Project Name : Aws Cloud based Calculator App Continuous Integration Pipeline

Description : we will deploy sample calculator app on AWS cloud using code-commit & through buildspec.yml & appsepc.yml file we will build our code using aws codebuild and using code deploy will deploy our calculator application using CI /CD pipeline on ec2 instance.

What is buildspec.yml file ?

It is a configuration file used by AWS CodeBuild to define the build and deployment steps for a project. It provides a set of instructions for CodeBuild to follow during the build and deployment process, including dependencies, run tests, and package the application for deployment.

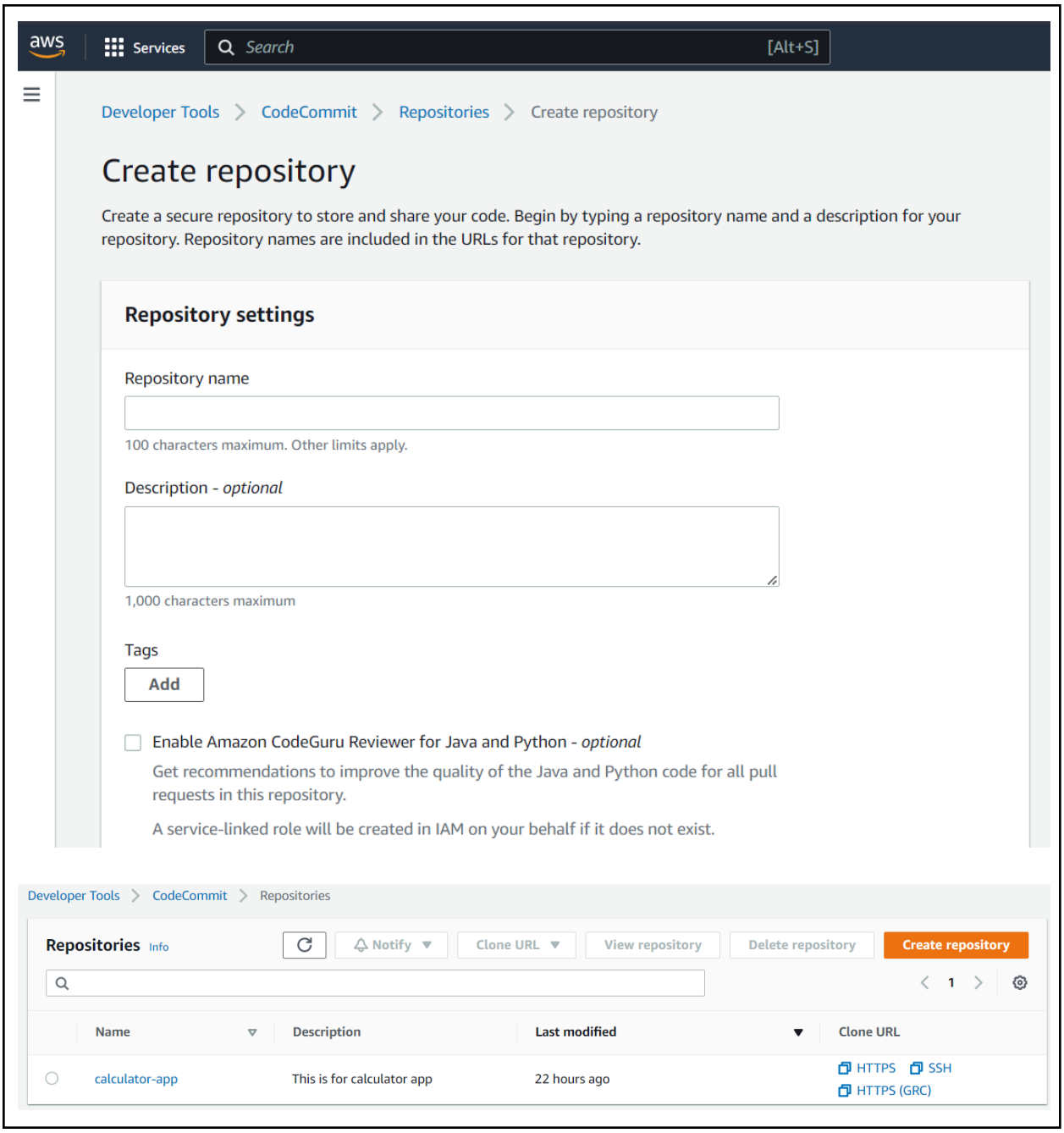
What is appspec.yml ?

It is a file used by AWS CodeDeploy to define the deployment instructions for an application. It is a YAML-formatted file that specifies the resources, files, permissions, and lifecycle events required for the deployment process.

Services required : IAM (user, roles , permissions) , CodeCommit ,Artifact(S3 bucket) CodeBuild , CodeDeploy , Pipeline

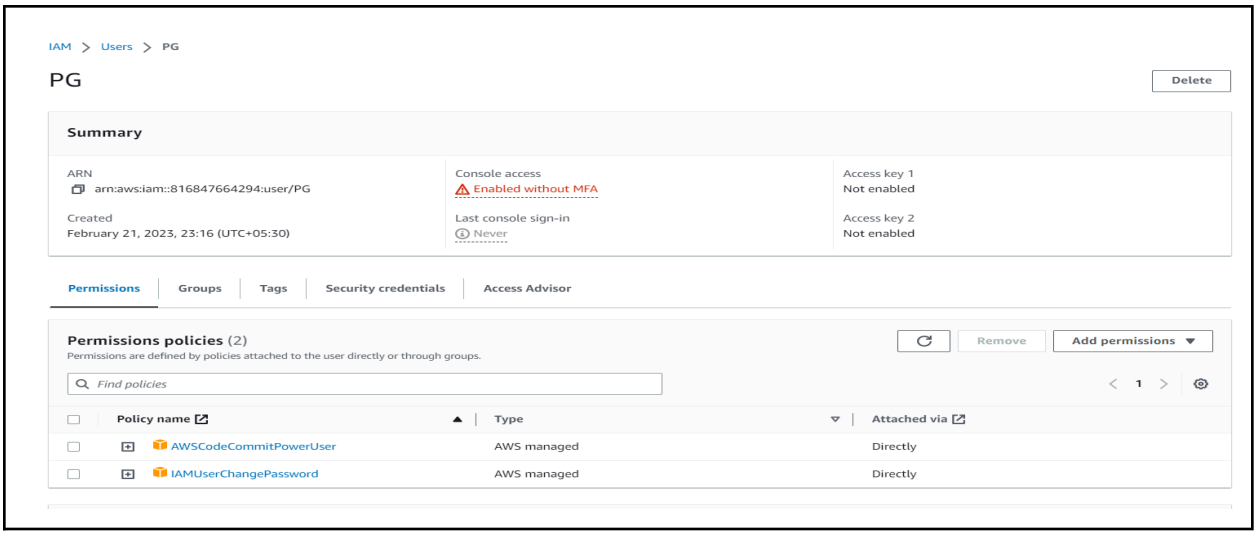
Steps:

1. First of all we need a repository where we will have our application & buildspec , appspec files , for that aws > codeCommit > repository > create New repo

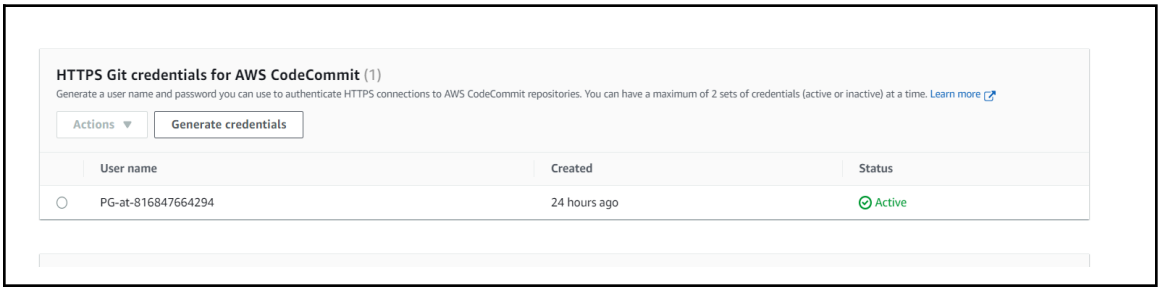


2. Now a repo has been created and we need to make this repo on the local system in order to add our application and configuration file . For that let's create a user which will access this repo on local . aws > services > IAM > user > create user

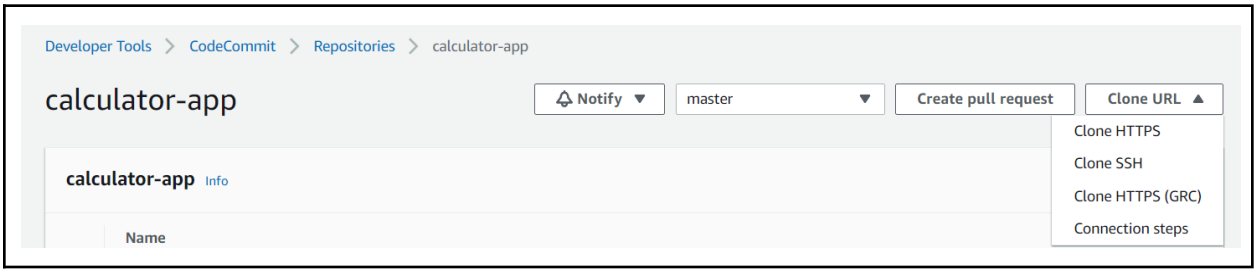
Note : Added AWSCodeCommitPowerUser permission in order to commit in this master repo from the local system.



3. Let's generate codecommit credentials in order to make changes into this repo .
   IAM > user > created User (in my case PG as you can see in above pic) > security cred > HTTPS Git credentials for AWS CodeCommit > generate



4. Let's clone this repo on local for that codeCommit > repo > Clone Url > Clone HTTPS > git link

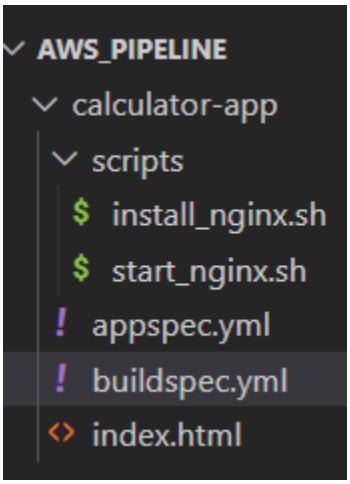


5. Open any IDE and make a directory integrated terminal and clone this repo :

Git clone <repo link> , it will ask for username and password that we have generated on user credentials .

```
PS C:\Users\my\OneDrive\Desktop\aws_pipeline> git clone https://git-codecommit.ap-northeast-1.amazonaws.com/v1/repos/calculator-app
Cloning into 'calculator-app'...
warning: You appear to have cloned an empty repository.
```

6. Let's add our application file along with configuration file and commit into aws
Inside folder on local machine create a index.html file , buildspec.yml and appspec.yml
And some nginx script files that our application will run up .



index.html × | buildspec.yml | appspec.yml | install_nginx.sh | start_nginx.sh

calculator-app > index.html > ...

```
1   <!DOCTYPE html>
2   <html>
3
4   <head>
5       <script src=
6   "https://cdnjs.cloudflare.com/ajax/libs/mathjs/10.6.4/math.js"
7           integrity=
8   "sha512-BbVEDjbqdN3Eow8+empLMrJlxXRj5nEitiCAK5A1pUr66+jLVejo3PmjIaucRnjlB0P9R3rBUs3g5jXc8ti+fQ=="
9           crossorigin="anonymous"
10          referrerpolicy="no-referrer"></script>
11      <script src=
12  "https://cdnjs.cloudflare.com/ajax/libs/mathjs/10.6.4/math.min.js"
13          integrity=
14  "sha512-iphNRh6dPbeuPGIrQbCdbBF/qcqadKWLa35YPVfMZMHBSI6PLJh1om2xCTWhpVpmUyb4IvVS9iYnnYMkleVXLA=="
15          crossorigin="anonymous"
16          referrerpolicy="no-referrer"></script>
17      <!-- for styling -->
18      <style>
19          table {
20              border: 1px solid black;
```

index.html

```
calculator-app > ! buildspec.yml
1  version: 0.2
2  phases:
3    install:
4      commands:
5        - echo installing NGINX
6        - sudo apt-get update
7        - sudo apt-get install nginx -y
8    build:
9      commands:
10       - echo Building started on `date`
11       - cp index.html /var/www/html/
12       - zip -r calapp.zip .
13   post_build:
14     command:
15       - echo Configuring nginx
16 artifacts:
17   files:
18     - '**/*'
```

Buildspec.yml file

```
alculator-app > ! appspec.yml
1  version: 0.0
2  os: linux
3  files:
4    - source: /
5      destination: /var/www/html
6  hooks:
7    AfterInstall:
8      - location: scripts/install_nginx.sh
9        timeout: 300
10       runas: root
11   ApplicationStart:
12     - location: scripts/start_nginx.sh
13       timeout: 300
14       runas: root
```

appspec.yml

```
alculator-app > scripts > $ install_nginx.sh
1  #!/bin/bash
2
3  sudo apt-get update
4  sudo apt-get install -y nginx
```

install_nginx.sh(script that will install all dependencies as per our application)

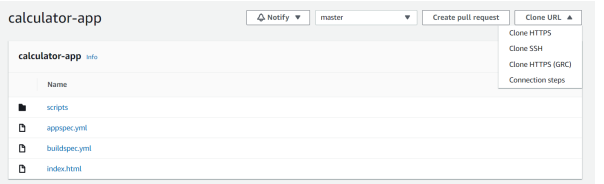
```
alculator-app > scripts > $ start_nginx.sh
1  #!/bin/bash
2
3  sudo service nginx start
```

Start_nginx.sh (script file which will start our service inorder to run up appspec.file)

7. Let’s commit all files into repo using the following steps :
    a. Git add . (adding all files into staging phase)
    b. Git commit -m “added all files” (commit version with specified message)
    c. Git push origin master (since our repo has master branch)

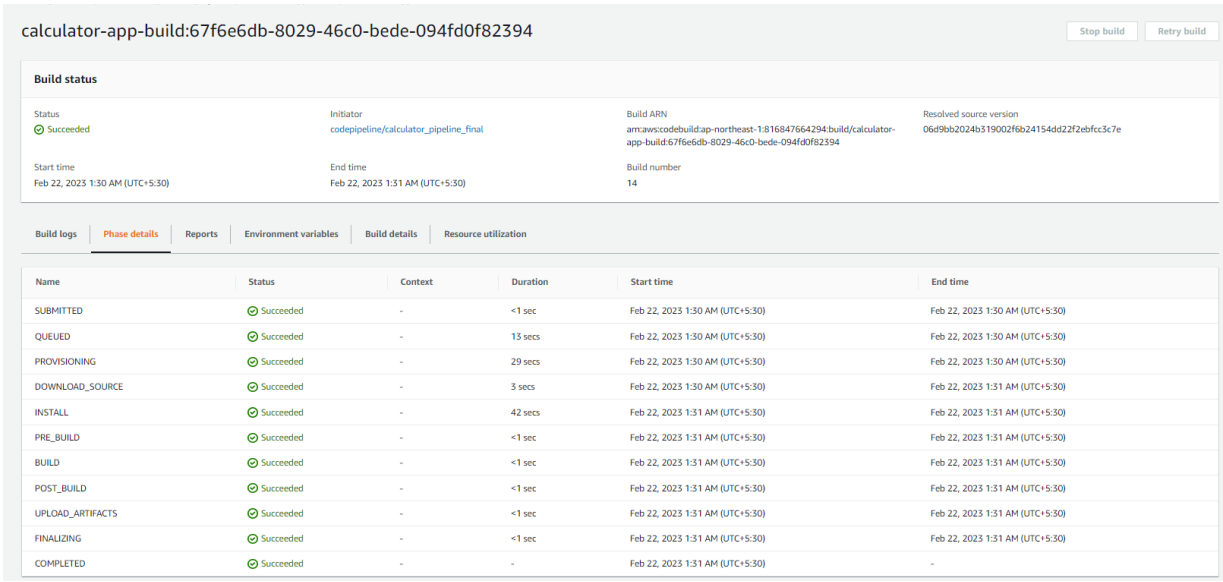
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\my\OneDrive\Desktop\aws_pipeline\calculator-app> git add .
PS C:\Users\my\OneDrive\Desktop\aws_pipeline\calculator-app> git commit -m "added index.html"
[master (root-commit) 2da2e3b] added index.html
 1 file changed, 141 insertions(+)
 create mode 100644 index.html
PS C:\Users\my\OneDrive\Desktop\aws_pipeline\calculator-app> git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 1.35 KiB | 460.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.ap-northeast-1.amazonaws.com/v1/repos/calculator-app
 * [new branch]      master -> master
PS C:\Users\my\OneDrive\Desktop\aws_pipeline\calculator-app> git add .
PS C:\Users\my\OneDrive\Desktop\aws_pipeline\calculator-app> git commit -m "buildspec"
```

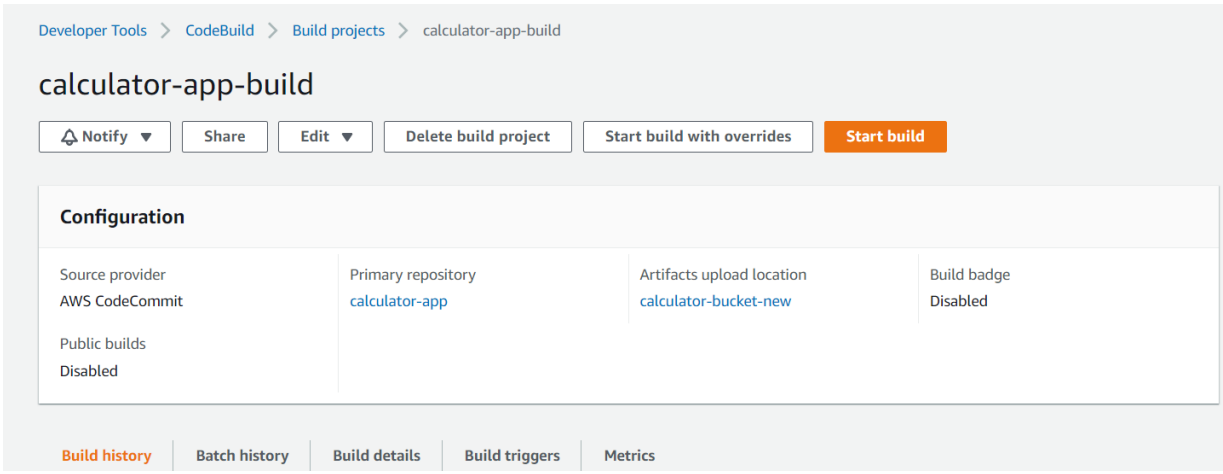


As we can see all files pushed successfully

8. It is time to build our code manually to check whether it is working or not . for that aws > codeBuild > build projects > specify your all details like code will come from aws codeCommit > build now

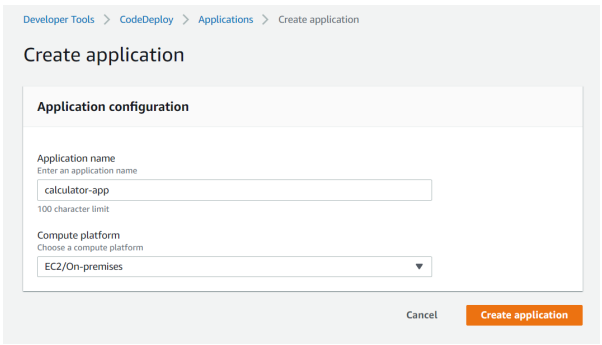


As you can see my code build status is successful.

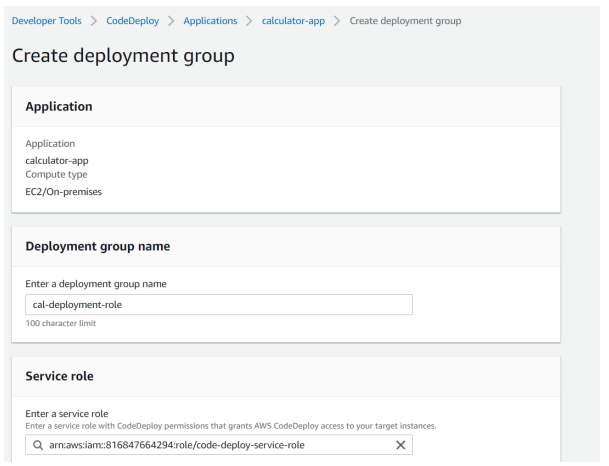


Here i have specified artifact location as well which is s3 bucket where my zip file will generated (basically my application image will be generated)

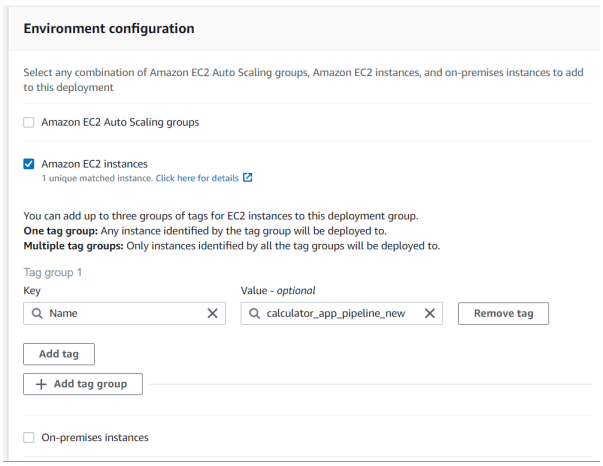
9. Now let’s make a deployment group and deploy our app . So a deployment group is needed in order to run up our deployment .



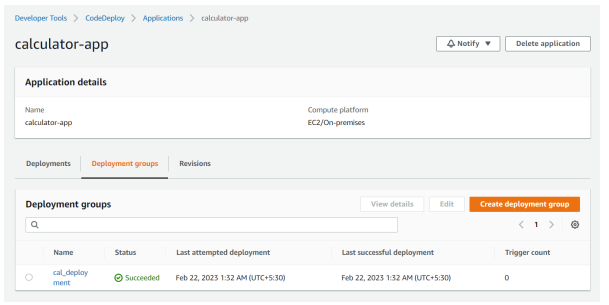
Here i have created deployment application



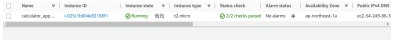
Here I have created a deployment group with code-deploy service role where all permission provided in this service.



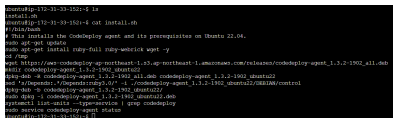
Provided my ec2 instance as well where we will deploy our application.



10. Now let's create ec2 application and install agent manually on instance for that :
Aws > ec2 > launch instance & we will run a script with specified region in my case it is Tokyo which is ap-northeast-1



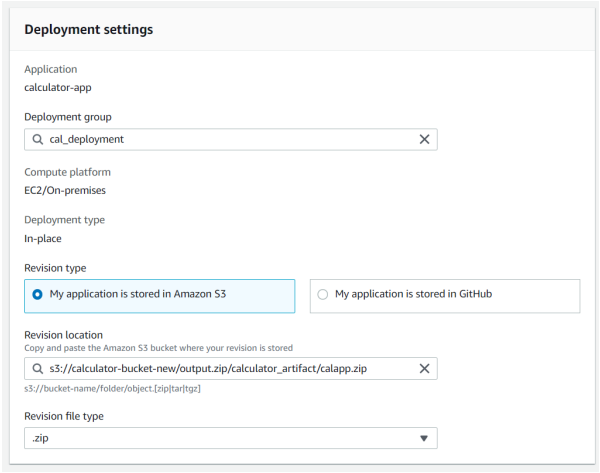
Ec2 instance running and up (t2.micro)



This is the script for installing agent on ec2

And here is our agent running and active

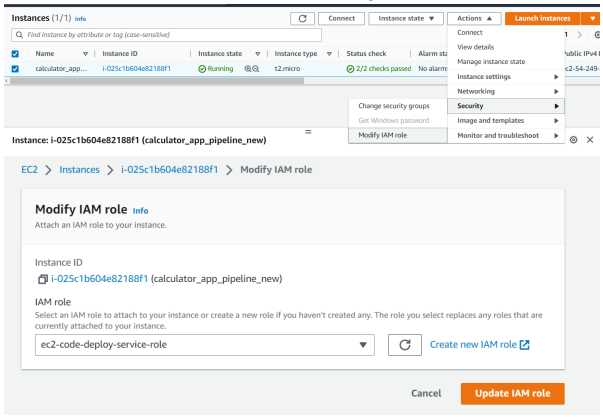
11. Let's make a deployment :
Aws > code deploy > application > create deployment



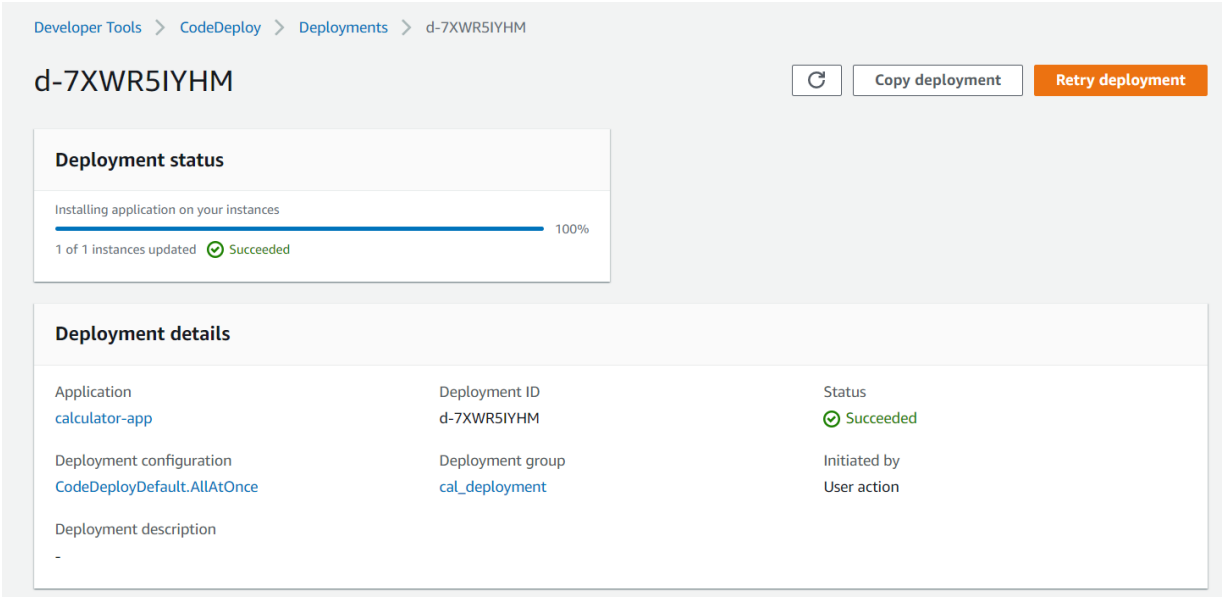
I have specified my artifact location and deployment group which we created earlier

Now we need ec2 code deploy service role in order to install all into deployment

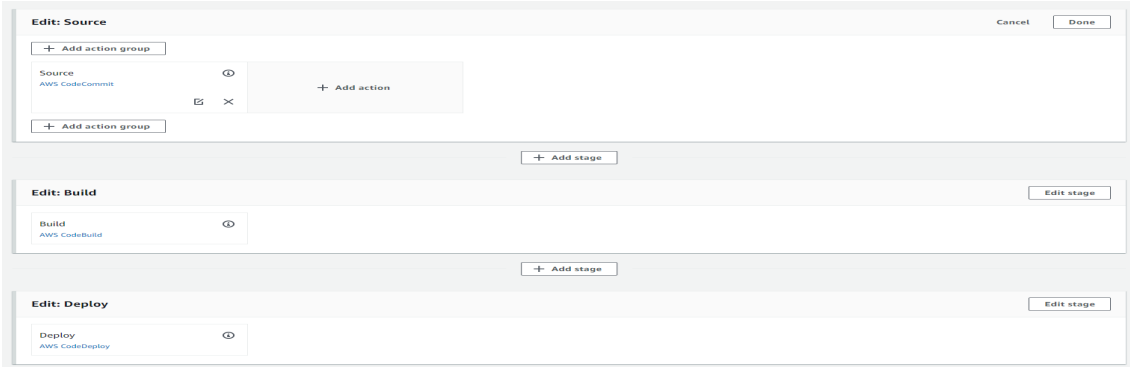
Ec2 > select instance > security > modify iam role > chose ec2 code deploy role



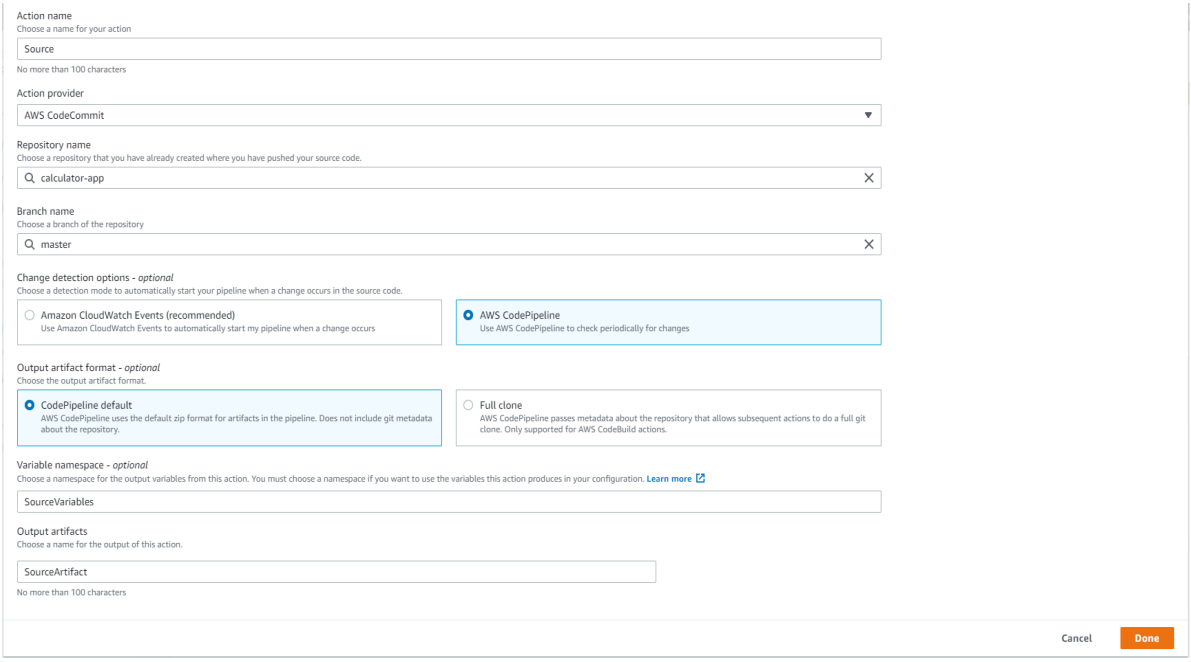
Now installation completed :



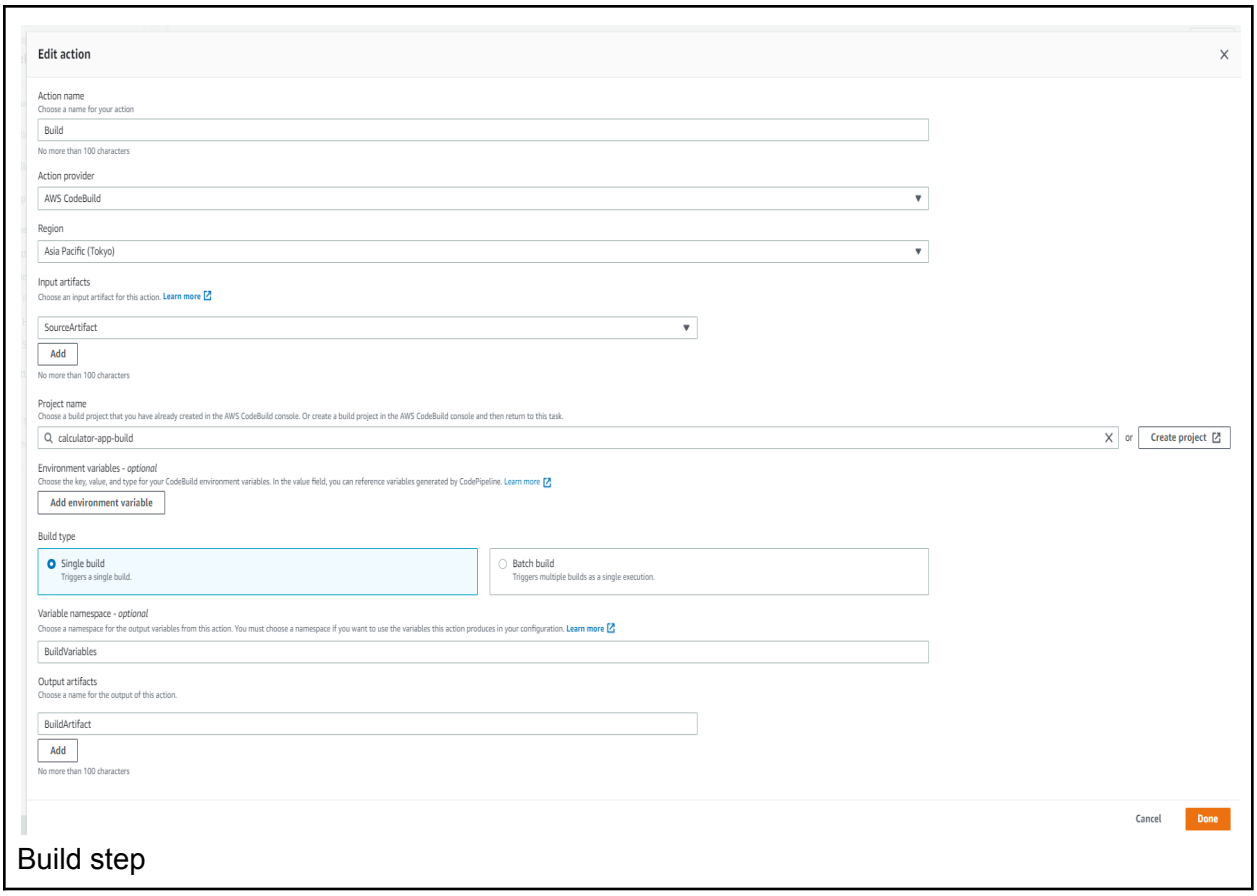
12. Now let's create a CI / CD pipeline into aws pipeline : aws > developer tool > codePipeline > create pipeline which will be source -> build -> deploy



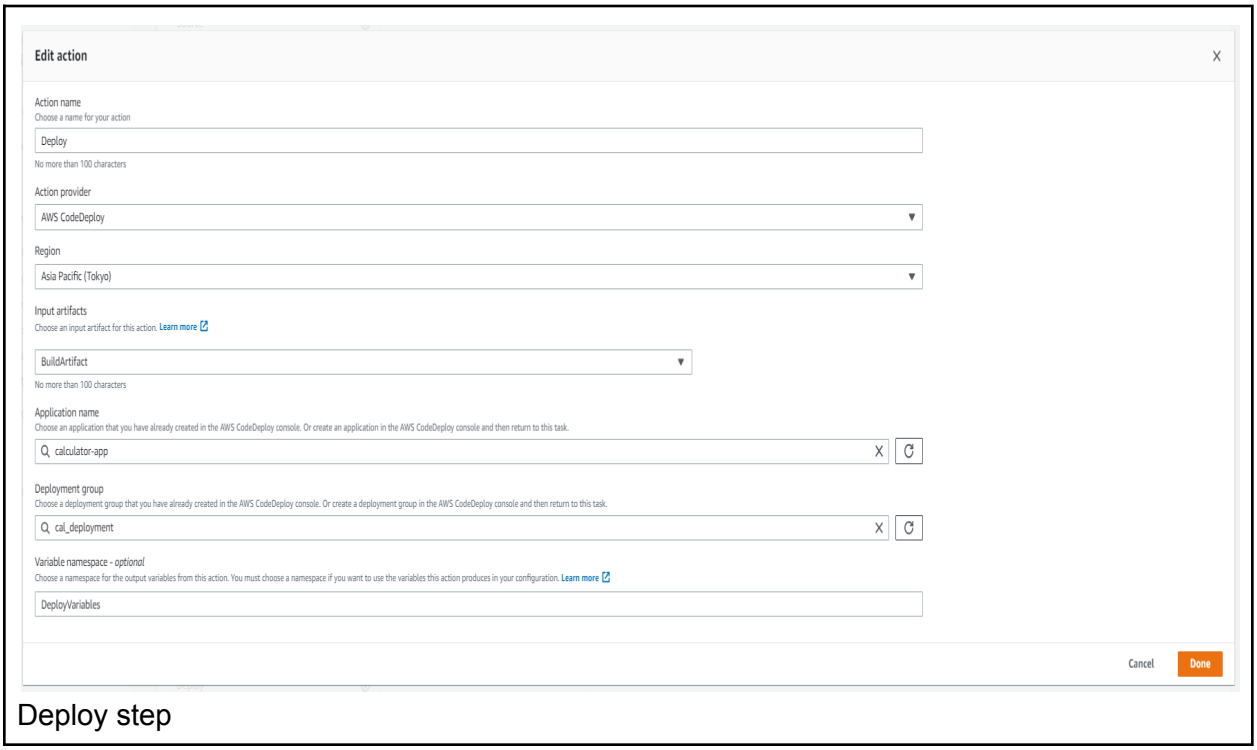
Overall steps in pipeline : source > build > deploy



Source step



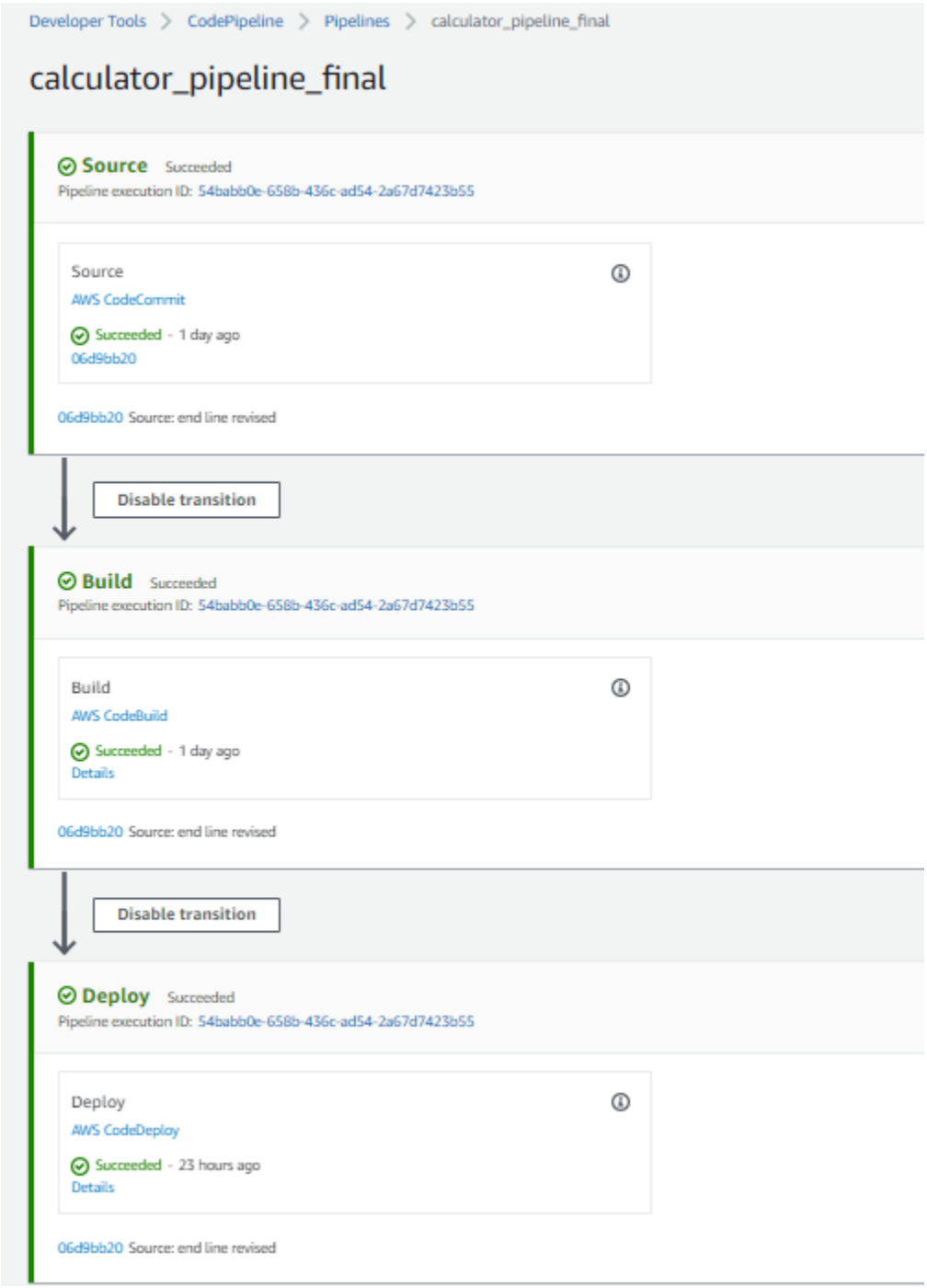
Build step



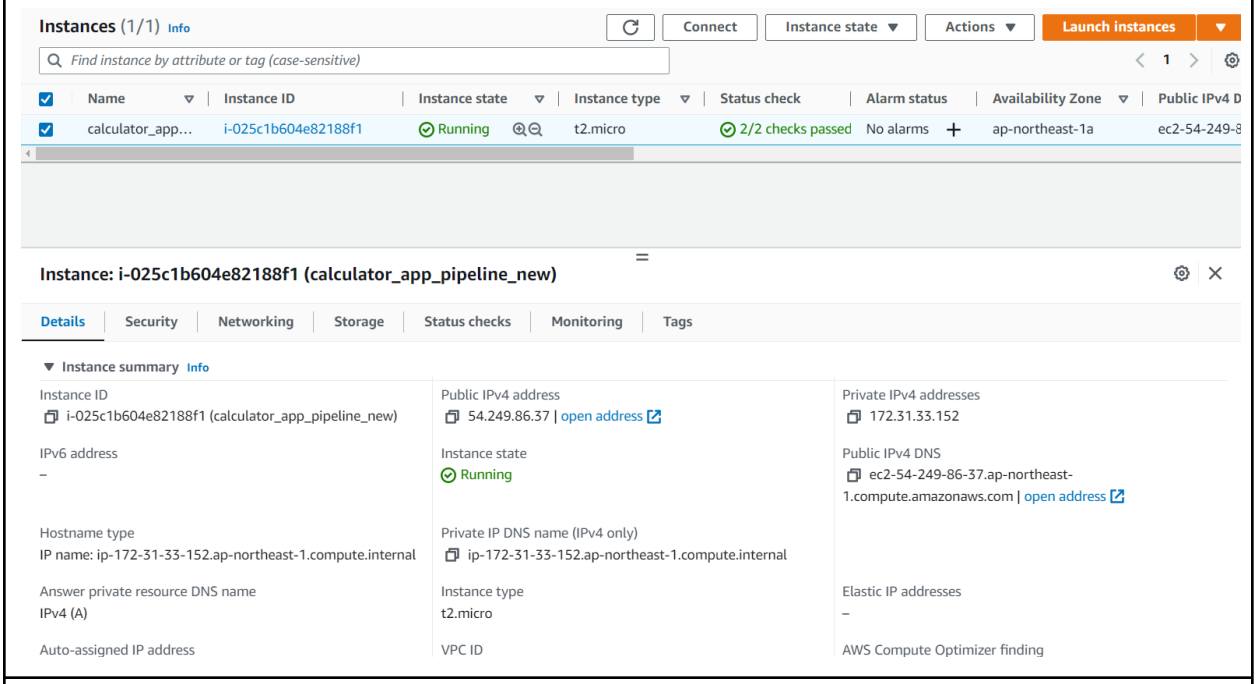
Deploy step

13. Final step let’s build our CI /CD pipeline automation :

Click save pipeline settings > start build and it will start automatically and in background it will tell us build process , steps and etc



Our pipeline is succeeded it means our all code successfully passed .Now let’s check our ec2 instance public IP our calculator app should be running :



Our ec2 public ip is 54.249.86.37 so hitting this ip into browser our calculator app running successfully