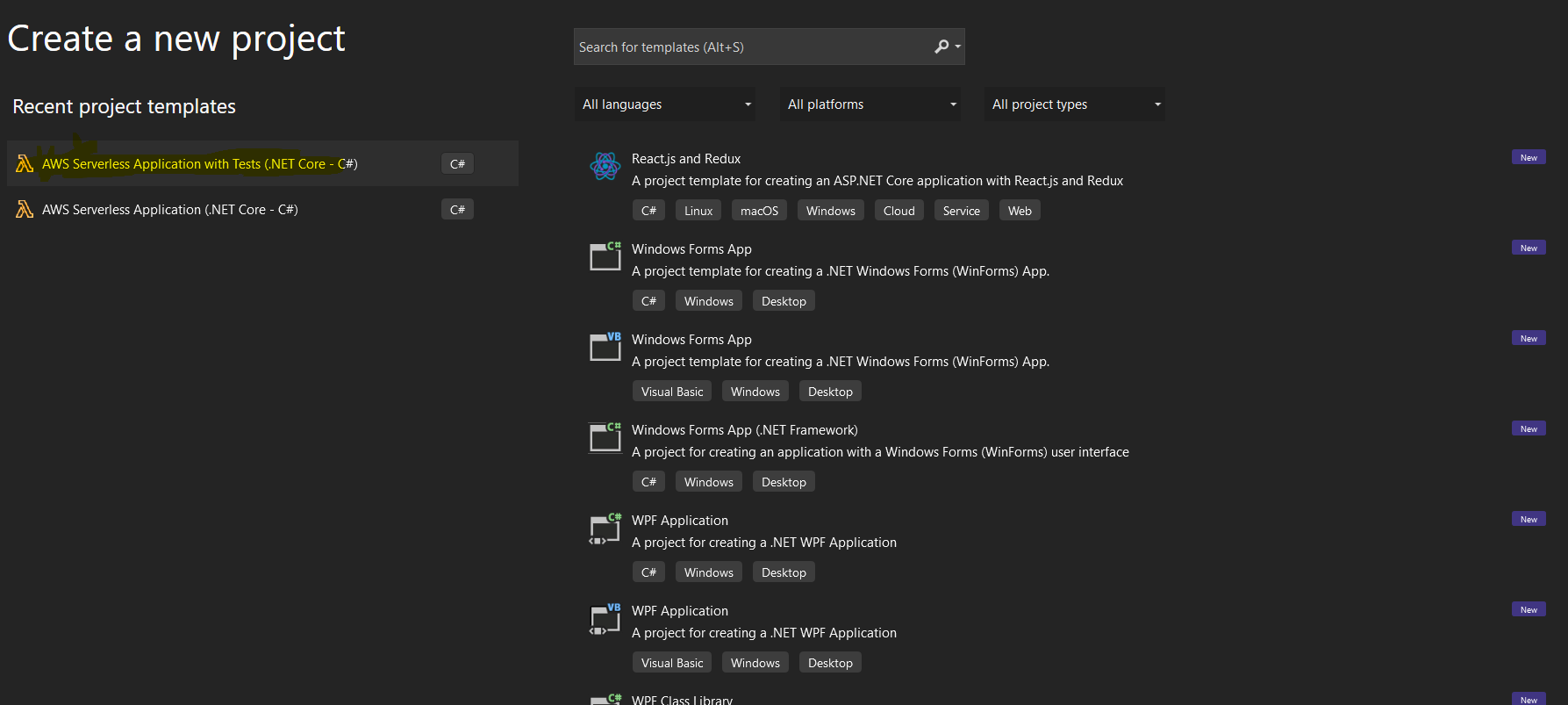
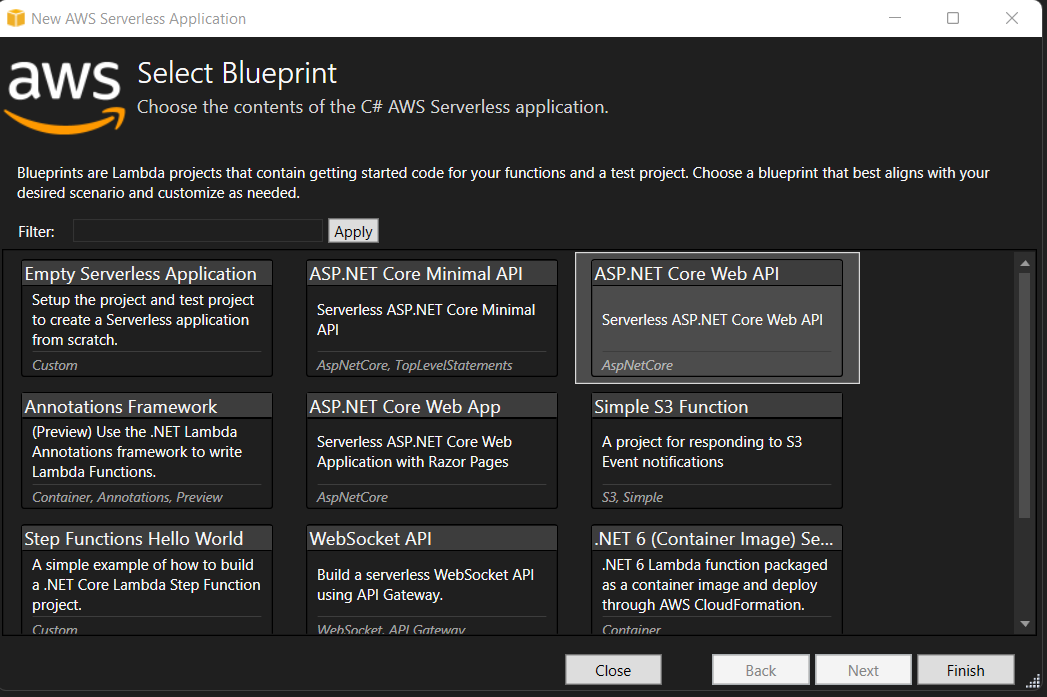
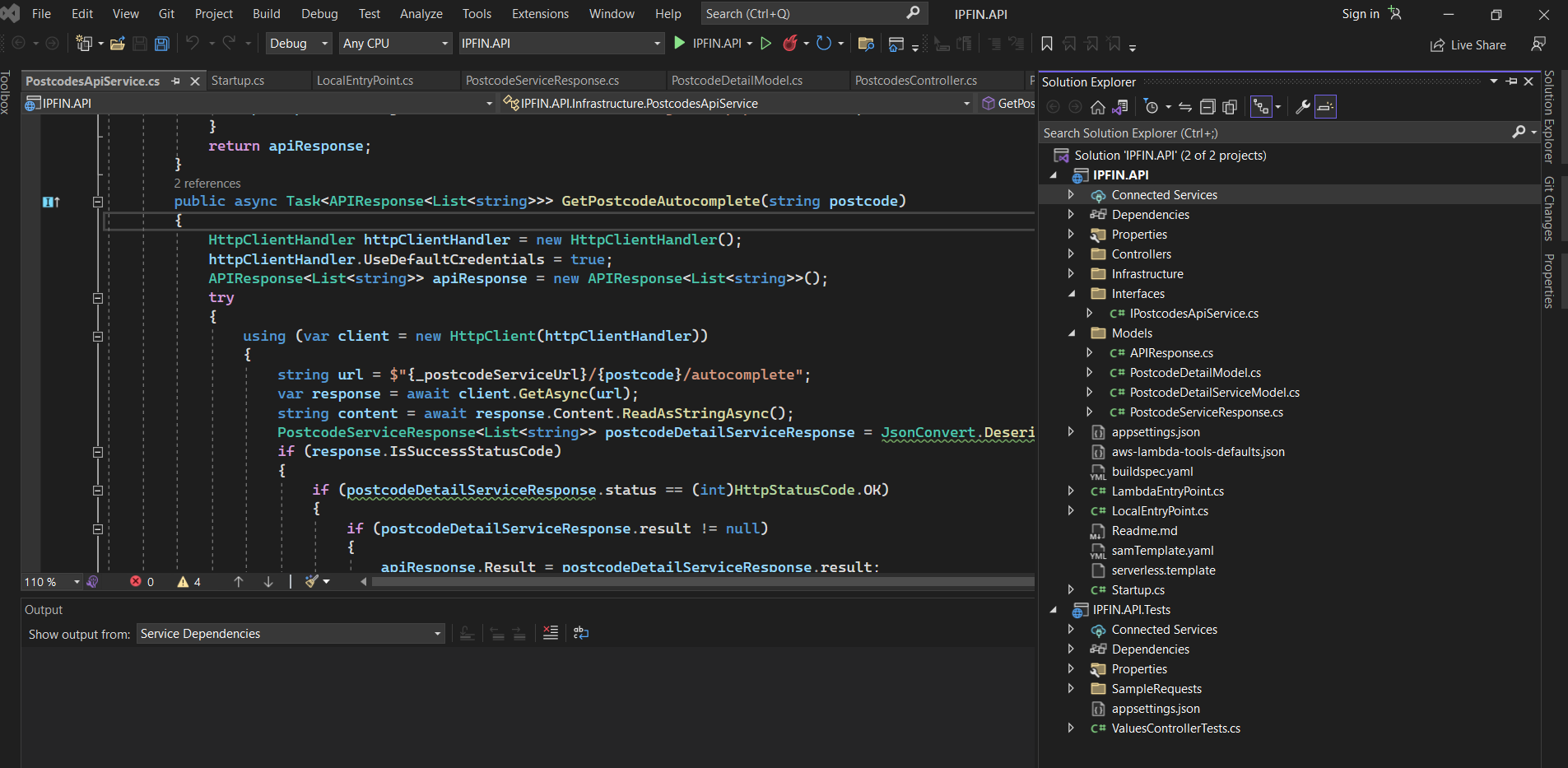
1. Getting started with AWS SAM Lambda
2. Create AWS serverless lambda application using visual studio.



1. Select ASP.NET Core Web API option for Blue print.





1. Create deployment package

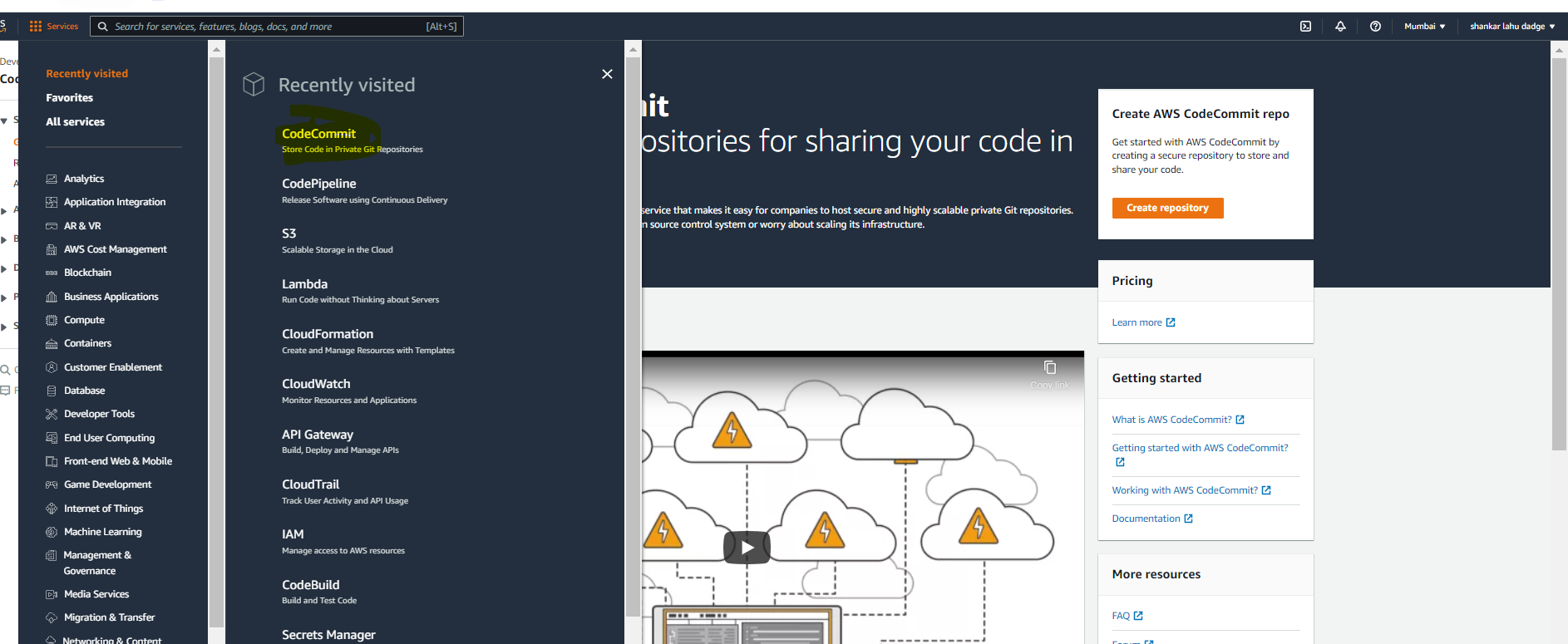
* Open command prompt, go to directory path
* Use command – dotnet build
* Then use dotnet publish command for create deployment - dotnet publish

1. We can also dploy code using serverless template throgh AWS CLI
2. Deploy code on AWS as serverless need to create AWS Pipeline

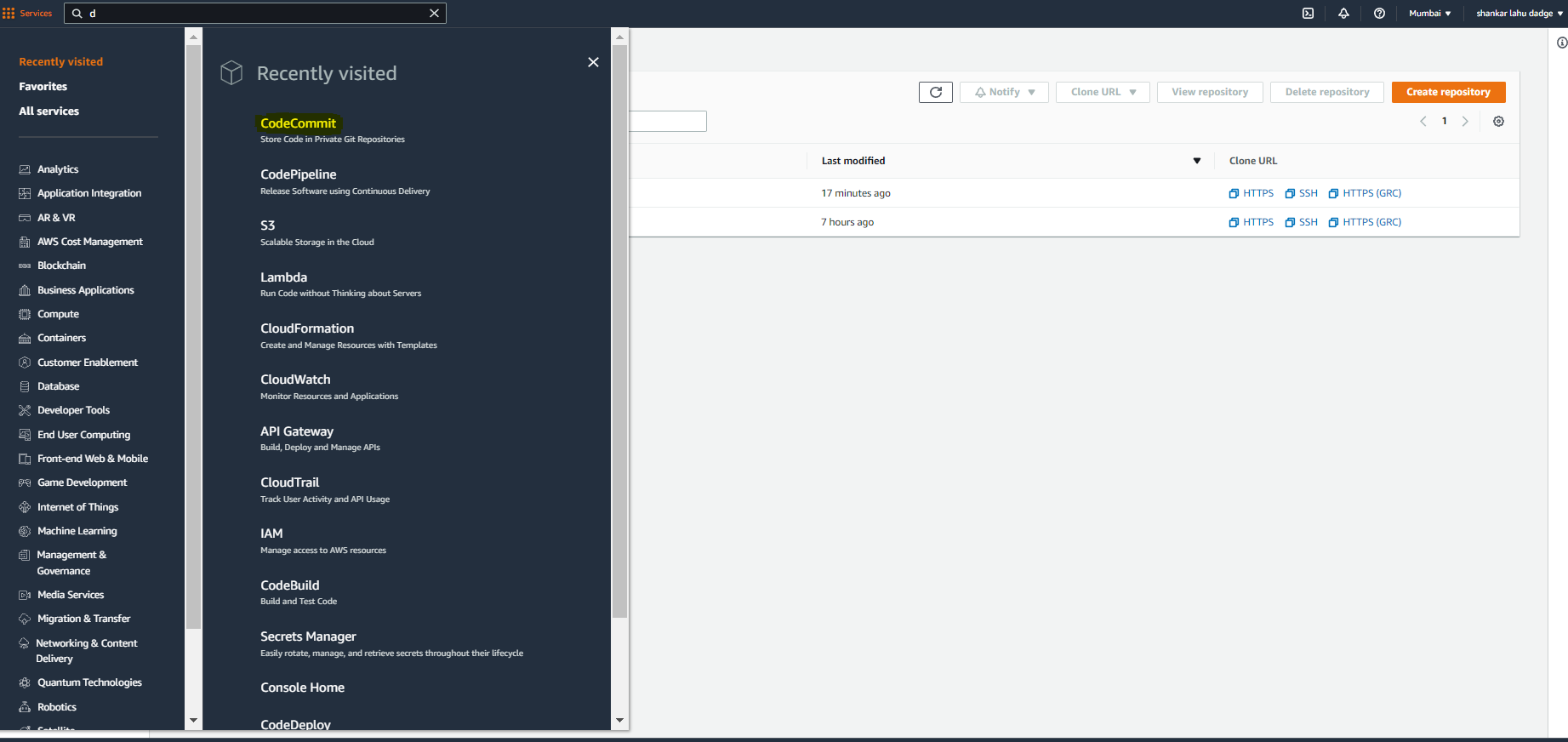
First need to create repository in AWS Code Commit service.

Steps :

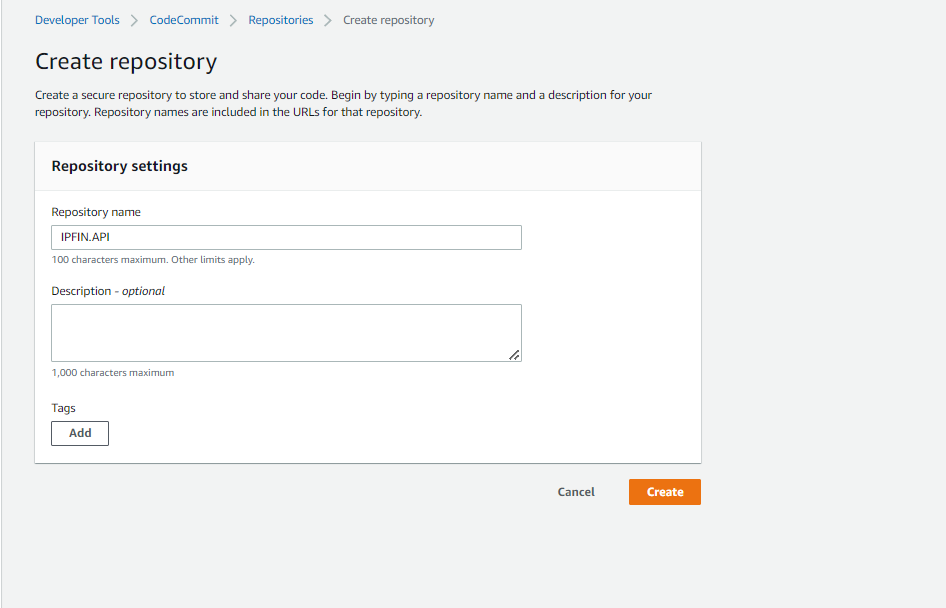
1. Login to aws console using IAM/Root user login, and select code commit service.



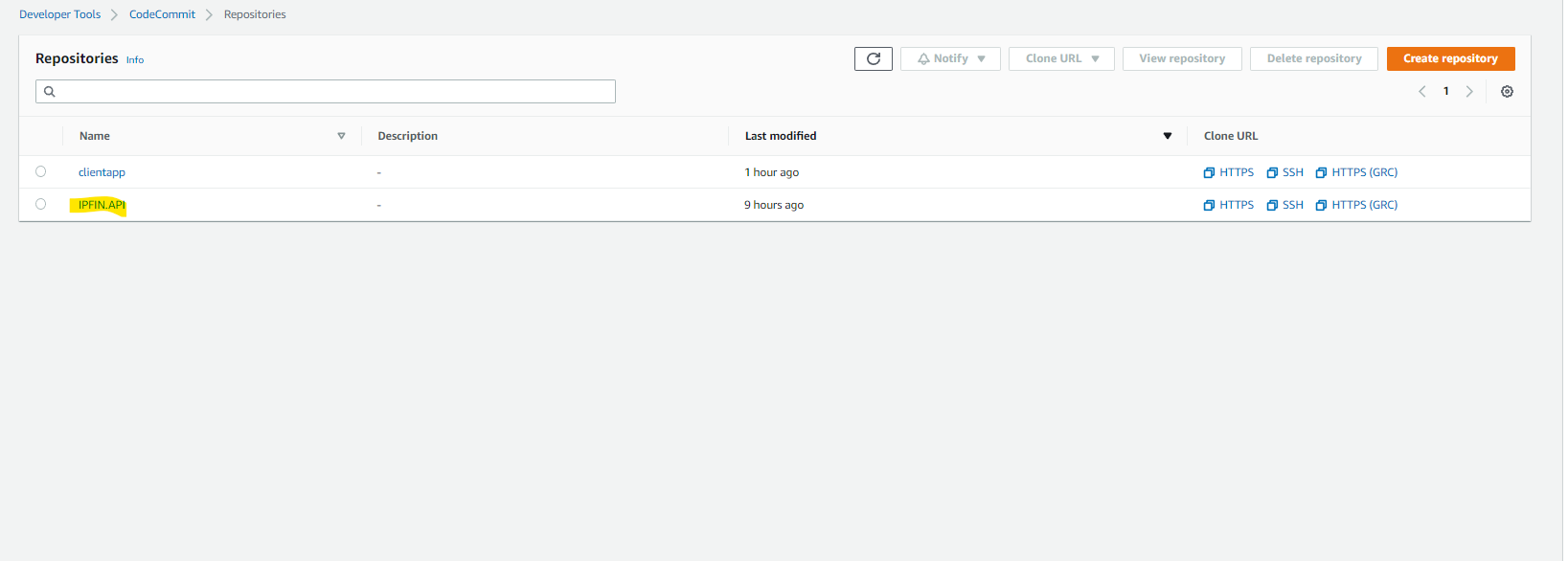
1. Need to select Create Repository option here



1. On click of create repository option, it will open create window. Here we can fill Repository name for our Service name and othe info is optional.

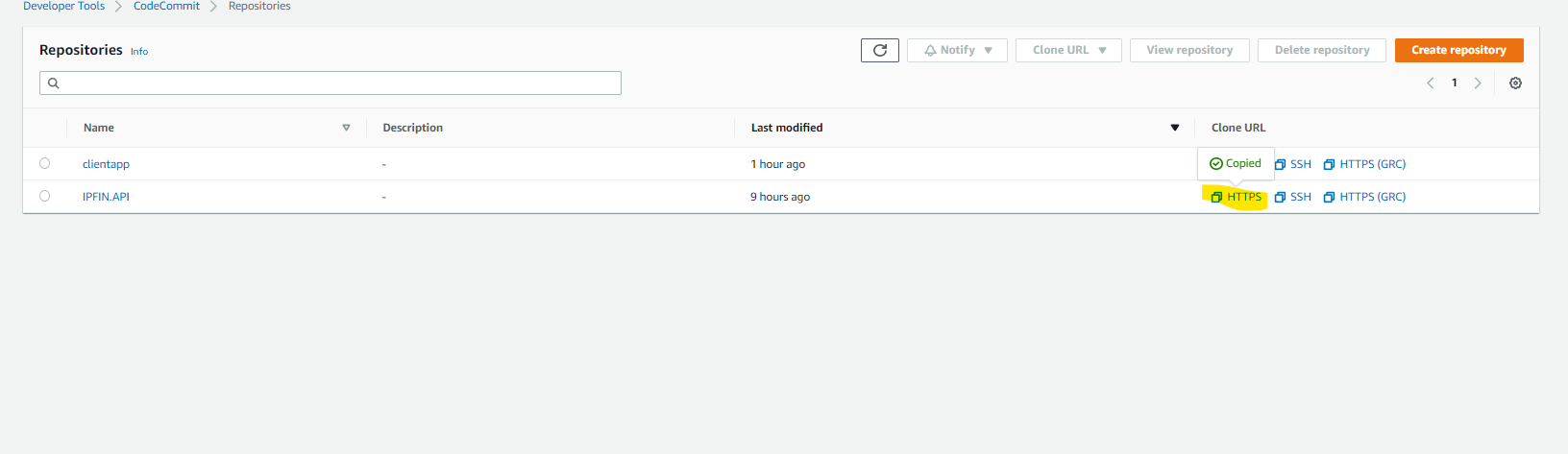


1. It will create source repository on AWS Code Commit

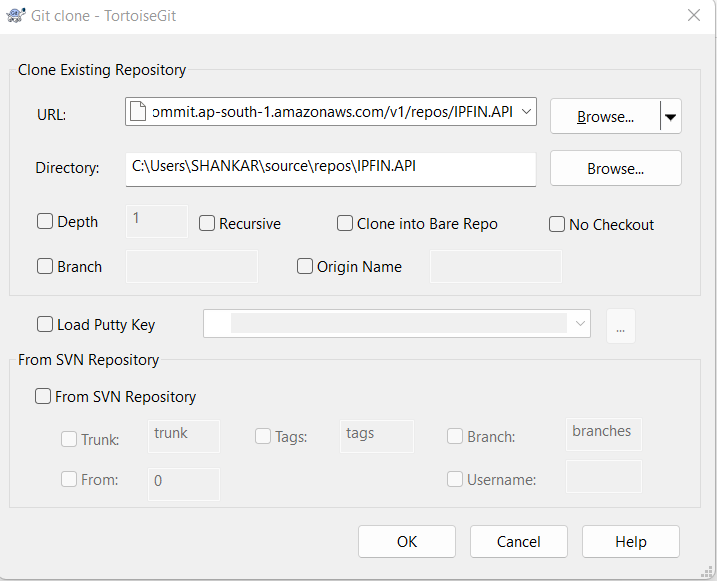


1. We can clone the repository in our local machine using Clone URL and CodeCommit User Credentials.

By clicking on https clone option against the repo we can copy the URL



1. Using git on local machine we can clone the repository.



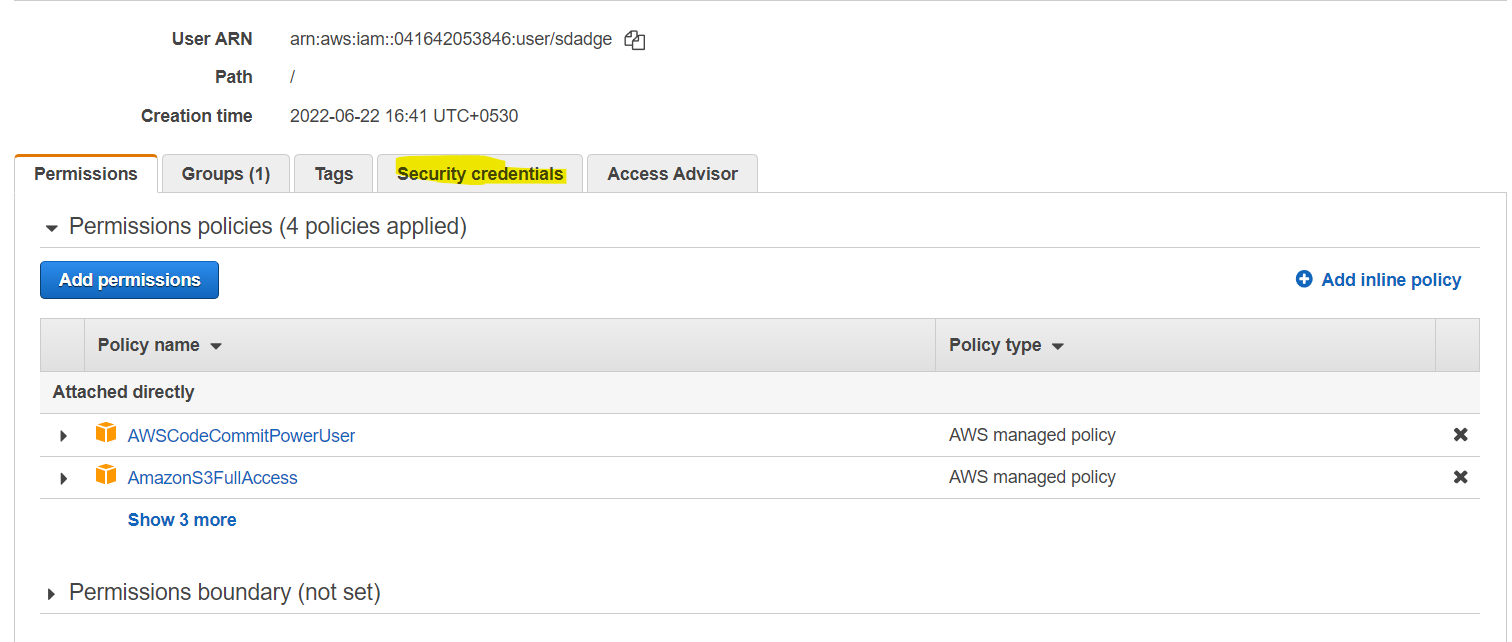
7.Creadentials for Code Commit repo checkout/commit need to create against the IAM user.

i) Open IAM aws service

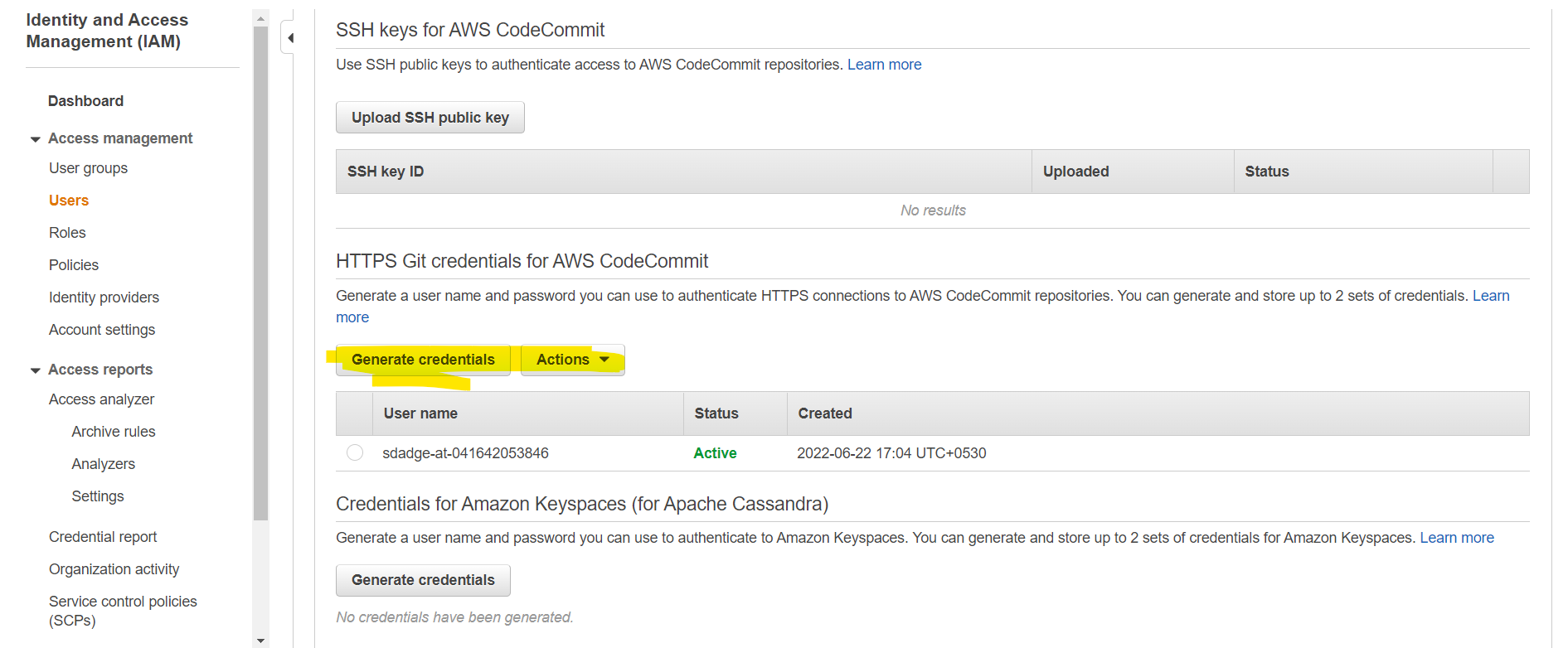
ii) Select users option

iii) Select IAM user

iv)Select security credentials option



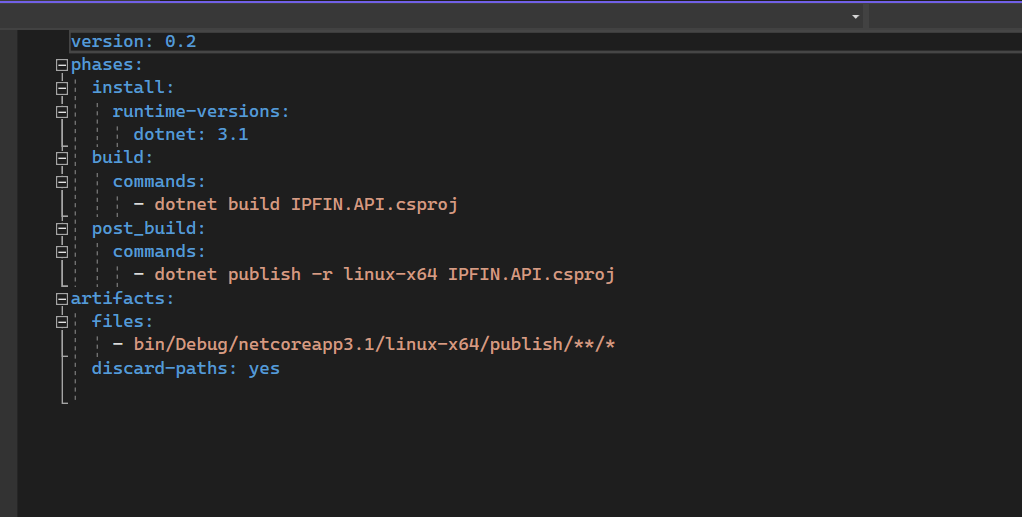
v) Genrate code commit git credentils using Genrate Credentails option



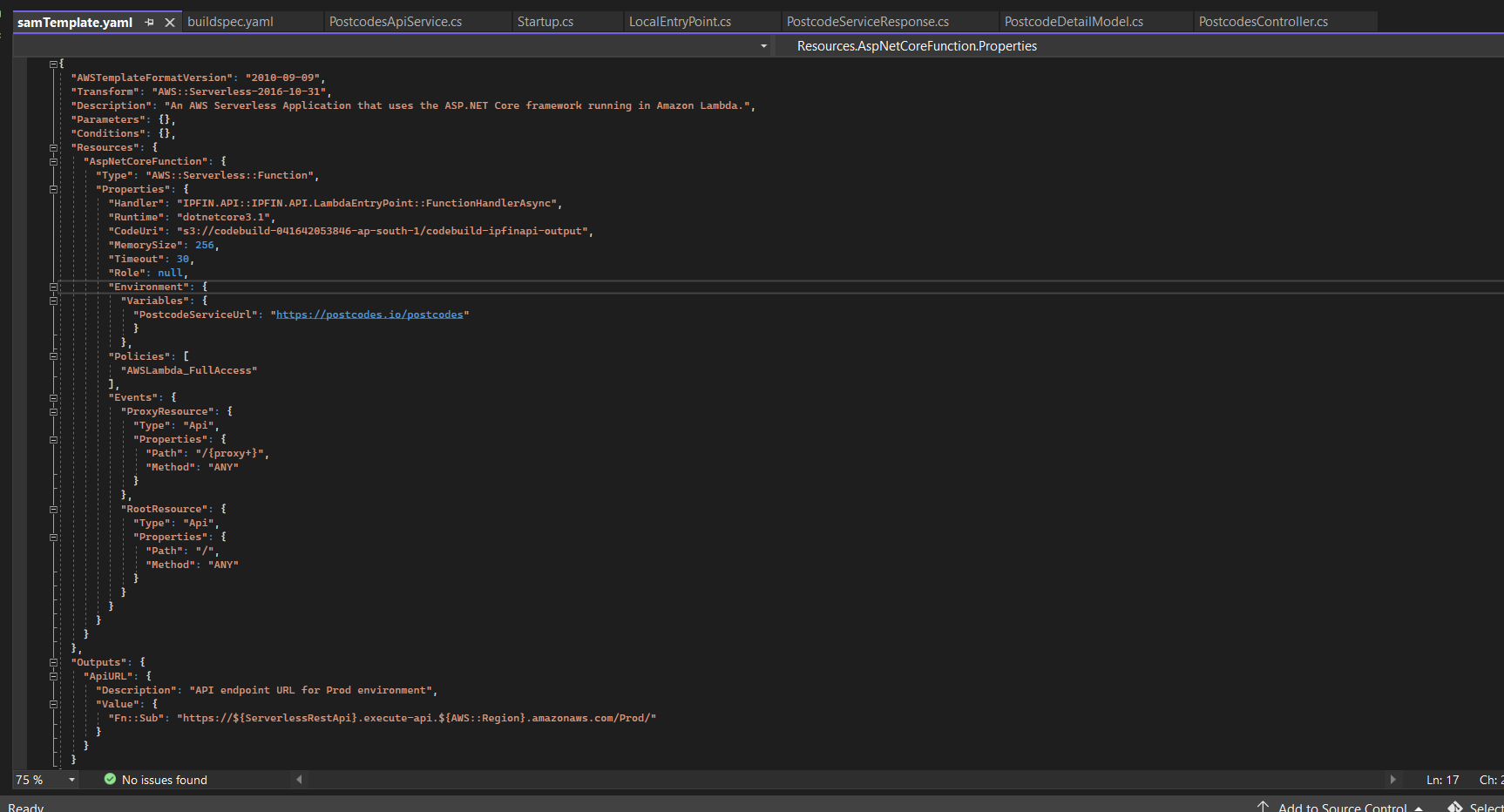
vi) Using thease credentails can be checkout and commit our Lambda.

8) To build our application on code pipeline and deploy on AWS, need to prepare buildspec.yml and need to commit in root directory of our project.

It will contaons basic prebuild, build and post build commands, artfact, stage and many more conformation.

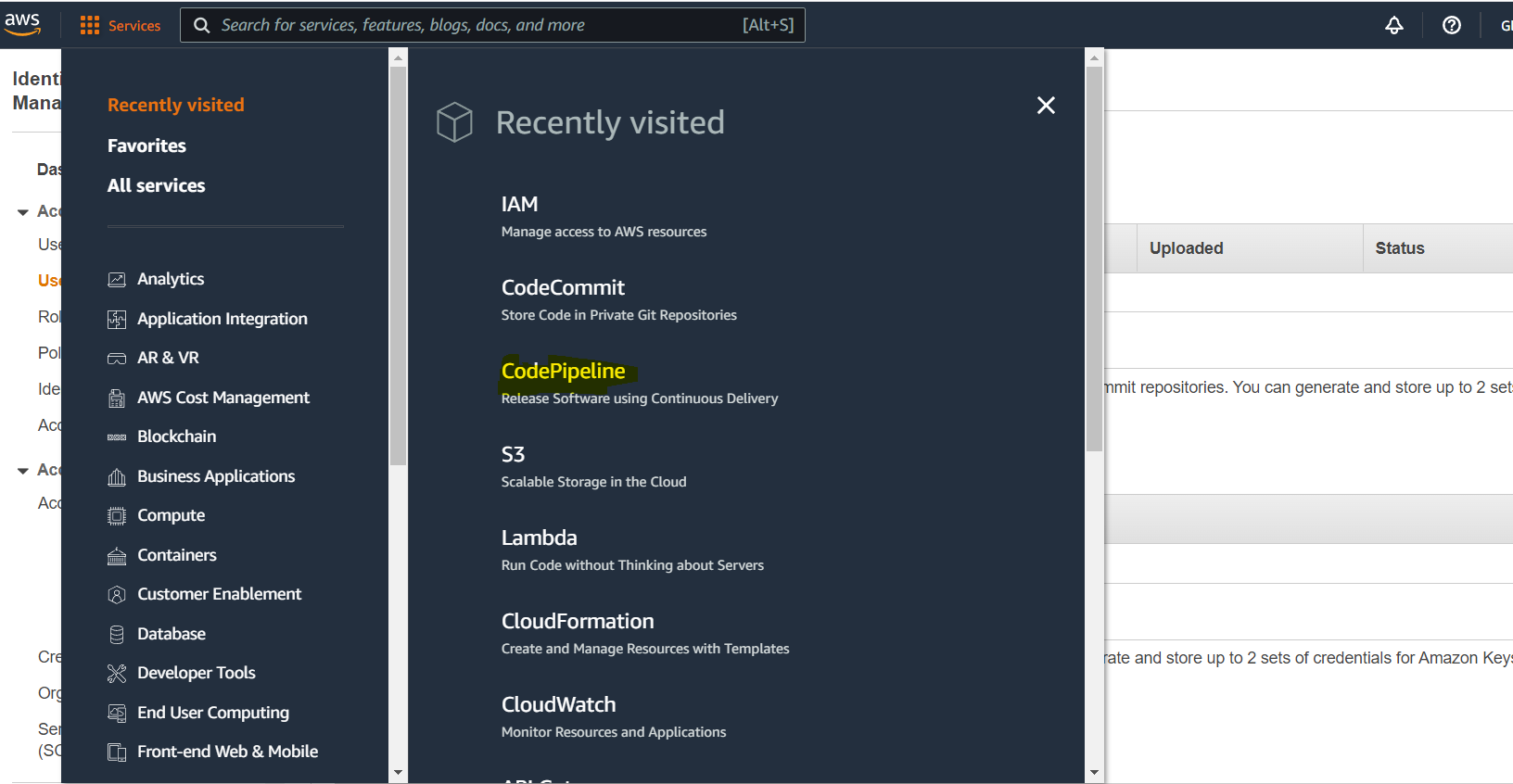


9) Prepare Cloud Formation template for creating AWS Lambda and API Gateway resource on AWS .

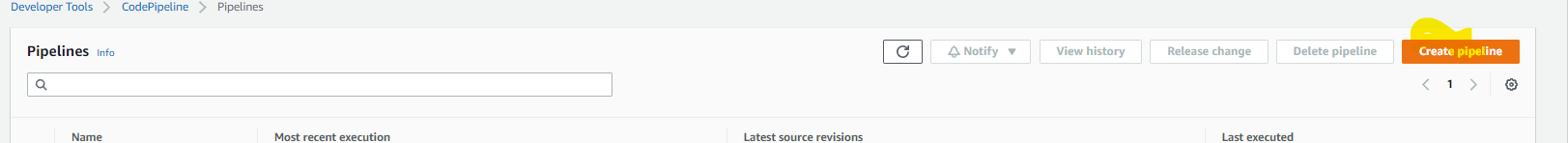


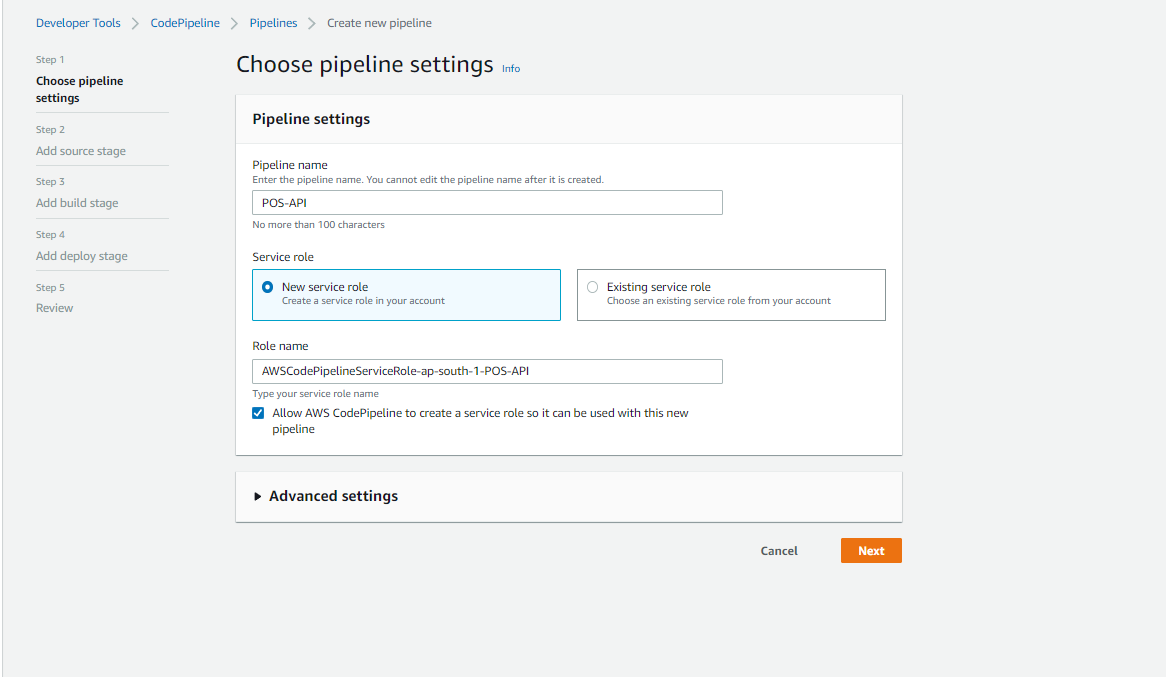
9) Now to deploy application through CICD, we need to create AWS Pipeline

i) Select Pipeline service

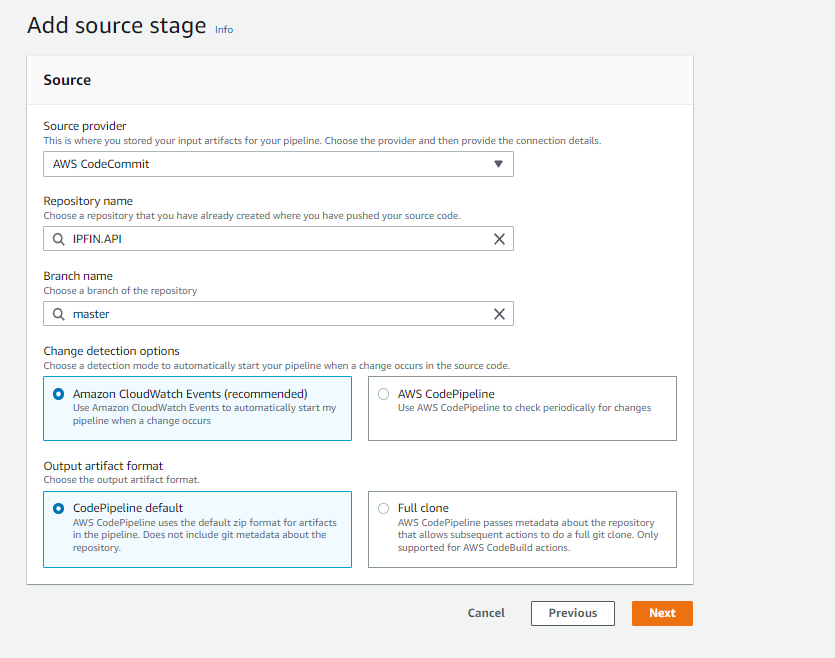


ii) On click of CodePipeline ite will open new window where developer can select Create New Pipeline option.

ii)It will open new window where nedd to fill Pipelinename, and need to select serivce role.

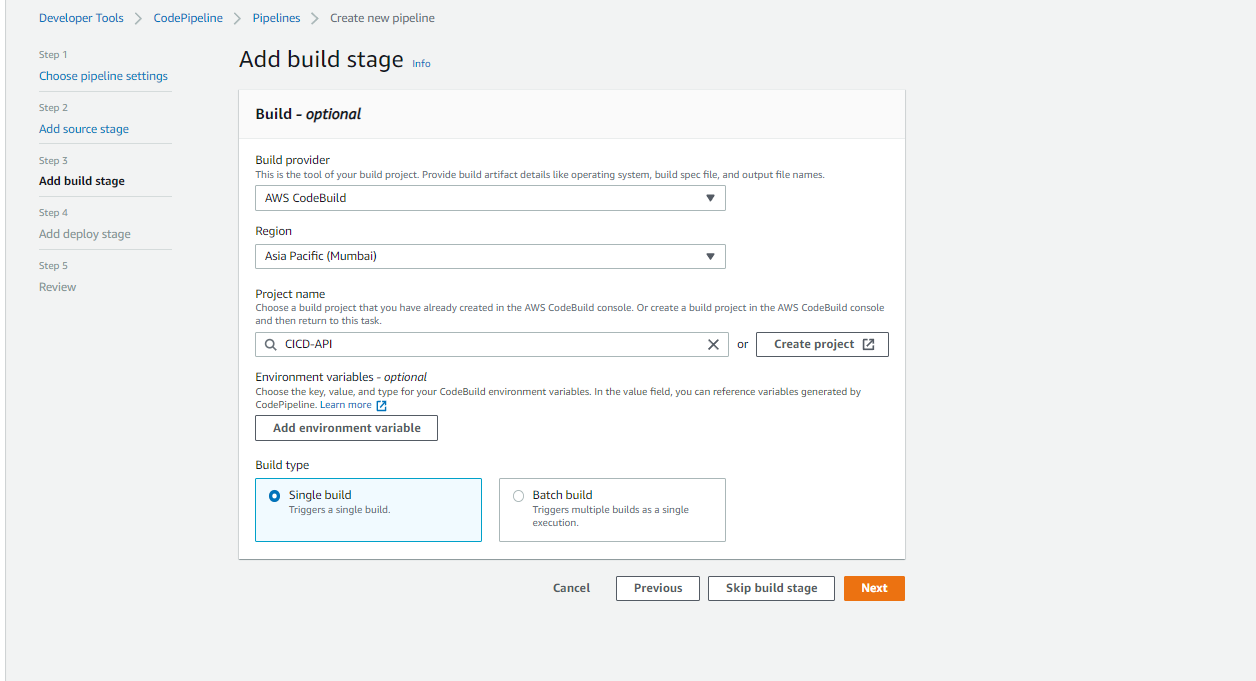


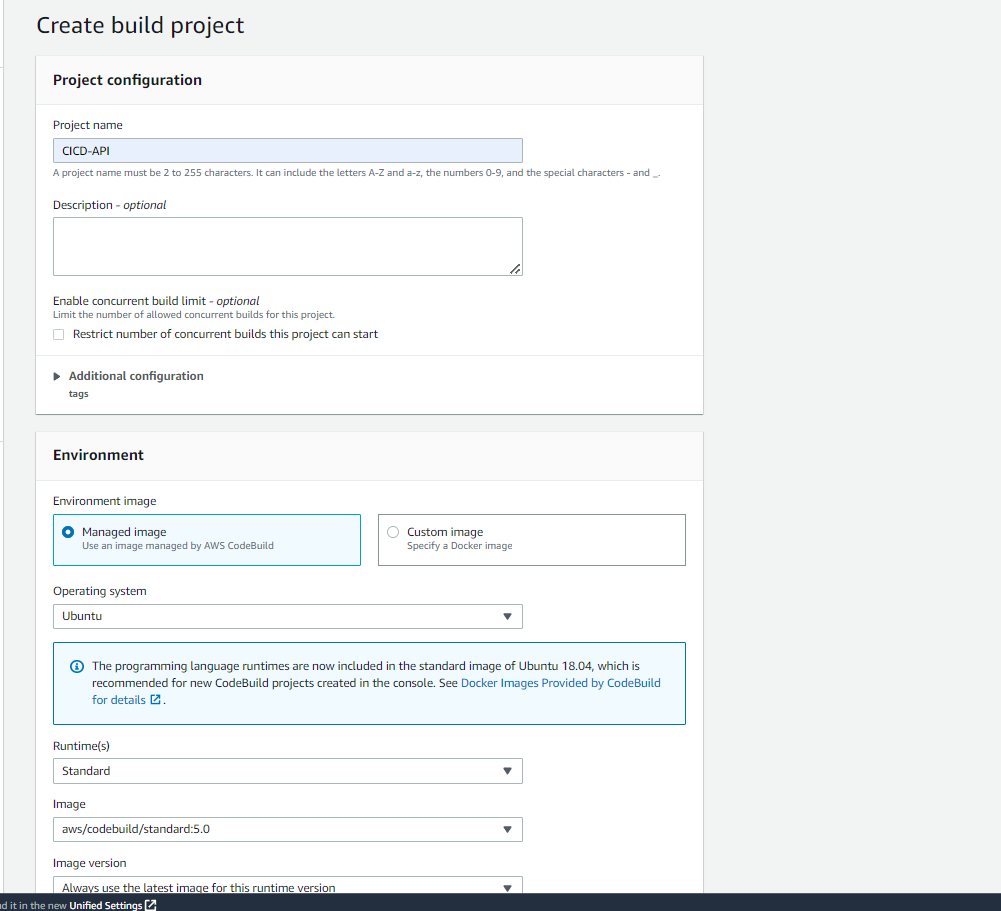
ii) click on next button and Fill source stage information, this will our Code Commit Repo information. Select appropriate repo and branch. Change detection option is pipeline triggering point. If any change commited against the repo it would be trigger pipeline.



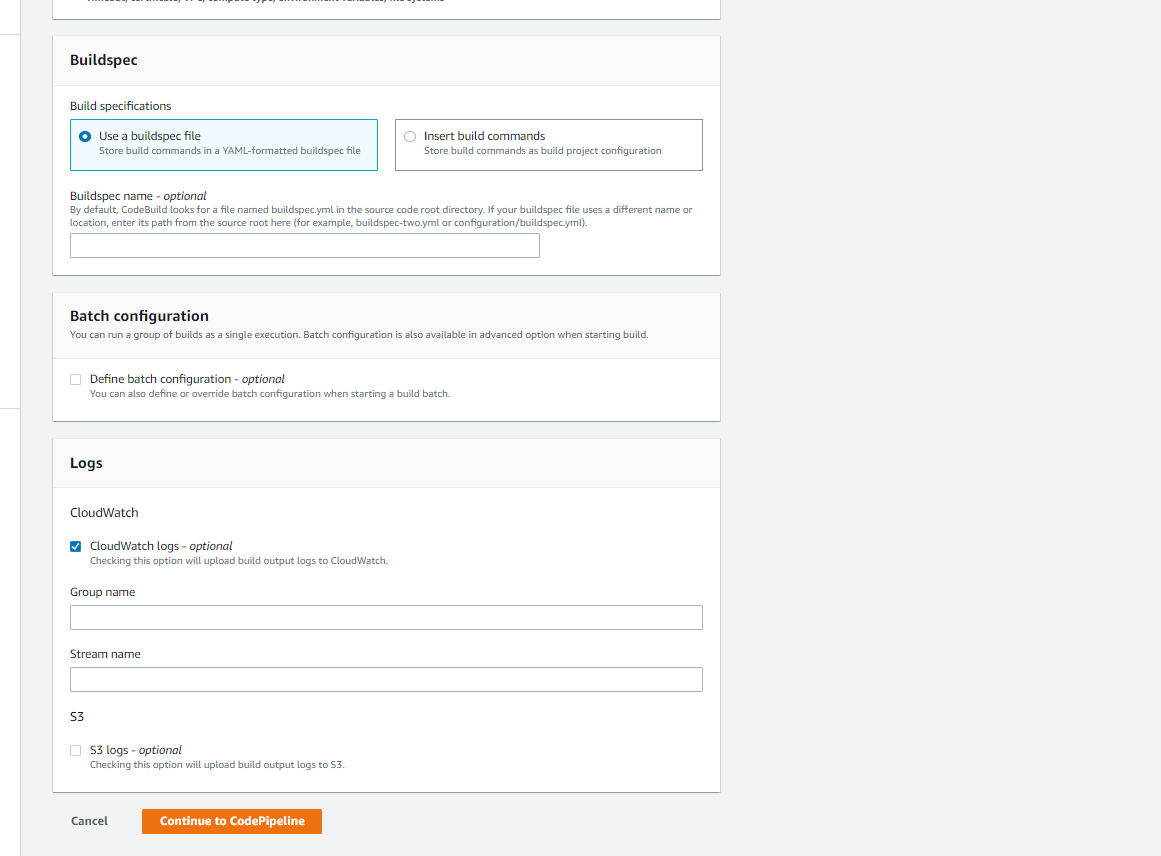
iii) Then we need to fill build stage information.

Here we need to select build proivder, region and project. If project is not there then need to select project option.



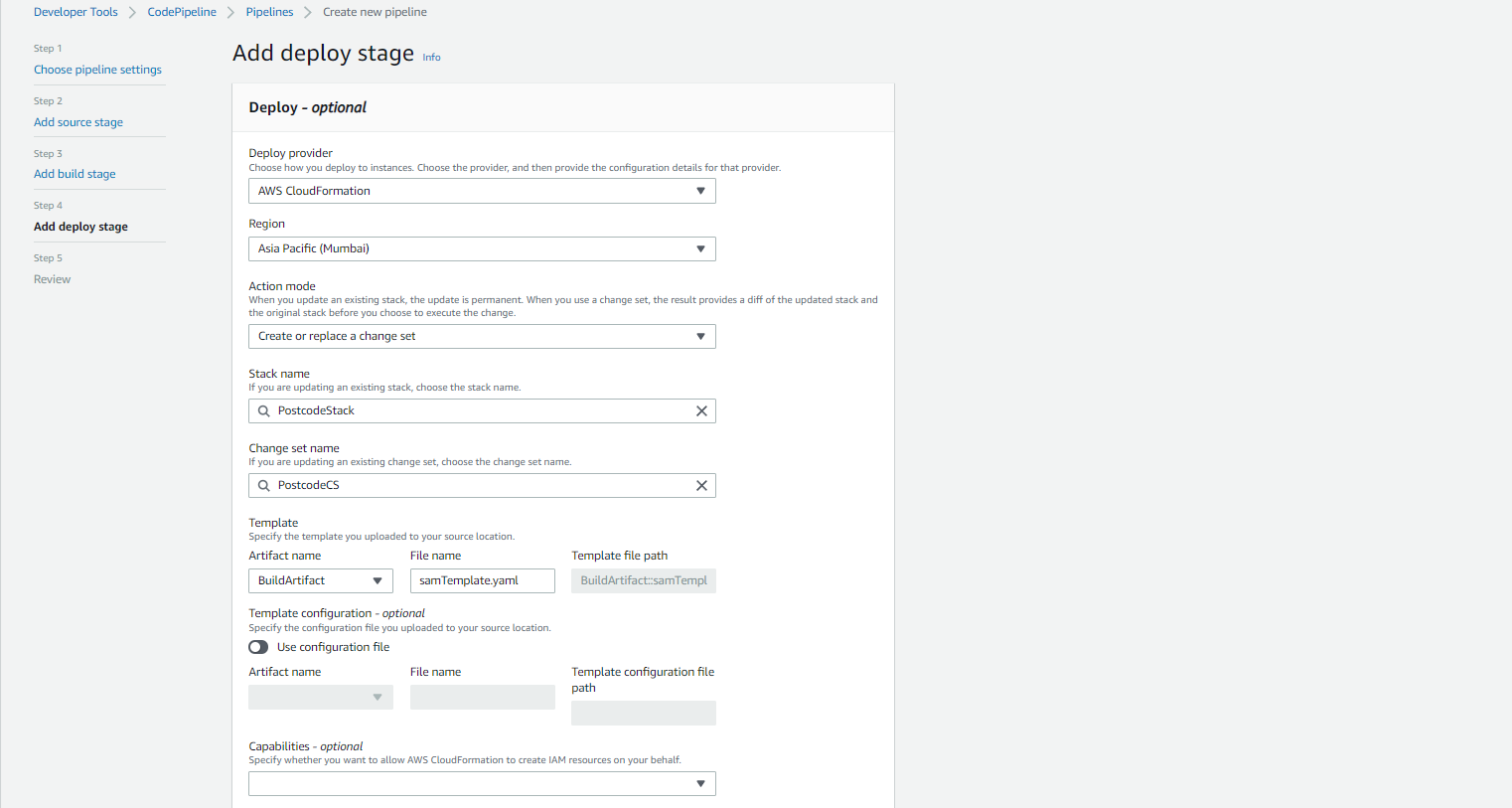
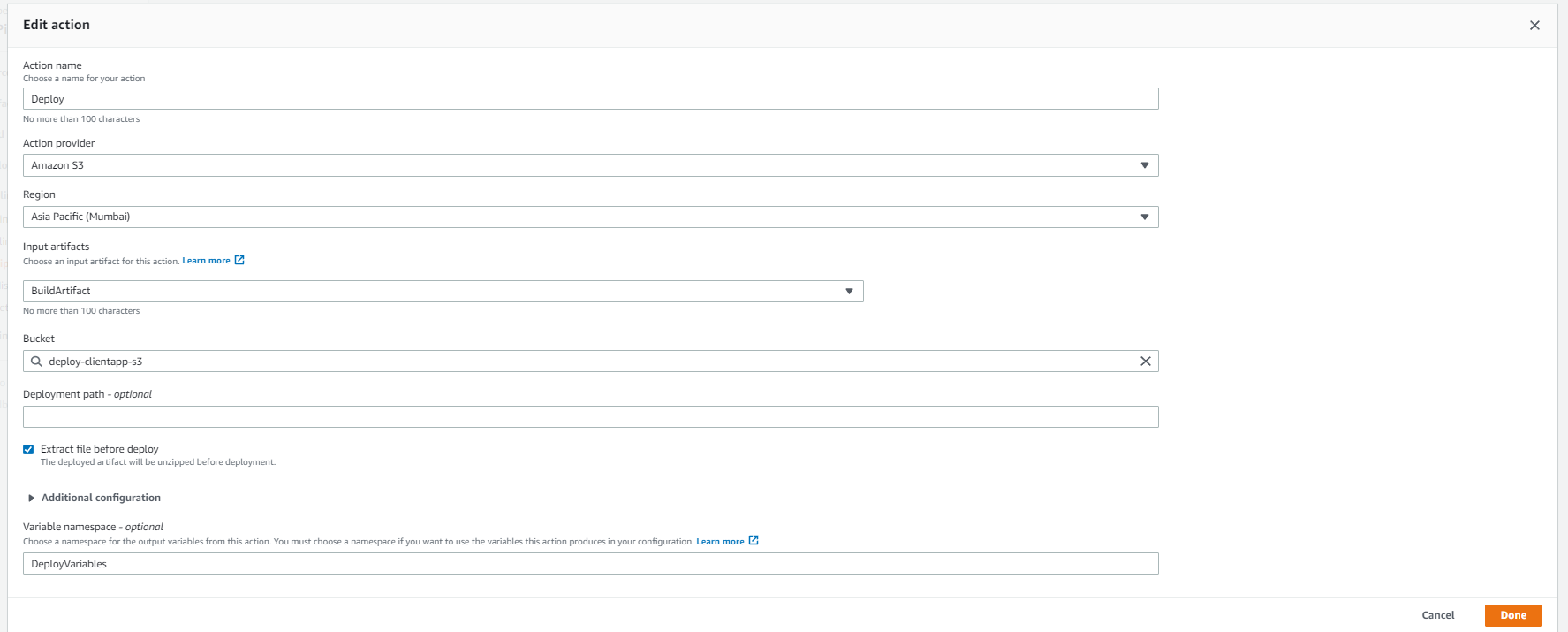
iv) While creating project of build stage, we need to provide project name, operating system, Image information, runtime, build spec and Log source. 

v) we have already Build Spect file in the root directory so select buildspec file option



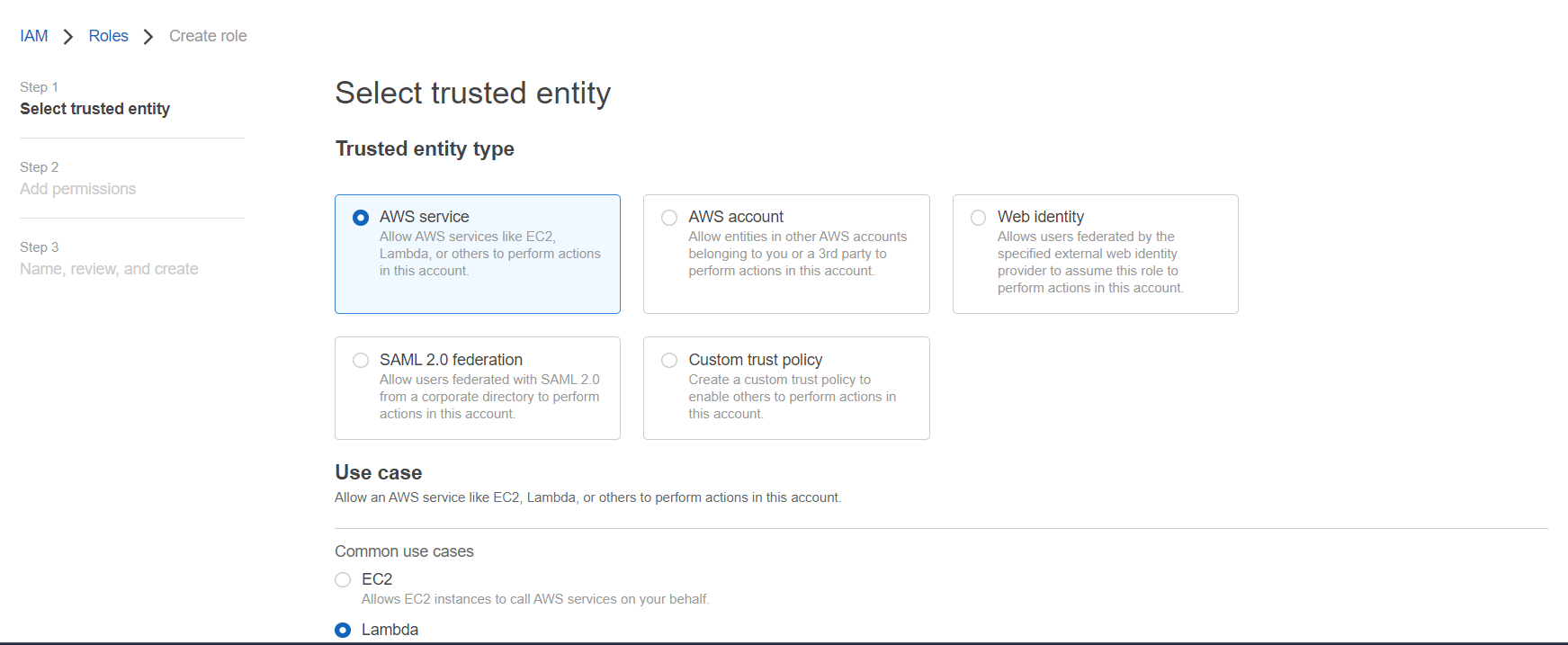
v) Add the deploy stage, which will deploy build aws stack, it will create AWS lambda and API Gateway endpoint.

Here we need to specify the Region, Action Mode, Stack Name, Build Artifact information and Cloud Formation Template file name.

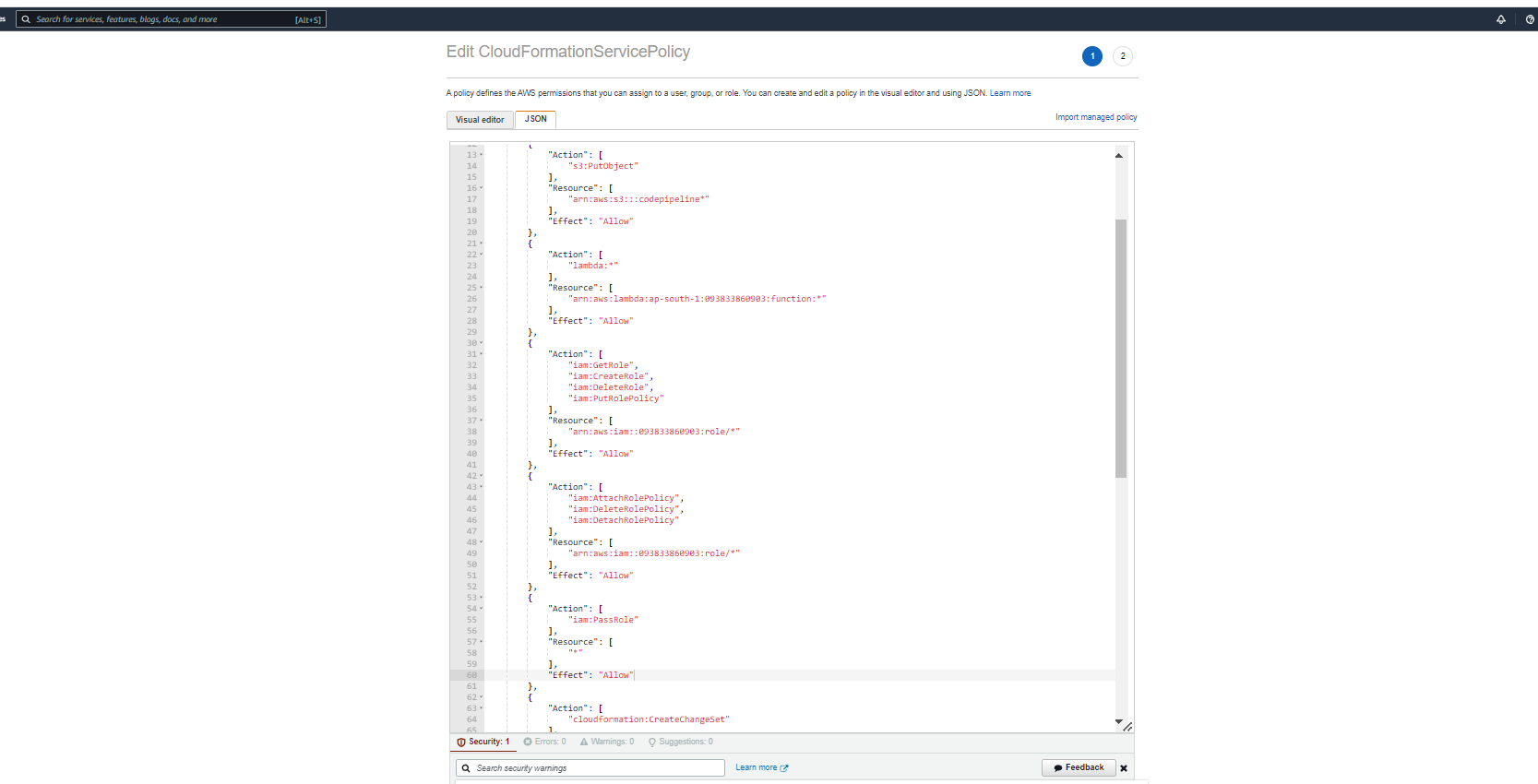
 

To deploy Lambda and API Gatway through the Cloud Formation Template, then need to create IAM role for this.

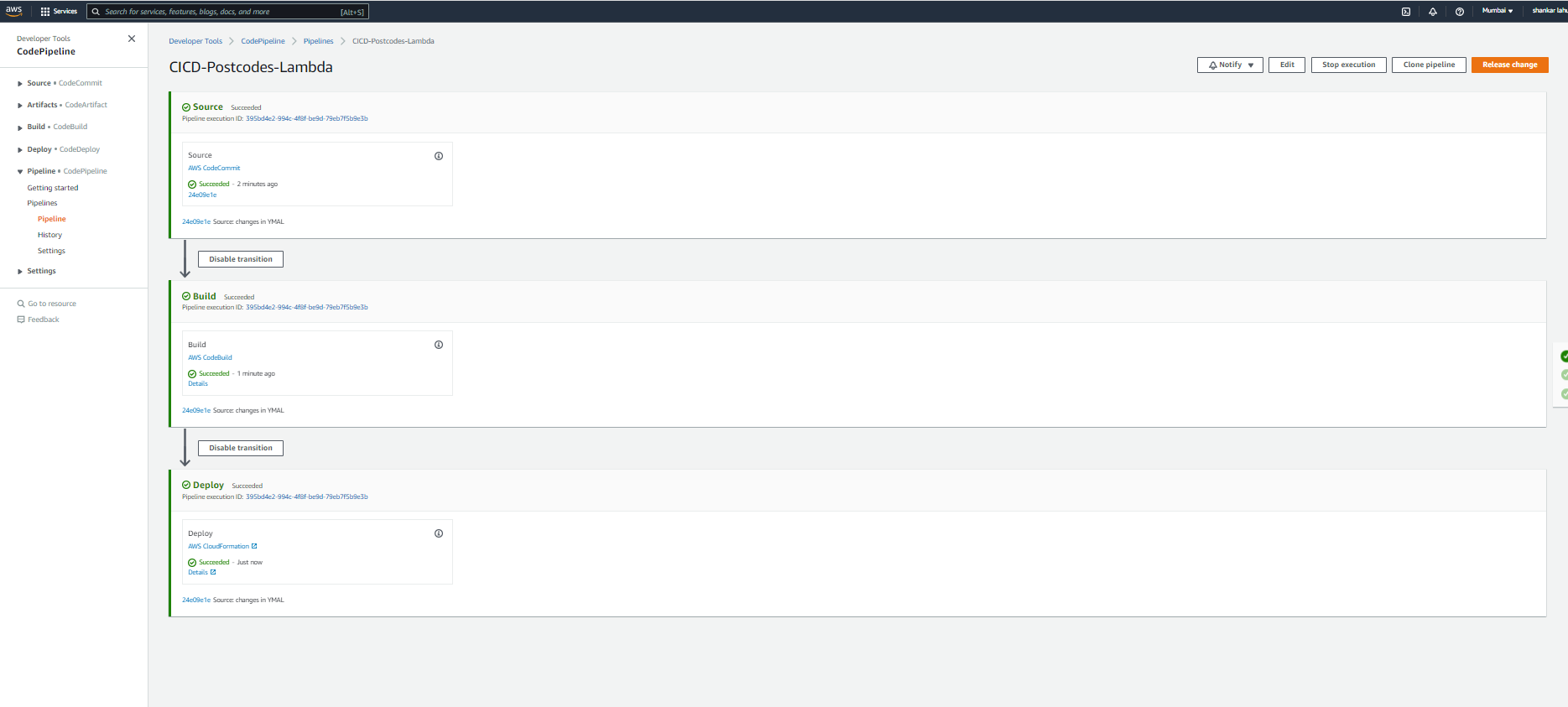
* Go in IAM and select Roles.
* Click on add new role



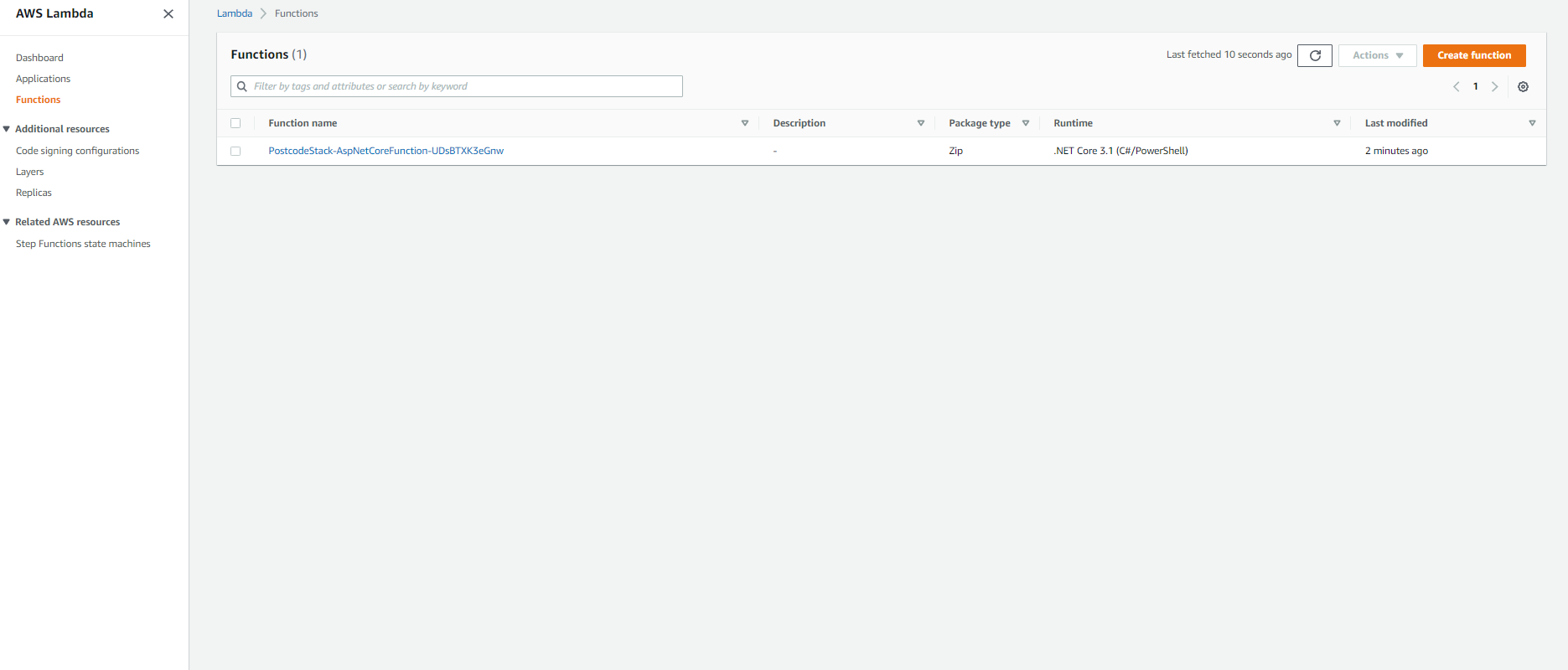
* Then add inline policy to access S3, Lambda, API Gatway and Stacks.

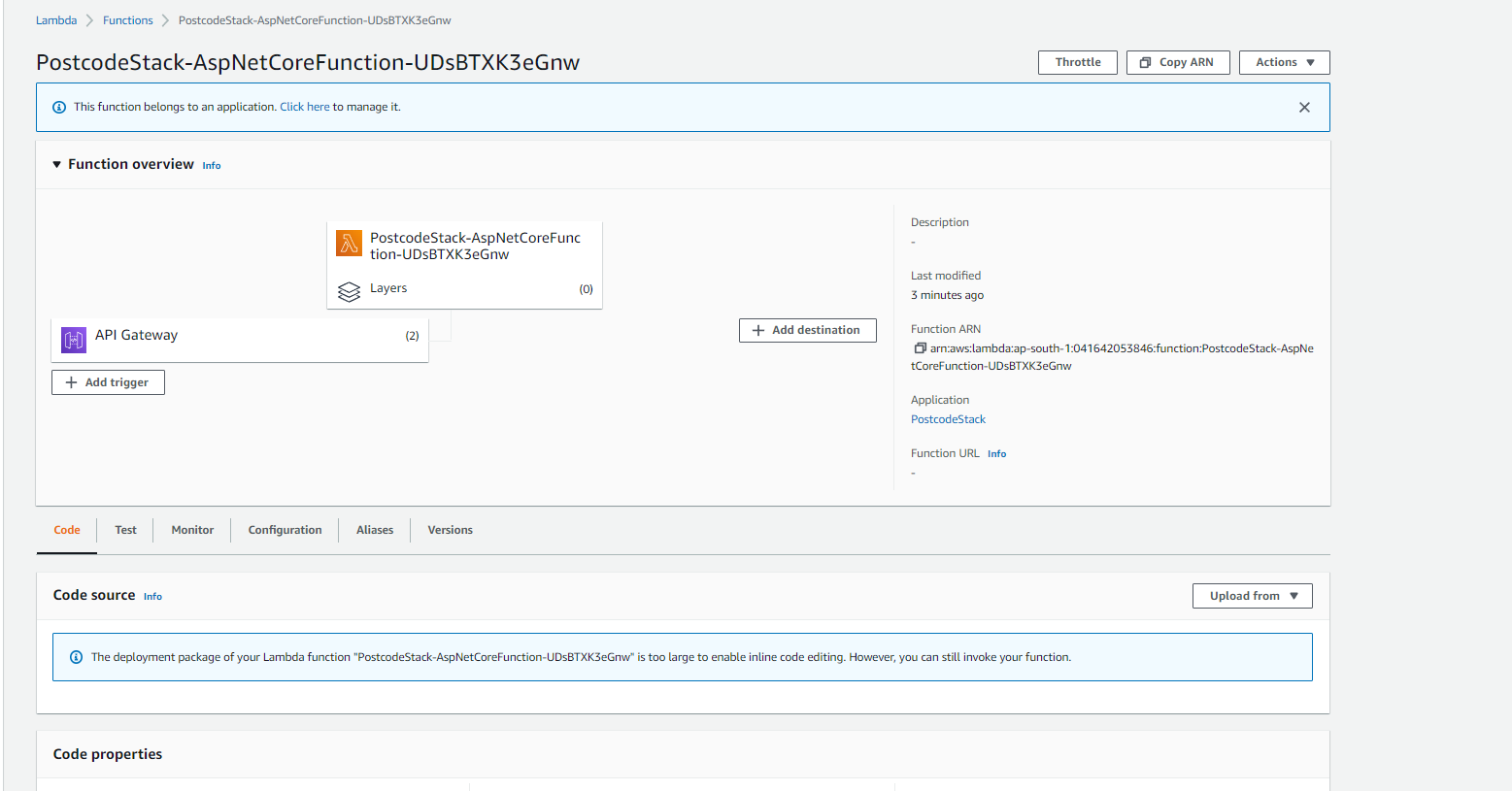


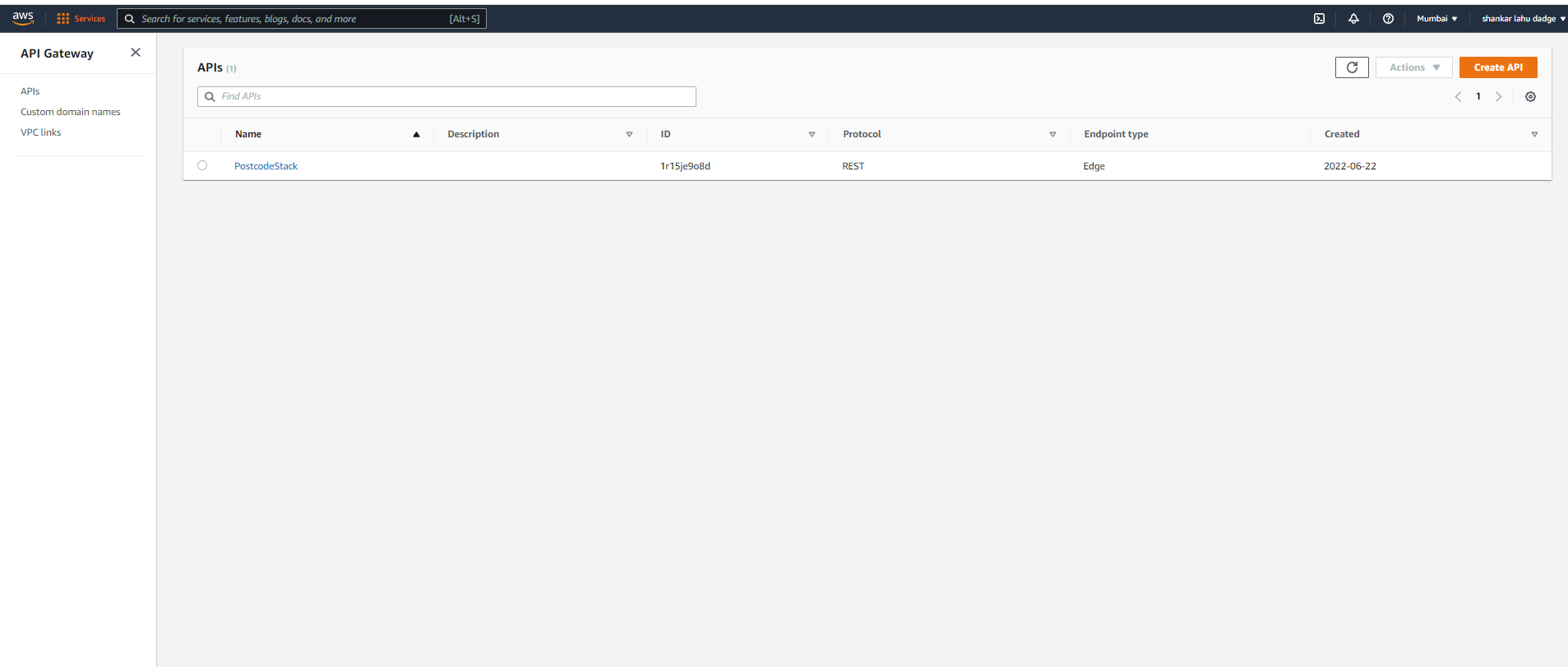
vi) Now our pipeline is ready , It will trigger on code commit or manualy.

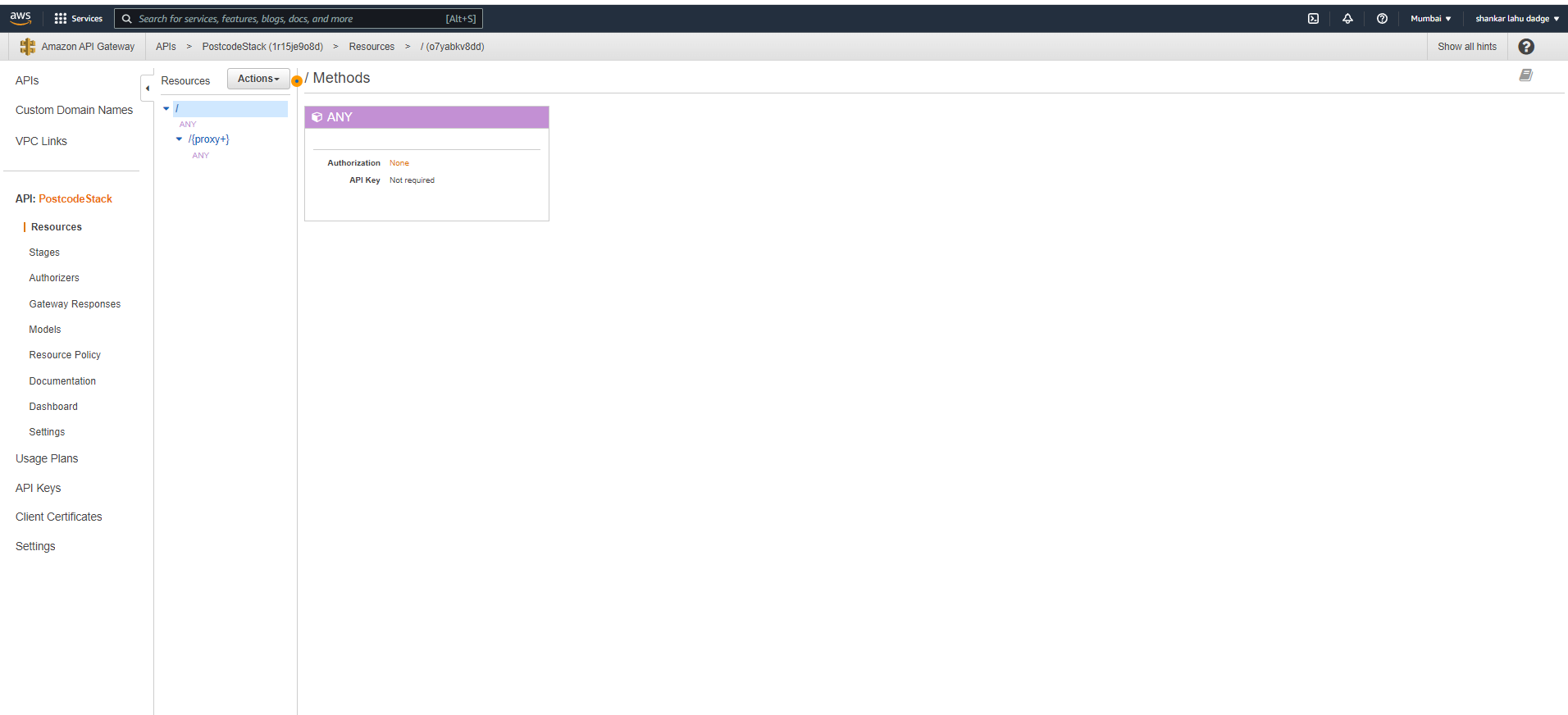


vii) After all stages run successfully, it will create AWS Lambda and API Gateway









viii) we can check the API endpoints using Postman.

