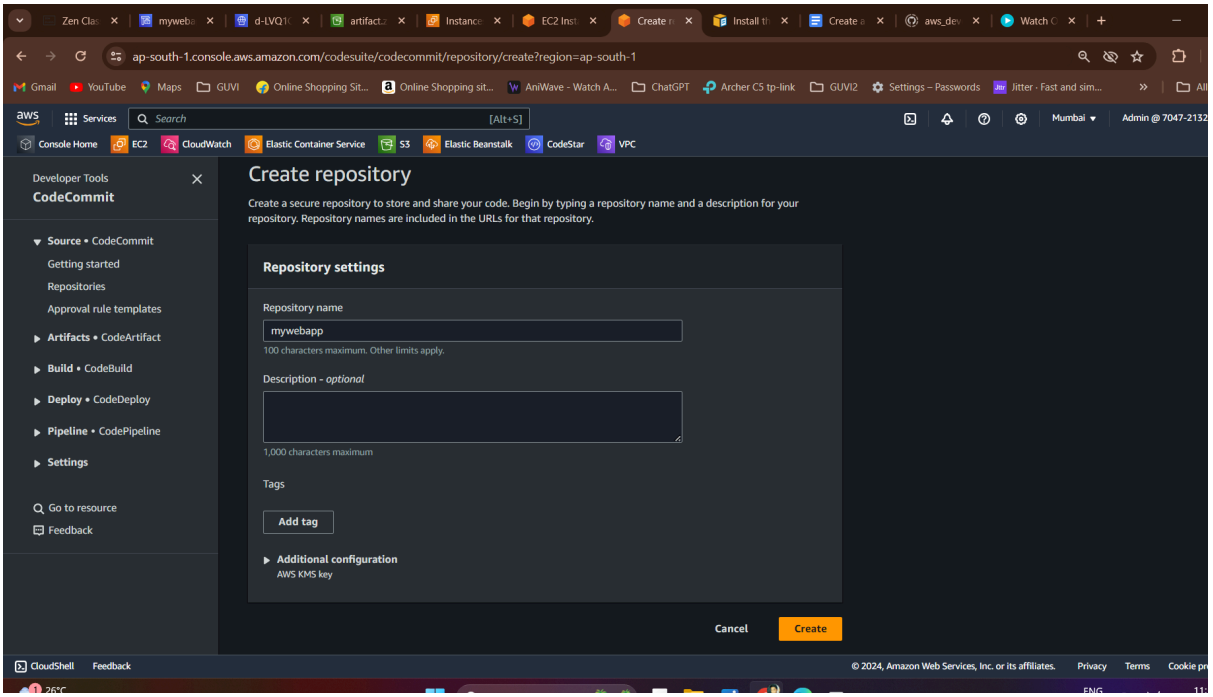
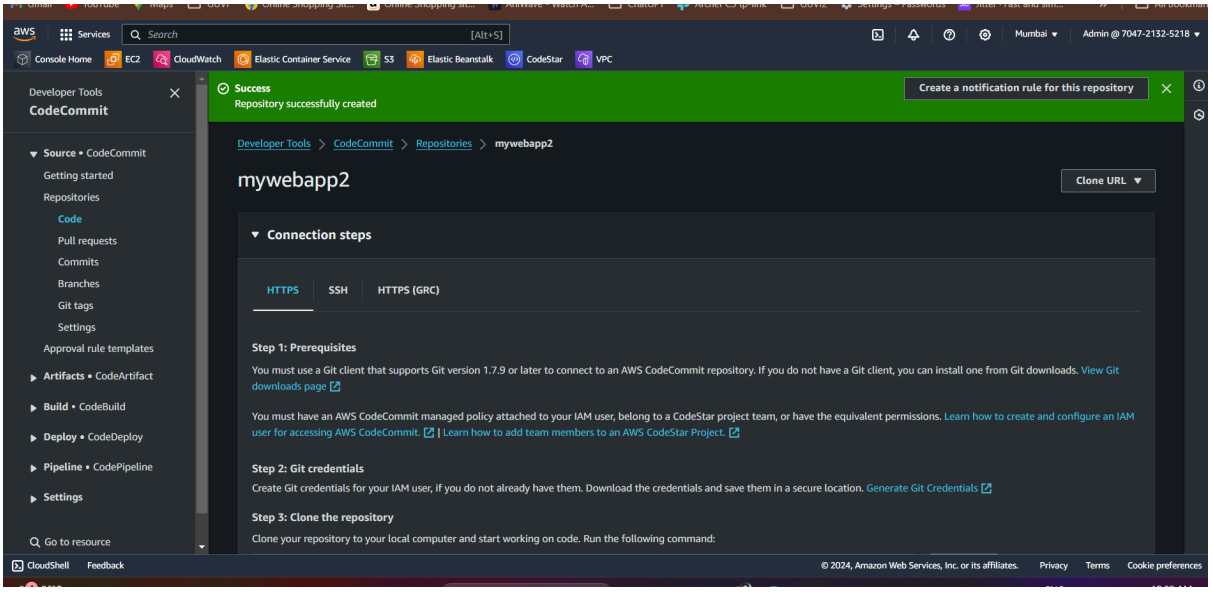
Deploy a simple Nginx application using AWS code commit and deploy & access via browser

Step 1: Create AWS CodeCommit repository - (**CC**).

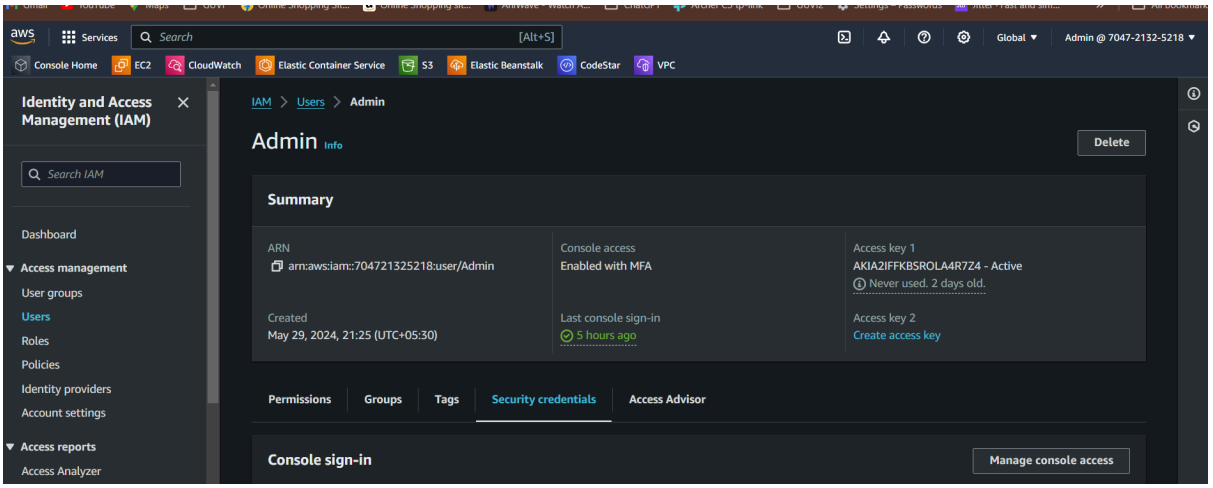


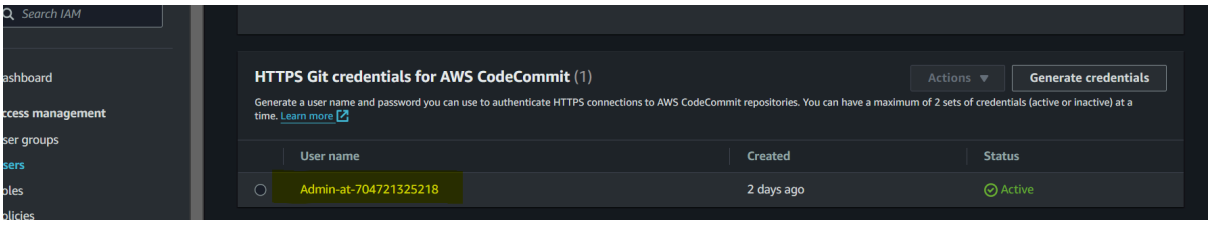


Note: It is not good to access **CodeCommit through root user**, so we have to **use IAM user to access CodeCommit**.

Step 2: Click on Clone URL —> Clone HTTPS —> Copy the URL link - (**CC**).

Step 1: Go to IAM user —> Security credentials —> HTTPS Git credentials for AWS CodeCommit —> Click Generate credentials —> Click Download - (**IAM**).





Step 2: Use the above credentials and authenticate Codecommit repositary - (**IAM**).

Step 3: Open Git Bash —> Enter Download folder —> Clone the HTTPS URL link from AWS CodeCommit as below (Actual folder is **mywebapp** and the one shown in **1** and **2** SS is example and Use above downloaded **HTTPS credentials as Password** for the below for git clone repositary link authentication) - (**IAM**).

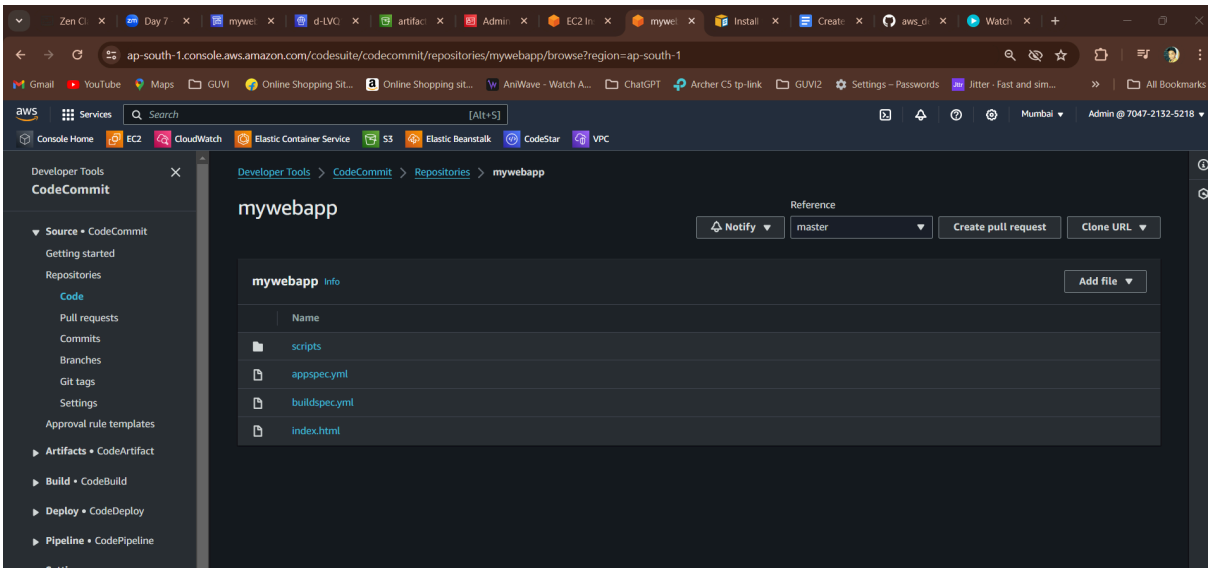
1.

```
MINGW64:/c/Users/Syed Abdul Rahim/Downloads/mywebapp2

Syed Abdul Rahim@LAPTOP-UDGP3QIT MINGW64 ~/Downloads/mywebapp2 (master)
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/mywebapp2
Cloning into 'mywebapp2'...
warning: You appear to have cloned an empty repository.

Syed Abdul Rahim@LAPTOP-UDGP3QIT MINGW64 ~/Downloads/mywebapp2 (master)
$ ls
mywebapp2/

Syed Abdul Rahim@LAPTOP-UDGP3QIT MINGW64 ~/Downloads/mywebapp2 (master)
$ ls
appspec.yml  buildspec.yml  index.html  scripts/

Syed Abdul Rahim@LAPTOP-UDGP3QIT MINGW64 ~/Downloads/mywebapp2 (master)
$ git add .
warning: in the working copy of 'Downloads/mywebapp2/appspec.yml', LF will be replaced by CRLF the next time Git touche
s it
warning: in the working copy of 'Downloads/mywebapp2/buildspec.yml', LF will be replaced by CRLF the next time Git touc
hes it
warning: in the working copy of 'Downloads/mywebapp2/index.html', LF will be replaced by CRLF the next time Git touches
 it
warning: in the working copy of 'Downloads/mywebapp2/scripts/install_nginx.sh', LF will be replaced by CRLF the next ti
me Git touches it
warning: in the working copy of 'Downloads/mywebapp2/scripts/start_nginx.sh', LF will be replaced by CRLF the next time
 Git touches it

Syed Abdul Rahim@LAPTOP-UDGP3QIT MINGW64 ~/Downloads/mywebapp2 (master)
$ git commit -m "application files"
[master 66c08df] application files
 5 files changed, 68 insertions(+)
 create mode 100644 Downloads/mywebapp2/appspec.yml
 create mode 100644 Downloads/mywebapp2/buildspec.yml
 create mode 100644 Downloads/mywebapp2/index.html
 create mode 100644 Downloads/mywebapp2/scripts/install_nginx.sh
 create mode 100644 Downloads/mywebapp2/scripts/start_nginx.sh

Syed Abdul Rahim@LAPTOP-UDGP3QIT MINGW64 ~/Downloads/mywebapp2 (master)
$ git push
```

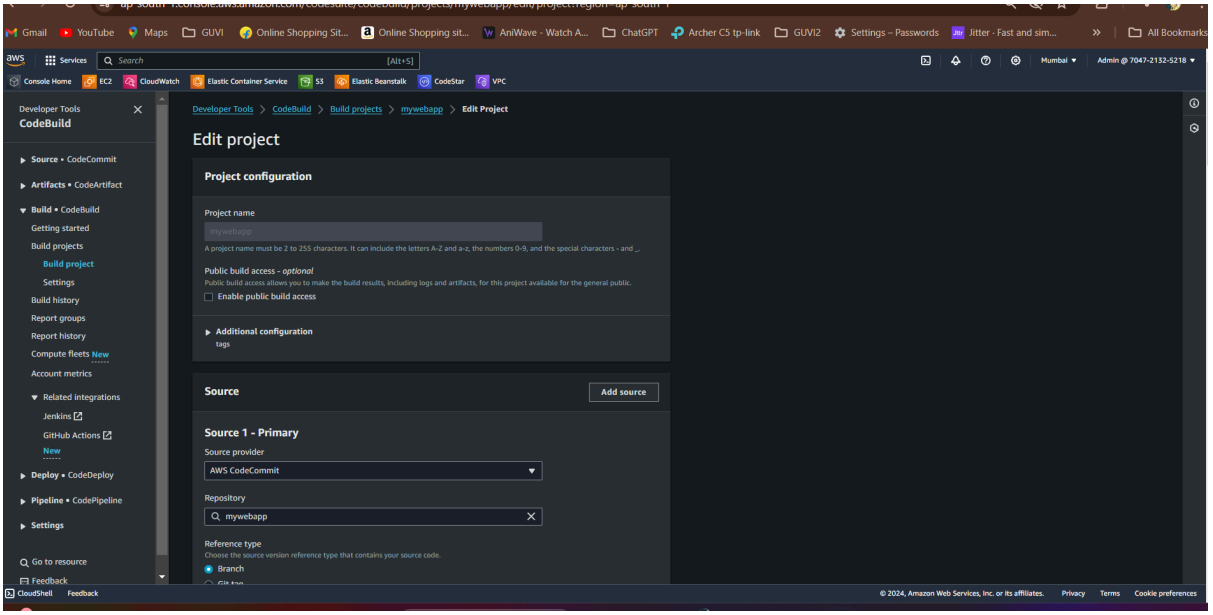
2.

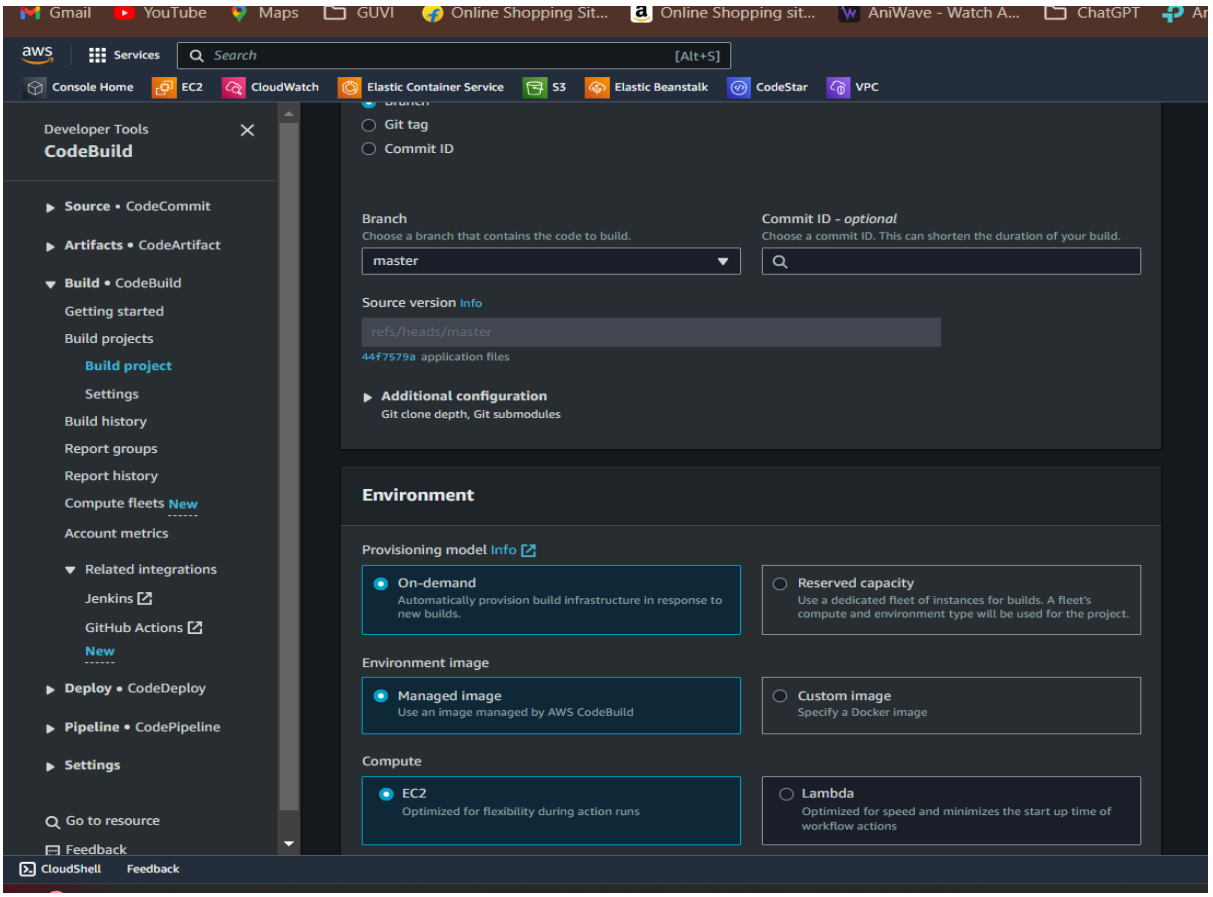
```
yasmin@abdul-personal demo % cd ..
yasmin@abdul-personal ~ % cd Downloads
yasmin@abdul-personal Downloads % git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/mywebapp
Cloning into 'mywebapp'...
Username for 'https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/mywebapp': yasmin-at-150878085644
Password for 'https://yasmin-at-150878085644@git-codecommit.ap-south-1.amazonaws.com/v1/repos/mywebapp':
warning: You appear to have cloned an empty repository.
yasmin@abdul-personal Downloads % cd mywebapp
yasmin@abdul-personal mywebapp % ls
yasmin@abdul-personal mywebapp % ls
appspec.yml     buildspec.yml   index.html      scripts
yasmin@abdul-personal mywebapp % git add .
yasmin@abdul-personal mywebapp % git commit -m "application files"
[master (root-commit) 378ebd9] application files
 5 files changed, 68 insertions(+)
 create mode 100644 appspec.yml
 create mode 100644 buildspec.yml
 create mode 100644 index.html
 create mode 100644 scripts/install_nginx.sh
 create mode 100644 scripts/start_nginx.sh
yasmin@abdul-personal mywebapp % git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 1.07 KiB | 546.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/mywebapp
 * [new branch]      master -> master
yasmin@abdul-personal mywebapp %
```

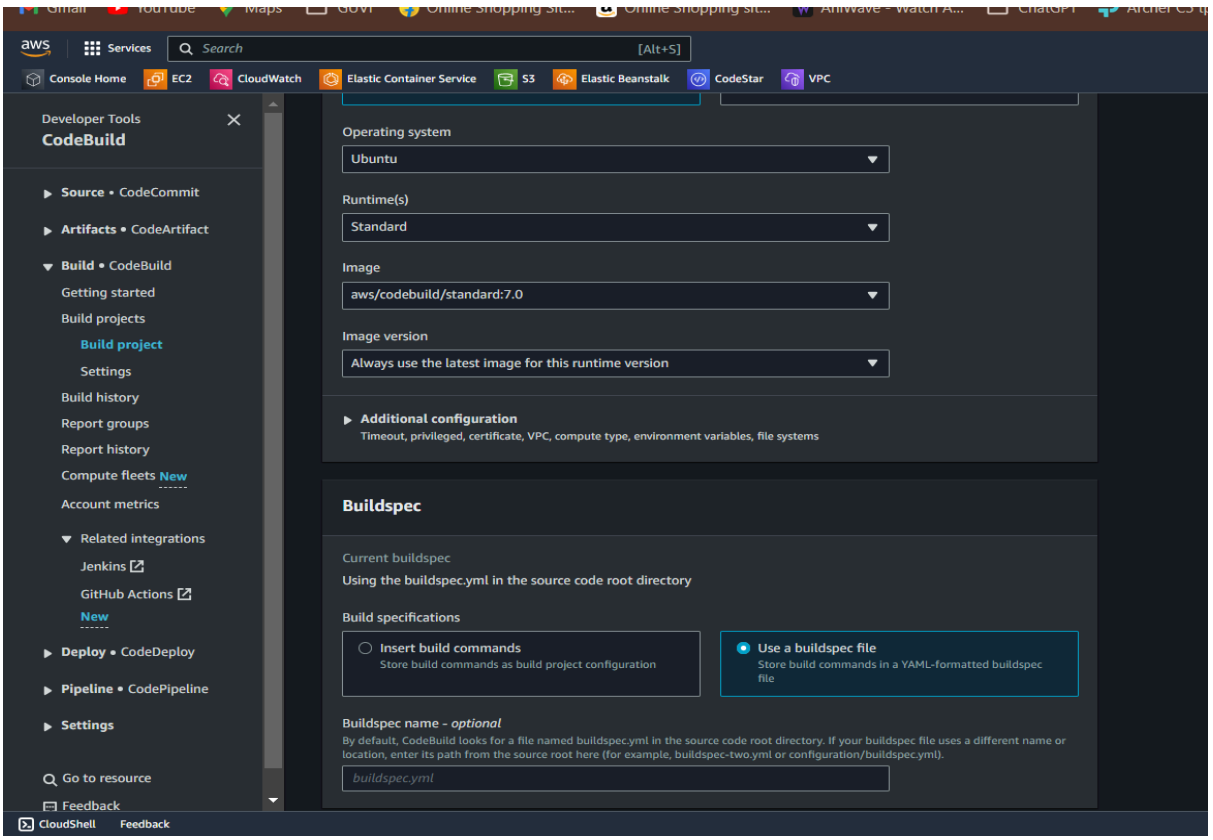
Step 4: Goto CodeCommit —> Check the CodeCommit repository —> It should have cloned the data from local git bash repository as per above **Step 3** - (**IAM**).

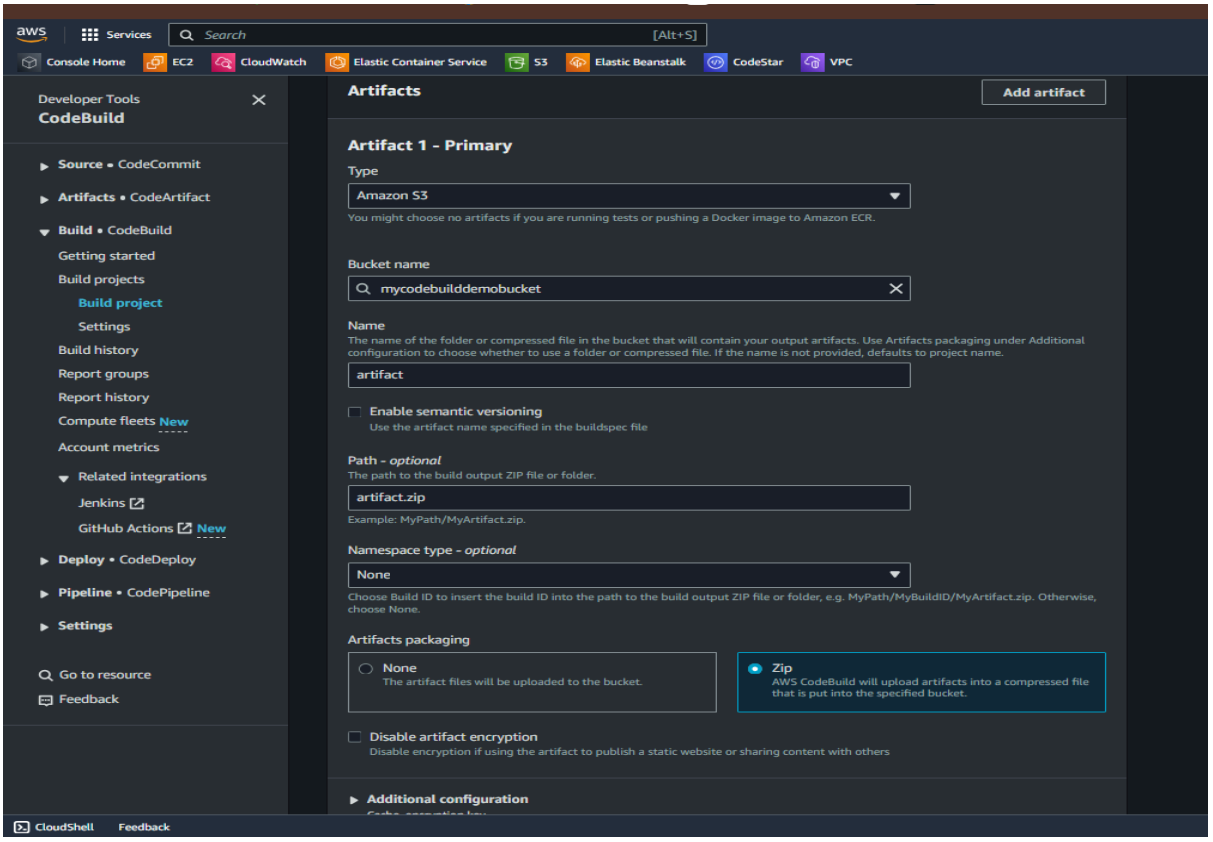


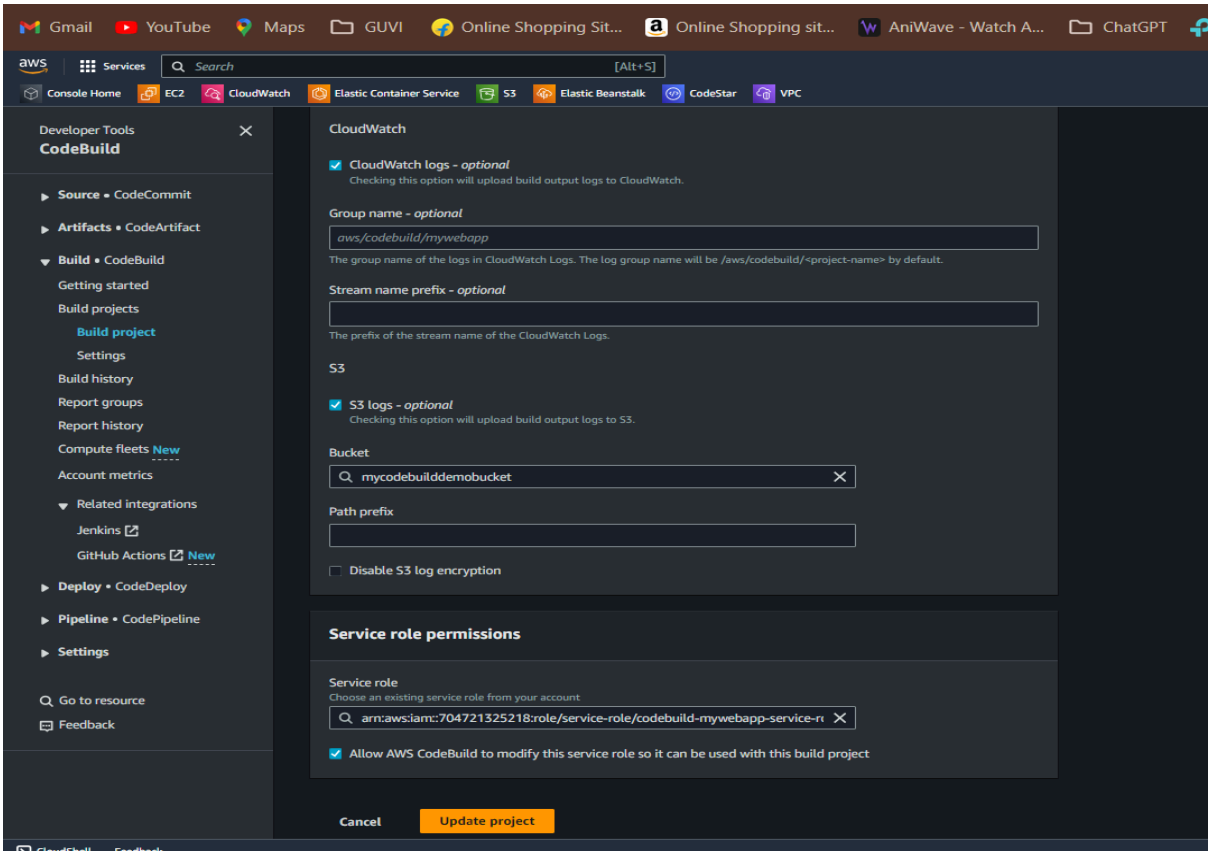
Step 1: Click on CodeBuild —> Create Project (Below is already filled data with **5 SS**) - (**CB**).




Gmail
YouTube
Maps
GUVI
Online Shopping Sit...
Online Shopping sit...
AniWave - Watch A...
ChatGPT
aws
Services
Search
[Alt+S]
Console Home
EC2
CloudWatch
Elastic Container Service
S3
Elastic Beanstalk
CodeStar
VPC
Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history
Compute fleets New
Account metrics
Related integrations
Jenkins
GitHub Actions
New
Deploy • CodeDeploy
Pipeline • CodePipeline
Settings
Go to resource
Feedback
CloudShell
Feedback
Git tag
Commit ID
Branch
Choose a branch that contains the code to build.
master
Commit ID - optional
Choose a commit ID. This can shorten the duration of your build.
Source version Info
refs/heads/master
44f7579a application files
Additional configuration
Git clone depth, Git submodules
Environment
Provisioning model Info
On-demand
Automatically provision build infrastructure in response to new builds.
Reserved capacity
Use a dedicated fleet of instances for builds. A fleet's compute and environment type will be used for the project.
Environment image
Managed image
Use an image managed by AWS CodeBuild
Custom image
Specify a Docker image
Compute
EC2
Optimized for flexibility during action runs
Lambda
Optimized for speed and minimizes the start up time of workflow actions

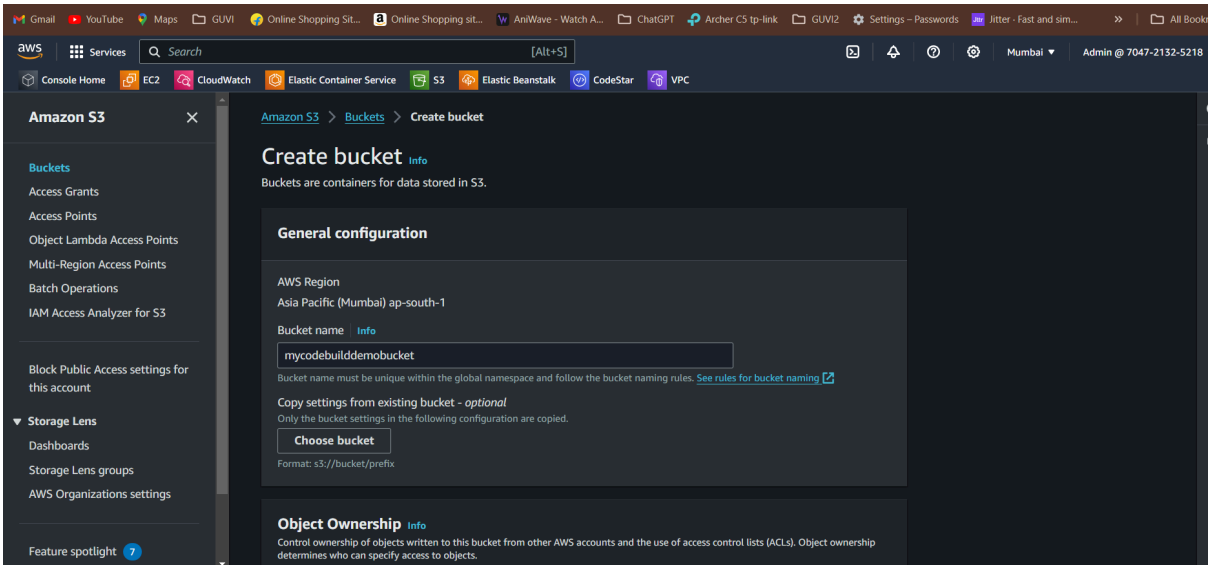

aws
Services
Search
[Alt+S]
Console Home
EC2
CloudWatch
Elastic Container Service
S3
Elastic Beanstalk
CodeStar
VPC
Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history
Compute fleets New
Account metrics
Related integrations
Jenkins
GitHub Actions
New
Deploy • CodeDeploy
Pipeline • CodePipeline
Settings
Go to resource
Feedback
CloudShell
Feedback
Operating system
Ubuntu
Runtime(s)
Standard
Image
aws/codebuild/standard:7.0
Image version
Always use the latest image for this runtime version
Additional configuration
Timeout, privileged, certificate, VPC, compute type, environment variables, file systems
Buildspec
Current buildspec
Using the buildspec.yml in the source code root directory
Build specifications
Insert build commands
Store build commands as build project configuration
Use a buildspec file
Store build commands in a YAML-formatted buildspec file
Buildspec name - optional
By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).
buildspec.yml

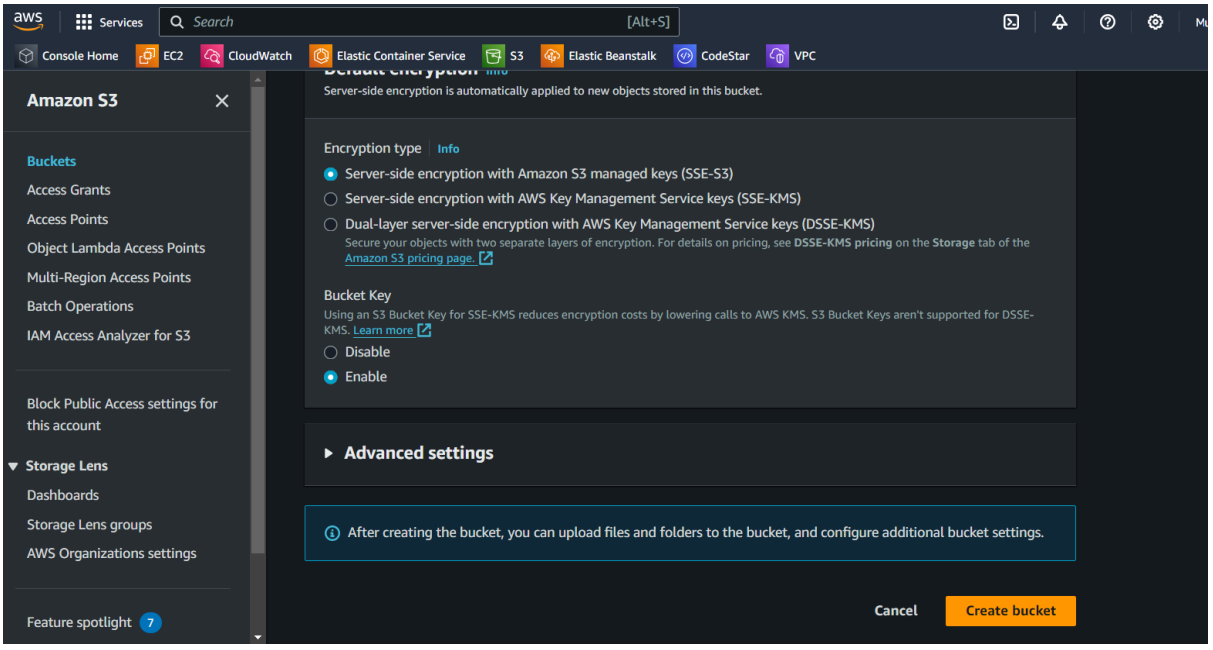

Services
Search
[Alt+S]
Console Home
EC2
CloudWatch
Elastic Container Service
S3
Elastic Beanstalk
CodeStar
VPC
Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history
Compute fleets New
Account metrics
Related integrations
Jenkins
GitHub Actions New
Deploy • CodeDeploy
Pipeline • CodePipeline
Settings
Go to resource
Feedback
Artifacts
Add artifact
Artifact 1 - Primary
Type
Amazon S3
You might choose no artifacts if you are running tests or pushing a Docker image to Amazon ECR.
Bucket name
mycodebuilddemobucket
Name
The name of the folder or compressed file in the bucket that will contain your output artifacts. Use Artifacts packaging under Additional configuration to choose whether to use a folder or compressed file. If the name is not provided, defaults to project name.
artifact
Enable semantic versioning
Use the artifact name specified in the buildspec file
Path - optional
The path to the build output ZIP file or folder.
artifact.zip
Example: MyPath/MyArtifact.zip.
Namespace type - optional
None
Choose Build ID to insert the build ID into the path to the build output ZIP file or folder, e.g. MyPath/MyBuildID/MyArtifact.zip. Otherwise, choose None.
Artifacts packaging
None
The artifact files will be uploaded to the bucket.
Zip
AWS CodeBuild will upload artifacts into a compressed file that is put into the specified bucket.
Disable artifact encryption
Disable encryption if using the artifact to publish a static website or sharing content with others
Additional configuration
CloudShell
Feedback


Gmail
YouTube
Maps
GUVI
Online Shopping Sit...
Online Shopping sit...
AniWave - Watch A...
ChatGPT
Services
Search
[Alt+S]
Console Home
EC2
CloudWatch
Elastic Container Service
S3
Elastic Beanstalk
CodeStar
VPC
Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history
Compute fleets New
Account metrics
Related integrations
Jenkins
GitHub Actions New
Deploy • CodeDeploy
Pipeline • CodePipeline
Settings
Go to resource
Feedback
CloudWatch
CloudWatch logs - optional
Checking this option will upload build output logs to CloudWatch.
Group name - optional
aws/codebuild/mywebapp
The group name of the logs in CloudWatch Logs. The log group name will be /aws/codebuild/<project-name> by default.
Stream name prefix - optional
The prefix of the stream name of the CloudWatch Logs.
S3
S3 logs - optional
Checking this option will upload build output logs to S3.
Bucket
mycodebuilddemobucket
Path prefix
Disable S3 log encryption
Service role permissions
Service role
Choose an existing service role from your account
arn:aws:iam::704721325218:role/service-role/codebuild-mywebapp-service-r
Allow AWS CodeBuild to modify this service role so it can be used with this build project
Cancel
Update project

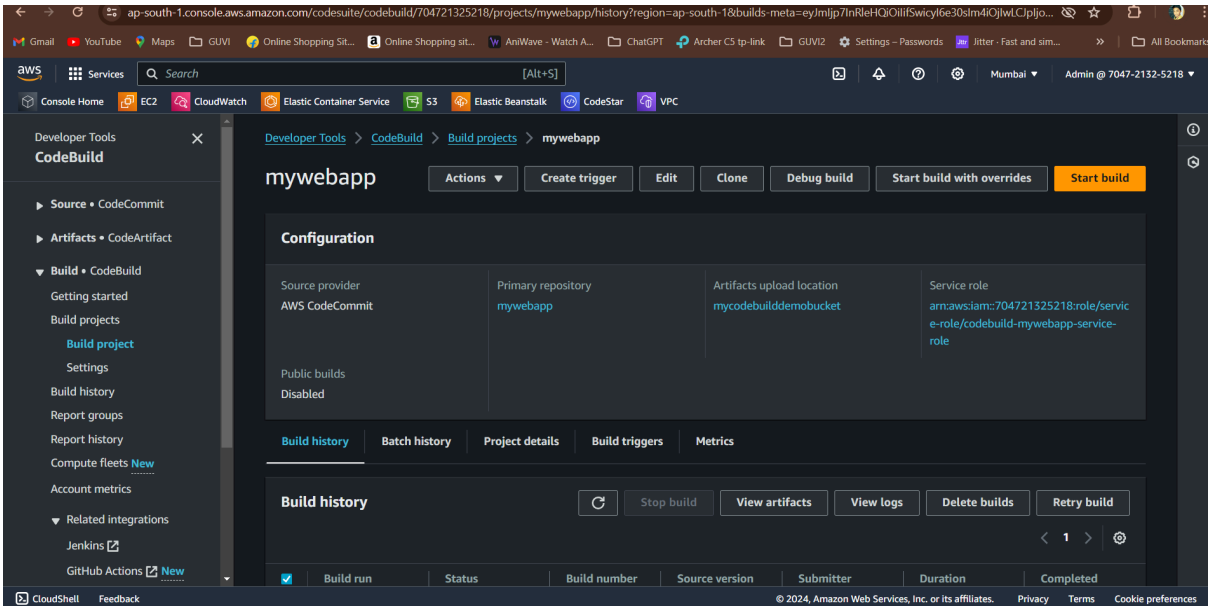
Step 2: Click on Create build project for above SS - (**CB**).

Step 1: Create a **s3 Bucket** —> Just give the Bucket name —-> Create Bucket - (**s3 B**).

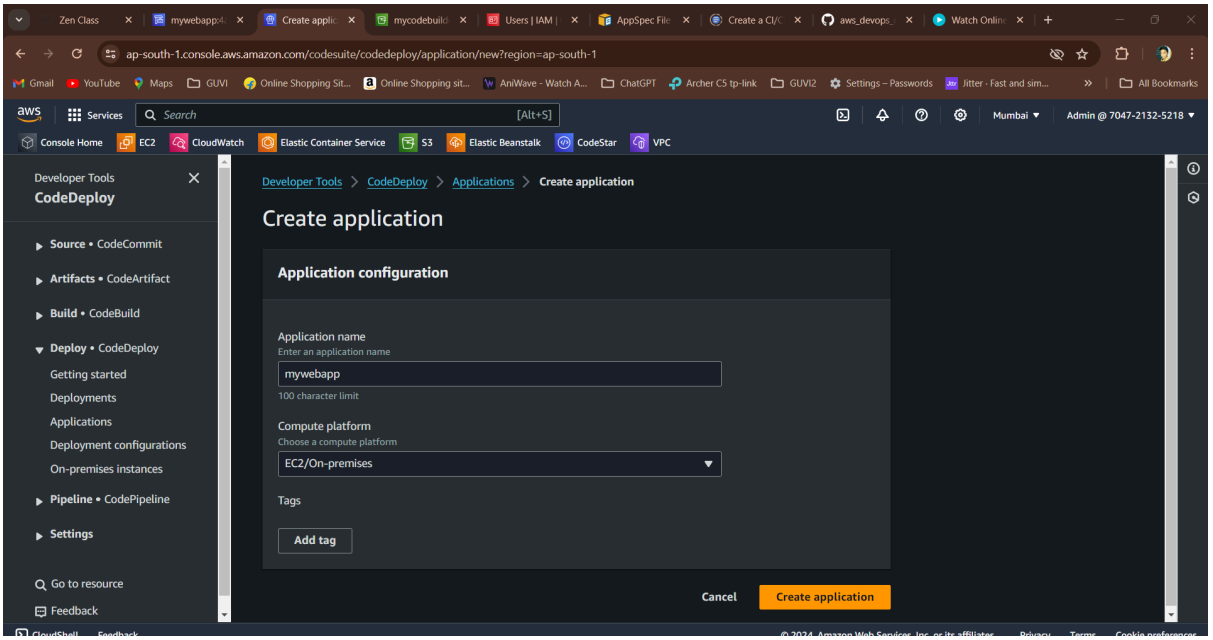




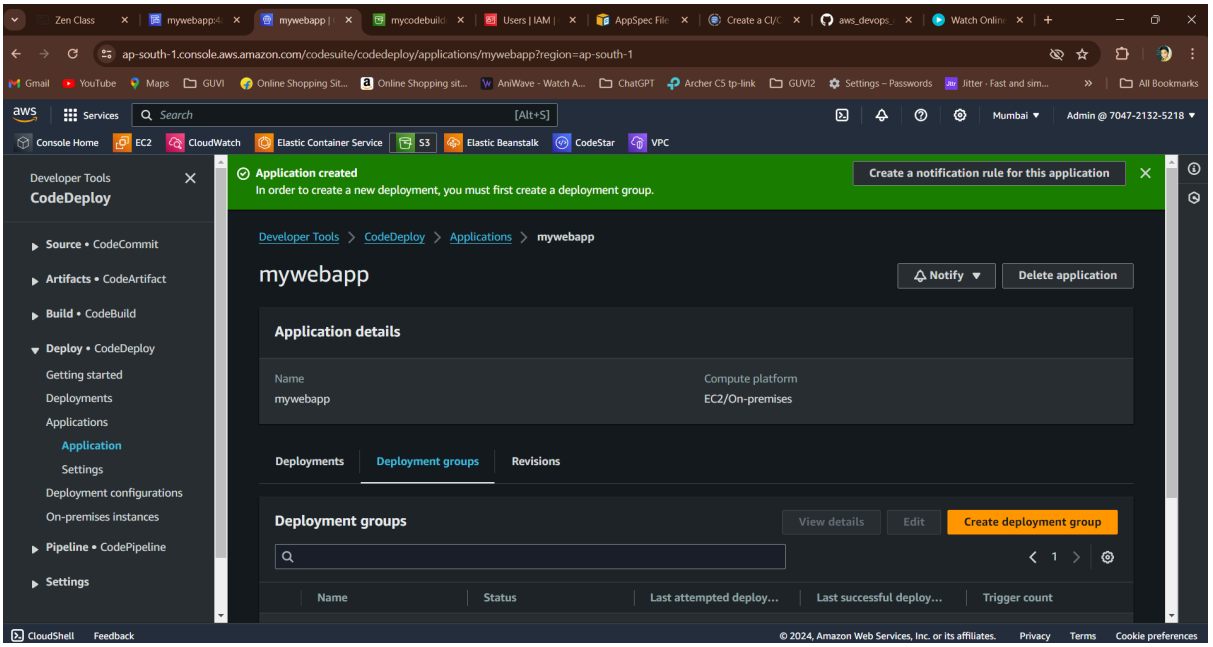
Step 2: Click on Start Build - (**s3 B**).



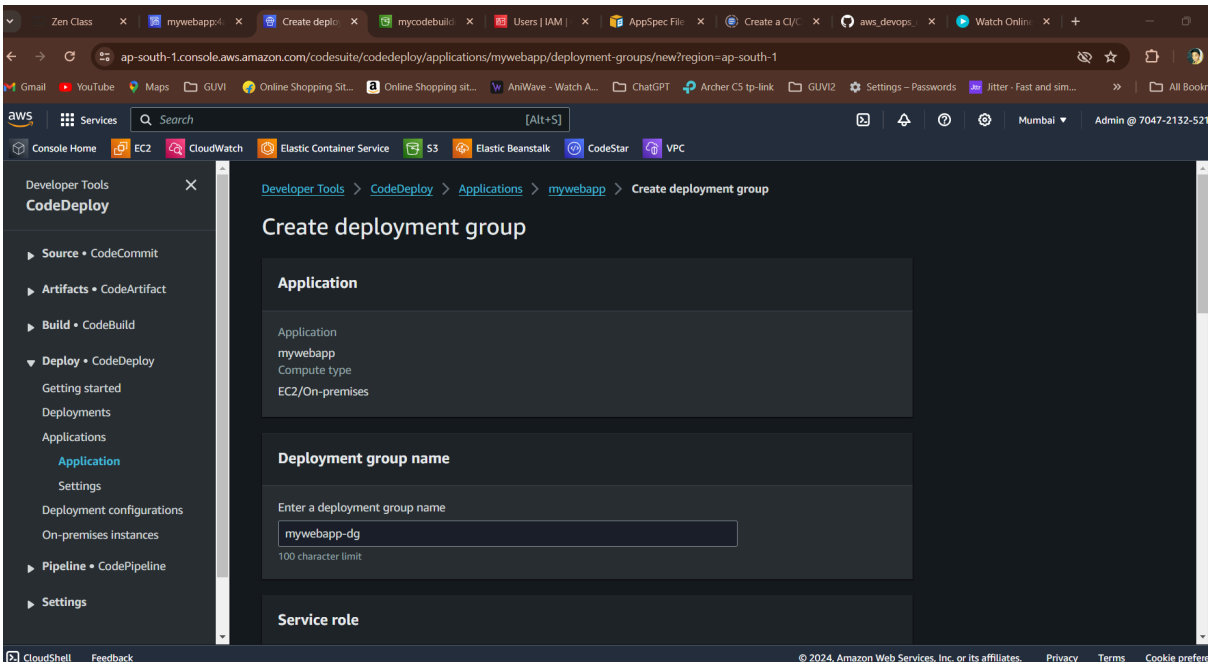
Step 1: Create **CodeDeploy** —> Create application - (**CD**).



Created application.

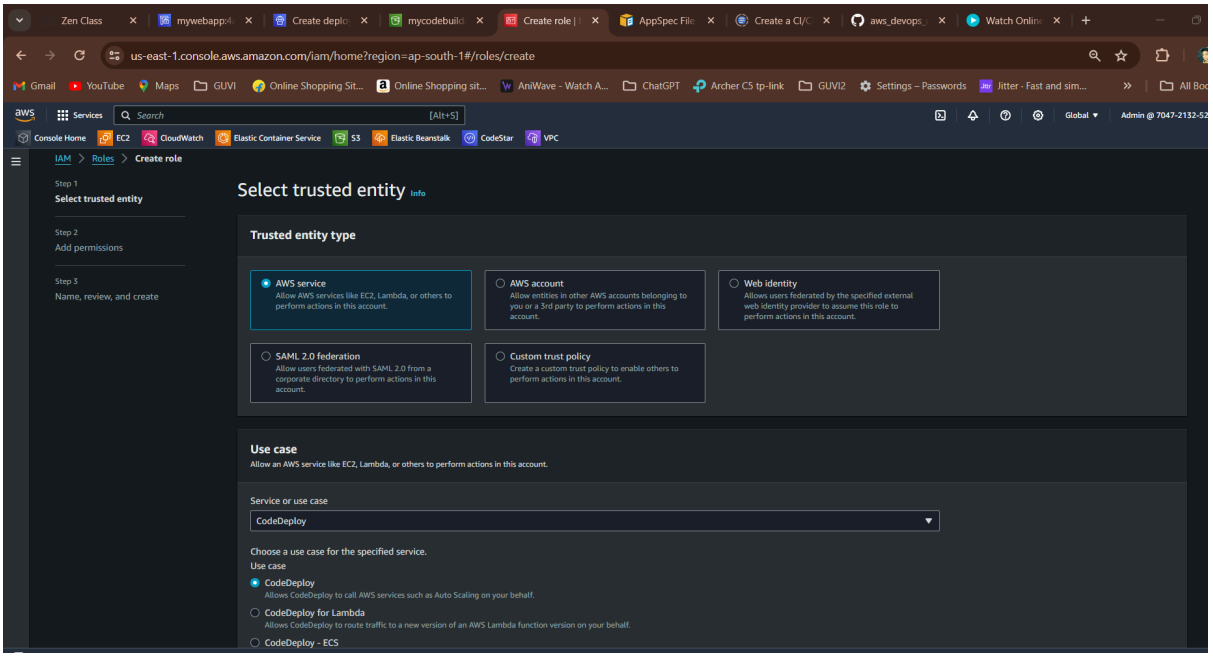


Step 2: Deploy the above application in EC2 —> Click on Create deployment group - (**CD**).

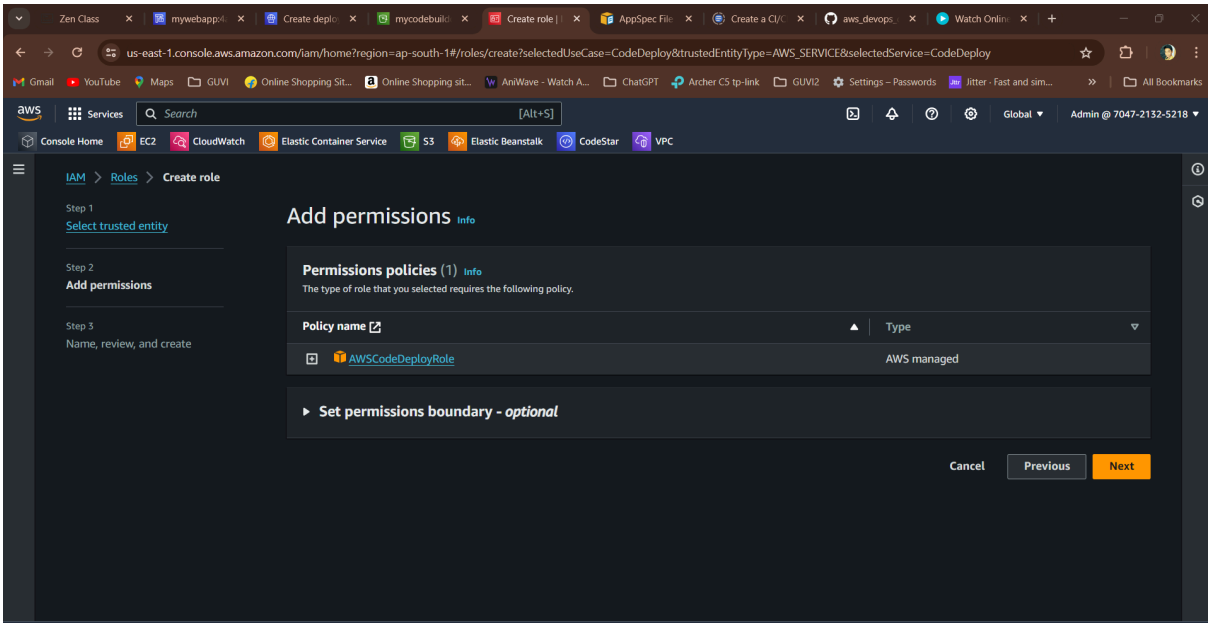


Step 1: In Service role (There won't be any default code deploy role options, so we create it manualy) —> Goto IAM —> Click on Roles —> Create Role - (**CD**).

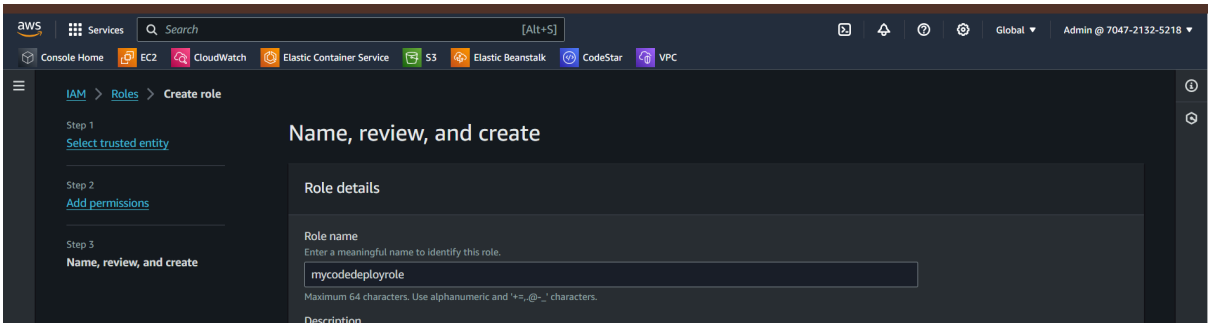
Step 2: Create IAM role (**01**) and fill as per requirements like below —> Click next - (**CD**).



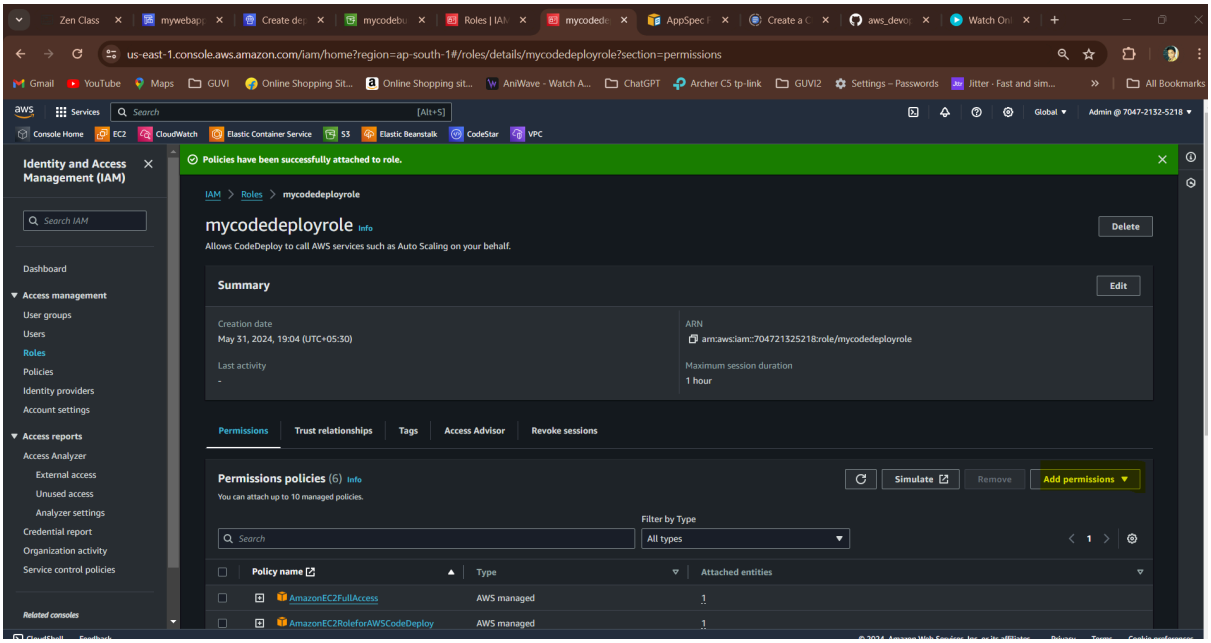
Step 3: As it is for codedeploy we make no changes - (**CD**).

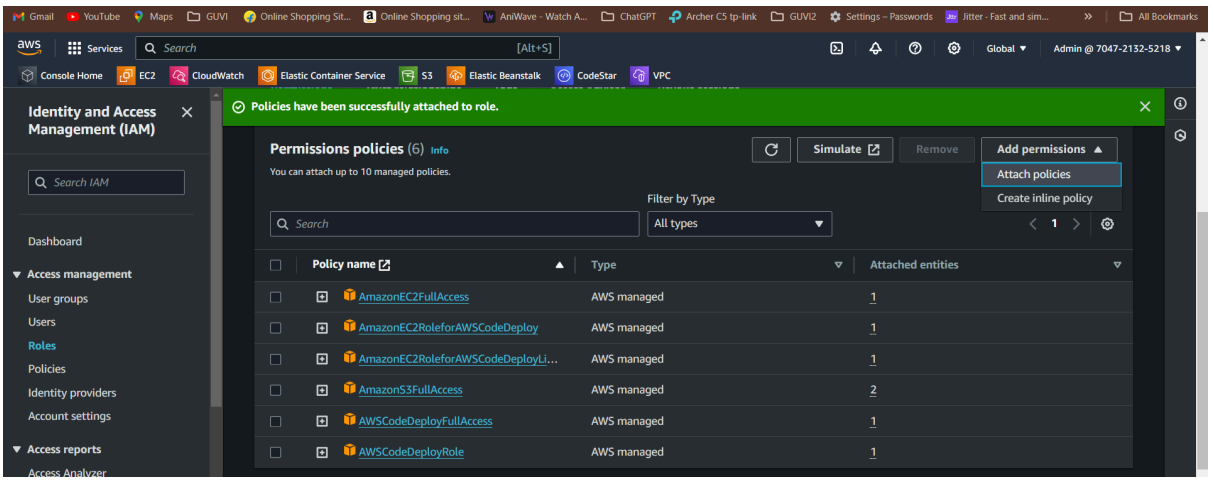


Step 4: Just give the Role name —> Click Create Role(Make no other changes) - (**CD**).

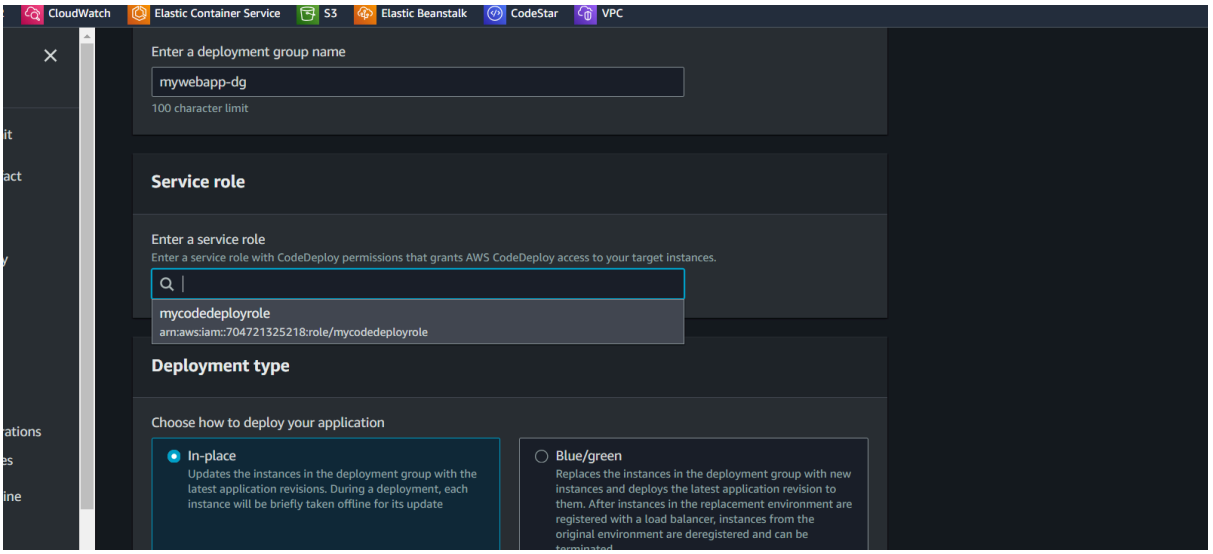


Step 5: As more permissions need to be added —> Click Add permissions —> Select Attach policies —> And select the below permissions - (**CD**).

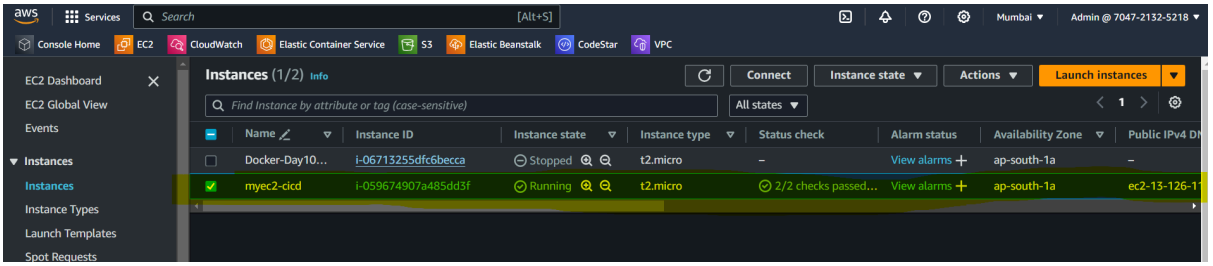




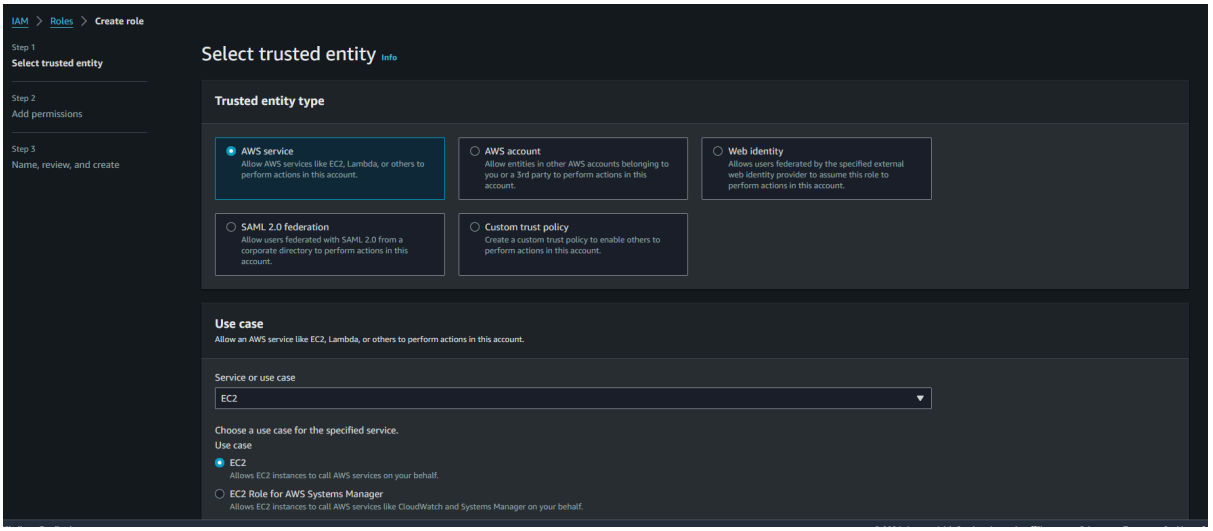
Step 3: Select the above Service role created from the option - (**CD**).



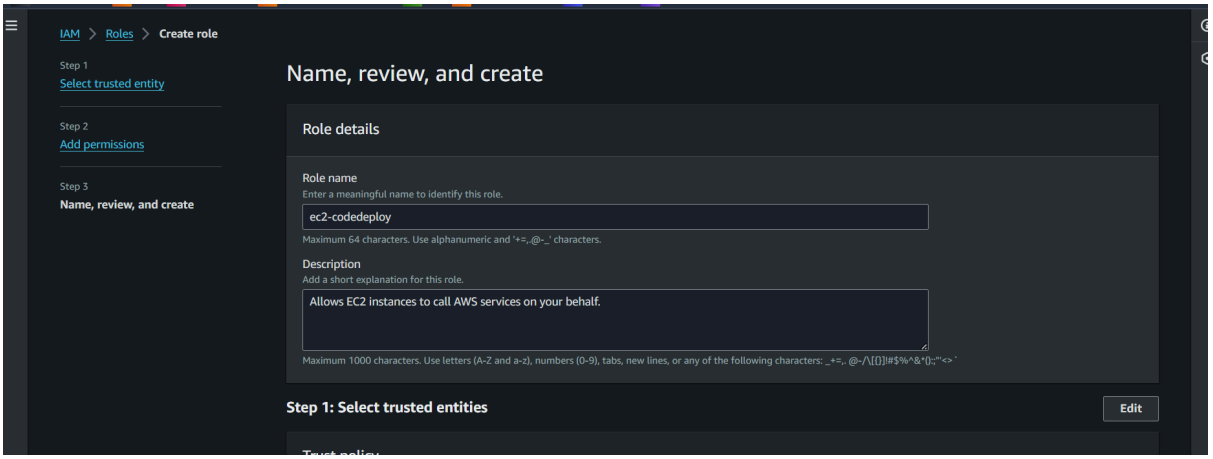
Step 4: Select Amazon EC2 Instance —> Select an EC2 instance (If not create one manualy with *version* **Ubuntu Server 22.04 LTS**) - (**CD**).

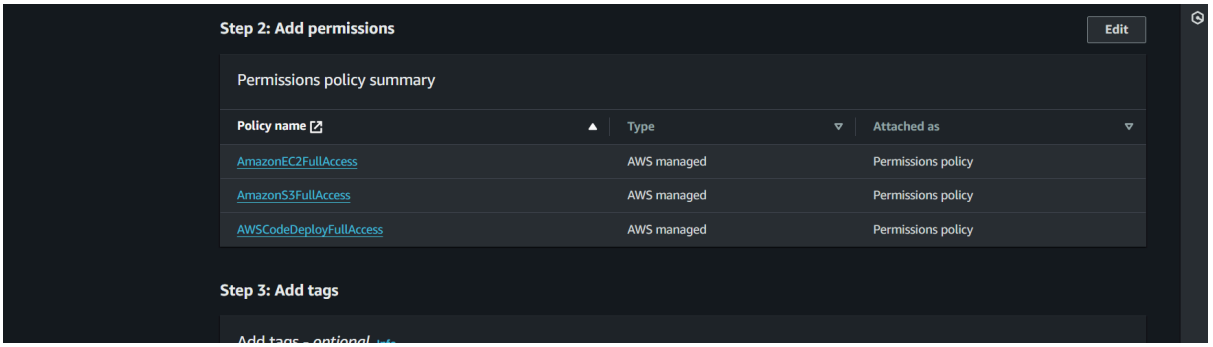


Step 1: Create a IAM role (**02**) for the above EC2 instance as per below —> Click next - (**CD**).

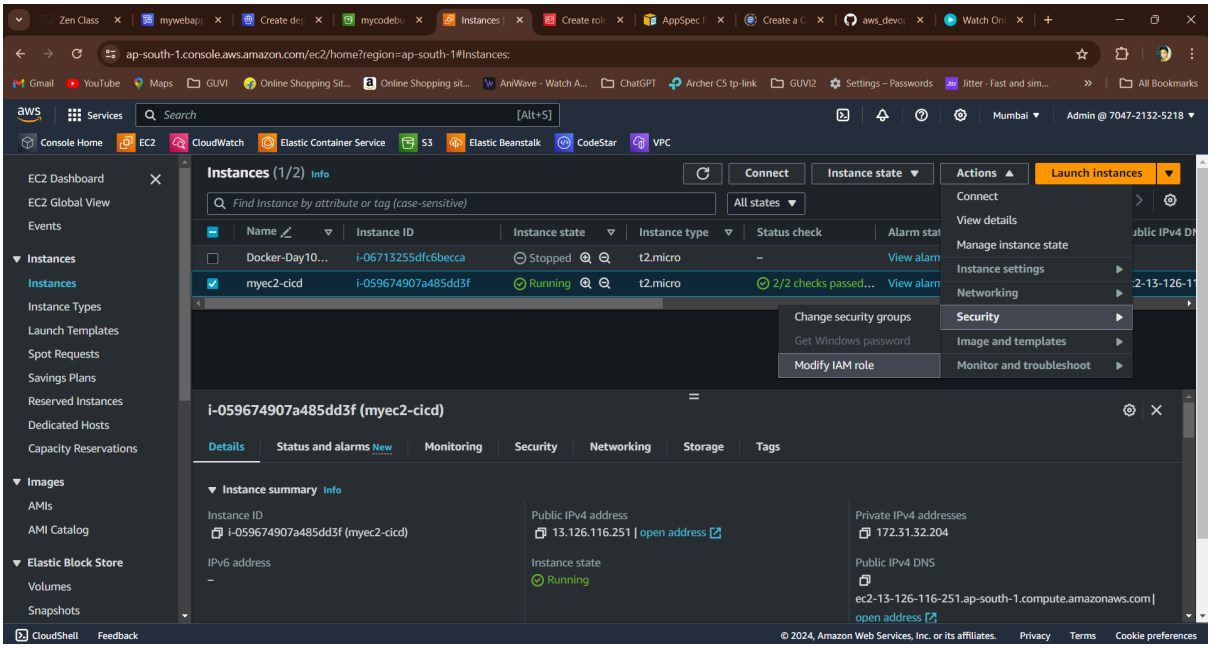


Step 2: As this is for EC2 instance, name and permissions added —> Click Create role - (**CD**).

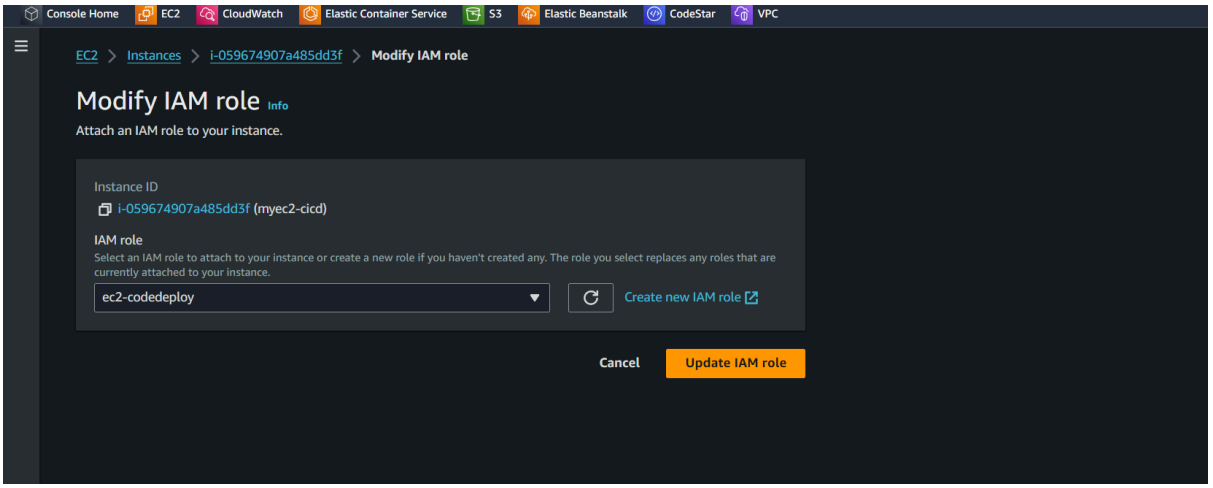




Step 5: How to attach EC2 IAM role —> Action —> Security —> Modify IAM role - (**CD**).



Step 6: Select the IAM role that was created for EC2 instance —> Click Update IAM role - (**CD**).

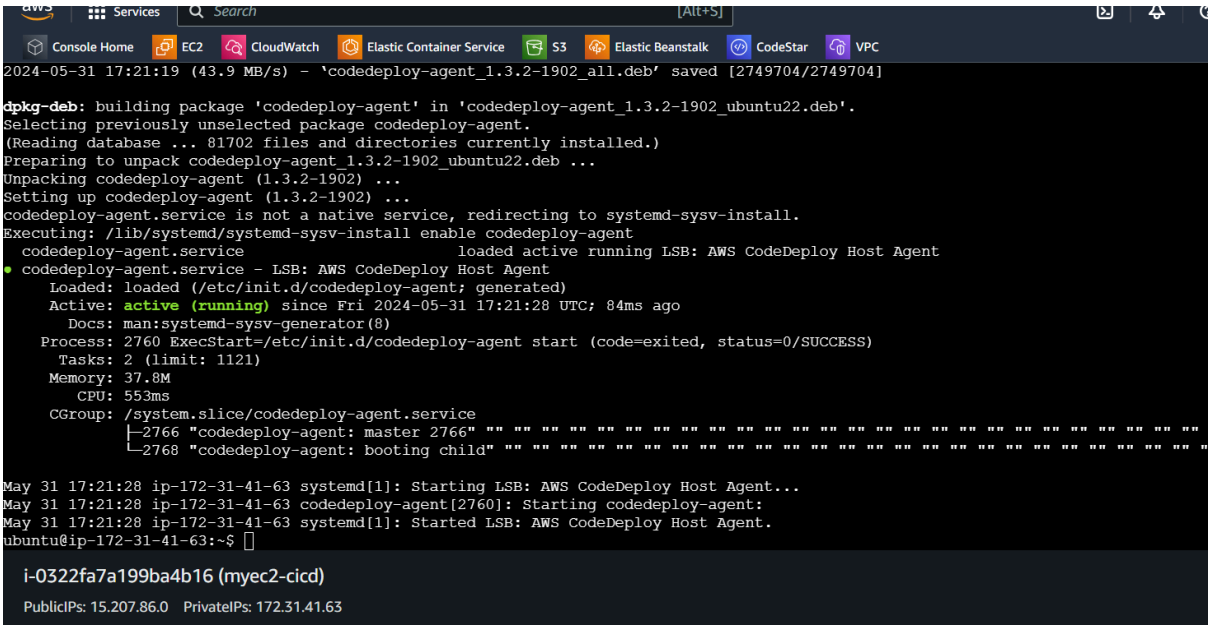


Step 7: Launch the created EC2 instance —> Install a CodeDeploy agent for this EC2 instance —> throught **one** of the below link —> After installation is complete it should show as per below ***active (running)*** - (**CD**).

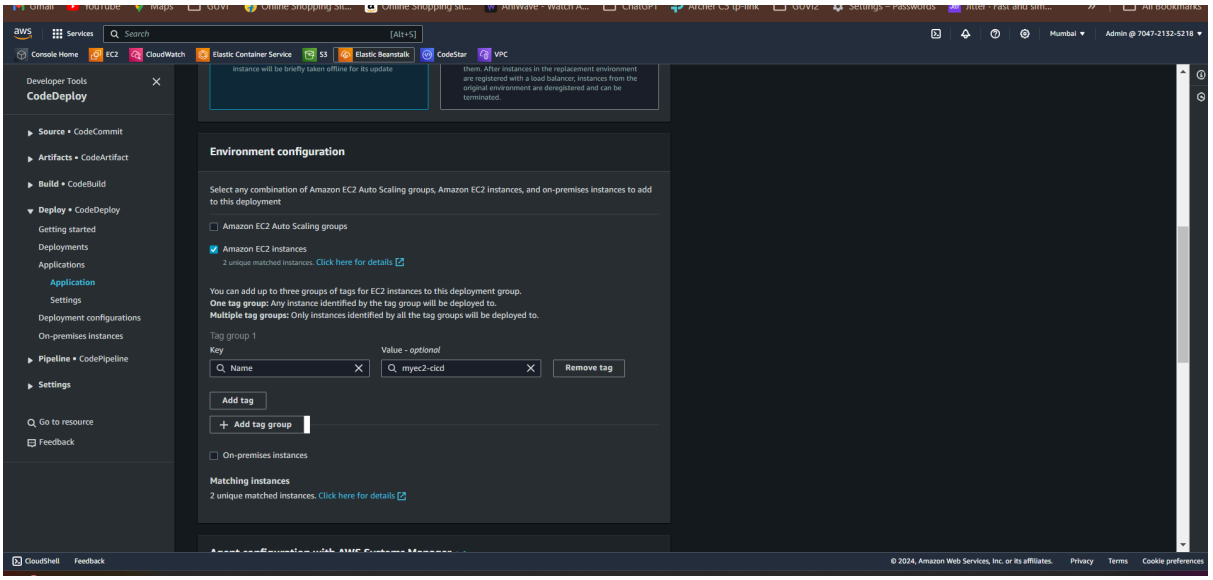
https://docs.aws.amazon.com/codedeploy/latest/userguide/codedeploy-agent-operations-install-ubuntu.html

OR

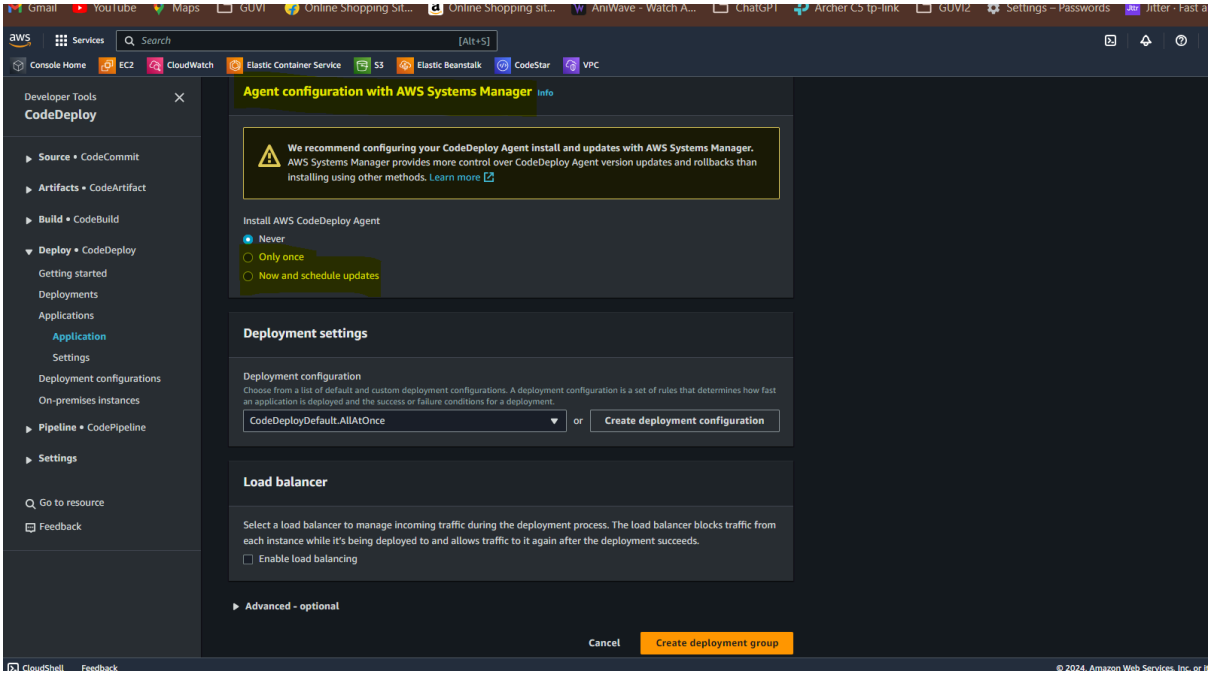
https://docs.google.com/document/d/1PWghD1uklVI37KiR92n9ln8WEUyO6bbbkXUU_ZtC13E/edit?usp=sharing



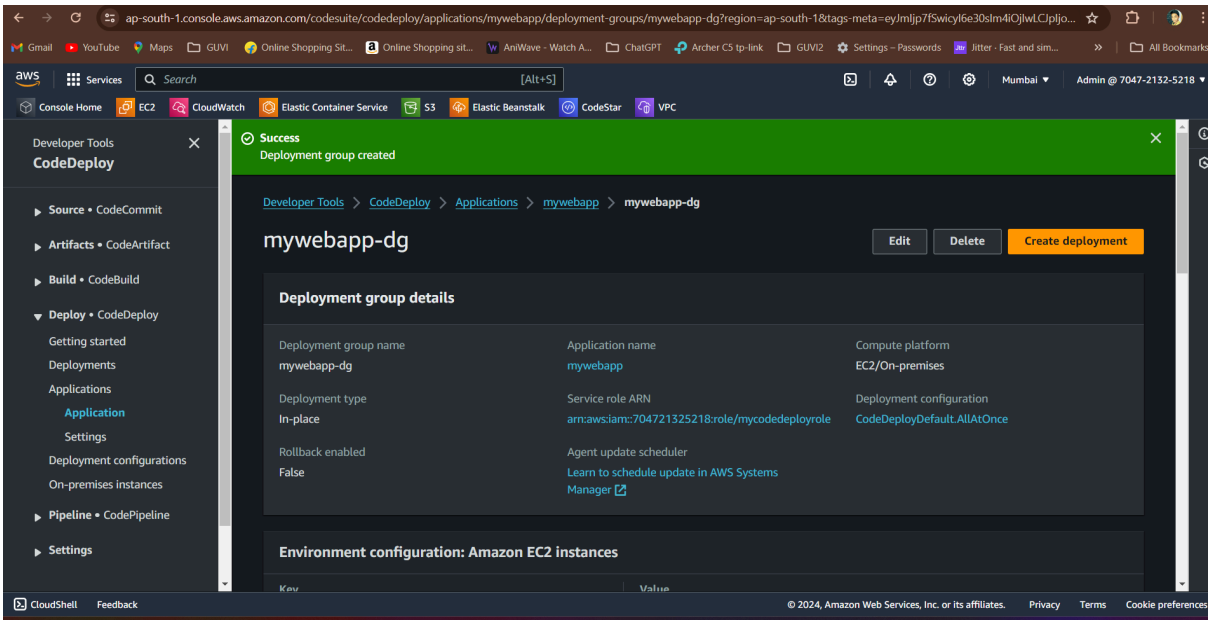
Step 8: Select the EC2 instance that was created with role - (**CD**).



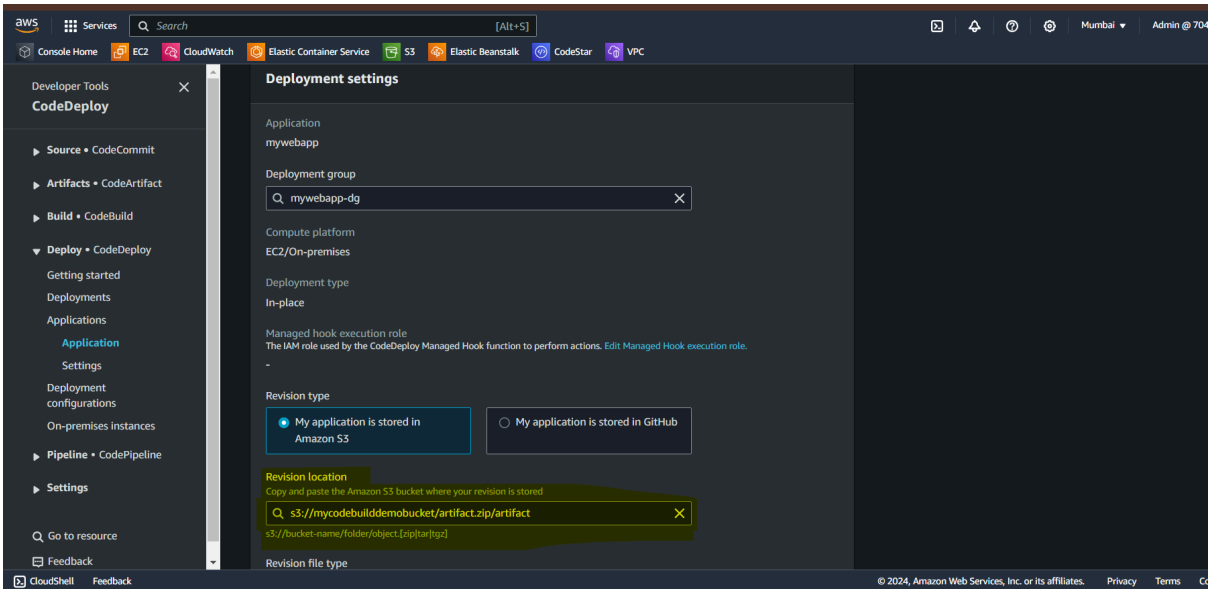
Step 9: We can also install CodeDeploy agent directly as per below, however if you want to manually install CodeDeploy agent then follow the link in ***Step 7*** —> Uncheck Load balancer —> Click Create deployment group - (**CD**).



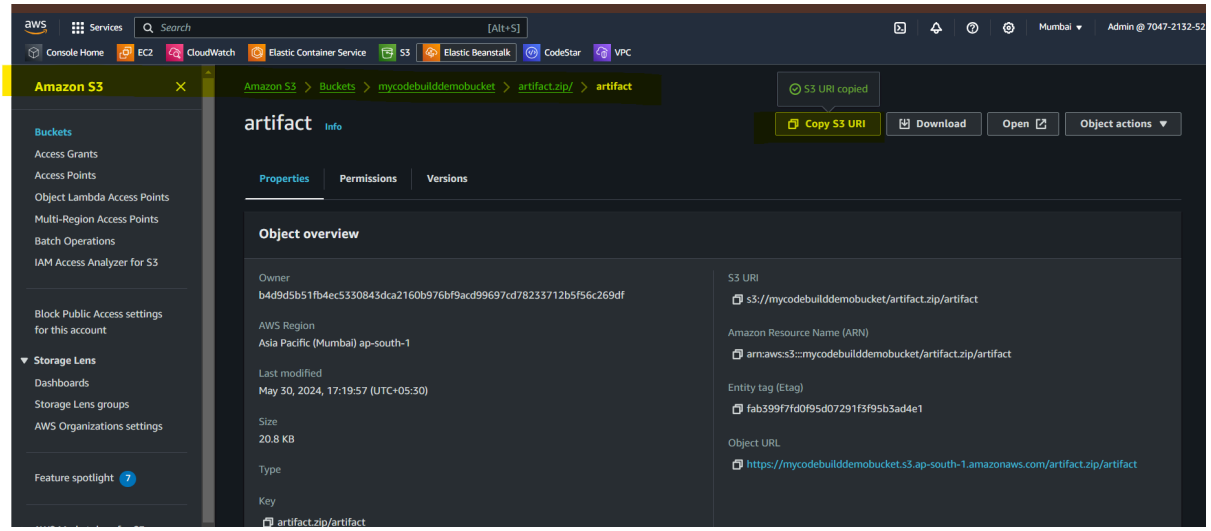
Step 10: Deployment group created —> Click Create deployment - (**CD**).



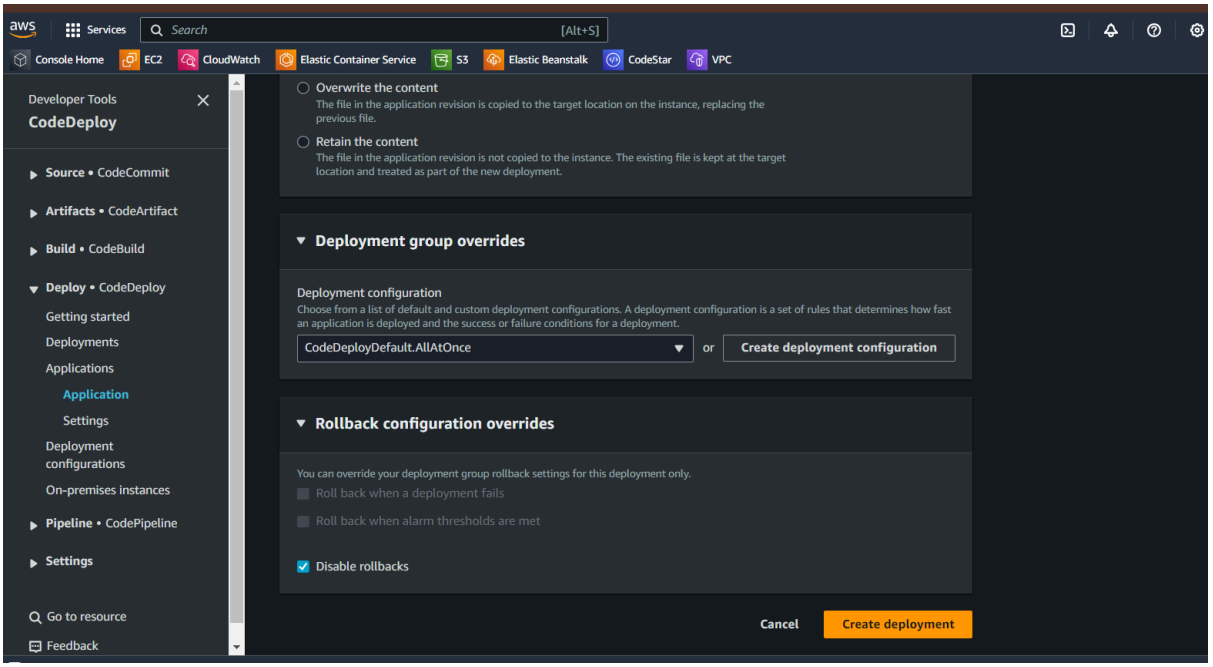
Step 11: Fill the Revision location with ***s3 URL link*** - (**CD**).



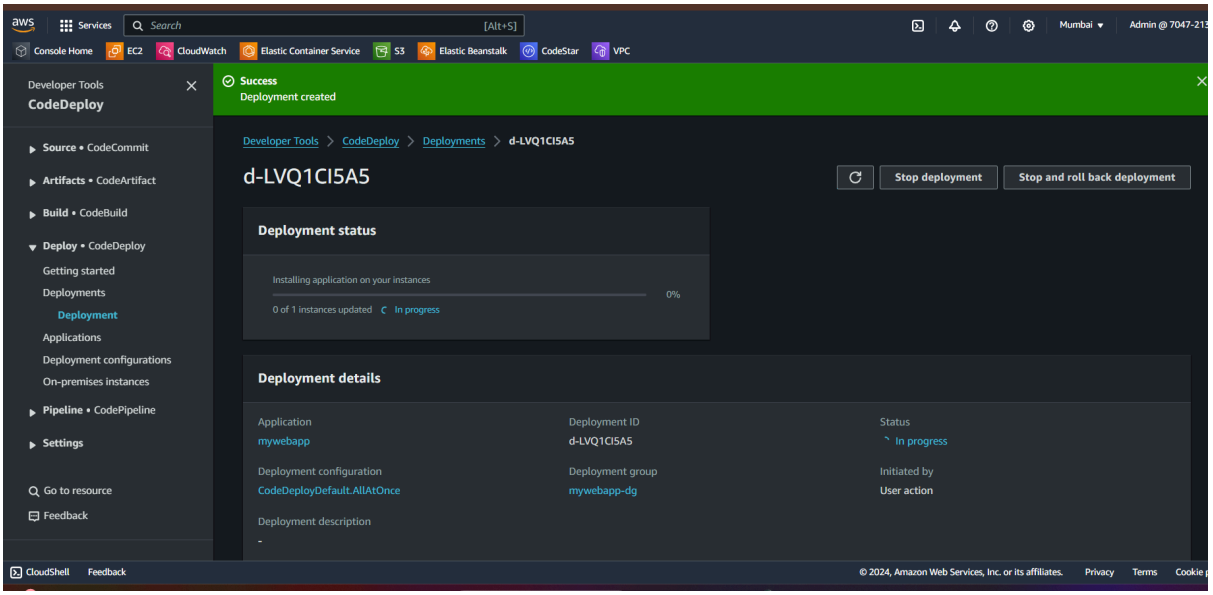
You can find the ***s3 URL link*** in below location.



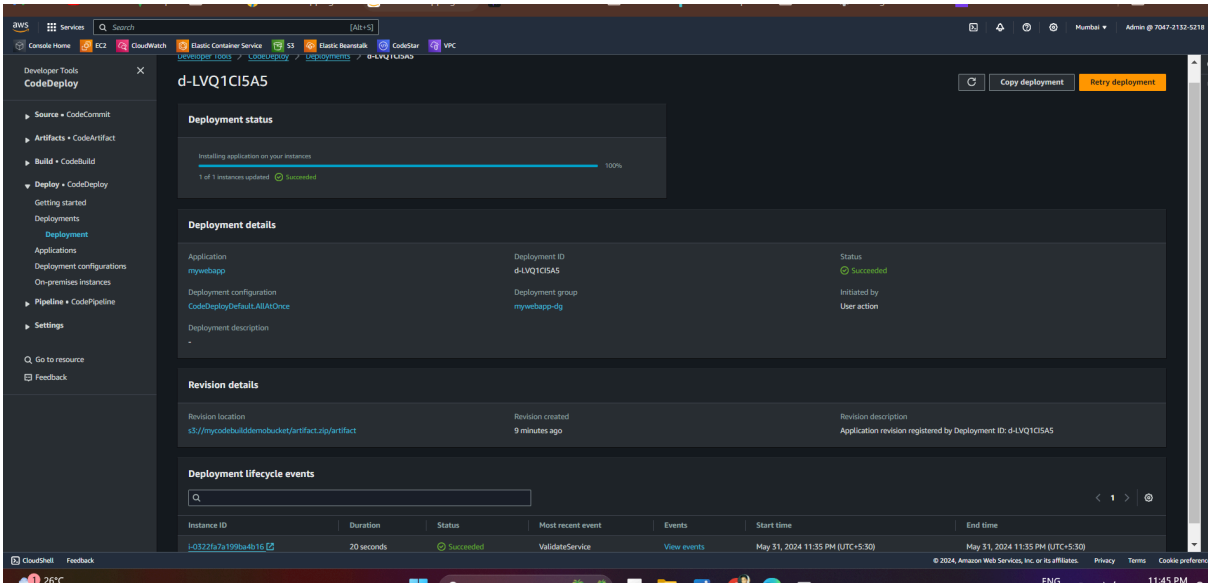
Step 12: Leave the rest as default —> Click Create deployment - (**CD**).



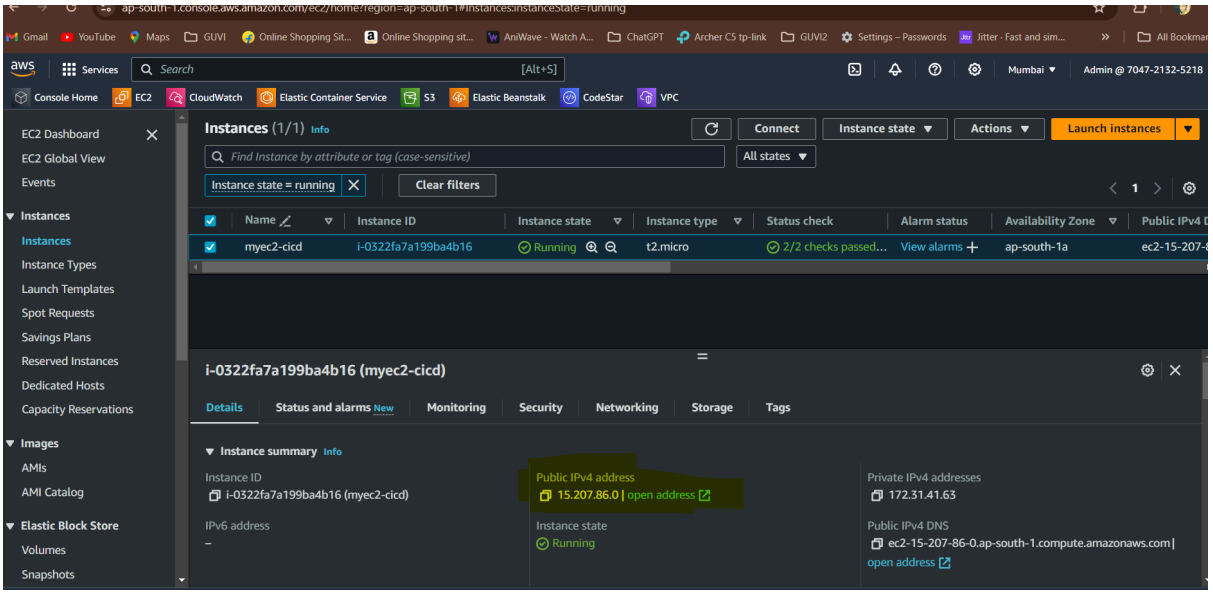
Step 13: CodeDeploy getting deployed - (**CD**).



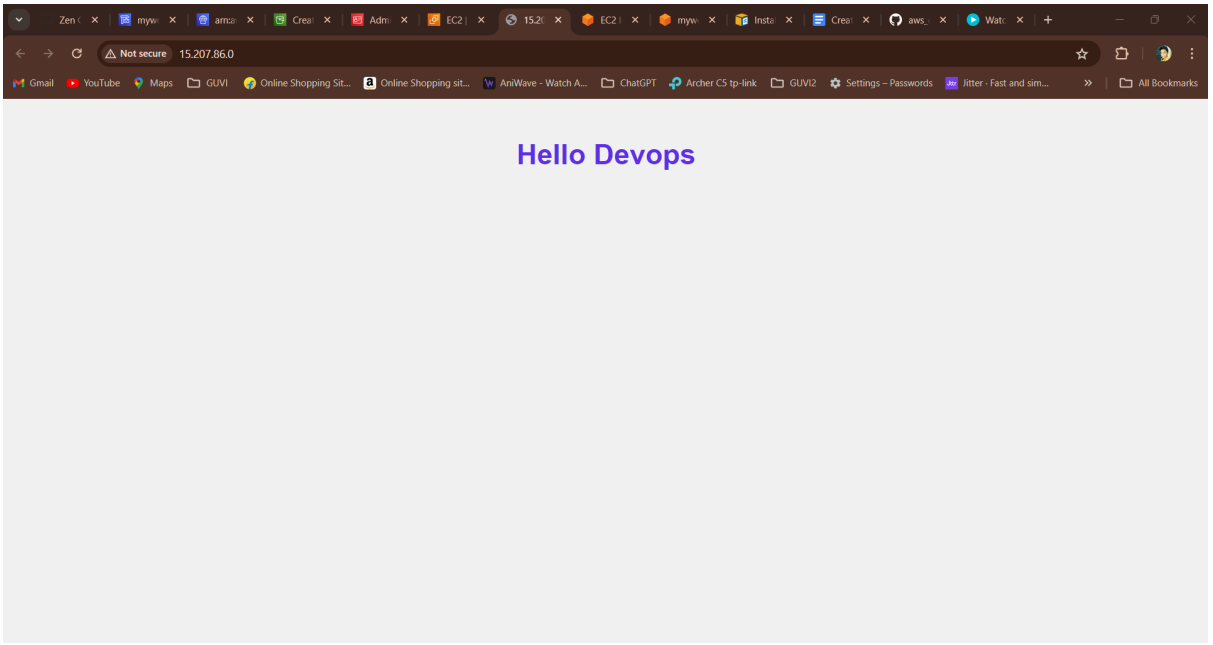
Deployment Completed Successfully.



Goto EC2 instance —> Click on the Public-Open address.

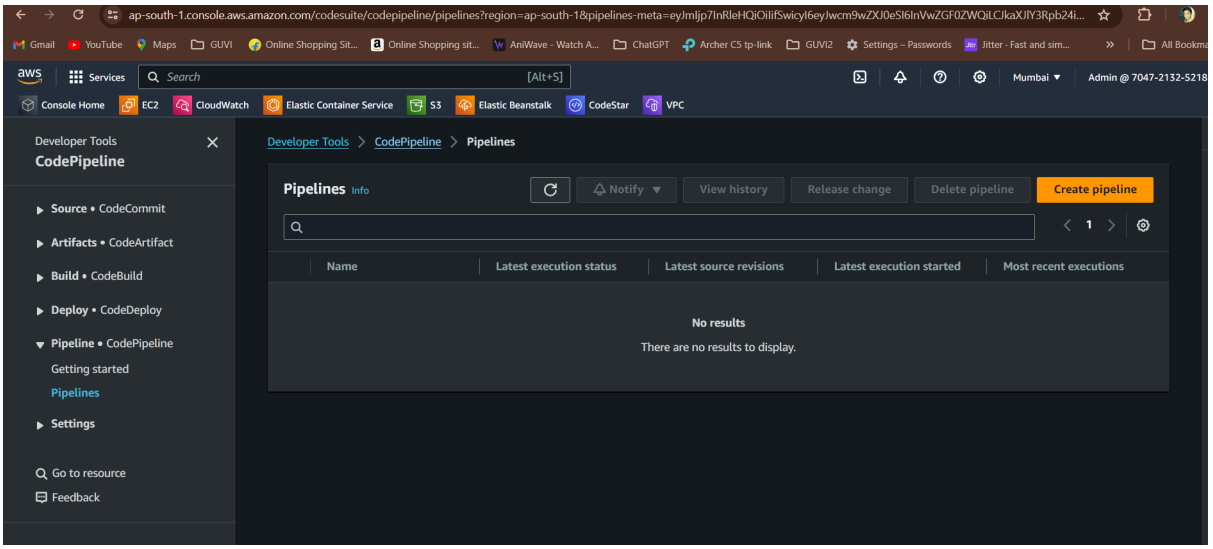


Output:

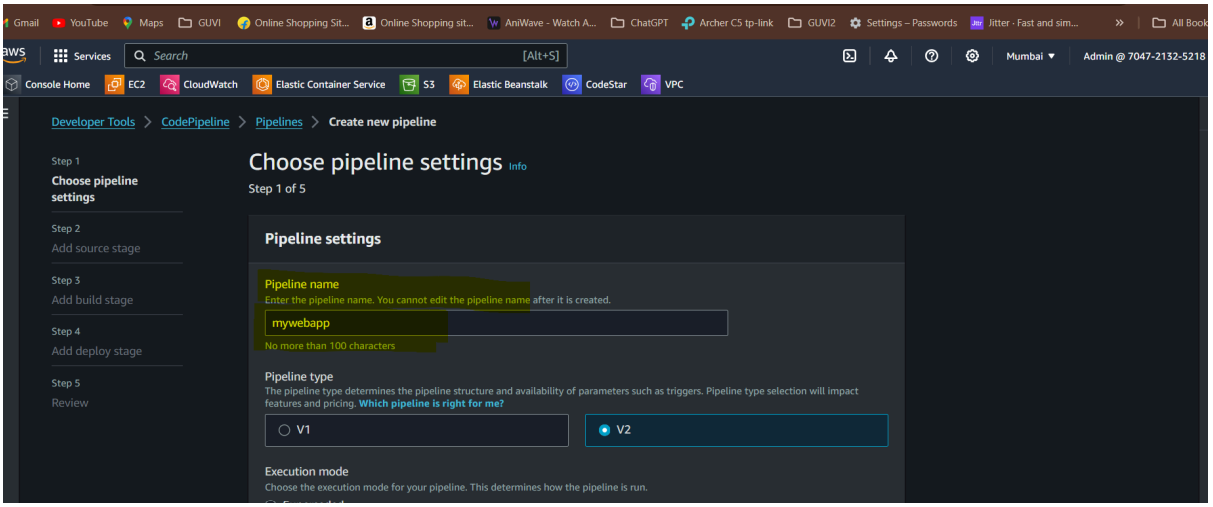


Note: Make sure the port no. is 80 in security —> Security group (If not, then add inbound rule 80 port with HTTP)

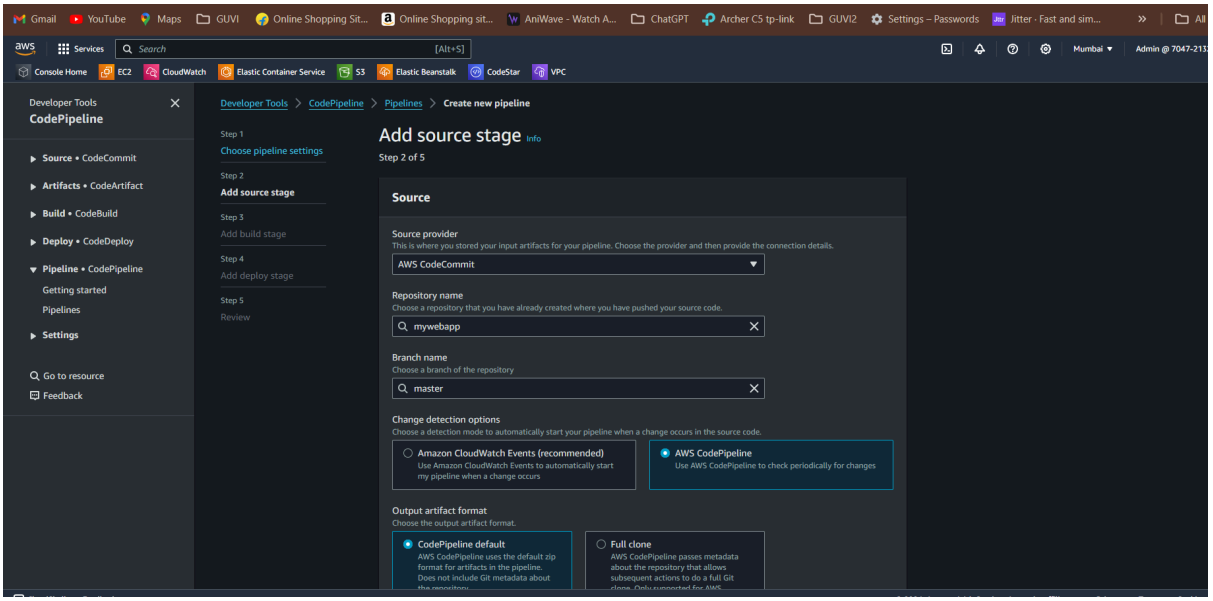
Step 1: Create a **Pipline** so that any changes to you main code will automatically be saved and updated - (**PI**).



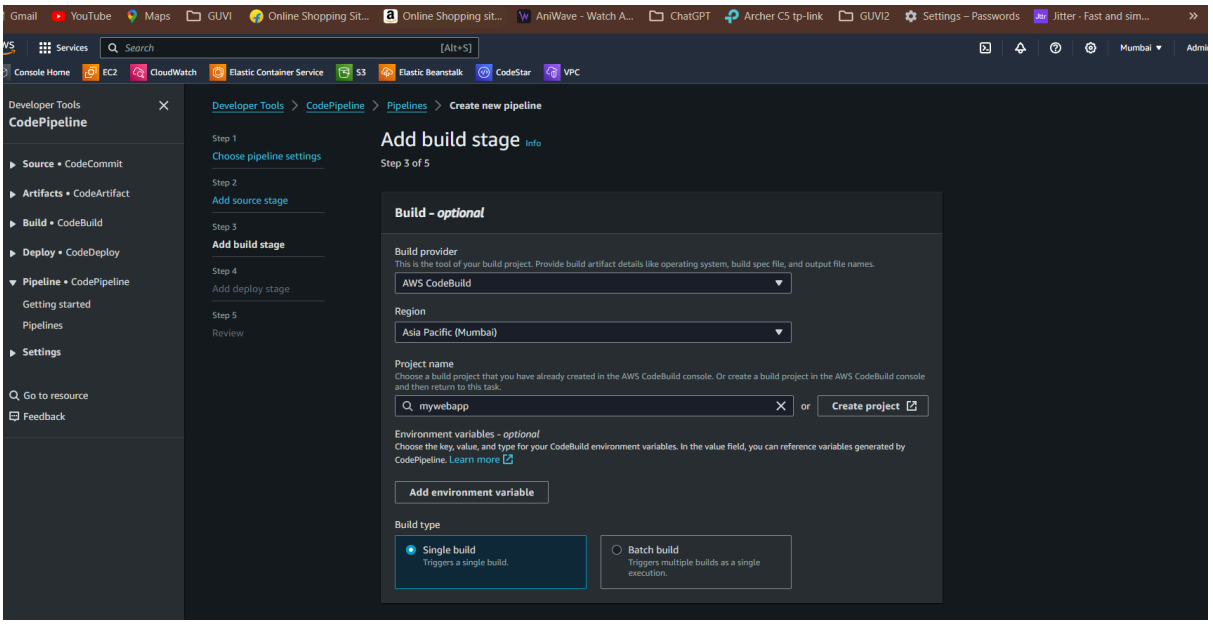
Step 2: Just give the Pipeline name —> next - (**PI**).



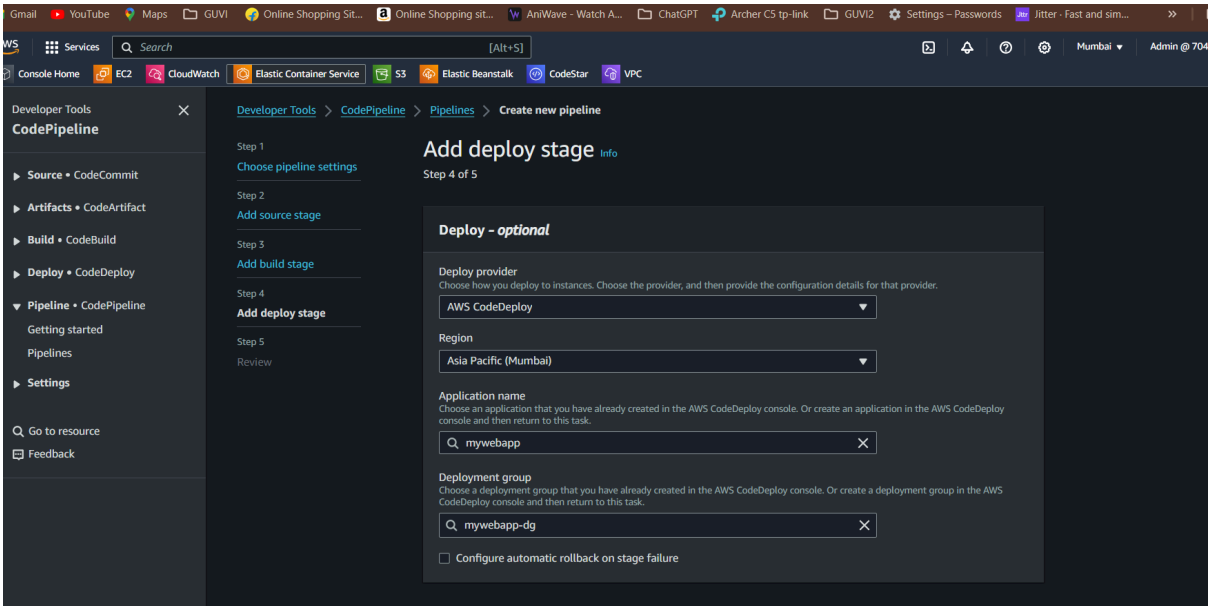
Step 3: Filled as per below, default info that is already created —> Click next.



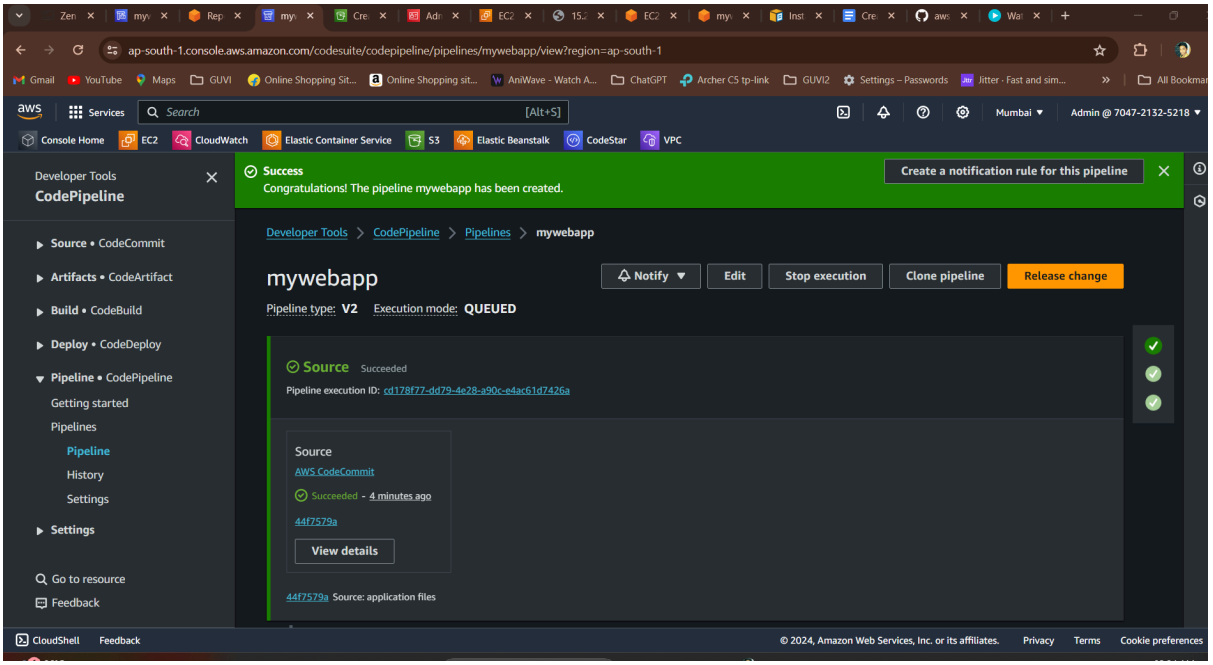
Step 4: Fill the info as per below —> Click next.



Step 5: Fill the info as per below —> Click next —> Click Create Piplile.



Pipline created successfully.



Pipeline Output: