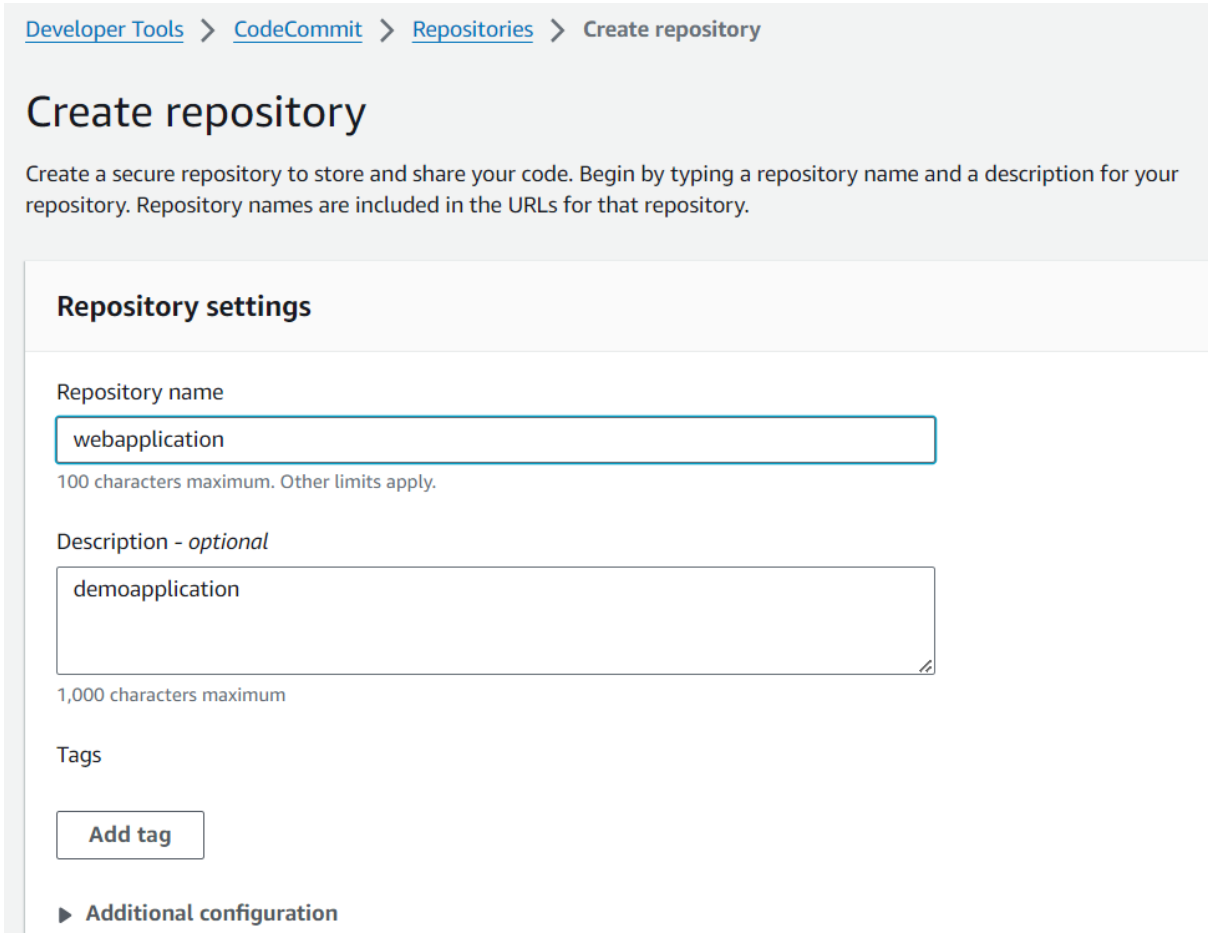
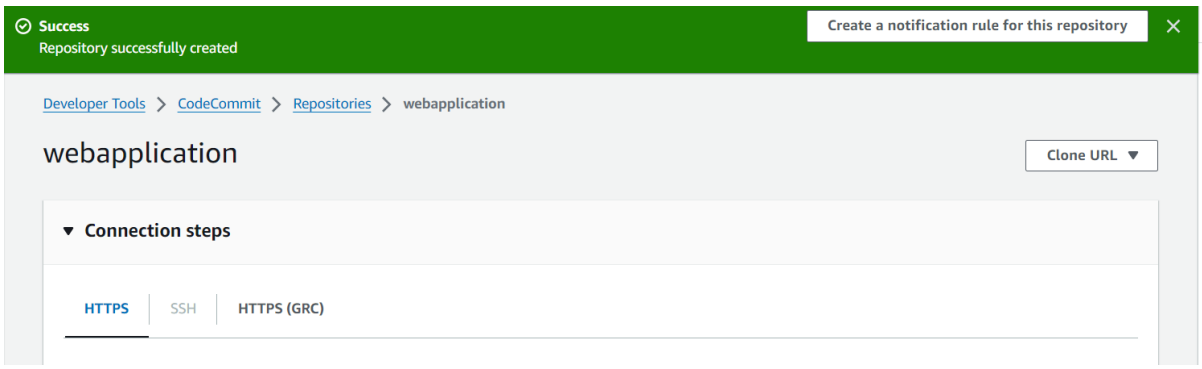
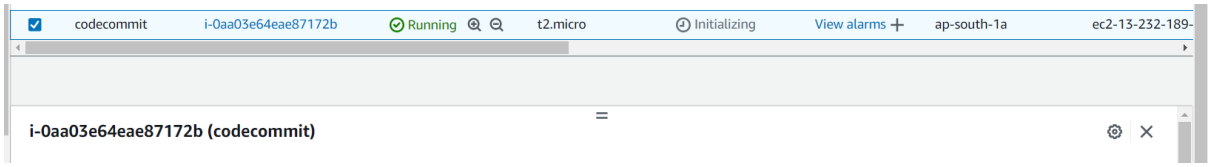
Task 16: Deploy a simple Nginx application using AWS code commit and deploy & access via browser

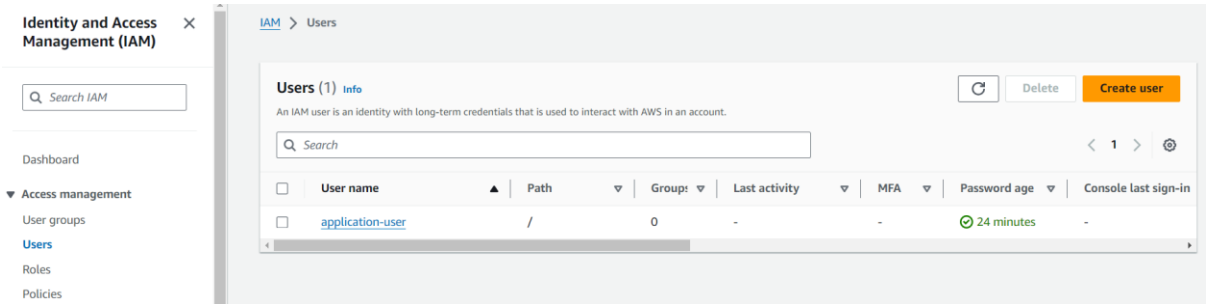
1. **<u>Create code commit repo:</u>**







IAM user:



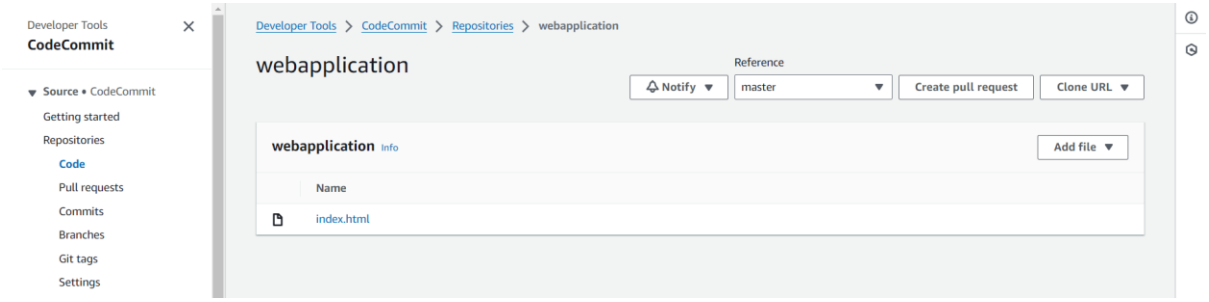
Install git, clone repo,add,push in code commit:

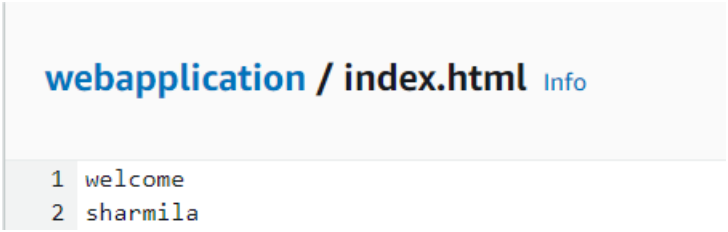
1. ## Index.html file:

```
[ec2-user@ip-172-31-39-139 ~]$ sudo yum install git -y
Last metadata expiration check: 0:00:17 ago on Tue Jul 16 10:34:56 2024.
Dependencies resolved.
================================================================================
 Package                 Architecture        Version                 Repository      Size
================================================================================
Installing:
 git                     x86_64              2.40.1-1.amzn2023.0.3   amazonlinux     54 k
Installing dependencies:
 git-core                x86_64              2.40.1-1.amzn2023.0.3   amazonlinux    4.3 M
 git-core-doc            noarch              2.40.1-1.amzn2023.0.3   amazonlinux    2.6 M
 perl-Error              noarch              1:0.17029-5.amzn2023.0.2 amazonlinux    41 k
 perl-File-Find          noarch              1.37-477.amzn2023.0.6   amazonlinux    26 k
 perl-Git                noarch              2.40.1-1.amzn2023.0.3   amazonlinux    42 k
 perl-TermReadKey        x86_64              2.38-9.amzn2023.0.2     amazonlinux    36 k
 perl-lib                x86_64              0.65-477.amzn2023.0.6   amazonlinux    15 k
```

```
[ec2-user@ip-172-31-39-139 ~]$ git clone https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/webapplication
Cloning into 'webapplication'...
Username for 'https://git-codecommit.ap-south-1.amazonaws.com': application-user-at-891377228959
Password for 'https://application-user-at-891377228959@git-codecommit.ap-south-1.amazonaws.com':
warning: You appear to have cloned an empty repository.
[ec2-user@ip-172-31-39-139 ~]$ ls
webapplication
[ec2-user@ip-172-31-39-139 ~]$ cd webapplication/
[ec2-user@ip-172-31-39-139 webapplication]$ vi index.html
[ec2-user@ip-172-31-39-139 webapplication]$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
        add
[ec2-user@ip-172-31-39-139 webapplication]$ git add .
[ec2-user@ip-172-31-39-139 webapplication]$ git commit -m "index file added"
```

```
[ec2-user@ip-172-31-39-139 webapplication]$ git push
Username for 'https://git-codecommit.ap-south-1.amazonaws.com': application-user-at-891377228959
Password for 'https://application-user-at-891377228959@git-codecommit.ap-south-1.amazonaws.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 264 bytes | 264.00 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/webapplication
 * [new branch]      master -> master
[ec2-user@ip-172-31-39-139 webapplication]$
```

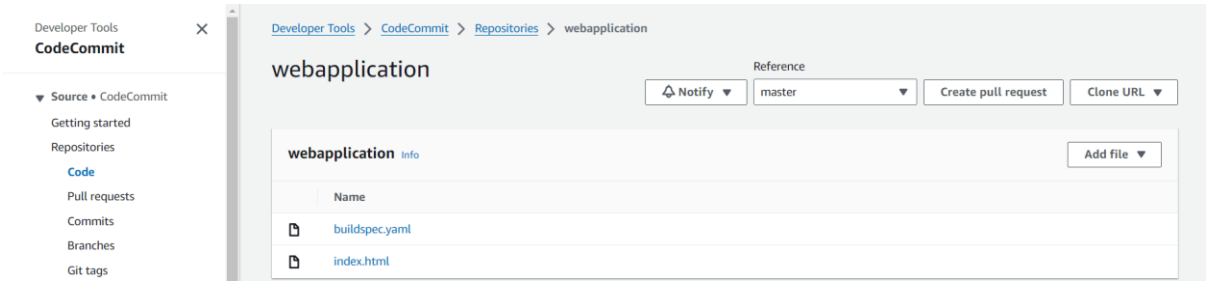


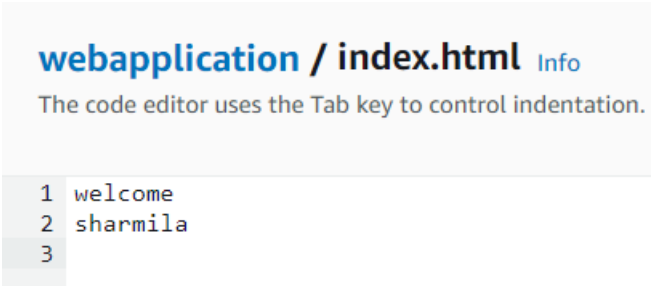


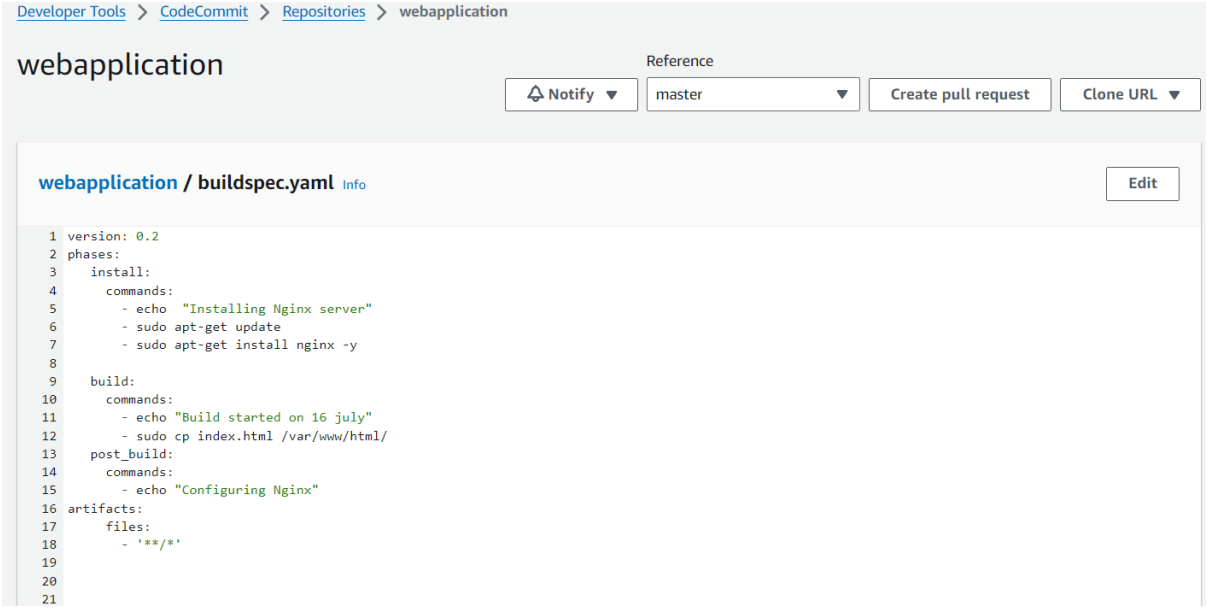
2. Buildspec.yaml file

```
[ec2-user@ip-172-31-39-139 webapplication]$ vi buildspec.yaml
[ec2-user@ip-172-31-39-139 webapplication]$ git add .
[ec2-user@ip-172-31-39-139 webapplication]$ git commit -m "buildspec file added"
[master e196f94] buildspec file added
 Committer: EC2 Default User <ec2-user@ip-172-31-39-139.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

 2 files changed, 23 insertions(+)
 create mode 100644 buildsepc.Yaml
 create mode 100644 buildspec.yaml
[ec2-user@ip-172-31-39-139 webapplication]$ git push
Username for 'https://git-codecommit.ap-south-1.amazonaws.com': application-user-at-891377228959
Password for 'https://application-user-at-891377228959@git-codecommit.ap-south-1.amazonaws.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (3/3), done.
```

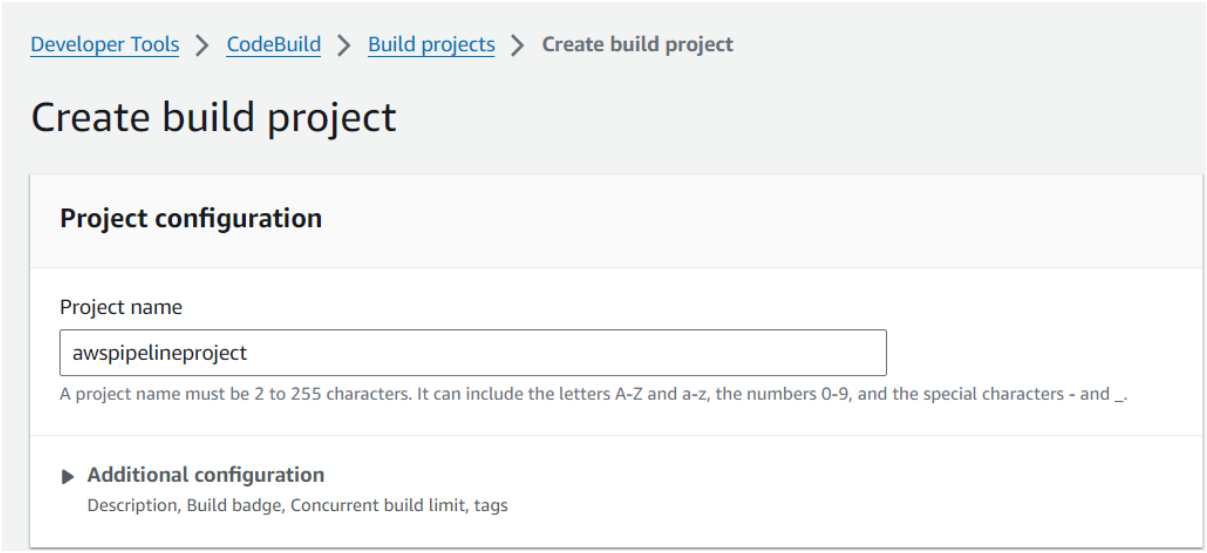
```
[ec2-user@ip-172-31-39-139 webapplication]$ git push
Username for 'https://git-codecommit.ap-south-1.amazonaws.com': application-user-at-891377228959
Password for 'https://application-user-at-891377228959@git-codecommit.ap-south-1.amazonaws.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 585 bytes | 585.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/webapplication
   c5f8e54..e196f94  master -> master
[ec2-user@ip-172-31-39-139 webapplication]$
```






Developer Tools > CodeCommit > Repositories > webapplication
webapplication
Reference
Notify
master
Create pull request
Clone URL
webapplication / buildspec.yaml Info
Edit
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 16 july"
- sudo cp index.html /var/www/html/
post_build:
commands:
- echo "Configuring Nginx"
artifacts:
files:
- '**/*'

3. Code build:


Developer Tools > CodeBuild > Build projects > Create build project
Create build project
Project configuration
Project name
awspipelineproject
A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.
Additional configuration
Description, Build badge, Concurrent build limit, tags

Source

Add source

Source 1 - Primary

Source provider

AWS CodeCommit

Repository

webapplication

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

refs/heads/master

`35810f3f` delete the file

Compute

- (•) EC2
  Optimized for flexibility during action runs
- ( ) Lambda
  Optimized for speed and minimizes the start up time of workflow actions

Operating system

Ubuntu

Runtime(s)

Standard

Image

aws/codebuild/standard:7.0

Image version

Always use the latest image for this runtime version

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

webapplication-user

Amazon S3 > Buckets > sharmi-pipelinebucket > Create folder

Create folder Info

Use folders to group objects in buckets. When you create a folder, S3 creates an object using the name that you specify followed by a slash (/). This object then appears as folder on the console. Learn more

Your bucket policy might block folder creation
If your bucket policy prevents uploading objects without specific tags, metadata, or access control list (ACL) grantees, you will not be able to create a folder using this configuration. Instead, you can use the upload configuration to upload an empty folder and specify the appropriate settings.

Folder

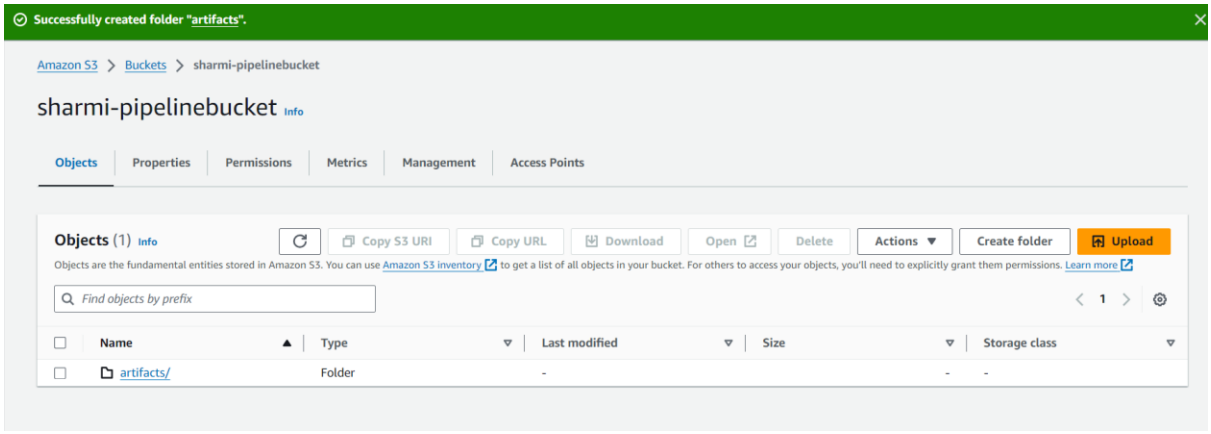
Folder name

artifacts /

Folder names can't contain "/". See rules for naming

Server-side encryption Info

Server-side encryption protects data at rest.



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

sharmi-pipelinebucket

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

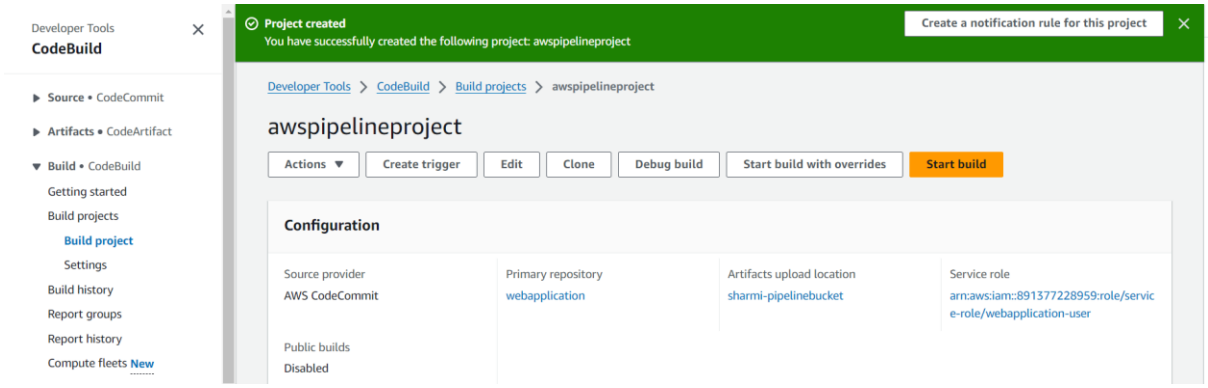
Enable semantic versioning

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.



Developer Tools > CodeCommit > Repositories > webapplication

webapplication

Reference

Notify | master | Create pull request | Clone URL

webapplication / buildspec.yaml Info

Edit

```
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 16 july"
       - sudo cp index.html /var/www/html/
   post_build:
     commands:
       - echo "Configuring Nginx"
artifacts:
     files:
       - '**/*'
```

Developer Tools > CodeBuild > Build projects

Build projects Info

Actions | Create trigger | View details | Start build | Create project

Your projects

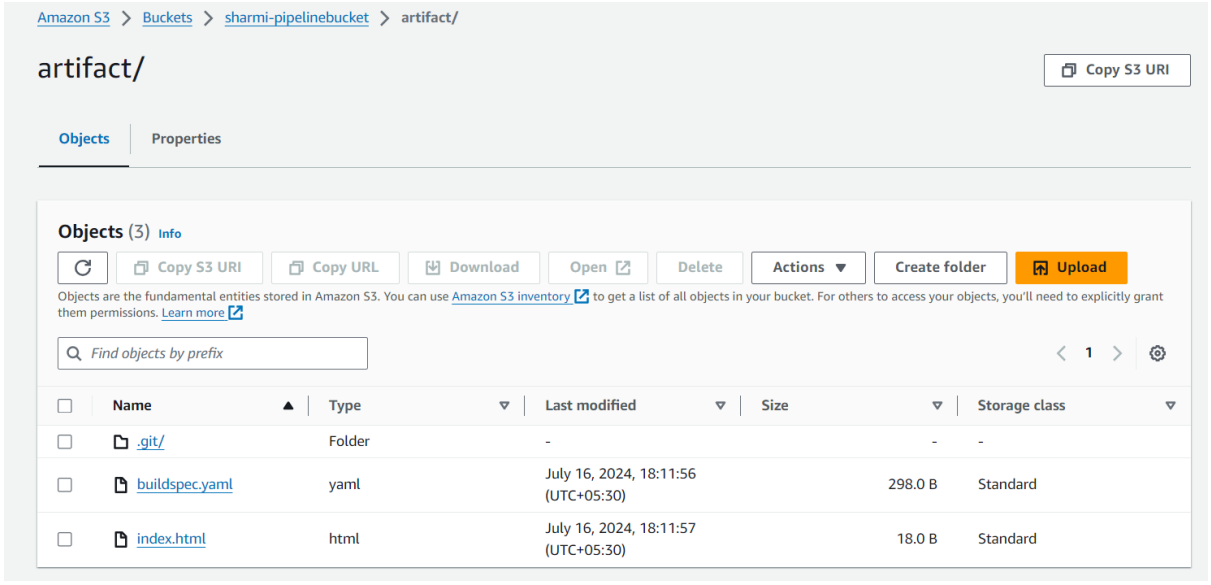
Name	Source provider	Repository	Latest build status	Description	Last Modified
pipelineaws	AWS CodeCommit	webapplication	Succeeded	-	36 minutes ago

Build history | Batch history | Project details | Build triggers | Metrics

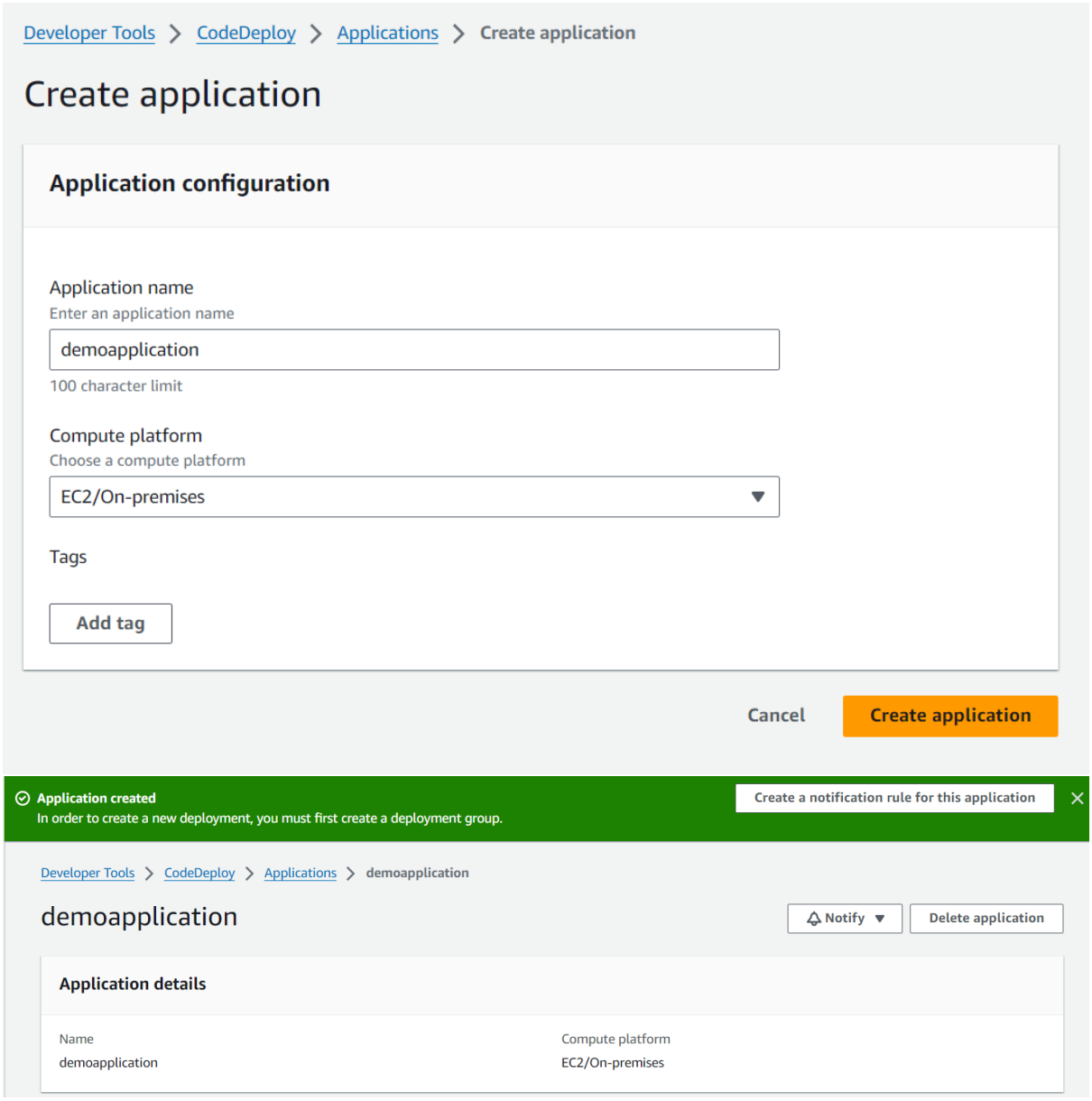
Build history

Stop build | View artifacts | View logs | Delete builds | Retry build

Build run	Status	Build number	Source version	Submitter	Duration	Completed
pipelineaws:730042a9-410d-4ba9-a0f5-c511fee0a87c	Succeeded	1	refs/heads/master	root	33 seconds	36 minutes ago


Amazon S3 > Buckets > sharmi-pipelinebucket > artifact/
artifact/
Copy S3 URI
Objects
Properties
Objects (3) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
1
Name
Type
Last modified
Size
Storage class
.git/
Folder
-
-
-
buildspec.yaml
yaml
July 16, 2024, 18:11:56 (UTC+05:30)
298.0 B
Standard
index.html
html
July 16, 2024, 18:11:57 (UTC+05:30)
18.0 B
Standard

4. Code deploy:



Create service role for code deploy:

Trusted entity type

- **AWS service** — Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- **AWS account** — Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- **Web identity** — Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- **SAML 2.0 federation** — Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- **Custom trust policy** — Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

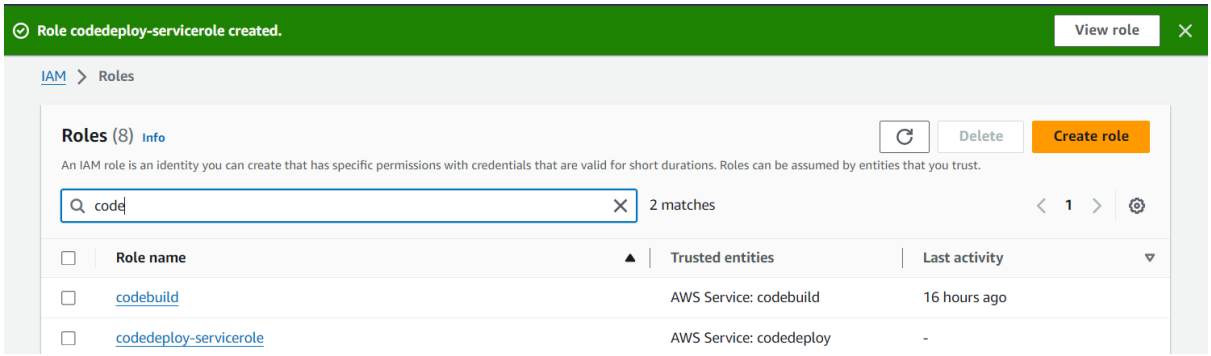
Service or use case

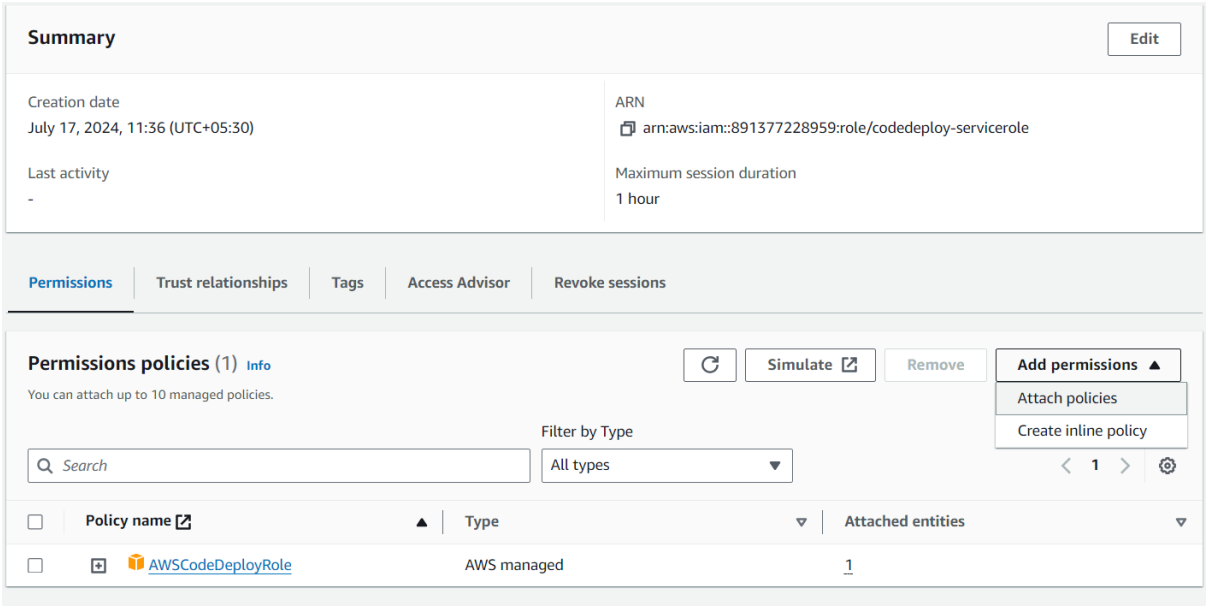
CodeDeploy

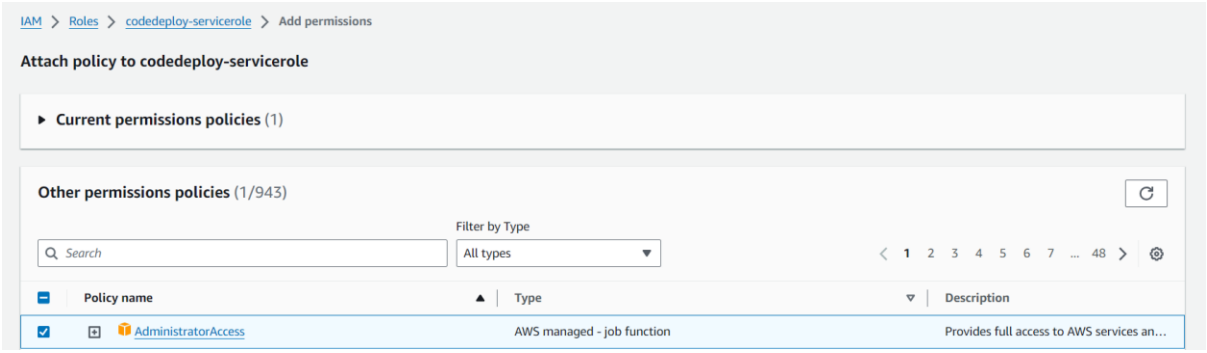
Choose a use case for the specified service.

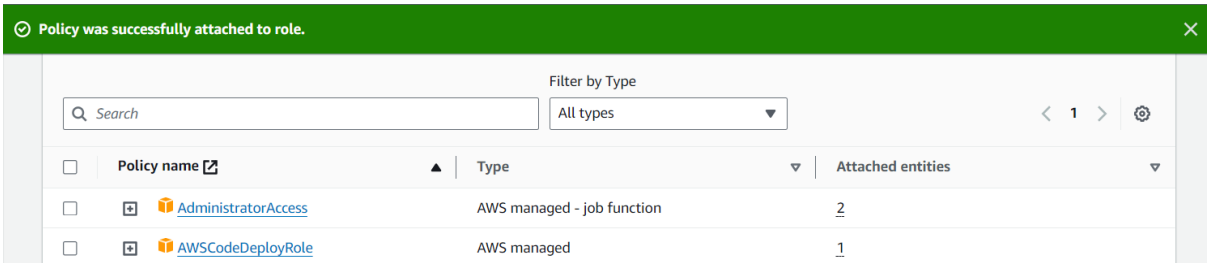
Use case

- **CodeDeploy**
  Allows CodeDeploy to call AWS services such as Auto Scaling on your behalf.

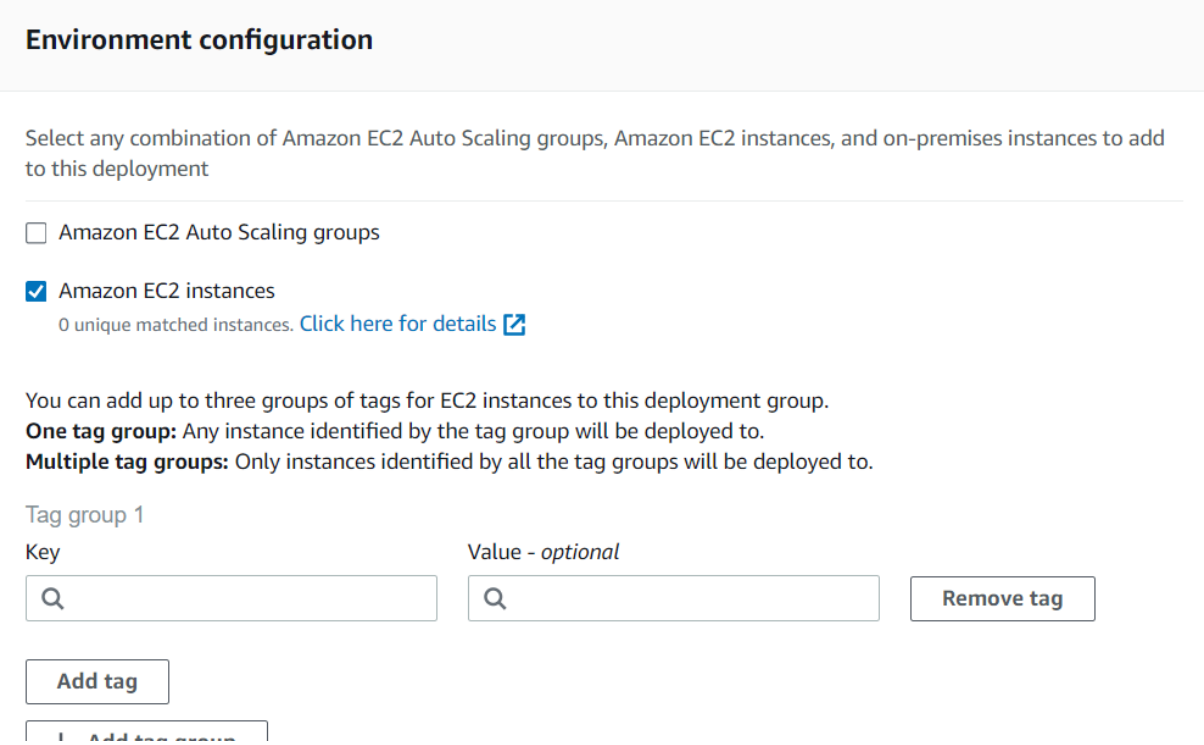



Summary
Edit
Creation date
July 17, 2024, 11:36 (UTC+05:30)
ARN
arn:aws:iam::891377228959:role/codedeploy-servicerole
Last activity
-
Maximum session duration
1 hour
Permissions
Trust relationships
Tags
Access Advisor
Revoke sessions
Permissions policies (1) Info
You can attach up to 10 managed policies.
Simulate
Remove
Add permissions
Attach policies
Create inline policy
Filter by Type
Search
All types
Policy name
Type
Attached entities
AWSCodeDeployRole
AWS managed
1

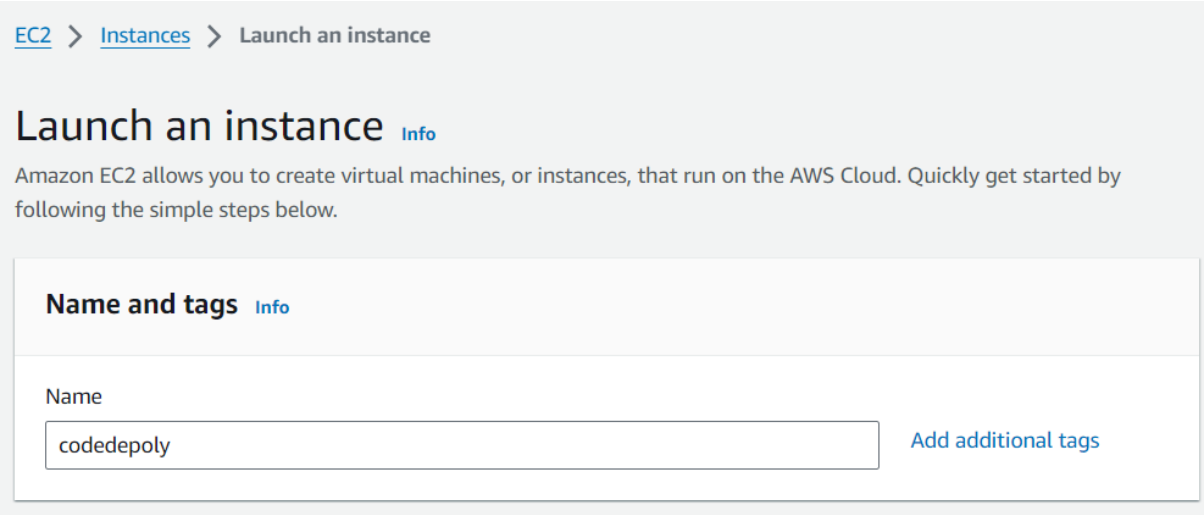

IAM
Roles
codedeploy-servicerole
Add permissions
Attach policy to codedeploy-servicerole
Current permissions policies (1)
Other permissions policies (1/943)
Filter by Type
Search
All types
Policy name
Type
Description
AdministratorAccess
AWS managed - job function
Provides full access to AWS services an...


Policy was successfully attached to role.
Filter by Type
Search
All types
Policy name
Type
Attached entities
AdministratorAccess
AWS managed - job function
2
AWSCodeDeployRole
AWS managed
1

In this place we have to give our code deploy instance.so created new instance



Create new instance for code deploy:



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-0c2af51e265bd5e0e (64-bit (x86)) / ami-0c938b21c7e598cd0 (64-bit (Arm))

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

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

Description

Ubuntu Server 22.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Architecture: 64-bit (x86)

AMI ID: ami-0c2af51e265bd5e0e — Verified provider

codedepoly i-04888a19787435f0a Running t2.micro – View alarms ap-south-1a ec2-13-233-17

i-04888a19787435f0a (codedepoly)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary Info

Instance ID
i-04888a19787435f0a (codedepoly)

Public IPv4 address
13.233.178.205 | open address

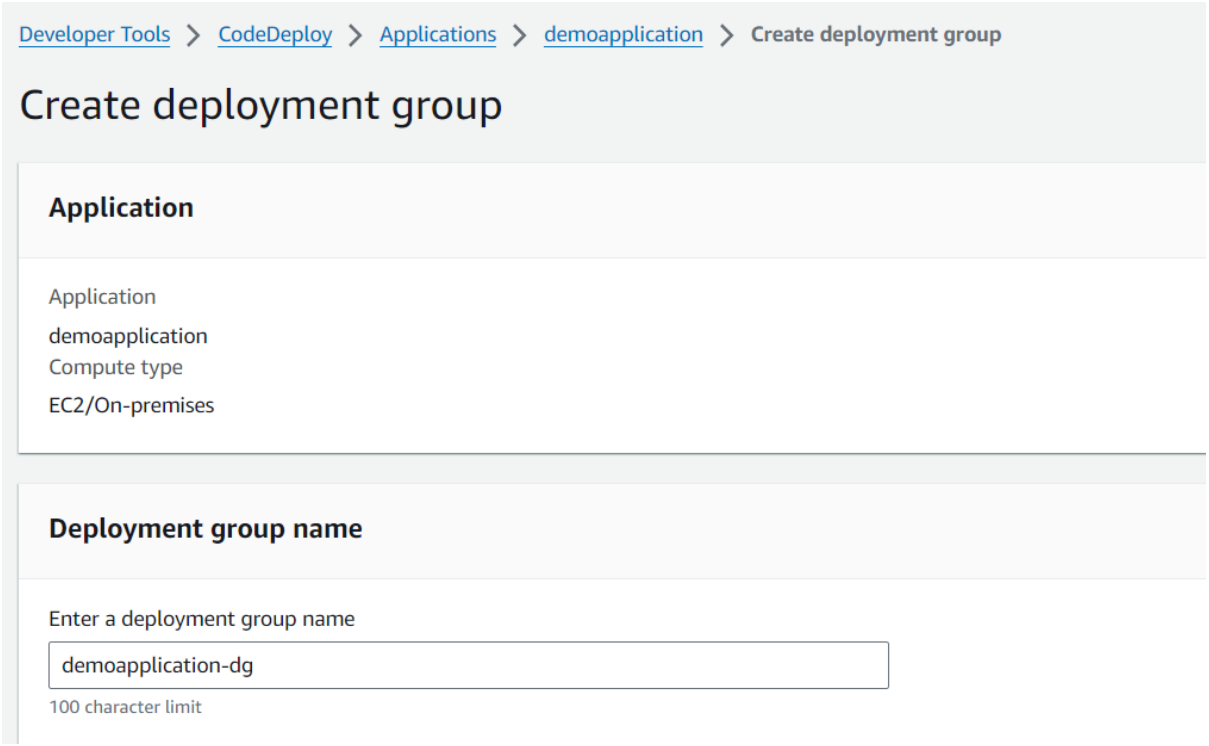
Private IPv4 addresses
172.31.39.248

IPv6 address
–

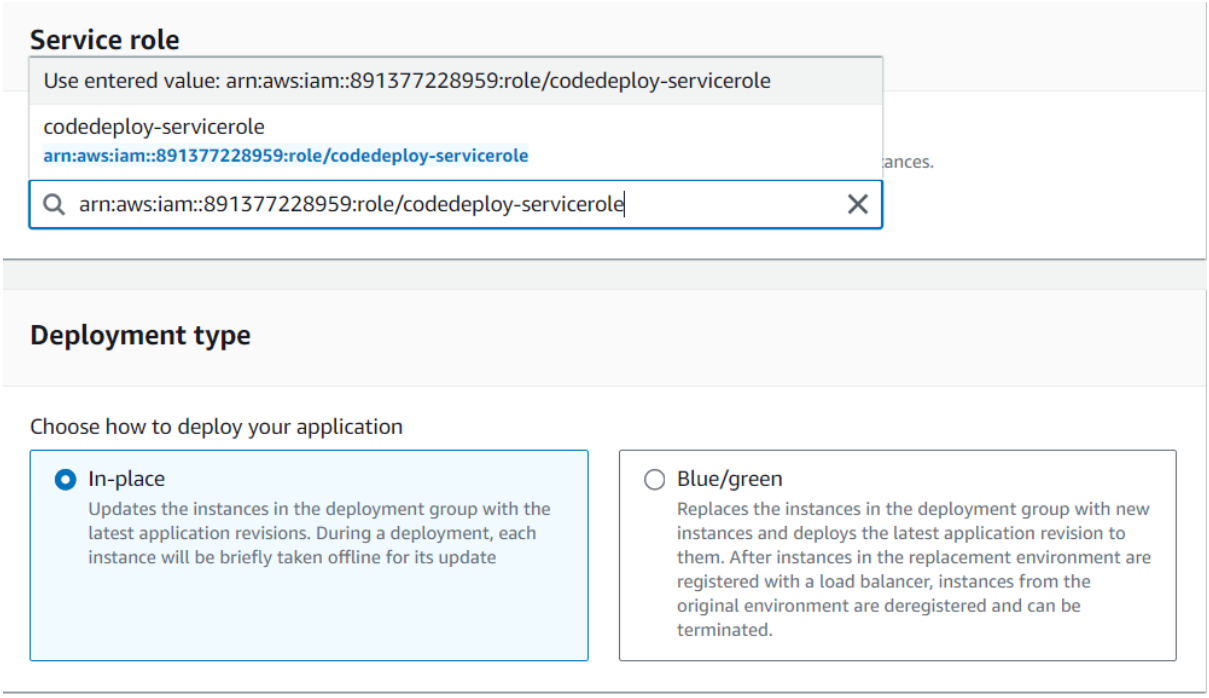
Instance state
Running

Public IPv4 DNS
ec2-13-233-178-205.ap-south-1.compute.amazonaws.com | open address

Create deployment group for my application:



For this we have to create service role for code deploy:



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*	
Name	codedepoly	Remove tag

Add tag

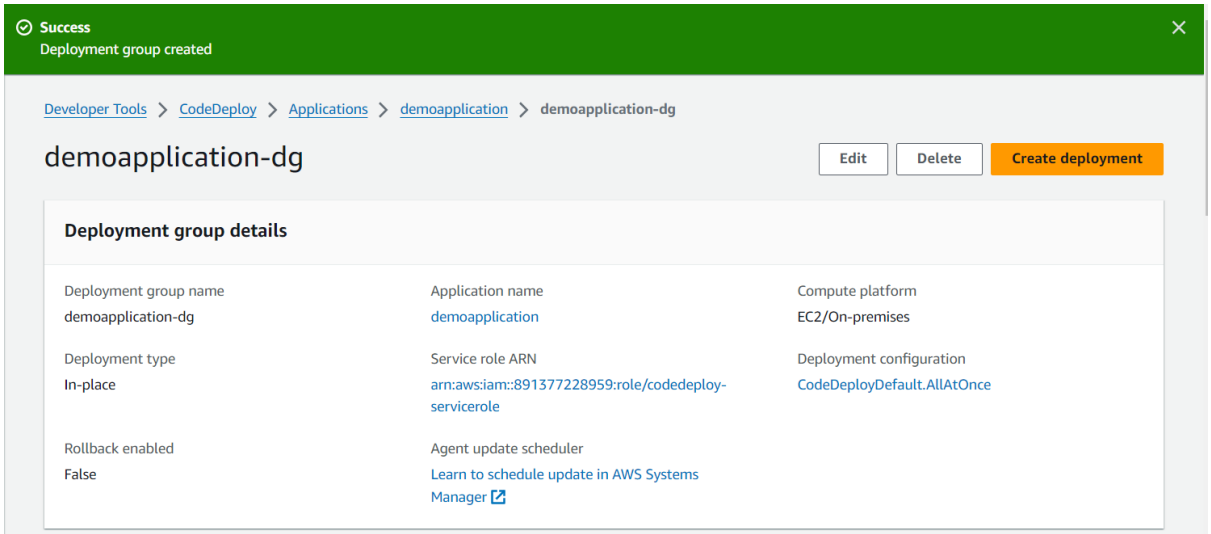
+ Add tag group

☐ On-premises instances

Matching instances

1 unique matched instance. Click here for details

Click create deployment group



Install code deploy agent. For this we write shell script

```
# Download the CodeDeploy agent package
wget https://aws-codedeploy-ap-south-1.s3.ap-south-1.amazonaws.com/releases/codedeploy-agent_1.3.2-1902_all.deb

# Create a directory for unpacking the package
mkdir -p codedeploy-agent_1.3.2-1902_ubuntu22

# Unpack the downloaded package
dpkg-deb -R codedeploy-agent_1.3.2-1902_all.deb codedeploy-agent_1.3.2-1902_ubuntu22

# Modify the control file to change dependencies
sed 's/Depends:.*/Depends: ruby3.0/' -i ./codedeploy-agent_1.3.2-1902_ubuntu22/DEBIAN/control

# Repack the modified package
dpkg-deb -b codedeploy-agent_1.3.2-1902_ubuntu22

# Install the modified package
sudo dpkg -i codedeploy-agent_1.3.2-1902_ubuntu22.deb

# Check if the CodeDeploy agent service is listed
systemctl list-units --type=service | grep codedeploy

# Check the status of the CodeDeploy agent service
sudo service codedeploy-agent status
```

```
  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 Wed 2024-07-17 09:37:57 UTC; 69ms ago
       Docs: man:systemd-sysv-generator(8)
    Process: 4553 ExecStart=/etc/init.d/codedeploy-agent start (code=exited, status=0/SUCCESS)
      Tasks: 2 (limit: 1120)
     Memory: 38.6M
        CPU: 553ms
     CGroup: /system.slice/codedeploy-agent.service
             ├─4559 "codedeploy-agent: master 4559" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
             └─4561 "codedeploy-agent: booting child" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "

Jul 17 09:37:56 ip-172-31-39-248 systemd[1]: Starting LSB: AWS CodeDeploy Host Agent...
Jul 17 09:37:57 ip-172-31-39-248 codedeploy-agent[4553]: Starting codedeploy-agent:
Jul 17 09:37:57 ip-172-31-39-248 systemd[1]: Started LSB: AWS CodeDeploy Host Agent.
```

Create another role for code deploy [give full access to ec2,s3,deployfullaccess,codedeployrole]

Trusted entity type

- (●) **AWS service** — Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- ( ) **AWS account** — Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- ( ) **Web identity** — Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- ( ) **SAML 2.0 federation** — Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- ( ) **Custom trust policy** — Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

EC2

Choose a use case for the specified service.

Use case

(●) **EC2**
Allows EC2 instances to call AWS services on your behalf.

Role codedeploy2-servicerole created. View role

IAM > Roles

Roles (9) Info

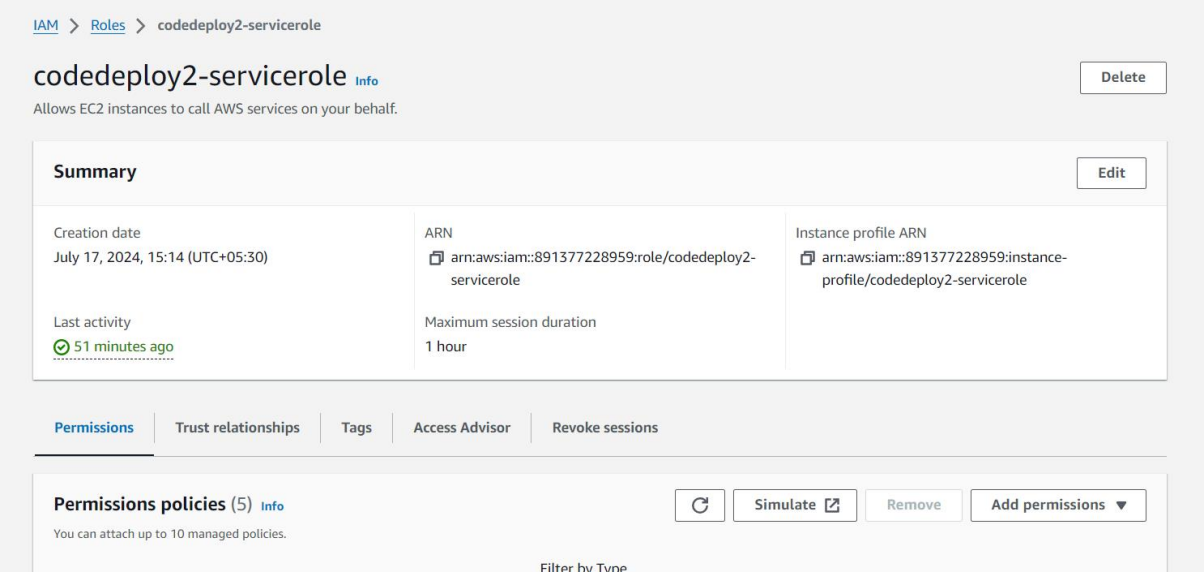
Delete | Create role

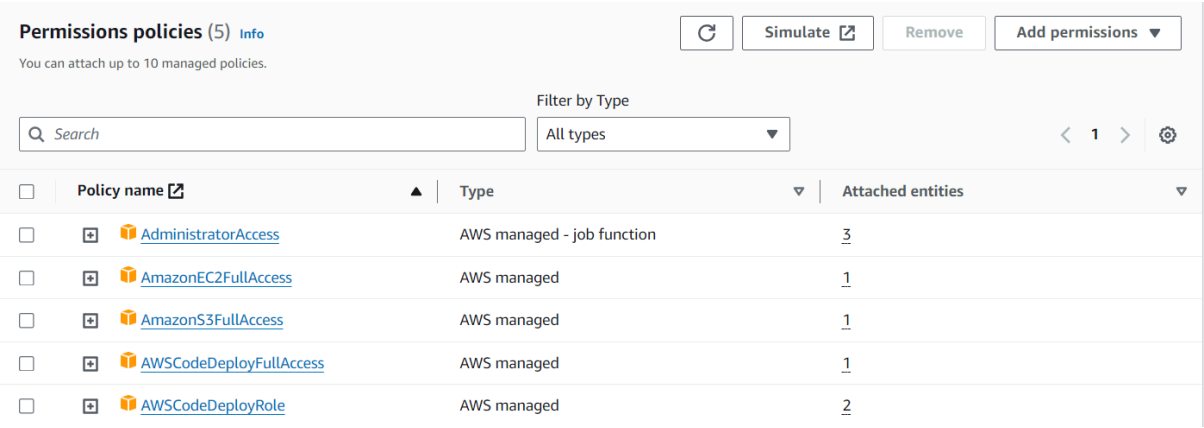
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

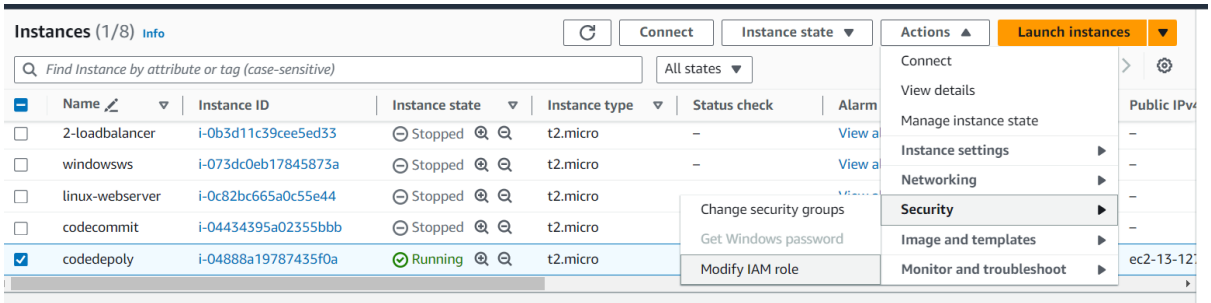
code | 3 matches | 1

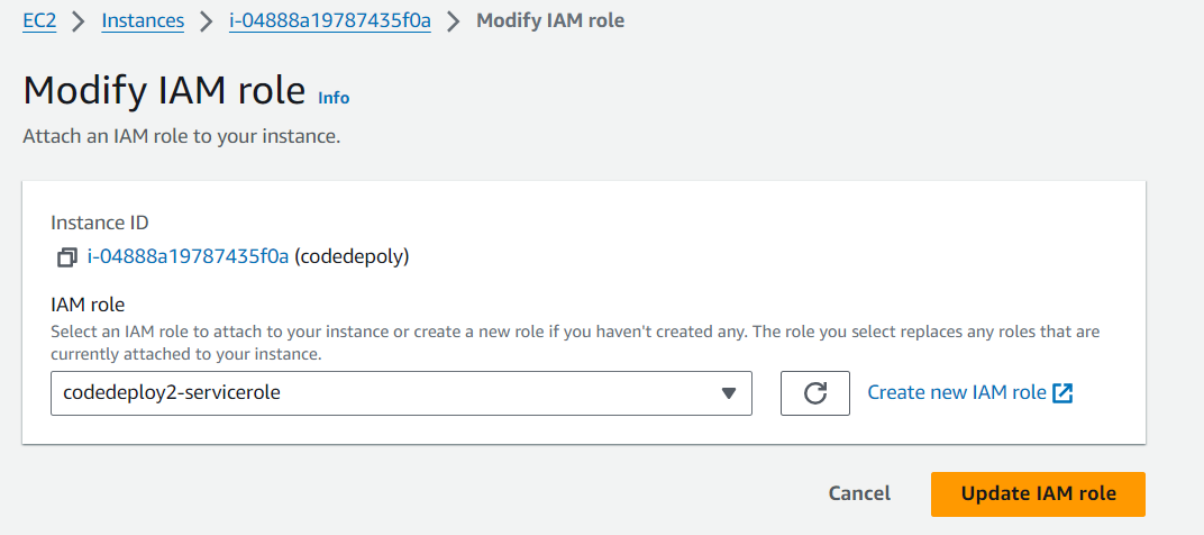
Role name	Trusted entities	Last activity
codebuild	AWS Service: codebuild	20 hours ago
codedeploy-servicerole	AWS Service: codedeploy	-
codedeploy2-servicerole	AWS Service: ec2	-

Add permissions for codedeploy2-servicerole:









Appspec.yml file

webapplication / appspec.yml Info

```yaml
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

webapplication / install_nginx.sh Info

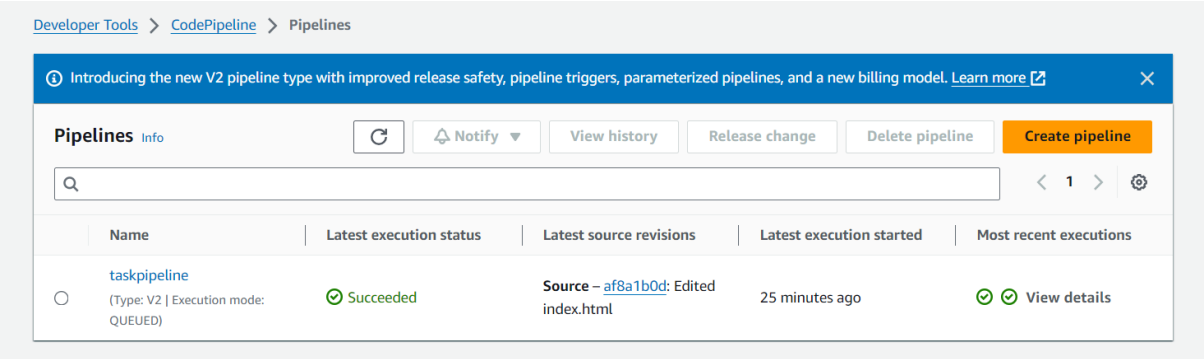
```
#!/bin/bash

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

Start_nginx.sh:

webapplication / start_nginx.sh Info

```
1 #!/bin/bash
2
3
4 sudo service nginx start
5
```

5. Code pipeline:



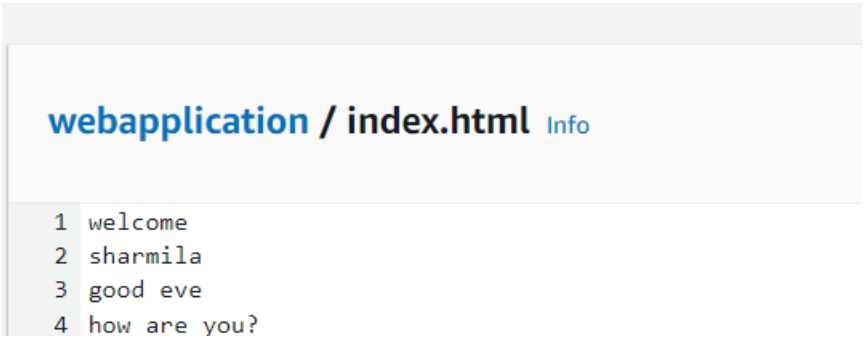
webapplication / index.html Info

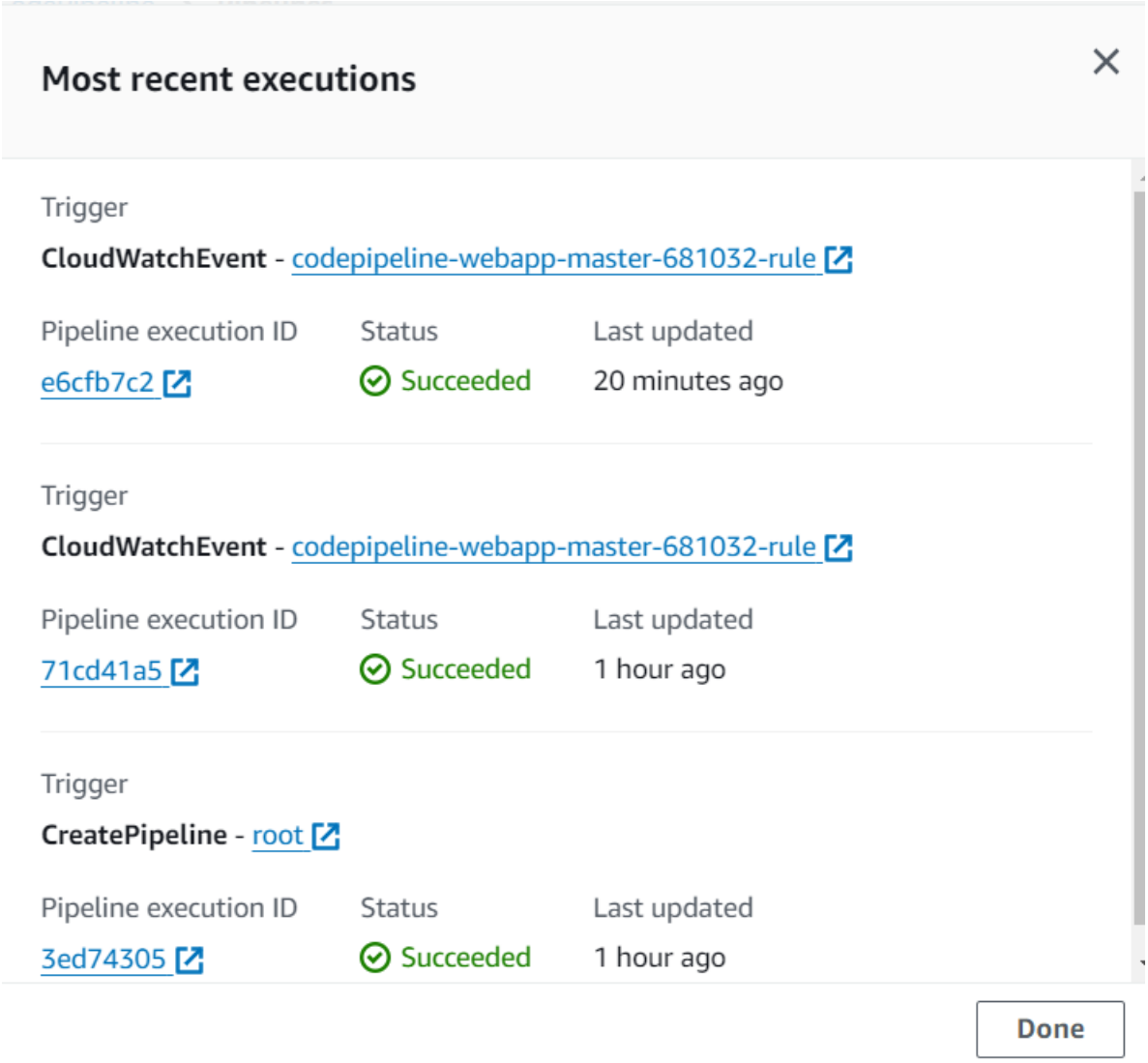
The code editor uses the Tab key to control indentation.

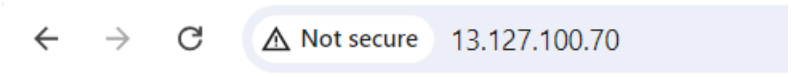
```
1 welcome
2 sharmila
3
```

welcome sharmila

I give a change in index.html.That change automatically done.







welcome sharmila good eve how are you?