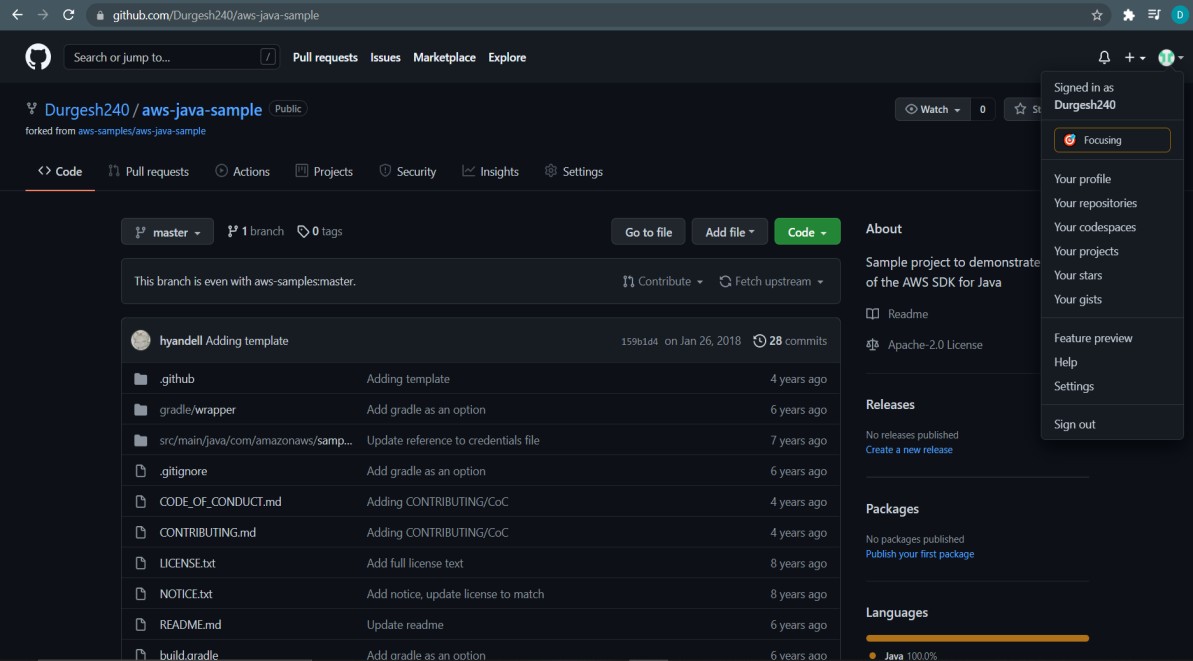
**Aim:** To Build Your Application using AWS CodeBuild and Deploy it on S3 bucket.

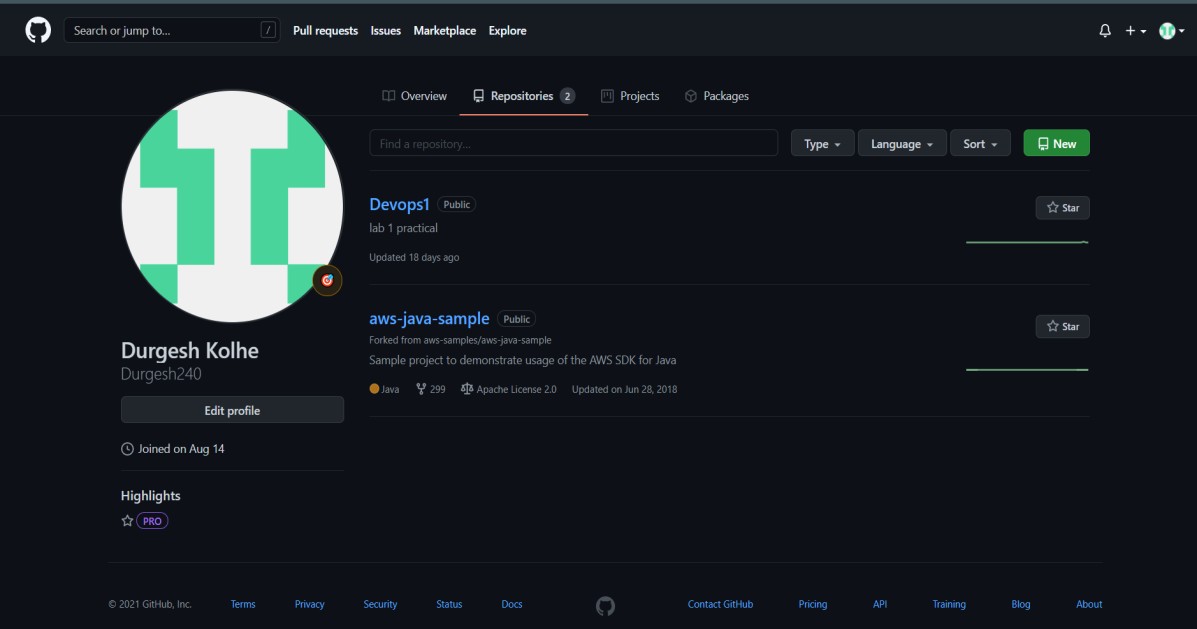
**Steps with screenshots:**

### STEP 1: INITIAL CONFIGURATION FOR CODECOMMIT

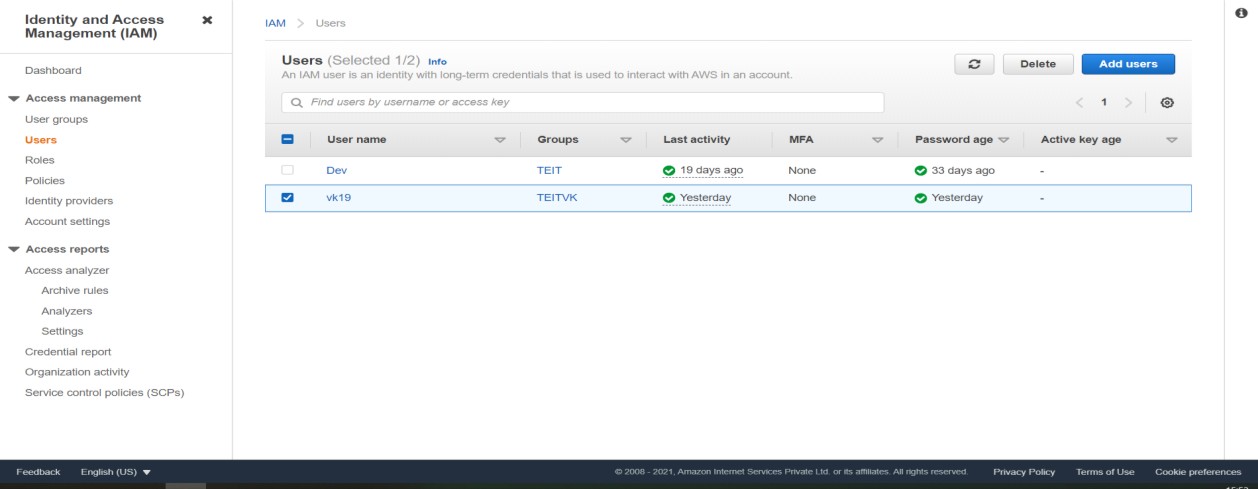
* Copy a repository in your GitHub account, so that this repository can be copied in aws acc.



* We can see other developer’s repository in “your repositories” section

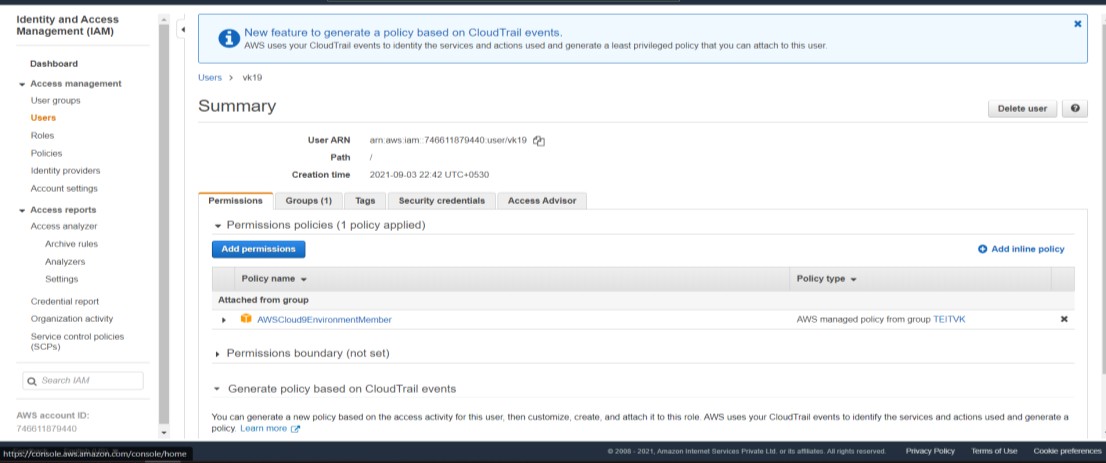


* Login as root user in AWS Console and then check account details of IAM User account details.



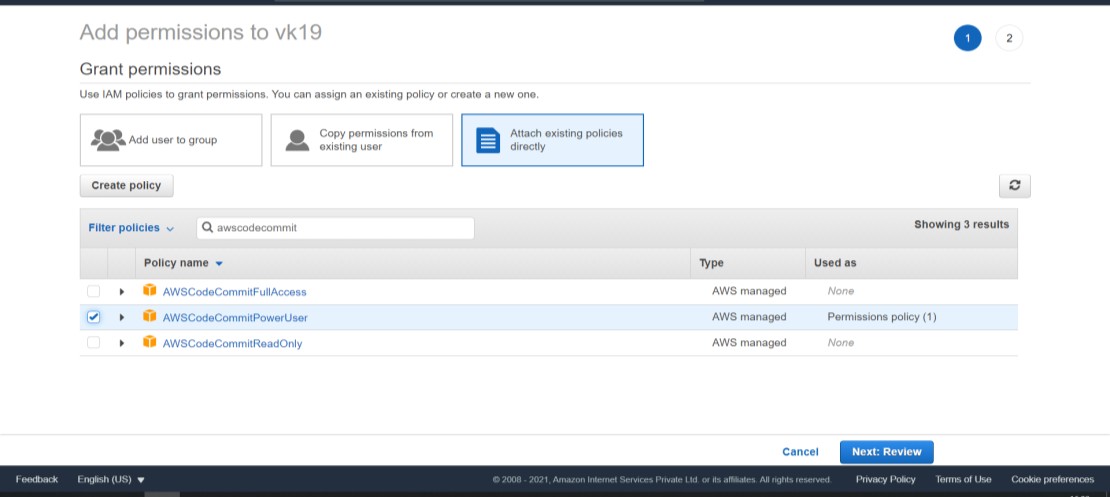
Dk19

* Click on your IAM User Account **(Dk19)** and then click on ADD PERMISSIONS.

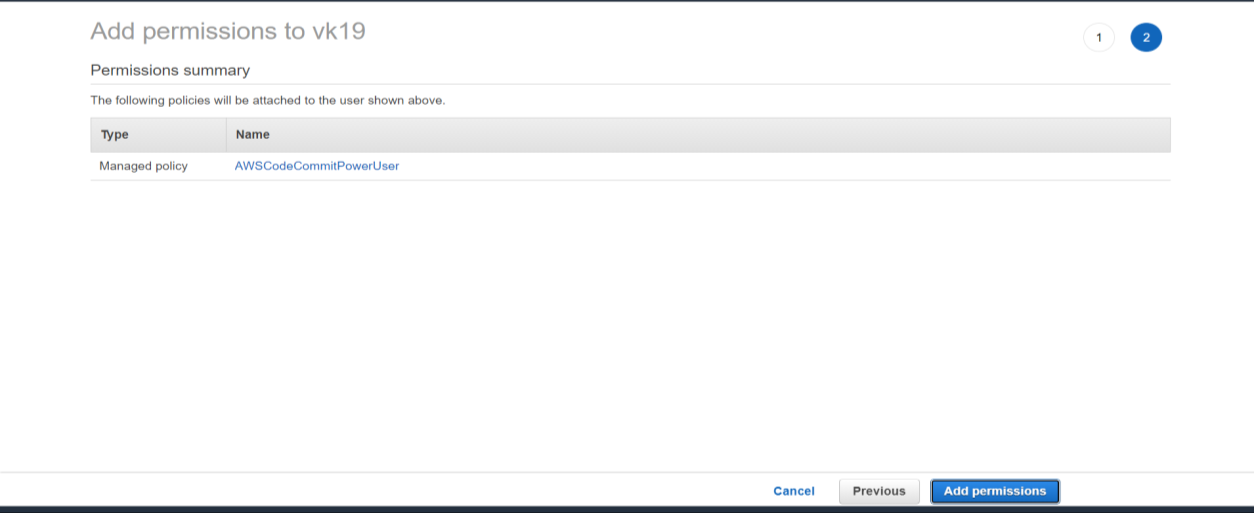


* In Add permissions select **Attach existing policies directly** and select [**AWSCodeCommitPowerUser**](https://console.aws.amazon.com/iam/home%23/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAWSCodeCommitPowerUser)

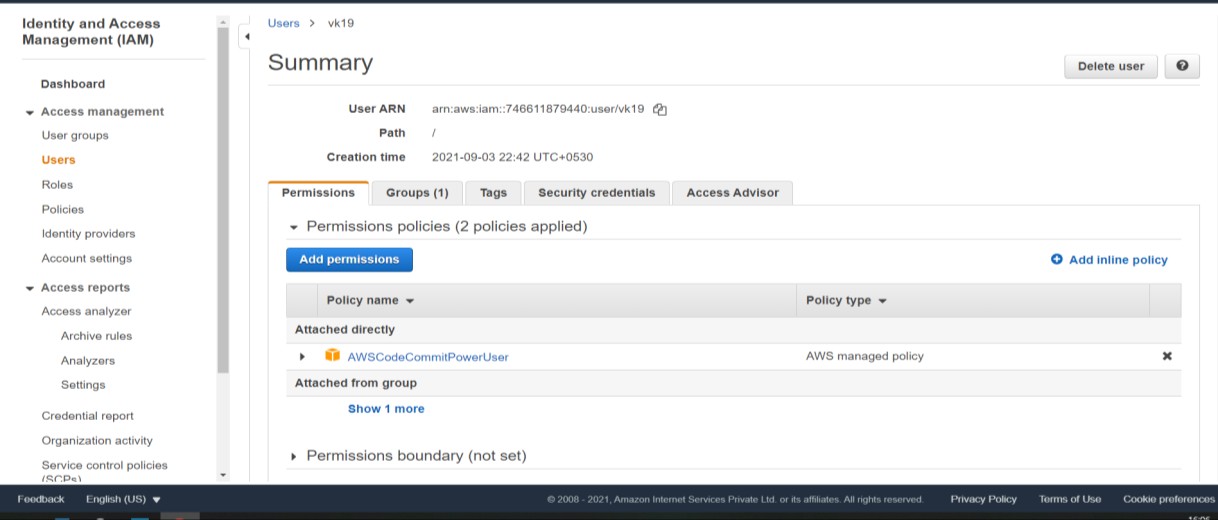
policy. Then click on **Next Review** button.



* After clicking on **Next Review** click on **Add Permission button**.

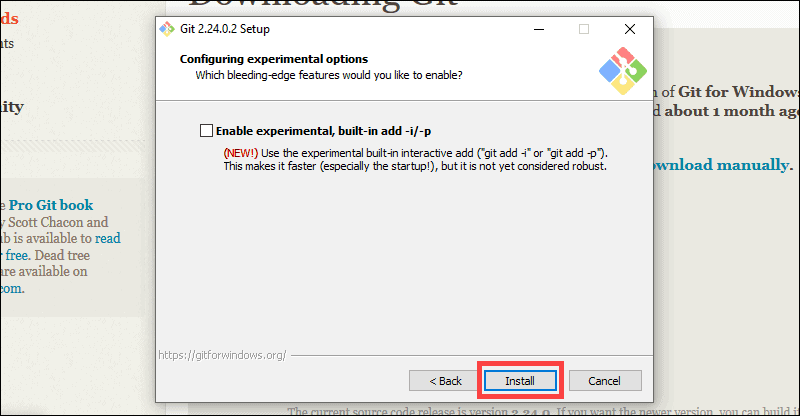


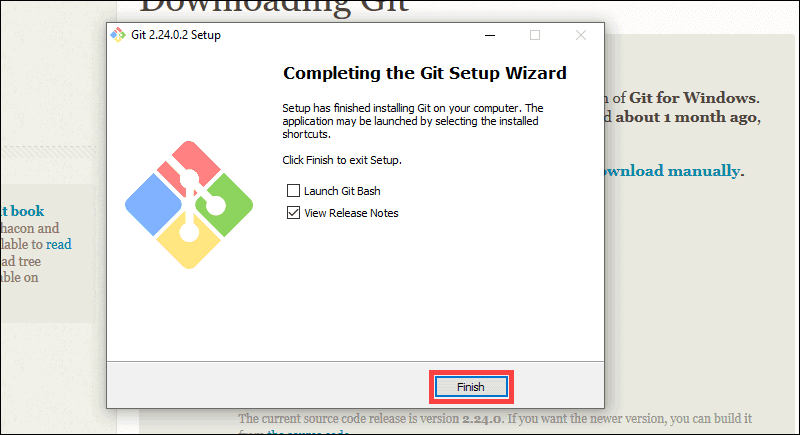
* Now you can see the permissions is set for **Dk19 user** under **SUMMARY** section.



### STEP 2: INSTALL GIT

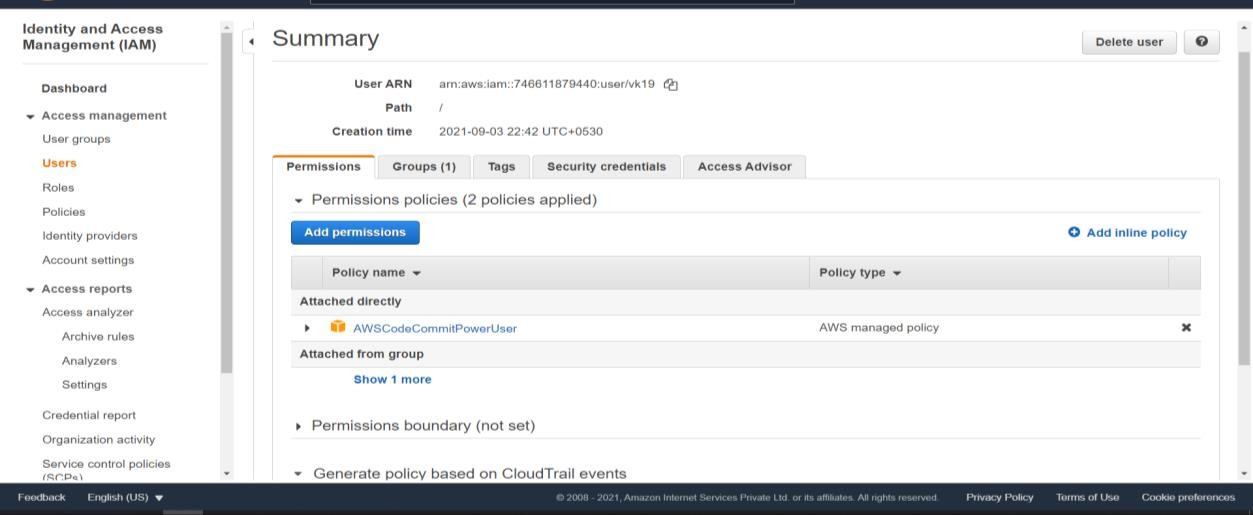
* To work with files, commits, and other information in CodeCommit repositories, you must install Git on your local machine. CodeCommit supports Git versions 1.7.9 and later. Git version 2.28 supports configuring the branch name for initial commits.



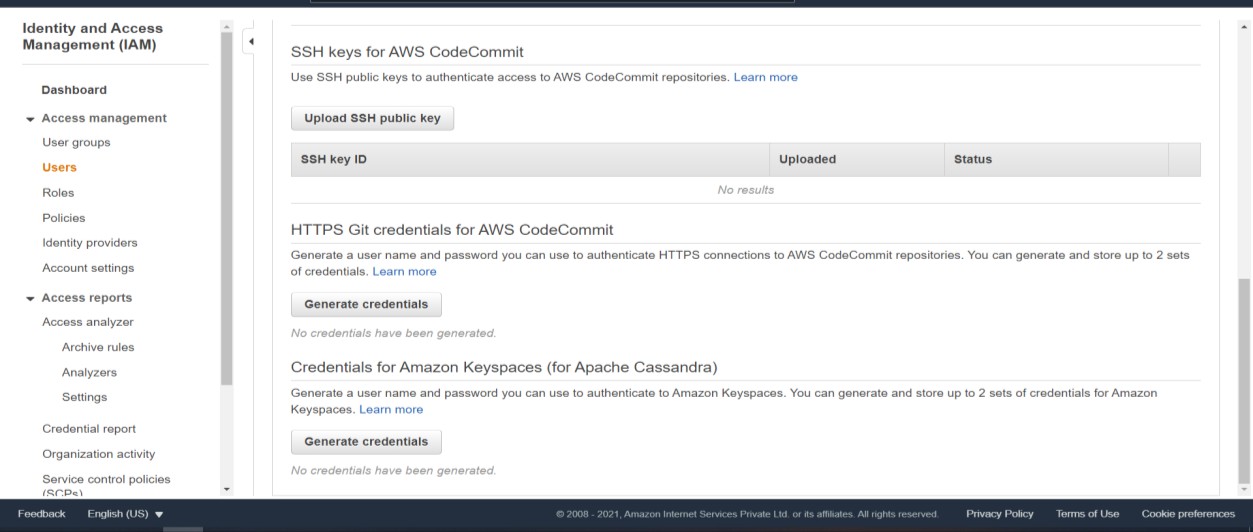


### STEP 3: CREATE GIT CREDENTIALS FOR HTTPS CONNECTIONS TO CODECOMMIT

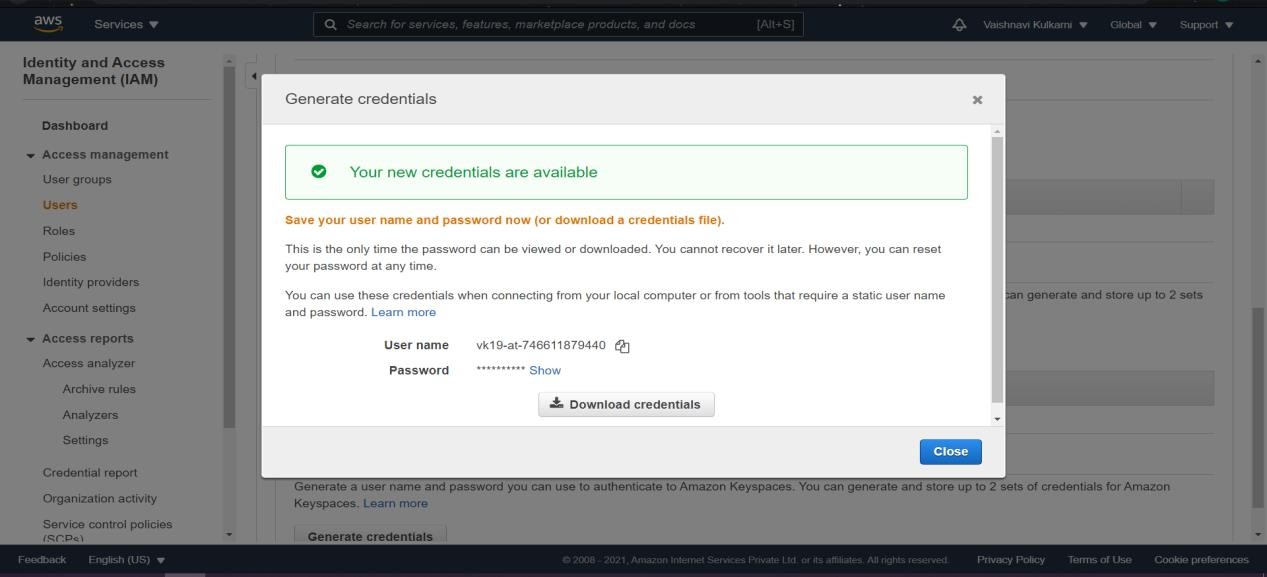
* In the IAM console, in the navigation pane, choose Users, and from the list of users, choose your IAM user.



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



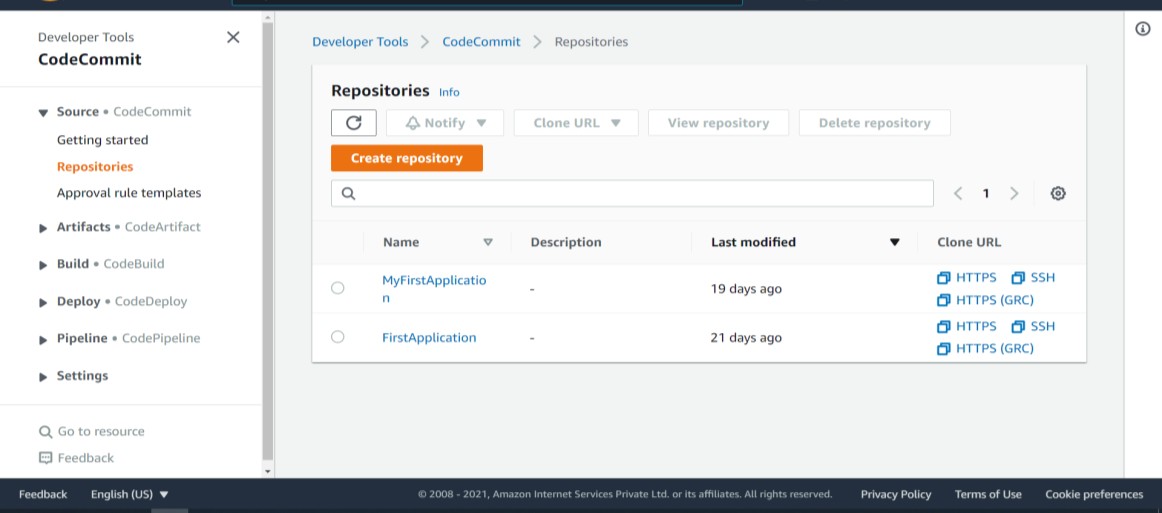
* Copy the user name and password that IAM generated for you.



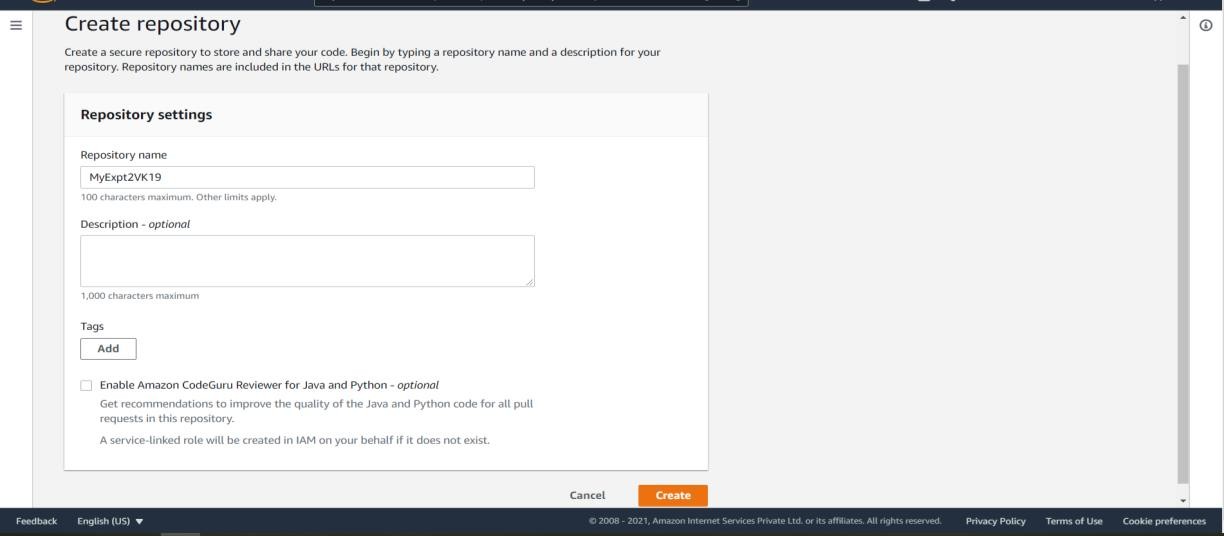
Dk19-at-746611879440

### STEP 4: CREATE A CODECOMMIT REPOSITORY

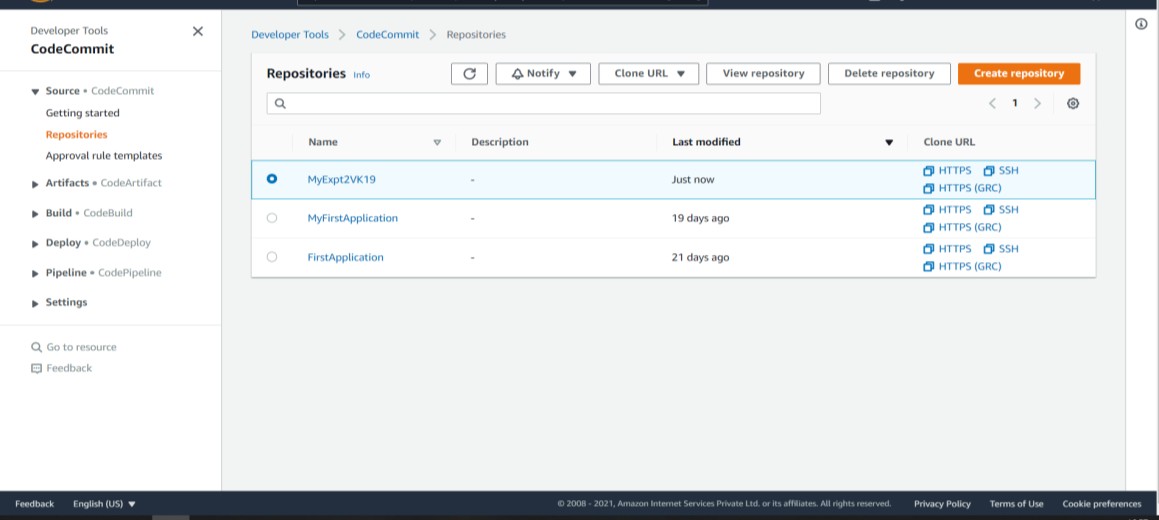
* Open the CodeCommit console, On the Repositories page, choose Create repository.



* On the Create repository page, in Repository name, enter a name for the repository. Choose Create.



* After it is created, the repository appears in the Repositories list**. (MyExpt2VK19)**

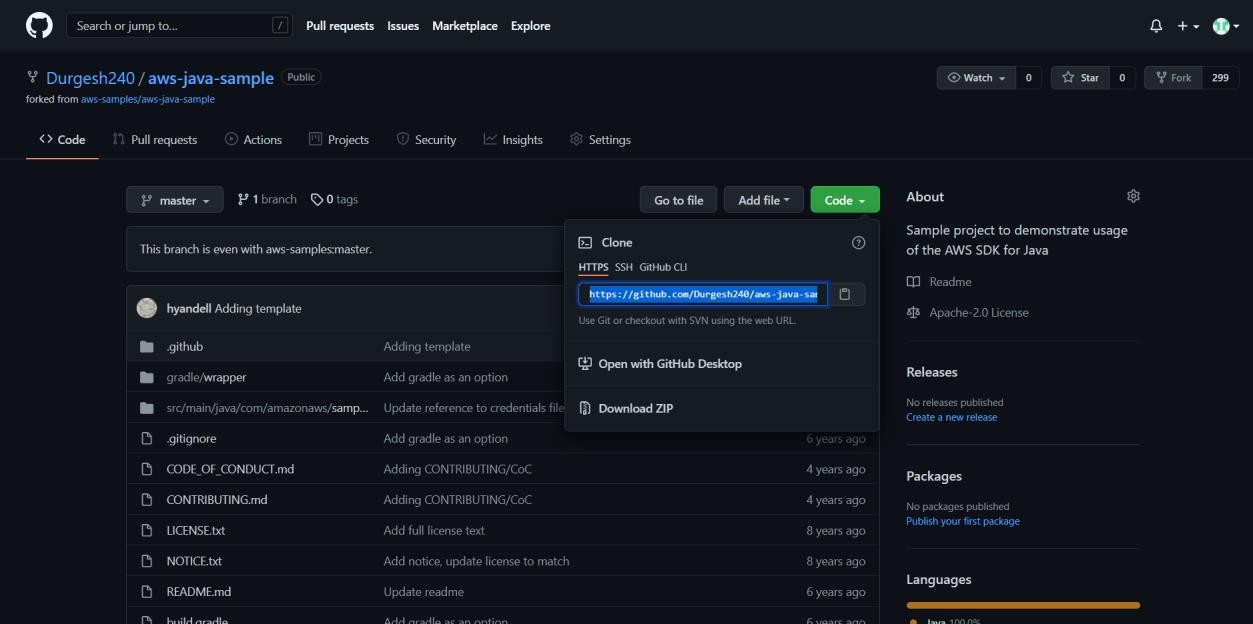


### STEP 5: CLONE THE REPOSITORY AND PUSH TO THE CODECOMMIT REPOSITORY

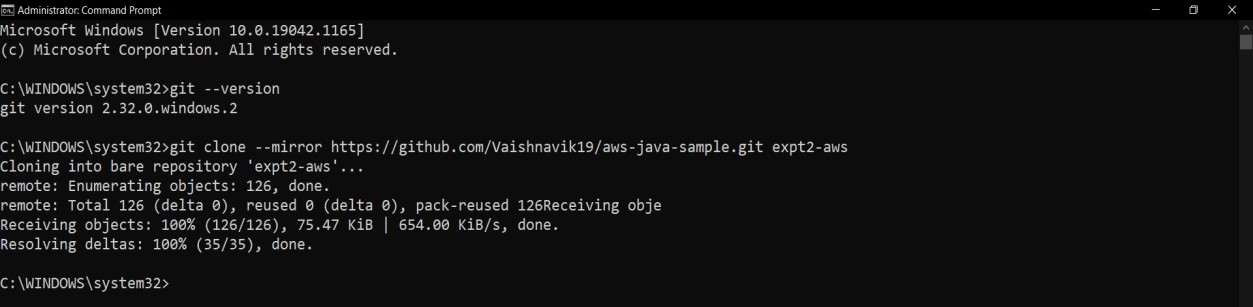
* Open terminal or command prompt on your local computer, run it as administrator. Check version of GIT.



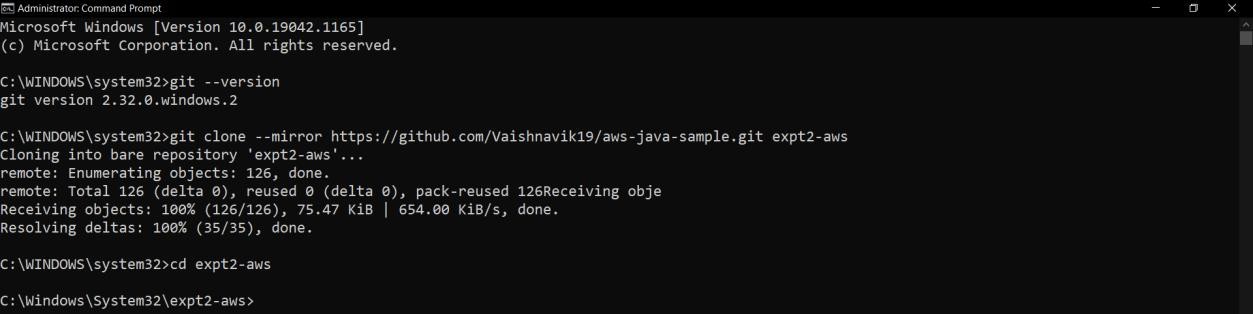
* Copy the URL of repository that you created on GIT SERVER.



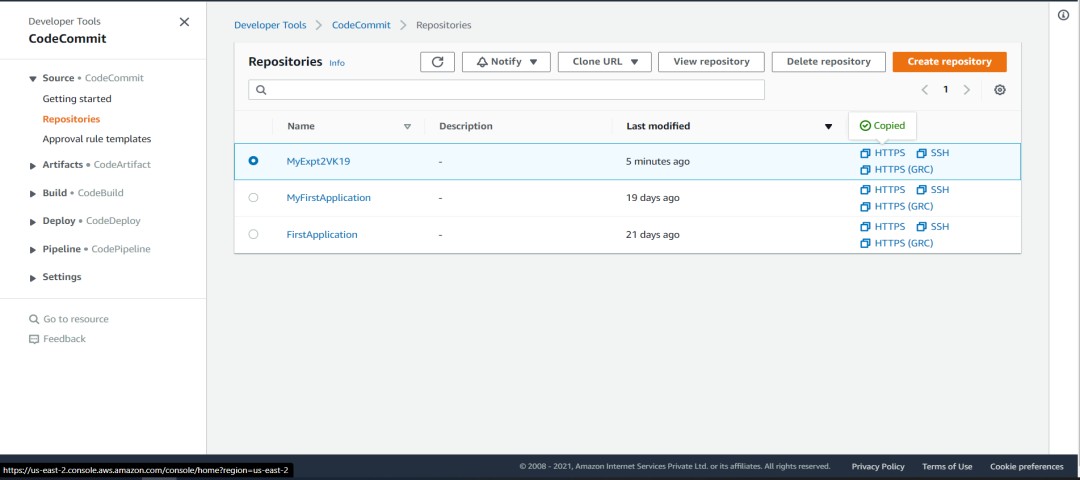
* Run the git clone command with the --mirror option to clone a bare copy of the remote repository into a new folder. **git clone --mirror url of github repo folder\_name**



* Change directories to the directory where you made the clone. **cd folder\_name**

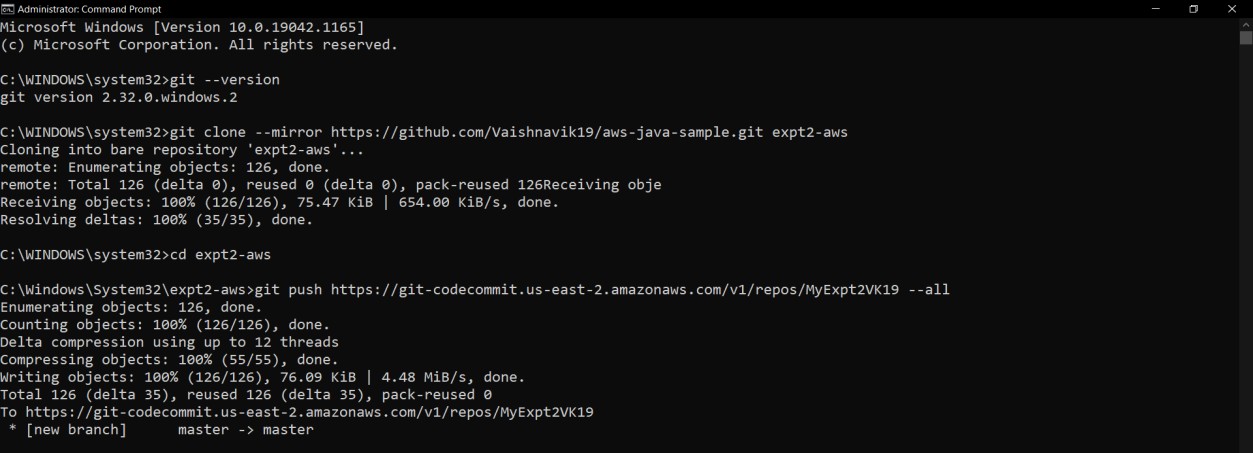


### Copy HTTPS URL of your repository for AWS CodeCommit.

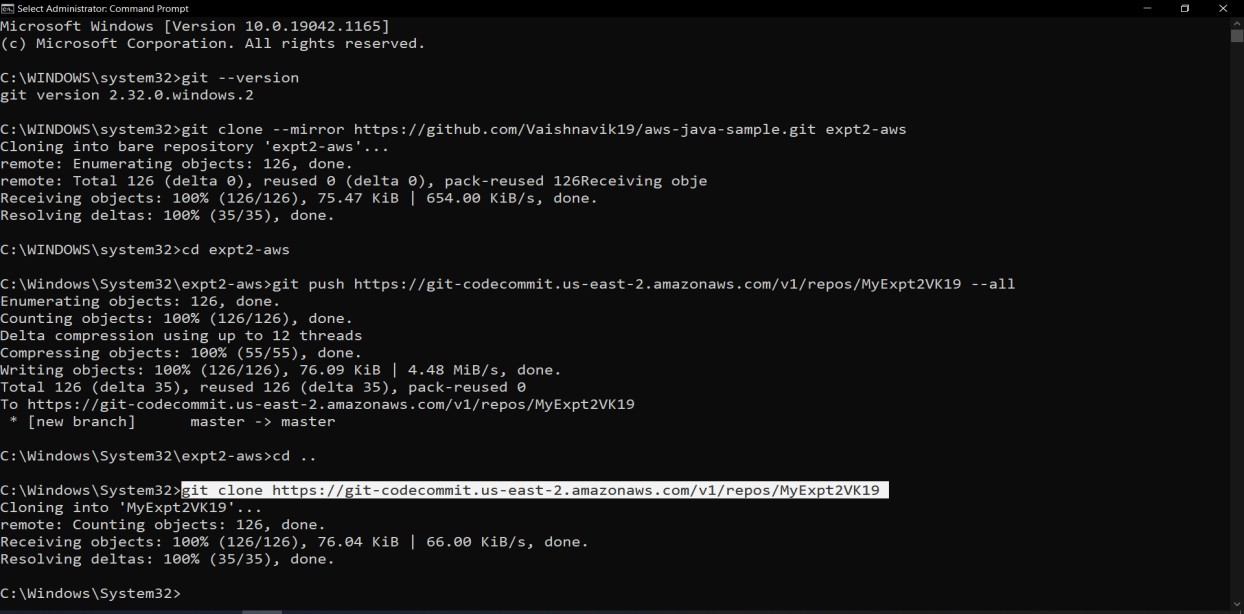


* Run the git push command, specifying the URL and name of the destination CodeCommit repository and the --all option. **Enter username and password of your IAM which you created.**

### git push url of codecommit –all

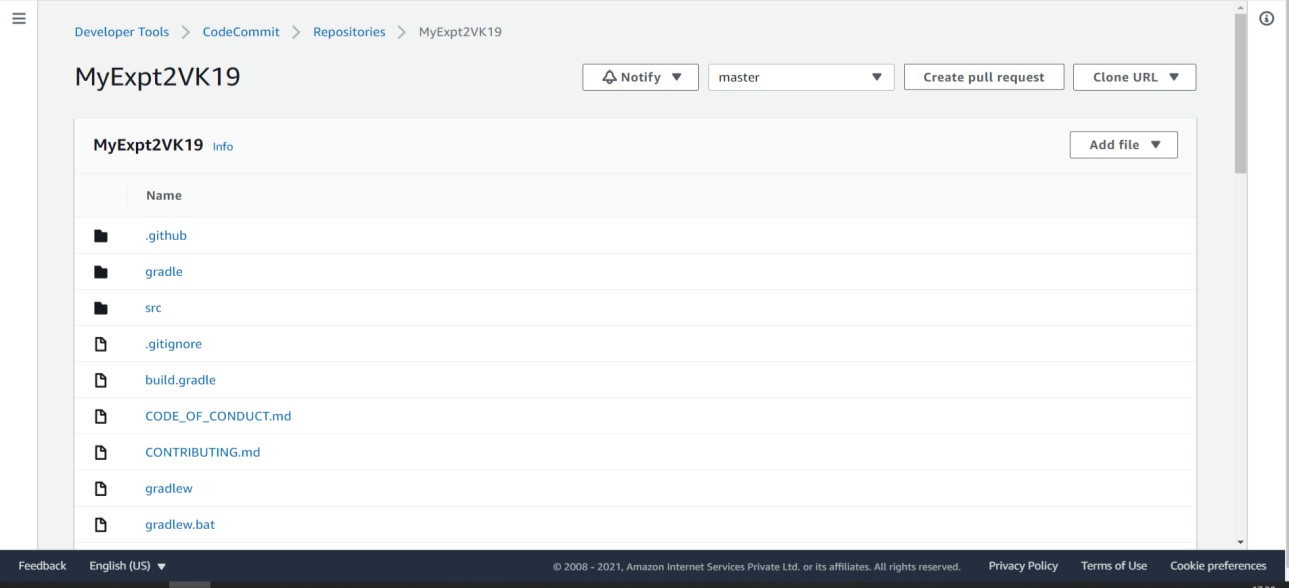


* To create a local repository with all the correct references for working with the repository in CodeCommit. **git clone url of codecommit**



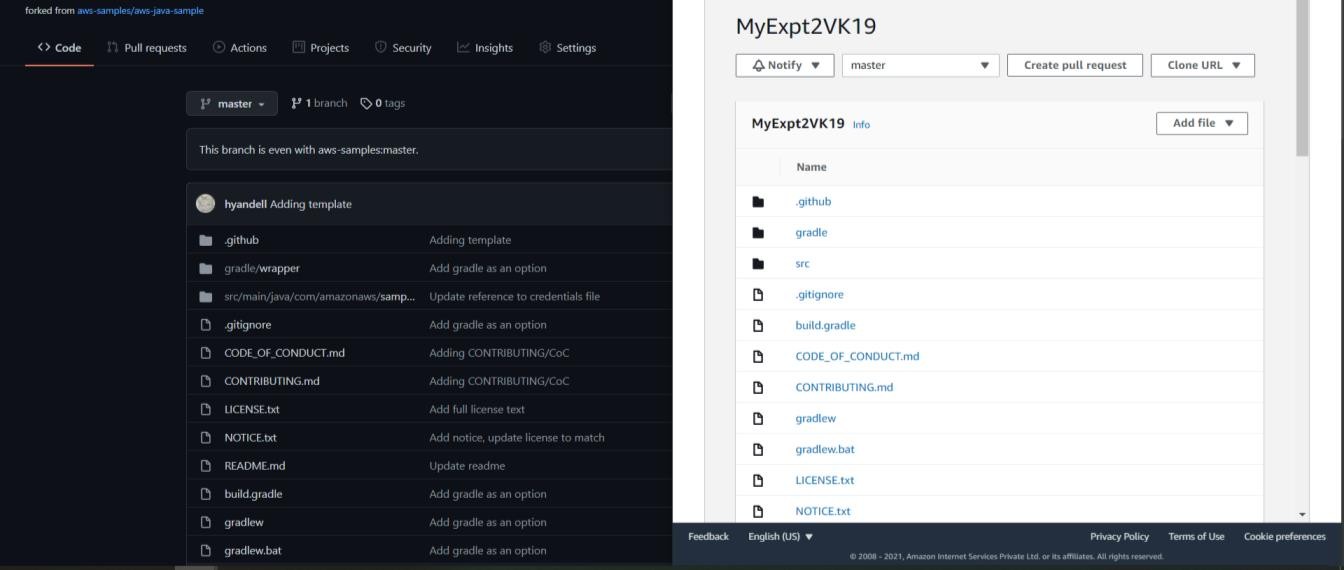
# STEP 6: VIEW FILES IN CODECOMMIT.

## Open the CodeCommit console. IN Repositories, choose the name of the repository. You should see the same files which were in GIT SERVER repository.



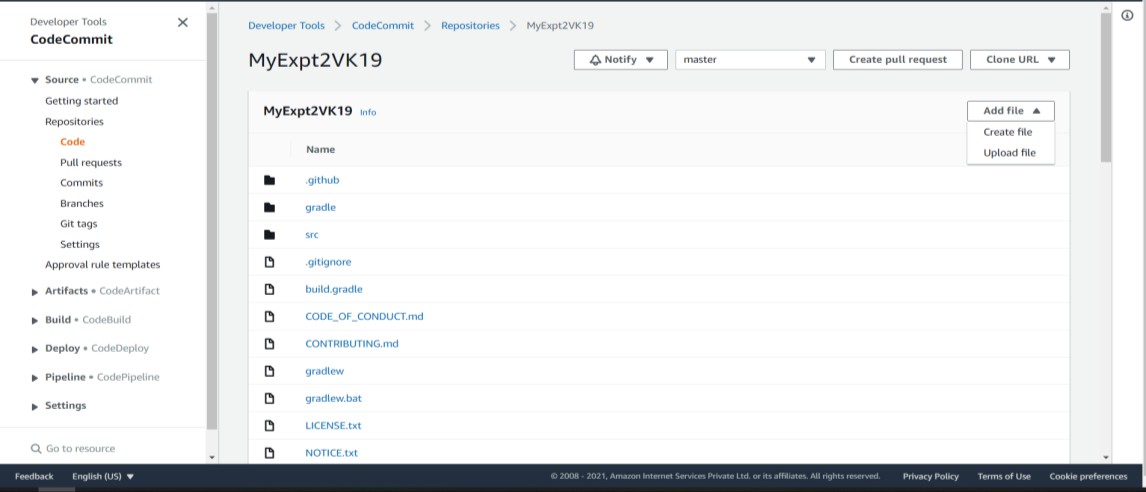
**(MyExpt2VK19)**

**As below you can see the same files from GIT SERVER has been transferred to AWS CodeCommit Repository.**



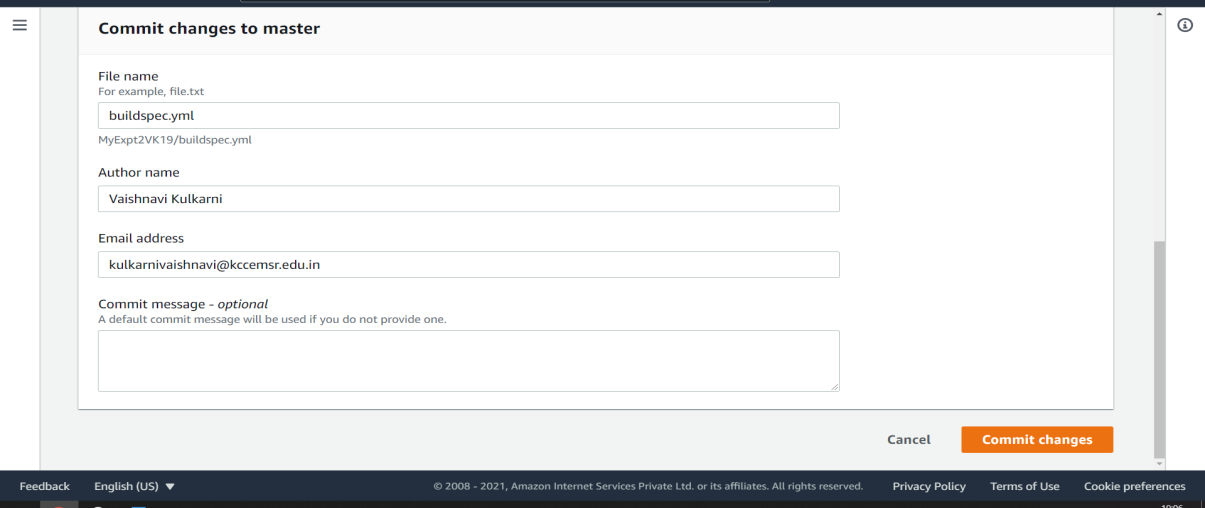
### Now we have to build the application using CodeBuild AWS Service and Deploy it on S3 Bucket STEP 7: Again, Login as a root user and select your repository from CodeCommit and click on

**ADD FILES** →**CREATE FILE**



**STEP 8: Add .yml code file in your project give the filename, author name and enter your email address then click on Commit changes.**

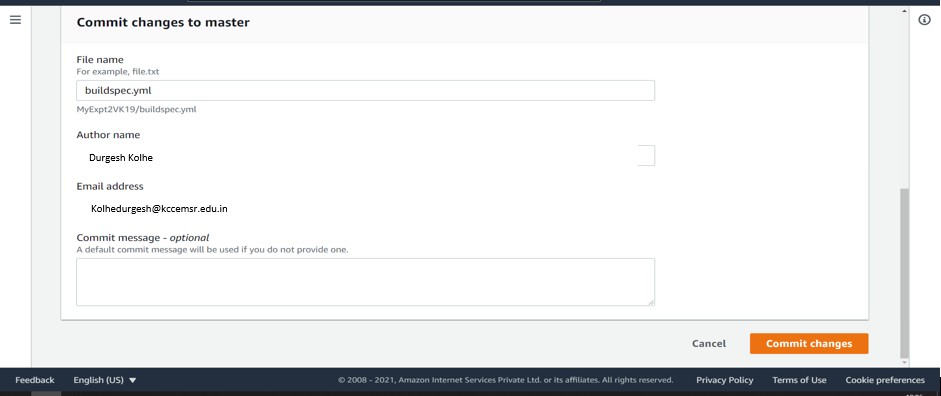




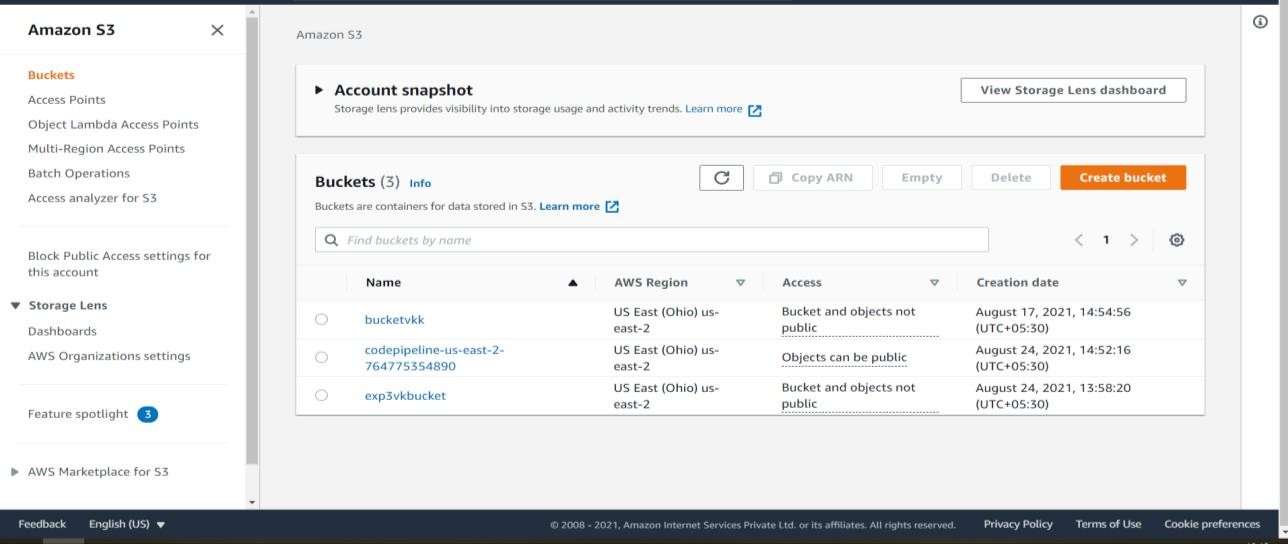
[Kolhedurgesh@kccemsr.edu.in](mailto:Kolhedurgesh@kccemsr.edu.in)

Durgesh Kolhe

* After adding check in your repository→under your sample project that buildspec.yml file is added or not.

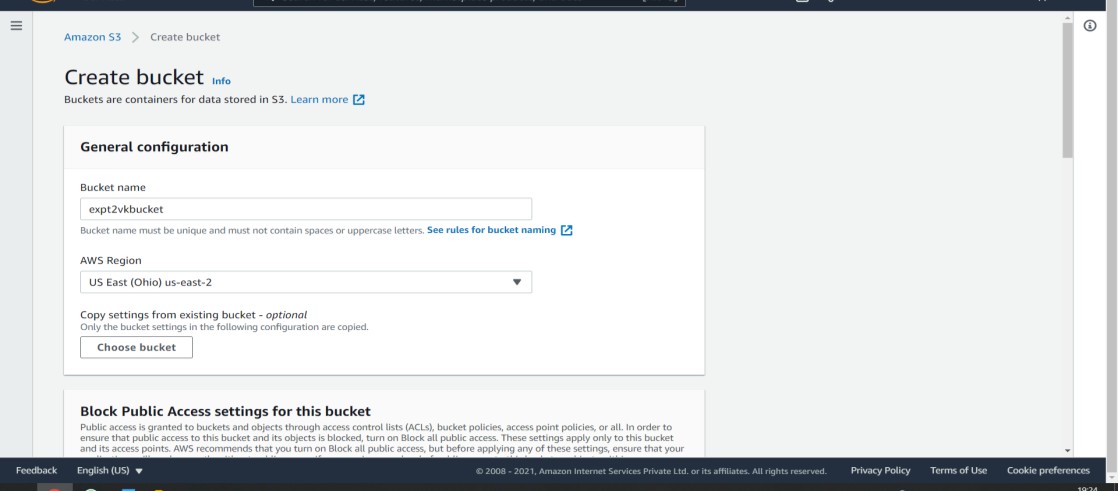


### Now for building our current project we will use S3 Bucket STEP 9: Select S3 Service for creating buckets in it → Click on create bucket button



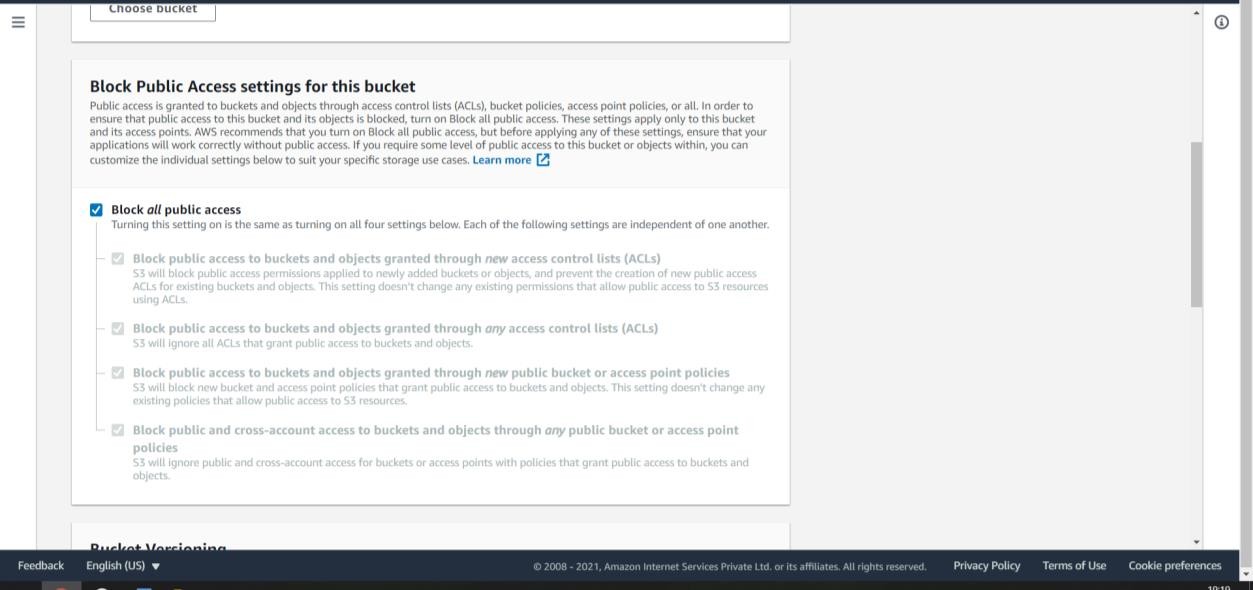
exp3dkbucket

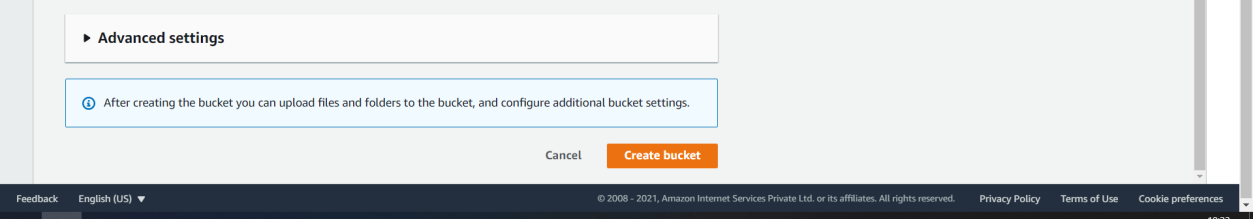
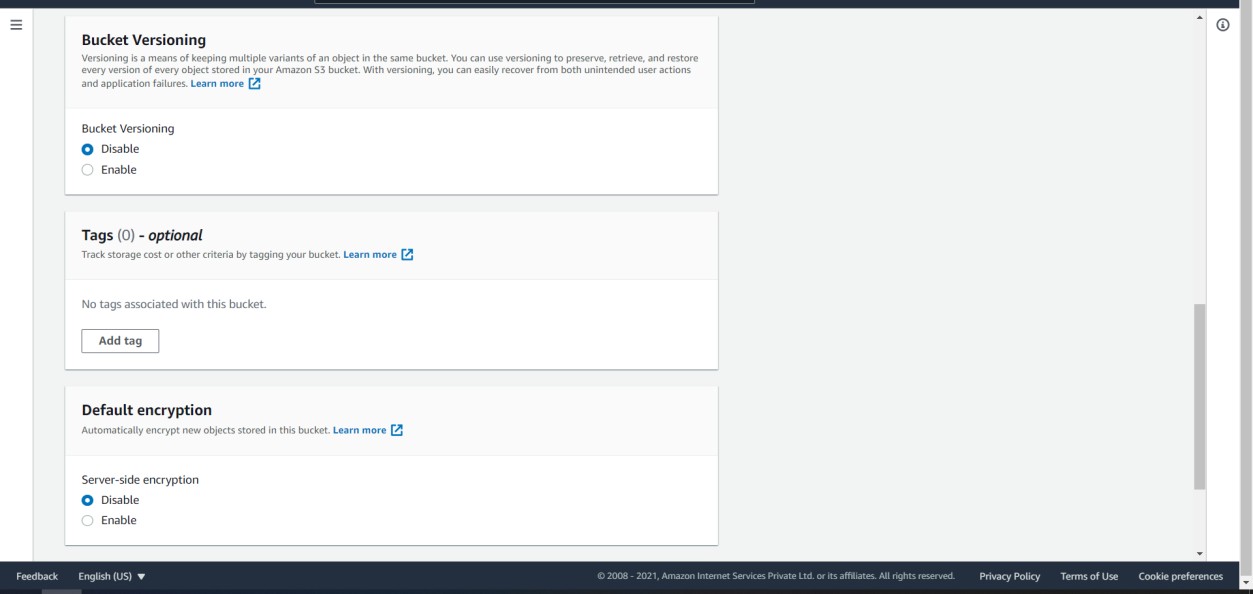
* Enter the bucket name of your wish and keep the location as it is. (expt2dkbucket)

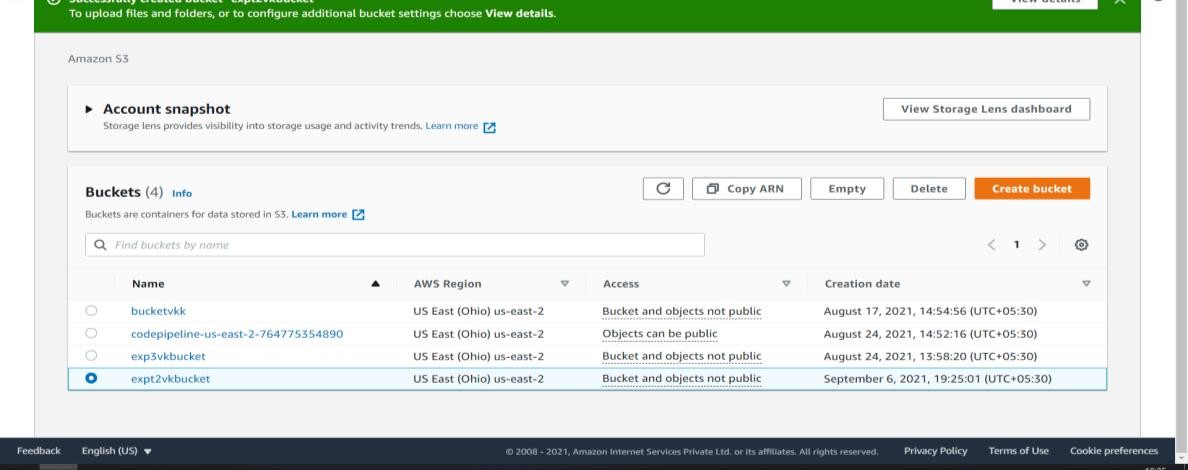


exp2dkbucket

* Keep “Block Public Access settings for this bucket” as it is, keep Bucket Versioning and Default encryption DISABLED and then click on create bucket.



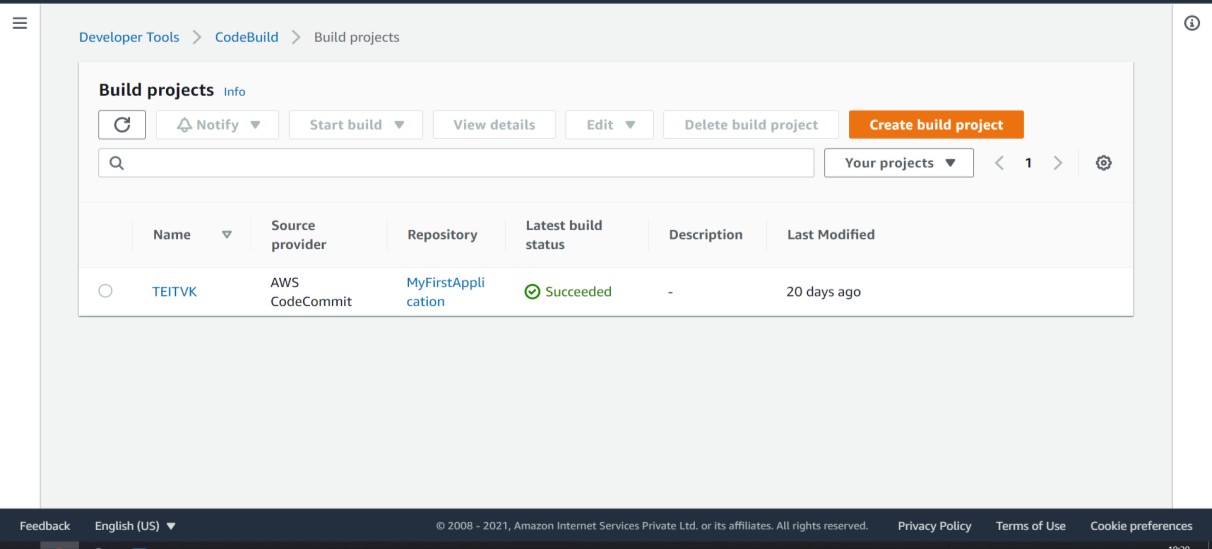




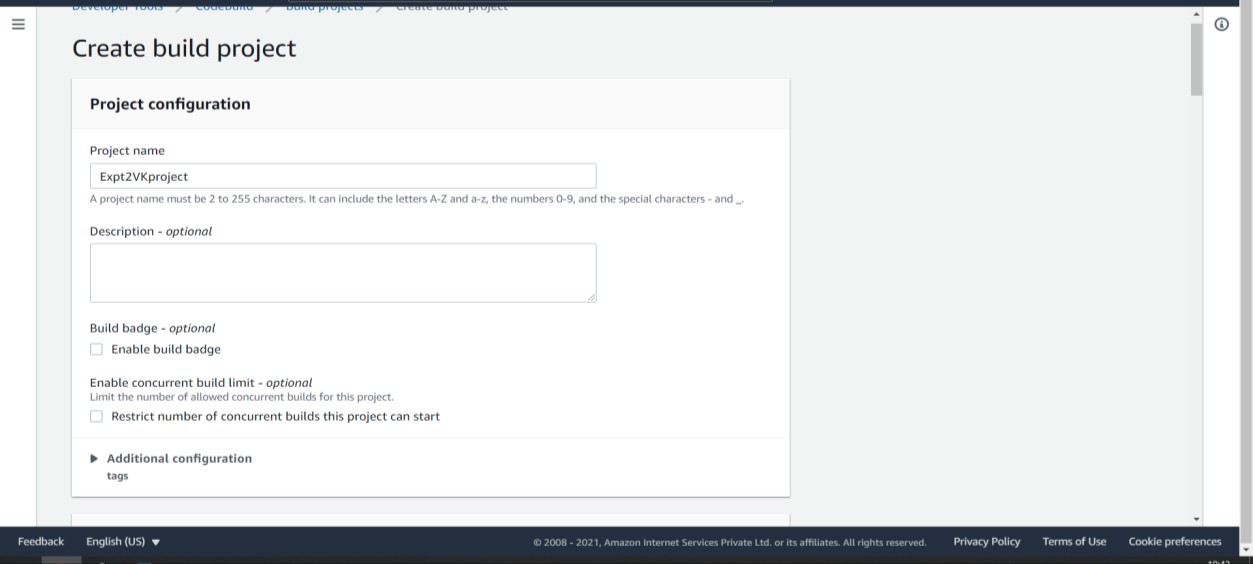
exp2dkbucket

This is how bucket got created which is used for storing our content.

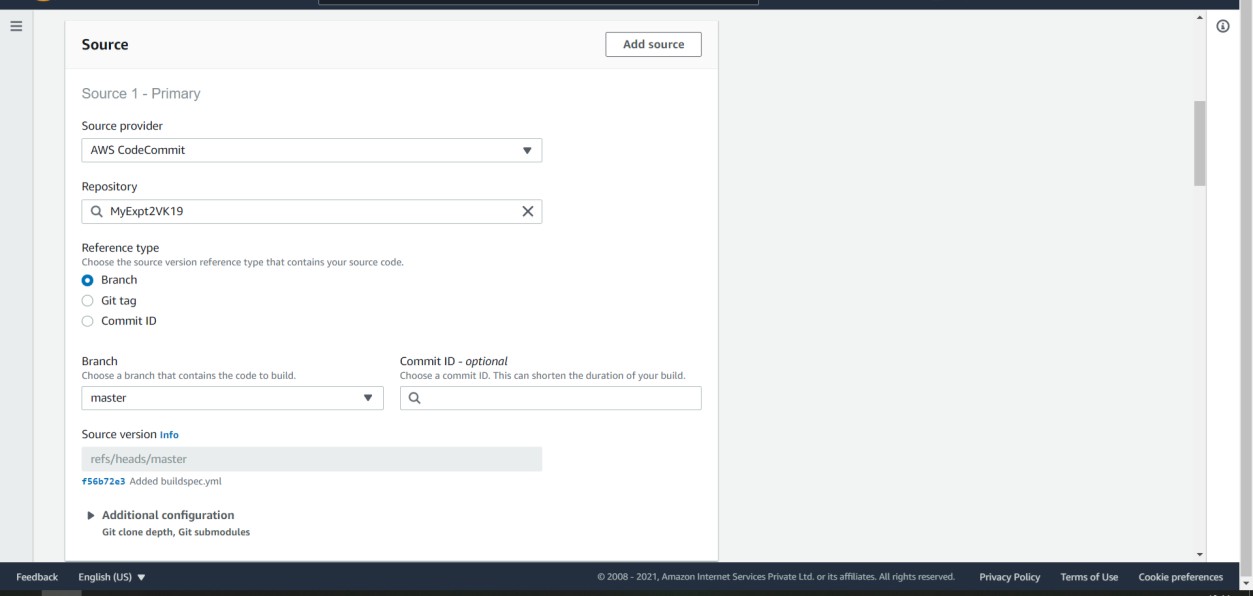
### STEP 10: Now go to CodeBuild AWS Service and then click on Create build Project.



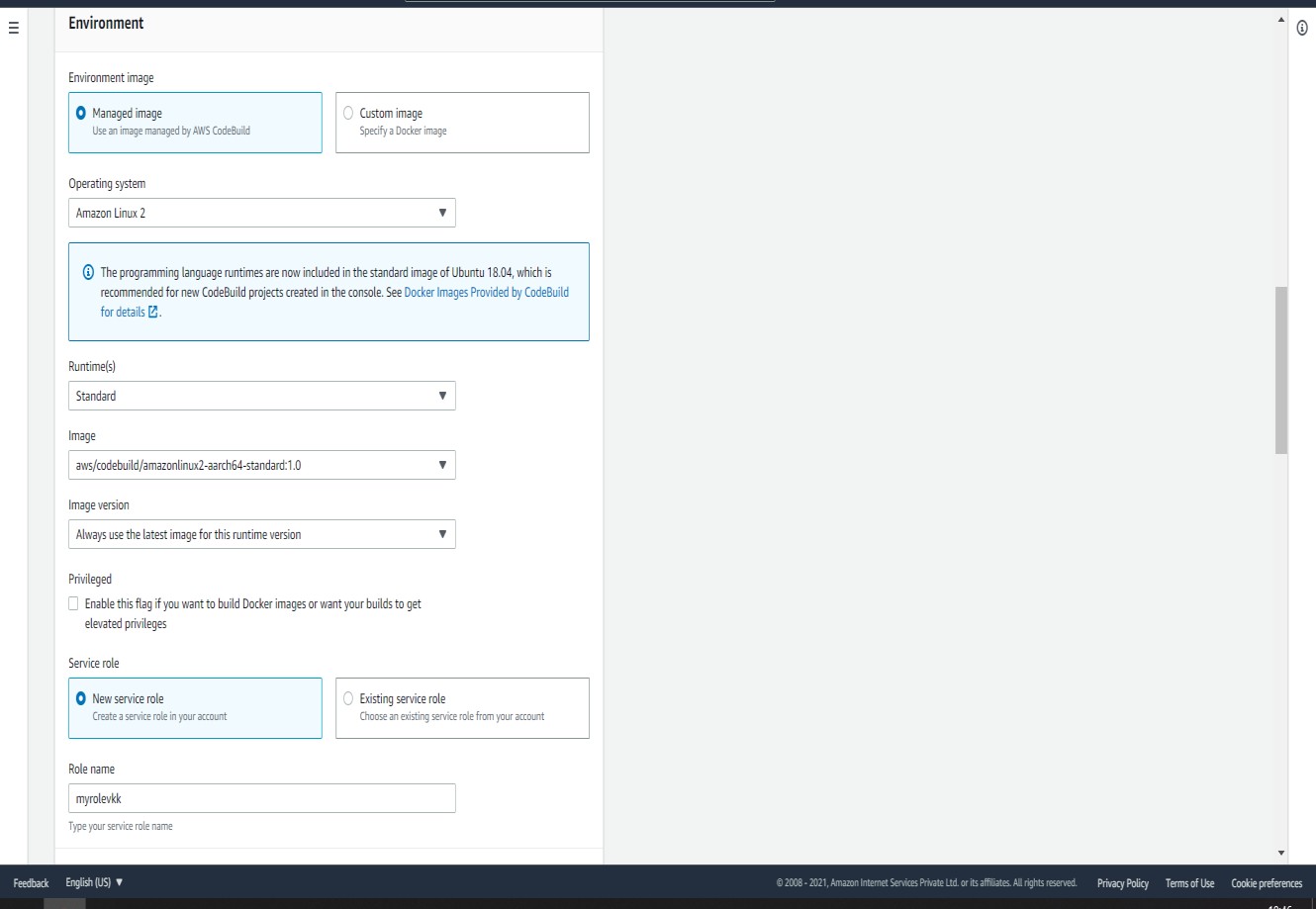
* Enter the name for your project (Expt2VKproject)



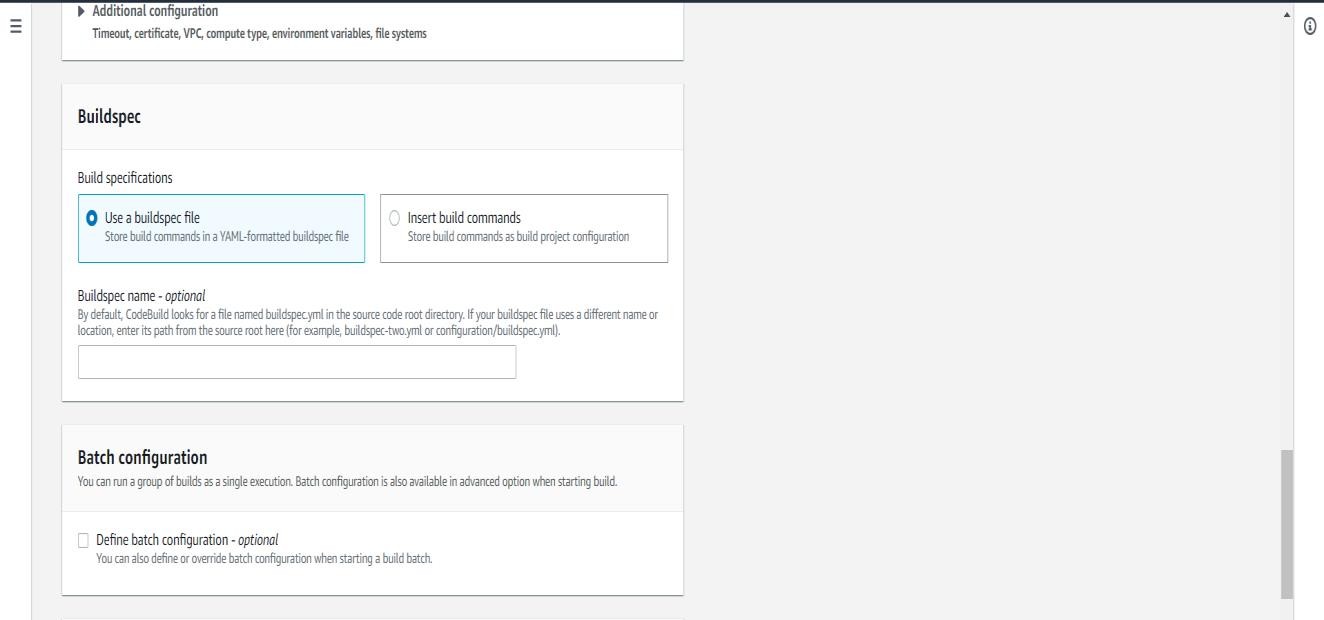
* In Source section Source provider→AWS CodeCommite, Repository→In which all the files of project are there select that repository only, Branch→ Master



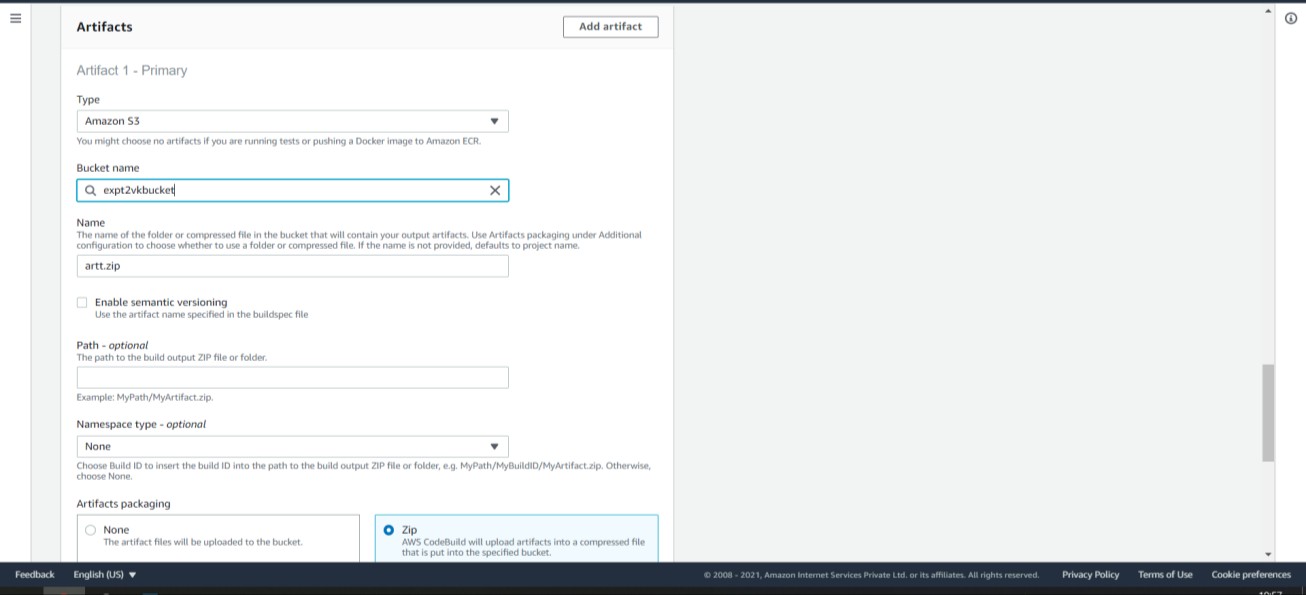
* In Environment Section Select→Managed Image, Operating system→Amazon Linux 2, Runtime→Standard, Image→ 1st option in list(1.0), Service role→New service role and enter any role name of your choice.



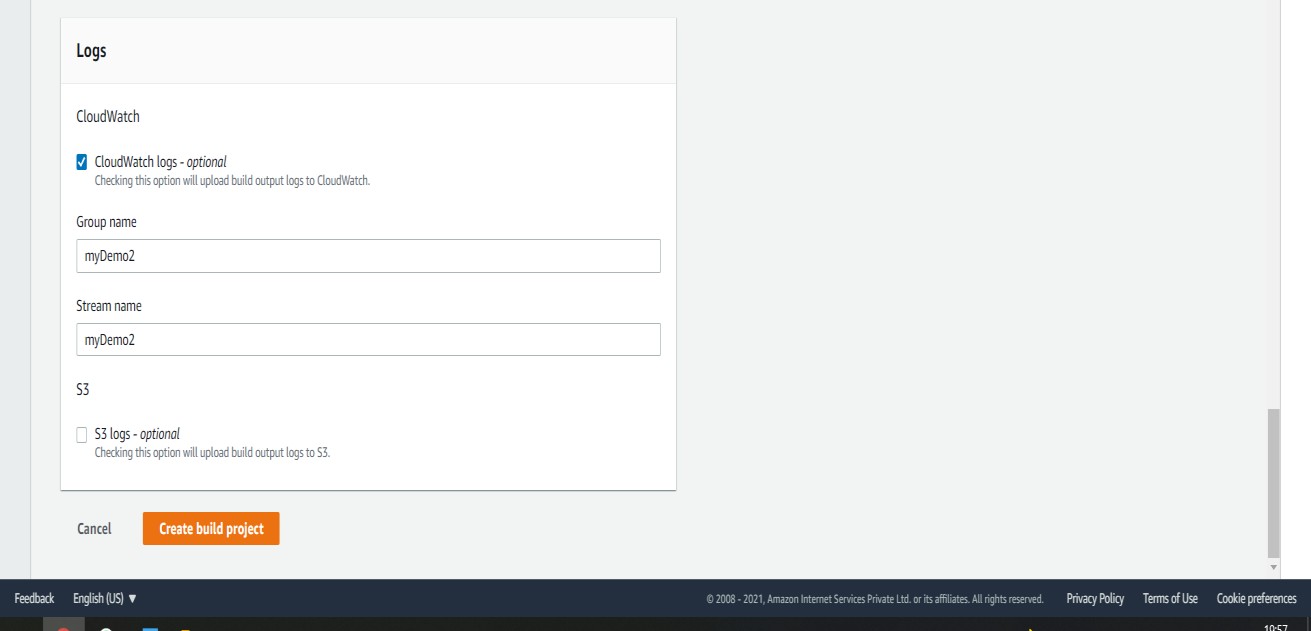
* In Buildspec Section Select→ Use a buildspec file (i.e. YAML-Formatted file)



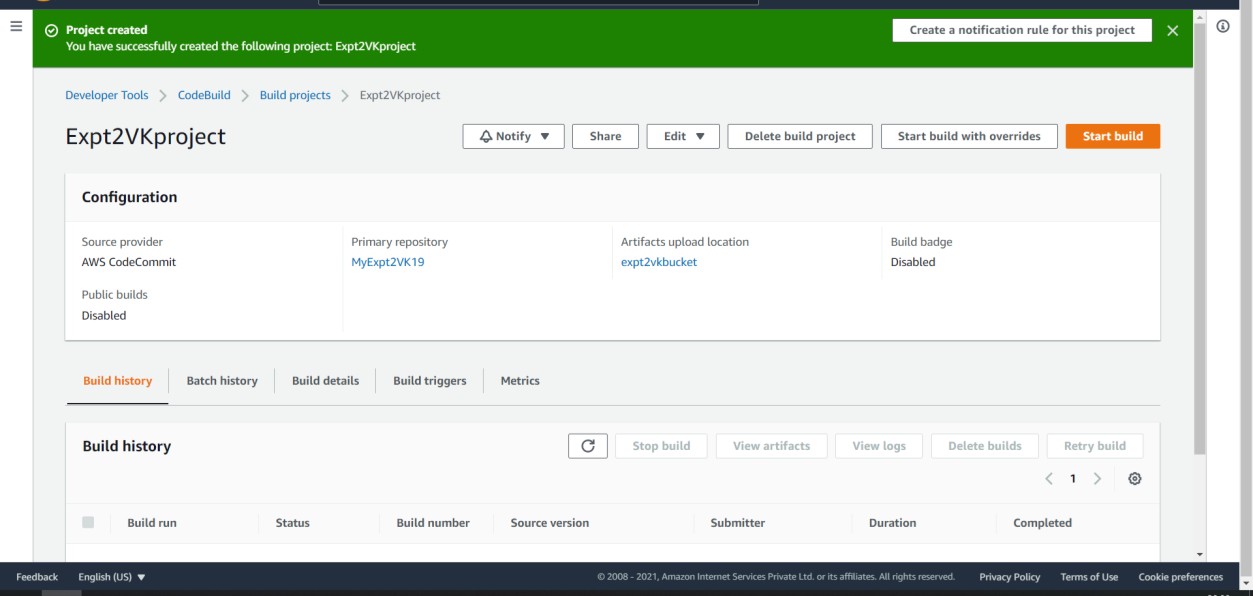
* In Artifacts Section Select Type→ Amazon S3, Bucket name→Previously which we created(expt2vkbucket), Name→Of your choice(artt.zip), Select Artifacts Packaging→Zip



* In logs section type Group name of your wish the same name can be of stream name as well

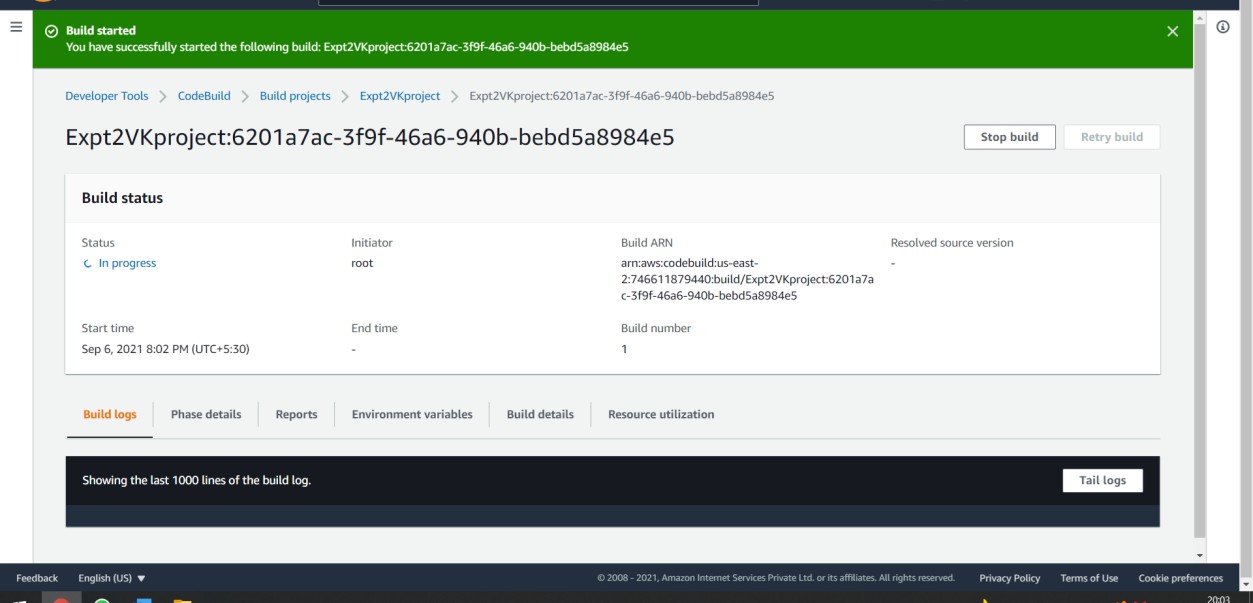


* This is how our Application is built with project name and the bucket which we created.

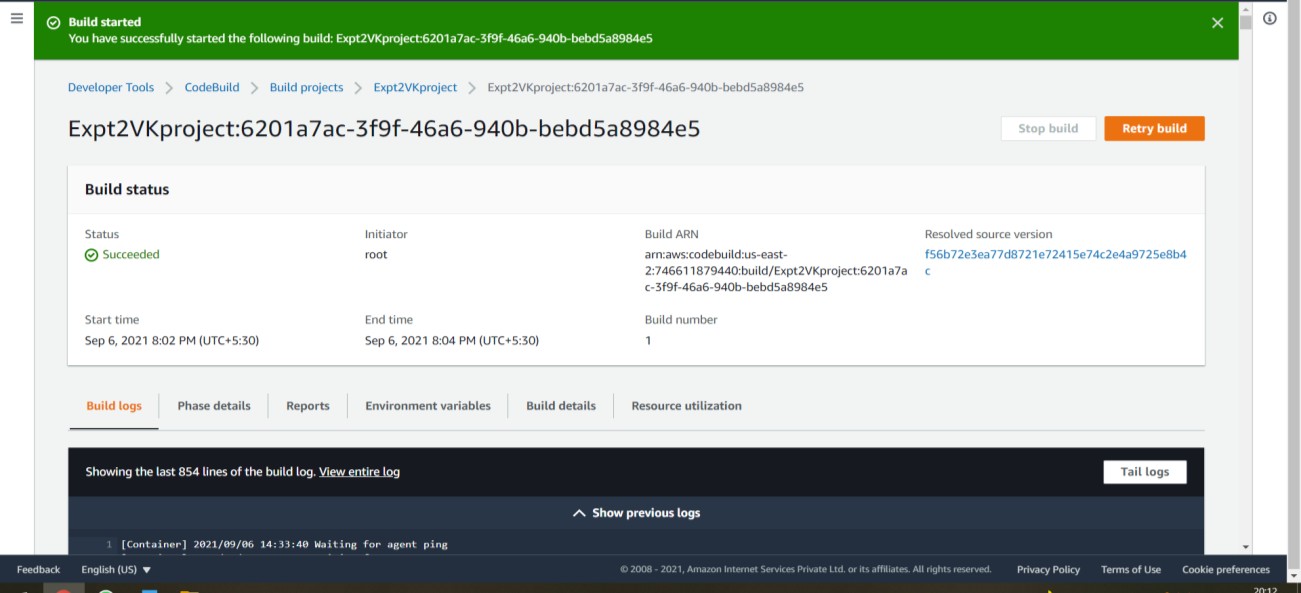


### STEP 11: NOW WE ARE GOING TO RUN THAT PROEJCT IN CODEBUILD FOR THAT CLICK ON START BUILD

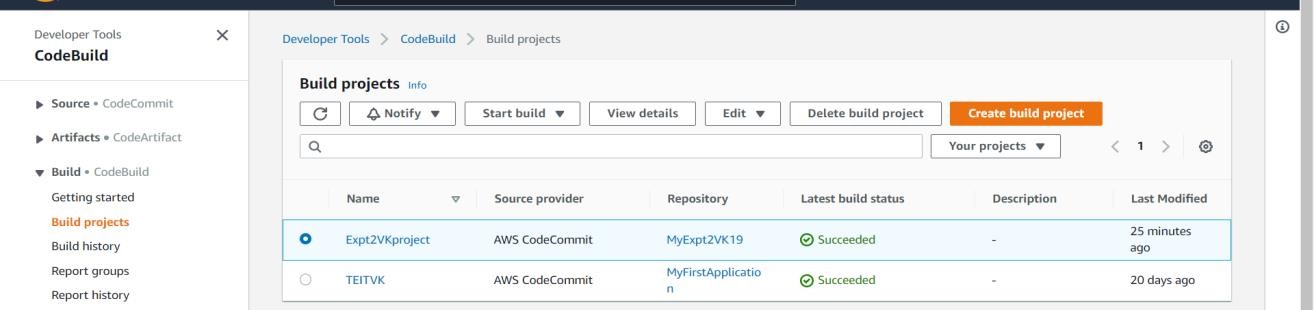
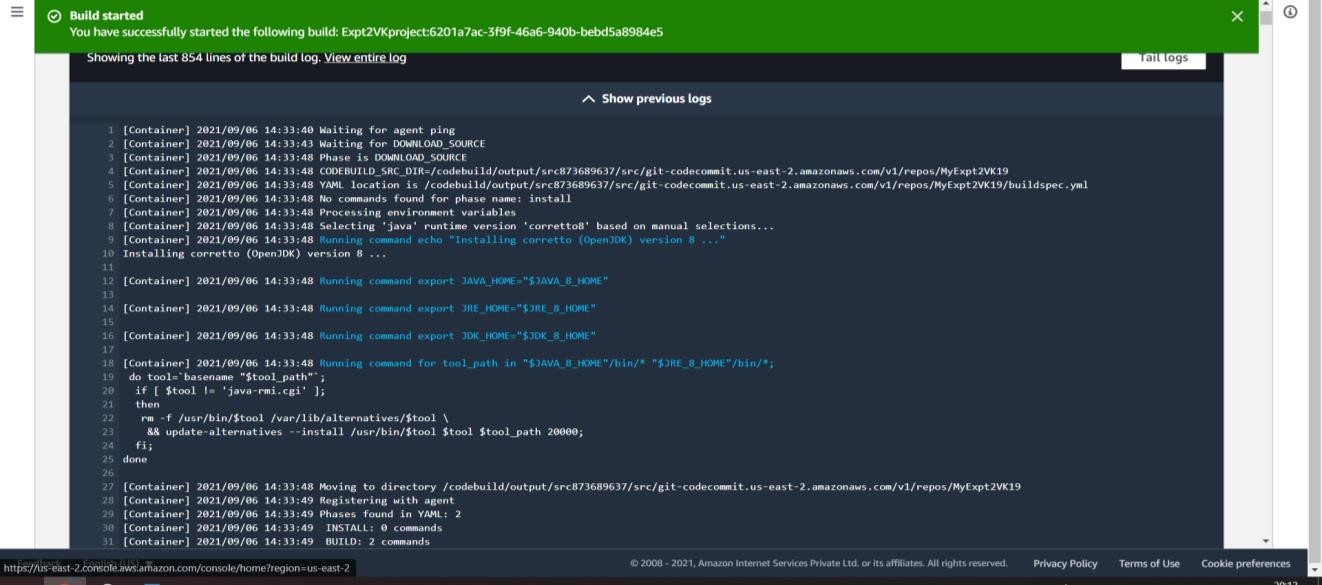
* + In starting the status will show In Progress if everything is correct the status will be Successful



* Building of my project in CodeBuild is now Successful



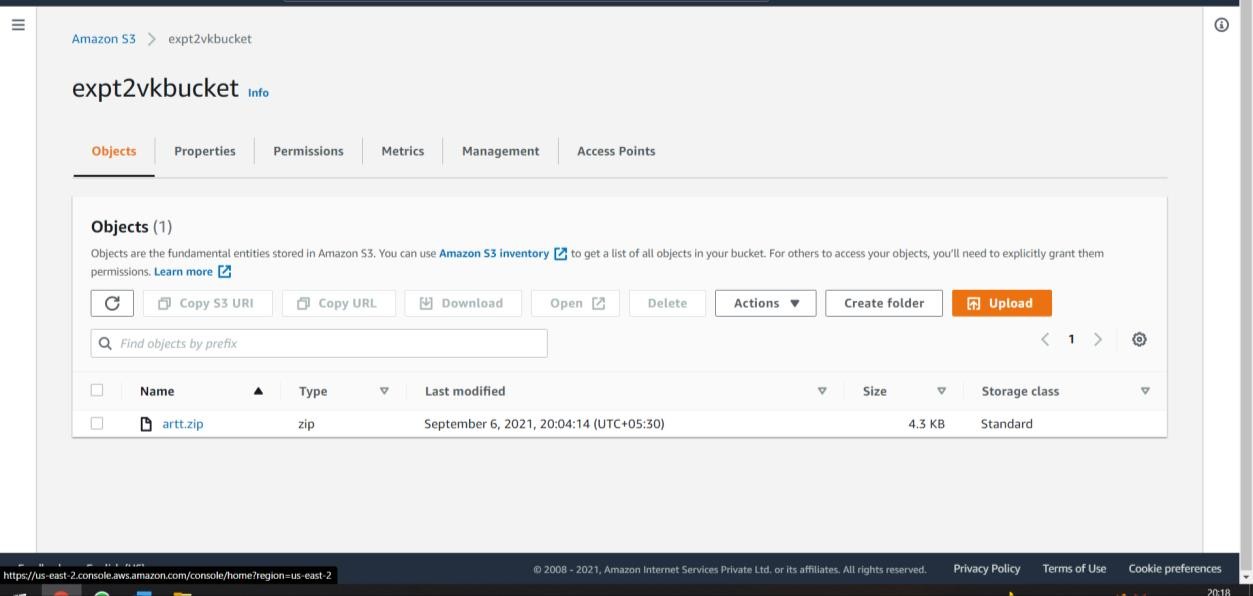
* If you want you can have a look on logs



### You can verify your status of project in codeBuild

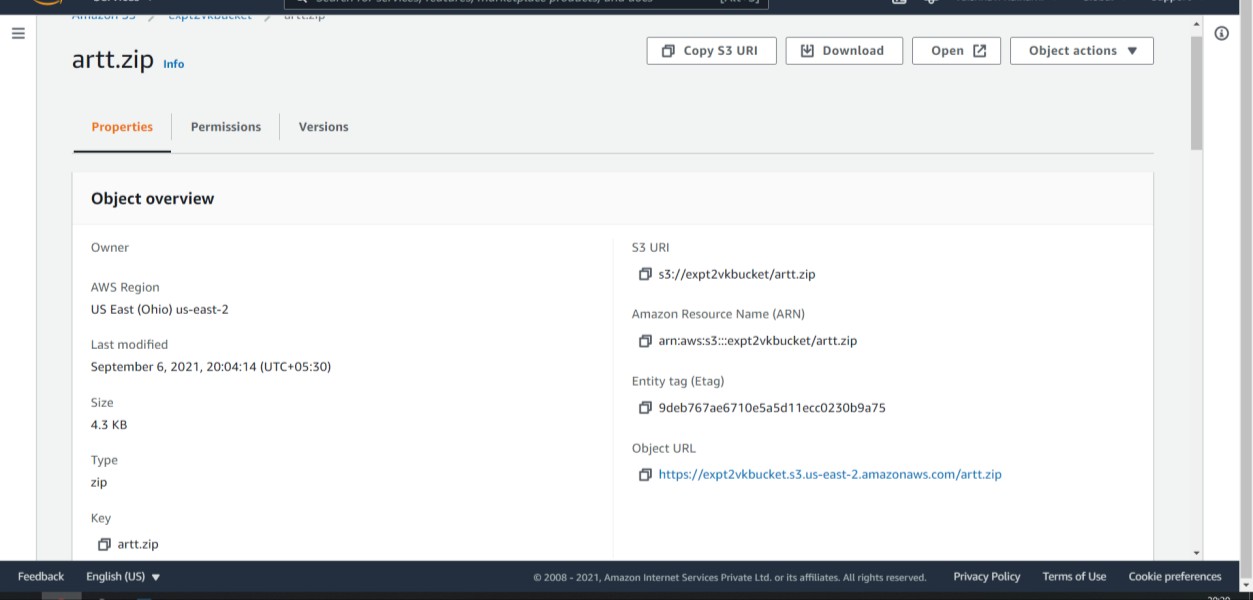
**STEP 12: NOW FOR FINAL STEP TO CHECK IF OUR OUTPUT FILE IS ACCESSIBLE OR NOT AFTER SUCCESSFUL BUILDING PROJECT IN CODEBUILT**

* Go to S3 Service and click on the bucket that you created for this project→You should see a zip file that you named while building your project in CodeBuilt Service



### artt.zip

* After clicking art.zip you can see entire logs i.e. output files are created successfully



### Zip folder is generated in S3 Bucket

**Conclusion:**

In the first part of this Experiment, we have successfully created a copy of a sample project i.e., we have successfully copied a repository from GIT SERVER to CODECOMMIT repository We can see all the files of GIT SERVER in our AWS CODECOMMIT repository.In the Second part of this Experiment, we have successfully added a YAML file on our existing project and we have compressed it in a zip file using CODEBUILD AWS SERVICE and we have created a bucket using S3 SERVICE to store that compressed zip file. In final output we can see the zip file which we created in CODEBUILD shall be accessible in our S3 BUCKET which we created.

**Lab Outcome: ITL504.1**

To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements.