

AWS CI/CD Pipeline

Overview

In this project I will be taking code from CodeCommit, building the project on CodeBuild, deploying the application on CodeDeploy and making a CI/CD pipeline using CodePipeline.

Goals

1. To create a pipeline manually that will take a code from CodeCommit then it will Build the code using CodeBuild then create the artifact and that artifact will get stored in AWS S3 and we will deploy the application on AWS EC2 .
2. Code -> CodeCommit -> CodeBuild -> AWS S3 -> CodeDeploy -> EC2
3. Then at last creating the CI/CD pipeline using AWS CodePipeline.

Skills

- AWS EC2
- AWS IAM User
- AWS IAM Role
- AWS S3
- AWS CodeCommit
- AWS CodeBuild
- AWS CodeDeploy
- AWS CodePipeline
- AWS KMS
- Artifact
- YAML
- VS Code



Milestones

I. AWS CodeCommit

AWS CodeCommit is a version control service hosted by Amazon Web Services that you can use to privately store and manage assets (such as documents, source code, and binary files) in the cloud.

II. AWS CodeBuild

AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers. CodeBuild scales continuously and processes multiple builds concurrently, so your builds are not left waiting in a queue. You can get started quickly by using prepackaged build environments, or you can create custom build environments that use your own build tools. With CodeBuild, you are charged by the minute for the compute resources you use.

III. AWS CodeDeploy

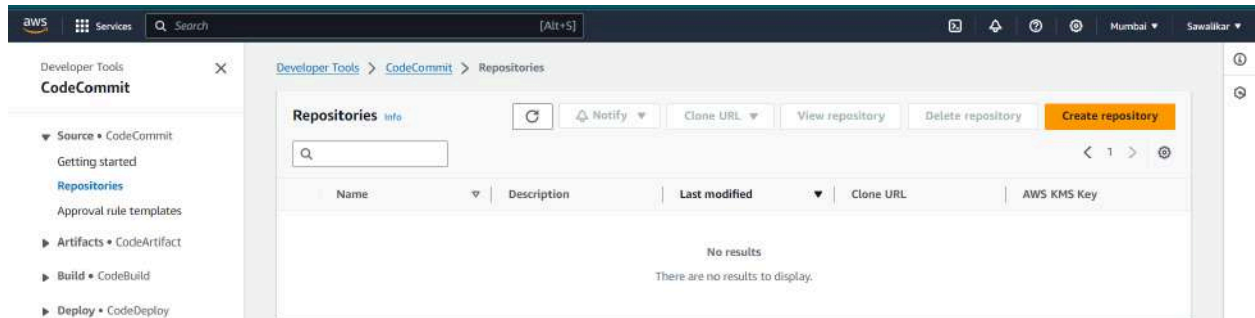
AWS CodeDeploy is a fully managed deployment service that automates software deployments to a variety of compute services such as Amazon EC2, AWS Fargate, AWS Lambda, and your on-premises servers. AWS CodeDeploy helps you to rapidly release new features, helps you avoid downtime during application deployment, and handles the complexity of updating your applications. You can use AWS CodeDeploy to automate software deployments, eliminating the need for error-prone manual operations. The service scales to match your deployment needs.

IV. AWS CodePipeline

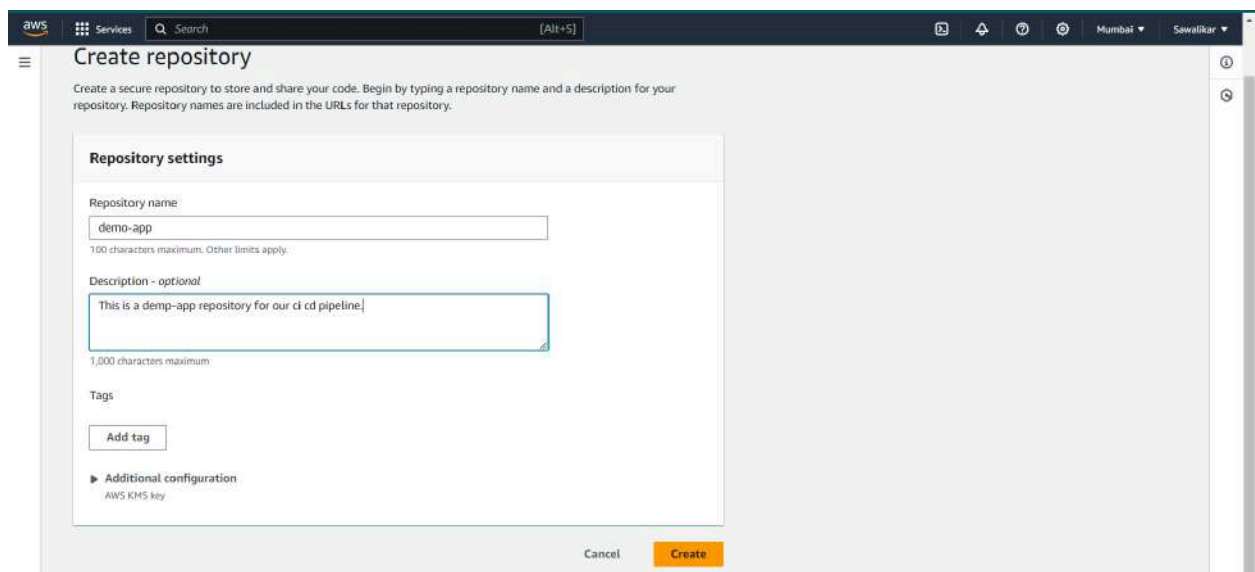
You can automate your release process by using AWS CodePipeline to test your code and run your builds with AWS CodeBuild.

Part 1 - AWS CodeCommit

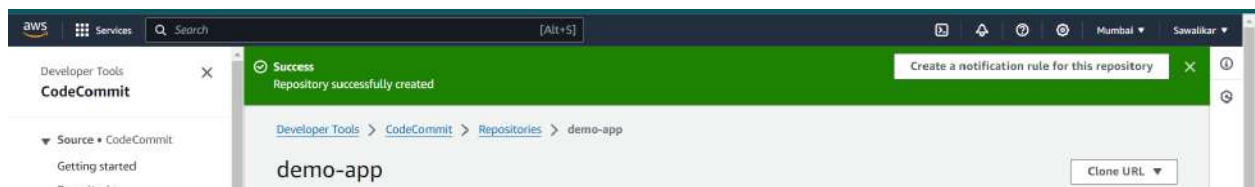
Step 1:- Search CodeCommit -> Click Create repository



Step 2:- Enter the name -> Write Description -> Press Create

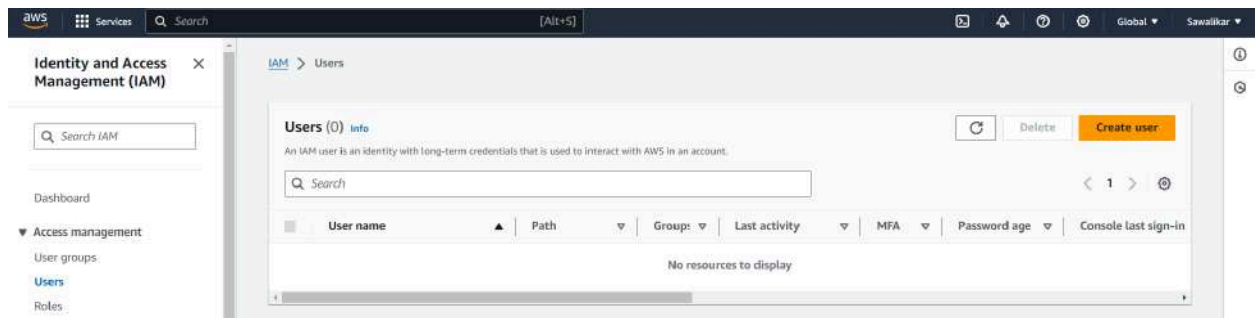


Step 3:- Repository created successfully , now I will create an IAM user in order to configure the SSH and HTTPS connections.

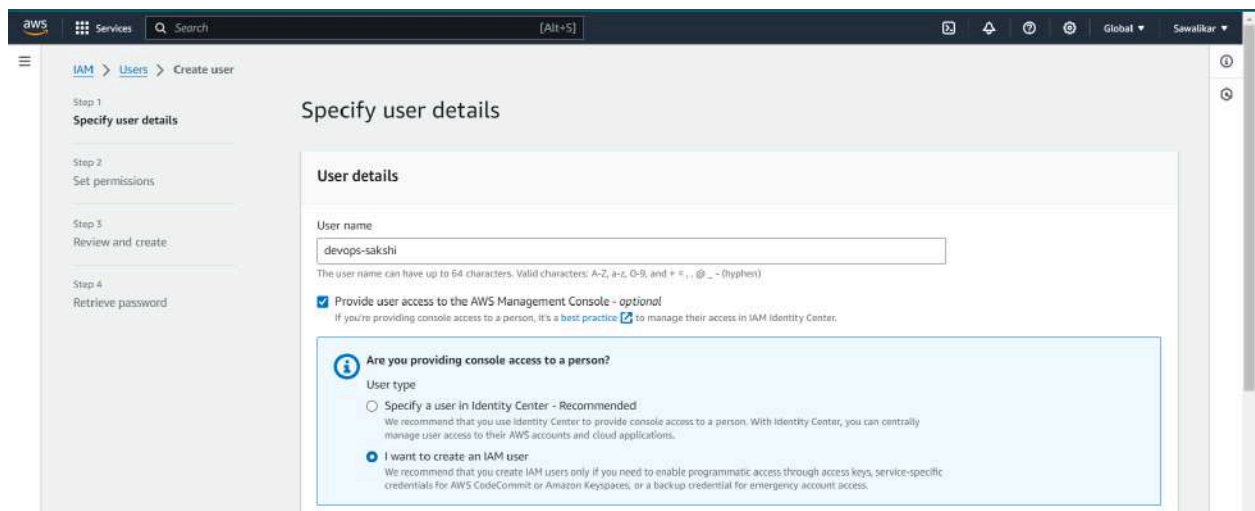


Step 4:- Creating IAM user

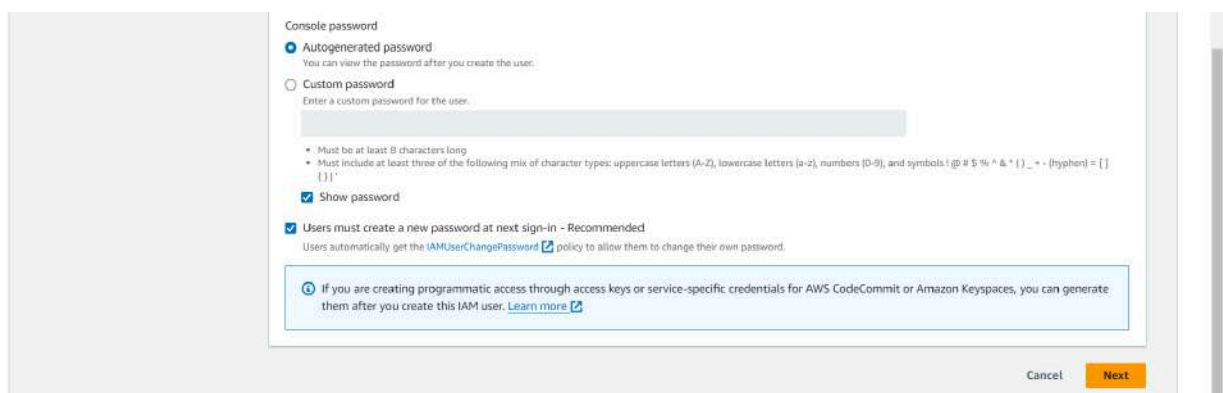
Search IAM -> Press Create User



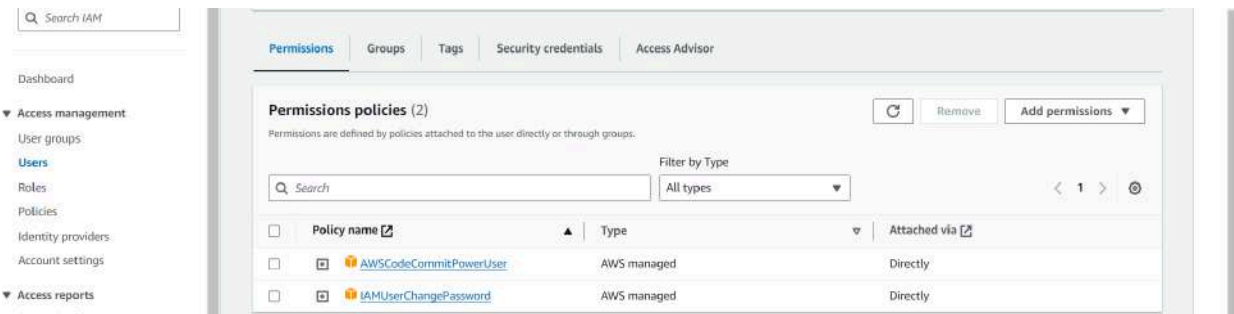
Give name to IAM user -> Select I want to create an IAM user



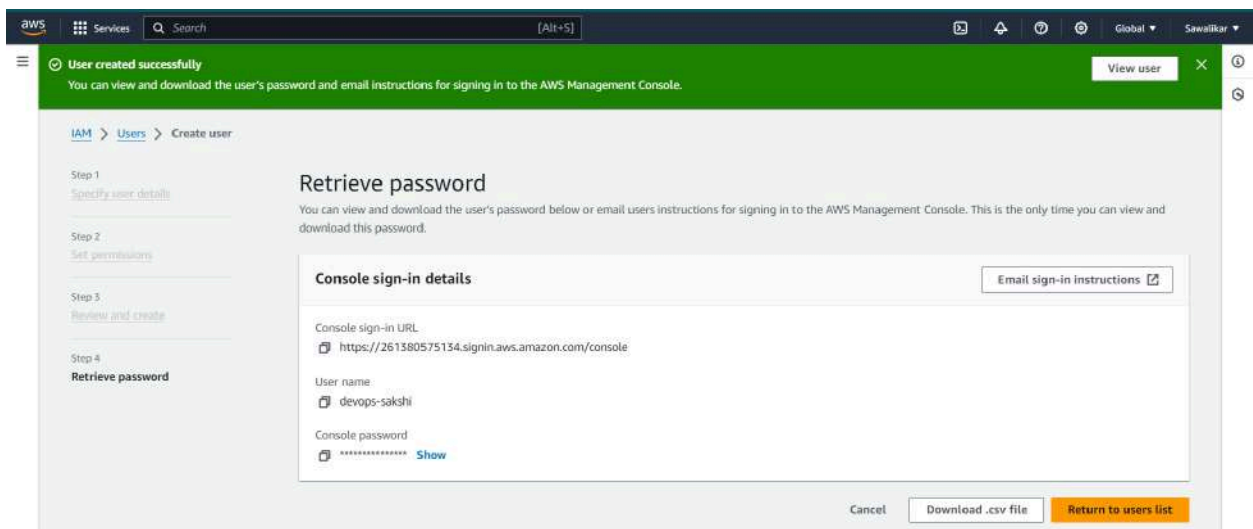
Select password -> Enable the show password -> Enable the Users must create a new password at next sign-in - Recommended



Add permissions -> Add AwsCodeCommitPowerUser permission



Next -> Create user -> Copy the credentials and download the csv



Console sign-in URL

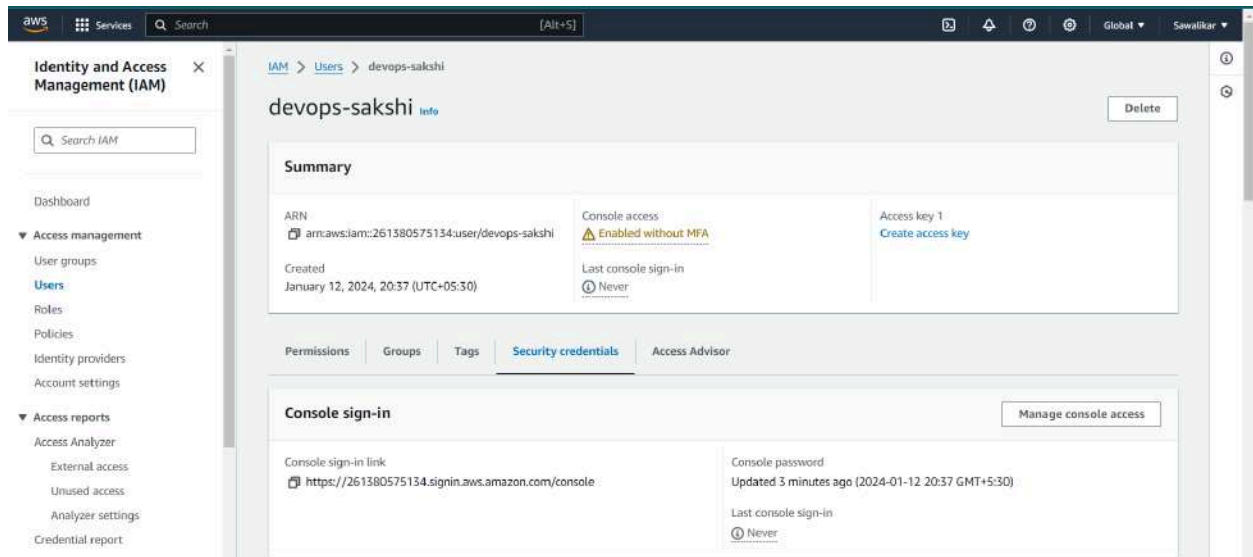
<https://261380575134.signin.aws.amazon.com/console>

User name

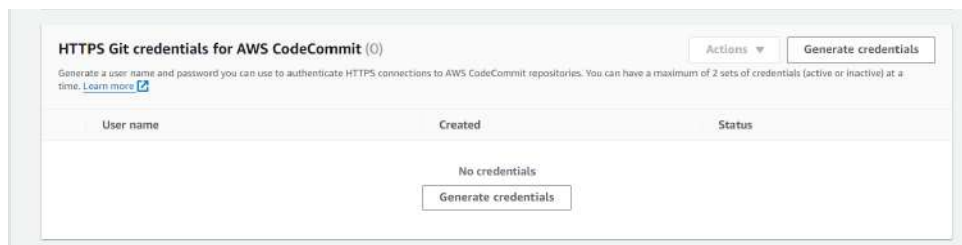
devops-sakshi

Console password

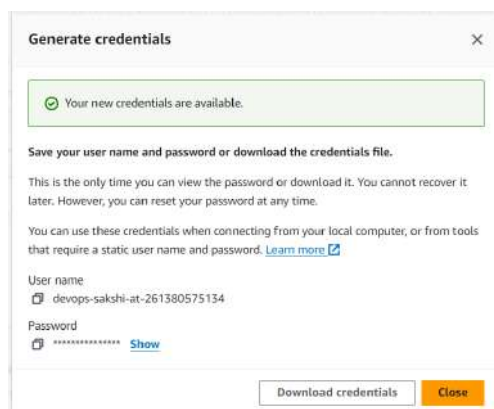
Step 5 :- Now select devops-sakshi -> Select security credentials



Scroll down and select HTTPS Git credentials for AWS CodeCommit -> Click generate credentials



Copy the credentials



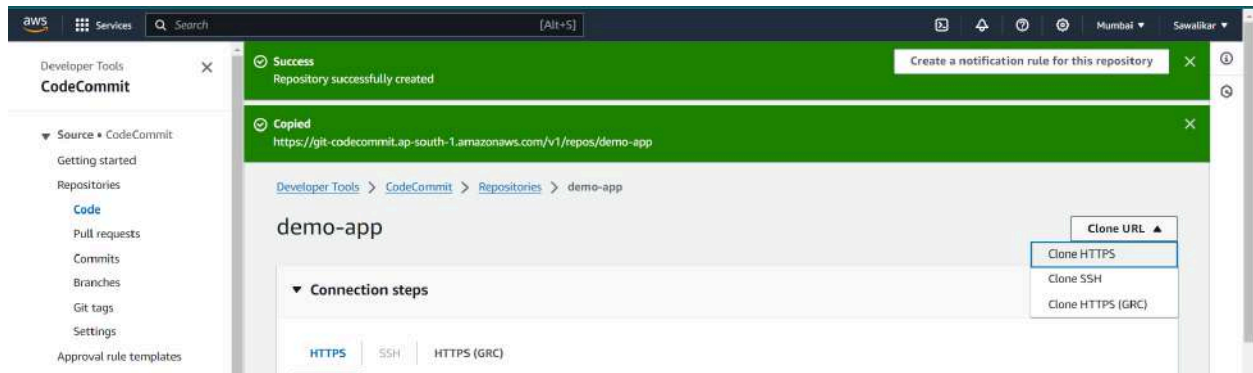
User name

devops-sakshi-at-261380575134

Password

Step 6:- Now again go to code commit our demo-app

Select Clone URL -> Select Clone HTTPS



Copy this url :- <https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/demo-app>

Step 7:- Now open a server

```
$ git --version
```

```
$ mkdir aws-devops
```

```
$ cd aws-devops
```

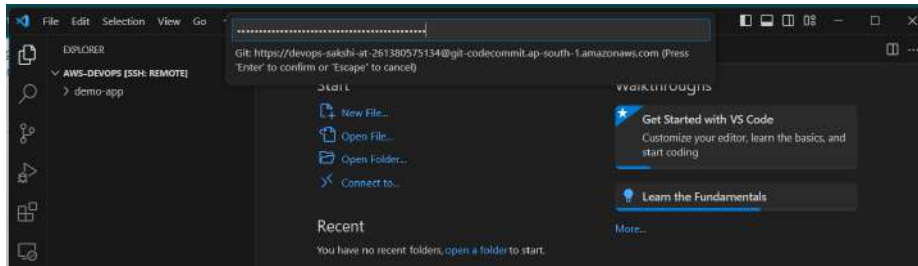
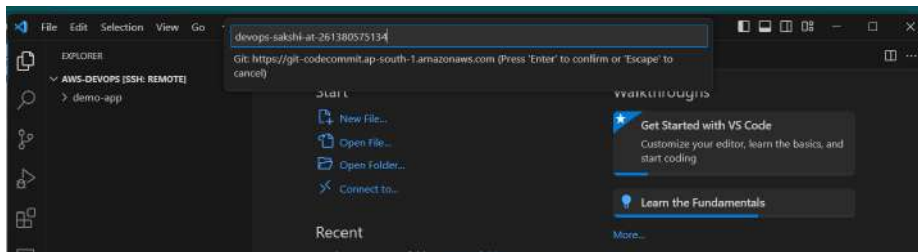
```
$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/demo-app
```

Fatal error



As we have not given the username and password hence fatal errors occur.

Now giving username and password



Successfully cloned the repository



Step 8:- Now I will create a new file in demo-app folder

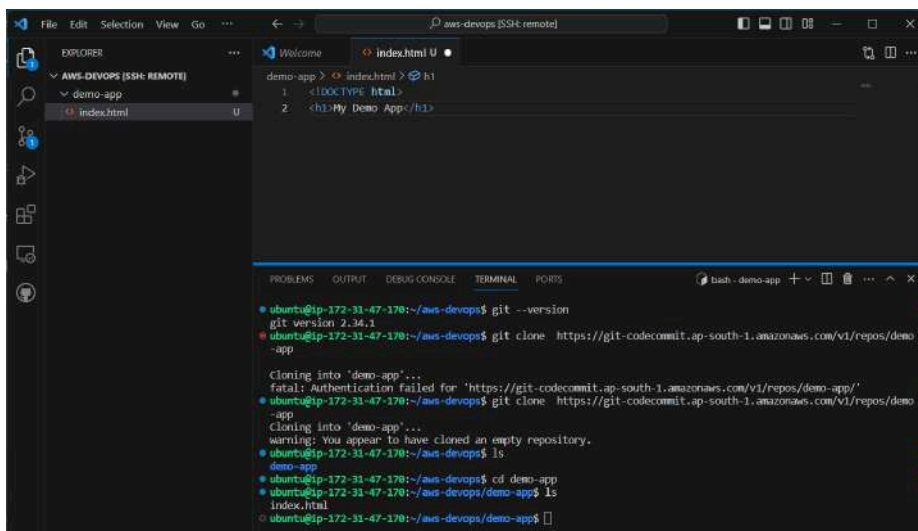
```
$ ls
```

```
$ cd demo-app
```

```
$ vi index.html
```

Or directly create a new file using vs code interface

Write some html code in index.html file




```
$ git status
```

(Making sure that I must be in demo-app)

```
● ubuntu@ip-172-31-47-170:~/aws-devops/demo-app$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

nothing added to commit but untracked files present (use "git add" to track)
```

```
$ git add .
```

```
$ git commit -m "Adding sample file index.html"
```

```
● ubuntu@ip-172-31-47-170:~/aws-devops/demo-app$ git add .
● ubuntu@ip-172-31-47-170:~/aws-devops/demo-app$ git commit -m "Adding sample file index.html"
[master (root-commit) 584dcaa] Adding sample file index.html
  Committer: Ubuntu <ubuntu@ip-172-31-47-170.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

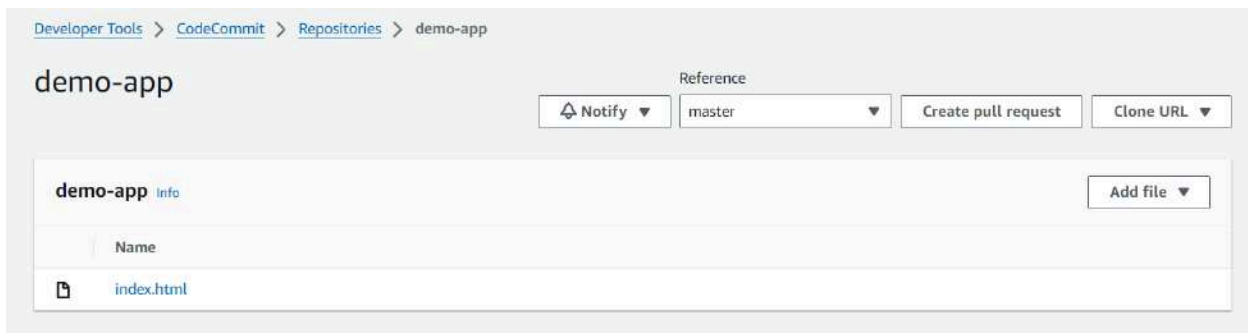
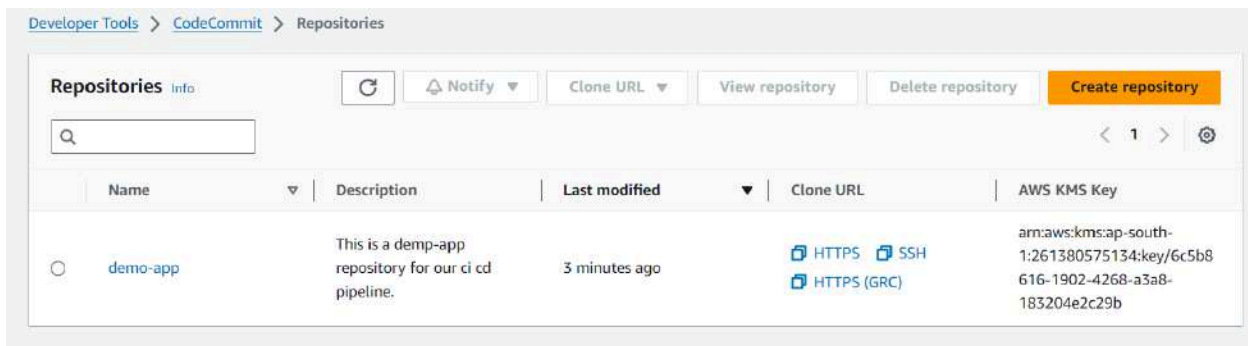
    git commit --amend --reset-author

1 file changed, 2 insertions(+)
create mode 100644 index.html
```

```
$ git push origin master
```

```
● ubuntu@ip-172-31-47-170:~/aws-devops/demo-app$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 280 bytes | 280.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/demo-app
* [new branch]      master -> master
```

Open the demo-app repository and you will see the index.html file



Step 9:- Now let's create a new branch 'dev'

```
$ git checkout -b dev
```

```
$ git commit -m "First time commit from dev branch"
```

```
ubuntu@ip-172-31-47-170:~/aws-devops/demo-app$ git checkout -b dev
Switched to a new branch 'dev'
ubuntu@ip-172-31-47-170:~/aws-devops/demo-app$ git add .
ubuntu@ip-172-31-47-170:~/aws-devops/demo-app$ git commit -m "First time commit from dev branch"
[dev 79cec42] First time commit from dev branch
Committer: Ubuntu <ubuntu@ip-172-31-47-170.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 2 insertions(+), 1 deletion(-)
```

```
$ git push origin dev
```

```
● ubuntu@ip-172-31-47-170:~/aws-devops/demo-app$ git push origin dev
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 331 bytes | 331.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/demo-app
* [new branch]      dev -> dev
```

Now go to demo-app repository, dev branch has been created successfully

Developer Tools > CodeCommit > Repositories > demo-app > Branches

demo-app

Branches [info](#) [Delete branch](#) [View branch](#) [View last commit](#) [Create pull request](#) [Create branch](#)

< 1 > ⚙

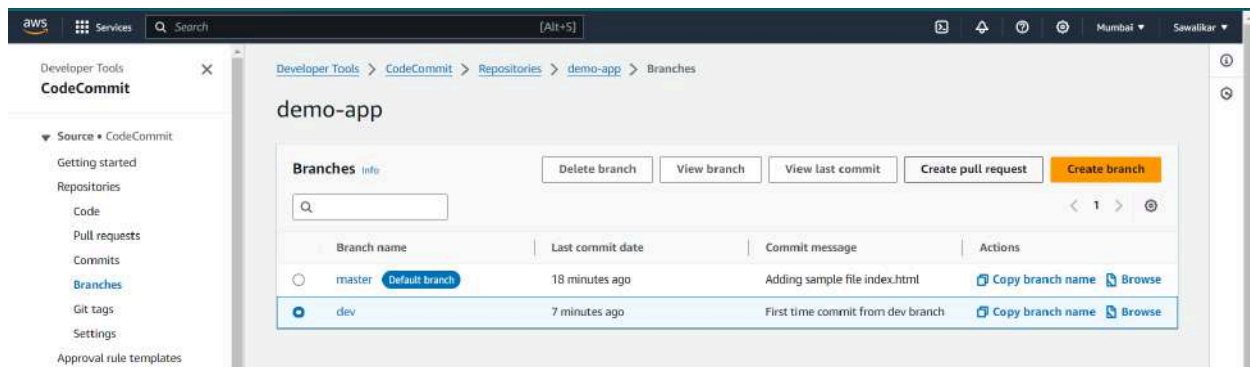
Branch name	Last commit date	Commit message	Actions
<input type="radio"/> master Default branch	16 minutes ago	Adding sample file index.html	Copy branch name Browse
<input type="radio"/> dev	5 minutes ago	First time commit from dev branch	Copy branch name Browse

Step 10:- Creating a pull request

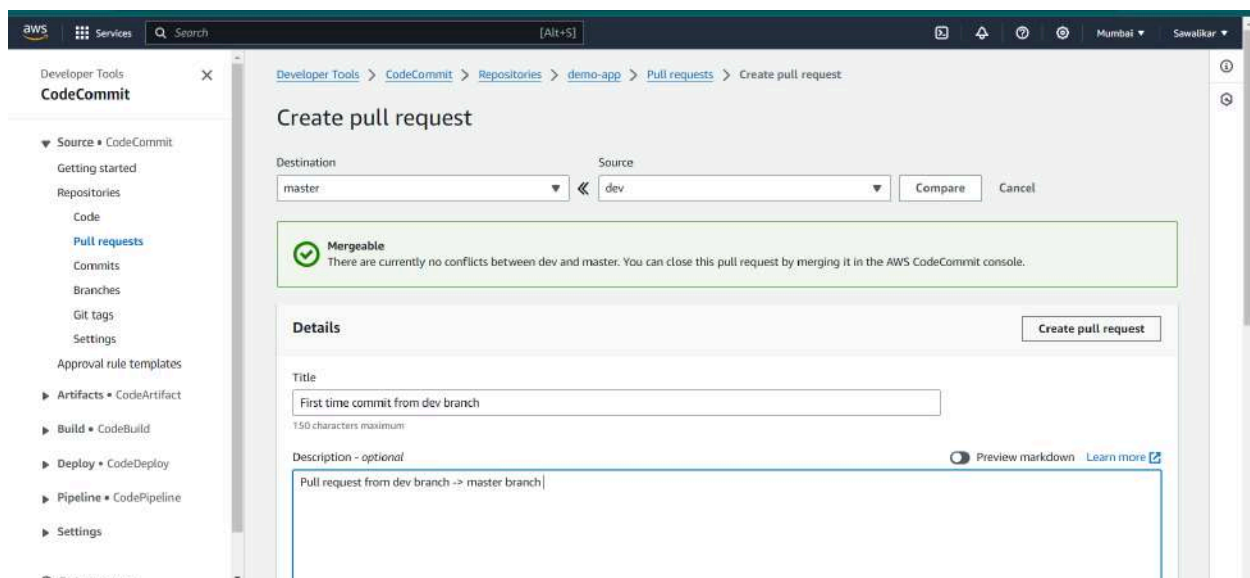
Source -> dev

Destination -> master

Select dev -> Press Create branch



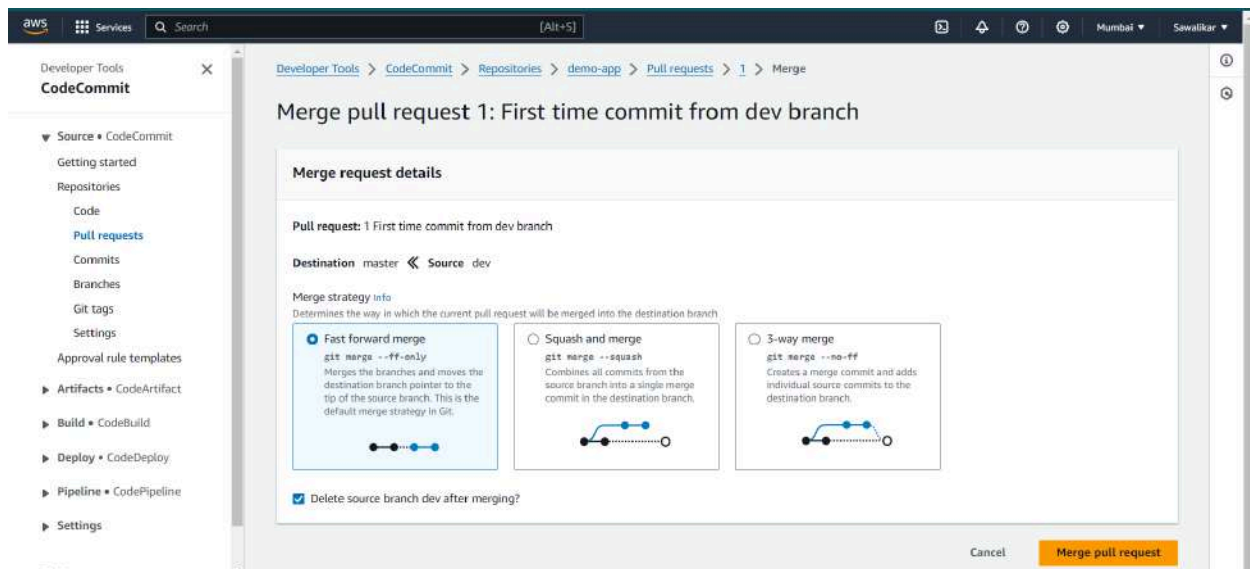
Add description -> Click create pull request



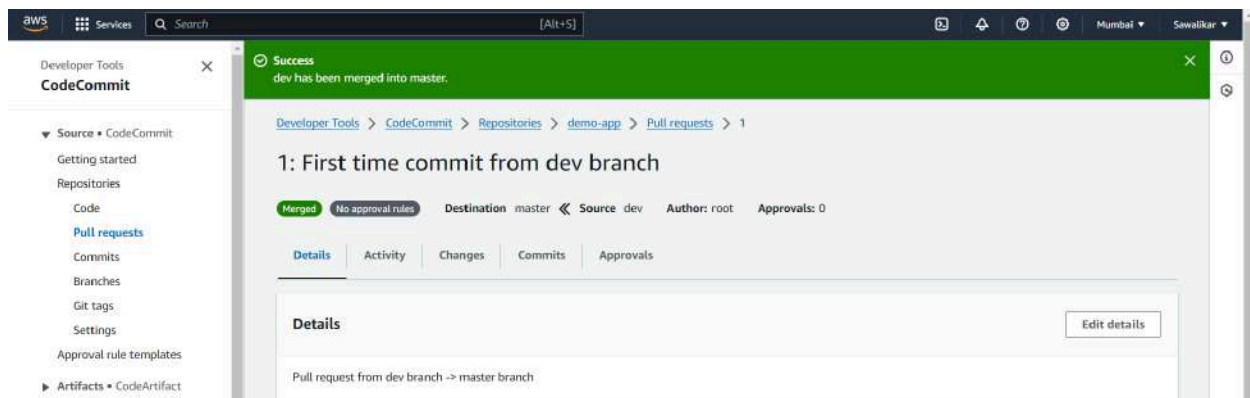
Click Merge



Now select Fast forward merge -> Enable the Delete source branch dev after merging box -> Click Merge pull request

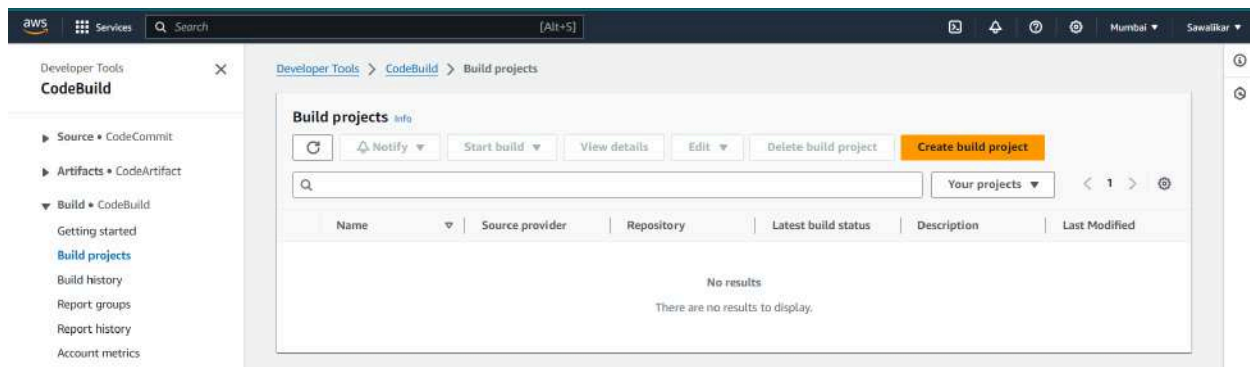


Successful

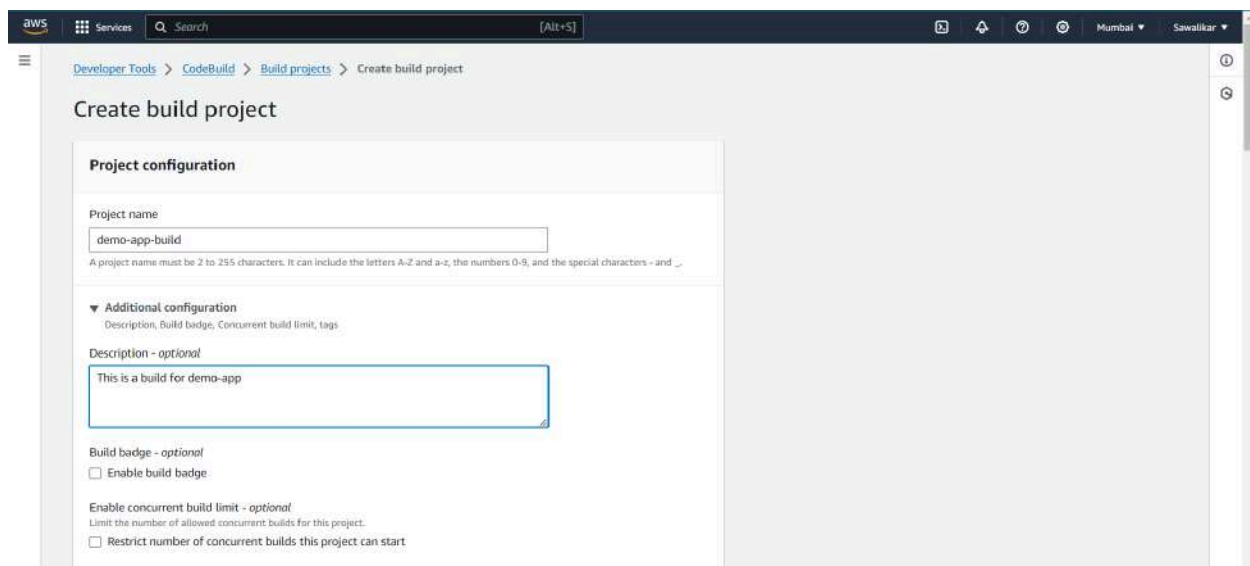


Part 2 - CodeBuild

Step 1:- Search and open CodeBuild ->Click on Create build project



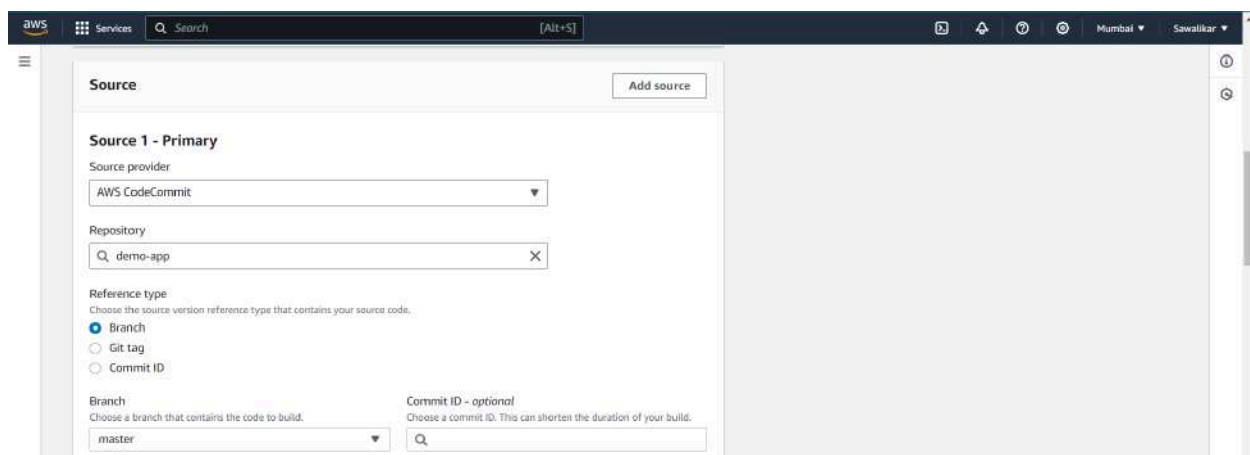
Give project name -> Give description



Source provider -> AWS CodeCommit

Repository -> demo app

Branch -> master



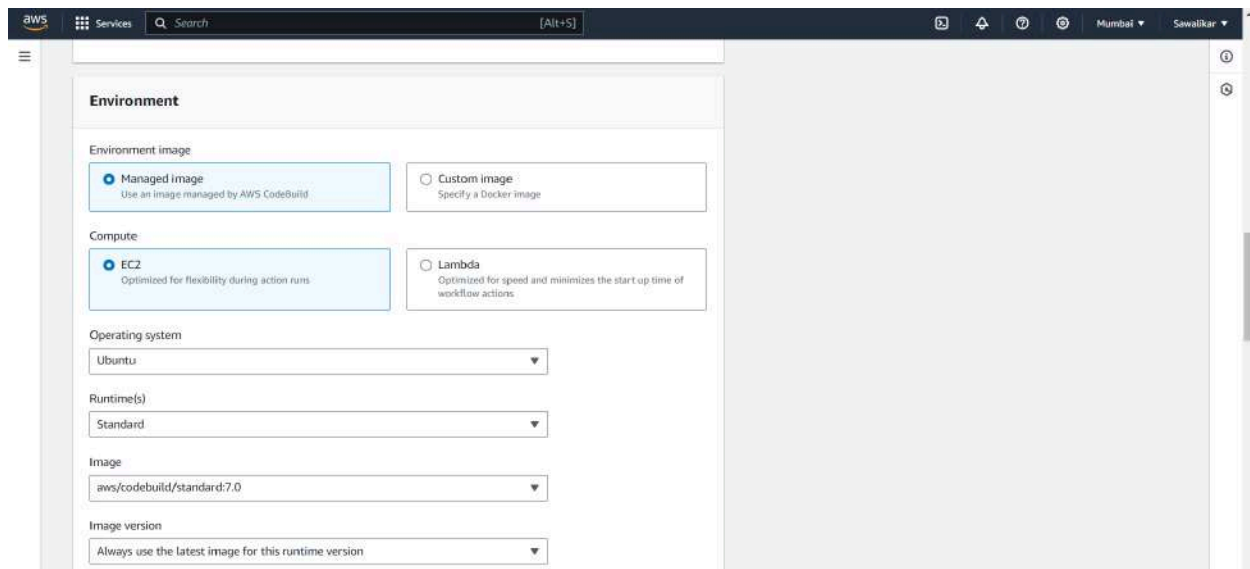
Environment -> Managed image

Compute -> EC2

Operating System -> Ubuntu

Runtime -> Standard

Image -> Select latest image version



Select New service role -> Role will get created automatically

Step 2:- Now create a Buildspec file

```

1  version: 0.2
2
3  phases:
4    install:
5      commands:
6        - echo Installing NGINX
7        - sudo apt-get update
8        - sudo apt-get install nginx-y
9    build:
10     commands:
11       - echo Build started on `date`
12       - cp index.html /var/www/html/
13    post_build:
14      commands:
15        - echo Configuring NGINX
16
17    artifacts:
18      files:
19        - '**/*'
20
21

```

A buildspec is a collection of build commands and related settings, in YAML format, that CodeBuild uses to run a build. Without a build spec, CodeBuild cannot successfully convert your build input into build output or locate the build output artifact in the build environment to upload to your output bucket.

*If you use another name instead of the Buildspec file then you have to mention it.

Step 3:-Now open the server

```
$ pwd
```

```
$ ls
```

```
$ cd demo-app
```

```
$ ls
```

```
$ git add .
```



```
$ git commit -m "Adding Buildspec.yml"
```

```
ubuntu@ip-172-31-47-170:~/aws-devops$ pwd
/home/ubuntu/aws-devops
ubuntu@ip-172-31-47-170:~/aws-devops$ ls
demo-app
ubuntu@ip-172-31-47-170:~/aws-devops$ cd demo-app
ubuntu@ip-172-31-47-170:~/aws-devops/demo-app$ ls
buildspec.yml index.html
ubuntu@ip-172-31-47-170:~/aws-devops/demo-app$ git add .
ubuntu@ip-172-31-47-170:~/aws-devops/demo-app$ git commit -m "Adding Buildspec.yml"
[dev 860ef6c] Adding Buildspec.yml
Committer: Ubuntu <ubuntu@ip-172-31-47-170.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 20 insertions(+)
create mode 100644 buildspec.yml
```

```
$ git push origin dev
```

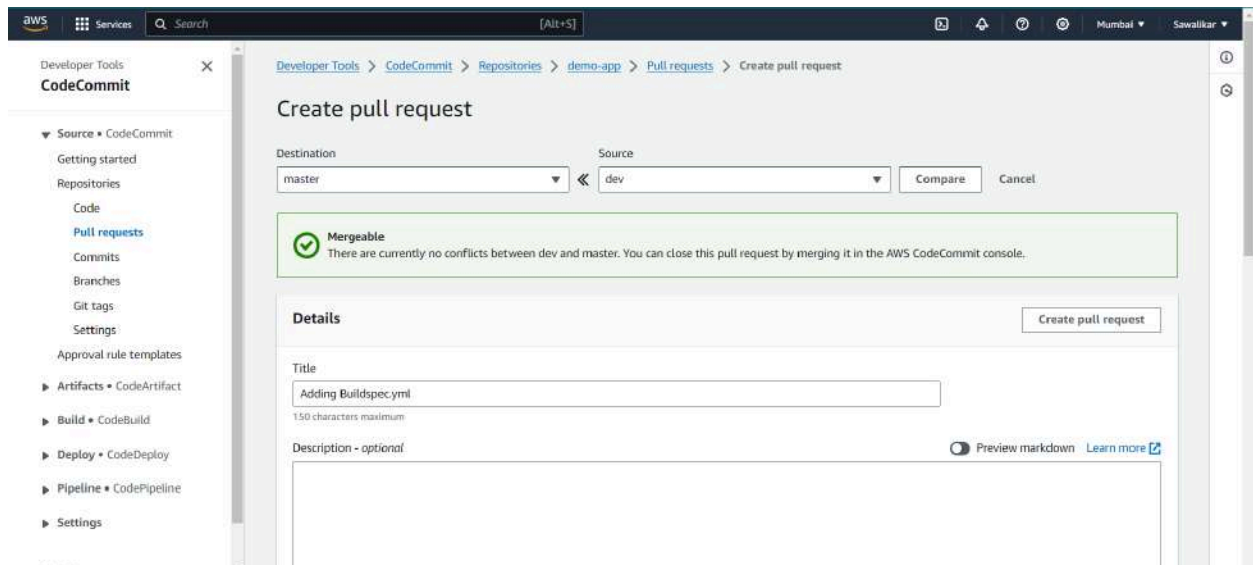
```
ubuntu@ip-172-31-47-170:~/aws-devops/demo-app$ git push origin dev
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 485 bytes | 485.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/demo-app
* [new branch]      dev -> dev
```



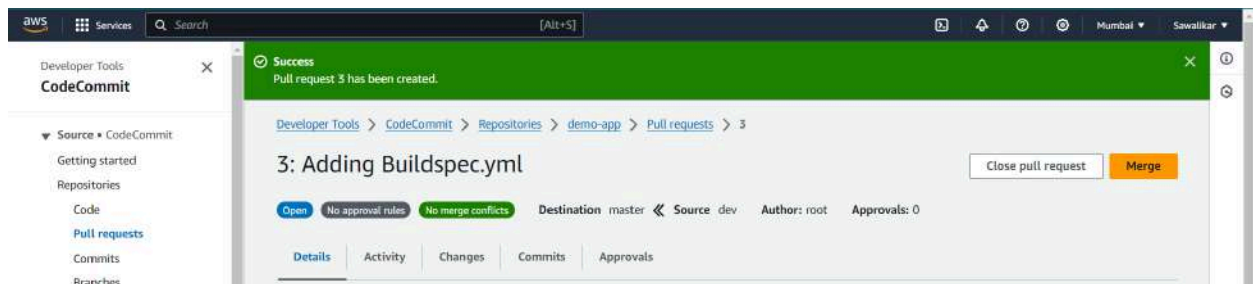
Step 4:-Now creating a pull request for that go to CodeCommit

Source -> dev

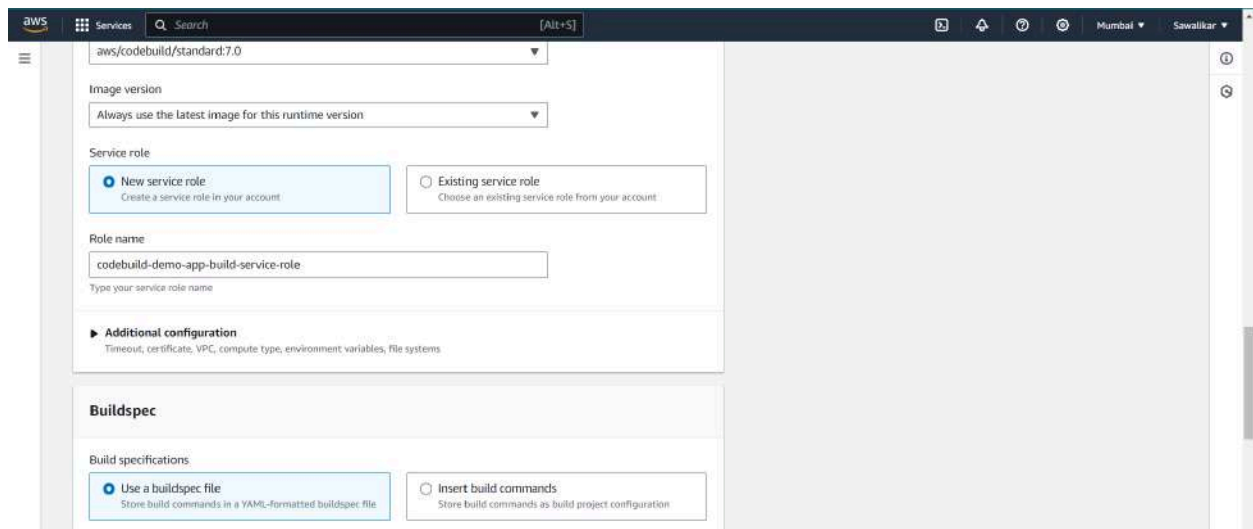
Destination -> master



Click Merge



Step 5:-Now resume the CodeBuild



Now scroll down and press Create build project

The screenshot shows the 'Artifacts' configuration page in the AWS CodeBuild console. The page has a dark blue header with the AWS logo, 'Services', a search bar, and a user profile 'Sawalikar'. The main content area is titled 'Artifacts' and includes an 'Add artifact' button. Below this, there's a section for 'Artifact 1 - Primary' with a 'Type' dropdown set to 'No artifacts'. A note states: 'You might choose no artifacts if you are running tests or pushing a Docker image to Amazon ECR.' There's an 'Additional configuration' section with 'Cache, encryption key'. Below that is a 'Logs' section with 'CloudWatch' and 'S3' options, each with a checkbox and a note: 'Checking this option will upload build output logs to CloudWatch.' At the bottom are 'Cancel' and 'Create build project' buttons.

Click Start build

The screenshot shows the 'demo-app-build' project page in the AWS CodeBuild console. A green banner at the top says 'Project created' and 'You have successfully created the following project: demo-app-build'. Below the banner, the breadcrumb trail is 'Developer Tools > CodeBuild > Build projects > demo-app-build'. The project name 'demo-app-build' is displayed with buttons for 'Notify', 'Share', 'Edit', 'Delete build project', 'Start build with overrides', and 'Start build'. The 'Configuration' section shows: 'Source provider: AWS CodeCommit', 'Primary repository: demo-app', 'Artifacts upload location: -', and 'Build badge: Disabled'. At the bottom are tabs for 'Build history', 'Batch history', 'Build details', 'Build triggers', and 'Metrics'.

The screenshot displays the AWS CodeBuild console interface. On the left, a sidebar menu under 'Developer Tools' includes 'CodeBuild', 'Source', 'Artifacts', 'Build', 'Deploy', 'Pipeline', and 'Settings'. The 'Build' section is expanded, showing options like 'Getting started', 'Build projects', 'Build project' (highlighted), 'Settings', 'Build history', 'Report groups', 'Report history', and 'Account metrics'. The main area shows a green notification banner: 'Build started' with the message 'You have successfully started the following build: demo-app-build:bd87a64c-d107'. Below this is a table of build steps:

Name	Status	Context	Duration
SUBMITTED	✓ Succeeded	-	<1 sec
QUEUED	✓ Succeeded	-	2 secs
PROVISIONING	✓ Succeeded	-	35 secs
DOWNLOAD_SOURCE	✓ Succeeded	-	6 secs
INSTALL	✓ Succeeded	-	25 secs
PRE_BUILD	✓ Succeeded	-	<1 sec
BUILD	✓ Succeeded	-	<1 sec
POST_BUILD	✓ Succeeded	-	<1 sec
UPLOAD_ARTIFACTS	✓ Succeeded	-	<1 sec
FINALIZING	✓ Succeeded	-	<1 sec
COMPLETED	✓ Succeeded	-	-

Step 6:- Now editing the build , adding artifact

Now open AWS S3 , I Am going to put everything in S3 .

Search AWS S3 -> Click on create bucket

The screenshot shows the AWS S3 console 'Create bucket' page. The breadcrumb navigation is 'Amazon S3 > Buckets > Create bucket'. The page title is 'Create bucket' with an 'Info' link. Below the title, it states 'Buckets are containers for data stored in S3. [Learn more](#)'. The 'General configuration' section includes:

- AWS Region:** A dropdown menu showing 'Asia Pacific (Mumbai) ap-south-1'.
- Bucket name:** A text input field containing 'devops-bucket-zero-to-hero'. An 'Info' link is next to the label. Below the field, a note states: 'Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)'.
- Copy settings from existing bucket - optional:** A section with the text 'Only the bucket settings in the following configuration are copied.' and a 'Choose bucket' button.
- Format:** A note indicating the format is 's3://bucket/prefix'.

aws

Services

Search

[Alt+S]

Global

Sawalikar

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☐ Block public access to buckets and objects granted through new access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☐ Block public access to buckets and objects granted through any access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

☐ Block public access to buckets and objects granted through new public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐ Block public and cross-account access to buckets and objects through any public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

⚠

Turning off block all public access might result in this bucket and the objects within becoming public

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

☐ Disable
☒ Enable

aws

Services

Search

[Alt+S]

Global

Sawalikar

Default encryption [Info](#)

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

☒ Server-side encryption with Amazon S3 managed keys (SSE-S3)
☐ Server-side encryption with AWS Key Management Service keys (SSE-KMS)
☐ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the Storage tab of the [Amazon S3 pricing page](#).

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

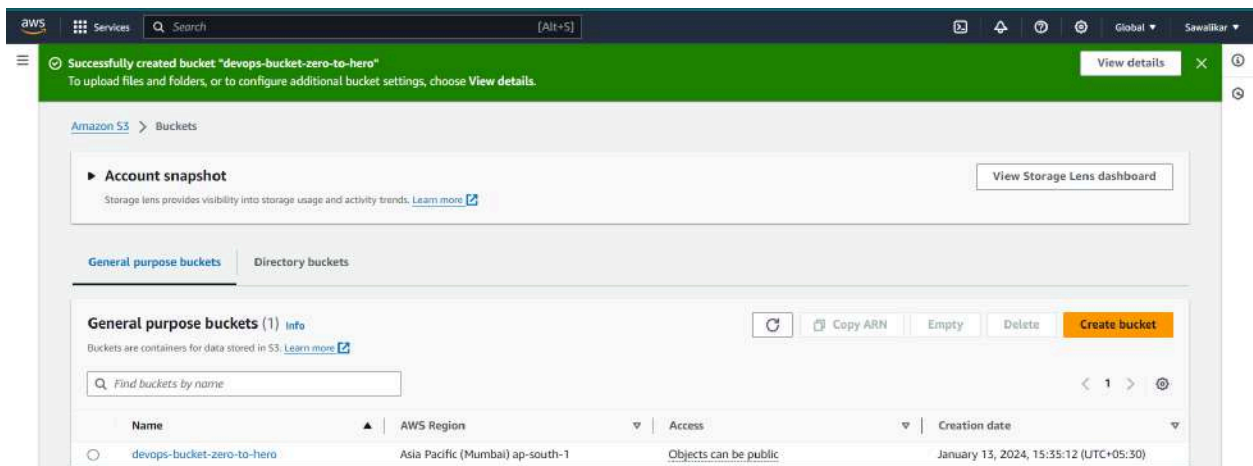
☐ Disable
☒ Enable

Advanced settings

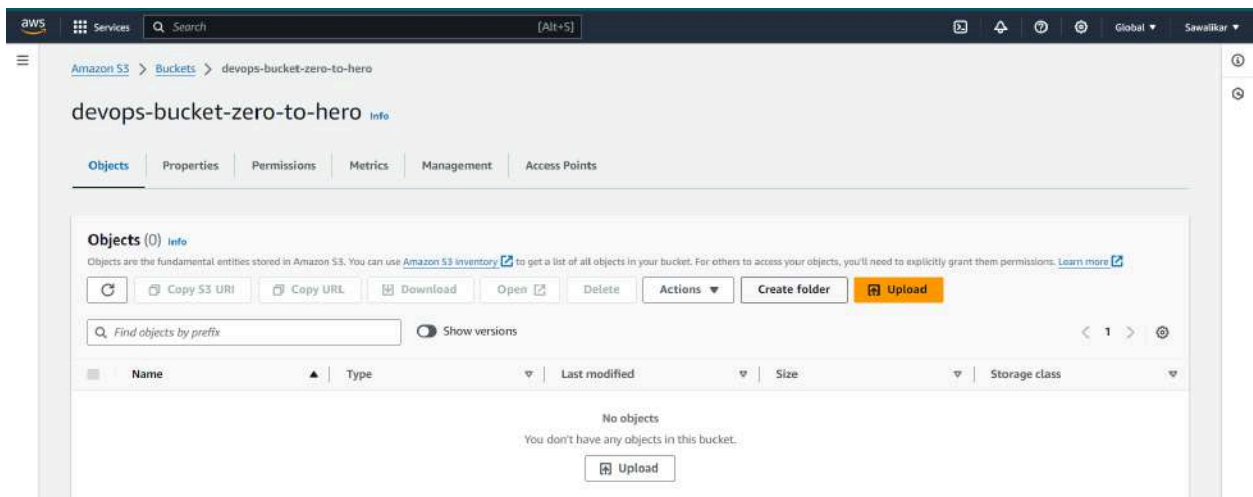
ⓘ After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel

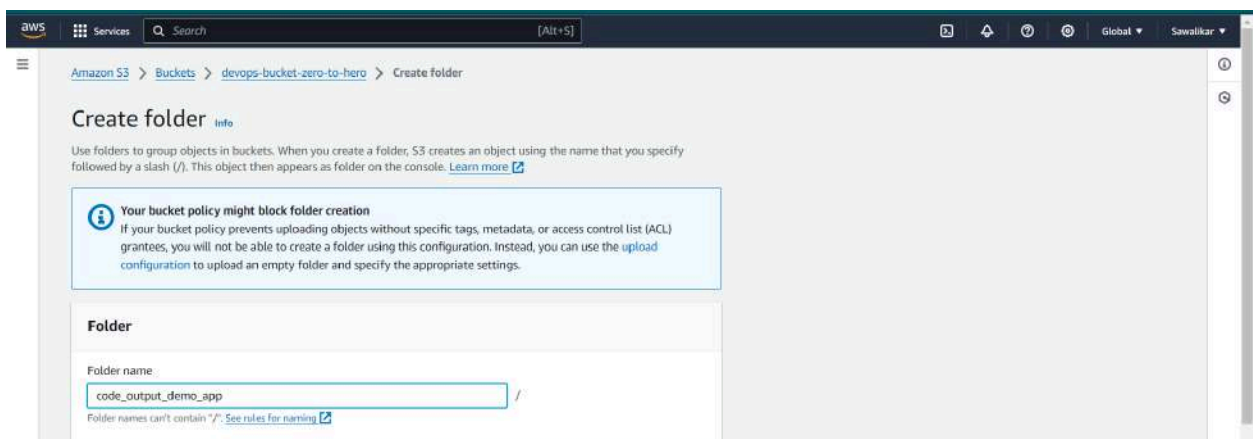
Create bucket

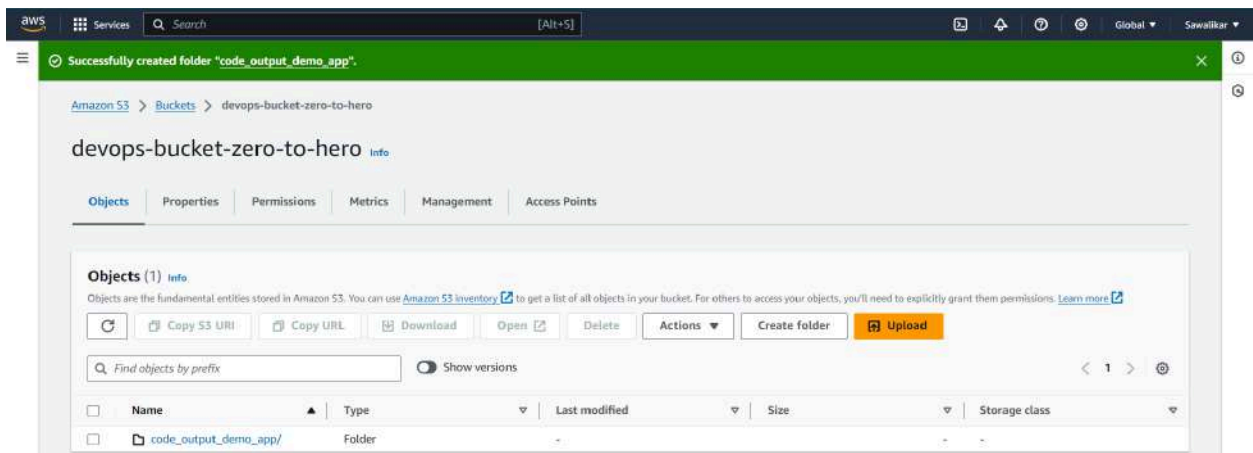


Step 7:- Going inside the bucket -> Click Create folder

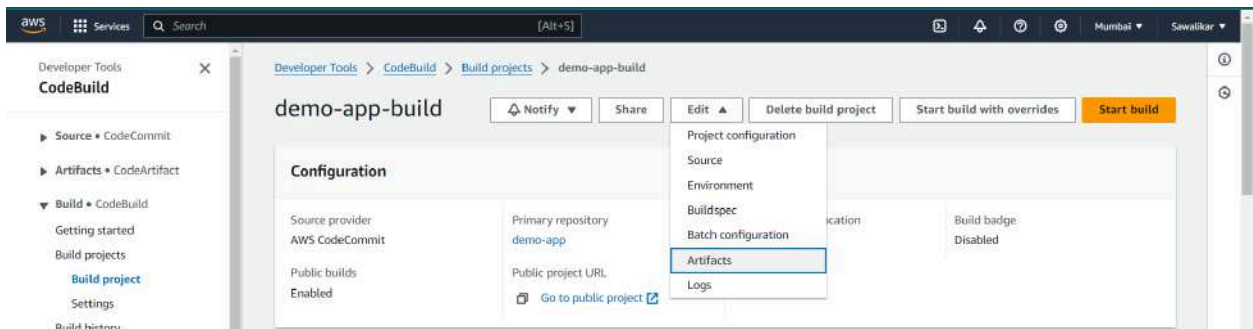


Give name -> code_output_demo_app





Go to demo-app-build -> Press Edit -> Click Artifacts



Type -> Amazon S3

Bucket name -> devops_bucket_zero_to_hero

Name -> code_output_demo_app

Path -> Go to S3 -> Select devops_bucket_zero_to_hero -> Select code_output_demo_app folder -> Copy path

Click Update artifact

Edit Artifacts

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: devops-bucket-zero-to-hero

Name: code_output_demo_app

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.

☐ Enable semantic versioning
Use the artifact name specified in the buildspec file.

Path - optional
The path to the build output ZIP file or folder.
s3://devops-bucket-zero-to-hero/code_output_demo_app/artifact.zip
Example: MyPath/MyArtifact.zip.

Artifacts updated successfully

Part 3 - CodeDeploy

Step 1:- Search codeDeploy -> Applications -> Press Create applications

Developer Tools

CodeDeploy

- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
- ▼ **Deploy • CodeDeploy**
 - Getting started
 - Deployments
 - Applications**

Applications Notify View details Deploy application Create application

Search:

Application name	Compute platform	Created
No results There are no results to display.		

Step 2:- Give application name -> Select compute platform as EC2

Developer Tools > CodeDeploy > Applications > Create application

Create application

Application configuration

Application name
Enter an application name
demo-app-application
100 character limit

Compute platform
Choose a compute platform
EC2/On-premises

Tags
Add tag

Cancel Create application

Developer Tools > CodeDeploy > Applications > demo-app-application

demo-app-application

Notify Delete application

Application details

Name	demo-app-application	Compute platform	EC2/On-premises
------	----------------------	------------------	-----------------

Developer Tools
CodeDeploy

- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
- ▼ Deploy • CodeDeploy
 - Getting started
 - Deployments
 - Applications

Step 3:- Create deployment group

Click deployment group

Developer Tools > CodeDeploy > Applications > demo-app-application

demo-app-application

Notify Delete application

Application details

Name	demo-app-application	Compute platform	EC2/On-premises
------	----------------------	------------------	-----------------

Deployments Deployment groups Revisions

Deployment groups

View details Edit Create deployment group

Q

Name	Status	Last attempted deploy...	Last successful deploy...	Trigger count
No deployment groups				

Before you can deploy your application using CodeDeploy, you must create a deployment group.

Create deployment group

Developer Tools
CodeDeploy


- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
- ▼ Deploy • CodeDeploy
 - Getting started
 - Deployments
 - Applications
 - Application
 - Settings
 - Deployment configurations
 - On-premises instances
- Pipeline • CodePipeline
- Settings

Go to resource

Before creating the deployment group we are going to create an IAM role. As an IAM role is required to create a deployment.

For that go to IAM and create a role named as “code-deploy-service-role”

Give the below accesses to IAM user

Permissions policy summary		
Policy name 	Type	Attached as
AmazonEC2FullAccess	AWS managed	Permissions policy
AmazonEC2RoleforAWSCodeDeploy	AWS managed	Permissions policy
AmazonEC2RoleforAWSCodeDeployLimited	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy
AWSCodeDeployFullAccess	AWS managed	Permissions policy
AWSCodeDeployRole	AWS managed	Permissions policy

IAM role created successfully . Now copy the ARN we are going to require it further to create deployment.

[IAM](#) > [Roles](#) > code-deploy-service-role

code-deploy-service-role Info

Allows EC2 instances to call AWS services on your behalf.

[Delete](#)


Summary Edit


Creation date

January 14, 2024, 13:35 (UTC+05:30)

Last activity

-


 ARN copied

 arn:aws:iam::261380575134:role/code-deploy-service-role








Maximum session duration

1 hour

Instance profile ARN

 arn:aws:iam::261380575134:instance-profile/code-deploy-service-role

Step 4:- I want to deploy our application on an EC2 instance , so now we are creating an EC2 instance.

Instances (1) <small>Info</small>							
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>							
running  Clear filters							
<input type="checkbox"/>	Name 	Instance ID	Instance state	Instance type	Status check	Alarm status	Public IPv4 DNS
<input type="checkbox"/>	demo-app-ec2	i-0c81fbf3653cc3164	 Running  	t2.micro	 Initializing View alarms 	ap-south-1b	ec2-15-206-205-

Step 5:- Now let's create a deployment group

Give name to the deployment group

Developer Tools > CodeDeploy > Applications > demo-app-application > Create deployment group

Create deployment group

Application

Application
demo-app-application
Compute type
EC2/On-premises

Deployment group name

Enter a deployment group name

demo-app-depl-gp

100 character limit

Select service role -> Copy the arn id of the IAM role "code-deploy-service-role"

Deployment type -> In place

Service role

Enter a service role

Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.

arn:aws:iam::261380575134:role/code-deploy-service-role

Deployment type

Choose how to deploy your application

☒ **In-place**
Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update.

☐ **Blue/green**
Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the original environment are deregistered and can be terminated.

Now write key-value

Key -> Name

Value -> demo-app-ec2 (The instance that we have create above)

Environment configuration

Select any combination of Amazon EC2 Auto Scaling groups, Amazon EC2 instances, and on-premises instances to add to this deployment

☐ Amazon EC2 Auto Scaling groups

☒ Amazon EC2 instances
1 unique matched instance. [Click here for details](#)

You can add up to three groups of tags for EC2 instances to this deployment group.
One tag group: Any instance identified by the tag group will be deployed to.
Multiple tag groups: Only instances identified by all the tag groups will be deployed to.

Tag group 1

Key: Value - optional:

☐ On-premises instances

Matching instances
1 unique matched instance. [Click here for details](#)

CodeDeploy Agent -> Never

Agent configuration with AWS Systems Manager [info](#)

⚠️ We recommend configuring your CodeDeploy Agent install and updates with AWS Systems Manager.
 AWS Systems Manager provides more control over CodeDeploy Agent version updates and rollbacks than installing using other methods. [Learn more](#)

Install AWS CodeDeploy Agent

☒ Never
☐ Only once
☐ Now and schedule updates

Deployment settings

Deployment configuration
 Choose from a list of default and custom deployment configurations. A deployment configuration is a set of rules that determines how fast an application is deployed and the success or failure conditions for a deployment.

or

Load balancer

Select a load balancer to manage incoming traffic during the deployment process. The load balancer blocks traffic from each instance while it's being deployed to and allows traffic to it again after the deployment succeeds.

☒ Enable load balancing

Load balancer type

☐ Application Load Balancer or Network Load Balancer
☐ Classic Load Balancer

► Advanced - optional

Step 6:- Now open the "demo-app-ec2" instance

I have chosen not to install the AWS CodeDeploy Agent during the deployment group configuration because of potential compatibility issues. While AWS has upgraded the EC2 service, the CodeDeployment service hasn't been upgraded as effectively, which might

result in version-related problems. To resolve this issue, I will run a script on the EC2 instance to manually set up the AWS CodeDeploy Agent and ensure a seamless integration.

Setting Up AWS CodeDeploy Agent on Ubuntu EC2

Setup an AWS CodeDeploy Agent in simple steps

In order to deploy your app to EC2, CodeDeploy needs an agent which actually deploys the code on your EC2.

So let's set it up.

Create a shell script with the below contents and run it.

```
$ vi install.sh
```

```
-----
```

```
#!/bin/bash
```

```
# This installs the CodeDeploy agent and its prerequisites on Ubuntu 22.04.
```

```
sudo apt-get update
```

```
sudo apt-get install ruby-full ruby-webrick wget -y
```

```
cd /tmp
```

```
wget
```

```
https://aws-codedeploy-ap-south-1.s3.ap-south-1.amazonaws.com/releases/codedeploy-agent\_1.3.2-1902\_all.deb
```

```
mkdir codedeploy-agent_1.3.2-1902_ubuntu22
```

```
dpkg-deb -R codedeploy-agent_1.3.2-1902_all.deb codedeploy-agent_1.3.2-1902_ubuntu22
```

```
sed 's/Depends:./Depends:ruby3.0/' -i
```

```
./codedeploy-agent_1.3.2-1902_ubuntu22/DEBIAN/control
```

```
dpkg-deb -b codedeploy-agent_1.3.2-1902_ubuntu22/
```

```
sudo dpkg -i codedeploy-agent_1.3.2-1902_ubuntu22.deb
```

```
systemctl list-units --type=service | grep codedeploy
```

```
sudo service codedeploy-agent status
```

```
-----
```

Copy the above code.

```

aws Services Search [Alt+S] Mumbai Sawalakar
# /bin/bash
# This installs the CodeDeploy agent and its prerequisites on Ubuntu 22.04.
sudo apt-get update
sudo apt-get install ruby-full ruby-webrick wget -y
cd /tmp
wget https://aws-codedeploy-ap-south-1.s3.ap-south-1.amazonaws.com/releases/codedeploy-agent_1.3.2-1902_all.deb
mkdir codedeploy-agent_1.3.2-1902_ubuntu22
dpkg-deb -x codedeploy-agent_1.3.2-1902_all.deb codedeploy-agent_1.3.2-1902_ubuntu22
sed 's/Depends:.*ruby3.0/' -i ./codedeploy-agent_1.3.2-1902_ubuntu22/DEBIAN/control
dpkg-deb -b codedeploy-agent_1.3.2-1902_ubuntu22/
sudo dpkg -i codedeploy-agent_1.3.2-1902_ubuntu22.deb
systemctl list-units --type=service | grep codedeploy
sudo service codedeploy-agent status

```

Status active

```

aws Services Search [Alt+S] Mumbai Sawalakar
2024-01-14 09:16:46 (159 MB/s) - 'codedeploy-agent_1.3.2-1902_all.deb' saved [2749704/2749704]

dpkg-deb: building package 'codedeploy-agent' in 'codedeploy-agent_1.3.2-1902_ubuntu22.deb'.
Selecting previously unselected package codedeploy-agent.
(Reading database ... 81218 files and directories currently installed.)
Preparing to unpack codedeploy-agent_1.3.2-1902_ubuntu22.deb ...
Unpacking codedeploy-agent (1.3.2-1902) ...
Setting up codedeploy-agent (1.3.2-1902) ...
codedeploy-agent.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable codedeploy-agent
codedeploy-agent.service loaded active running LSB: AWS CodeDeploy Host Agent
* codedeploy-agent.service - LSB: AWS CodeDeploy Host Agent
   Loaded: loaded (/etc/init.d/codedeploy-agent; generated)
   Active: active (running) since Sun 2024-01-14 09:16:57 UTC; 373ms ago
     Docs: man:systemd-sysv-generator(8)
  Process: 2482 ExecStart=/etc/init.d/codedeploy-agent start (code=exited, status=0/SUCCESS)
    Tasks: 2 (limit: 1121)
   Memory: 49.5M
      CPU: 729ms
   CGroup: /system.slice/codedeploy-agent.service
           └─2480 "codedeploy-agent: master 2480"
             └─2490 "codedeploy-agent: booting child"

Jan 14 09:16:56 ip-172-31-6-68 systemd[1]: Starting LSB: AWS CodeDeploy Host Agent...
Jan 14 09:16:57 ip-172-31-6-68 codedeploy-agent[2482]: Starting codedeploy-agent:
Jan 14 09:16:57 ip-172-31-6-68 systemd[1]: Started LSB: AWS CodeDeploy Host Agent.
ubuntu@ip-172-31-6-68:~$

```

Step 7:- Now under demo-app folder -> Create new folder scripts -> Under scripts create two files

install_nginx.sh and start_nginx.sh

Write below code in install_nginx.sh

```

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

```

Write below code in start_nginx.sh

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

Now create a new file under demo-app folder -> appspec.yml

If your application uses the EC2/On-Premises compute platform, the AppSpec file must be a YAML-formatted file named appspec.yml and it must be placed in the root of the directory structure of an application's source code. Otherwise, deployments fail. It is used by CodeDeploy to determine:

- What it should install onto your instances from your application revision in Amazon S3 or GitHub.
- Which lifecycle event hooks to run in response to deployment lifecycle events.

After you have a completed AppSpec file, you bundle it, along with the content to deploy, into an archive file (zip, tar, or compressed tar).

```

demo-app > ! appspec.yml > {} hooks > [ ] AfterInstall > {} 0 > location
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

```

\$ git pull origin master

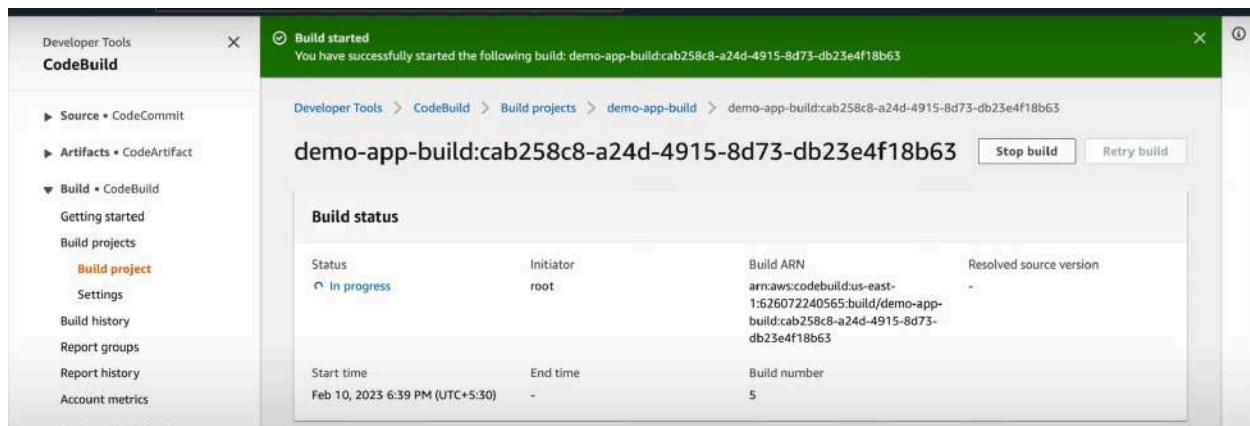
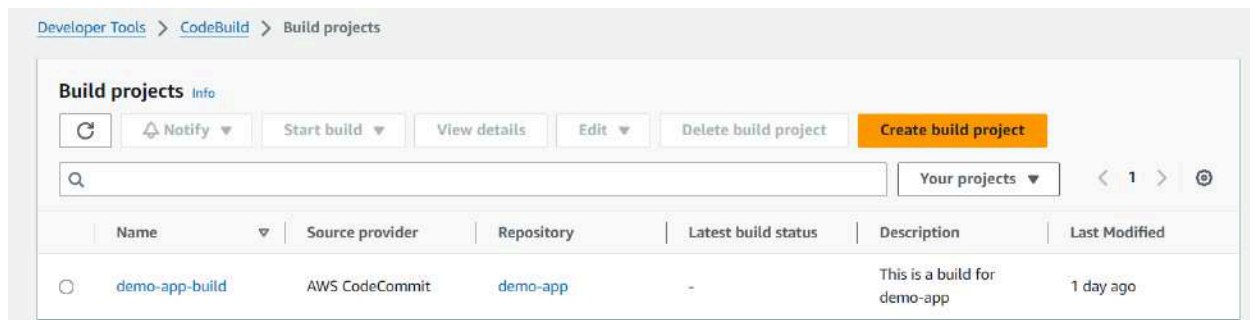
\$ git add .

\$ git commit -m "Adding appspec.yml file"

\$ git push origin master

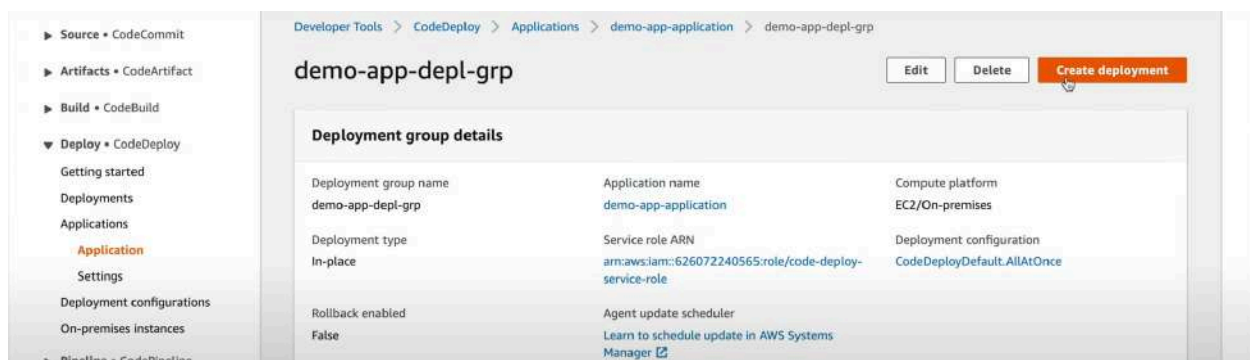


Now go to CodeBuild and build the application



Step 8 :- Now lets create deployment

Click on Create deployment



Select deployment group -> demo-app-depl-grp

Revision type -> AWS S3

Developer Tools > CodeDeploy > Applications > demo-app-application > Create deployment

Create deployment

Deployment settings

Application
demo-app-application

Deployment group
demo-app-depl-grp

Compute platform
EC2/On-premises

Deployment type
In-place

Revision type
☒ My application is stored in Amazon S3
 ☐ My application is stored in GitHub

Revision location
 Copy and paste the Amazon S3 bucket where your revision is stored
 s3://bucket-name/folder/object.[zip|tar|tgz]

Now go to Amazon S3 -> Buckets -> devops-zero-to-hero -> demo-app.zip -> Copy S3 URL

Amazon S3 > Buckets > devops-zero-to-hero > demo-app.zip

demo-app.zip Info

Copy S3 URI Download Open Object actions

Object overview

Owner trainwithshubham	S3 URI s3://devops-zero-to-hero/demo-app.zip
AWS Region US East (N. Virginia) us-east-1	Amazon Resource Name (ARN) arn:aws:s3:::devops-zero-to-hero/demo-app.zip
Last modified February 10, 2023, 17:56:37 (UTC+05:30)	Entity tag (Etag) b27bb32806c99295a16ed54e8d25443d
Size	

Now come back and put it in Revision location. Scroll down and press create deployment.

Deployment settings

Application
demo-app-application

Deployment group
demo-app-depl-grp

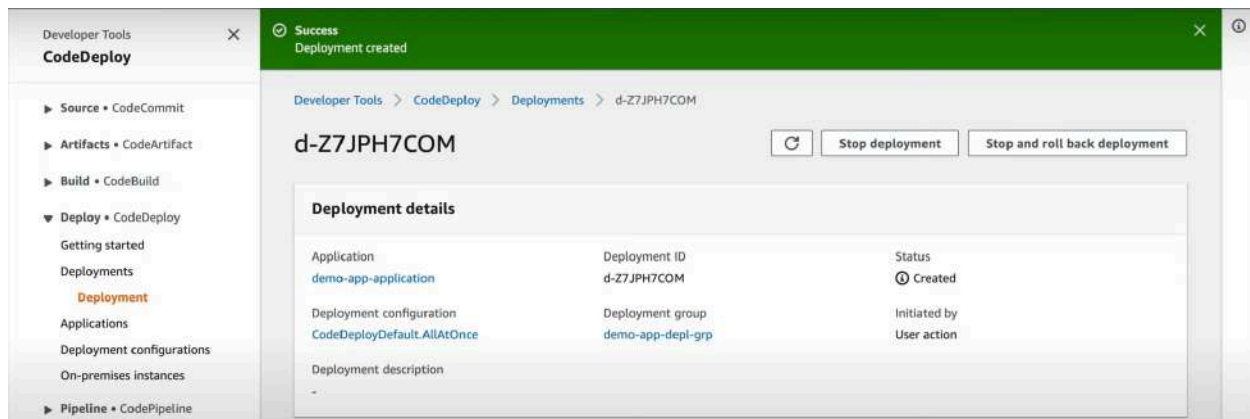
Compute platform
EC2/On-premises

Deployment type
In-place

Revision type
☒ My application is stored in Amazon S3
 ☐ My application is stored in GitHub

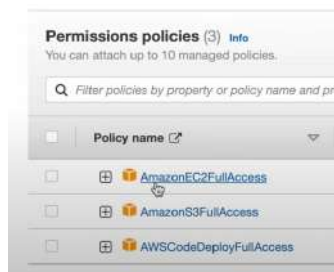
Revision location
 Copy and paste the Amazon S3 bucket where your revision is stored
 s3://devops-zero-to-hero/demo-app.zip

No previous revisions



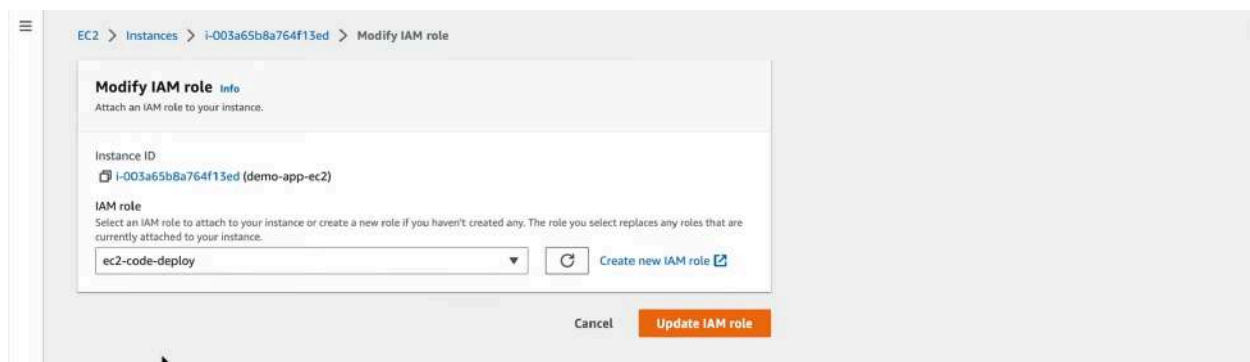
The status is still created , because EC2 does not have required access .

Hence we need to create an IAM role and we have to assign permissions related to EC2 ,S3 and CodeDeploy. Then we have to assign that role to EC2 on which we are deploying the application.



Step 9:- Now go to EC2 -> Select demo-app ec2 -> Under actions -> Security -> Modify IAM

Now select the IAM role with above specified permissions -> Click Update IAM role

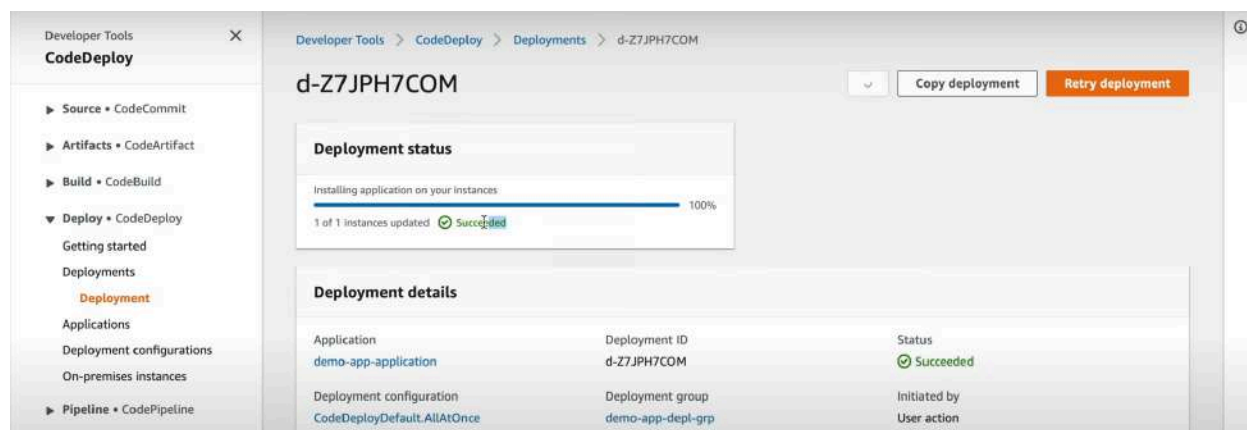


Now go to ec2-code-deploy server and restart the server

```
$ sudo service codedeploy-agent restart
```

```
$ sudo service codedeploy-agent status
```

Status should be active

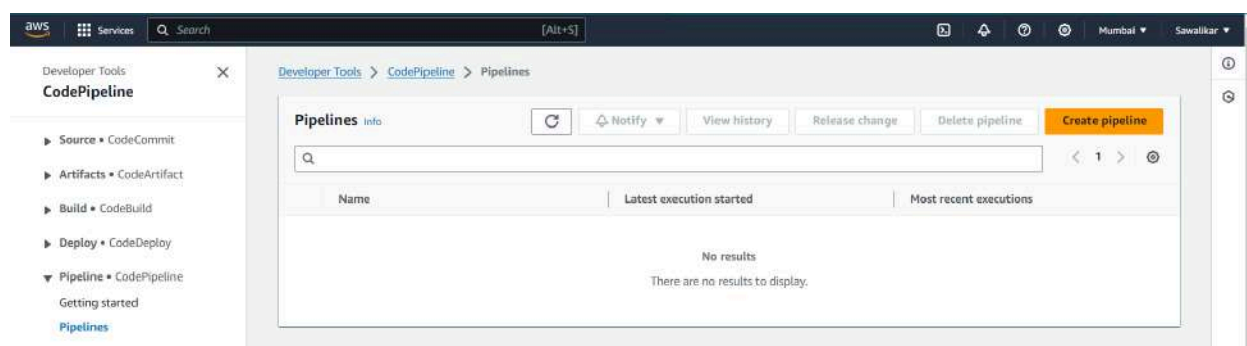


Now open the app on browser



Part 4 - CodePipeline

Search CodePipeline -> Click Create Pipeline



Give name to pipeline -> Select New service role -> Click

The screenshot shows the 'Choose pipeline settings' screen in the AWS CodePipeline console. The left sidebar lists the steps: Step 1: Choose pipeline settings (selected), Step 2: Add source stage, Step 3: Add build stage, Step 4: Add deploy stage, and Step 5: Review. The main content area is titled 'Choose pipeline settings' and 'Step 1 of 5'. It contains the following fields and options:

- Pipeline name:** A text input field containing 'demo-app-pipeline'. Below it, a note states: 'No more than 100 characters.'
- Pipeline type:** A section with the text: 'The pipeline type determines the pipeline structure and availability of parameters such as triggers. Pipeline type selection will impact features and pricing. Which pipeline is right for me?'. It has two radio buttons: 'V1' (unselected) and 'V2' (selected).
- Service role:** Two radio buttons: 'New service role' (selected) with the subtext 'Create a service role in your account.', and 'Existing service role' (unselected) with the subtext 'Choose an existing service role from your account.'
- Role name:** A text input field containing 'AWSCodePipelineServiceRole-ap-south-1-demo-app-pipeline'. Below it, a note says: 'Type your service role name'.
- Permissions:** A checkbox labeled 'Allow AWS CodePipeline to create a service role so it can be used with this new pipeline' is checked.

Source provider -> AWS CodeCommit

Repository name -> demo-app

Branch -> master

The screenshot shows the 'Add source stage' screen in the AWS CodePipeline console. The left sidebar lists the steps: Step 1: Choose pipeline settings, Step 2: Add source stage (selected), Step 3: Add build stage, Step 4: Add deploy stage, and Step 5: Review. The main content area is titled 'Add source stage' and 'Step 2 of 5'. It contains the following fields and options:

- Source provider:** A dropdown menu with 'AWS CodeCommit' selected. Below it, a note states: 'This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.'
- Repository name:** A text input field containing 'demo-app' with a clear (X) button. Below it, a note says: 'Choose a repository that you have already created where you have pushed your source code.'
- Branch name:** A text input field containing 'master' with a clear (X) button. Below it, a note says: 'Choose a branch of the repository.'

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

☐ Amazon CloudWatch Events (recommended)
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs

☒ AWS CodePipeline
Use AWS CodePipeline to check periodically for changes

Output artifact format
Choose the output artifact format.

☒ CodePipeline default
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

☐ Full clone
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

Cancel Previous Next

Select AWS CodeBuild -> Select project name -> Single build -> Next

Build - optional

Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

AWS CodeBuild

Region
Asia Pacific (Mumbai)

Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.

demo-app-build or Create project

Environment variables - optional
Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. [Learn more](#)

Add environment variable

Build type

☒ Single build
Triggers a single build.

☐ Batch build
Triggers multiple builds as a single execution.

Cancel Previous Skip build stage Next

Add deploy stage

Deploy - optional

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS CodeDeploy

Region
US East (N. Virginia)

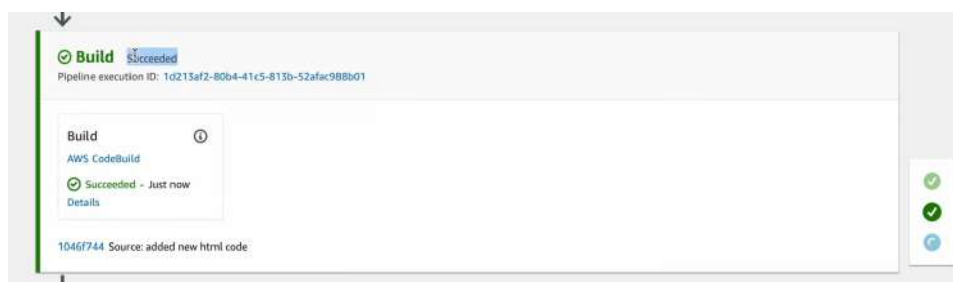
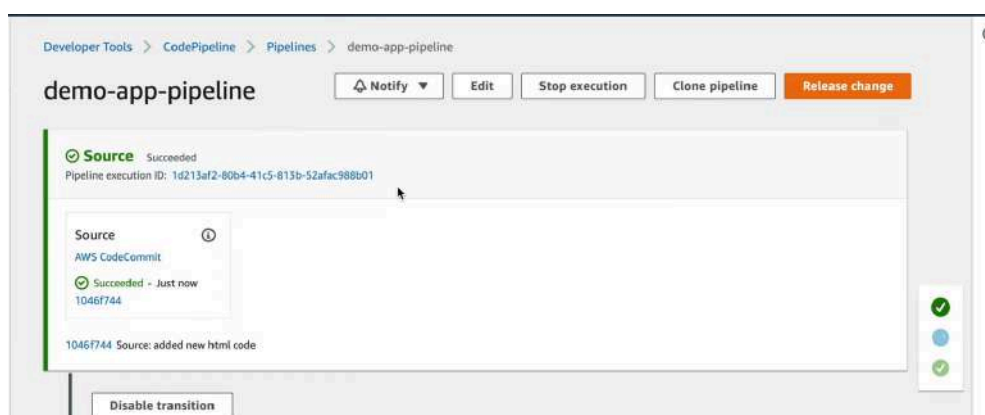
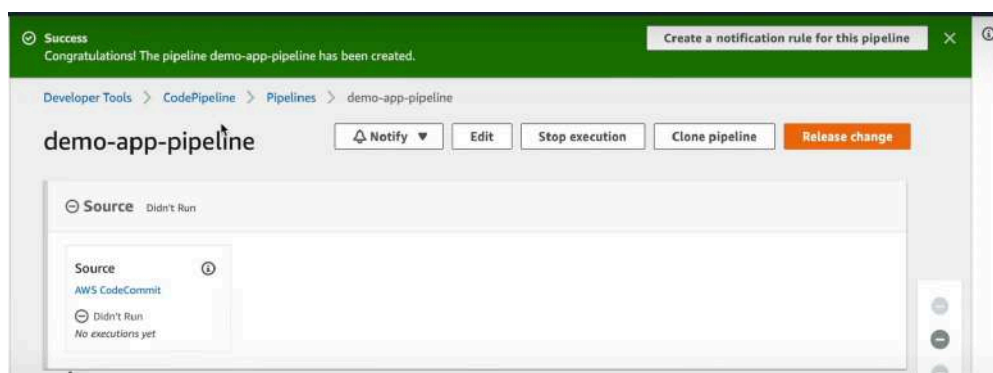
Application name
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.

demo-app-application

Deployment group
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.

demo-app-depl-grp

Review the pipeline and create



Copy the ip address of EC2 and paste on browser

My Demo App

this is nice