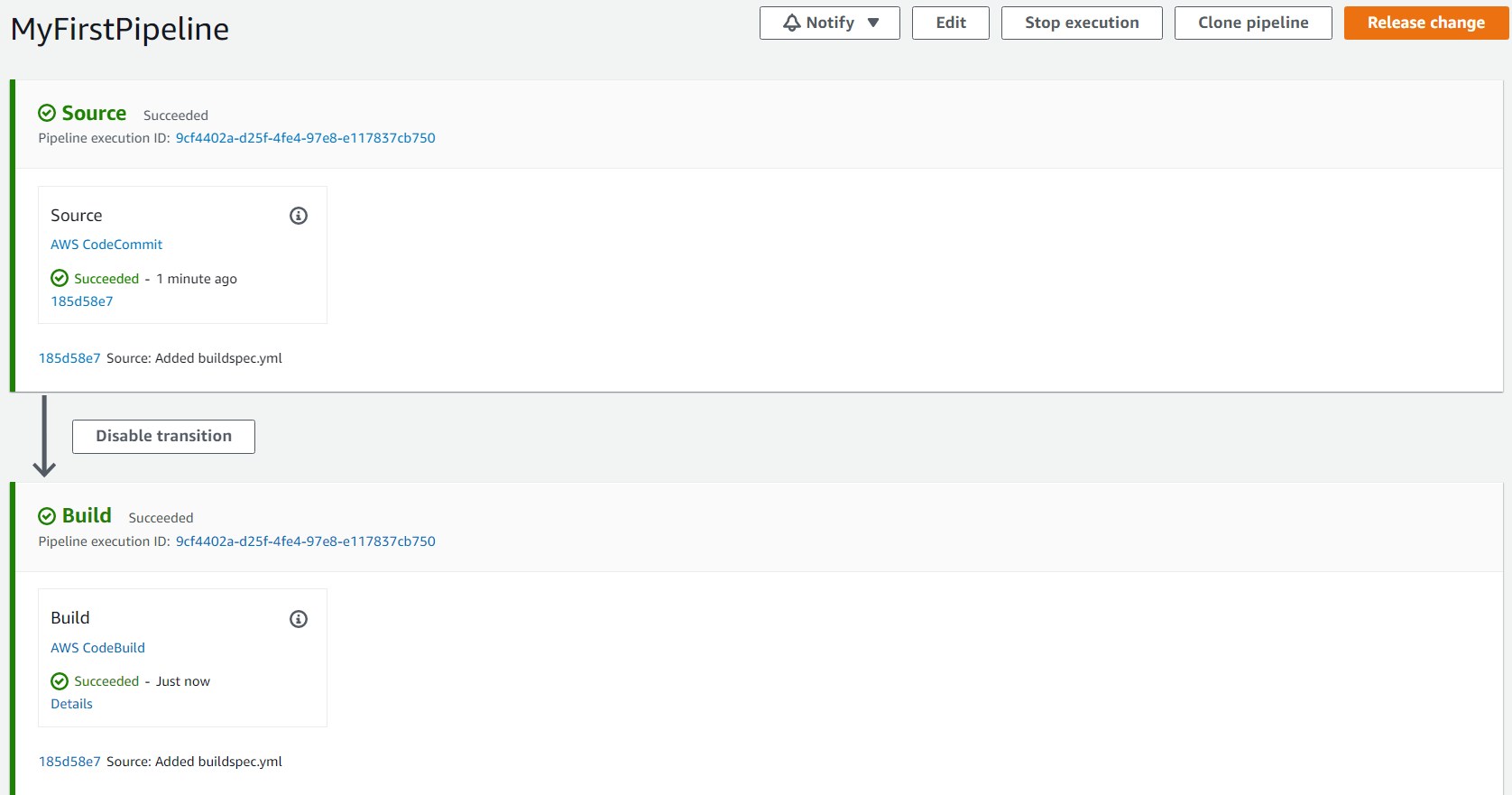
1. In **Step: Review**, review the information and then choose Create pipeline.



1. Once the pipeline is created, you will see a success message. The pipeline automatically starts and runs the sample through the pipeline. You can view progress and success and failure messages as the pipeline builds.



1. After the pipeline runs successfully, you will see a "Succeeded" message across both the Source and Build stage.

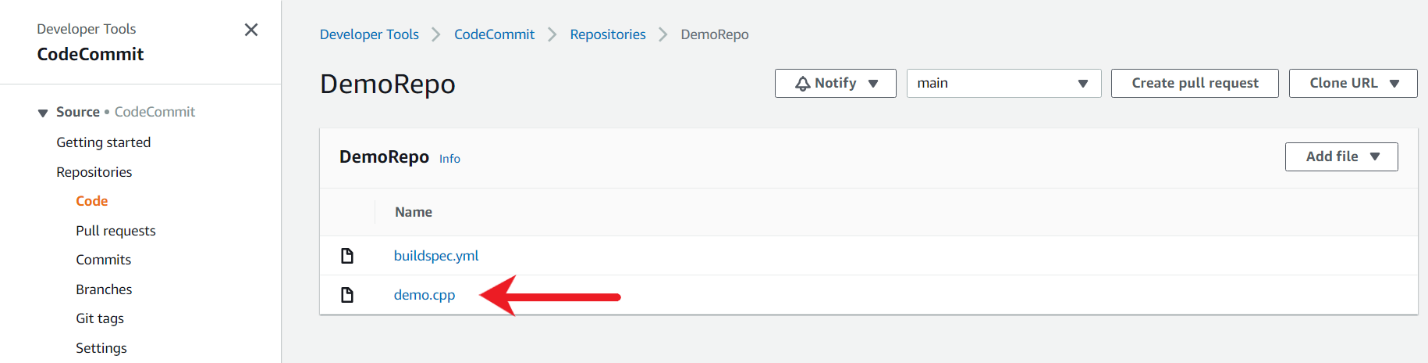


<https://docs.aws.amazon.com/codepipeline/latest/userguide/getting-started-codepipeline.html>

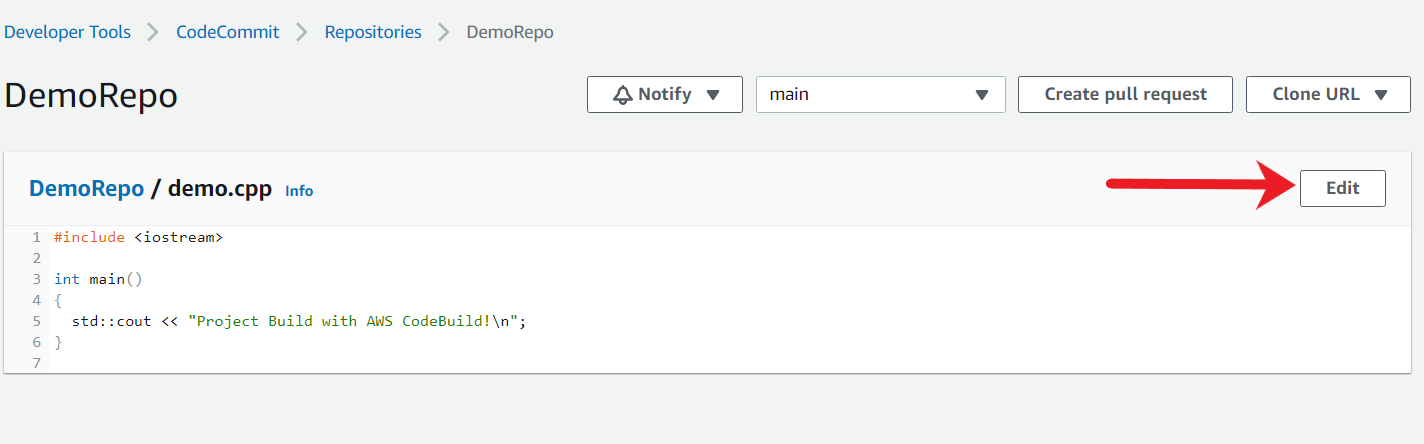
Verify Fail Notification

After the CodePipeline ran successfully, you now need to verify if failure in the build process leads to email notification. To test this, modify the demo.cpp code and remove a few lines so that the build process will fail. Once the file is modified, monitor the pipeline to verify the failed build and verify if the email notification has been sent over email.

1. To verify the notification functionality on fail builds, go back to the CodeCommit repository and click on the demo.cpp file.



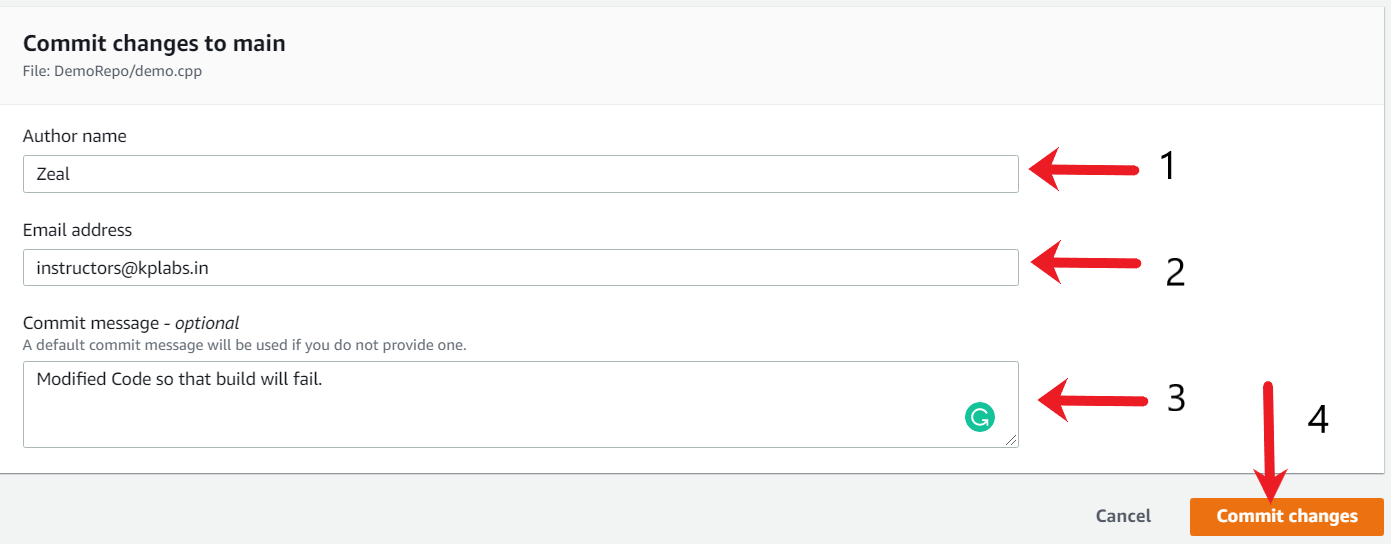
1. Once the contents of the file are displayed, click on **Edit**.



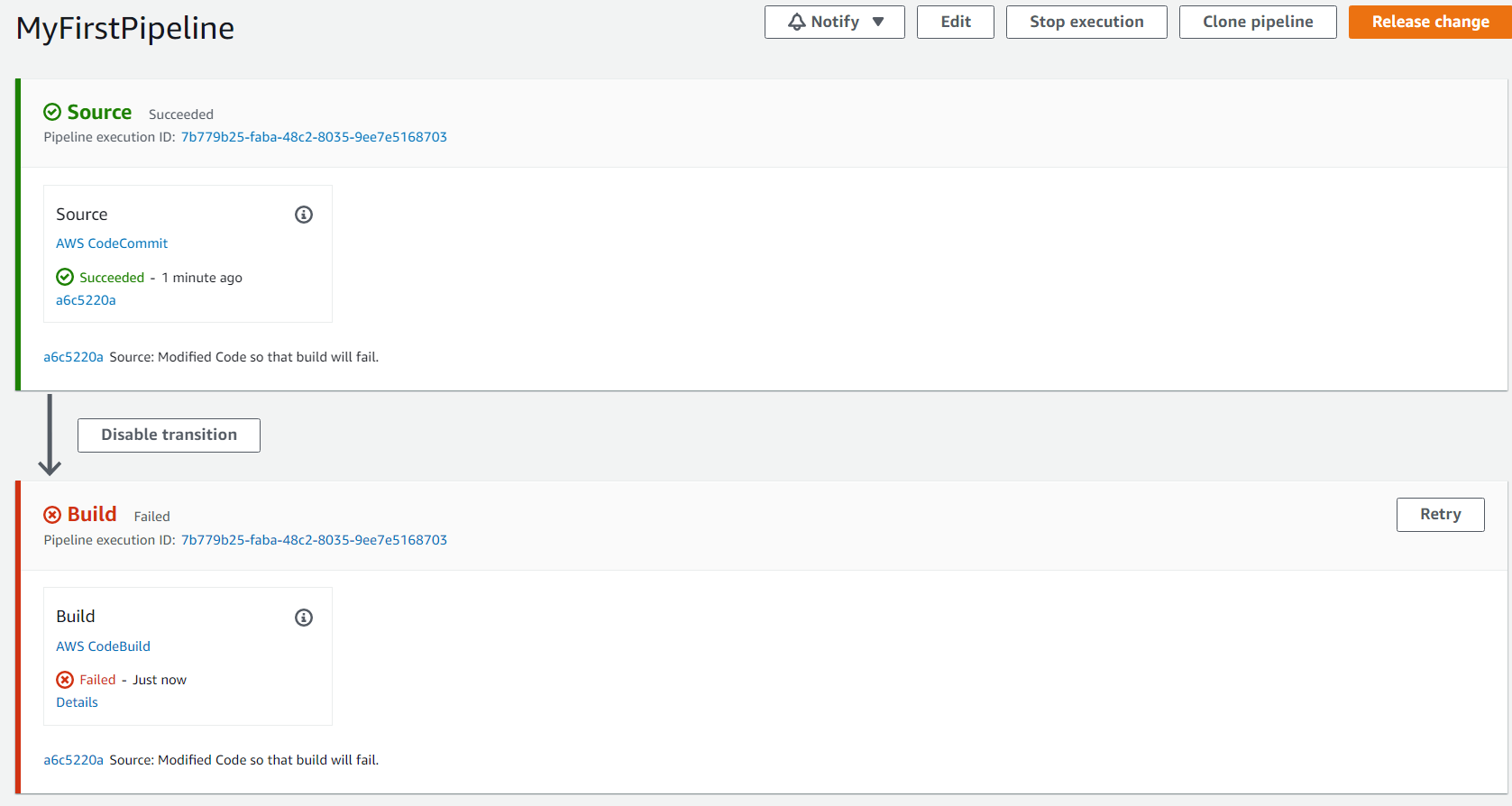
1. Let us remove the last line containing **}** within the code. This will lead to build failures.



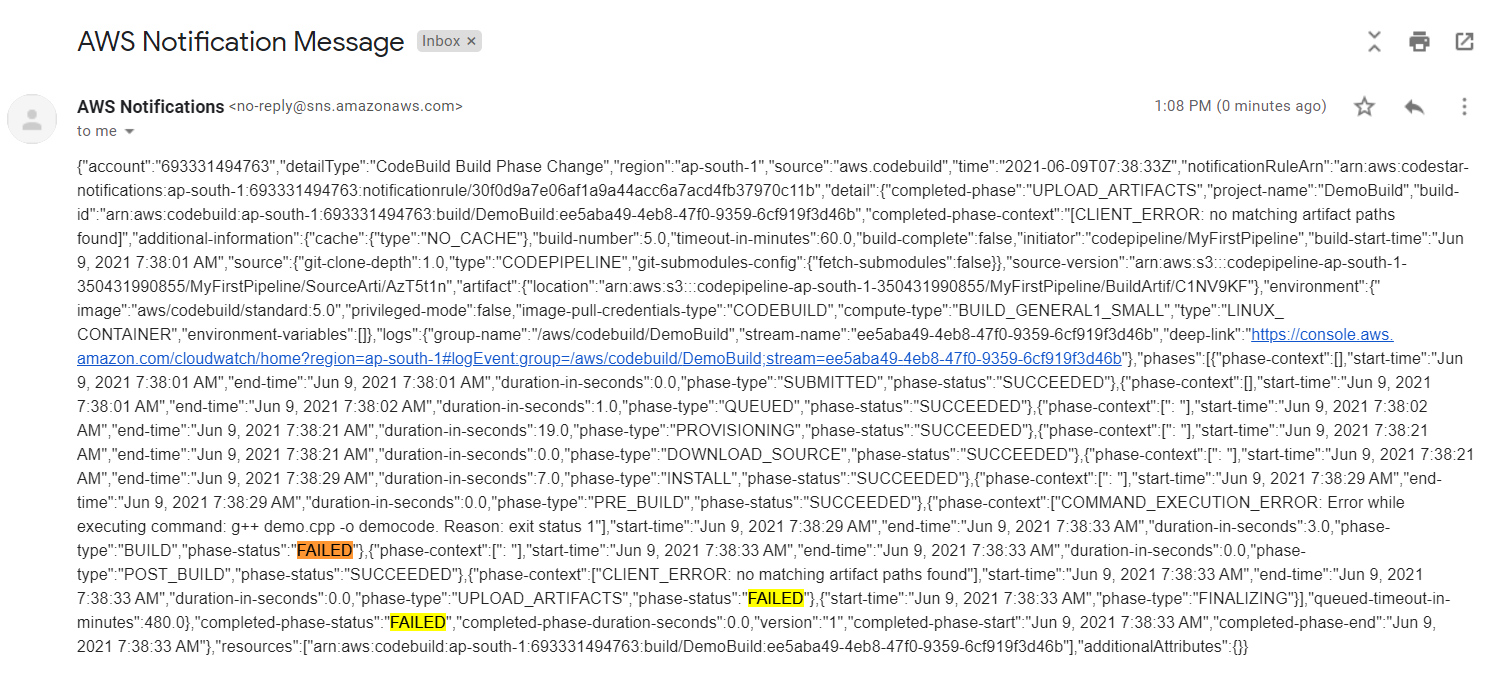
1. Once the application code is modified, scroll to the bottom of the page and fill in the details and the Commit message.



1. Once the commit is successful, go to the AWS CodePipeline console, and within few minutes, you should see a "Failure" message under the Build stage.



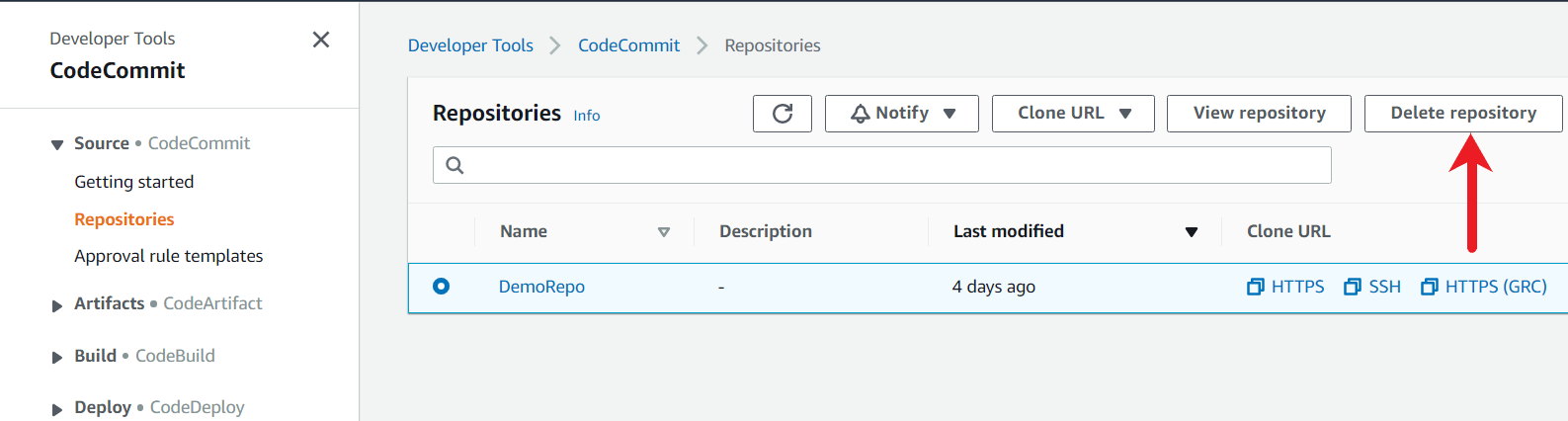
1. Check your email and verify you have received a new notification from SNS which contains details associated with the Failed deployment.



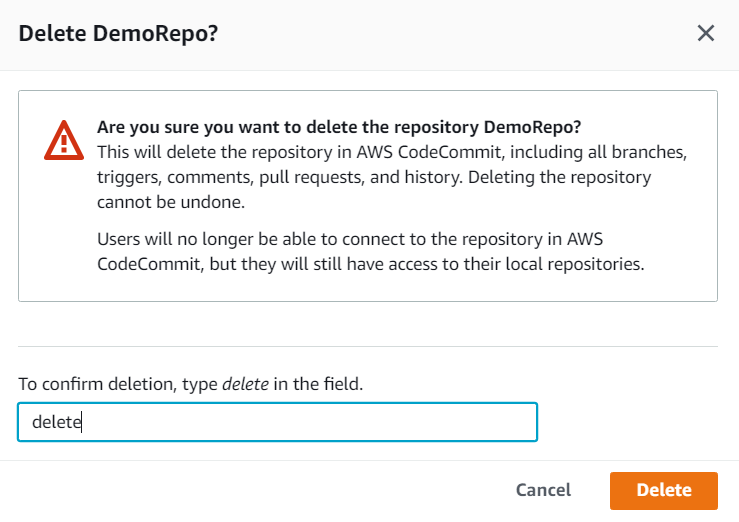
Cleanup

As we have successfully demonstrated a proof of concept for a continuous delivery pipeline, we'll clean up our AWS resources at this time. Delete all the resources created for your proof of concept.

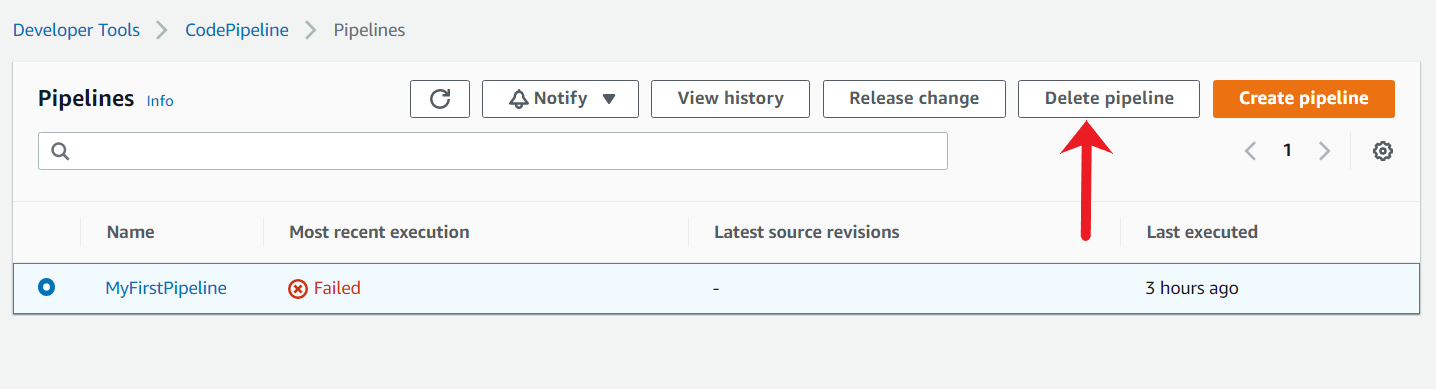
1. To delete the CodeCommit repository, go to the CodeCommit console and select your repository. Click on the Delete Repository option.



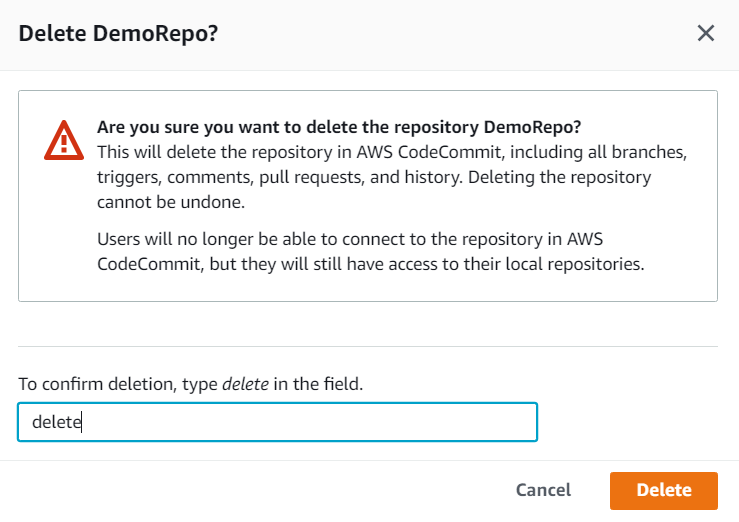
In the popup window, enter delete, and then choose Delete. The repository is permanently deleted.



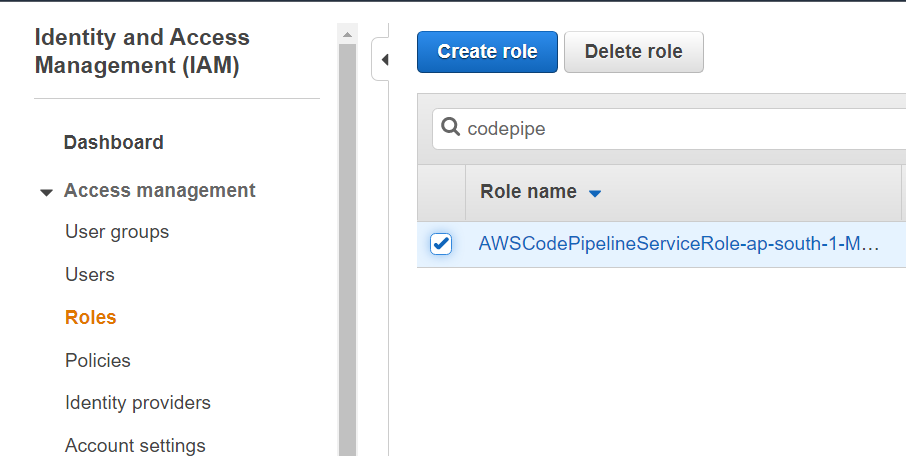
1. To delete CodePipeline, go to the AWS CodePipeline console and select the Pipeline sub-tab. Select your pipeline and click on Delete.



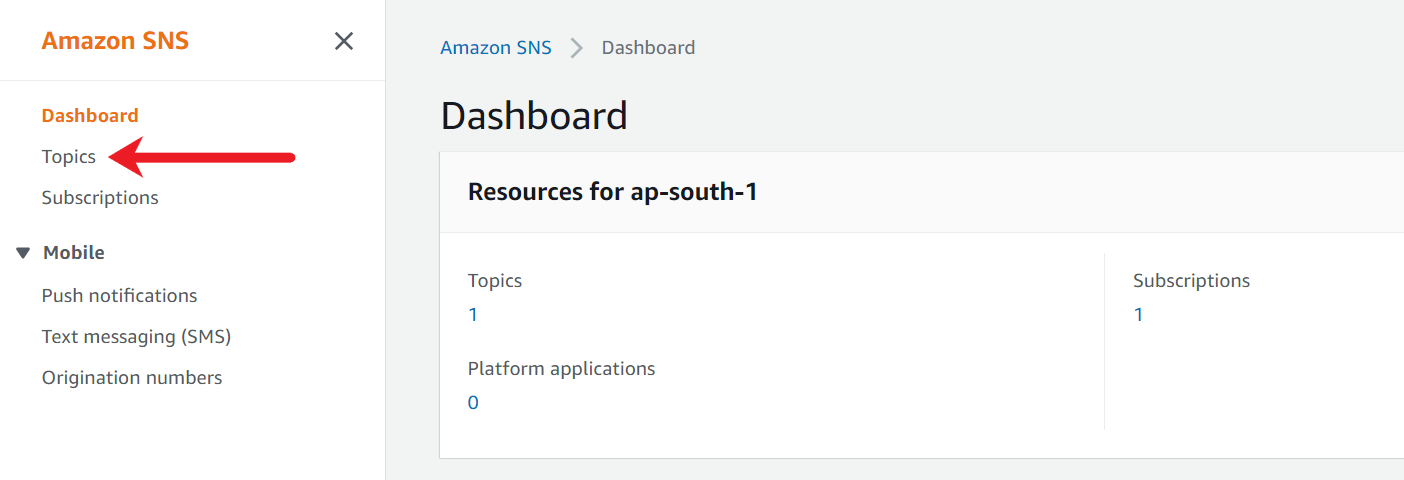
On the confirmation window, type delete and press and click on the Delete button.



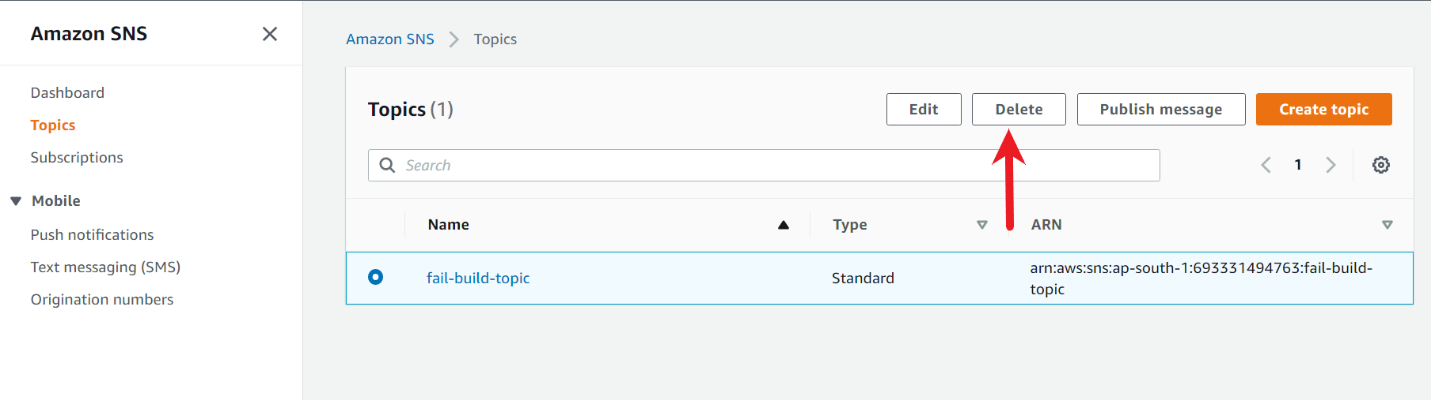
1. To delete the IAM role, go to the IAM console. In the navigation pane, choose Roles,In the navigation pane, choose Roles, and then select the check box next to the role name that you want to delete. The following role needs to be deleted.
   * Role for CodePipeline (name starts with AWSCodePipelineServiceRole\*)



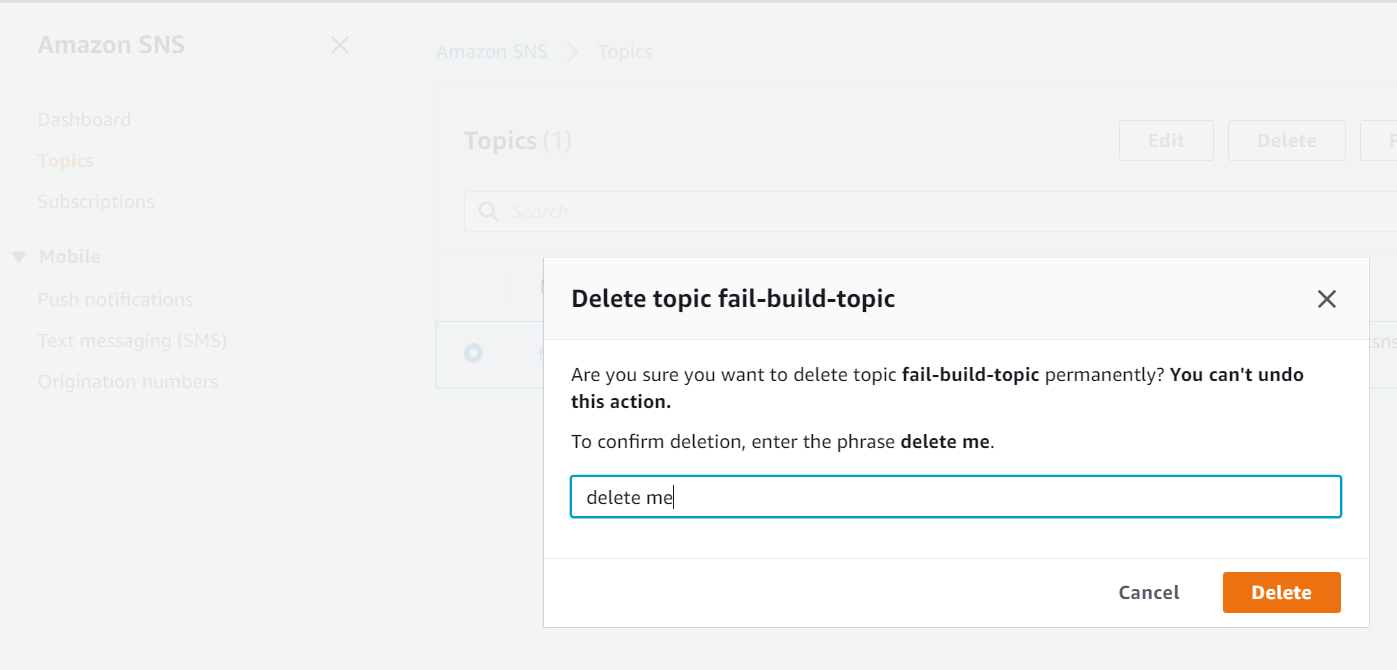
1. To Delete SNS topic, In the left navigation pane, choose Topics.



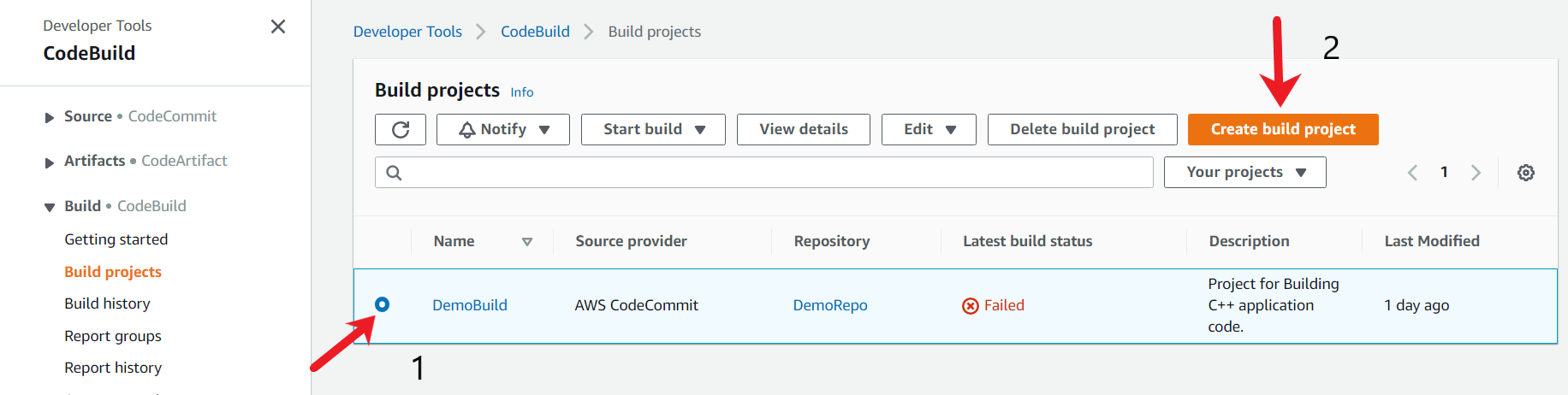
1. On the Topics page, select a topic, and then choose Delete.

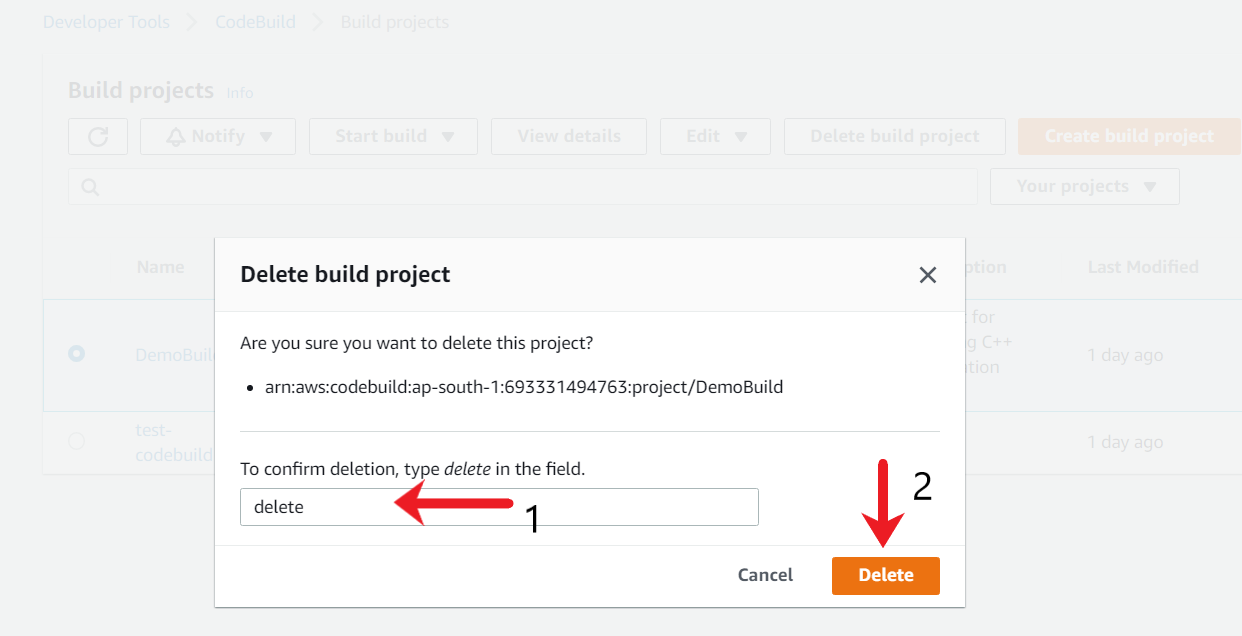


In the Delete topic dialog box, enter delete me, and then choose Delete.

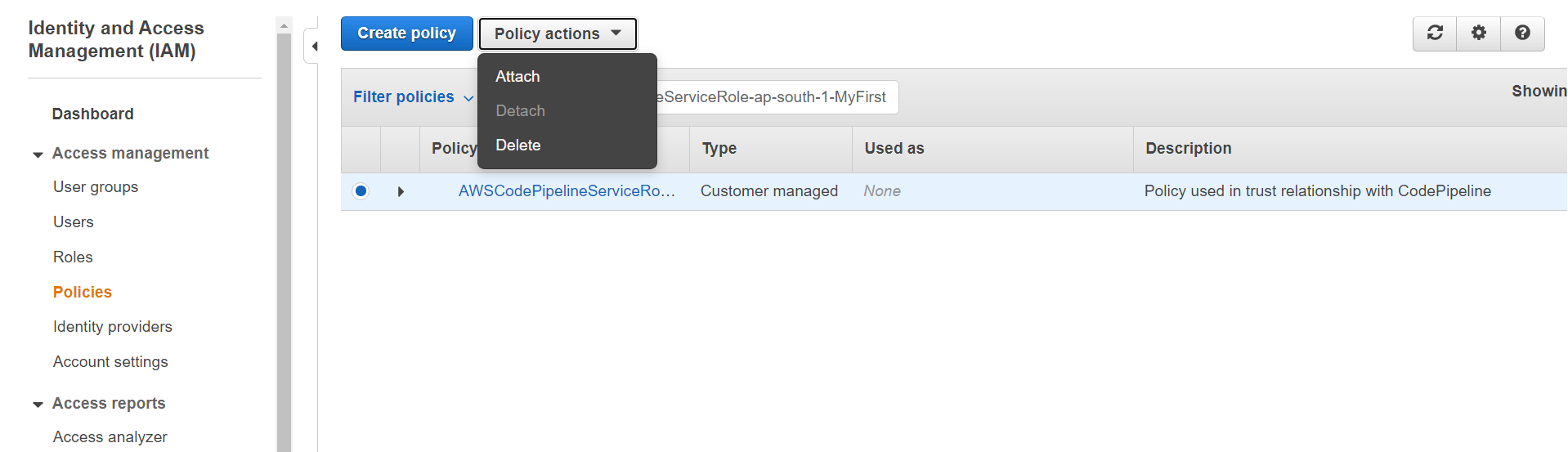


1. To delete the CodeBuild environment, go to the AWS CodeBuild console. Select the Build project you want to delete, and then choose Delete.





1. Ensure also to delete the IAM policy created by CodePipeline Service. The Policy name for this case is AWSCodePipelineServiceRole-ap-south-1-MyFirstPipeline. If your pipeline name and selected region are different, the IAM policy name will need to be computed accordingly. To Delete IAM Policy, select the policy, click on Policy Actions and select Delete.



A new confirmation box will open. Click on Delete to Confirm.

