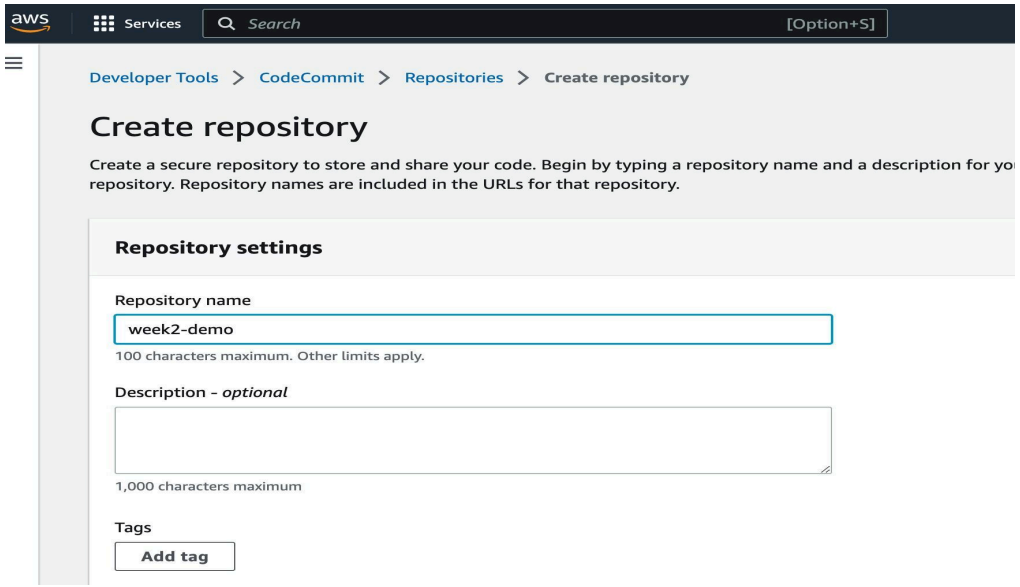
JOIN THE DevOps/SRE BATCH 8.0 - https://bit.ly/devopsbatch8
Check the DevOps Placements - https://www.youtube.com/watch?v=zm5jAUbDXL0

AWS DEVOPS CODEPIPELINE

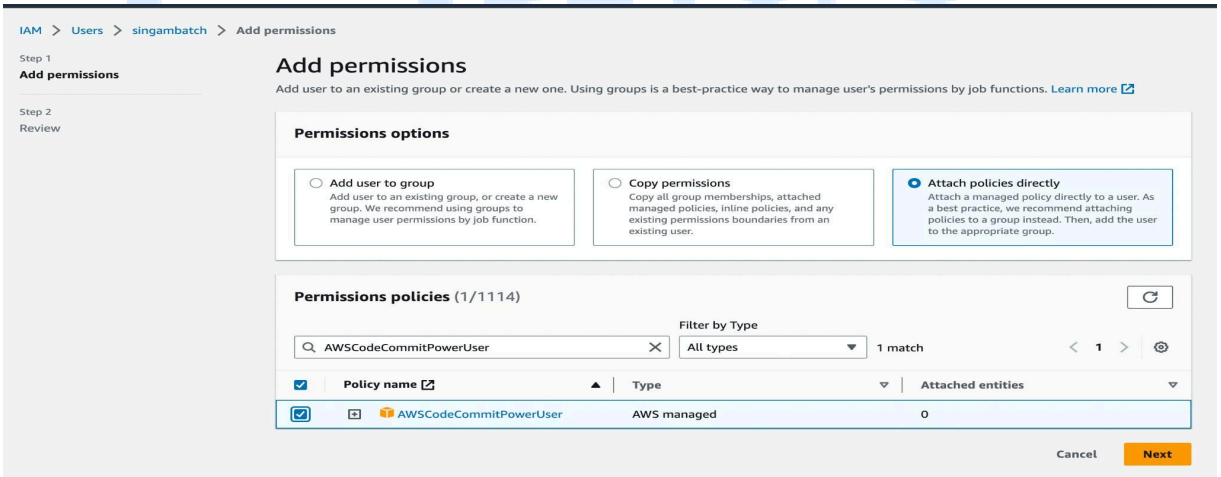
Stage1

Step 1 - Use the CodeCommit console to create the CodeCommit repository



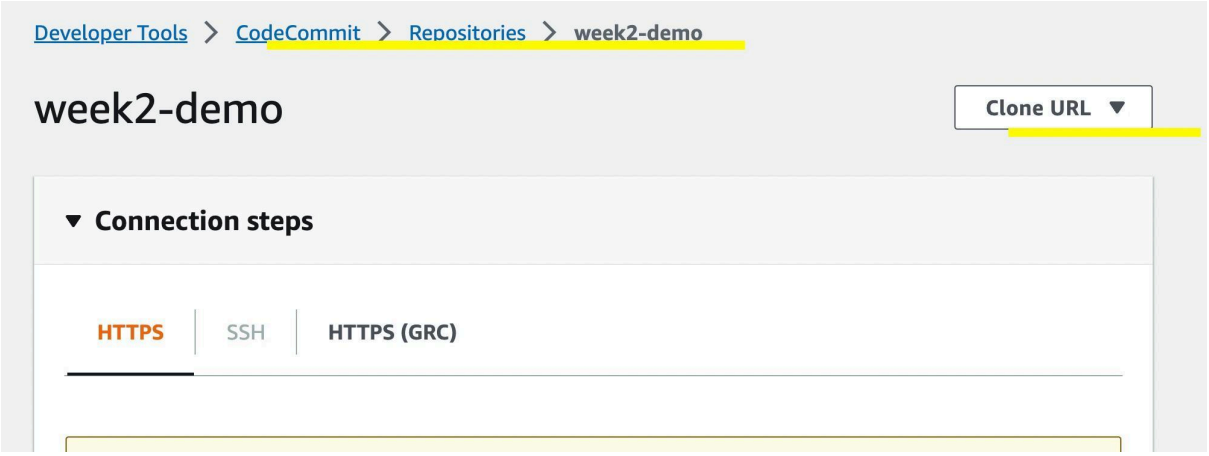
Step 2- Create the IAM user role for codecommit and add the policy as permission: [singambatch]

AWSCodeCommitPowerUser - Add as permission to user

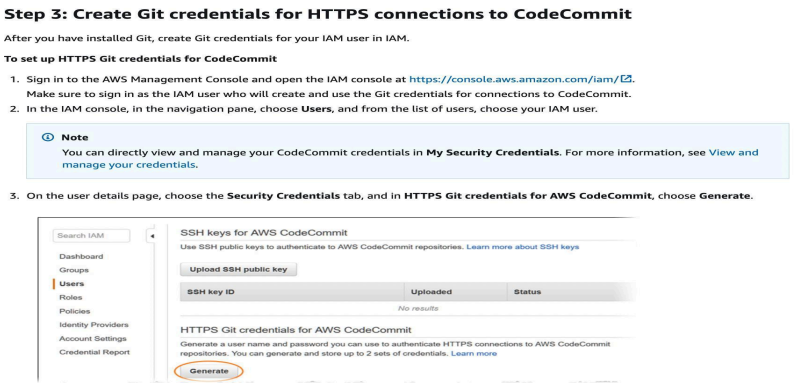


Step3: Clone your aws codecommit repo in your local, Make sure you have the aws configure command already set up to access aws cli.

Clone repo -



Step 3.1:



Step 3: Create Git credentials for HTTPS connections to CodeCommit

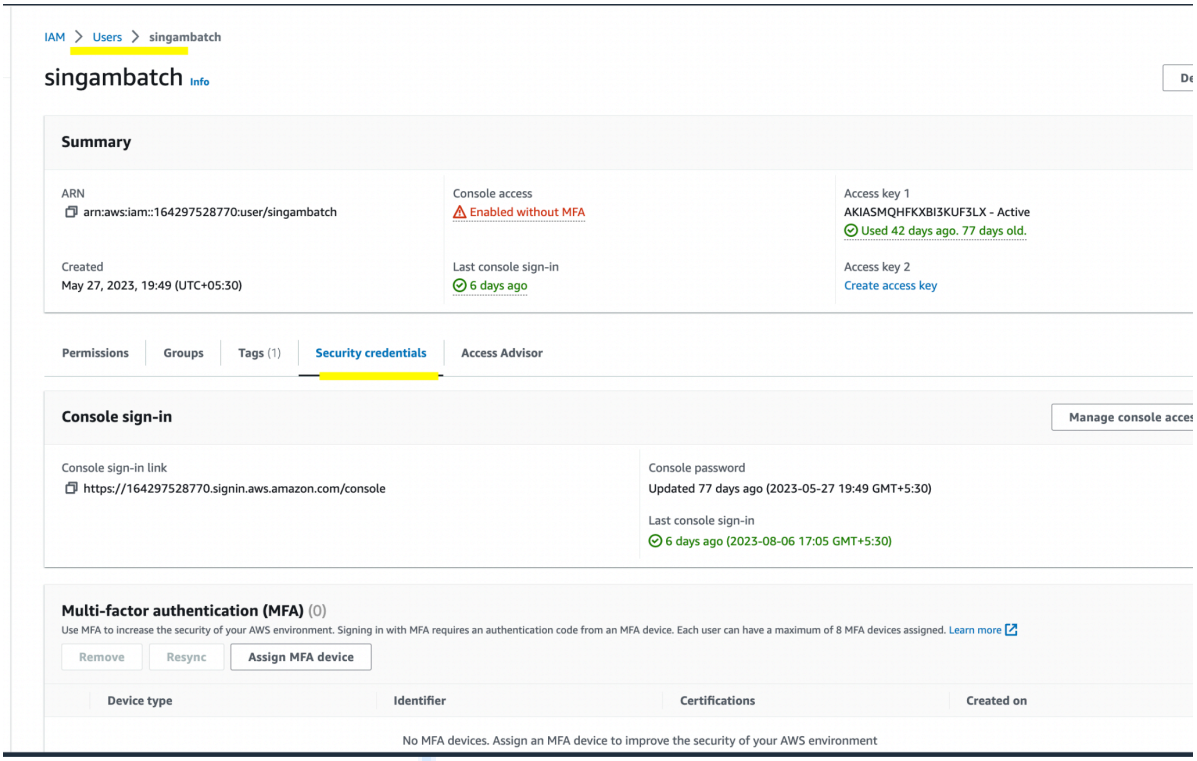
After you have installed Git, create Git credentials for your IAM user in IAM.

To set up HTTPS Git credentials for CodeCommit

1. 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.
2. In the IAM console, in the navigation pane, choose **Users**, and from the list of users, choose your IAM user.

Note
You can directly view and manage your CodeCommit credentials in **My Security Credentials**. For more information, see View and manage your credentials.

3. On the user details page, choose the **Security Credentials** tab, and in **HTTPS Git credentials for AWS CodeCommit**, choose **Generate**.



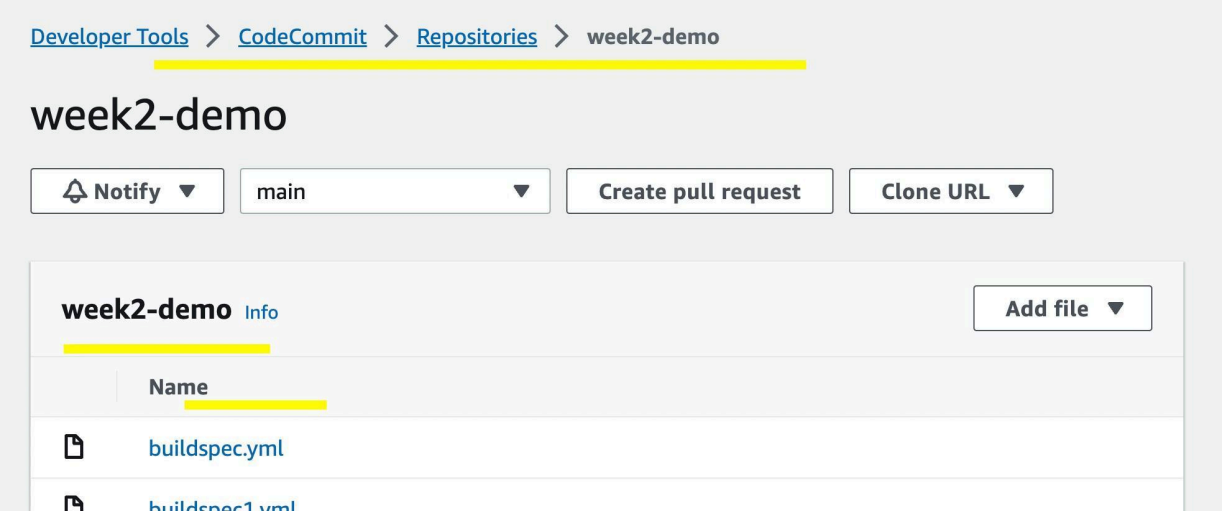
Step 4 -
git clone
https://git-codecommit.us-west-1.amazonaws.com/v1/repos/we e k2-demo

git clone --mirror https://github.com/praveen1994dec/aws_pipeline.git github-repo

cd github-repo

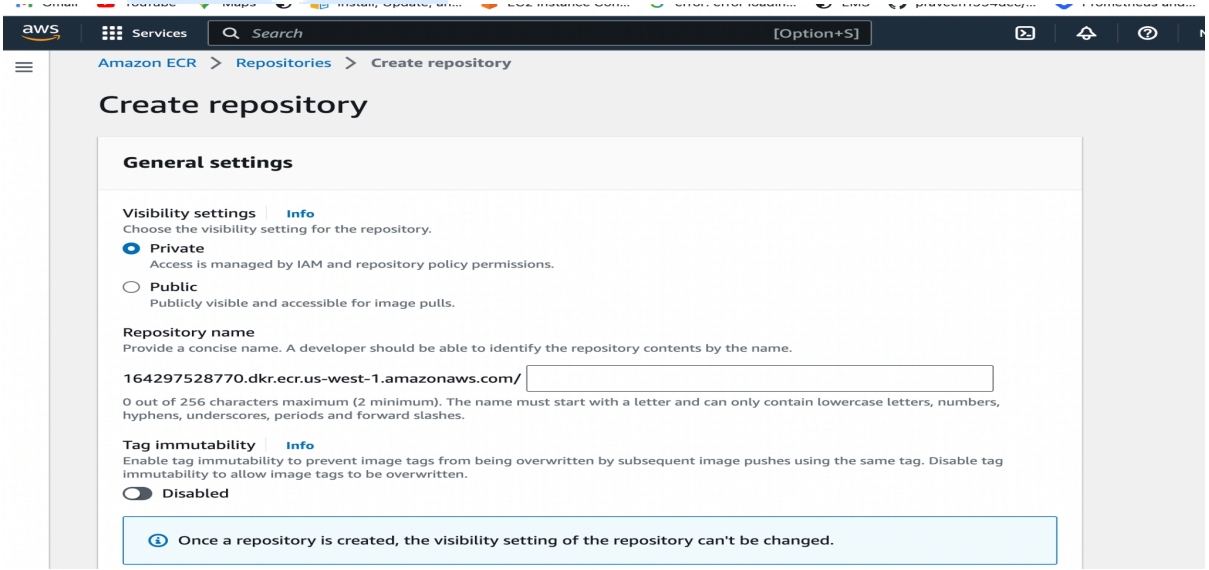
git push **https://git-codecommit.us-west-1.amazonaws.com/v1/repos/we e k2-demo** --all

Step 5- Go to code commit and check the repo details



Stage2: Configure CodeBuild

Create Repository in Elastic Container Registry

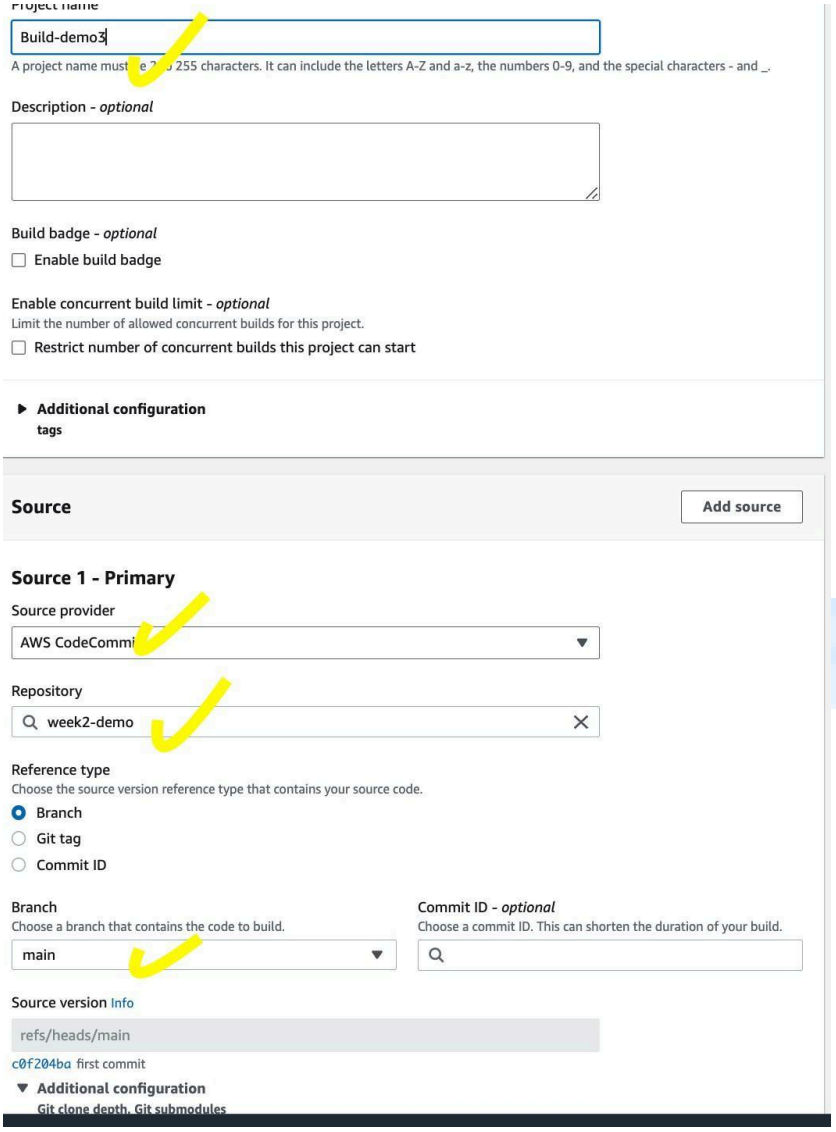


164297528770.dkr.ecr.us-west-1.amazonaws.com/singambatch

2.1 -> #Setup a CodeBuild project

Click on create code build project -> select repository -> select branch -> and add the below data

Select the PRIVILEGED checkbox



New environment image

Managed image
Use an image managed by AWS CodeBuild

Custom image
Specify a Docker image

Operating system

Amazon Linux 2

Runtime(s)

Standard

Image

aws/codebuild/amazonlinux2-x86_64-standard:4.0

Image version

aws/codebuild/amazonlinux2-x86_64-standard:4.0-23.02.16

Privileged

Enable this flag if you want to build Docker images or want your builds to get elevated privileges.

Service role

Choose an existing service role from your account

arn:aws:iam::164297528770:role/service-role/codebuild-service-rolec

FILL THE ENV VARIABLES:

7 GB memory, 4 vCPUs

15 GB memory, 8 vCPUs

145 GB memory, 72 vCPUs

Environment variables

Name	Value	Type	
AWS_DEFAULT_REGION	us-west-1	Plaintext	Remove
AWS_ACCOUNT_ID	164297528770	Plaintext	Remove
IMAGE_TAG	latest	Plaintext	Remove
IMAGE_REPO_NAME	demo3	Plaintext	Remove

Add environment variable

Create parameter

Logs

CloudWatch

CloudWatch logs - *optional*
Checking this option will upload build output logs to CloudWatch.

Group name

demo3

Stream name

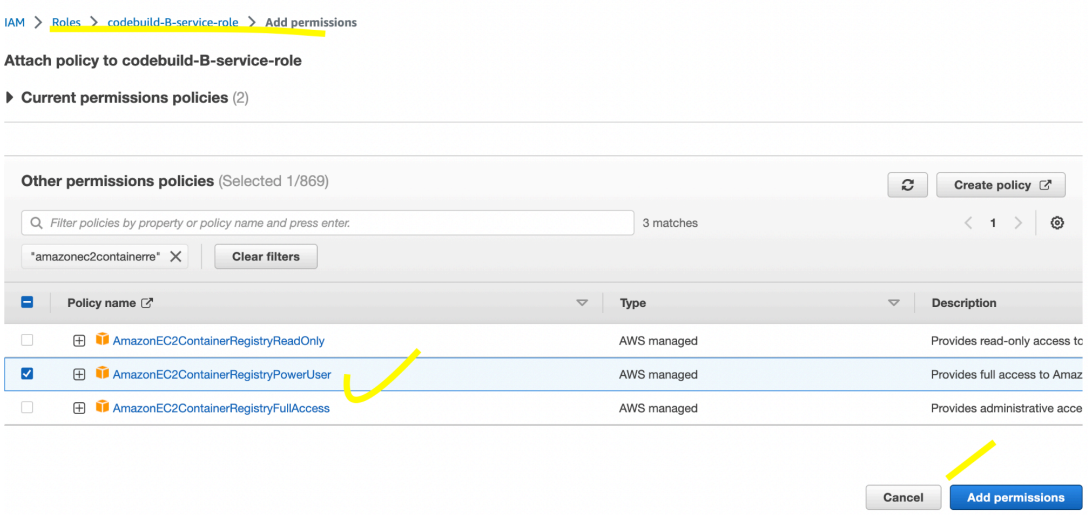
demo3

S3

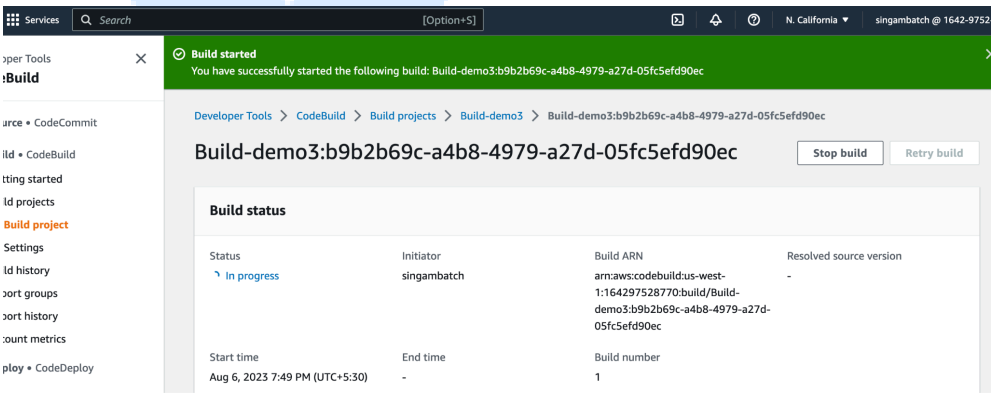
S3 logs - *optional*
Checking this option will upload build output logs to S3.

Cancel | Create build project

2.2 -> Go to codebuild IAM role and attach this policy so that the codebuild can access ECR - AmazonEC2ContainerRegistryPowerUser

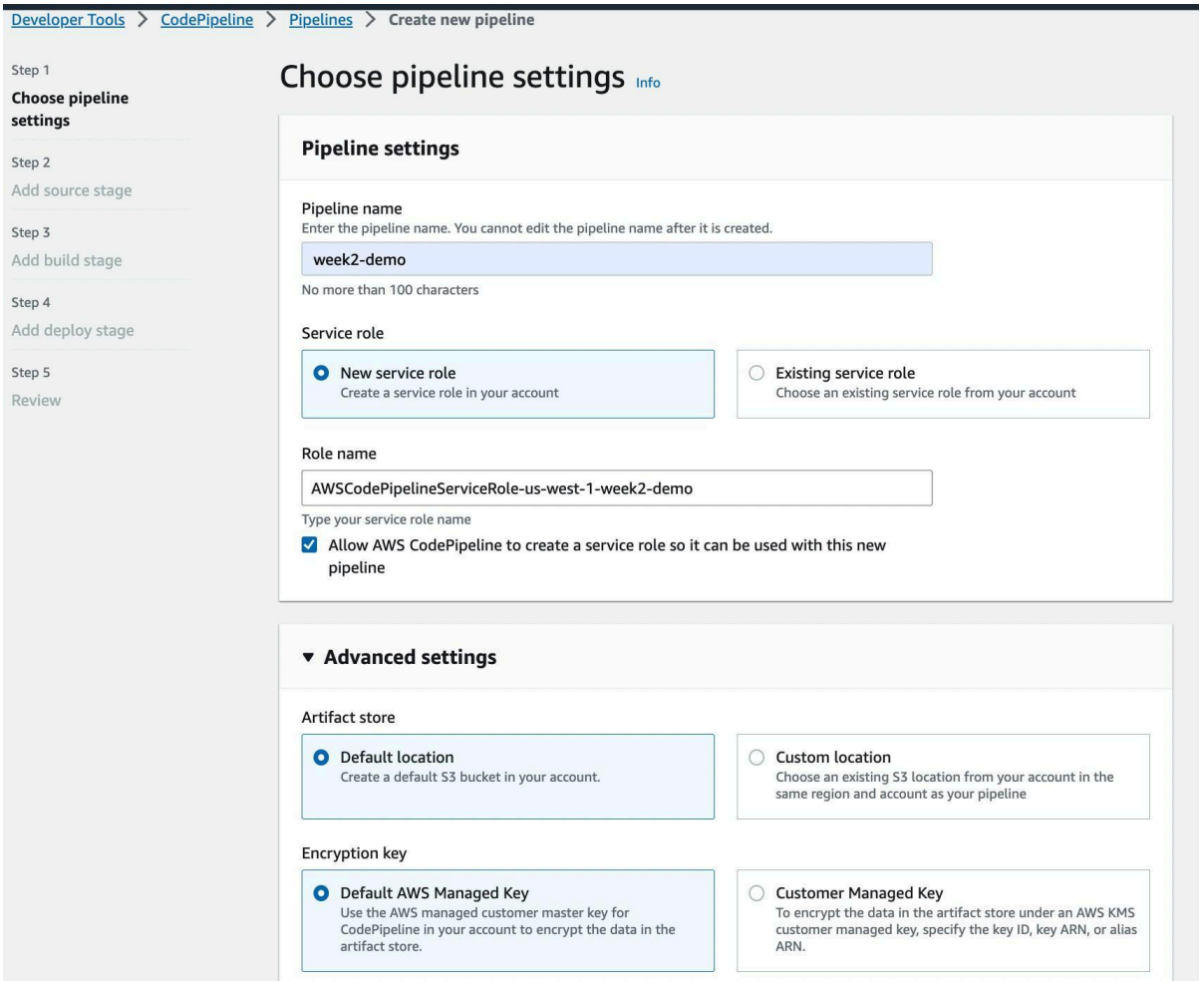


2.3 - Go to codebuild -> Run the build

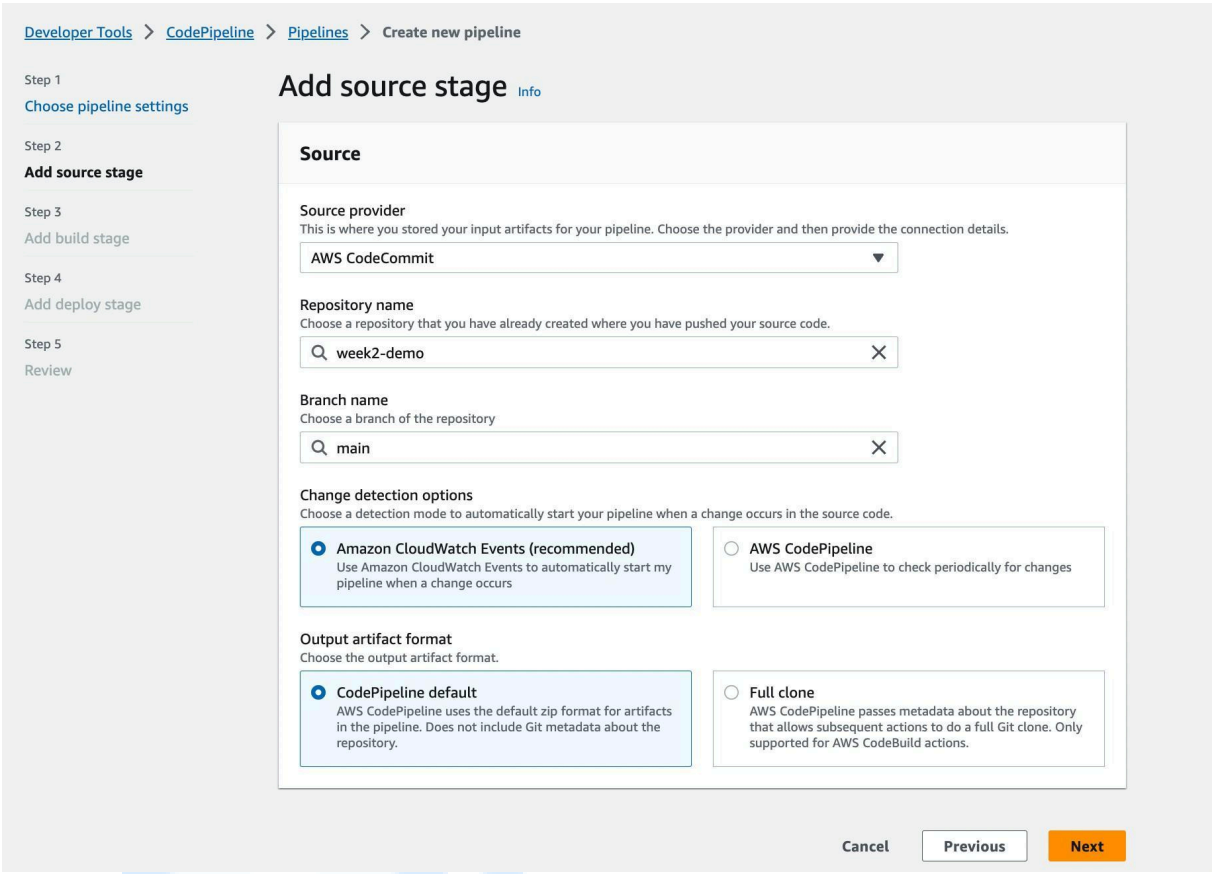


Stage3 - Create a CodePipeline

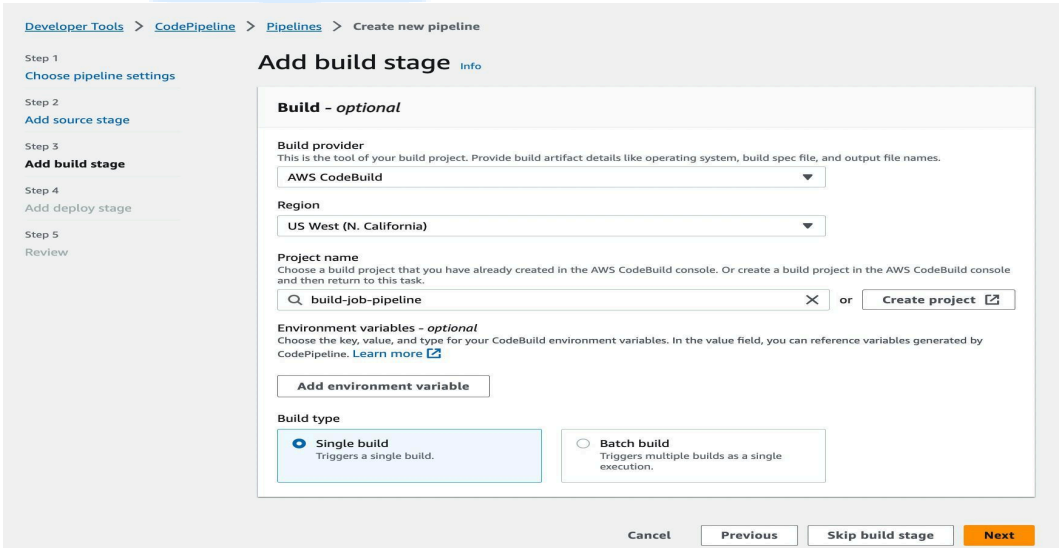
Step1 - Create the code pipeline



Step2-



Step3 -



Step4 -

Skip the deploy stage and save [The pipeline will trigger]

Happy

Learning,

Singam