Day 52: Your CI/CD pipeline on AWS - Part 3 🚀 ☁️

On your journey of making a CI/CD pipeline on AWS with these tools, you completed AWS CodeCommit & CodeBuild.

Next few days you'll learn these tools/services:

- CodeDeploy
- CodePipeline
- S3

What is CodeDeploy ?

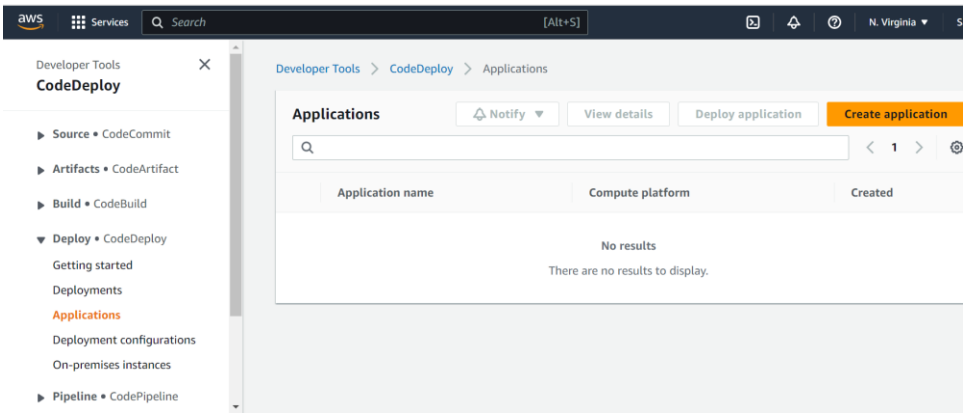
- AWS CodeDeploy is a deployment service that automates application deployments to Amazon EC2 instances, on-premises instances, serverless Lambda functions, or Amazon ECS services.

CodeDeploy can deploy application content that runs on a server and is stored in Amazon S3 buckets, GitHub repositories, or Bitbucket repositories. CodeDeploy can also deploy a serverless Lambda function. You do not need to make changes to your existing code before you can use CodeDeploy.

Task-01:

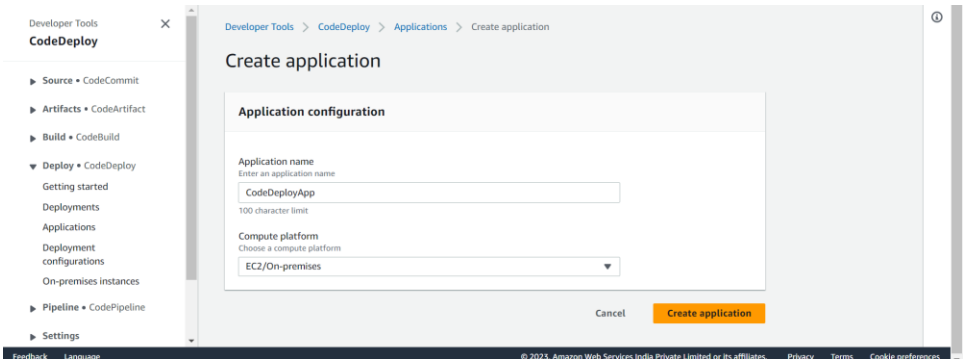
- Read about Appspec.yaml file for CodeDeploy.

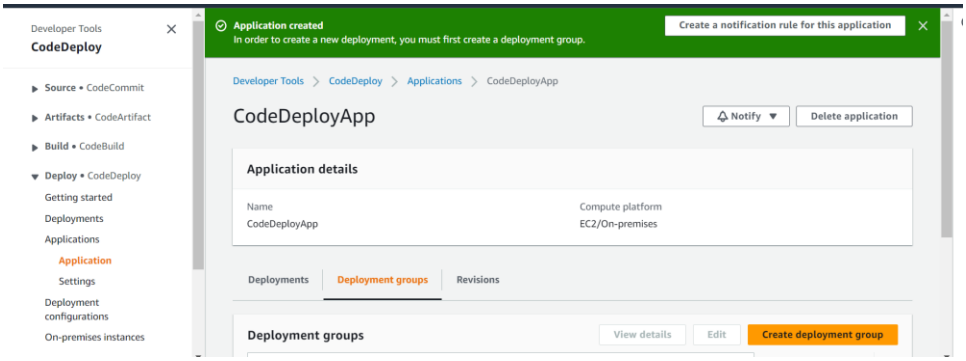
Goto CodeDeploy Tab and First Create an application



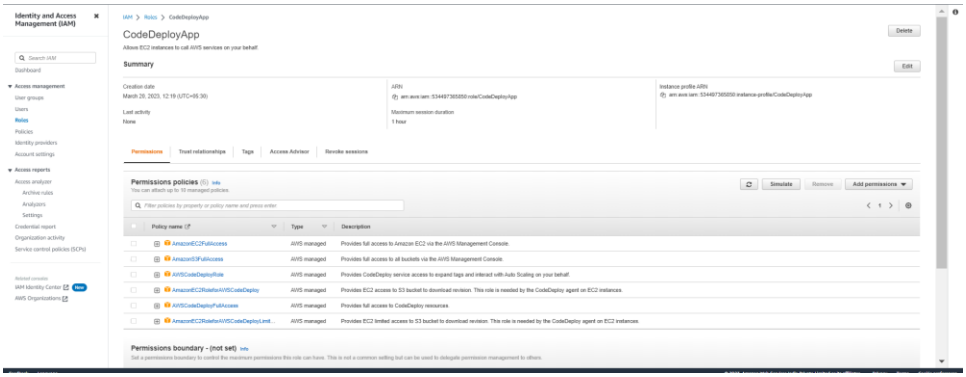
Give Application Name and Select Compute Platform.

Click on Create Application.

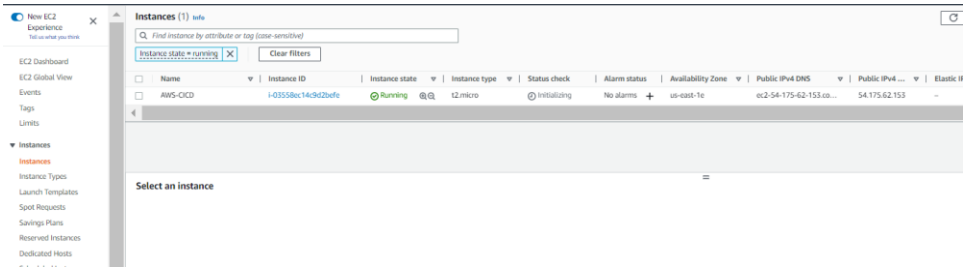




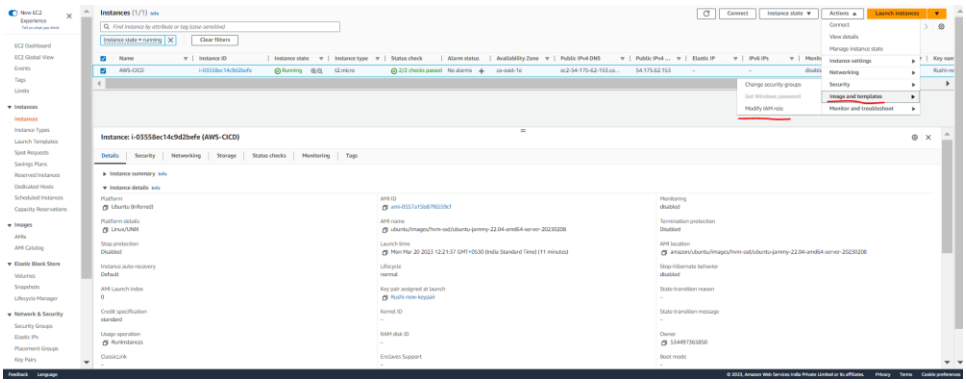
For Deployment, we need an IAM Role, Using This role we can establish a connection between CodeDeploy and EC2 Instance.

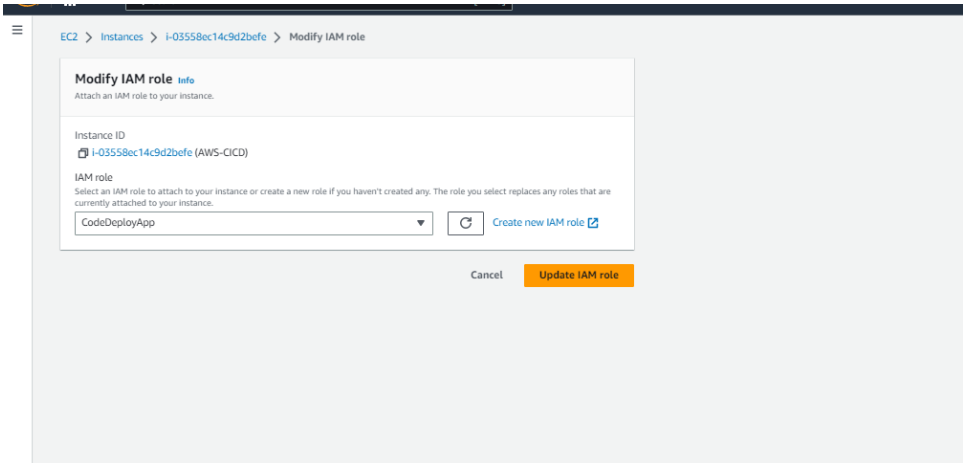


Create a one Instance



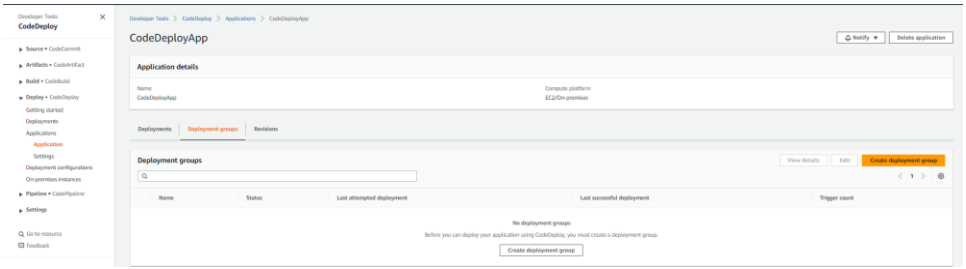
Attache newly created IAM Role to this Instance.

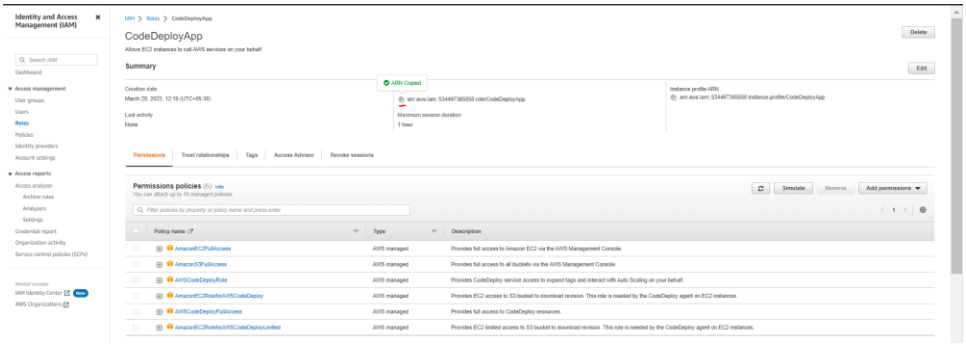




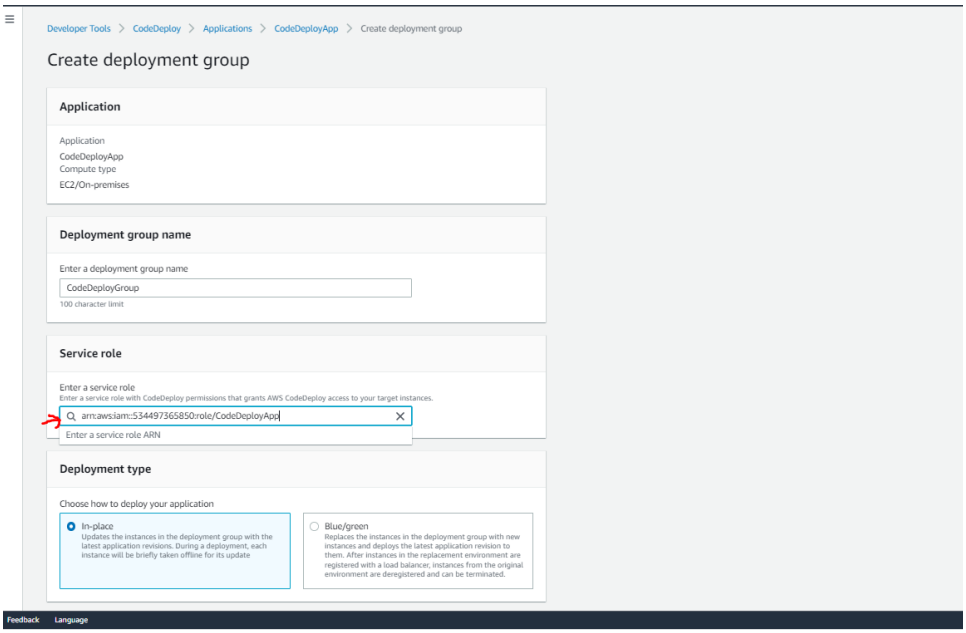
Ok, Now the Second Step is to Create a Deployment group.

Click on create Deployment group button

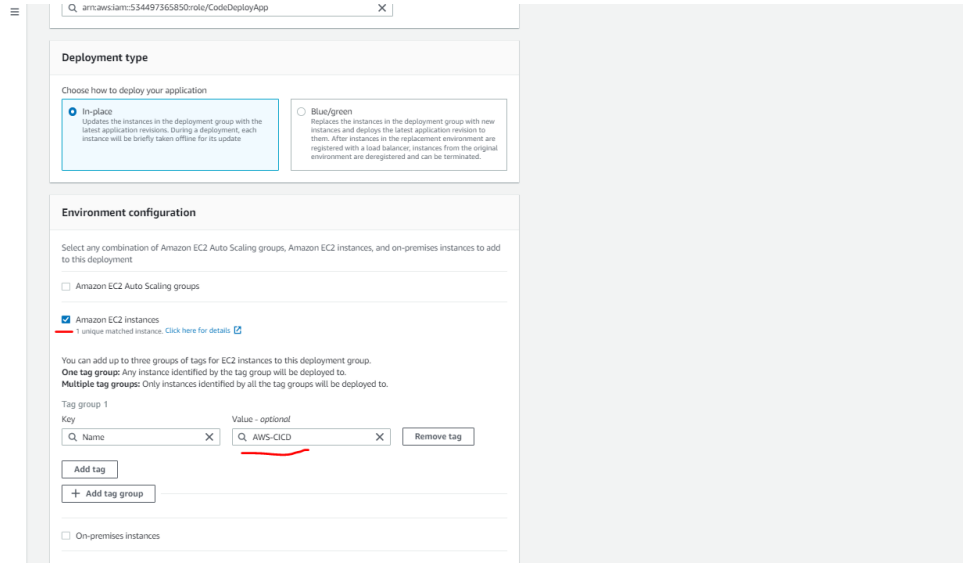




Give Deployment group name and IAM Service Role ARN.

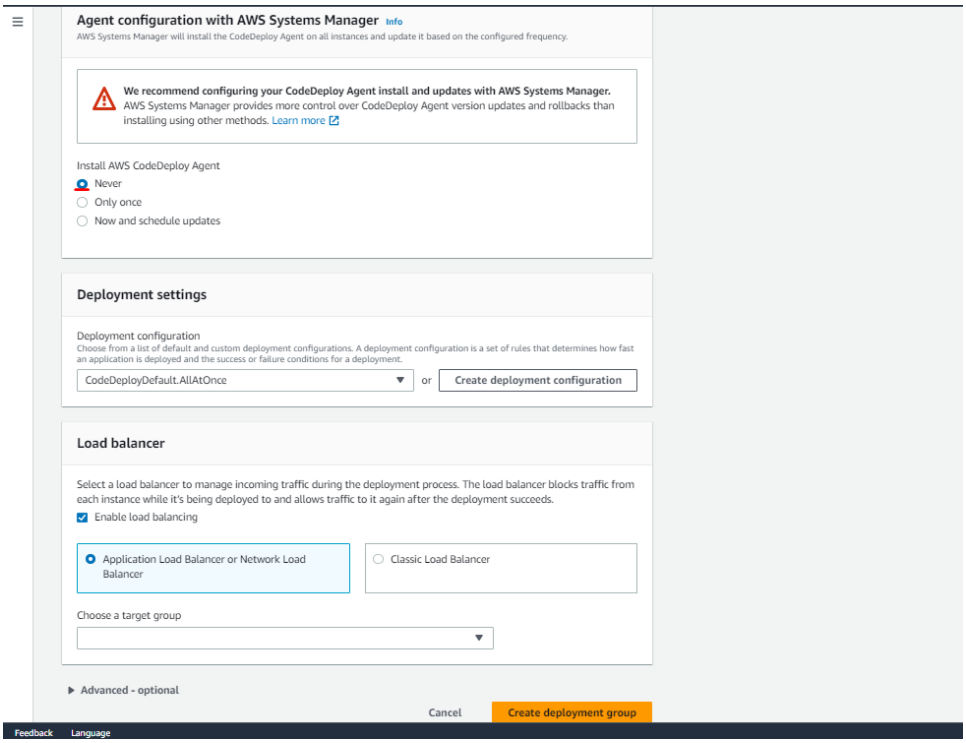


Choose **In-place** as a Deployment type and In Environment Configuration click on **Amazon EC2 Instance** checkbox and add Running EC2 instance name.

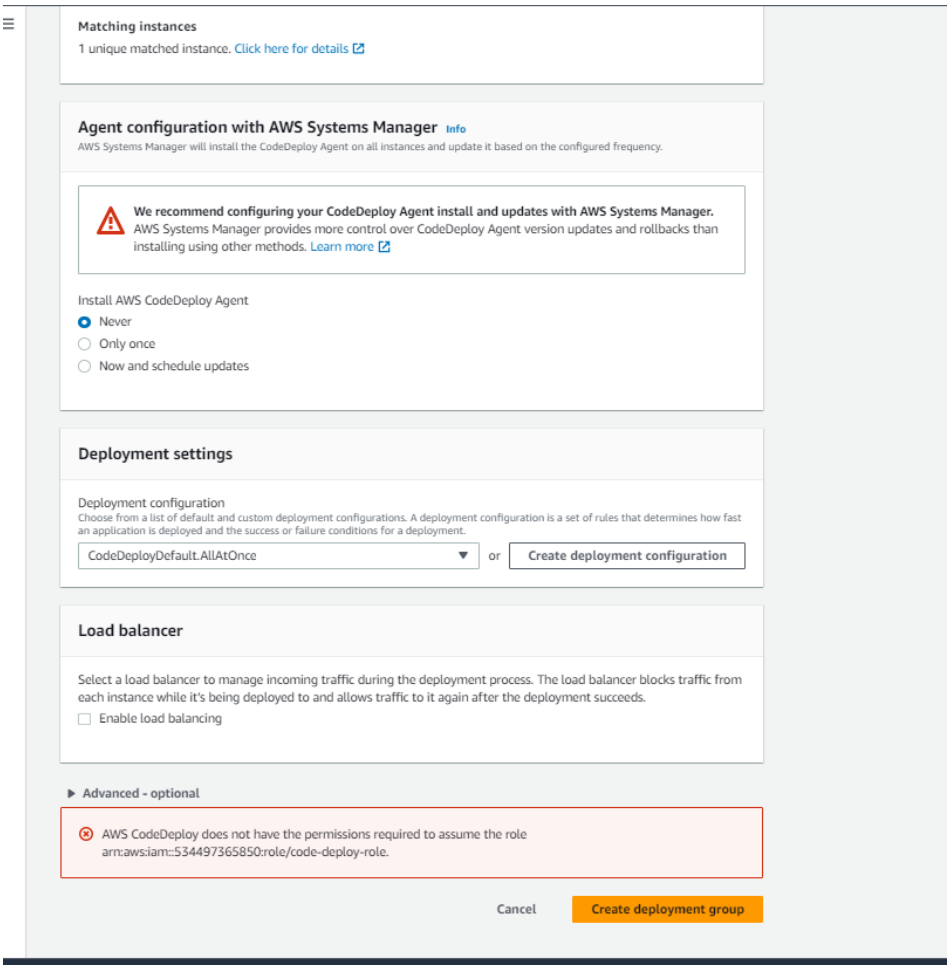


Here in this step, we don’t want to install CodeDeploy Agent.

We can directly install the CodeDeploy agent on EC2 instance later.



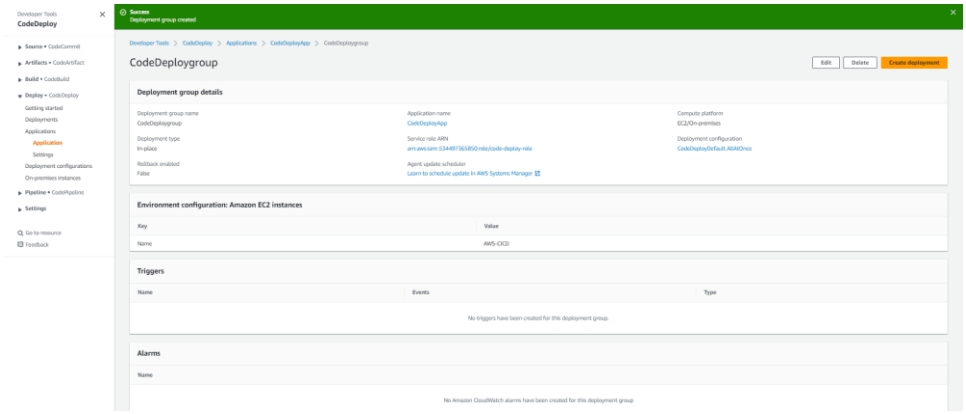
Note: - if you are getting this type of error, when creating a deployment group, follow below Reference document to solve this type of issue.



Solution: -

[AWS CodeDeploy does not have the permissions required to assume the role]

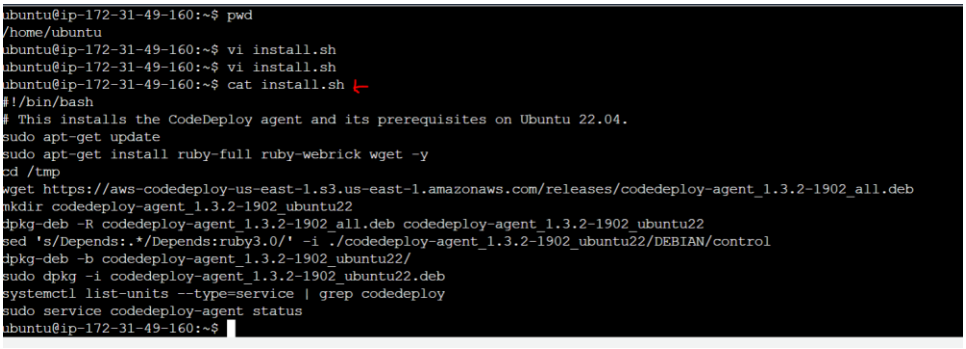
Deployment Group is Created Successfully.

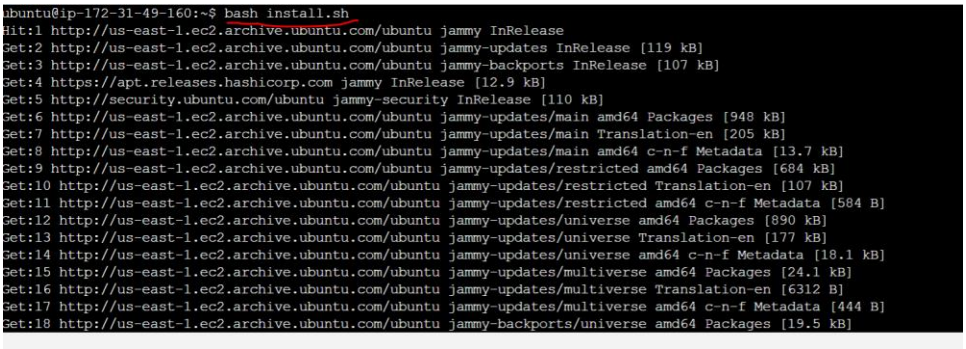


Here in this Step, we are setting up the CodeDeploy Agent.

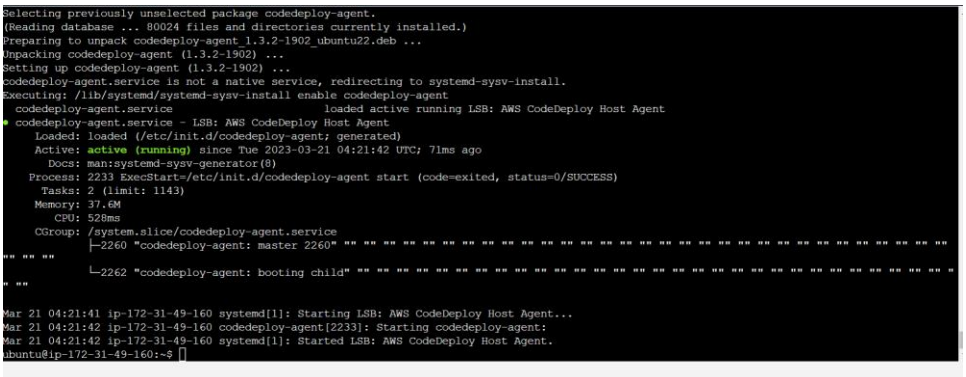
Refer Below Document, to get the All commends to Setup agent on EC2 instance.

[Setting Up AWS CodeDeploy Agent on Ubuntu EC2]





Your Code Deploy Agent is Running.



Create a Basic index page on local

```
Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ vi index.html

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ cat index.html
<!DOCTYPE html>
<html>
    <head>
        <title>Basic Web Page</title>
    </head>
    <body>
Hello Rushikesh!
Day 52 Task is Completed
    </body>
</html>
```

For Run the application, we need **appspec.yaml** file

Version: specifying version
OS: specifying Required OS
Files: Specifying Remote file path
Source: current Remote location
Destination: Default Nginx index path
hooks:
AfterInstall:
location: specify the nginx install script path
timeout: after 300 sec, it terminates automatically.
runas: Run as a Root user.
ApplicationStart:
location: Specify the nginx start script path.

```
Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ vi appspec.yaml

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ cat appspec.yaml
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html
hooks:
  AfterInstall:
    - location: scripts/install_nginx.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: scripts/start_nginx.sh
      timeout: 300
      runas: root
```

```
Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ mkdir scripts

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ vi ^C

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ vi install_nginx.sh

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ cat install_nginx.sh
#!/bin/bash
sudo apt-get update
sudo apt-get install nginx -y

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ mv install_nginx.sh scripts/

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ cd scripts/

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo/scripts (master)
$ vi start_nginx.sh

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo/scripts (master)
$ cat start_nginx.sh
#!/bin/bash

sudo service nginx start
```

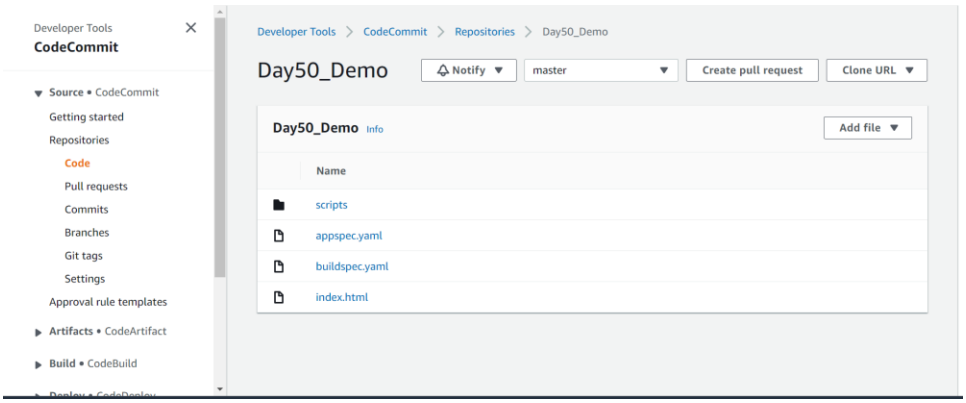
Add and commit to all local changes.

```
Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        appspec.yaml
        scripts/

no changes added to commit (use "git add" and/or "git commit -a")

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ git add .
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'appspec.yaml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'scripts/install_nginx.sh', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'scripts/start_nginx.sh', LF will be replaced by CRLF the next time Git touches it

Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   appspec.yaml
        modified:   index.html
        new file:   scripts/install_nginx.sh
        new file:   scripts/start_nginx.sh


Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ git commit -m "Added Appspec and Scripts"
[master 7618e98] Added Appspec and Scripts
 4 files changed, 21 insertions(+)
 create mode 100644 appspec.yaml
 create mode 100644 scripts/install_nginx.sh
 create mode 100644 scripts/start_nginx.sh
```

Push Committed Changes into Remote Repository.

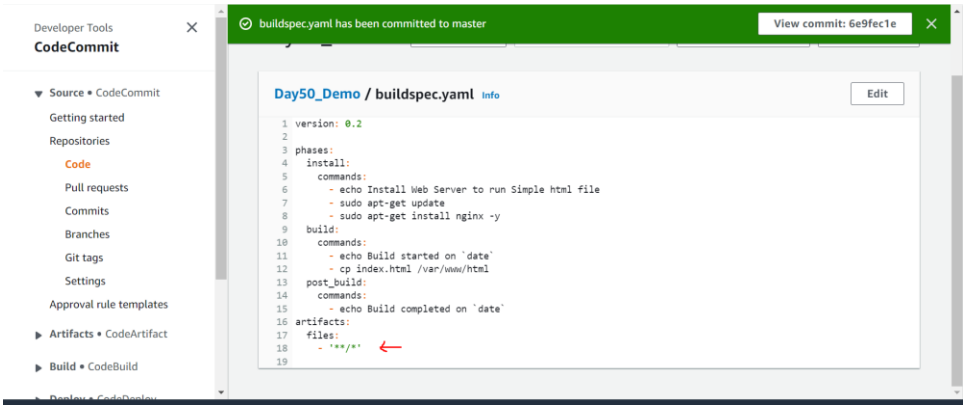
```
Rushikesh@DESKTOP-OJSE6R3 MINGW64 ~/Desktop/DevOps/Day50/Day50_Demo (master)
$ git push origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 755 bytes | 53.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/Day50_Demo
   4620fbb..7618e98  master -> master
```

All local files are successfully pushed to Code Commit Repo.

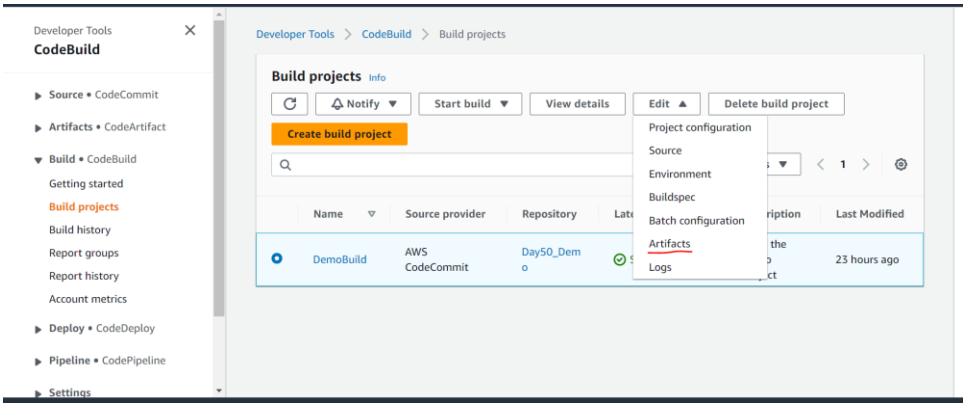


Change the **buildSpec** files.

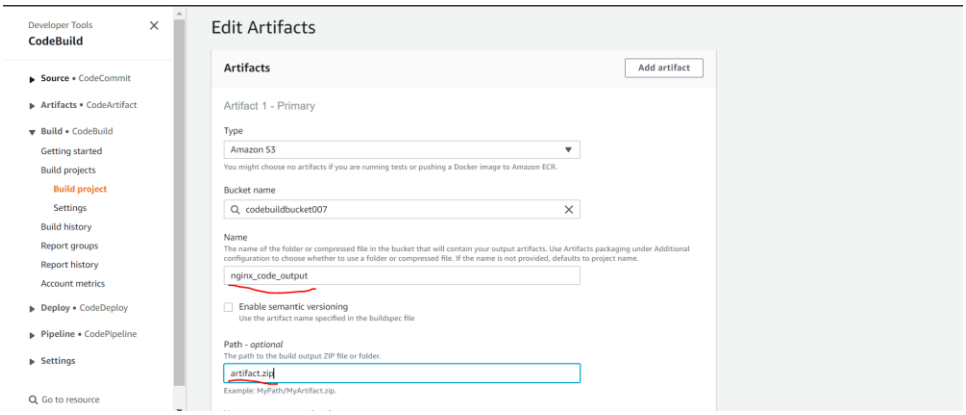
‘**/*’ means add all files in an artifact.



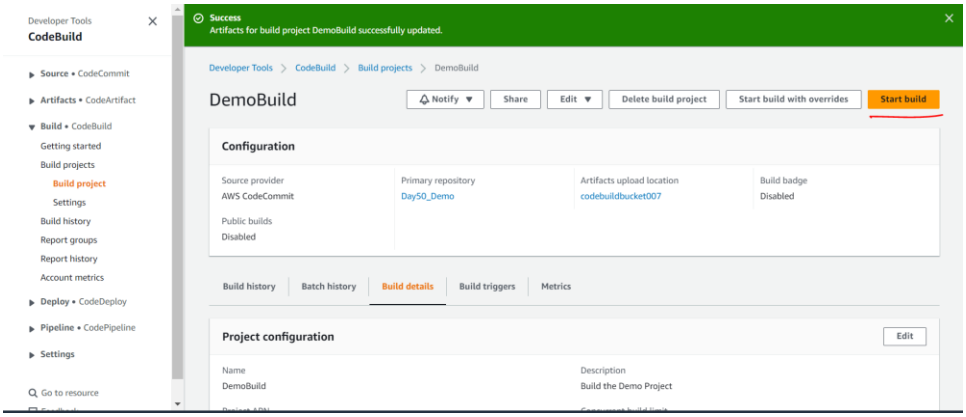
Edit the Artifacts



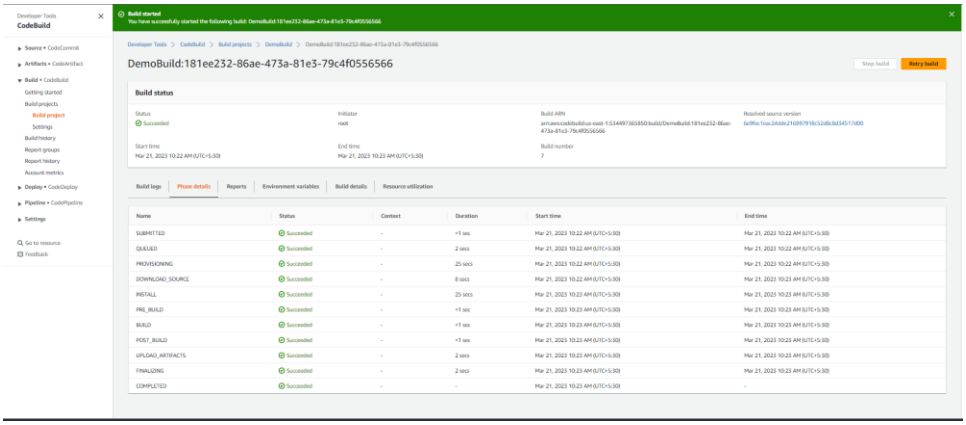
Add Artifact Name and path



Click on Start Build



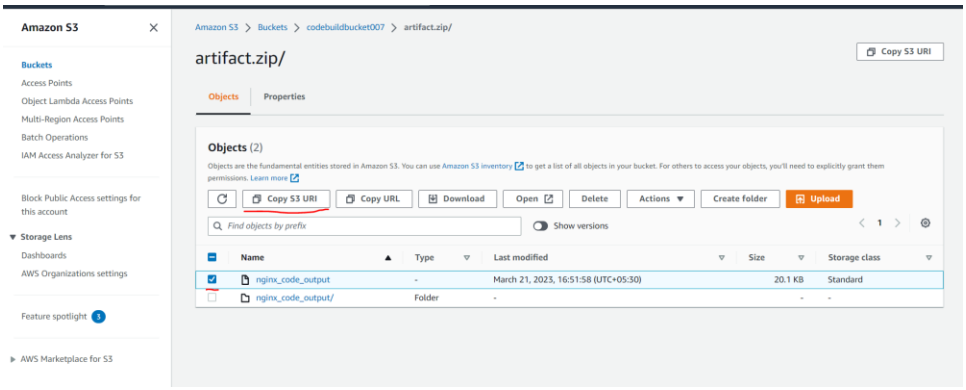
Build Successful

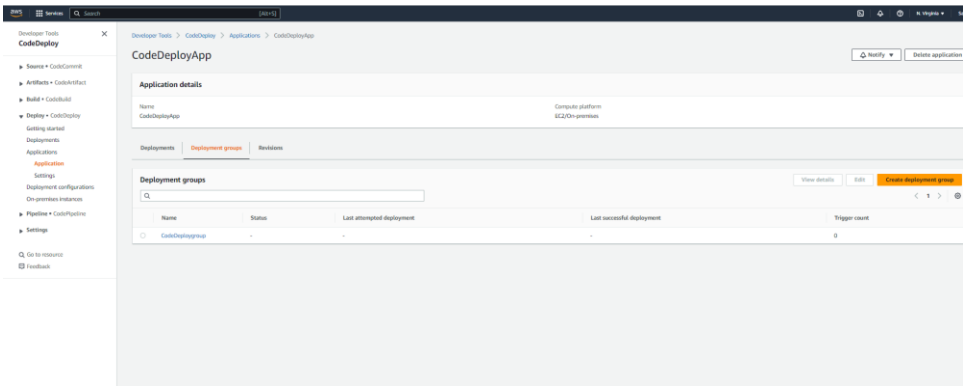


The Third step is creating a Deployment.

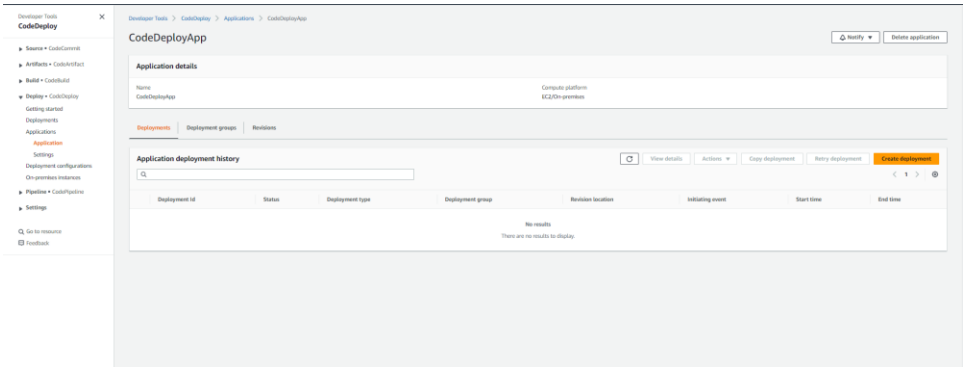
For this deployment, need Artifact URI.

Goto **s3 and copy the S3 URI.**

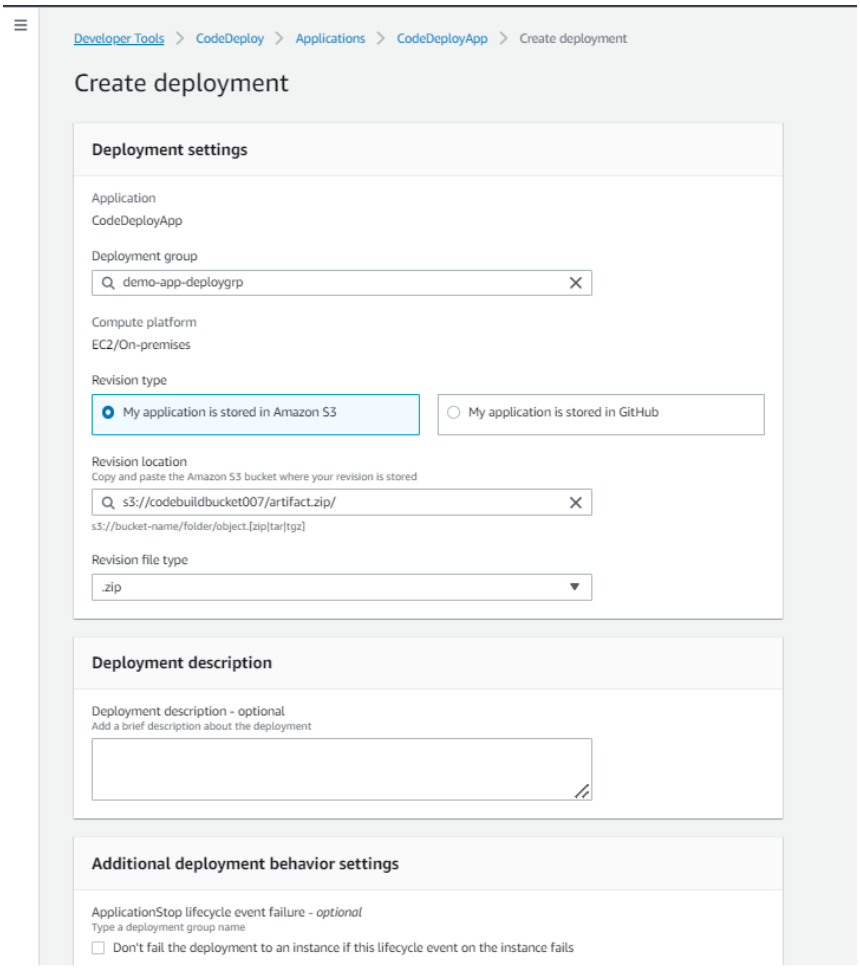




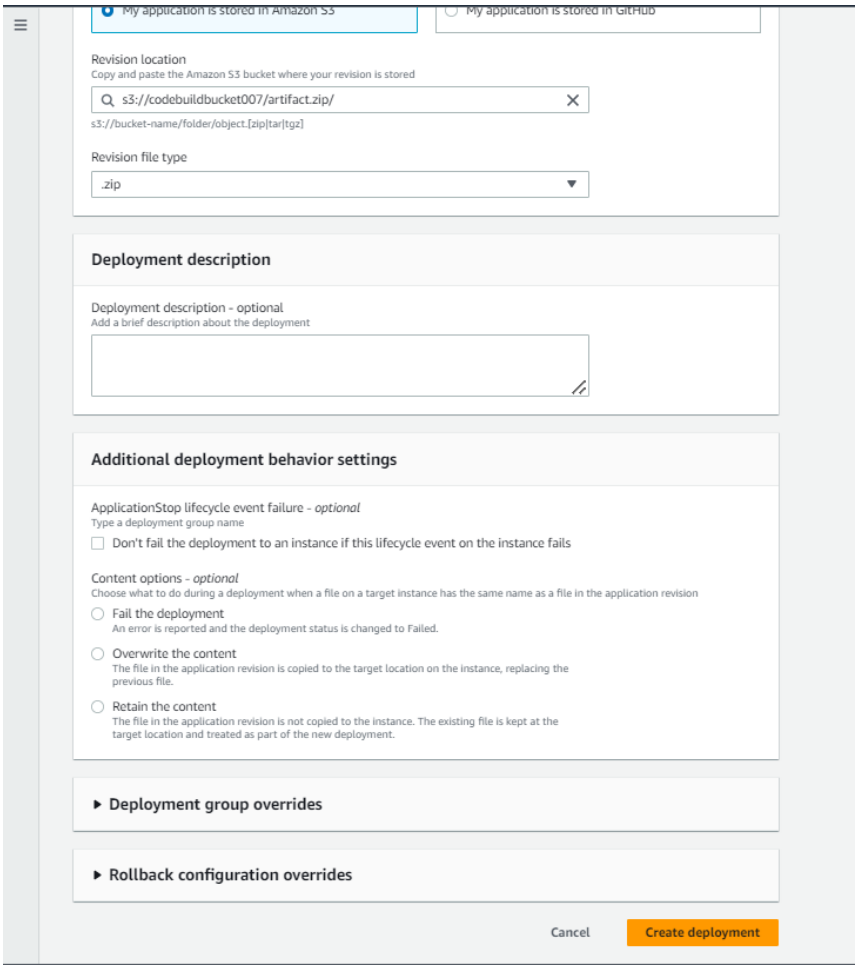
Click on Create Deployment.

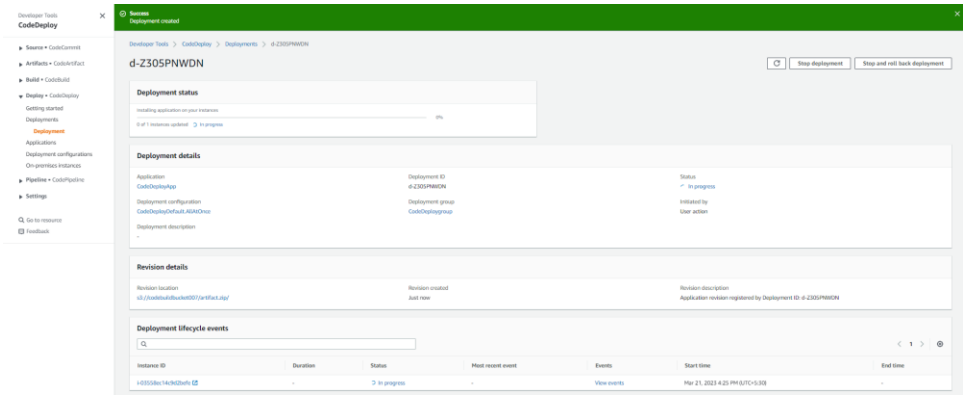


Add Revision location and Revision type.

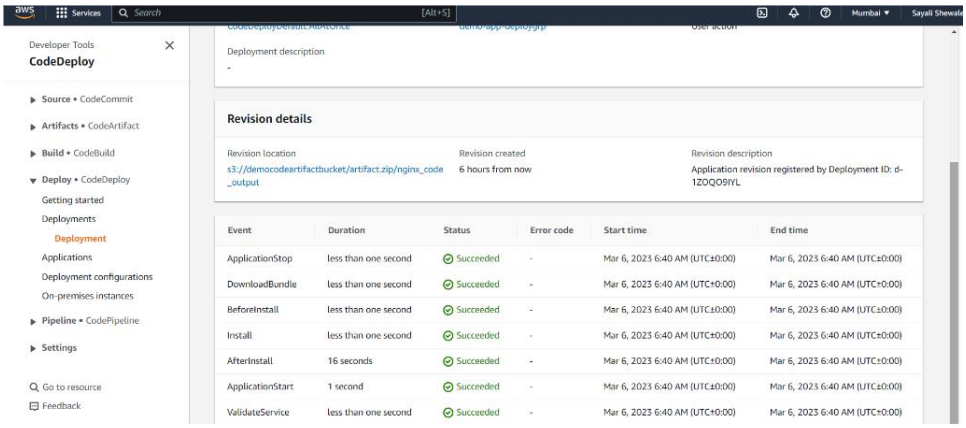


Click on Create Deployment.





All Events are Succeeded.



Happy Learning :)