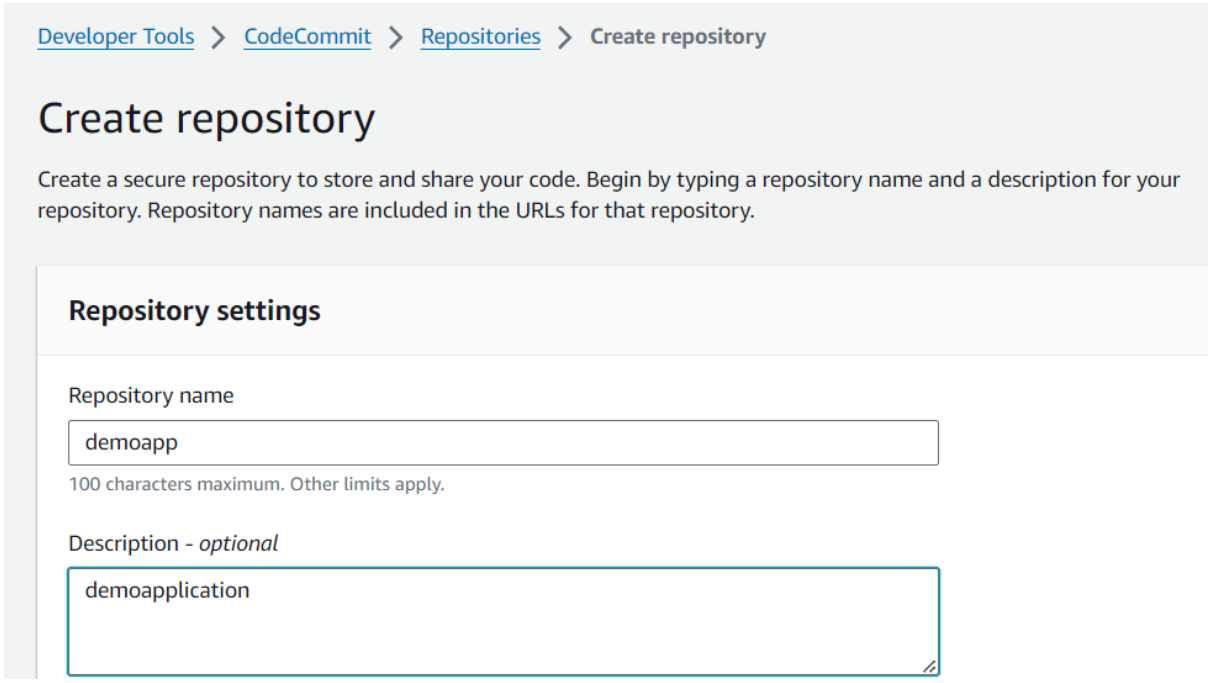
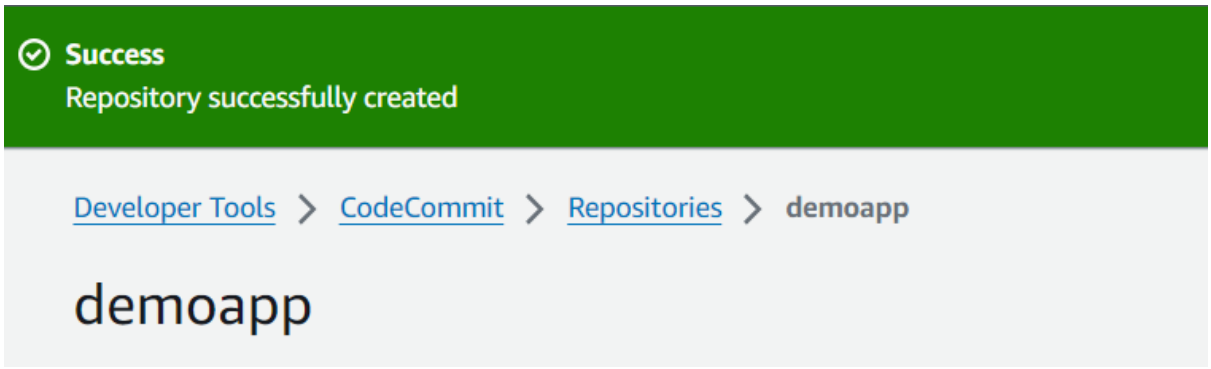
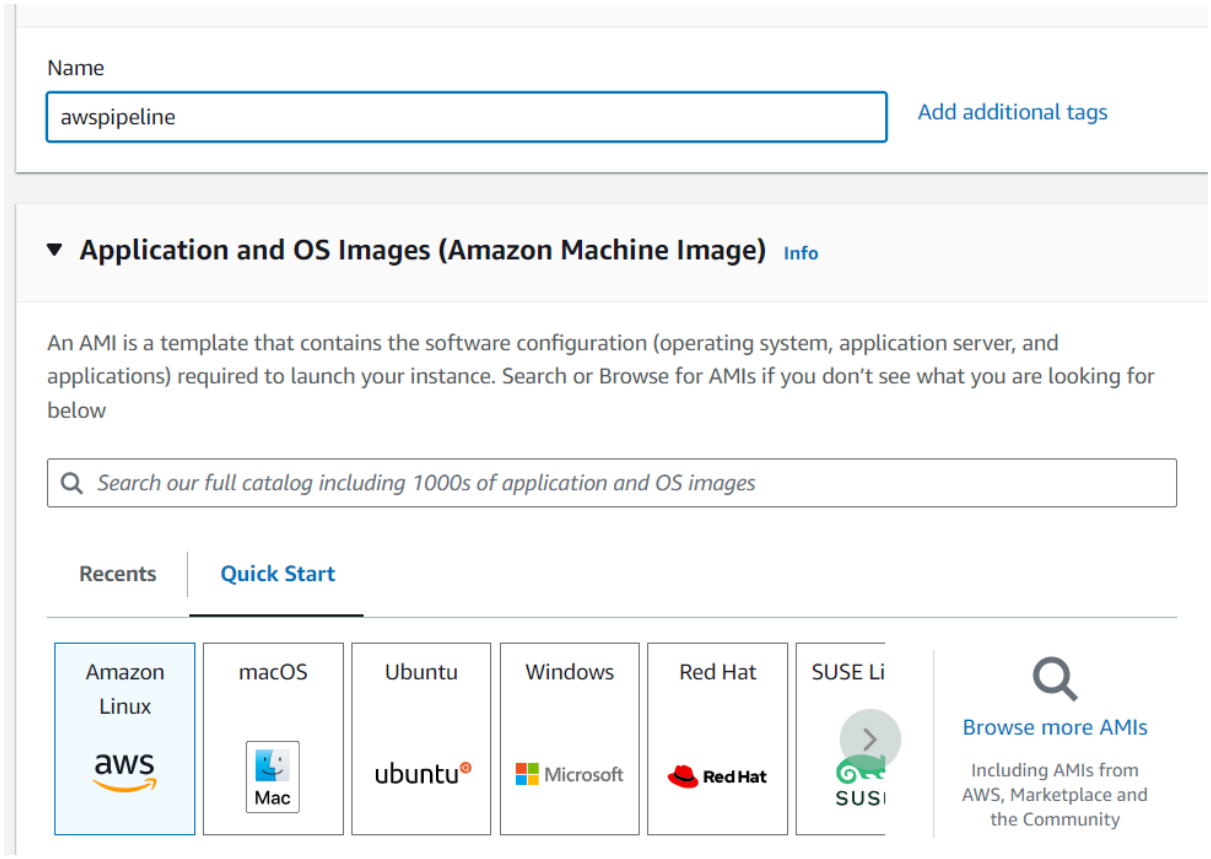
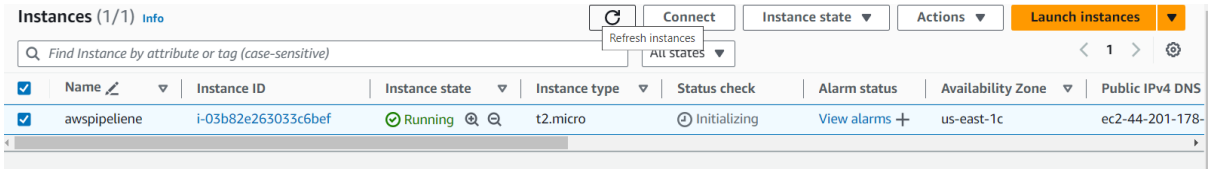
Task 16: Deploy a simple Nginx application using AWS code commit and deploy & access via browser

1. Create code commit repo:

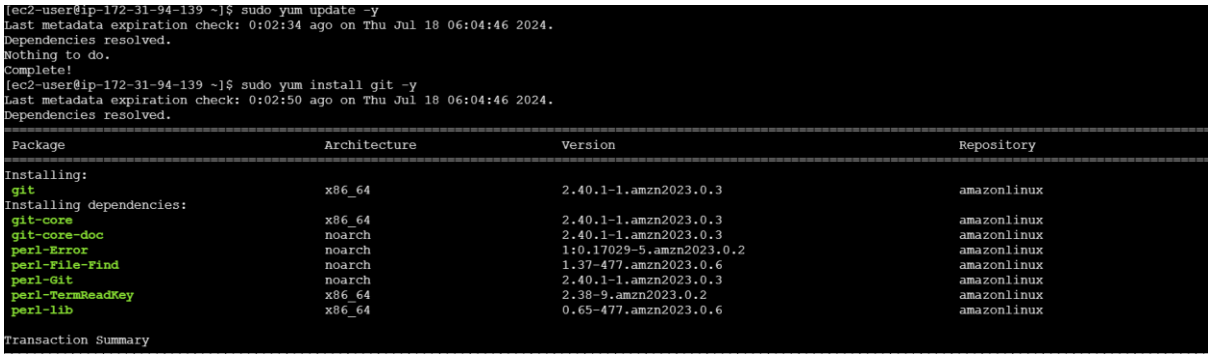


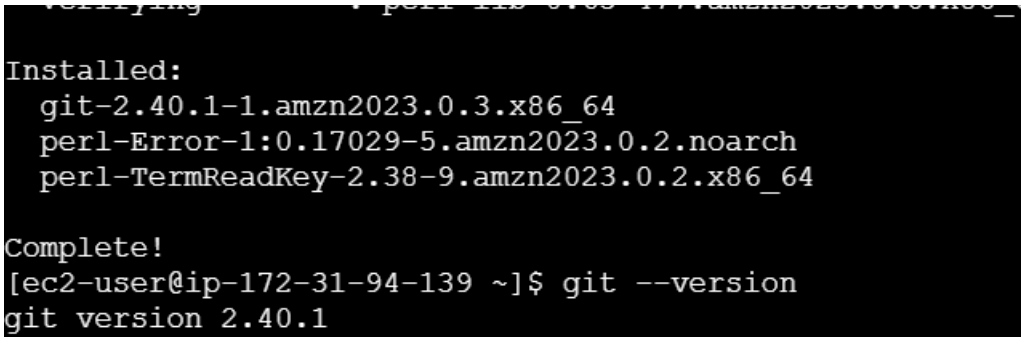




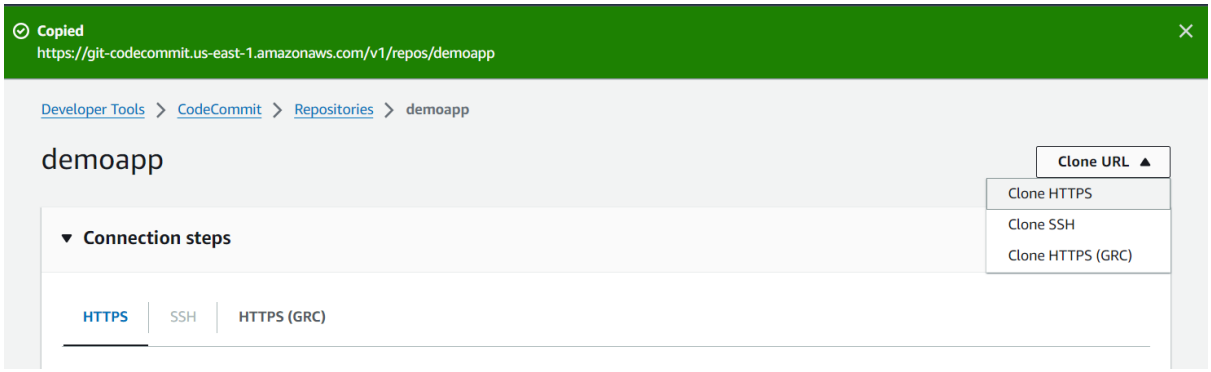


Connect with that instance and install git

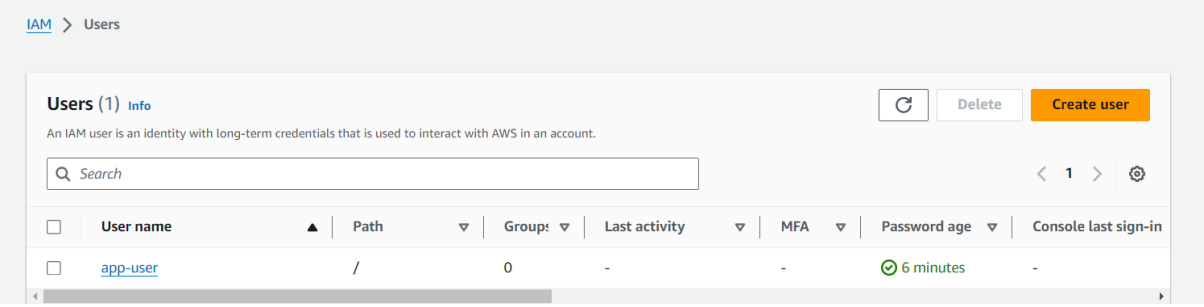




Clone the created repo in amazon linux



IAM user can be created [for clone the repo]. During clone ask username and password so only we created IAM user



Now clone that repo.

```
[ec2-user@ip-172-31-94-139 ~]$ git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/demoapp
Cloning into 'demoapp'...
Username for 'https://git-codecommit.us-east-1.amazonaws.com': app-user-at-381492100082
Password for 'https://app-user-at-381492100082@git-codecommit.us-east-1.amazonaws.com':
warning: You appear to have cloned an empty repository.
[ec2-user@ip-172-31-94-139 ~]$
```

Inside demoapp repo we added few files [**All files written in amazon linux instance]**

I. Index.html
II. Buildspec.yaml
III. Appspec.yml

- install_nginx.sh
- Start_nginx.sh:

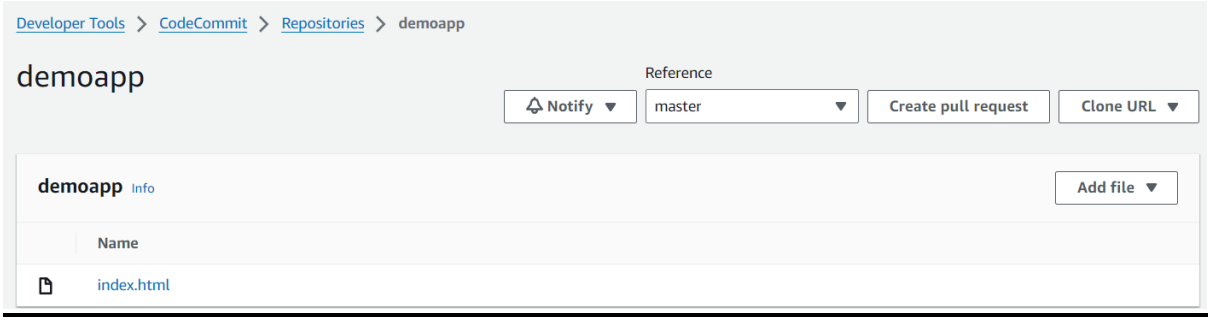
i. <u>Index.html:</u>

```
[ec2-user@ip-172-31-94-139 ~]$ ls
demoapp
[ec2-user@ip-172-31-94-139 ~]$ vi index.html
```

```
hai
good evening
how are you?
~
~
```

```
[ec2-user@ip-172-31-94-139 ~]$ git add index.html
[ec2-user@ip-172-31-94-139 ~]$ git commit -m "add index.html file"
[master (root-commit) 970c701] add index.html file
 Committer: EC2 Default User <ec2-user@ip-172-31-94-139.ec2.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, 3 insertions(+)
 create mode 100644 index.html
[ec2-user@ip-172-31-94-139 ~]$
```

```
[ec2-user@ip-172-31-94-139 ~]$ git remote -v
[ec2-user@ip-172-31-94-139 ~]$ git remote add origin https://git-codecommit.us-east-1.amazonaws.com/v1/repos/demoapp
[ec2-user@ip-172-31-94-139 ~]$ git push --set-upstream origin master
Username for 'https://git-codecommit.us-east-1.amazonaws.com': app-user-at-381492100082
Password for 'https://app-user-at-381492100082@git-codecommit.us-east-1.amazonaws.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 270 bytes | 270.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/demoapp
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
[ec2-user@ip-172-31-94-139 ~]$ git push
Username for 'https://git-codecommit.us-east-1.amazonaws.com': app-user-at-381492100082
Password for 'https://app-user-at-381492100082@git-codecommit.us-east-1.amazonaws.com':
Everything up-to-date
[ec2-user@ip-172-31-94-139 ~]$
```

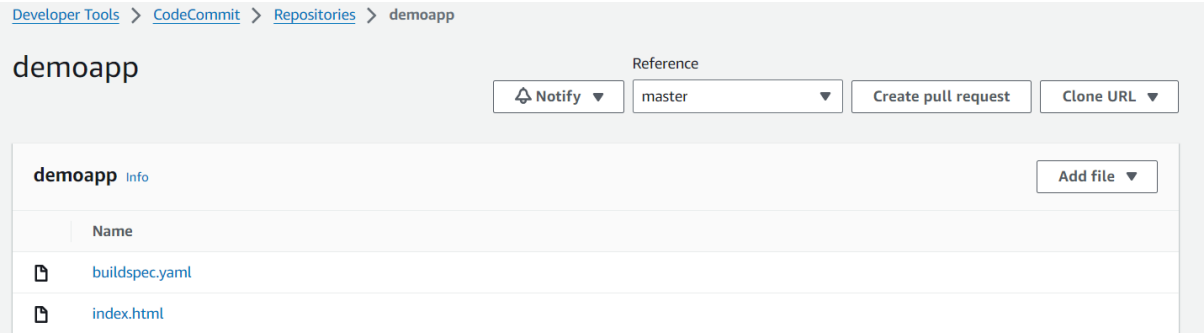


II) Buildspec.yaml

```
version: 0.2
phases:
   install:
     commands:
       - echo  "Installing Nginx server"
       - sudo apt-get update
       - sudo apt-get install nginx -y

   build:
     commands:
       - echo "Build started on 18 july"
       - sudo cp index.html /var/www/html/
   post_build:
     commands:
       - echo "Configuring Nginx"
artifacts:
     files:
       - '**/*'
```

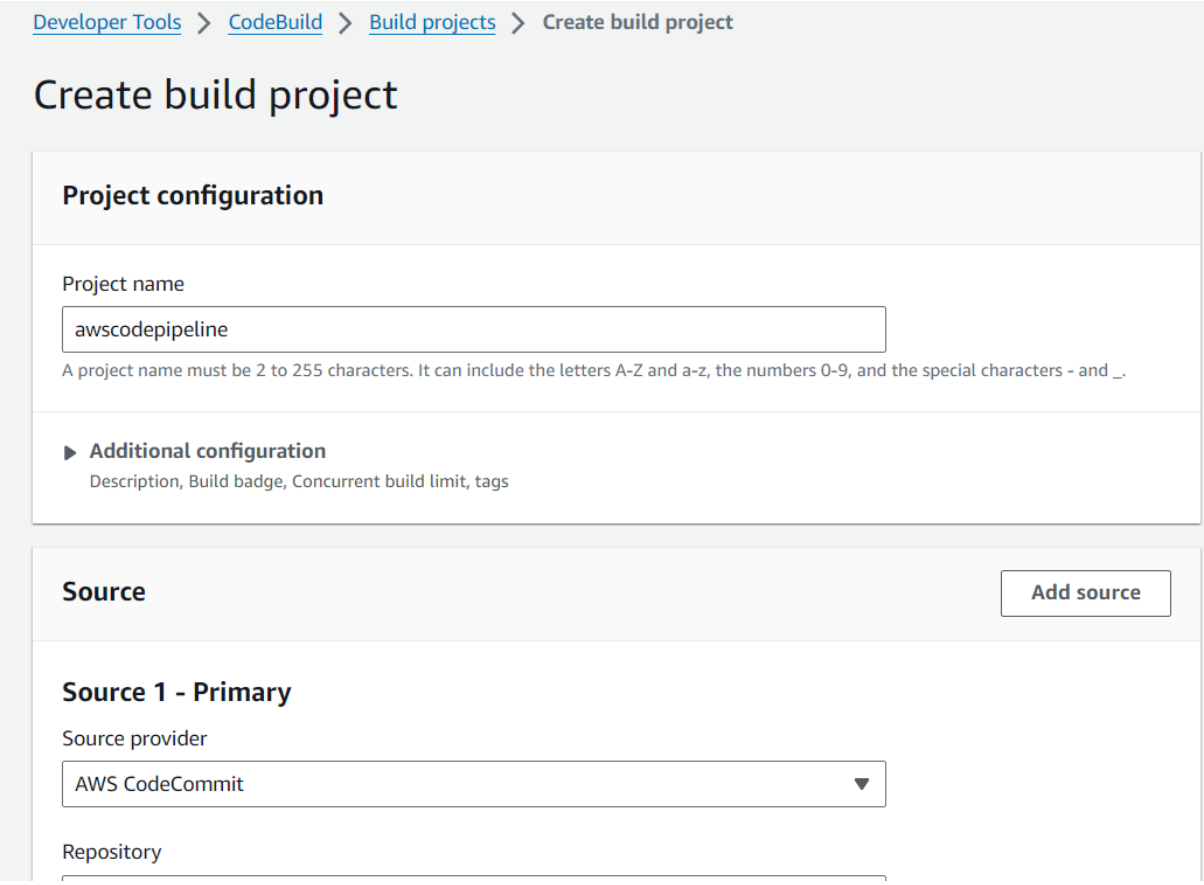
```
[ec2-user@ip-172-31-94-139 ~]$ vi buildspec.yaml
[ec2-user@ip-172-31-94-139 ~]$ git add buildspec.yaml
[ec2-user@ip-172-31-94-139 ~]$ git commit -m "add buildspec.yaml"
[master 5765ef5] add buildspec.yaml
 Committer: EC2 Default User <ec2-user@ip-172-31-94-139.ec2.internal>
Your name and email address 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, 19 insertions(+)
 create mode 100644 buildspec.yaml
[ec2-user@ip-172-31-94-139 ~]$
```

```
[ec2-user@ip-172-31-94-139 ~]$ git push
Username for 'https://git-codecommit.us-east-1.amazonaws.com': app-user-at-381492100082
Password for 'https://app-user-at-381492100082@git-codecommit.us-east-1.amazonaws.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 498 bytes | 498.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/demoapp
   970c701..5765ef5  master -> master
[ec2-user@ip-172-31-94-139 ~]$
```



2. Code build:

Goto code build → click create project



Source

Add source

Source 1 - Primary

Source provider

AWS CodeCommit

Repository

demoapp

Reference type

Choose the source version reference type that contains your source code.

- (•) Branch
- ( ) 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

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

Operating system

Amazon Linux

Runtime(s)

Standard

Image version

Always use the latest image for this runtime version

Use GPU-enhanced compute

Service role

New service role
Create a service role in your account

Existing service role
Choose an existing service role from your account

Role name

codepipeline-servicerole

Type your service role name

Additional configuration
Timeout, privileged, certificate, VPC, compute type, environment variables, file systems

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

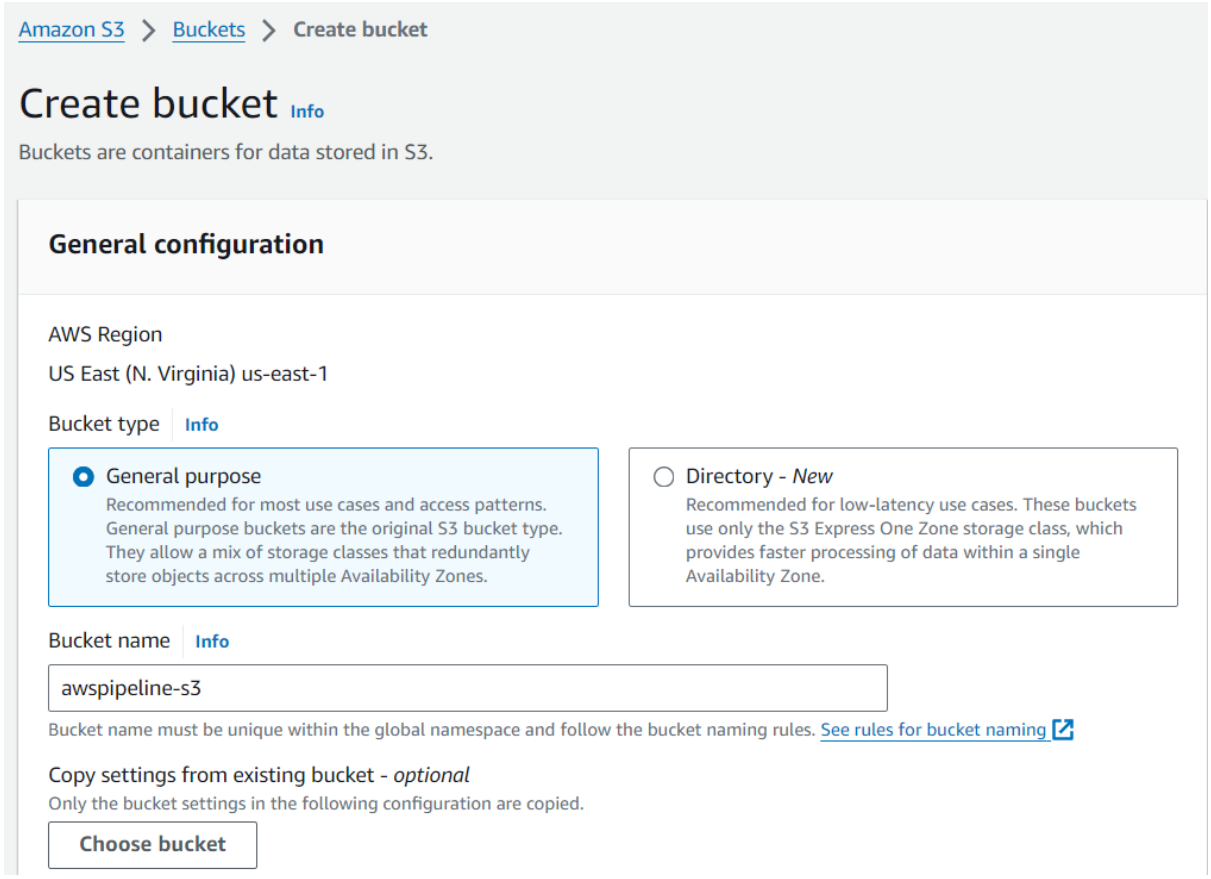
Buildspec name - *optional*

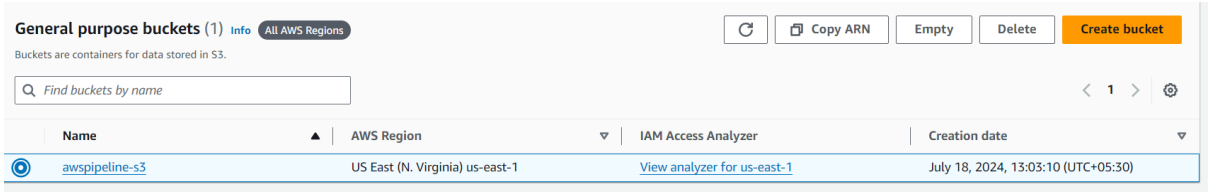
By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).

buildspec.yml

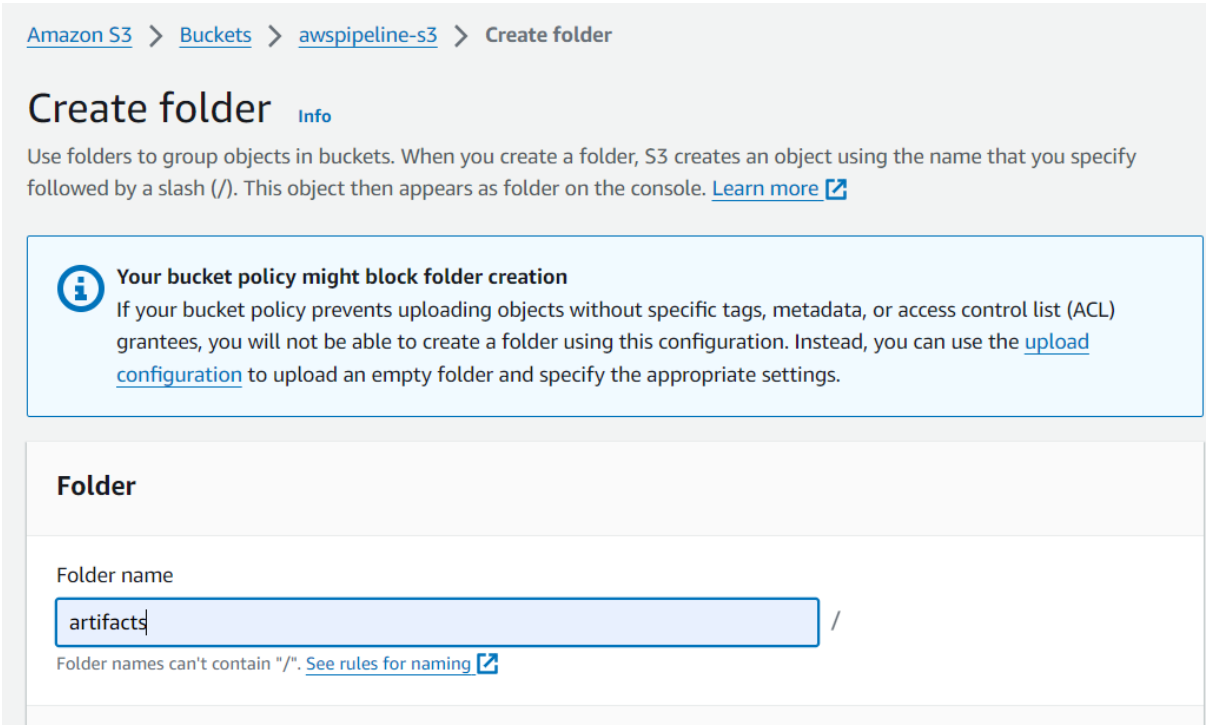
Artifact asks in which s3 bucket stores this.so we create s3 bucket

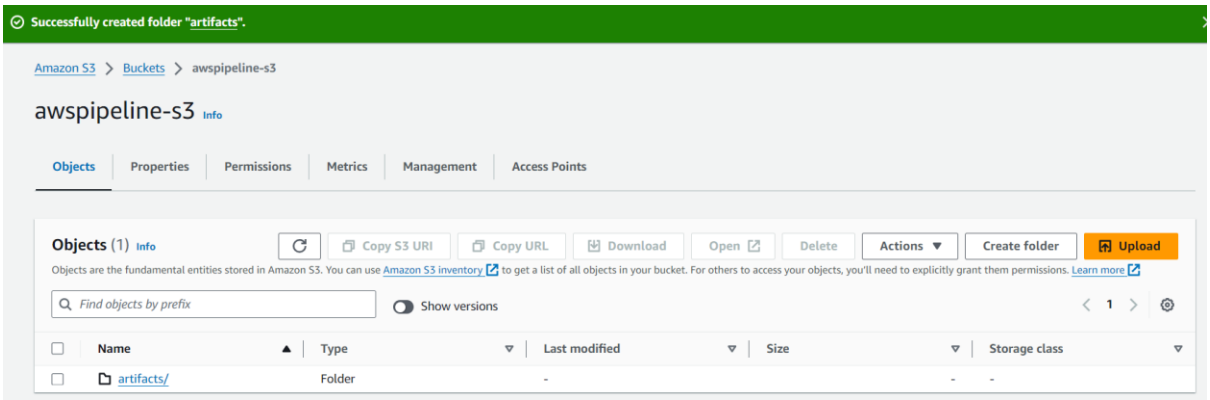
Create s3 bucket [awspipeline-s3]





Click create folder.





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

awspipeline-s3

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

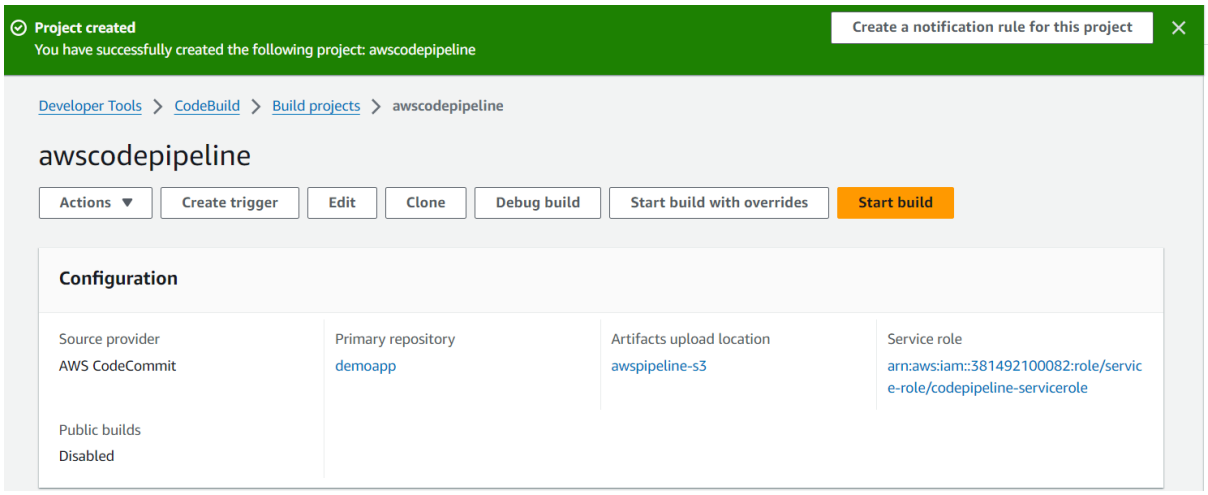
Use the artifact name specified in the buildspec file

Path - *optional*

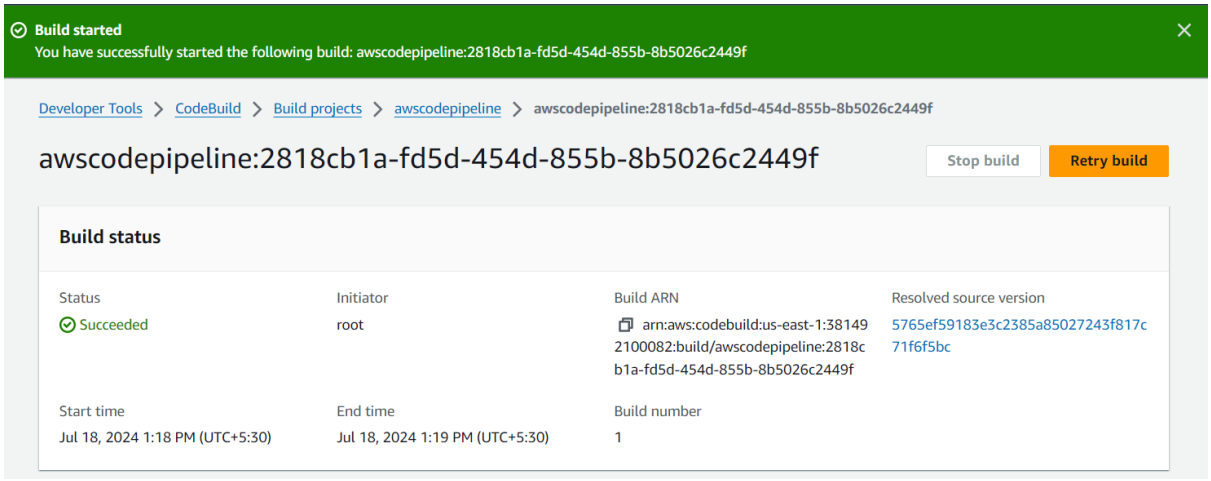
The path to the build output ZIP file or folder.

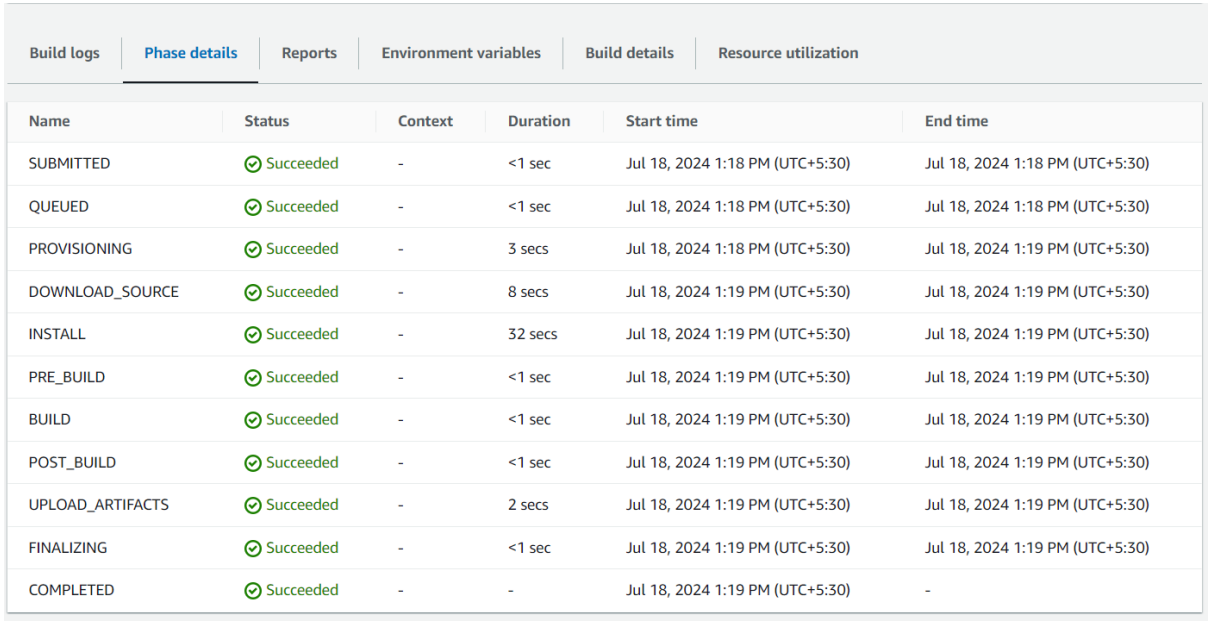
Example: MyPath/MyArtifact.zip.

Click create build project. Project created



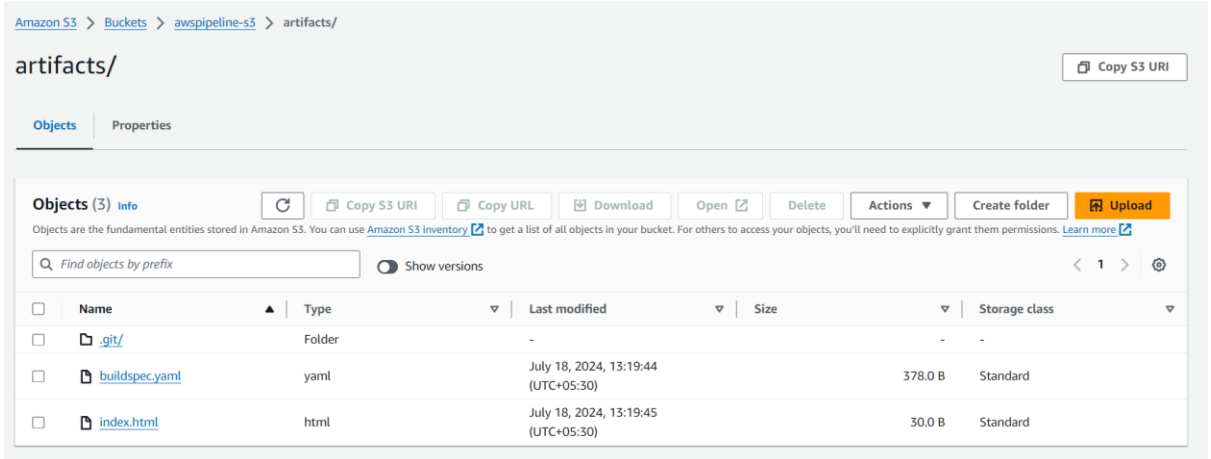
Click start build





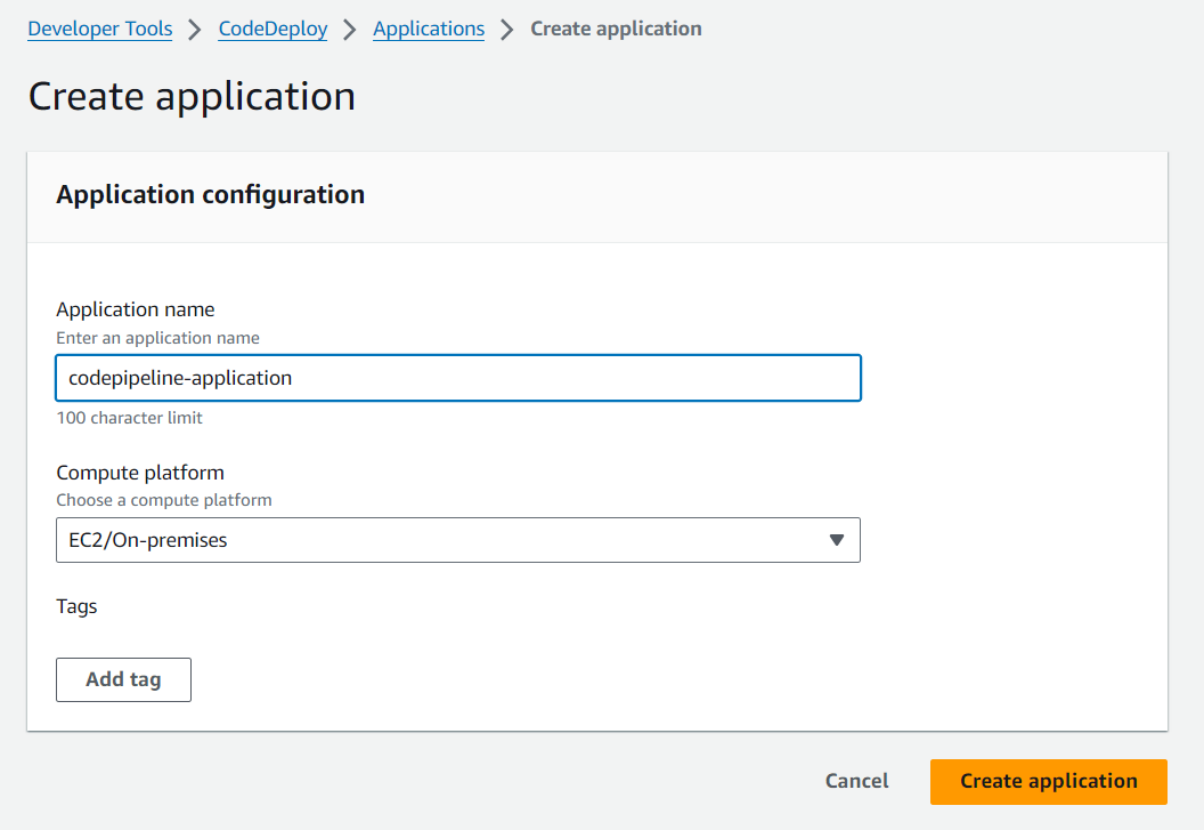
Name	Status	Context	Duration	Start time	End time
SUBMITTED	Succeeded	-	<1 sec	Jul 18, 2024 1:18 PM (UTC+5:30)	Jul 18, 2024 1:18 PM (UTC+5:30)
QUEUED	Succeeded	-	<1 sec	Jul 18, 2024 1:18 PM (UTC+5:30)	Jul 18, 2024 1:18 PM (UTC+5:30)
PROVISIONING	Succeeded	-	3 secs	Jul 18, 2024 1:18 PM (UTC+5:30)	Jul 18, 2024 1:19 PM (UTC+5:30)
DOWNLOAD_SOURCE	Succeeded	-	8 secs	Jul 18, 2024 1:19 PM (UTC+5:30)	Jul 18, 2024 1:19 PM (UTC+5:30)
INSTALL	Succeeded	-	32 secs	Jul 18, 2024 1:19 PM (UTC+5:30)	Jul 18, 2024 1:19 PM (UTC+5:30)
PRE_BUILD	Succeeded	-	<1 sec	Jul 18, 2024 1:19 PM (UTC+5:30)	Jul 18, 2024 1:19 PM (UTC+5:30)
BUILD	Succeeded	-	<1 sec	Jul 18, 2024 1:19 PM (UTC+5:30)	Jul 18, 2024 1:19 PM (UTC+5:30)
POST_BUILD	Succeeded	-	<1 sec	Jul 18, 2024 1:19 PM (UTC+5:30)	Jul 18, 2024 1:19 PM (UTC+5:30)
UPLOAD_ARTIFACTS	Succeeded	-	2 secs	Jul 18, 2024 1:19 PM (UTC+5:30)	Jul 18, 2024 1:19 PM (UTC+5:30)
FINALIZING	Succeeded	-	<1 sec	Jul 18, 2024 1:19 PM (UTC+5:30)	Jul 18, 2024 1:19 PM (UTC+5:30)
COMPLETED	Succeeded	-	-	Jul 18, 2024 1:19 PM (UTC+5:30)	-

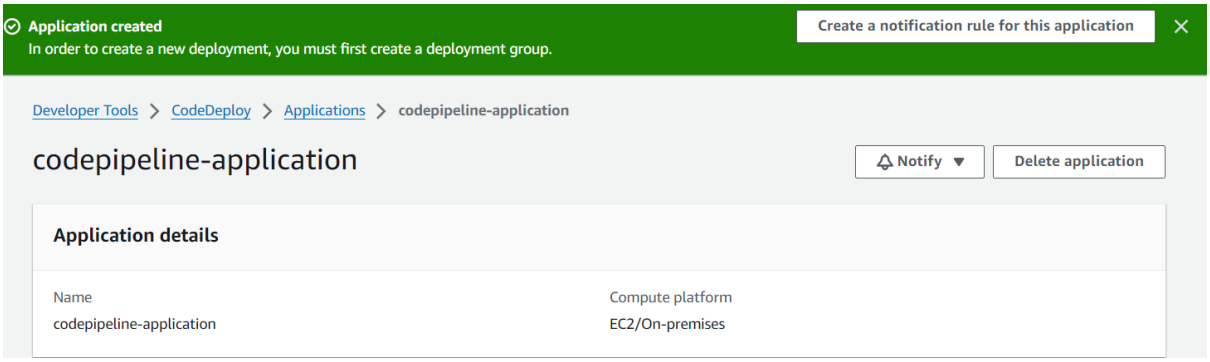
Once the application build – file stored in artifacts folder



3. Code deploy:

Click code deploy →create an application



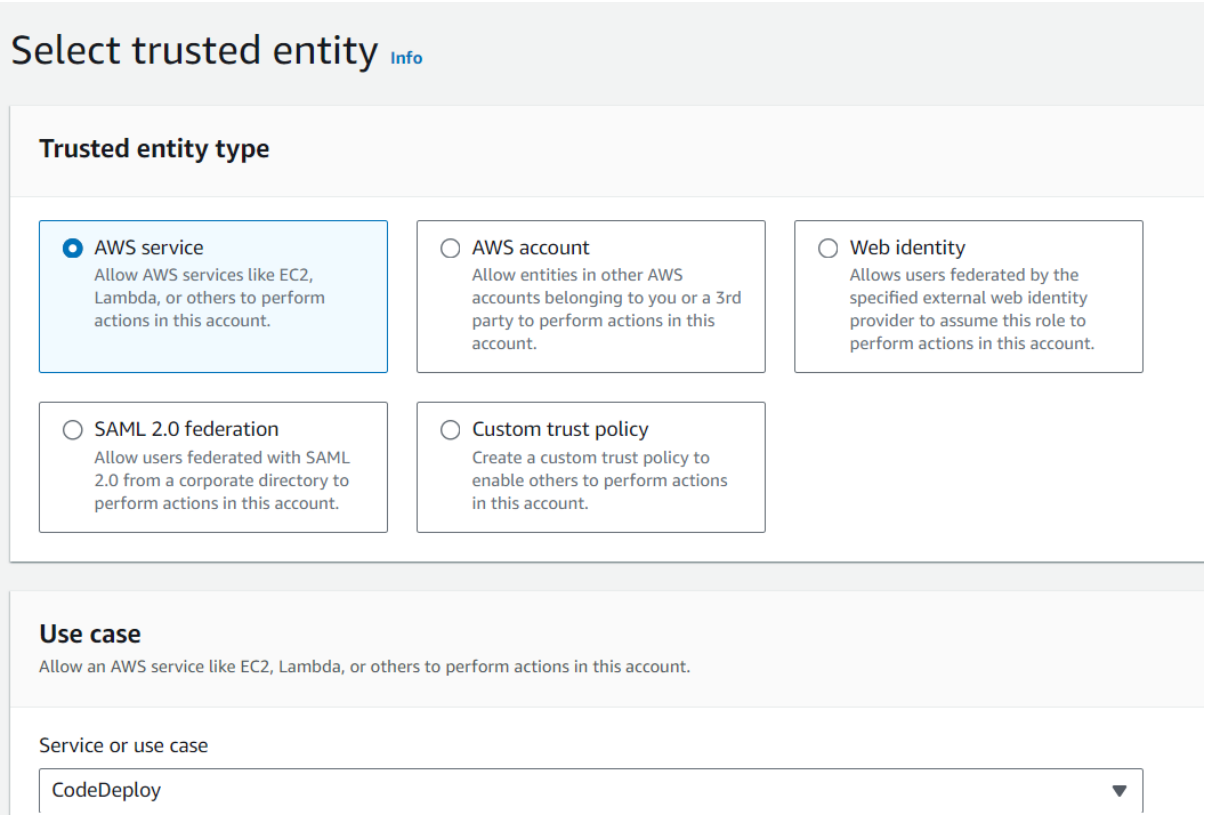


Then create deployment group. In d.g ask about

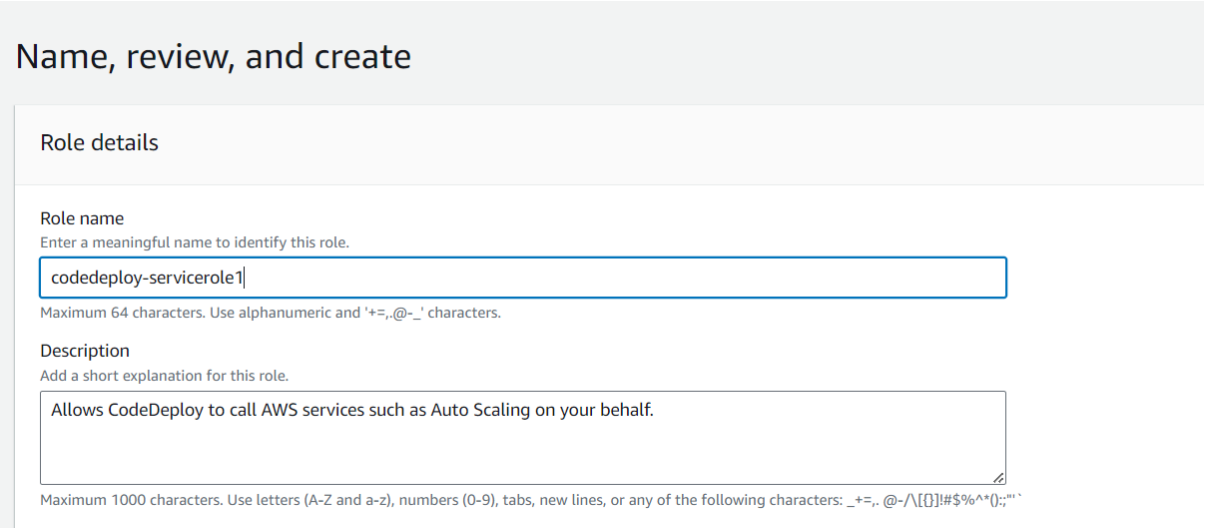
1. **2 service role has to create.so we create this role in IAM**
2. **ec2 instance**

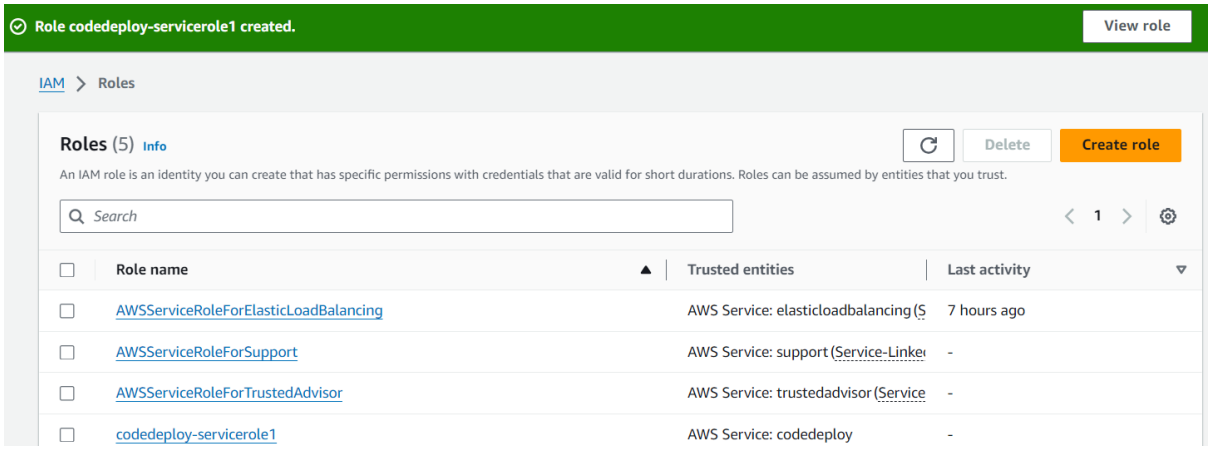
Code deploy service role -1:

Goto IAM .click roles →create roles

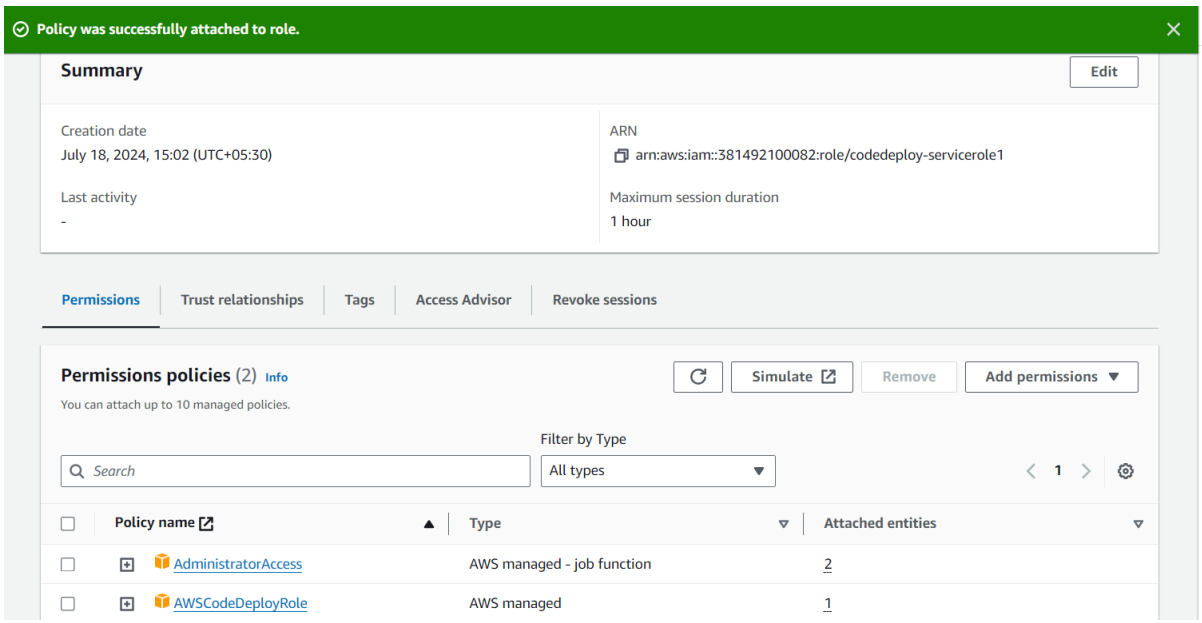


Click next

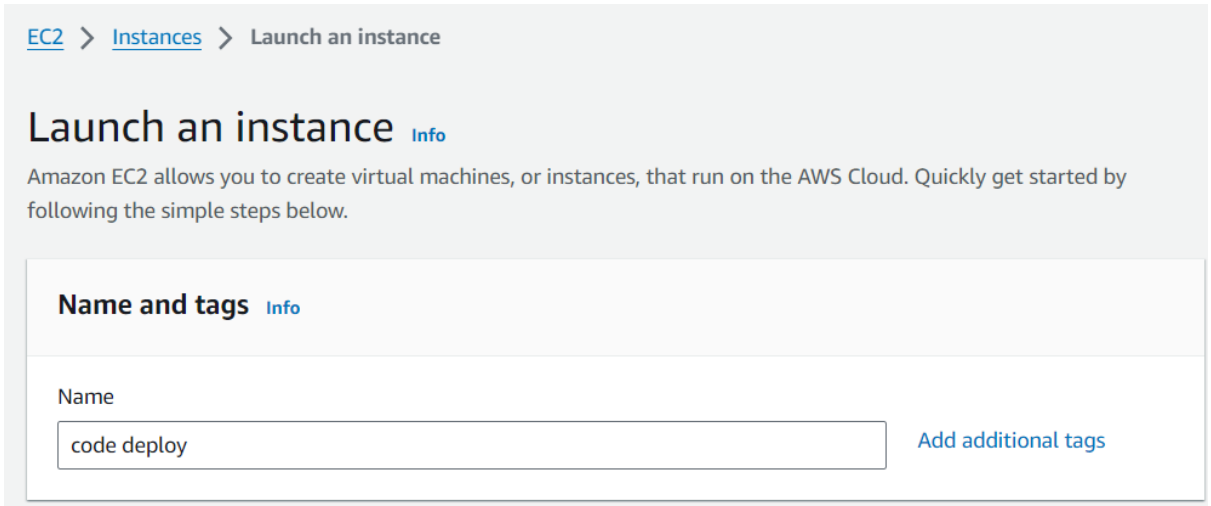




Attach the permissions[administration access] to this roles



Create new instance for code deploy:



Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Li

Browse more AMIs

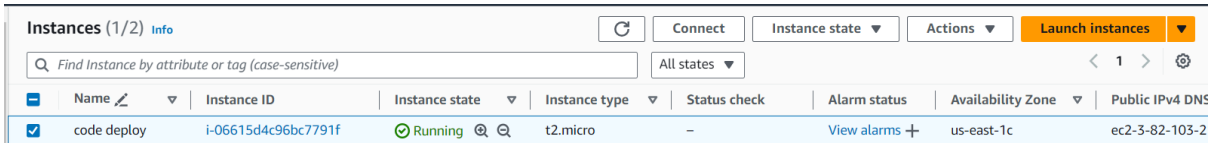
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

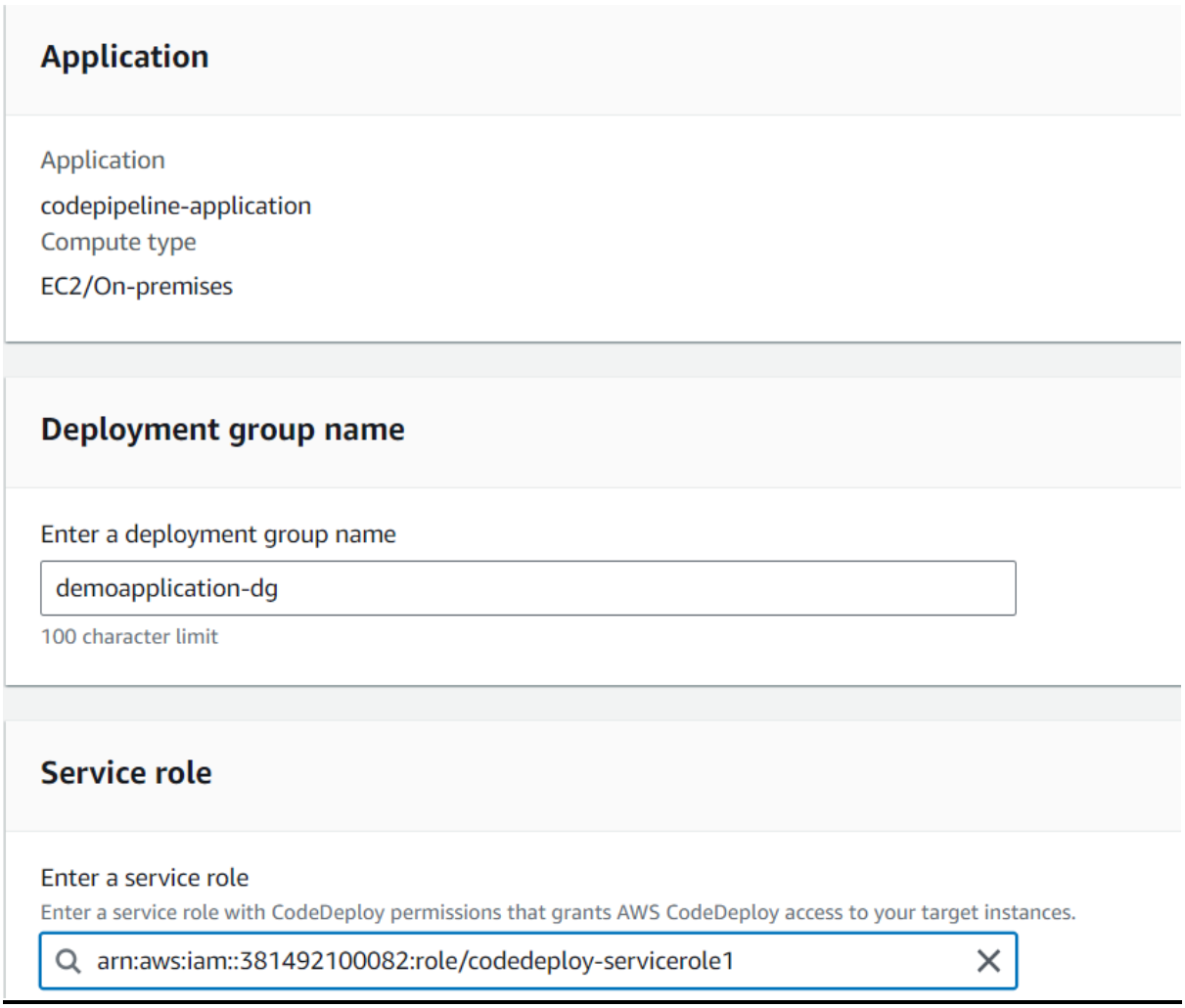
Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible

ami-0a0e5d9c7acc336f1 (64-bit (x86)) / ami-070f589e4b4a3fece (64-bit (Arm))

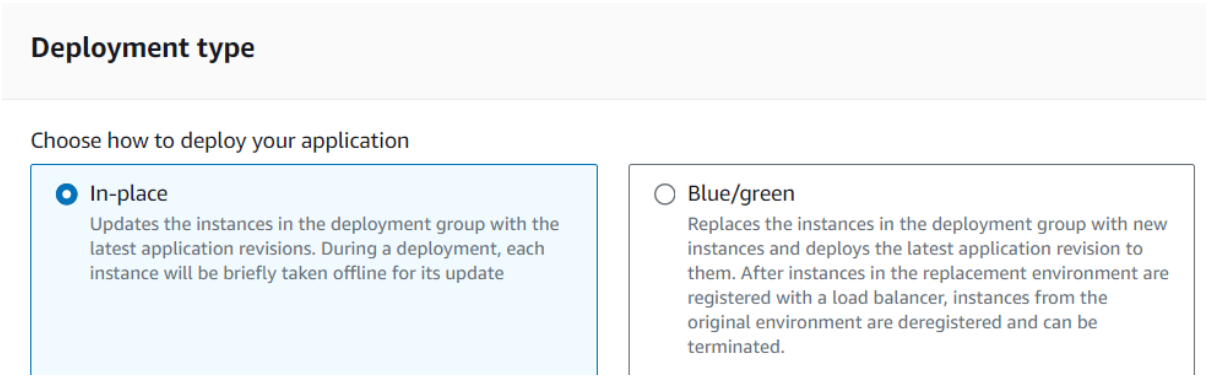
Virtualization: hvm ENA enabled: true Root device type: ebs

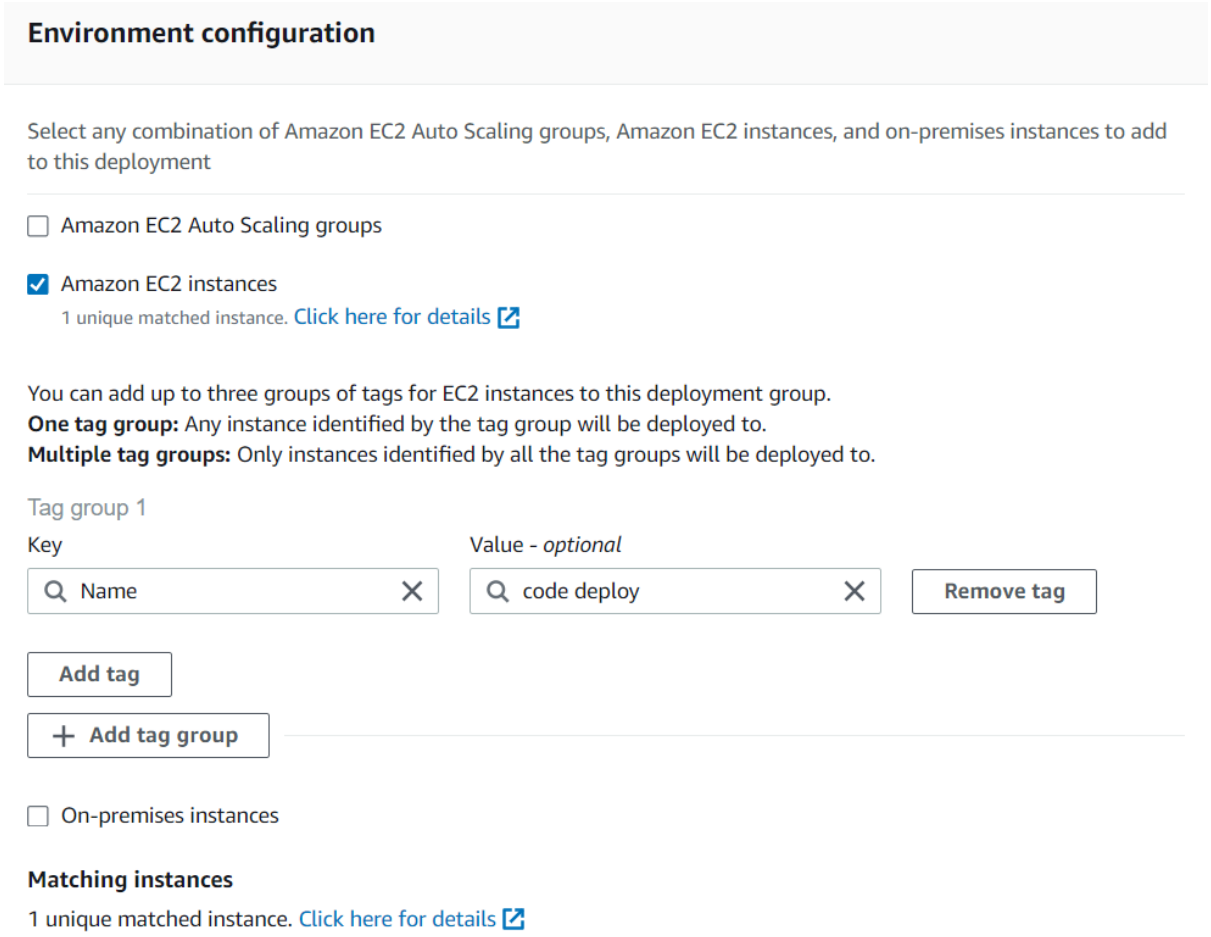


create deployment group:

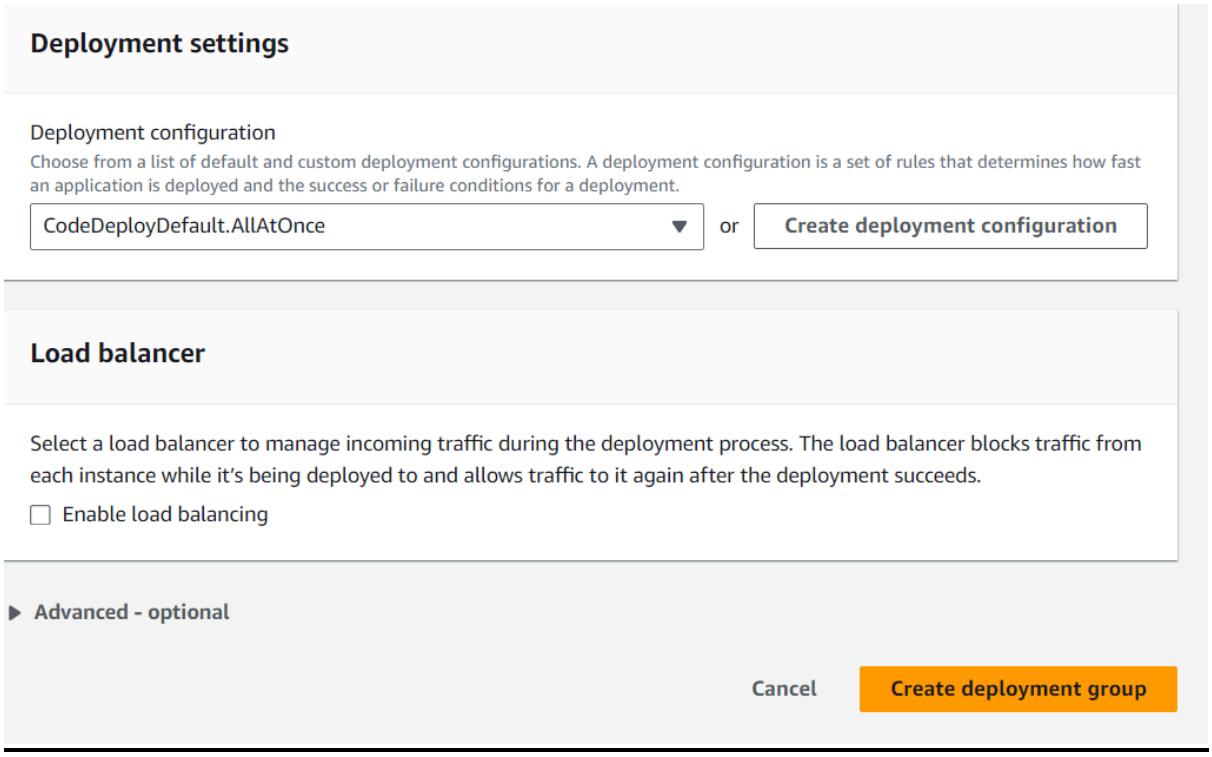


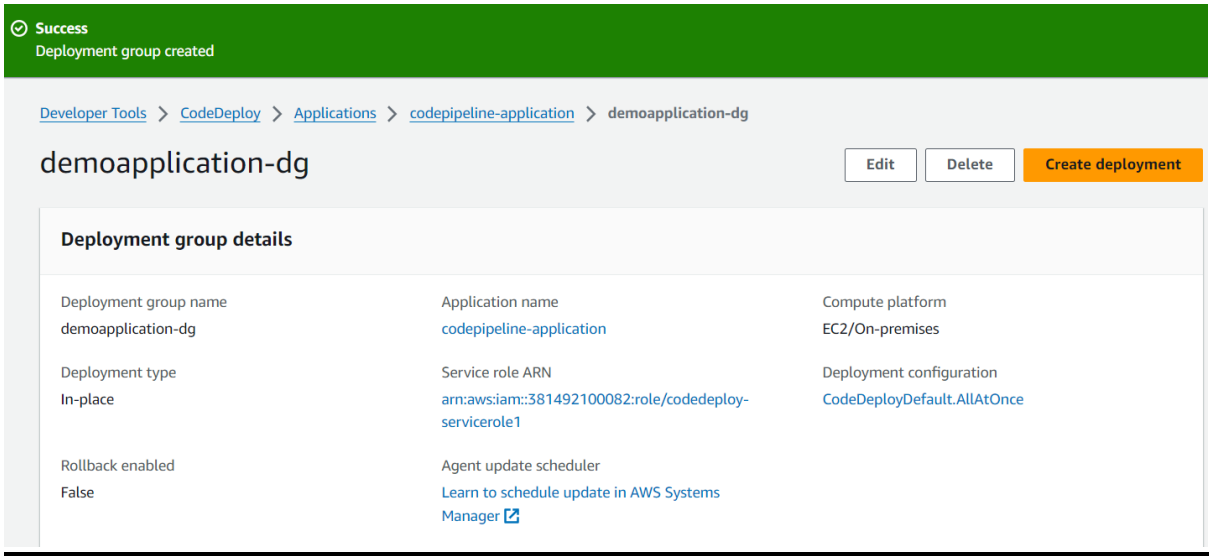
Attached the service role we already created





Click create deployment group





Install code deploy agent. For this we write shell script.connect code deploy ec2[ubuntu server]

Install.sh :

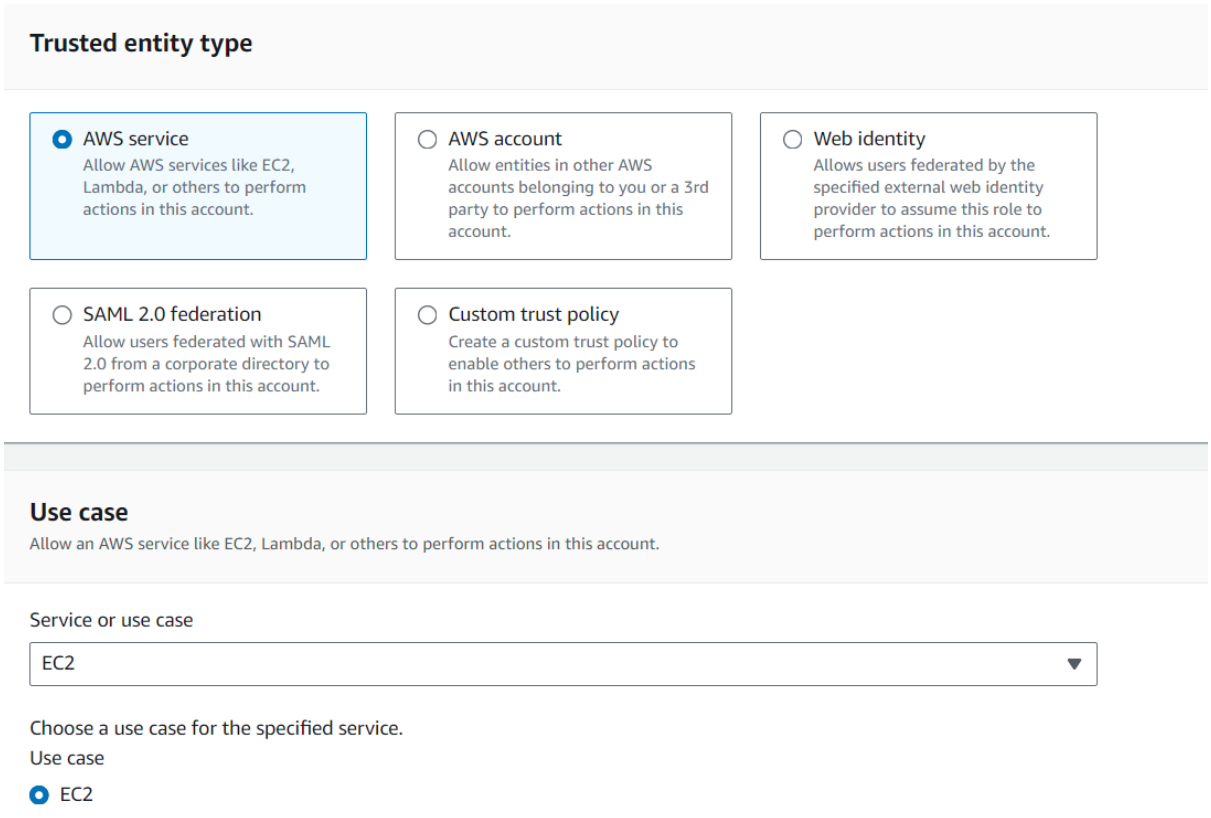
```
#!/bin/bash

# Download and install CodeDeploy agent
sudo yum update -y
sudo yum install -y ruby wget
wget https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install
chmod +x ./install
sudo ./install auto

# Start CodeDeploy agent
sudo service codedeploy-agent start

# Optional: Verify CodeDeploy agent status
sudo service codedeploy-agent status
```

```
[ec2-user@ip-172-31-94-139 ~]$ vi install.sh
[ec2-user@ip-172-31-94-139 ~]$ chmod +x install.sh
[ec2-user@ip-172-31-94-139 ~]$ ./install.sh
Last metadata expiration check: 4:28:03 ago on Thu Jul 18 06:04:46 2024.
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 4:28:04 ago on Thu Jul 18 06:04:46 2024.
Package wget-1.21.3-1.amzn2023.0.3.x86_64 is already installed.
Dependencies resolved.
================================================================================
 Package                      Architecture  Version                    Repository
================================================================================
Installing:
 ruby3.2                      x86_64        3.2.2-180.amzn2023.0.2     amazonlinux
Installing dependencies:
 ruby3.2-default-gems         noarch        3.2.2-180.amzn2023.0.2     amazonlinux
 ruby3.2-libs                 x86_64        3.2.2-180.amzn2023.0.2     amazonlinux
 ruby3.2-rubygem-io-console   x86_64        0.6.0-180.amzn2023.0.2     amazonlinux
 ruby3.2-rubygem-json         x86_64        2.6.3-180.amzn2023.0.2     amazonlinux
 ruby3.2-rubygem-psych        x86_64        5.0.1-180.amzn2023.0.2     amazonlinux
Installing weak dependencies:
 ruby3.2-rubygem-bigdecimal   x86_64        3.1.3-180.amzn2023.0.2     amazonlinux
 ruby3.2-rubygem-bundler      noarch        2.4.10-180.amzn2023.0.2    amazonlinux
 ruby3.2-rubygem-rdoc         noarch        6.5.0-180.amzn2023.0.2     amazonlinux
 ruby3.2-rubygems             noarch        3.4.10-180.amzn2023.0.2    amazonlinux
```

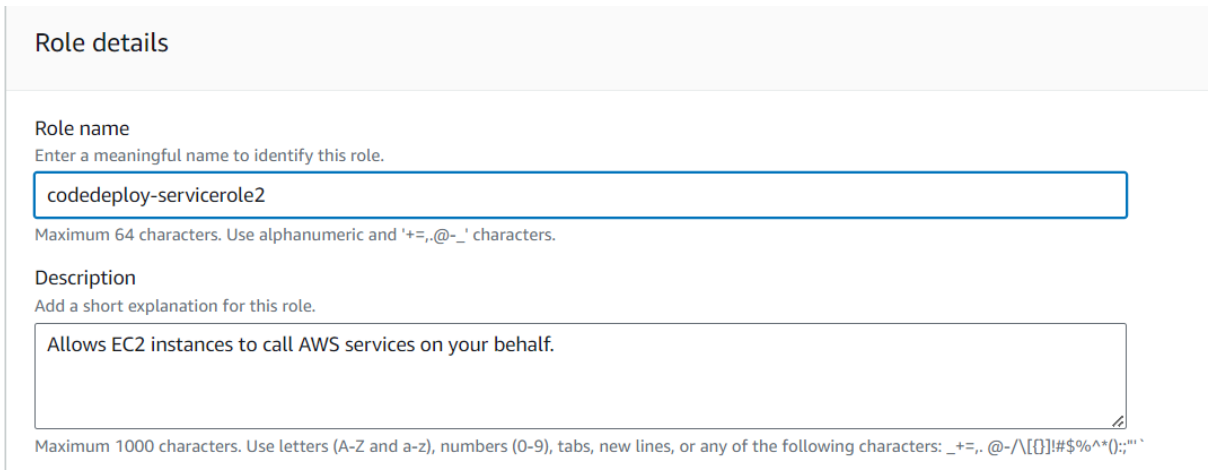
```
ubuntu@ip-172-31-80-155:~$ vi install.sh
ubuntu@ip-172-31-80-155:~$ chmod +x install.sh
ubuntu@ip-172-31-80-155:~$ ./install.sh
```

```
● codedeploy-agent.service - LSB: AWS CodeDeploy Host Agent
     Loaded: loaded (/etc/init.d/codedeploy-agent; generated)
     Active: active (running) since Thu 2024-07-18 09:58:00 UTC; 1s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 2511 ExecStart=/etc/init.d/codedeploy-agent start (code=exited, status=0/SUCCESS)
      Tasks: 3 (limit: 1120)
     Memory: 57.0M
        CPU: 997ms
     CGroup: /system.slice/codedeploy-agent.service
```

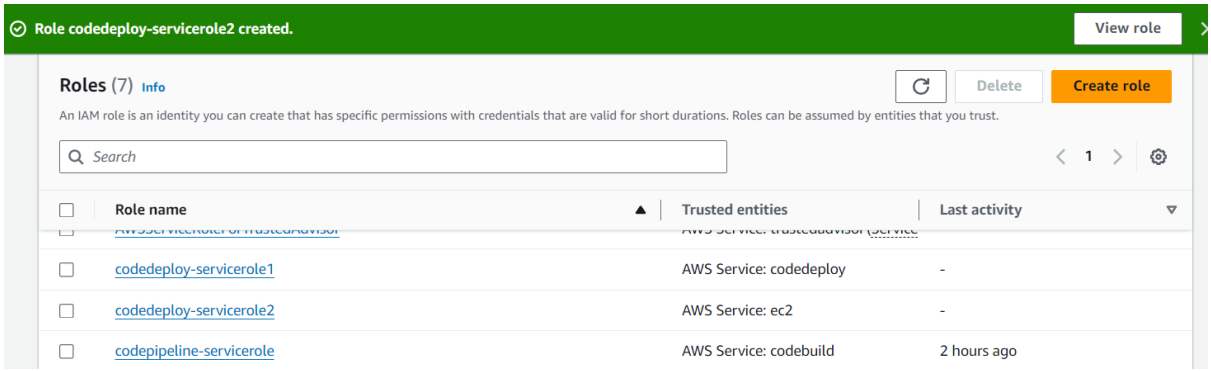
Code deploy service role -2: [for ec2]



Click next. Later on attach permissions .click next

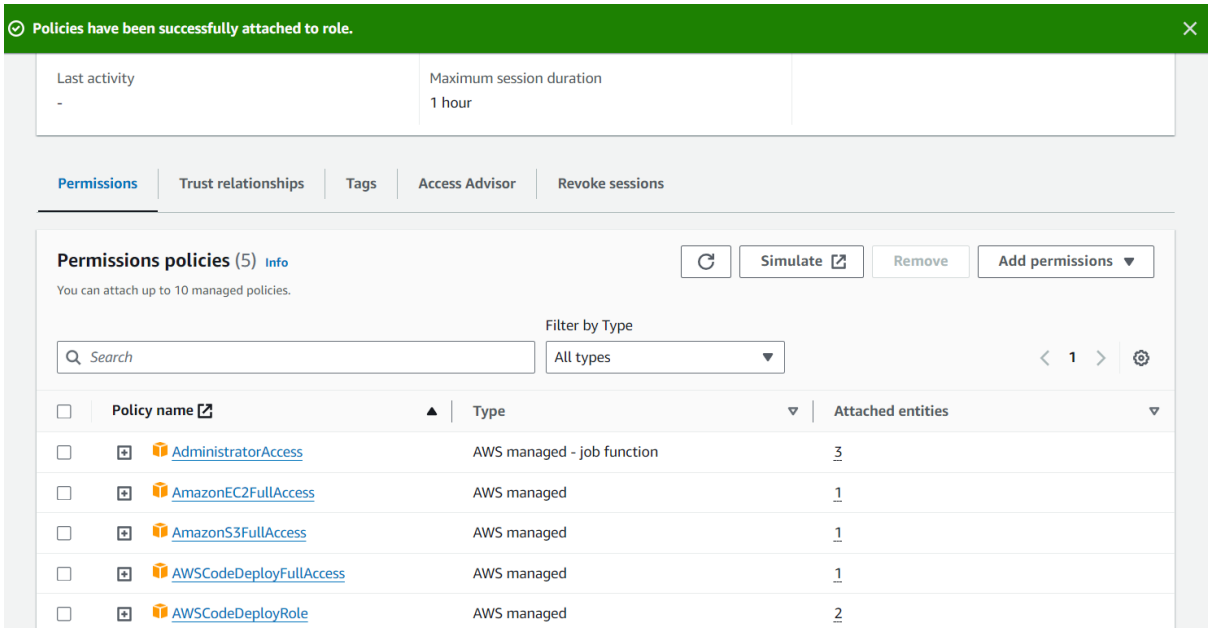


Click create role.role created.

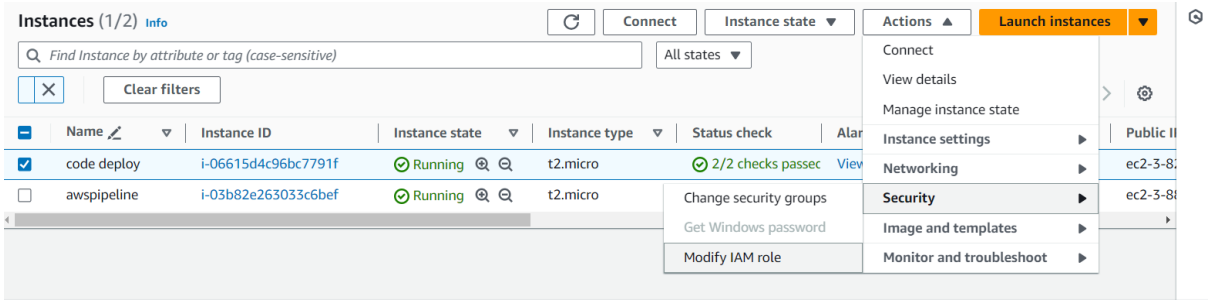


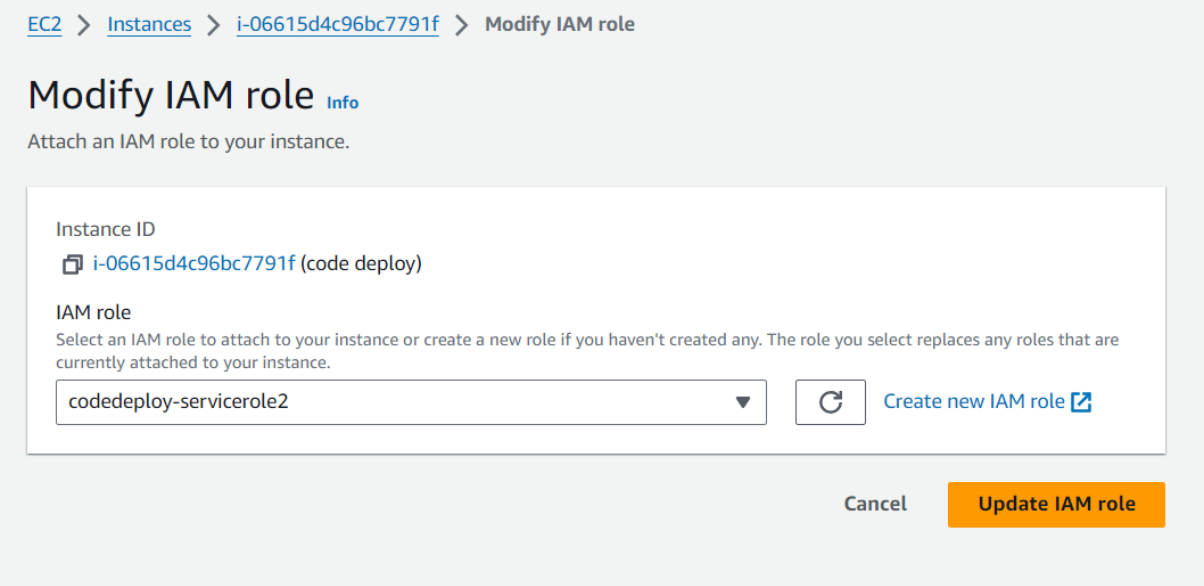
Attach permissions for this role [give full access to ec2, s3, deploy full access, code deploy role]

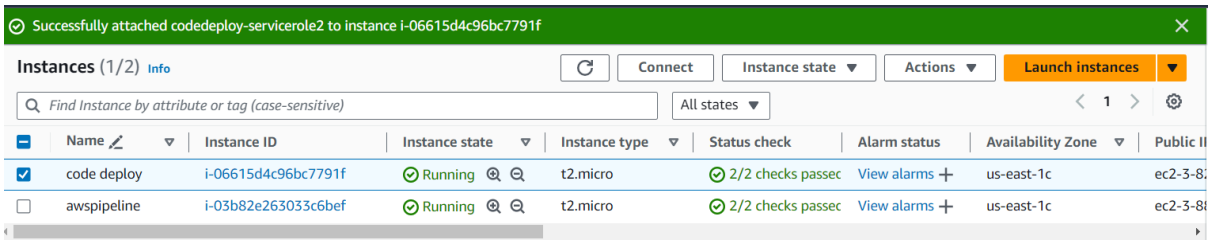
Get into that role click attach policies.



Goto instance give modify iam role







After that restart code deploy agent

```
ubuntu@ip-172-31-80-155:~$ sudo service codedeploy-agent restart
ubuntu@ip-172-31-80-155:~$
```

iii) Appspec.yml

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html
hooks:
  AfterInstall:
    - location: install_nginx.sh
      timeout: 300
      runas: root
  ApplicationStart:
    - location: start_nginx.sh
      timeout: 300
      runas: root
```

- **install_nginx.sh**

```
#!/bin/bash

sudo apt-get update
sudo apt-get install nginx -y
```

- **Start_nginx.sh:**

```
#!/bin/bash
sudo service nginx start
```

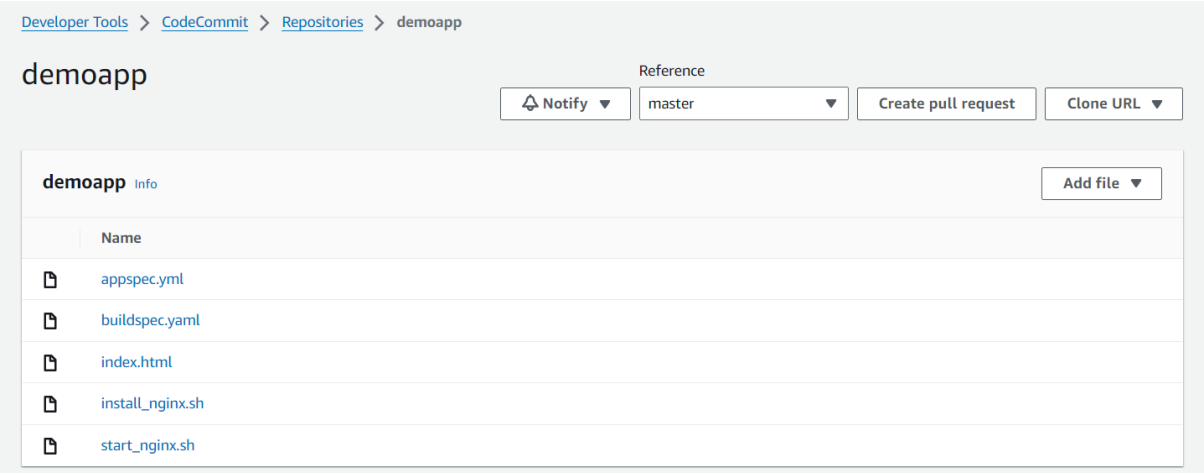
```
[ec2-user@ip-172-31-94-139 ~]$ vi install_nginx.sh
[ec2-user@ip-172-31-94-139 ~]$ vi start_nginx.sh
```

```
[ec2-user@ip-172-31-94-139 ~]$ git add appspec.yml
[ec2-user@ip-172-31-94-139 ~]$ git add install_nginx.sh
[ec2-user@ip-172-31-94-139 ~]$ git add start_nginx.sh
```

```
[ec2-user@ip-172-31-94-139 ~]$ git commit -m "add all files"
[master 12bf0b8] add all files
 Committer: EC2 Default User <ec2-user@ip-172-31-94-139.ec2.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

 3 files changed, 21 insertions(+)
 create mode 100644 appspec.yml
 create mode 100644 install_nginx.sh
 create mode 100644 start_nginx.sh
[ec2-user@ip-172-31-94-139 ~]$ git push
Username for 'https://git-codecommit.us-east-1.amazonaws.com': app-user-at-381492100082
Password for 'https://app-user-at-381492100082@git-codecommit.us-east-1.amazonaws.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Compressing objects: 100% (4/4), done.
```

```
[ec2-user@ip-172-31-94-139 ~]$ git push
Username for 'https://git-codecommit.us-east-1.amazonaws.com': app-user-at-381492100082
Password for 'https://app-user-at-381492100082@git-codecommit.us-east-1.amazonaws.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 641 bytes | 641.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/demoapp
   5765ef5..12bf0b8  master -> master
[ec2-user@ip-172-31-94-139 ~]$
```

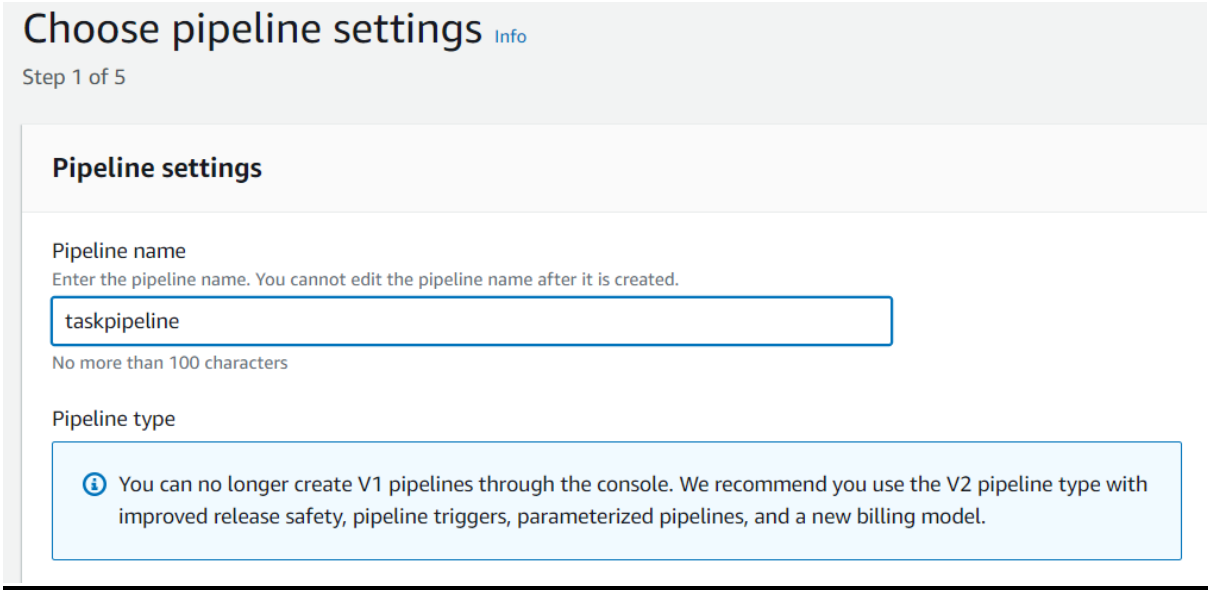
All files written in amazon linux instance



Open the port no: 80 for codedeploy ec2 instance

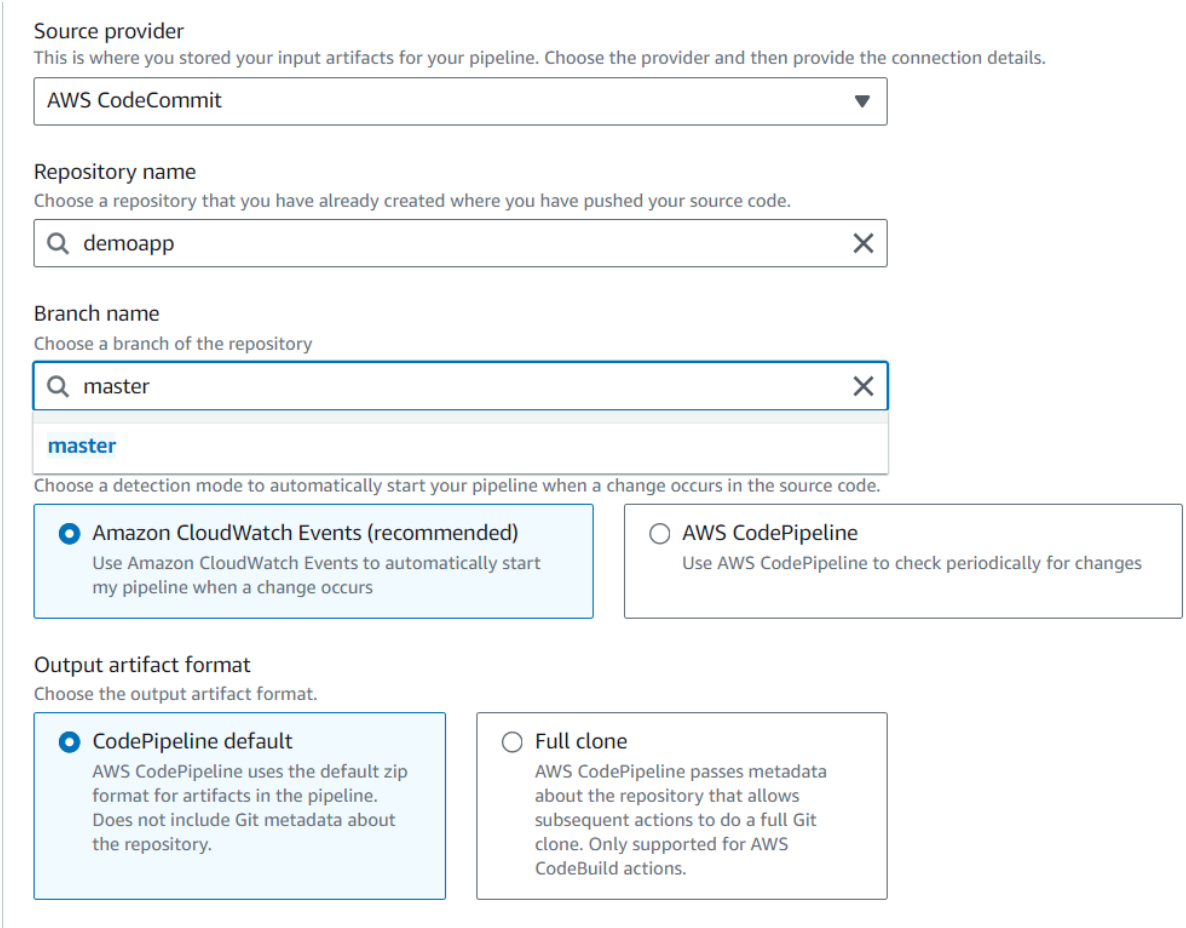
4. Code pipeline:

Click create pipeline



Remaining are same .click next. pipeliene role also assigned automatically

Select aws code commit.All names comes under [already we created]



Step 3 of 5

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

US East (N. Virginia)

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.

awscodepipeline

or

Create project

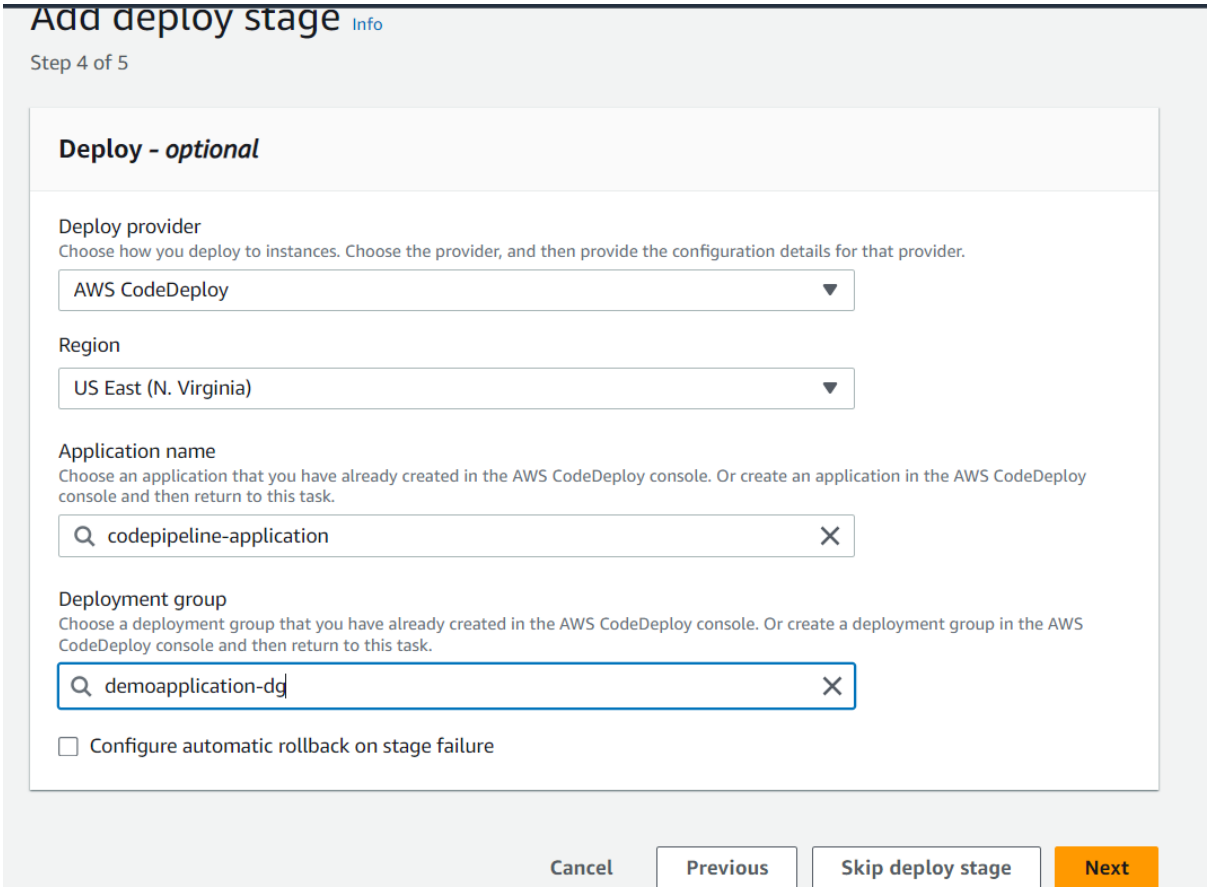
Environment variables - *optional*

Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. Learn more

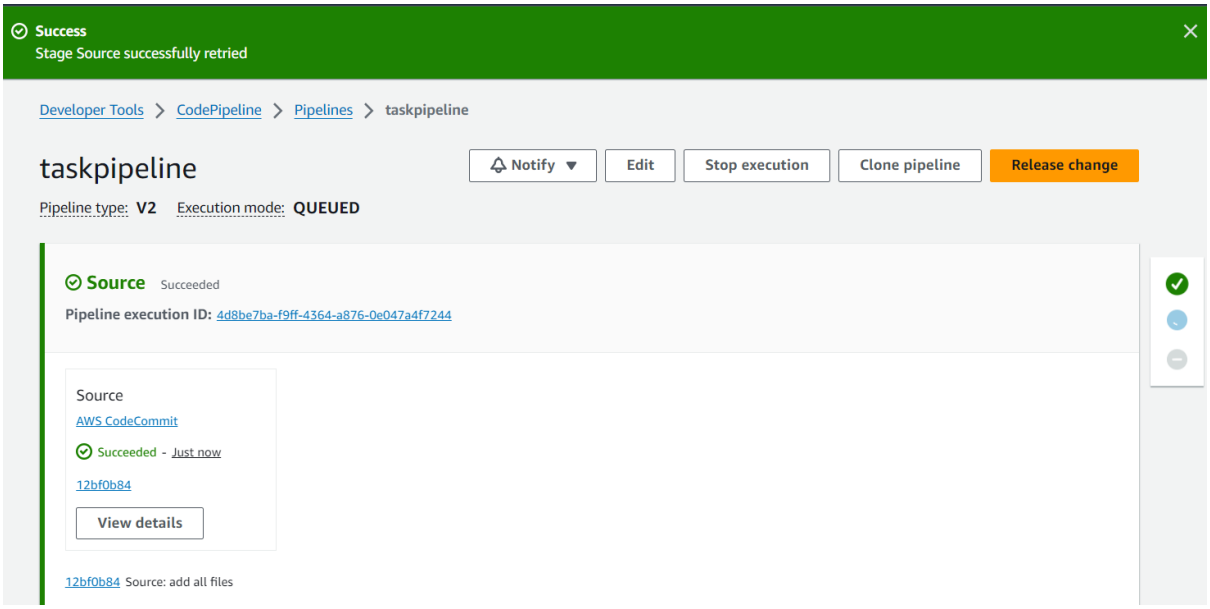
Add environment variable

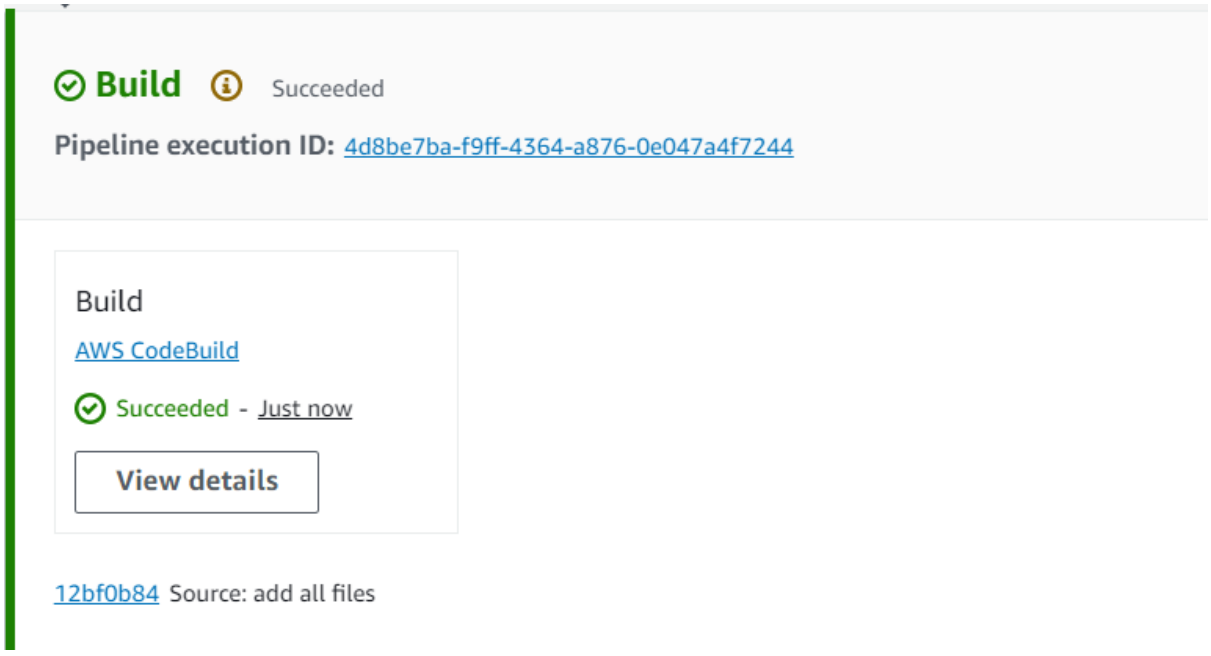
Build type

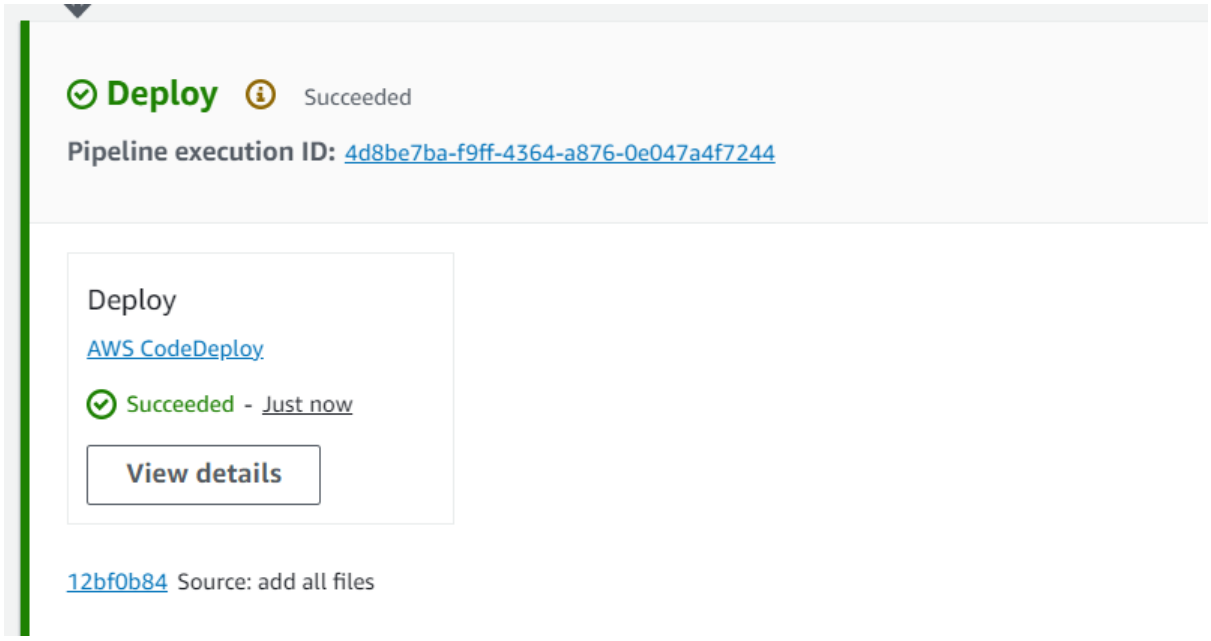
- Single build
  Triggers a single build.
- Batch build
  Triggers multiple builds as a single execution.

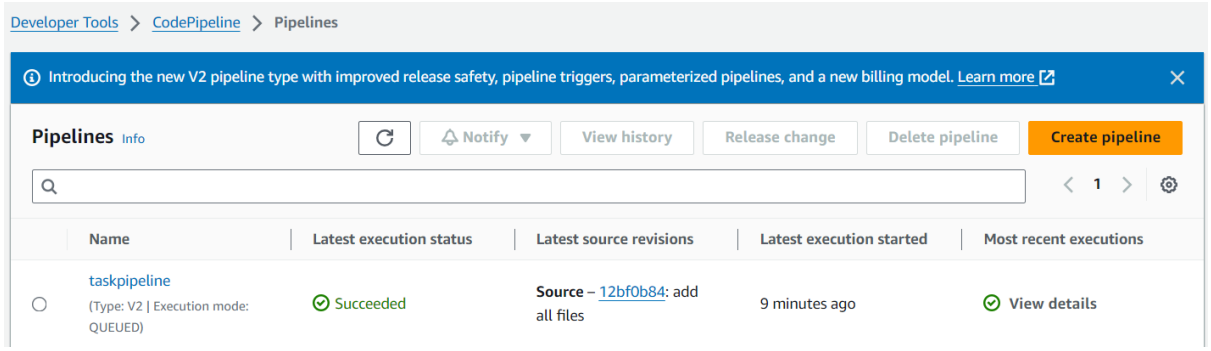


Click next .click create pipeline



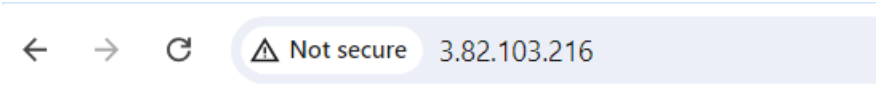

Build
Succeeded
Pipeline execution ID: 4d8be7ba-f9ff-4364-a876-0e047a4f7244
Build
AWS CodeBuild
Succeeded - Just now
View details
12bf0b84 Source: add all files


Deploy
Succeeded
Pipeline execution ID: 4d8be7ba-f9ff-4364-a876-0e047a4f7244
Deploy
AWS CodeDeploy
Succeeded - Just now
View details
12bf0b84 Source: add all files


Developer Tools > CodePipeline > Pipelines
Introducing the new V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model. Learn more
Pipelines Info
Notify
View history
Release change
Delete pipeline
Create pipeline
Name
Latest execution status
Latest source revisions
Latest execution started
Most recent executions
taskpipeline
(Type: V2 | Execution mode: QUEUED)
Succeeded
Source – 12bf0b84: add all files
9 minutes ago
View details

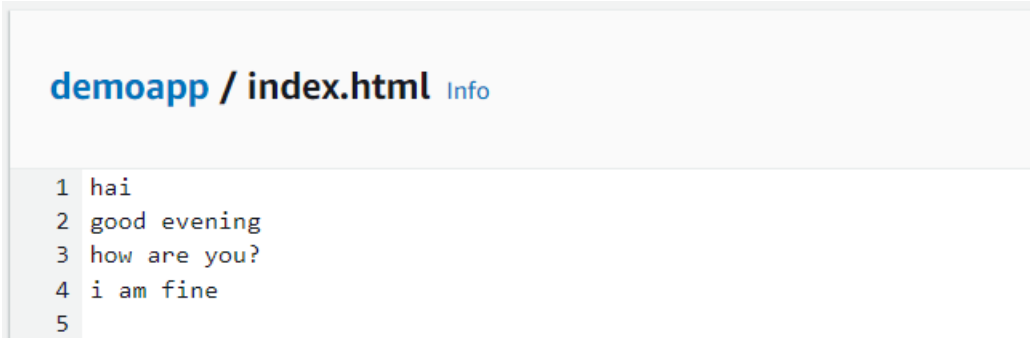
Goto code deploy ec2.open the port no:80..copy the address run in browser.

This is the content I gave in index.html

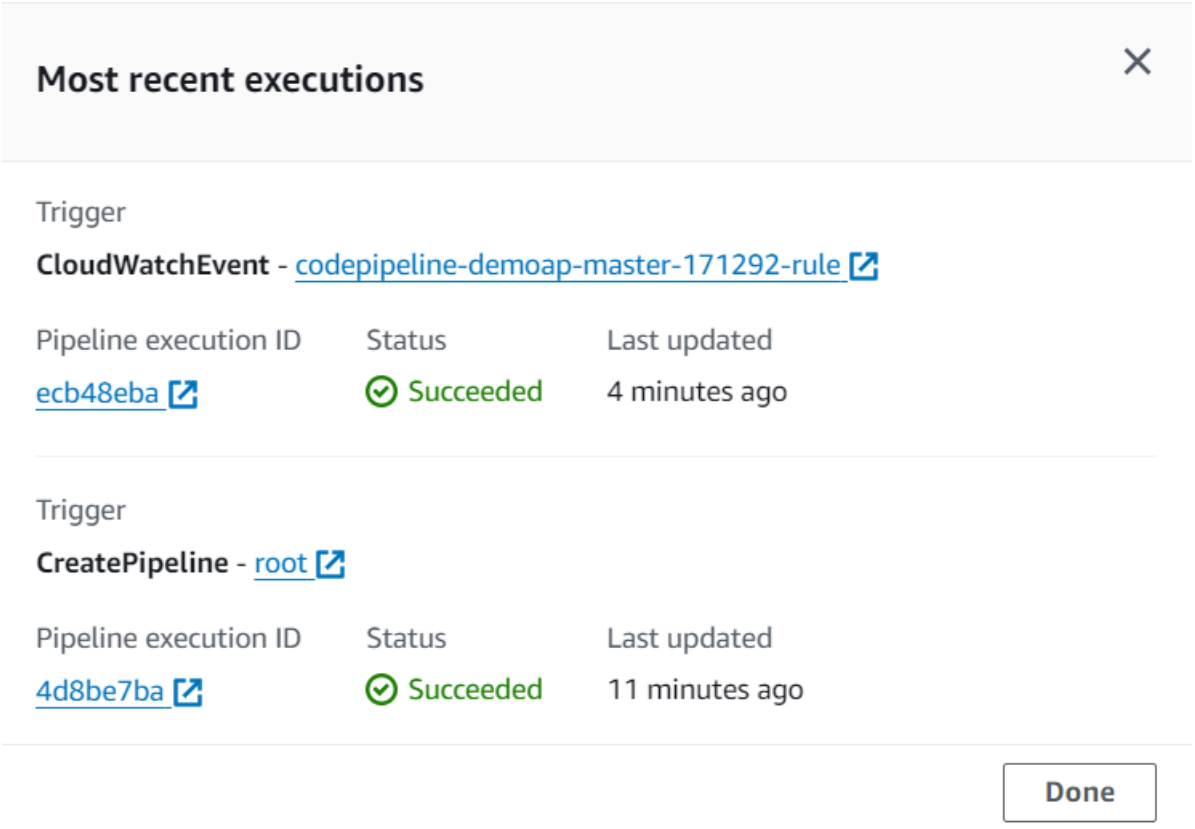


hai good evening how are you?

I gave a change in index.html file.



That chance automatically build



hai good evening how are you? i am fine