

# **Project Code-Pipeline**

AWS Code Pipeline is a fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates. Code Pipeline automates the build, test, and deploy phases of your release process every time there is a code change, based on the release model you define. This enables you to rapidly and reliably deliver features and updates. you can easily integrate AWS Code Pipeline with third-party services such as GitHub or with your own custom plugin. With AWS Code Pipeline, you only pay for what you use. There are no upfront fees or long-term commitments

This activity guide cover steps for:

- 1.Create a Code Commit repository
- 2.Add sample code to your Code Commit repository
- 3.Create an EC2 Linux instance and install the Code Deploy agent
- 4.To launch an instance
- 5.Create an application in Code Deploy
- 6.Create. your first pipeline in Code Pipeline
- 7.To verify that your pipeline ran successfully
- 8.Modify code code in your Code Commit repository
- 9.To verify your pipeline ran successfully
- 10.Clean up resources

a. Unzip the files from

[https://docs.aws.amazon.com/codepipeline/latest/userguide/samples/SampleApp\\_Linux.zipp](https://docs.aws.amazon.com/codepipeline/latest/userguide/samples/SampleApp_Linux.zipp) into the local directory (for example, /tmp/MyDemoRepo or c:\temp\MyDemoRepo).

Be sure to place the files directly into your local repository. Do not include a SampleApp\_Linux folder. On your local machine for example, your directory and file hierarchy should look like this:

b. Use git commands to upload the code in the local directory to public repository in Code commit

appspec.yml

index.html

LICENSE.txt

scripts

install dependencies

start server

stop server

c. Create and configure Code Deploy (Amazon EC2 instance)

d. Create a pipeline using AWS Code Pipeline , AWS Code Commit and Code Deploy to deploy index.html to Amazon Linux ec2 instance

e. Check the o/p, by accessing the public id of the ec2 instance. Contents of the index.html must be displayed

## Step 1: Creating IAM User and Roles.

The screenshot shows the AWS IAM console interface. The left sidebar contains the 'Identity and Access Management (IAM)' menu with options like Dashboard, Access management, Users, Roles, Policies, and Access reports. The main content area displays the details for a user named 'Suraj'. The 'Summary' section shows the user's ARN, console access status (Disabled), and creation date. The 'Permissions policies' section shows one policy attached: 'AWSCodeCommitFullAccess'.

**Suraj** Info

Summary

ARN arn:aws:iam::229569186836:user/Suraj	Console access Disabled	Access key 1 <a href="#">Create access key</a>
Created November 02, 2023, 11:07 (UTC+05:30)	Last console sign-in -	

Permissions policies (1)

Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via
<a href="#">AWSCodeCommitFullAccess</a>	AWS managed	Directly

## Created a role for EC2-CodeDeploy

The screenshot shows the AWS IAM console interface. The left sidebar contains the 'Identity and Access Management (IAM)' menu. The main content area displays the details for a role named 'TEST-EC2-CODEDEPOLY'. The 'Summary' section shows the role's ARN, creation date, and last activity. The 'Permissions policies' section shows four policies attached: 'AWSCodeCommitFullAccess', 'AWSCodeDeployFullAccess', 'AWSCodeDeployReadOnlyAccess', and 'AWSCodeCommitReadOnlyAccess'.

**TEST-EC2-CODEDEPOLY** Info

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date November 02, 2023, 11:09 (UTC+05:30)	ARN arn:aws:iam::229569186836:role/TEST-EC2-CODEDEPOLY	Instance profile ARN arn:aws:iam::229569186836:instance-profile/TEST-EC2-CODEDEPOLY
Last activity 54 minutes ago	Maximum session duration 1 hour	

Permissions policies (4)

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
<a href="#">AWSCodeCommitFullAccess</a>	AWS managed	
<a href="#">AWSCodeDeployFullAccess</a>	AWS managed	
<a href="#">AWSCodeDeployReadOnlyAccess</a>	AWS managed	
<a href="#">AWSCodeCommitReadOnlyAccess</a>	AWS managed	

## Created a role for Code-Deploy

The screenshot shows the AWS IAM console for the role 'TEST-CODE-DEPLOY'. The left sidebar contains the 'Identity and Access Management (IAM)' menu with options like Dashboard, Access management, Users, Roles, Policies, Identity providers, Account settings, Access reports, Access analyzer, Archive rules, Analyzers, Settings, Credential report, Organization activity, and Service control policies (SCPs). The main content area shows the role details:

- Summary:** Creation date: November 02, 2023, 11:10 (UTC+05:30); Last activity: 1 hour ago; ARN: `arn:aws:iam::229569186836:role/TEST-CODE-DEPLOY`; Maximum session duration: 1 hour.
- Permissions:** A section titled 'Permissions policies (1)' with a search bar and a filter by type dropdown set to 'All types'. A table lists the attached policies:

Policy name	Type	Attached entities
<a href="#">AWSCodeDeployRole</a>	AWS managed	2

The bottom of the screenshot shows the Windows taskbar with the taskbar search bar, taskbar icons, and the system tray showing the date and time as 12:42 PM.

## Step 2: Creating a Code-Commit Repository

The screenshot shows the AWS CodeCommit console for the 'test' repository. The left sidebar contains the 'Developer Tools' menu with options like Source, CodeCommit, Getting started, Repositories, Approval rule templates, Artifacts, CodeArtifact, Build, CodeBuild, Deploy, CodeDeploy, Pipeline, CodePipeline, and Settings. The main content area shows the repository details:

- Repositories:** A section with a search bar and a table listing the repositories.

Name	Description	Last modified	Clone URL
<a href="#">test</a>	-	46 minutes ago	<a href="#">HTTPS</a> <a href="#">SSH</a> <a href="#">HTTPS (GRC)</a>

The bottom of the screenshot shows the Windows taskbar with the taskbar search bar, taskbar icons, and the system tray showing the date and time as 12:46 PM.

## Step 3: Add sample code to your Code-Commit repository

```
C:\Windows\System32\cmd.exe
D:\Demo12>git clone https://git-codecommit.us-east-2.amazonaws.com/v1/repos/test
Cloning into 'test'...
warning: You appear to have cloned an empty repository.

D:\Demo12>cd test

D:\Demo12\test>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        LICENSE.txt
        appspec.yml
        index.html
        scripts/

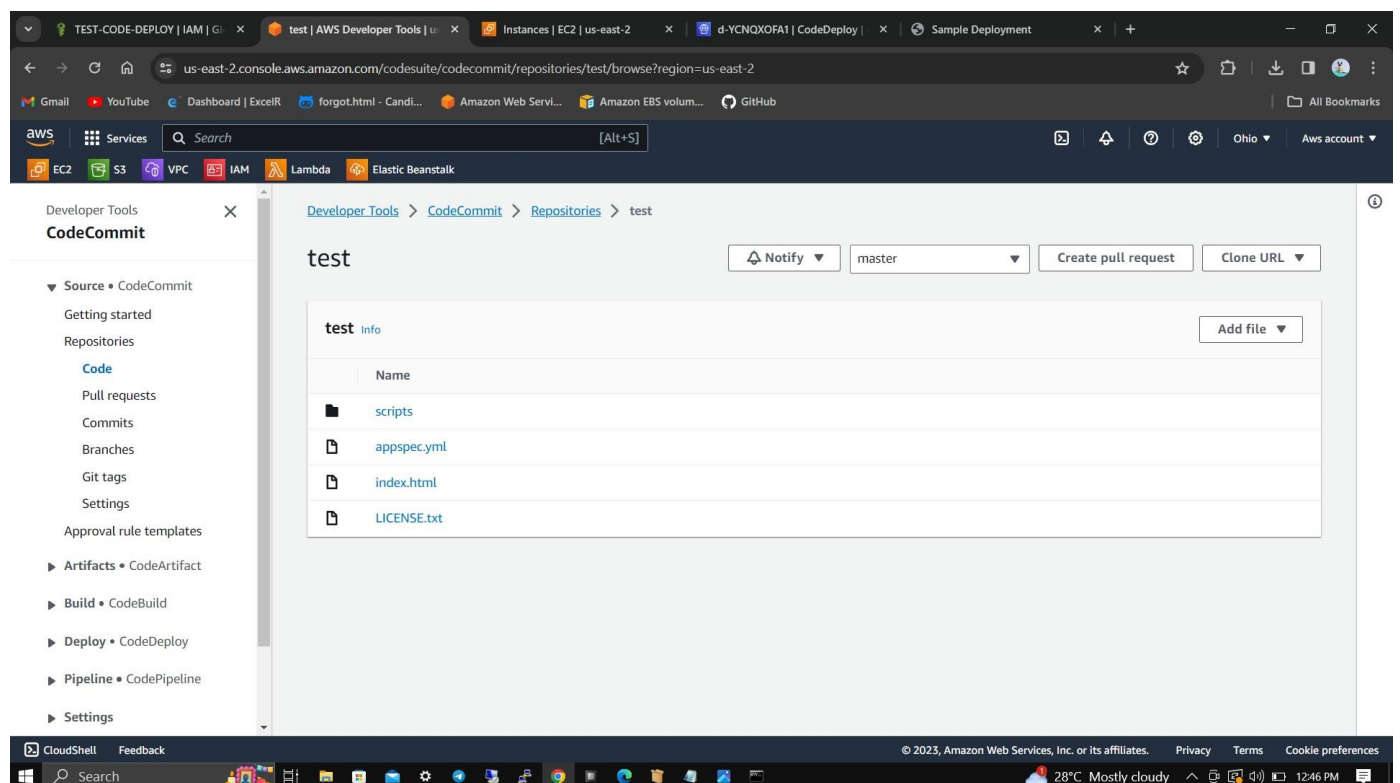
nothing added to commit but untracked files present (use "git add" to track)

D:\Demo12\test>git add -A
warning: in the working copy of 'LICENSE.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'appspec.yml', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'scripts/install_dependencies', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'scripts/start_server', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'scripts/stop_server', LF will be replaced by CRLF the next time Git touches it

D:\Demo12\test>git commit -m"Code Git successs"
[master (root-commit) 2e129df] Code Git successs
 6 files changed, 266 insertions(+)
 create mode 100644 LICENSE.txt
 create mode 100644 appspec.yml
 create mode 100644 index.html
 create mode 100644 scripts/install_dependencies
 create mode 100644 scripts/start_server
 create mode 100644 scripts/stop_server

D:\Demo12\test>git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 5.01 KiB | 1.25 MiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Validating objects: 100%
To https://git-codecommit.us-east-2.amazonaws.com/v1/repos/test
 * [new branch]      master -> master

D:\Demo12\test>
```



## Step 4: Creating Amazon Linux EC2 instance:

The screenshot shows the AWS Management Console for the us-east-2 region. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, and Snapshots. The main content area displays the 'Instances (1/1)' page. A table lists the instance 'code-dep' with ID 'i-0cbce29ed59be862b', state 'Running', type 't2.medium', and public IP 'ec2-3-145-218-4'. Below the table, the 'Instance: i-0cbce29ed59be862b (code-dep)' details are shown, including the instance ID, public IP address (3.145.218.4), private IP address (172.31.23.115), instance state (Running), public IP DNS (ec2-3-145-218-4.us-east-2.compute.amazonaws.com), and private IP DNS name (ip-172-31-23-115.us-east-2.compute.internal).

The screenshot shows the AWS Management Console for the us-east-2 region, specifically the 'Instance details' page for the instance 'code-dep' (i-0cbce29ed59be862b). The page displays the 'Instance summary for i-0cbce29ed59be862b (code-dep)' with various details. The instance is in a 'Running' state. The public IP address is 3.145.218.4, and the private IP address is 172.31.23.115. The public IP DNS is ec2-3-145-218-4.us-east-2.compute.amazonaws.com, and the private IP DNS name is ip-172-31-23-115.us-east-2.compute.internal. The instance type is t2.medium, and the VPC ID is vpc-0125ee7beb20a0c43. The subnet ID is subnet-0424919d440566c4c. The IAM Role is TEST-EC2-CODEDEPOLY, and the IMDSv2 is Required. The page also includes an 'AWS Compute Optimizer finding' section with a recommendation to 'Opt-in to AWS Compute Optimizer for recommendations'.



## Step 5: Creating Code-Deploy Application and Code-Deployment Group.

The screenshot shows the AWS CodeDeploy console in the 'us-east-2' region. The left sidebar displays the 'Developer Tools' menu with 'CodeDeploy' selected. The main content area shows the 'Applications' page with a search bar, a 'Notify' button, and a 'View details' button. A table lists the applications:

Application name	Compute platform	Created
code-dep-app123	EC2/On-premises	42 minutes ago

At the bottom of the console, the Windows taskbar is visible, showing the system clock at 12:47 PM and the SPBSMCSI stock price at +1.01%.

The screenshot shows the AWS CodeDeploy console for the application 'code-dep-app123'. The left sidebar displays the 'Developer Tools' menu with 'CodeDeploy' selected. The main content area shows the 'code-dep-app123' application details page with a search bar, a 'Notify' button, and a 'Delete application' button. The 'Application details' section shows the application name 'code-dep-app123' and the compute platform 'EC2/On-premises'. The 'Deployment groups' section shows a table with the following data:

Name	Status	Last attempted deploy...	Last successful deploy...	Trigger count
dep-test-12	Succeeded	Nov 2, 2023 12:06 PM (U...	Nov 2, 2023 12:06 PM (U...	0

At the bottom of the console, the Windows taskbar is visible, showing the system clock at 12:47 PM and the SPBSMCSI stock price at +1.01%.

TEST-CODE-DEPLOY | IAM | Gl... x dep-test-12 | CodeDeploy | us... x Instances | EC2 | us-east-2 x d-YCNQXOFA1 | CodeDeploy | x Sample Deployment x +

us-east-2.console.aws.amazon.com/codesuite/codedeploy/applications/code-dep-app123/deployment-groups/dep-test-12?region=us-east-2&tags-meta=eyJmIjp7fSwicyl6e3...

Gmail YouTube Dashboard | ExcelR forgot.html - Candi... Amazon Web Servi... Amazon EBS volum... GitHub

aws Services Search [Alt+S]

EC2 S3 VPC IAM Lambda Elastic Beanstalk

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

## dep-test-12

Edit Delete Create deployment

### Deployment group details

Deployment group name	Application name	Compute platform
dep-test-12	code-dep-app123	EC2/On-premises
Deployment type	Service role ARN	Deployment configuration
In-place	arn:aws:iam::229569186836:role/TEST-CODE-DEPLOY	CodeDeployDefault.OneAtATime
Rollback enabled	Agent update scheduler	
False	Learn to schedule update in AWS Systems Manager	

### Environment configuration: Amazon EC2 instances

Key	Value
Name	code-dep

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

SPBSMCSI +1.01% 12:47 PM

## Step 6: Creating Pipeline.

TEST-CODE-DEPLOY | IAM | Gl... x Pipelines | CodePipeline | us-e... x Instances | EC2 | us-east-2 x d-YCNQXOFA1 | CodeDeploy | x Sample Deployment x +

us-east-2.console.aws.amazon.com/codesuite/codepipeline/pipelines?region=us-east-2&pipelines-meta=eyJmIjp7InRleHQjOiIifSwicyl6eyJwcm9wZXJ0eSl6InVwZGF0ZWQlClJk...

Gmail YouTube Dashboard | ExcelR forgot.html - Candi... Amazon Web Servi... Amazon EBS volum... GitHub

aws Services Search [Alt+S]

EC2 S3 VPC IAM Lambda Elastic Beanstalk

Developer Tools CodePipeline

- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
- Deploy • CodeDeploy
- Pipeline • CodePipeline
  - Getting started
  - Pipelines
  - Settings

Go to resource Feedback

## Pipelines

info

Refresh Notify View history Release change Delete pipeline Create pipeline

Search

Name	Type	Most recent execution	Latest source revisions	Last executed
code-dep-pip1	V1	Succeeded	Source - 2e129df4: Code Git success	41 minutes ago

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

SPBSMCSI +1.01% 12:47 PM

## Step 7: Verifying that pipeline ran successfully.

The screenshot shows the AWS CodePipeline console for the pipeline named 'code-dep-pip1'. The pipeline is in a 'Success' state, indicated by a green banner at the top that says 'Success: Congratulations! The pipeline code-dep-pip1 has been created.' and a 'Release change' button. The pipeline type is 'V1'. The left sidebar shows the 'CodePipeline' section with options for Source, Artifacts, Build, Deploy, and Pipeline. The main area shows the pipeline stages: 'Source' (Succeeded) and 'Deploy' (Succeeded). The 'Source' stage is expanded, showing a successful execution with ID 'e488abda-d5ba-48f6-b694-23ad4d153899' and commit '2e129df4'. A 'Disable transition' button is visible below the stage. The bottom of the screen shows the Windows taskbar with the time 12:45 PM.

The screenshot shows the AWS CodePipeline console for the pipeline named 'code-dep-pip1'. The pipeline is in a 'Success' state, indicated by a green banner at the top that says 'Success: Congratulations! The pipeline code-dep-pip1 has been created.' and a 'Release change' button. The pipeline type is 'V1'. The left sidebar shows the 'CodePipeline' section with options for Source, Artifacts, Build, Deploy, and Pipeline. The main area shows the pipeline stages: 'Source' (Succeeded) and 'Deploy' (Succeeded). The 'Source' stage is expanded, showing a successful execution with ID 'e488abda-d5ba-48f6-b694-23ad4d153899' and commit '2e129df4'. The 'Deploy' stage is also expanded, showing a successful execution with ID 'e488abda-d5ba-48f6-b694-23ad4d153899' and commit '2e129df4'. A 'Disable transition' button is visible between the stages. The bottom of the screen shows the Windows taskbar with the time 12:45 PM.



## Step 8: Checking the o/p using public Ip of EC2 instance.

