

Hands-on tutorials

# Set Up a Continuous Deployment Pipeline Using AWS CodePipeline



# Set Up a Continuous Deployment Pipeline Using AWS CodePipeline:

## Hands-on tutorials

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

# Table of Contents

<b>Set Up a Continuous Deployment Pipeline Using AWS CodePipeline .....</b>	<b>i</b>
Overview .....	1
What you will accomplish .....	1
Prerequisites .....	2
Implementation .....	2
Conclusion .....	66

# Set Up a Continuous Deployment Pipeline Using AWS CodePipeline

<b>AWS experience</b>	Beginner
<b>Minimum time to complete</b>	30 minutes
<b>Cost to complete</b>	<a href="#">Free Tier</a> eligible
<b>Services used</b>	<a href="#">AWS CodePipeline</a> <a href="#">AWS Elastic Beanstalk</a>
<b>Last updated</b>	February 14, 2023

## Overview

In this tutorial, you will learn how to create an automated software release pipeline that deploys a live sample app. You will create the pipeline using AWS CodePipeline, a service that builds, tests, and deploys your code every time there is a code change. You will use your GitHub account, an Amazon Simple Storage Service (Amazon S3) bucket, or an AWS CodeCommit repository as the source location for the sample app's code. You will also use AWS Elastic Beanstalk as the deployment target for the sample app. Your completed pipeline will be able to detect changes made to the source repository containing the sample app and then automatically update your live sample app.

Continuous deployment allows you to deploy revisions to a production environment automatically without explicit approval from a developer, making the entire software release process automated.

Everything done in this tutorial is Free Tier eligible.

## What you will accomplish

In this tutorial, you will:

- create an automated software release pipeline that deploys a live sample app



- create the pipeline using AWS CodePipeline
- use AWS Elastic Beanstalk as the deployment target for the sample app

## Prerequisites

Before starting this tutorial, you will need an AWS account. If you don't already have one, follow the [Setting Up Your AWS Environment](#) getting started guide for a quick overview.

## Implementation

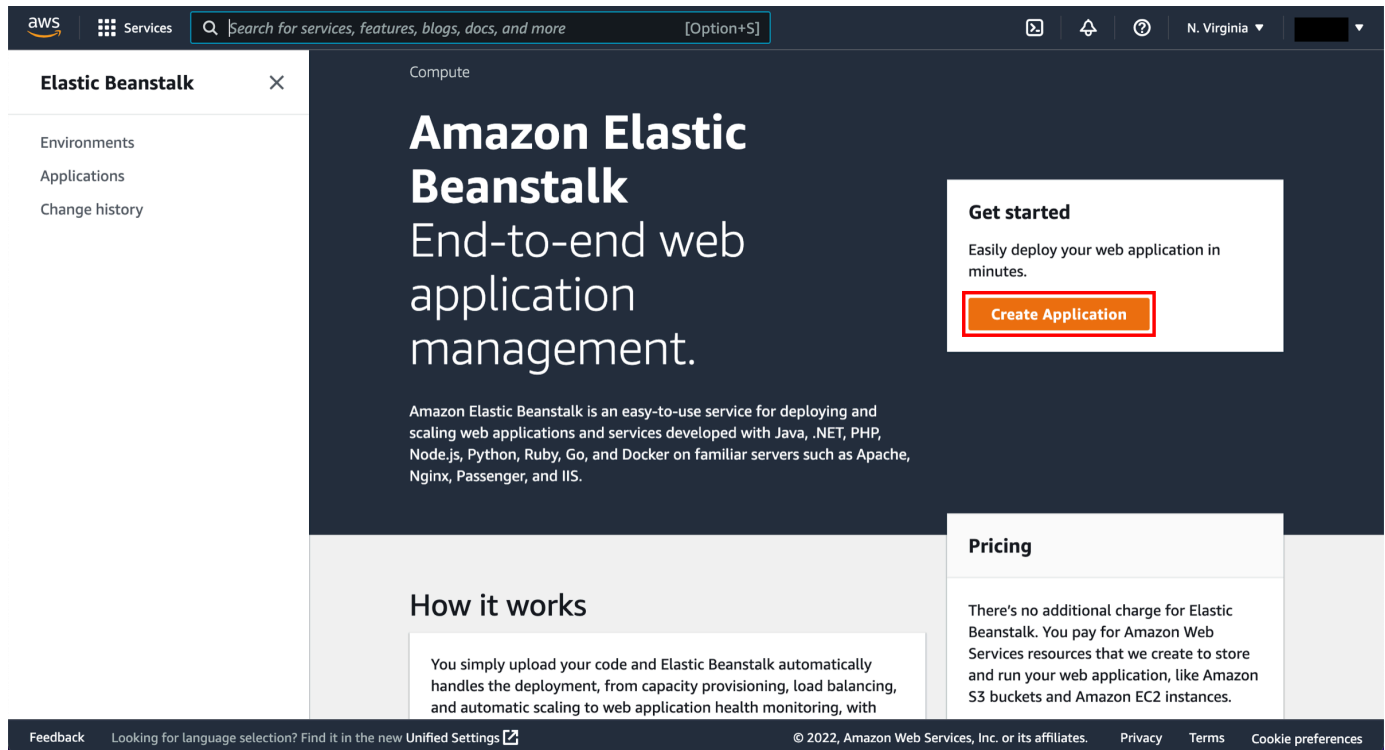
### Step 1: Create a deployment environment

Your continuous deployment pipeline will need a target environment containing virtual servers, or Amazon EC2 instances, where it will deploy sample code. You will prepare this environment before creating the pipeline.

To simplify the process of setting up and configuring EC2 instances for this tutorial, you will spin up a sample environment using AWS Elastic Beanstalk. With Elastic Beanstalk you can easily host web applications without needing to launch, configure, or operate virtual servers on your own. It automatically provisions and operates the infrastructure (such as virtual servers and load balancers) and provides the application stack (such as OS, language and framework, and web and application server) for you.

#### 1. Create an application

To start, open the [Elastic Beanstalk console](#) and choose **Create Application**.



## 2. Configure the application

For **Application name**, enter **Deployment Tutorial**. Select **PHP** from the dropdown menu under **Platform**, and choose **Create application**.

### Note

If you have created an Elastic Beanstalk application before, choose **Create New Application** on the upper-right corner. Name your application and create a new **web server environment**. Select **PHP** as your **Platform** and **Single Instance** as your **Environment type**. If you are planning to remote login to your instances, select a key pair. Otherwise, leave default values for the remaining options and create the environment for your continuous deployment pipeline.

**Elastic Beanstalk** ×

Environments  
Applications  
Change history

Elastic Beanstalk > Getting started

## Create a web app

Create a new application and environment with a sample application or your own code. By creating an environment, you allow Amazon Elastic Beanstalk to manage Amazon Web Services resources and permissions on your behalf. [Learn more](#)

### Application information

Application name  
**Deployment Tutorial**  
Up to 100 Unicode characters, not including forward slash (/).

### Application tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

Key	Value	
<input type="text"/>	<input type="text"/>	<button>Remove tag</button>

Add tag  
50 remaining

### Platform

Platform  
**PHP**

Platform branch  
PHP 8.1 running on 64bit Amazon Linux 2

Platform version  
3.4.0 (Recommended)

### Application code

☒ **Sample application**  
Get started right away with sample code.

☐ **Upload your code**  
Upload a source bundle from your computer or copy one from Amazon S3.

Cancel Configure more options **Create application**

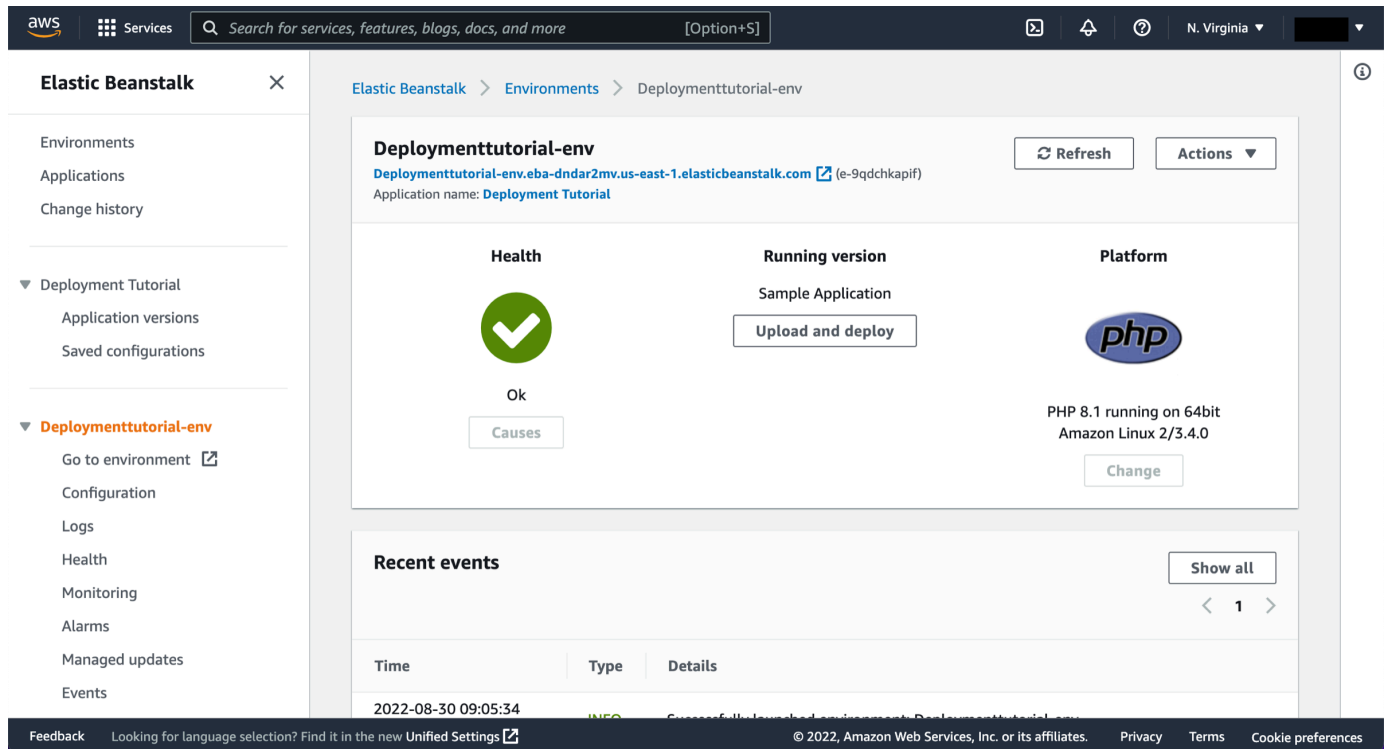
Feedback Looking for language selection? Find it in the new Unified Settings [↗](#) © 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

### 3. Create a sample environment

Elastic Beanstalk will begin creating a sample environment for you to deploy your application to. It will create an Amazon EC2 instance, a security group, an Auto Scaling group, an Amazon S3 bucket, Amazon CloudWatch alarms, and a domain name for your application.

**Note**

This will take several minutes to complete.



## Step 2: Get a copy of the sample code

In this step, you will retrieve a copy of the sample app's code and choose a source to host the code. The pipeline takes code from the source and then performs actions on it.

You can use one of three options as your source: a GitHub repository, an Amazon S3 bucket, or an AWS CodeCommit repository. Select your preference and follow the steps.

### GitHub

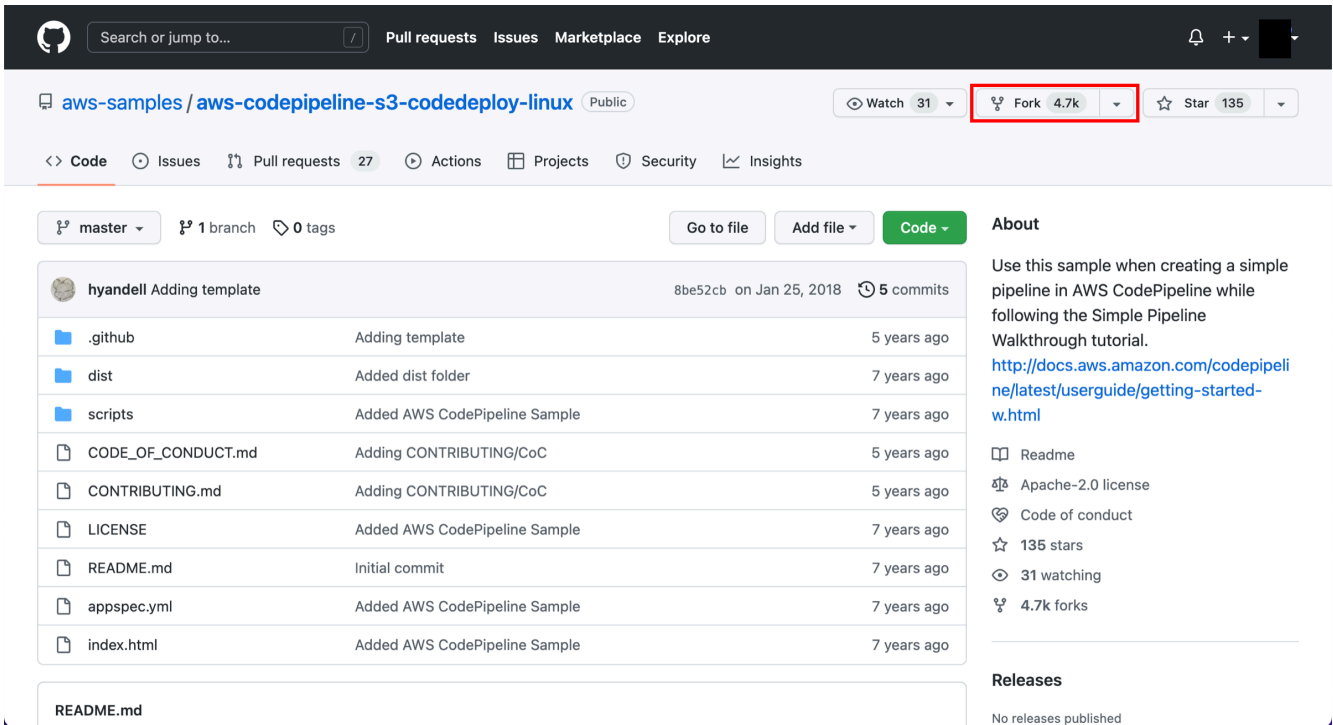
Use this procedure if you would like to use your GitHub account as your source.

- Fork the repository

If you would like to use your GitHub account:

- Visit our GitHub repository containing the sample code at <https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux>.
- Fork a copy of the repository to your own GitHub account by choosing the **Fork** button in the upper-right corner.

Then, go to **Create your pipeline**.



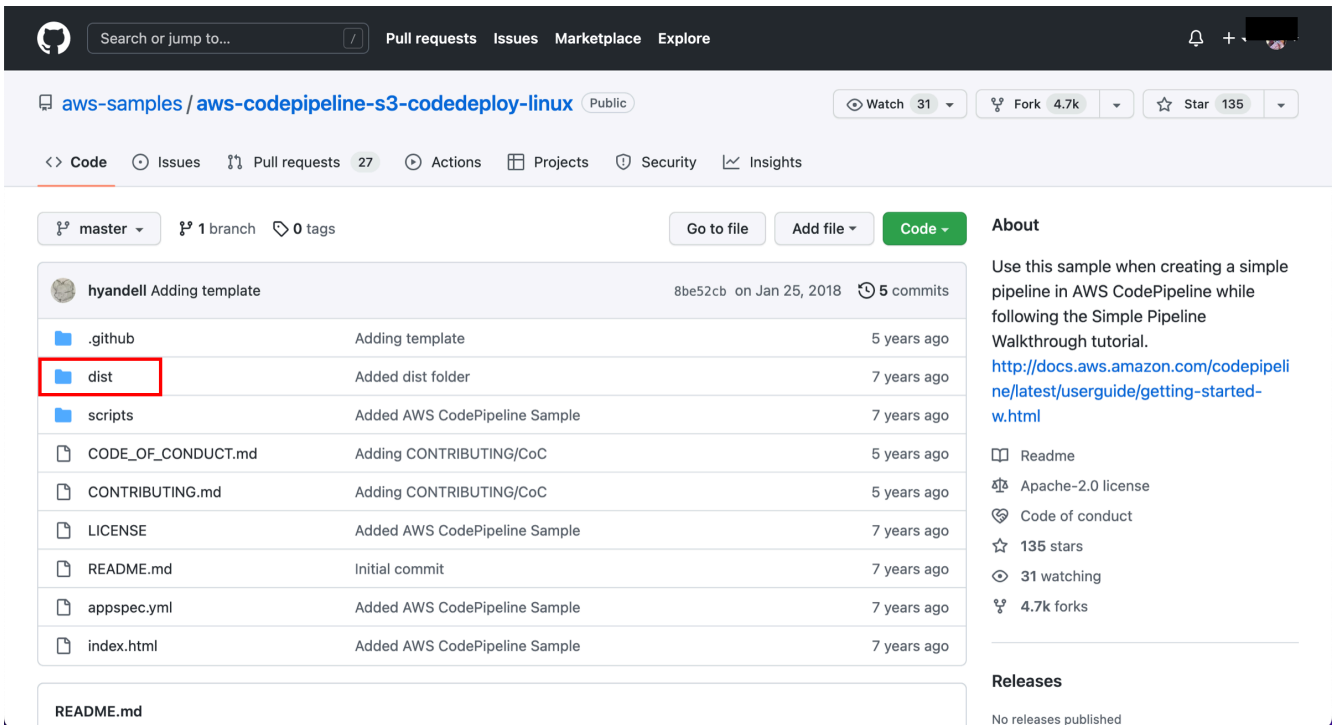
## Amazon S3

Use this procedure if you would like to use Amazon S3 as your source.

### 1. Navigate to the sample code

If you plan to use Amazon S3 as your source, you will retrieve the sample code from the AWS GitHub repository, save it to your computer, and upload it to an Amazon S3 bucket.

- Visit our GitHub repository containing the sample code at <https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux>
- Select the **dist** folder.



Search or jump to... Pull requests Issues Marketplace Explore

aws-samples / aws-codepipeline-s3-codedeploy-linux Public

Watch 31 Fork 4.7k Star 135

Code Issues Pull requests 27 Actions Projects Security Insights

master 1 branch 0 tags

Go to file Add file Code

hyandell Adding template 8be52cb on Jan 25, 2018 5 commits

.github	Adding template	5 years ago
dist	Added dist folder	7 years ago
scripts	Added AWS CodePipeline Sample	7 years ago
CODE_OF_CONDUCT.md	Adding CONTRIBUTING/CoC	5 years ago
CONTRIBUTING.md	Adding CONTRIBUTING/CoC	5 years ago
LICENSE	Added AWS CodePipeline Sample	7 years ago
README.md	Initial commit	7 years ago
appspec.yml	Added AWS CodePipeline Sample	7 years ago
index.html	Added AWS CodePipeline Sample	7 years ago

README.md

About

Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walkthrough tutorial.  
<http://docs.aws.amazon.com/codepipeline/latest/userguide/getting-started-w.html>

Readme Apache-2.0 license Code of conduct 135 stars 31 watching 4.7k forks

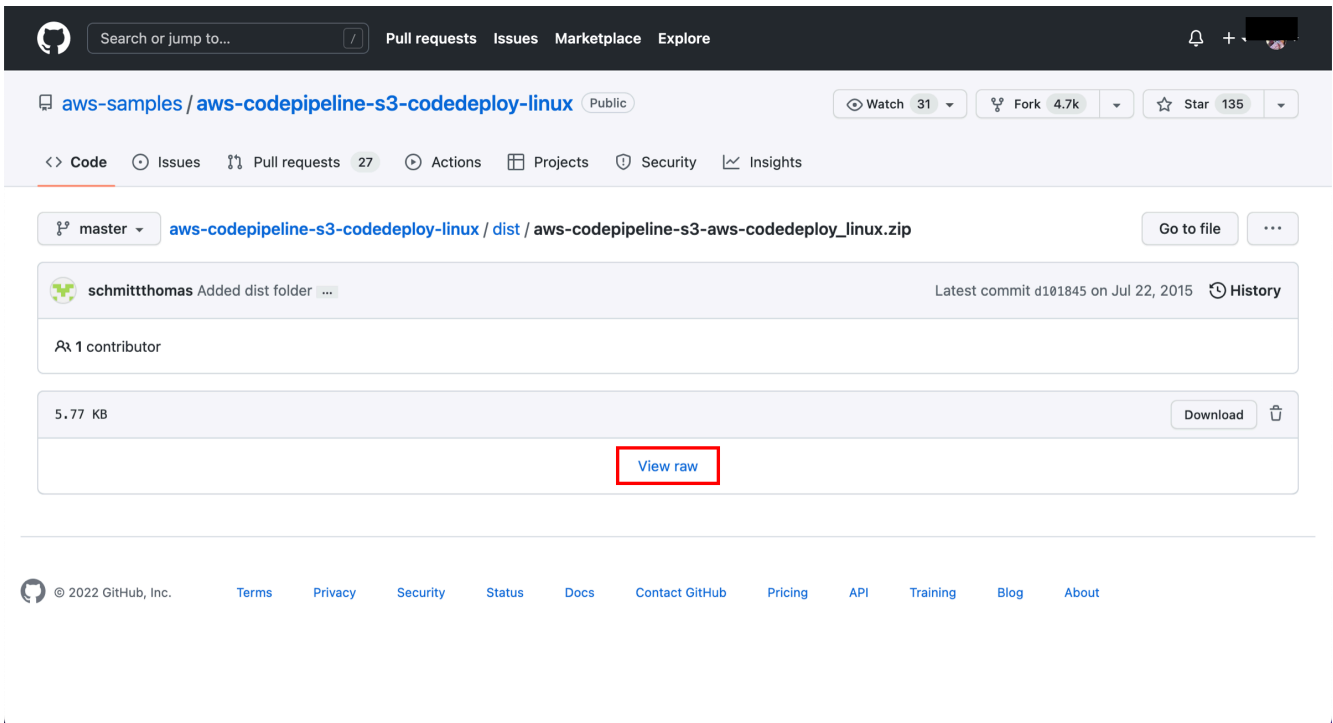
Releases

No releases published

## 2. Download the files

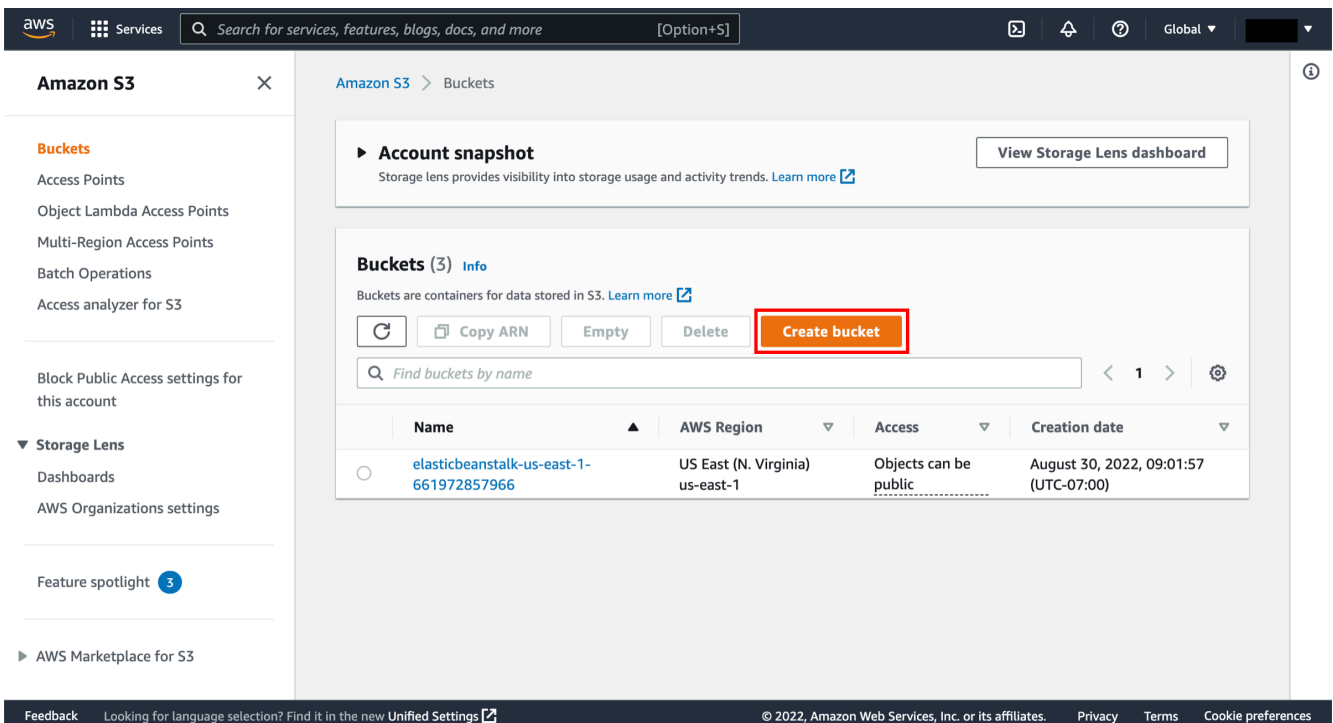
Save the source files to your computer:

- Select the file named **aws-codepipeline-s3-aws-codedeploy\_linux.zip**.
- Choose **View raw**.
- Save the sample file to your local computer.



### 3. Create a bucket

Open the [Amazon S3 console](#) and choose **Create bucket**.



### 4. Configure bucket details

**Bucket name:** Enter a unique name for your bucket, such as **awscodepipeline-demobucket-variables**. All bucket names in Amazon S3 must be unique, so use one of your own, not one with the name shown in the example.

**Region:** In the dropdown, select the Region where you will create your pipeline, such as US East (N. Virginia).

Choose **Create bucket**.



aws

Services

Search for services, features, blogs, docs, and more

[Option+S]

Global

Amazon S3

Buckets

Create bucket

Create bucket

Info

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

awscodepipeline-demobucket-830

Bucket name must be globally unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

US East (N. Virginia) us-east-1

Copy settings from existing bucket - optional

Only the bucket settings in the following configuration are copied.

Choose bucket

Object Ownership

Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

☒ ACLs disabled (recommended)  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☐ ACLs enabled  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced

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.

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

Tags (0) - optional

Track storage cost or other criteria by tagging your bucket. [Learn more](#)

No tags associated with this bucket.

Add tag

Default encryption

Automatically encrypt new objects stored in this bucket. [Learn more](#)

## 5. View the created bucket

The console displays the newly created bucket, which is empty.

**Amazon S3** ×

Search for services, features, blogs, docs, and more [Option+S]

Global ▼

Successfully created bucket "awscodepipeline-demobucket-8302022"  
To upload files and folders, or to configure additional bucket settings choose [View details](#).

View details ×

Amazon S3 > Buckets

**Account snapshot**  
Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

View Storage Lens dashboard

**Buckets (4)** [Refresh](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

	Name ▲	AWS Region ▼	Access ▼	Creation date ▼
<input type="radio"/>	<a href="#">awscodepipeline-demobucket-8302022</a>	US East (N. Virginia) us-east-1	Bucket and objects not public	August 30, 2022, 10:02:19 (UTC-07:00)
<input type="radio"/>	<a href="#">elasticbeanstalk-us-east-1-661972857966</a>	US East (N. Virginia) us-east-1	Objects can be public	August 30, 2022, 09:01:57 (UTC-07:00)

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

## 6. Open the bucket

You will now upload the sample code to the Amazon S3 bucket. Select the Amazon S3 bucket.

**Amazon S3**

**Buckets**

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

Access analyzer for S3

Block Public Access settings for this account

**Storage Lens**

Dashboards

AWS Organizations settings

Feature spotlight 3

AWS Marketplace for S3

**Successfully created bucket "awscodepipeline-demobucket-8302022"**

To upload files and folders, or to configure additional bucket settings choose [View details](#).

[View details](#)

Amazon S3 > Buckets

**Account snapshot**

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

**Buckets (4)**

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

	Name	AWS Region	Access	Creation date
<input type="radio"/>	awscodepipeline-demobucket-8302022	US East (N. Virginia) us-east-1	Bucket and objects not public	August 30, 2022, 10:02:19 (UTC-07:00)
<input type="radio"/>	elasticbeanstalk-us-east-1-661972857966	US East (N. Virginia) us-east-1	Objects can be public	August 30, 2022, 09:01:57 (UTC-07:00)

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 7. Upload the sample code

### Select Upload.

**Amazon S3**

**Buckets**

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

Access analyzer for S3

Block Public Access settings for this account

**Storage Lens**

Dashboards

AWS Organizations settings

Feature spotlight 3

AWS Marketplace for S3

Amazon S3 > Buckets > awscodepipeline-demobucket-8302022

**awscodepipeline-demobucket-8302022**

**Objects** Properties Permissions Metrics Management Access Points

**Objects (0)**

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](#)

Copy S3 URI Copy URL Download Open Delete Actions

Create folder Upload

Find objects by prefix Show versions

	Name	Type	Last modified	Size	Storage class
No objects					
You don't have any objects in this bucket.					

Upload

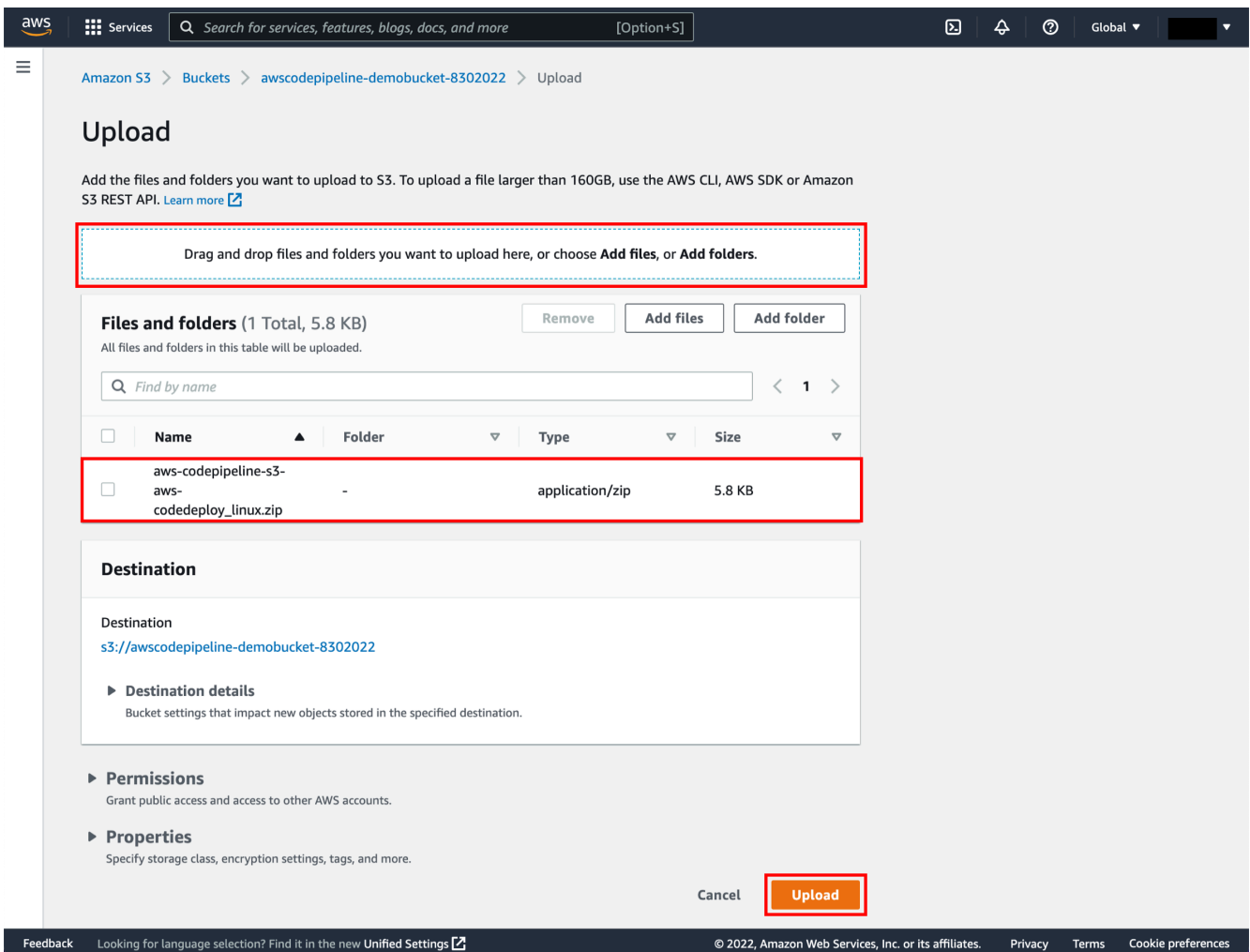
<https://us-east-1.console.aws.amazon.com/s3/#> Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 8. Add files

Select **Add files** to upload the zip file you downloaded earlier or drag and drop the file. Then select **Upload**.

Then, go to **Create your pipeline**.



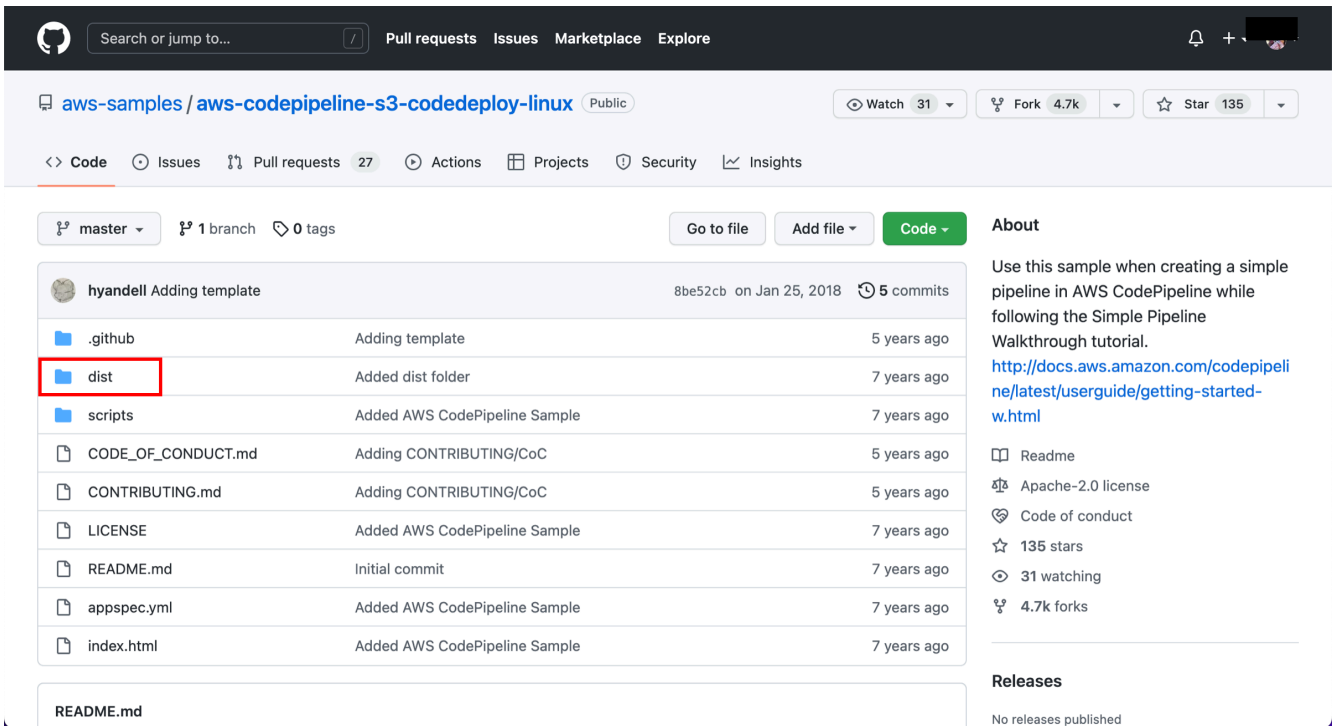
## AWS CodeCommit

Use this procedure if you would like to use AWS CodeCommit as your source.

### 1. Navigate to the sample code

If you plan to use AWS CodeCommit as your source, you will retrieve the sample code from the AWS GitHub repository, save it to your computer, and upload it to an AWS CodeCommit repository.

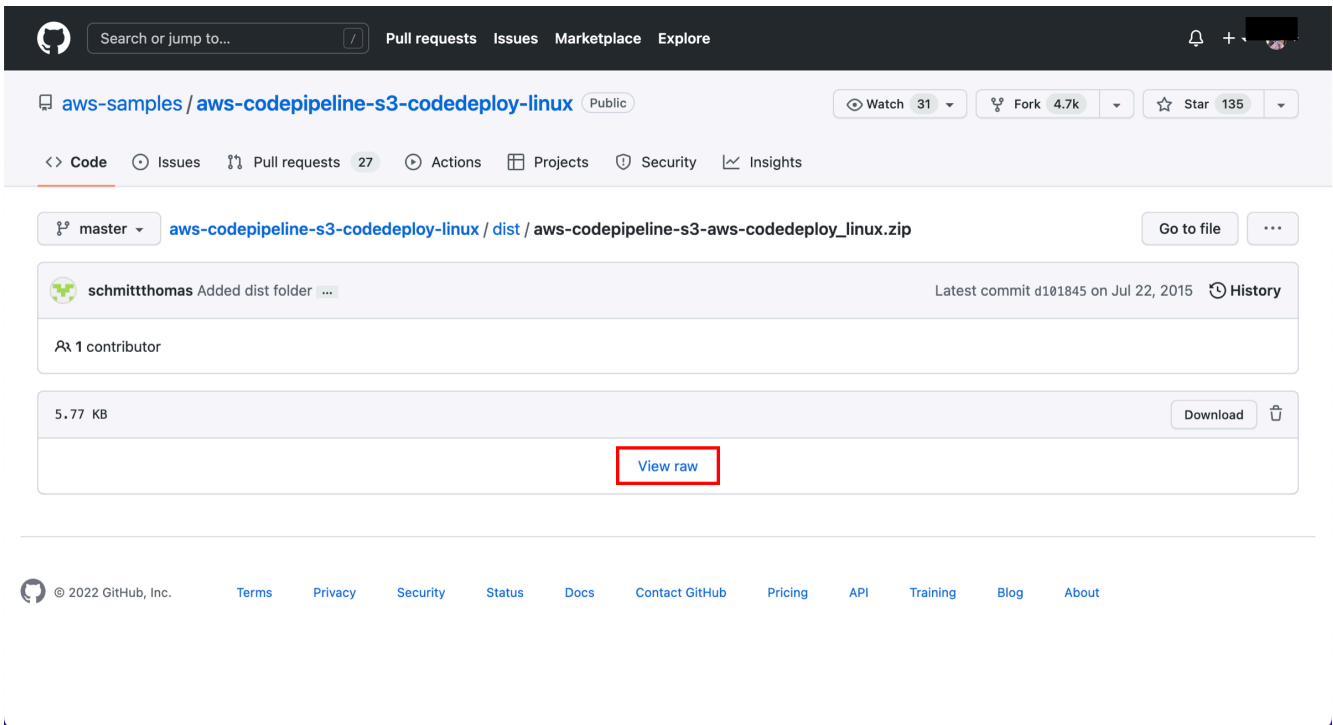
- Visit our GitHub repository containing the sample code at <https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux>
- Select the **dist** folder.



## 2. Download the files

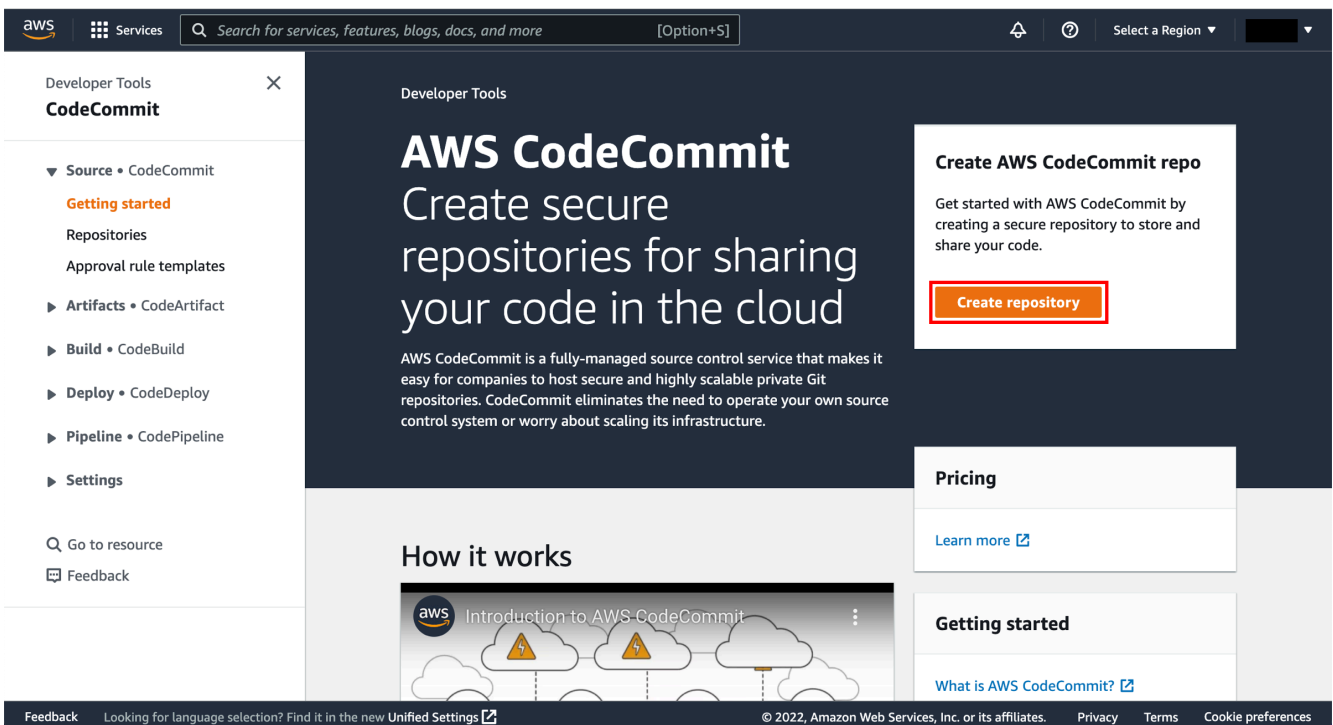
Save the source files to your computer:

- a. Select the file named **aws-codepipeline-s3-aws-codedeploy\_linux.zip**.
- b. Choose **View raw**.
- c. Save the sample file to your local computer.



### 3. Create a repository

Open the [AWS CodeCommit console](#) and choose **Create repository**.



### 4. Configure repository settings

On the **Create repository** page:

Enter **PipelineRepo** for **Repository name**.

Choose **Create**.

The screenshot shows the AWS CodeCommit 'Create repository' page. The breadcrumb navigation at the top indicates the path: Developer Tools > CodeCommit > Repositories > Create repository. The main heading is 'Create repository', followed by a subheading: 'Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.'

The 'Repository settings' section contains the following fields and options:

- Repository name:** A text input field containing 'PipelineRepo'. A red box highlights this field. Below the field, it says '100 characters maximum. Other limits apply.'
- Description - optional:** A text area for a description. Below it, it says '1,000 characters maximum'.
- Tags:** A section with an 'Add' button.
- Enable Amazon CodeGuru Reviewer for Java and Python - optional:** An unchecked checkbox. Below it, text reads: 'Get recommendations to improve the quality of the Java and Python code for all pull requests in this repository. A service-linked role will be created in IAM on your behalf if it does not exist.'

At the bottom right of the settings section, there are 'Cancel' and 'Create' buttons. The 'Create' button is highlighted with a red box.

The footer of the page includes a 'Feedback' link, a link to 'Unified Settings', and copyright information: '© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

## 5. Upload sample code

Once the repository is successfully created, scroll down to the **PipelineRepo** section and select **Add file**, then choose **Upload file**.

Developer Tools  
**CodeCommit**

- Source • CodeCommit
  - Getting started
  - Repositories
  - Code**
  - Pull requests
  - Commits
  - Branches
  - Git tags
  - Settings
- Approval rule templates
- Artifacts • CodeArtifact
- Build • CodeBuild
- Deploy • CodeDeploy
- Pipeline • CodePipeline
- Settings

**Success**  
Repository successfully created

Create a notification rule for this repository

**Step 2: Set up the AWS CLI Credential Helper**  
Set up your connection to AWS CodeCommit repositories using the credential helper included in the AWS CLI. This is the only connection method for AWS CodeCommit repositories that does not require an IAM user, so it is the only method that supports root access, federated access, and temporary credentials. [Learn more](#)

**Additional details**  
You can find more detailed instructions in the documentation. [View documentation](#)

**PipelineRepo** Info

Name	Size	Actions
<p>Empty repository</p> <p>Your repository is currently empty. You can add files to it directly from the console or by cloning the repository to your local computer, creating commits, and pushing content to the remote repository in AWS CodeCommit.</p> <p>Create file</p>		

Buttons: Add file, Create file, Upload file

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 6. Select the sample code file

On the **Upload a file** page, choose the **Choose file** button and select the downloaded `aws-codepipeline-s3-aws-codedeploy_linux.zip` file.

Developer Tools  
**CodeCommit**

- Source • CodeCommit
  - Getting started
  - Repositories
  - Code**
  - Pull requests
  - Commits
  - Branches
  - Git tags
  - Settings
- Approval rule templates
- Artifacts • CodeArtifact
- Build • CodeBuild
- Deploy • CodeDeploy
- Pipeline • CodePipeline
- Settings

Developer Tools > CodeCommit > Repositories > PipelineRepo > File

**Upload a file**

**PipelineRepo** Info

Name	Size	Actions
<p>Upload file</p> <p>Choose a file to upload.</p> <p>Choose file</p>		

**Commit changes to main**

Author name

Email address

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## 7. Commit changes to main



Enter an **Author name** and **Email address**, then choose **Commit changes**.

Then, go to **Create your pipeline**.

Developer Tools **CodeCommit**

Source • CodeCommit

- Getting started
- Repositories
- Code**
- Pull requests
- Commits
- Branches
- Git tags
- Settings

Approval rule templates

Artifacts • CodeArtifact

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Settings

Developer Tools > CodeCommit > Repositories > PipelineRepo > File

### Upload a file

**PipelineRepo** Info

Name	Size	Actions
aws-codepipeline-s3-aws-codedeploy_linux.zip	6 KB	<button>Remove file</button>

**Commit changes to main**  
File: PipelineRepo/aws-codepipeline-s3-aws-codedeploy\_linux.zip

Author name

Email address

Commit message - optional  
A default commit message will be used if you do not provide one.

Cancel **Commit changes**

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

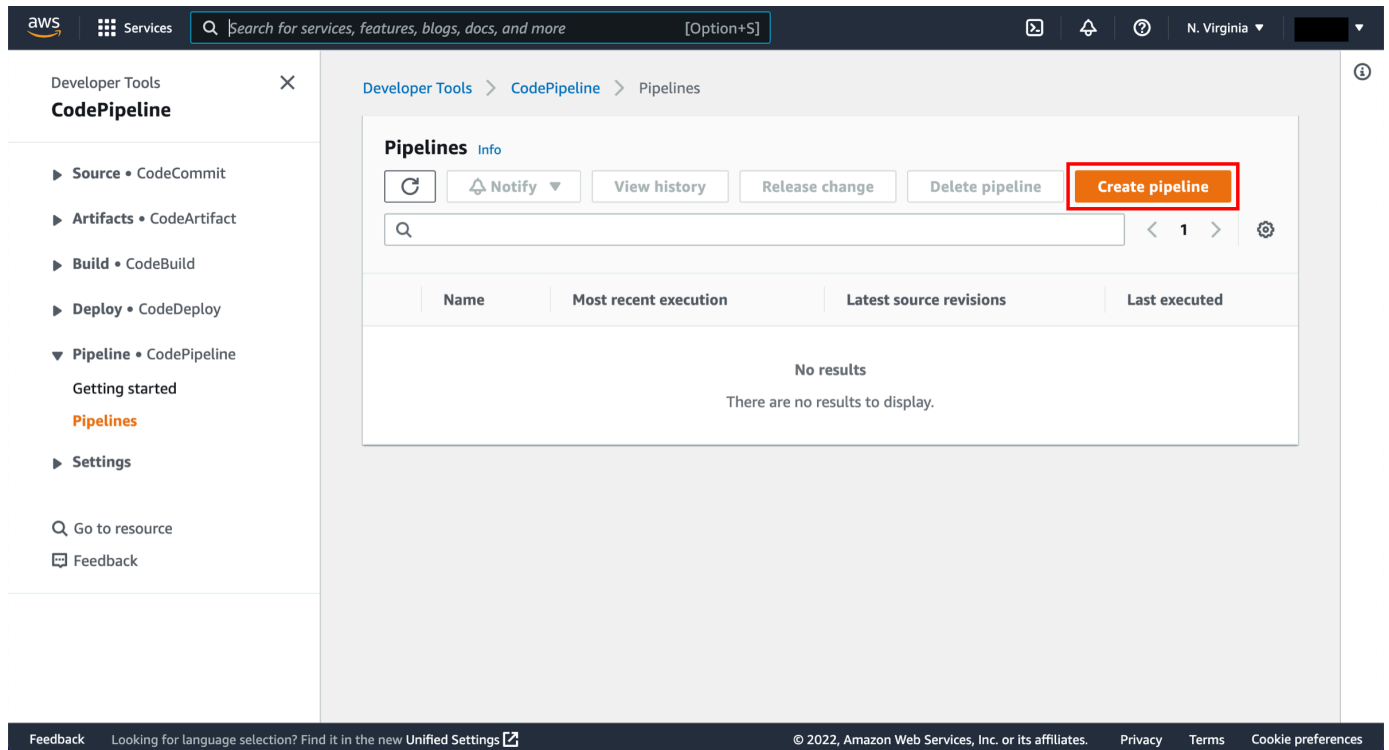
## Step 3: Create your pipeline

In this step, you will create and configure a simple pipeline with two actions: source and deploy. You will provide CodePipeline with the locations of your source repository and deployment environment.

### 1. Create a pipeline

Open the [AWS CodePipeline console](#).

On the **Welcome** page, choose **Create pipeline**.



## 2. Configure pipeline settings

On the **Step 1: Choose pipeline settings** page:

- **Pipeline name:** Enter the name for your pipeline, **DemoPipeline**.
- Choose **Next**.

### Note

After you create a pipeline, you cannot change its name.

aws Services Search for services, features, blogs, docs, and more [Option+S] N. Virginia

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1  
**Choose pipeline settings**

Step 2  
Add source stage

Step 3  
Add build stage

Step 4  
Add deploy stage

Step 5  
Review

### Choose pipeline settings [Info](#)

#### Pipeline settings

**Pipeline name**  
Enter the pipeline name. You cannot edit the pipeline name after it is created.

**DemoPipeline**

No more than 100 characters

**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**

**AWSCodePipelineServiceRole-us-east-1-DemoPipeline**

Type your service role name

☒ **Allow AWS CodePipeline to create a service role so it can be used with this new pipeline**

► **Advanced settings**

Cancel **Next**

Feedback Looking for language selection? Find it in the new Unified Settings © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

### 3. Choose your source

On the **Step 2: Add source stage** page, select the location of the source you selected using the following instructions.

#### GitHub

##### 1. Add source

Select **GitHub (Version 2)** for the **Source provider**.

Choose **Connect to GitHub**.

The screenshot shows the AWS CodePipeline console interface. The breadcrumb navigation at the top reads: Developer Tools > CodePipeline > Pipelines > Create new pipeline. On the left sidebar, the steps are listed: Step 1: Choose pipeline settings, Step 2: Add source stage (selected), Step 3: Add build stage, Step 4: Add deploy stage, Step 5: Review. The main content area is titled 'Add source stage' with an 'Info' link. It contains the following sections:

- Source**
  - Source provider**: This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details. The dropdown menu is set to 'GitHub (Version 2)'.
  - New GitHub version 2 (app-based) action**: To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. [Learn more](#)
  - Connection**: Choose an existing connection that you have already configured, or create a new one and then return to this task. There is a search input field and a button labeled 'Connect to GitHub' which is highlighted with a red box.
  - Repository name**: Choose a repository in your GitHub account. There is a search input field and a placeholder text '<account>/<repository-name>'.

The footer of the console shows: Feedback, Looking for language selection? Find it in the new Unified Settings, © 2022, Amazon Web Services, Inc. or its affiliates, Privacy, Terms, Cookie preferences.

## 2. Enter a connection name

Enter **Deployment Tutorial** for **Connection name** and choose **Connect to GitHub**.

The screenshot shows the AWS Developer Tools console interface. The browser address bar displays the URL: `us-east-1.console.aws.amazon.com/codesuite/settings/connections/create?origin...`. The page title is "Settings - AWS Developer Tools". The navigation bar includes the AWS logo, "Services", a search icon, a code icon, a notification bell, and a "More" dropdown. The breadcrumb trail shows "Developer Tools" > "..." > "Create connection".

## Create a connection [Info](#)

### Create GitHub App connection [Info](#)

Connection name

Deployment Tutorial

► Tags - optional

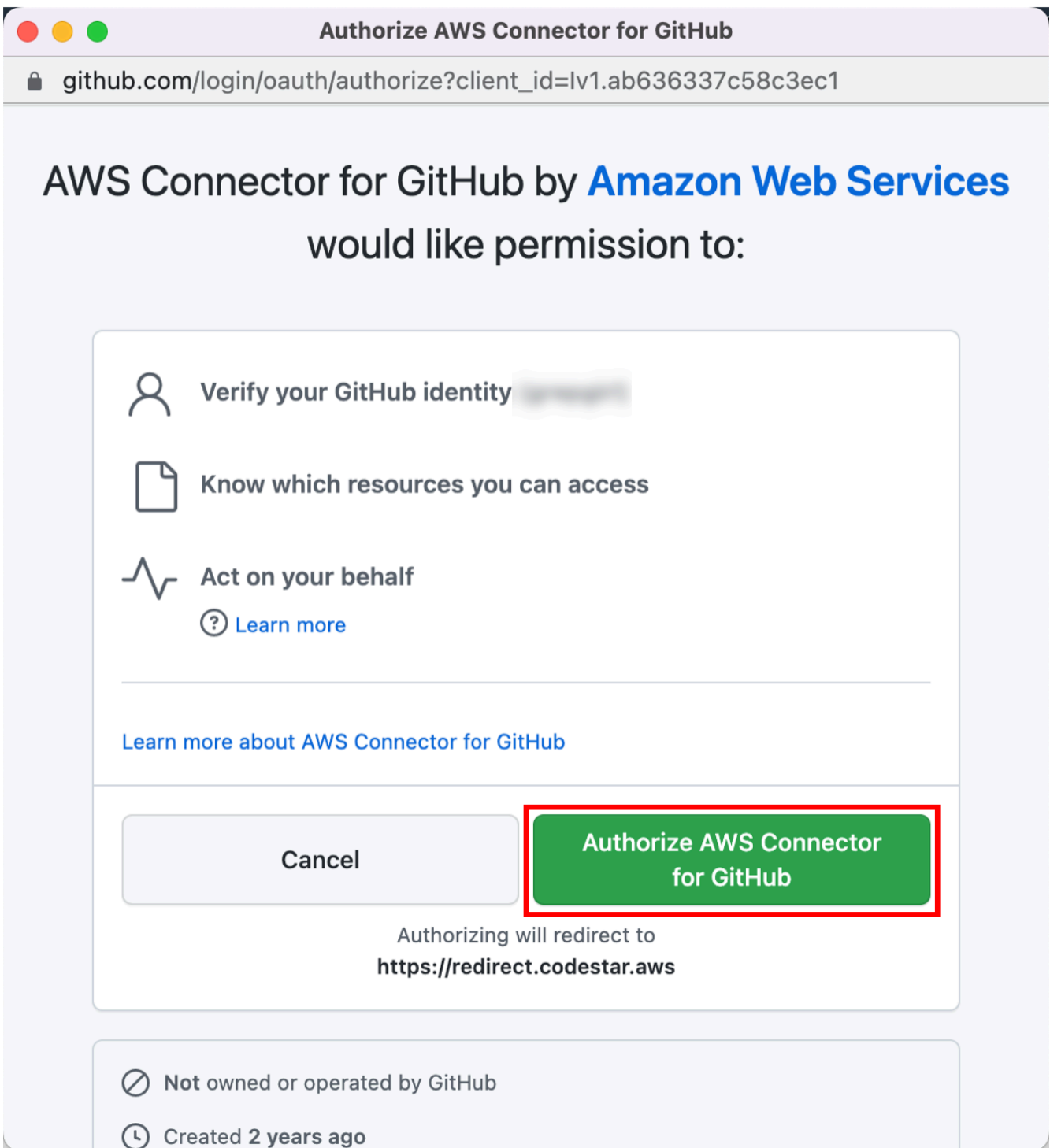
**Connect to GitHub**

Feedback   Change language [↗](#)   Privacy   Terms   Cookie preferences

© 2022, Amazon Web Services, Inc. or its affiliates.

### 3. Grant permissions

Select **Authorize AWS Connector for GitHub**.



#### 4. Install a new app

When redirected back to the connection screen, choose **Install a new app**.

## 5. Install the forked repository

On the **Install AWS Connector for GitHub** page, choose **Only select repositories** and select the **aws-codepipeline-s3-codedeploy-linux** repository forked in the previous step.

Choose **Install**.

Installing AWS Connector for GitHub

github.com/apps/aws-connector-for-github/installations/new/permissions?target...

## Install AWS Connector for GitHub


Install on your personal account [redacted]

☐ **All repositories**  
This applies to all current *and* future repositories.

☒ **Only select repositories**  
Select at least one repository.

Select repositories ▾

Search for a repository

with  [redacted] /aws-codepipeline-s3-codedeploy-linux

Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walkthrough tutorial.

✓ <http://docs.aws.amazon.com/codepipeline/latest/userguide/getting-started-w.html>

✓

✓ **Read and write access to administration, code, and pull requests**

**Install** Cancel

Next: you'll be directed to the GitHub App's site to complete setup.

## 6. Connect to GitHub



Once redirected back to the **Connect to GitHub** page, choose **Connect**.

The screenshot shows the AWS Developer Tools console interface. The browser address bar displays the URL: `us-east-1.console.aws.amazon.com/codesuite/settings/connections/create/github...`. The page title is "Settings - AWS Developer Tools". The navigation bar includes the AWS logo, "Services", a search icon, a code icon, a bell icon, and a "More" dropdown. The breadcrumb trail shows "Developer Tools > ... > Create connection". The main heading is "Connect to GitHub". Below this is the "GitHub connection settings" section, which includes an "Info" link. The "Connection name" field is filled with "Deployment Tutorial". The "GitHub Apps" section explains that GitHub Apps create a link for the connection and provides a search bar with the value "28929444" and an "Install a new app" button. At the bottom right, the "Connect" button is highlighted with a red rectangle. The footer contains links for "Feedback", "Change language", "Privacy", "Terms", and "Cookie preferences", along with the copyright notice "© 2022, Amazon Web Services, Inc. or its affiliates."

7. Specify a repository and branch

The **Add source** page will be updated to reflect GitHub is ready to connect. Specify the repository and branch:

**Repository name:** In the dropdown list, select the GitHub repository you want to use as the source location for your pipeline. Select the forked repository in your GitHub account named `aws-codepipeline-s3-codedeploy-linux`.

**Branch name:** In the dropdown list, select the branch you want to use, **master**.

**Output artifact format:** Select **CodePipeline default**.

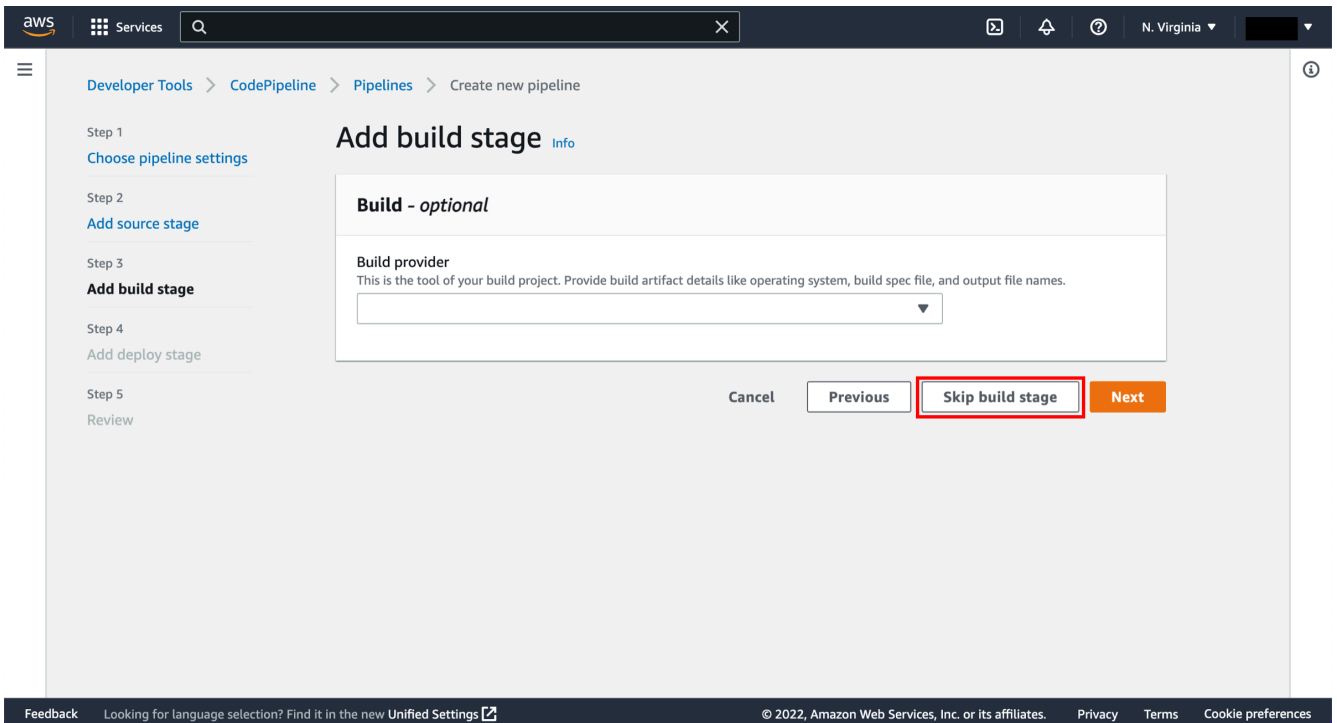
Choose **Next**.

The screenshot shows the AWS CodePipeline console interface for configuring a new pipeline. The 'Connection' section indicates that the GitHub connection is ready. The 'Repository name' field is populated with 'grepgirl/aws-codepipeline-s3-codedeploy-linux'. The 'Branch name' field is set to 'master'. The 'Change detection options' section has the checkbox 'Start the pipeline on source code change' checked. The 'Output artifact format' section has 'CodePipeline default' selected. The 'Next' button is highlighted in orange, indicating the next step in the process.

## 8. Skip build stage

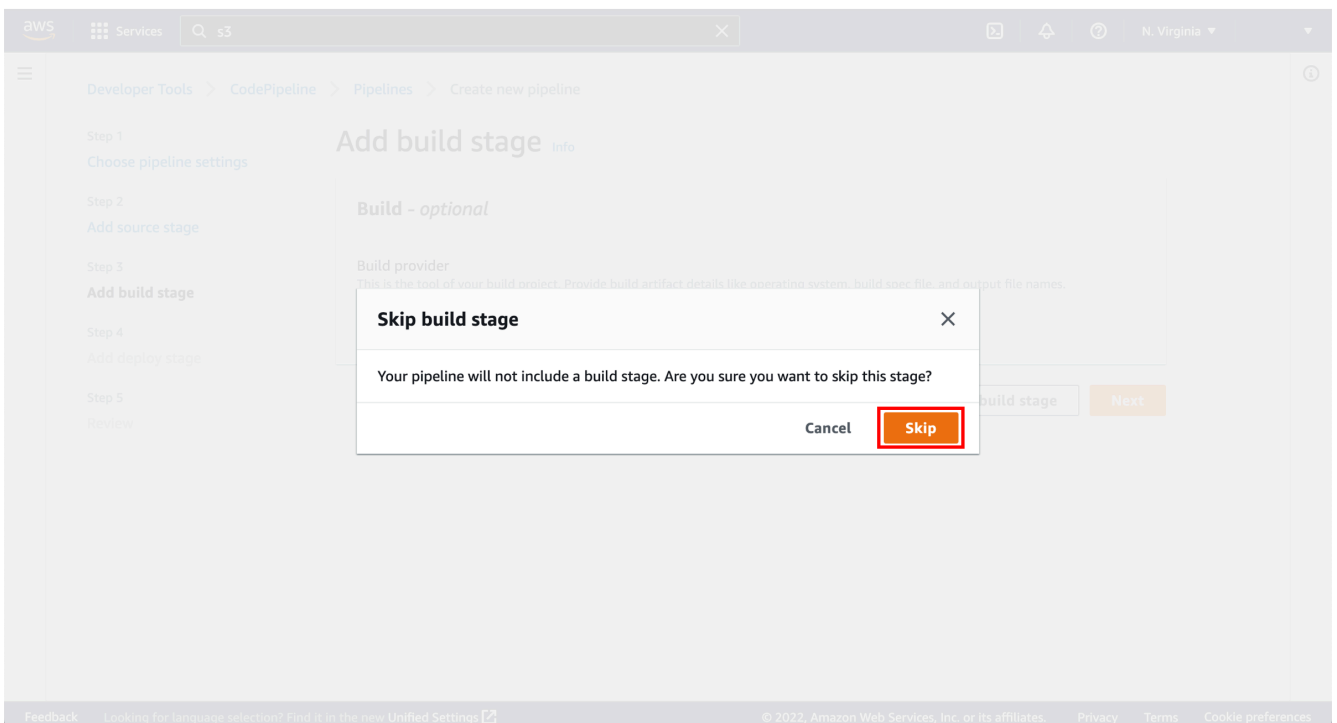
A true continuous deployment pipeline requires a build stage, where code is compiled and unit tested. CodePipeline lets you plug your preferred build provider into your pipeline. However, in this tutorial you will skip the build stage.

In **Step 3: Add build stage**, choose **Skip build stage**.



## 9. Choose Skip

In the confirmation dialog, select **Skip**.



## 10. Configure deploy stage

In the **Step 4: Add deploy stage** page:

**Deploy provider:** Select **AWS Elastic Beanstalk**.

**Region:** Retain the default region.

**Application name:** Select **Deployment Tutorial**.

**Environment name:** Select **Deploymenttutorial-env**.

Click **Next**.

Continue to **Activate your pipeline to deploy your code**.

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1  
Choose pipeline settings

Step 2  
Add source stage

Step 3  
Add build stage

Step 4  
**Add deploy stage**

Step 5  
Review

### Add deploy stage Info

**You cannot skip this stage**  
Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.

#### Deploy

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

AWS Elastic Beanstalk

**Region**  
US East (N. Virginia)

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

Deployment Tutorial

**Environment name**  
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.

Deploymenttutorial-env

Cancel Previous **Next**

Feedback Looking for language selection? Find it in the new Unified Settings [\[?\]](#) © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Amazon S3

### 1. Add source

Select **Amazon S3** for the **Source provider**, select the Amazon S3 bucket you created, and then enter the S3 object key for the file uploaded, for example: **aws-codepipeline-s3-aws-codedeploy\_linux.zip**.

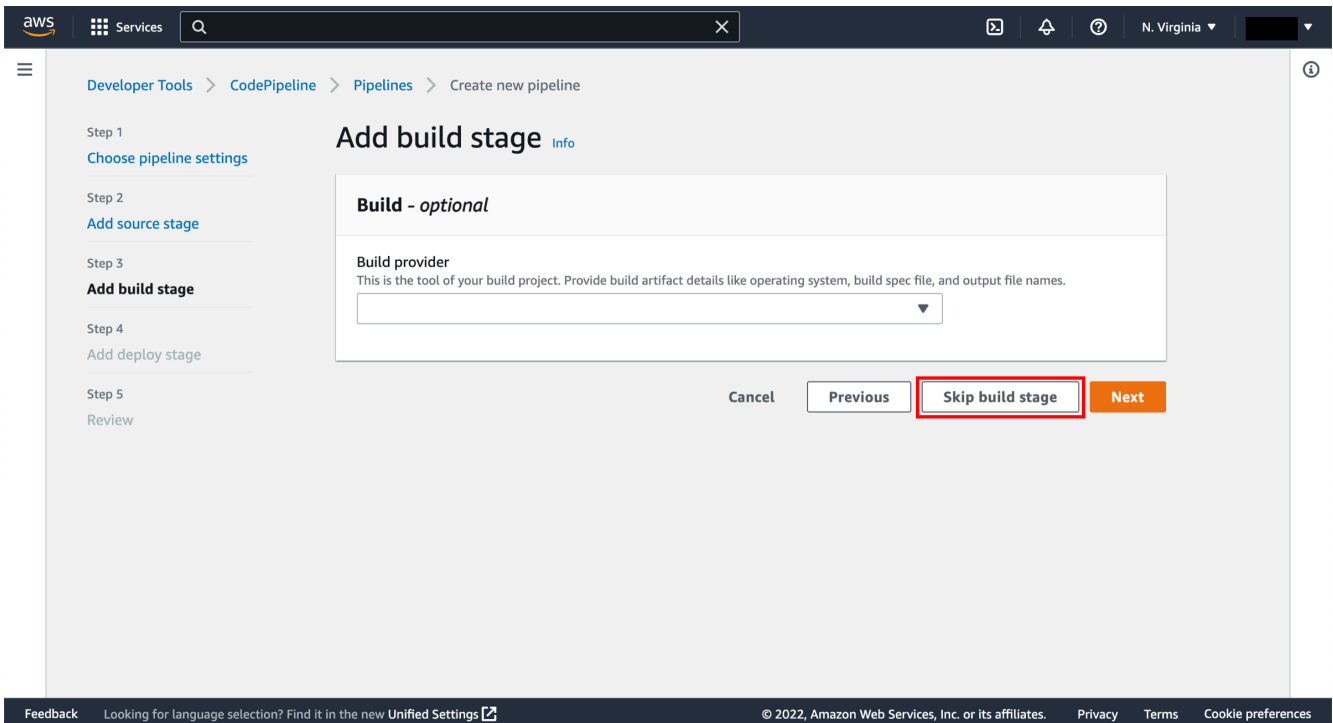
## Choose Next.

The screenshot shows the AWS CodePipeline console interface for adding a source stage. The breadcrumb navigation indicates the path: Developer Tools > CodePipeline > Pipelines > Create new pipeline. On the left, a sidebar lists the steps: Step 1 (Choose pipeline settings), 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 'Add source stage' with an 'Info' link. It contains a 'Source' section with the following fields: 'Source provider' (Amazon S3), 'Bucket' (awscodepipeline-demobucket-8302022), and 'S3 object key' (aws-codepipeline-s3-aws-codedeploy\_linux.zip). Below these fields is a 'Change detection options' section with two radio buttons: 'Amazon CloudWatch Events (recommended)' (selected) and 'AWS CodePipeline'. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next' (highlighted in red).

## 2. Skip build stage

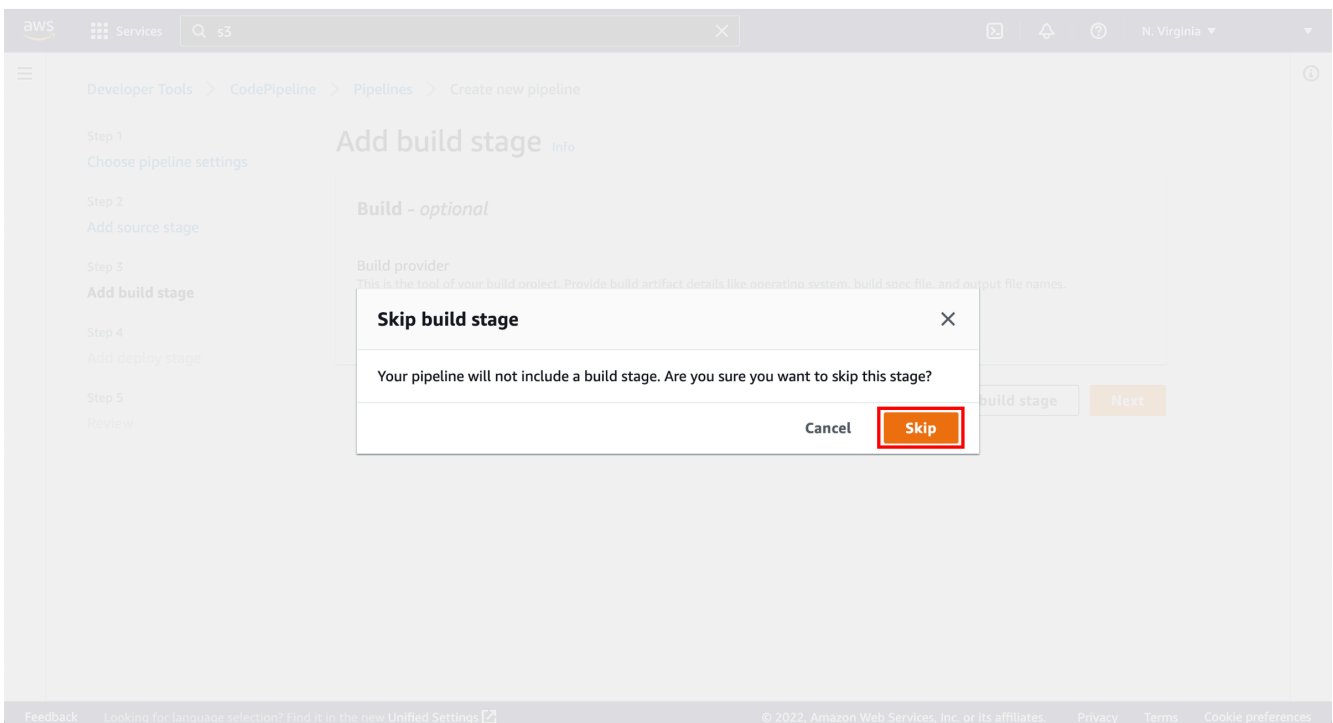
A true continuous deployment pipeline requires a build stage, where code is compiled and unit tested. CodePipeline lets you plug your preferred build provider into your pipeline. However, in this tutorial you will skip the build stage.

**In Step 3: Add build stage, choose Skip build stage.**



### 3. Choose Skip

In the confirmation dialog, select **Skip**.



### 4. Configure deploy stage

In the **Step 4: Add deploy stage** page:

**Deploy provider:** Select **AWS Elastic Beanstalk**.

**Region:** Retain the default region.

**Application name:** Select **Deployment Tutorial**.

**Environment name:** Select **Deploymenttutorial-env**.

Click **Next**.

Continue to **Activate your pipeline to deploy your code**.

The screenshot shows the AWS CodePipeline console interface for adding a new deployment stage. The breadcrumb navigation indicates the path: Developer Tools > CodePipeline > Pipelines > Create new pipeline. The left sidebar shows the steps of the pipeline: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), Step 5 (Review), and a final Review step. The main content area is titled 'Add deploy stage' with an 'Info' link. A blue information box states: 'You cannot skip this stage. Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.' Below this, the 'Deploy' stage configuration is shown. The 'Deploy provider' dropdown is set to 'AWS Elastic Beanstalk'. The 'Region' dropdown is set to 'US East (N. Virginia)'. The 'Application name' dropdown is set to 'Deployment Tutorial'. The 'Environment name' dropdown is set to 'Deploymenttutorial-env'. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next' (which is highlighted in orange).

## AWS CodeCommit

### 1. Add source

Select **AWS CodeCommit** for the **Source provider**.

**Repository name:** In the dropdown list, choose the **PipelineRepo** repository you created to use as the source location for your pipeline.

**Branch name:** In the dropdown list, choose the branch you want to use, **main**.

**Output artifact format:** Choose **CodePipeline default**.

Choose **Next**.

The screenshot shows the 'Add source stage' configuration page in the AWS CodePipeline console. The left sidebar shows the pipeline steps: Step 1 (Choose pipeline settings), 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 'Add source stage' and includes the following sections:

- Source provider:** A dropdown menu with 'AWS CodeCommit' selected.
- Repository name:** A text input field with 'PipelineRepo' entered.
- Branch name:** A text input field with 'main' entered.
- Change detection options:** Two radio button options. 'Amazon CloudWatch Events (recommended)' is selected, with a description: 'Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs'. The other option is 'AWS CodePipeline' with the description: 'Use AWS CodePipeline to check periodically for changes'.
- Output artifact format:** Two radio button options. 'CodePipeline default' is selected, with a description: 'AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include git metadata about the repository.' The other option is 'Full clone' with the description: 'AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full git clone. Only supported for AWS CodeBuild actions.'

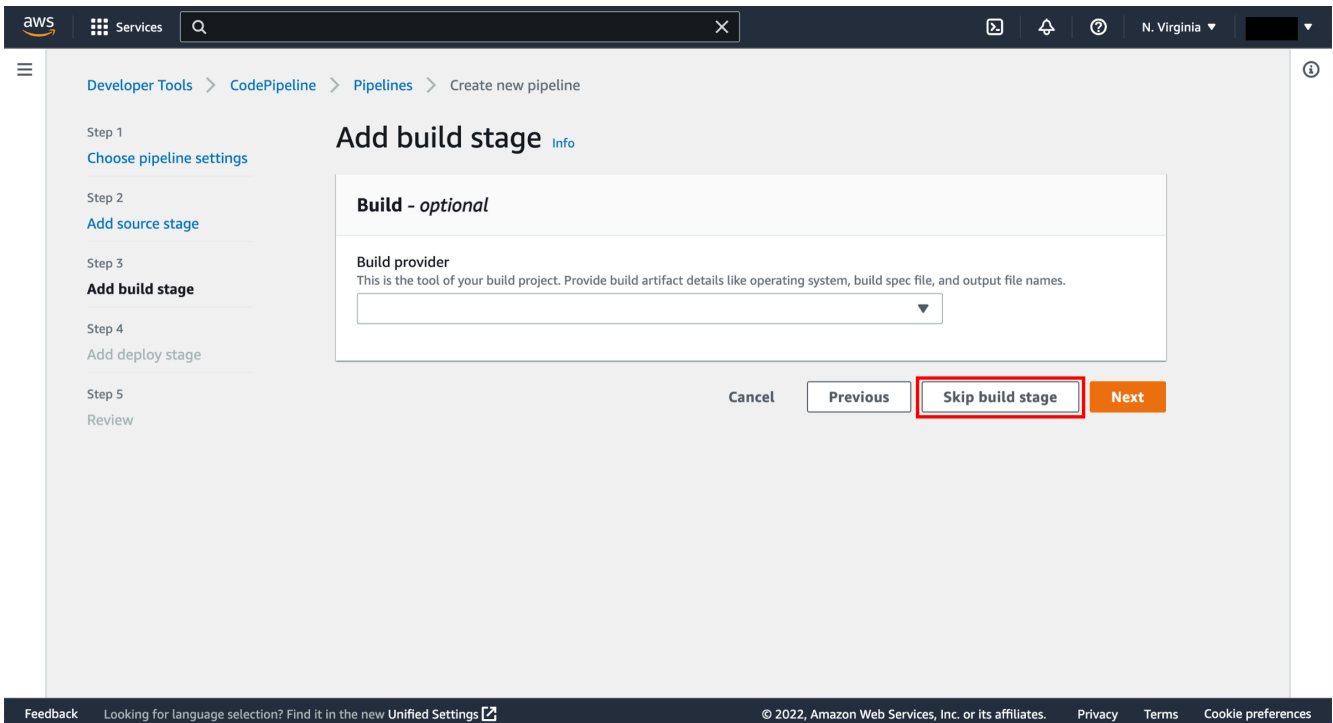
At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'. The 'Next' button is highlighted with a red border.

## 2. Skip build stage

A true continuous deployment pipeline requires a build stage, where code is compiled and unit tested. CodePipeline lets you plug your preferred build provider into your pipeline. However, in this tutorial you will skip the build stage.

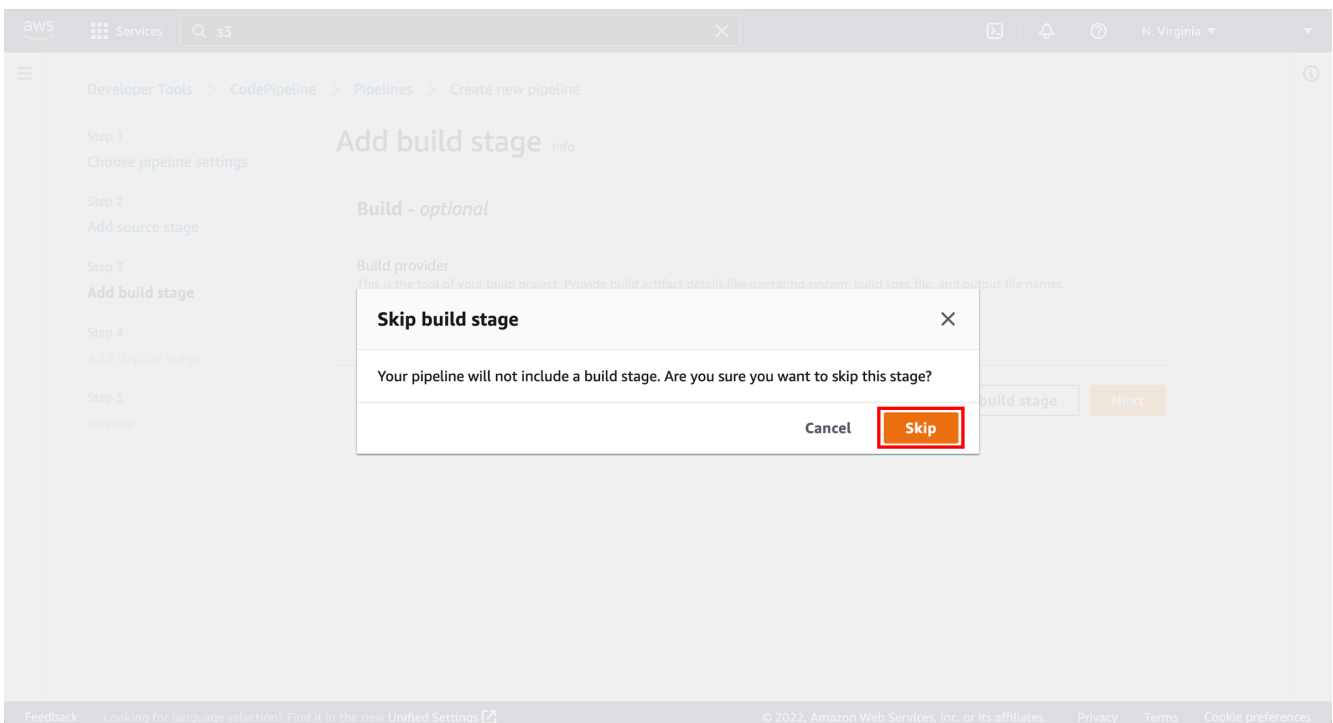
In **Step 3: Add build stage**, choose **Skip build stage**.





### 3. Choose Skip

In the confirmation dialog, select **Skip**.



### 4. Configure deploy stage

In the **Step 4: Add deploy stage** page:

**Deploy provider:** Select **AWS Elastic Beanstalk**.

**Region:** Retain the default region.

**Application name:** Select **Deployment Tutorial**.

**Environment name:** Select **Deploymenttutorial-env**.

Click **Next**.

Continue to **Activate your pipeline to deploy your code**.

The screenshot shows the AWS CodePipeline console interface. The breadcrumb navigation is 'Developer Tools > CodePipeline > Pipelines > Create new pipeline'. The left sidebar shows the steps of the pipeline creation process: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), Step 5 (Review), and a final Review step. The main content area is titled 'Add deploy stage' with an 'Info' link. A blue information box states: 'You cannot skip this stage. Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.' Below this, the 'Deploy' configuration section is shown. It includes a 'Deploy provider' dropdown set to 'AWS Elastic Beanstalk', a 'Region' dropdown set to 'US East (N. Virginia)', an 'Application name' search box containing 'Deployment Tutorial', and an 'Environment name' search box containing 'Deploymenttutorial-env'. At the bottom right of the configuration section are 'Cancel', 'Previous', and 'Next' buttons, with 'Next' being highlighted in orange.

## Step 4: Activate your pipeline to deploy your code

In this step, you will launch your pipeline. Once your pipeline has been created, it will start to run automatically. First, it detects the sample app code in your source location, bundles up the files, and then moves them to the second stage that you defined. During this stage, it passes the code to Elastic Beanstalk, which contains the EC2 instance that will host your code. Elastic Beanstalk handles deploying the code to the EC2 instance.

## 1. Review configuration and create pipeline

In the **Step 5: Review** page, review the information and choose **Create pipeline**.

The screenshot shows the AWS CodePipeline console interface. At the top, there's a navigation bar with the AWS logo, 'Services' menu, a search bar, and regional settings for 'N. Virginia'. Below the navigation bar, a breadcrumb trail reads 'Developer Tools > CodePipeline > Pipelines > Create new pipeline'. On the left, a sidebar lists five steps: 'Step 1: Choose pipeline settings', 'Step 2: Add source stage', 'Step 3: Add build stage', 'Step 4: Add deploy stage', and 'Step 5: Review'. The 'Review' step is currently selected and highlighted.

The main content area is titled 'Review' with an 'Info' link. It displays the configuration for the pipeline across four steps:

- Step 1: Choose pipeline settings**
  - Pipeline settings**
    - Pipeline name: DemoPipeline
    - Artifact location: A new Amazon S3 bucket will be created as the default artifact store for your pipeline
    - Service role name: AWSCodePipelineServiceRole-us-east-1-DemoPipeline
- Step 2: Add source stage**
  - Source action provider**
    - Source action provider: Amazon S3
    - PollForSourceChanges: false
    - S3Bucket: awscodepipeline-demobucket-8302022
    - S3ObjectKey: aws-codepipeline-s3-aws-codedeploy\_linux.zip
- Step 3: Add build stage**
  - Build action provider**
    - Build stage: No build
- Step 4: Add deploy stage**
  - Deploy action provider**
    - Deploy action provider: AWS Elastic Beanstalk
    - ApplicationName: Deployment Tutorial
    - EnvironmentName: Deploymenttutorial-env

At the bottom right of the main content area, there are three buttons: 'Cancel', 'Previous', and 'Create pipeline' (which is highlighted with a red border).

The footer of the console shows a 'Feedback' link, a language selection prompt, copyright information for 2022, and links for 'Privacy', 'Terms', and 'Cookie preferences'.

## 2. Monitor the pipeline status

After your pipeline is created, the pipeline status page appears and the pipeline automatically starts to run. You can view progress as well as success and failure messages as the pipeline performs each action.

To verify your pipeline ran successfully, monitor the progress of the pipeline as it moves through each stage. The status of each stage will change from No executions yet to **In progress**, and then to either **Succeeded** or **Failed**. The pipeline should complete the first run within a few minutes.

The screenshot displays the AWS CodePipeline console interface. At the top, a green banner indicates 'Success' and 'Congratulations! The pipeline DemoPipeline has been created.' Below this, the breadcrumb navigation shows 'Developer Tools > CodePipeline > Pipelines > DemoPipeline'. The main content area shows the 'DemoPipeline' with a 'Notify' dropdown, 'Edit', 'Stop execution', 'Clone pipeline', and 'Release change' buttons. The pipeline consists of two stages: 'Source' and 'Deploy'. Both stages are marked as 'Succeeded'. The 'Source' stage is linked to 'Amazon S3' and completed '3 minutes ago'. The 'Deploy' stage is linked to 'AWS Elastic Beanstalk' and completed '2 minutes ago'. A 'Disable transition' button is visible between the stages. The left sidebar shows the 'CodePipeline' section with options like 'Source', 'Artifacts', 'Build', 'Deploy', 'Pipeline', and 'Settings'. The bottom of the console features a footer with 'Feedback', a language selection link, copyright information, and links to 'Privacy', 'Terms', and 'Cookie preferences'.

### 3. Select Elastic Beanstalk

In the status area for the Beta stage, select **AWS Elastic Beanstalk**.

The screenshot displays the AWS CodePipeline console interface. On the left, a sidebar menu shows 'Developer Tools' with 'CodePipeline' selected. Below it, a list of pipeline stages is visible: Source (CodeCommit), Artifacts (CodeArtifact), Build (CodeBuild), Deploy (CodeDeploy), and Pipeline (CodePipeline). The 'Pipeline' section is expanded, showing 'Getting started', 'Pipelines', 'Pipeline' (highlighted), 'History', and 'Settings'. The main panel shows the 'DemoPipeline' execution details. It includes a 'Source' stage that succeeded, with a 'Source' action using 'Amazon S3' as the provider. Below this, a 'Deploy' stage is shown, also succeeded, with a 'Deploy' action using 'AWS Elastic Beanstalk' as the provider. The 'AWS Elastic Beanstalk' link is highlighted with a red box. The pipeline execution ID is 'b3115f96-4ec4-4c70-90ce-228bb7747b82'. The bottom of the console shows a footer with 'Feedback', 'Looking for language selection? Find it in the new Unified Settings', and copyright information for 2022.

#### 4. Select the environment

The AWS Elastic Beanstalk console opens with the details of the deployment.

Select the environment you created earlier, called **Default-Environment Deploymenttutorial-env**.

The screenshot shows the AWS Elastic Beanstalk console. On the left, the 'Elastic Beanstalk' sidebar is visible with options for Environments, Applications, and Change history. Under 'Deployment Tutorial', there are links for Application versions, Saved configurations, and Recent environments (Deploymenttutorial-env). The main content area displays 'Application 'Deployment Tutorial' environments'. A table lists the environment details:

Environment name	Health	Date created	Last modified	URL	Running versions
Deploymenttutorial-env	Ok	2022-08-30 09:02:06 UTC-0700	2022-09-05 10:35:58 UTC-0700	Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com	code-pipeline-16623992711 Tzv1VxUki0BP4Y44gSQoCG7

The 'Deploymenttutorial-env' entry is highlighted with a red box. The top of the console shows a notification about AWS Graviton support and a search bar. The bottom of the console has a footer with 'Feedback', 'Looking for language selection? Find it in the new Unified Settings', and copyright information.

## 5. Select the URL of the sample website

Select the URL to view the sample website you deployed.

A webpage with a congratulations message indicating you successfully created a pipeline from your source to Amazon EC2 will open.

The screenshot shows the AWS Elastic Beanstalk console. At the top, there's a navigation bar with the AWS logo, 'Services' link, a search bar, and a region dropdown set to 'N. Virginia'. Below the navigation bar, the left sidebar shows the 'Elastic Beanstalk' menu with options like 'Environments', 'Applications', and 'Change history'. The main content area shows the 'Deploymenttutorial-env' environment. A blue banner at the top of the main area mentions 'AWS Graviton now supported'. Below this, the breadcrumb 'Elastic Beanstalk > Environments > Deploymenttutorial-env' is visible. The environment details section shows the URL 'Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com' highlighted with a red box. Below the URL, the application name is 'Deployment Tutorial'. The 'Health' section shows a green checkmark and 'Ok' status. The 'Running version' section shows the version 'code-pipeline-1662399271185-Tzv1VxUkl0BP4Y44gSQoCG78YgS8R58I' and an 'Upload and deploy' button. The 'Platform' section shows the PHP logo and 'PHP 8.1 running on 64bit Amazon Linux 2/3.4.0', with a warning 'Different version recommended' and a 'Change' button. At the bottom, there's a 'Recent events' section and a footer with 'Feedback', 'Looking for language selection? Find it in the new Unified Settings', and copyright information.

## Step 5: Commit a change and then update your app

In this step, you will revise the sample code and commit the change to your repository. CodePipeline will detect your updated sample code and then automatically initiate deploying it to your EC2 instance by way of Elastic Beanstalk.

Note that the sample web page you deployed refers to AWS CodeDeploy, a service that automates code deployments. In CodePipeline, CodeDeploy is an alternative to using Elastic Beanstalk for deployment actions. Let's update the sample code so that it correctly states that you deployed the sample using Elastic Beanstalk.

Choose the appropriate tab based on the code source you used.

### GitHub

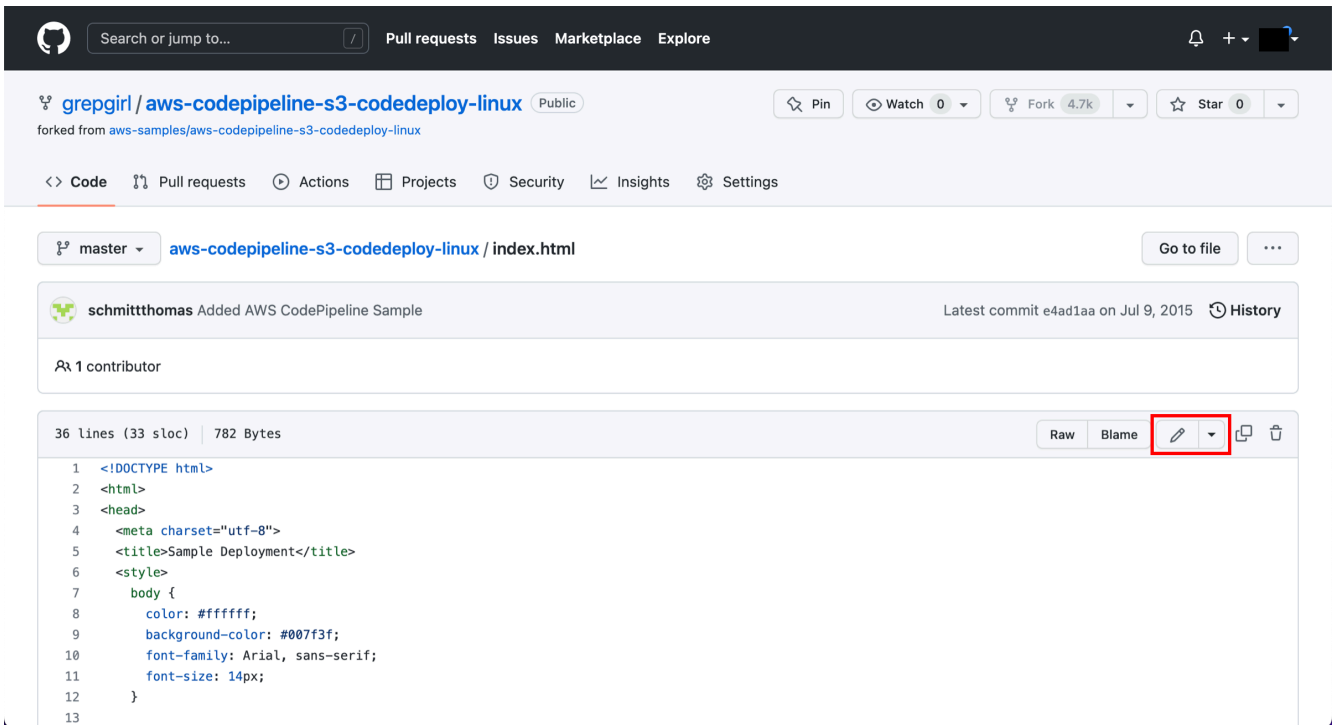
#### 1. Edit the code

Visit your own copy of the repository that you forked in GitHub.

Open **index.html**.

Select the **Edit** icon.

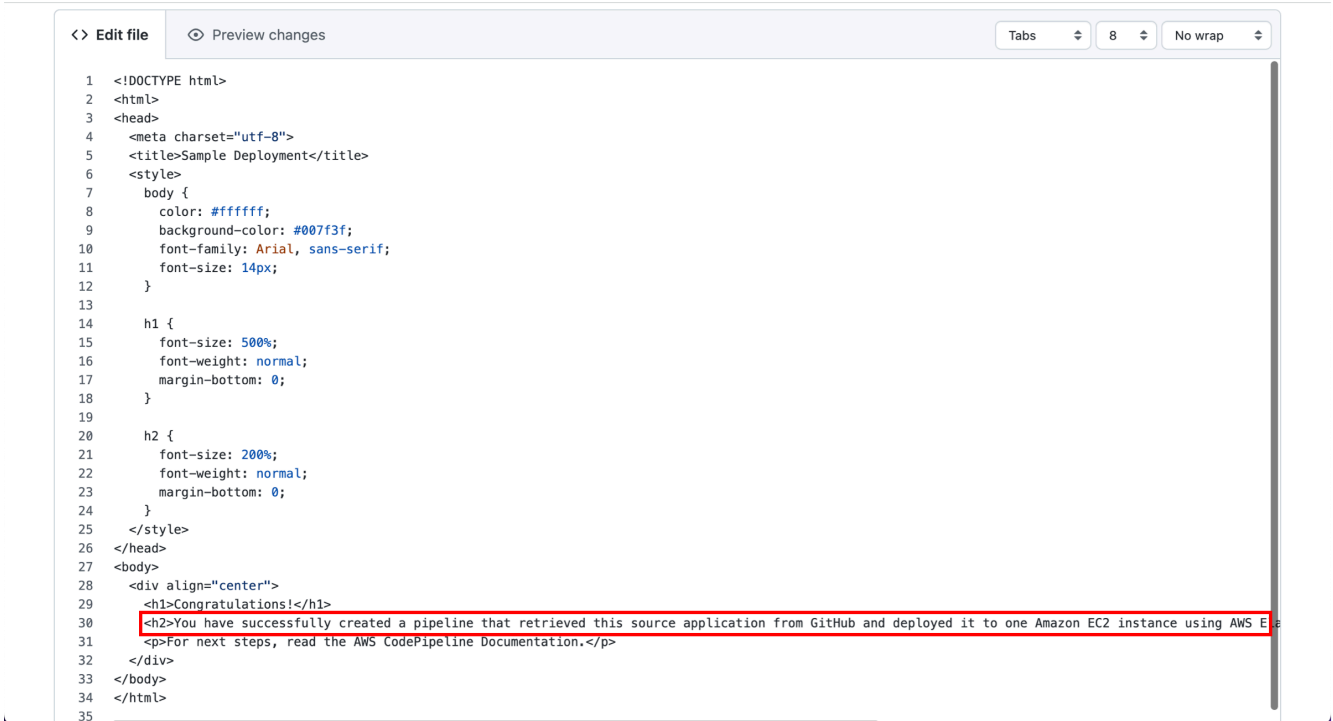




## 2. Insert text

Update the webpage by copying and pasting the following text on line 30:

You have successfully created a pipeline that retrieved this source application from GitHub and deployed it to one Amazon EC2 instance using AWS Elastic Beanstalk. You're one step closer to practicing continuous deployment!

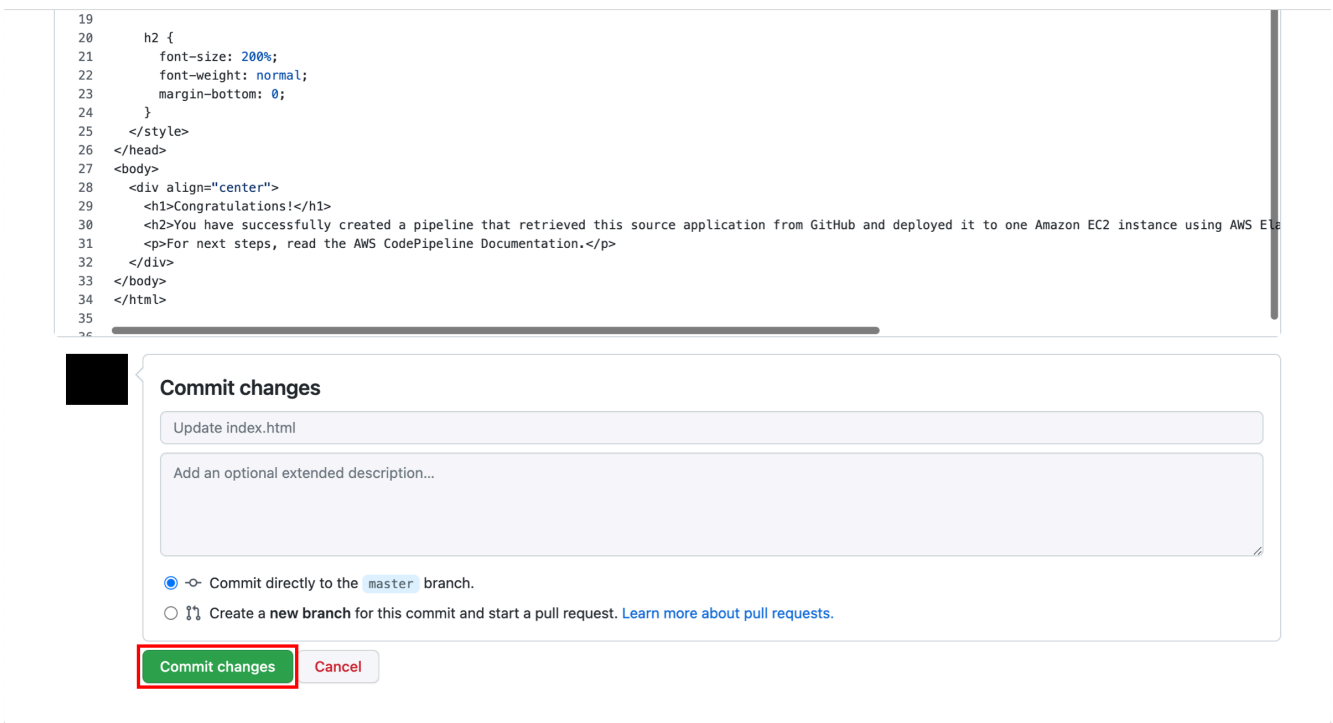


```
<> Edit file    Preview changes    Tabs 8 No wrap
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Sample Deployment</title>
6 </head>
7 <body>
8   <div>
9     <h1>Congratulations!</h1>
10    <p>You have successfully created a pipeline that retrieved this source application from GitHub and deployed it to one Amazon EC2 instance using AWS Elastic Beanstalk.</p>
11    <p>For next steps, read the AWS CodePipeline Documentation.</p>
12  </div>
13 </body>
14 </html>
```

### 3. Commit the change

Commit the change to your repository.

Then, go to View the page you updated with GitHub.



```
19
20   h2 {
21     font-size: 200%;
22     font-weight: normal;
23     margin-bottom: 0;
24   }
25 </style>
26 </head>
27 <body>
28   <div align="center">
29     <h1>Congratulations!</h1>
30     <h2>You have successfully created a pipeline that retrieved this source application from GitHub and deployed it to one Amazon EC2 instance using AWS Elastic Beanstalk.</h2>
31     <p>For next steps, read the AWS CodePipeline Documentation.</p>
32   </div>
33 </body>
34 </html>
```

**Commit changes**  
Update index.html  
Add an optional extended description...  
☒ Commit directly to the master branch.  
☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)  
**Commit changes** Cancel

## Amazon S3

### 1. Edit the code

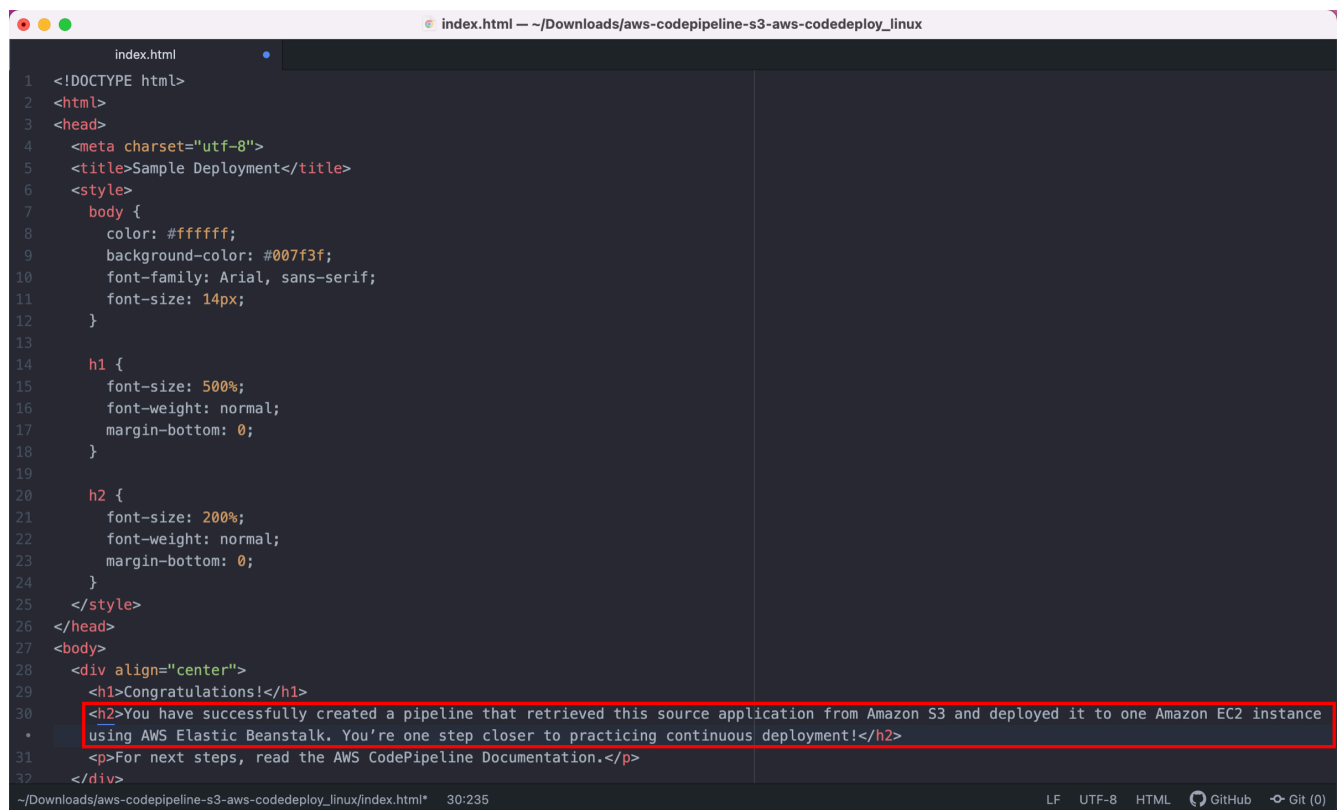
On your desktop, visit the zip file you downloaded called **aws-codepipeline-s3-aws-codedeploy\_linux.zip**.

Edit the sample web app code:

- Extract index.html from the zip file and open it using your preferred text editor.
- Update the header text that comes after **Congratulations!** so that it reads:

You have successfully created a pipeline that retrieved this source application from Amazon S3 and deployed it to one Amazon EC2 instance using AWS Elastic Beanstalk. You're one step closer to practicing continuous deployment!

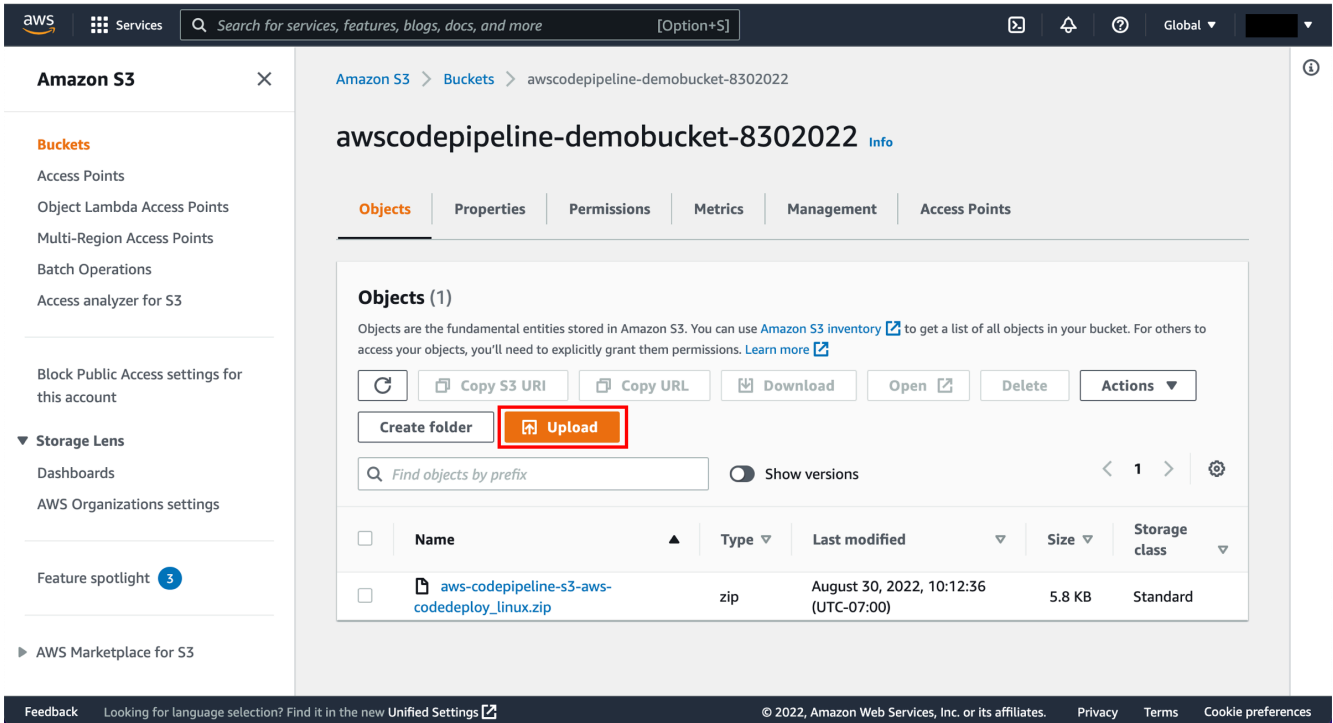
- Copy the updated index.html file back into **aws-codepipeline-s3-aws-codedeploy\_linux.zip** and replace the older version of index.html.



```
index.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Sample Deployment</title>
6   <style>
7     body {
8       color: #ffffff;
9       background-color: #007f3f;
10      font-family: Arial, sans-serif;
11      font-size: 14px;
12    }
13
14    h1 {
15      font-size: 500%;
16      font-weight: normal;
17      margin-bottom: 0;
18    }
19
20    h2 {
21      font-size: 200%;
22      font-weight: normal;
23      margin-bottom: 0;
24    }
25  </style>
26 </head>
27 <body>
28   <div align="center">
29     <h1>Congratulations!</h1>
30     <h2>You have successfully created a pipeline that retrieved this source application from Amazon S3 and deployed it to one Amazon EC2 instance
31     using AWS Elastic Beanstalk. You're one step closer to practicing continuous deployment!</h2>
32   </div>
33   <p>For next steps, read the AWS CodePipeline Documentation.</p>
34 </body>
35 </html>
```

### 2. Upload the file to your bucket

Return to the S3 bucket that you created earlier and select **Upload**.



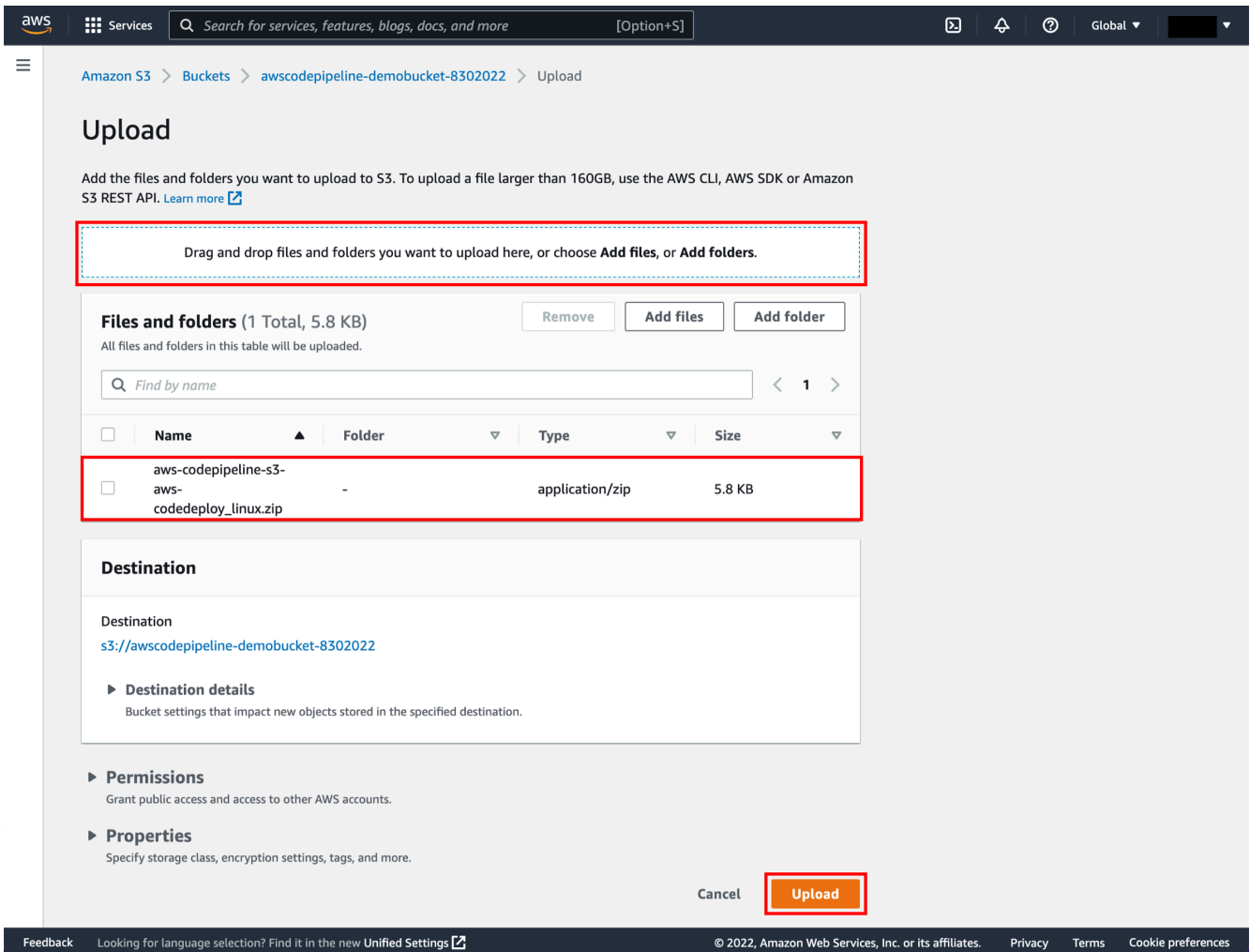
### 3. Upload the file to your bucket

Select **Add files** to upload the updated **aws-codepipeline-s3-aws-codedeploy\_linux.zip** file or drag and drop the file. Then choose **Upload**.

#### Note

Because you enabled versioning when you first created the S3 bucket, S3 will save a copy of every version of your files.

Then, go to View the page you updated in Amazon S3.



## AWS CodeCommit

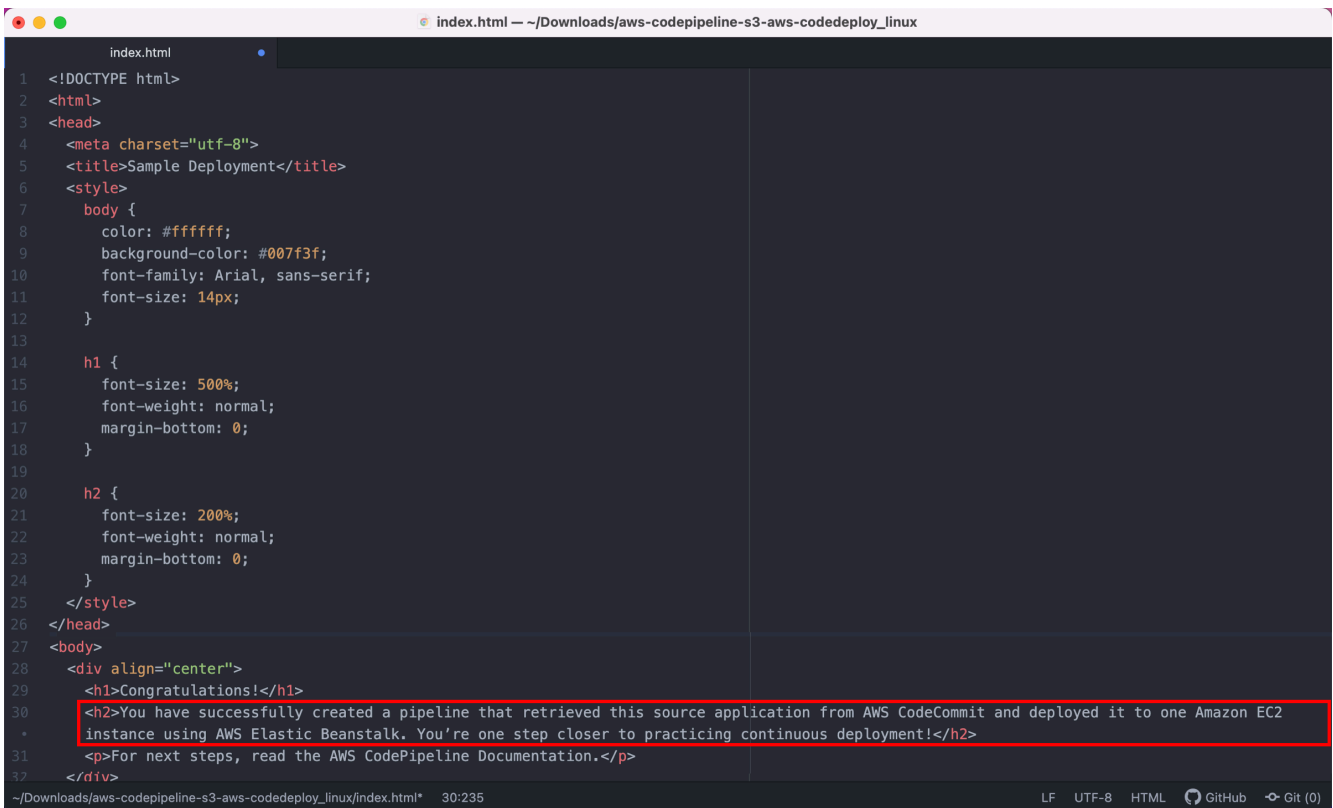
### 1. Edit the code

On your desktop, visit the zip file you downloaded called **aws-codepipeline-s3-aws-codedeploy\_linux.zip**.

Edit the sample web app code:

- Extract index.html from the zip file and open it using your preferred text editor.
- Update the header text that comes after **Congratulations!** so that it reads:

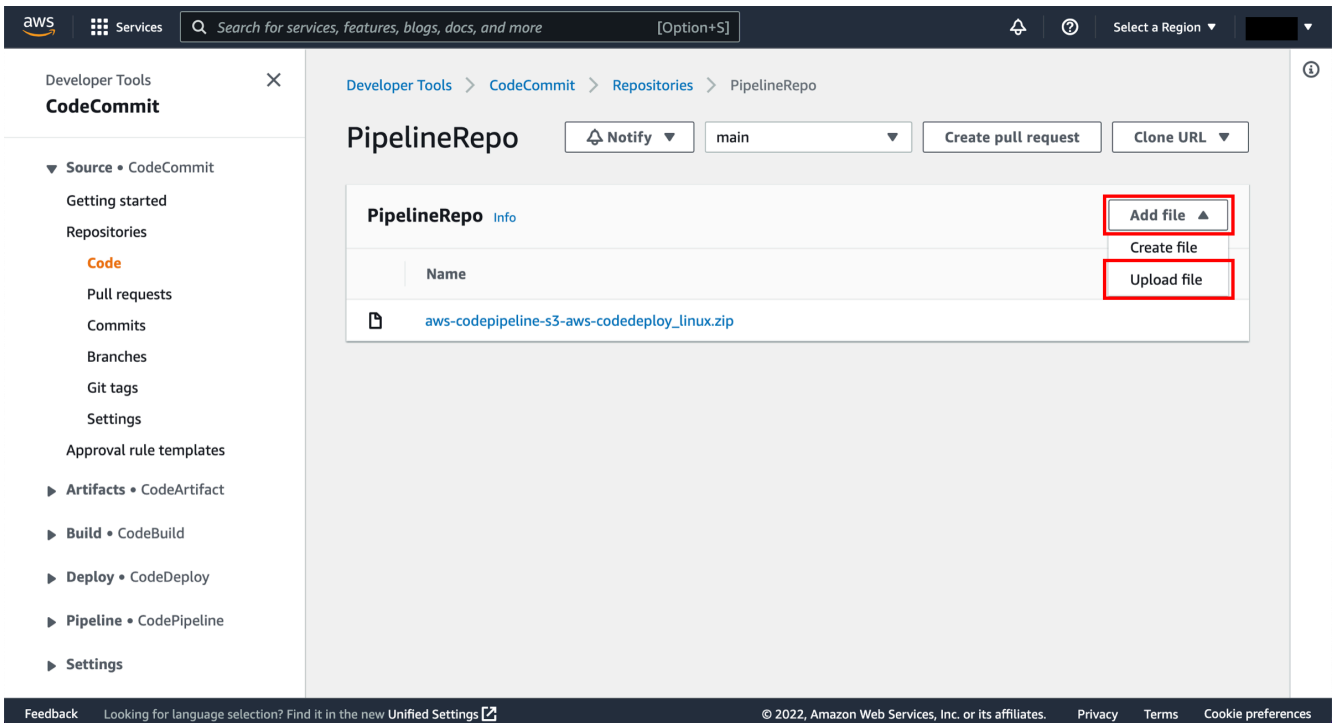
You have successfully created a pipeline that retrieved this source application from AWS CodeCommit and deployed it to one Amazon EC2 instance using AWS Elastic Beanstalk. You're one step closer to practicing continuous deployment!



```
index.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Sample Deployment</title>
6   <style>
7     body {
8       color: #ffffff;
9       background-color: #007f3f;
10      font-family: Arial, sans-serif;
11      font-size: 14px;
12    }
13
14    h1 {
15      font-size: 500%;
16      font-weight: normal;
17      margin-bottom: 0;
18    }
19
20    h2 {
21      font-size: 200%;
22      font-weight: normal;
23      margin-bottom: 0;
24    }
25  </style>
26 </head>
27 <body>
28   <div align="center">
29     <h1>Congratulations!</h1>
30     <h2>You have successfully created a pipeline that retrieved this source application from AWS CodeCommit and deployed it to one Amazon EC2
    * instance using AWS Elastic Beanstalk. You're one step closer to practicing continuous deployment!</h2>
31     <p>For next steps, read the AWS CodePipeline Documentation.</p>
32   </div>
```

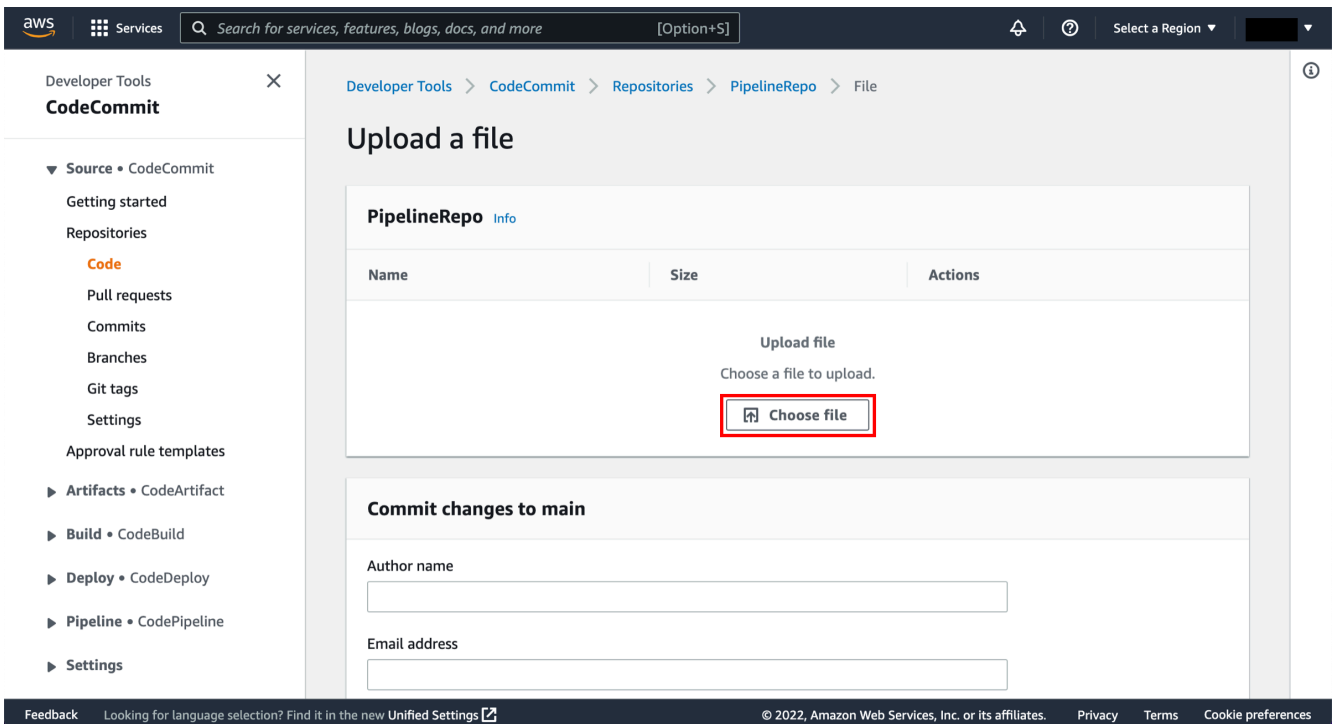
## 2. Upload the file

From the **CodeCommit PipelineRepo** page, choose **Add file** and select **Upload file**.



### 3. Upload the file

On the **Upload a file** page, choose the **Choose file** button and select the updated **aws-codepipeline-s3-aws-codedeploy\_linux.zip** file.



### 4. Commit changes

## Enter an Author name and Email address, then choose **Commit changes**.

Developer Tools

CodeCommit

Source • CodeCommit

Getting started

Repositories

Code

Pull requests

Commits

Branches

Git tags

Settings

Approval rule templates

Artifacts • CodeArtifact

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Settings

Developer Tools > CodeCommit > Repositories > PipelineRepo > File

### Upload a file

**PipelineRepo** Info

Name	Size	Actions
aws-codepipeline-s3-aws-codedeploy_linux.zip	6 KB	<button>Remove file</button>

**Commit changes to main**

File: PipelineRepo/aws-codepipeline-s3-aws-codedeploy\_linux.zip

Author name

AWS User

Email address

aws-user@amazon.com

Commit message - optional

A default commit message will be used if you do not provide one.

Cancel Commit changes

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Step 6: View the page you updated

In this step, you will view the page you updated.

Choose the appropriate tab based on the code source you used.

### GitHub

#### 1. Choose Elastic Beanstalk

Return to your pipeline in the CodePipeline console. In a few minutes, you should see the Source change to blue, indicating that the pipeline has detected the changes you made to your source repository. Once this occurs, it will automatically move the updated code to Elastic Beanstalk.

After the pipeline status displays **Succeeded**, in the status area for the Beta stage, choose **AWS Elastic Beanstalk**.



The screenshot shows the AWS CodePipeline console for a pipeline named "DemoPipeline". The left sidebar contains navigation links for Developer Tools, CodePipeline, Source (CodeCommit), Artifacts (CodeArtifact), Build (CodeBuild), Deploy (CodeDeploy), Pipeline (CodePipeline), Getting started, Pipelines, Pipeline (highlighted), History, Settings, and Settings. The main area displays the pipeline execution details. The "Source" stage is marked as "Succeeded" and used "Amazon S3" as the provider. The "Deploy" stage is also marked as "Succeeded" and used "AWS Elastic Beanstalk" as the provider. The "AWS Elastic Beanstalk" link in the Deploy stage is highlighted with a red box. The pipeline execution ID is "b3115f96-4ec4-4c70-90ce-228bb7747b82".

## 2. Select the environment

The AWS Elastic Beanstalk console opens with the details of the deployment. Select the environment you created earlier, called **Deploymenttutorial-env**.

The screenshot shows the AWS Elastic Beanstalk console. On the left, the 'Elastic Beanstalk' sidebar is visible with options like 'Environments', 'Applications', and 'Change history'. The main content area is titled 'Application 'Deployment Tutorial' environments'. It features a table with columns: Environment name, Health, Date created, Last modified, URL, and Running versions. The 'Deploymenttutorial-env' is highlighted with a red box. Below the table, there are links for 'Go to environment', 'Configuration', 'Logs', 'Health', 'Monitoring', 'Alarms', 'Managed updates', and 'Events'.

Environment name	Health	Date created	Last modified	URL	Running versions
Deploymenttutorial-env	Ok	2022-08-30 09:02:06 UTC-0700	2022-09-05 11:15:30 UTC-0700	Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com	code-pipeline-16624017111 ABLhKicTNwtrXSA5t1hGb4Y

### 3. Select the URL

Select the URL to view the sample website again.

The screenshot shows the AWS Elastic Beanstalk console with the 'Deploymenttutorial-env' environment selected. The environment details are displayed, including the URL 'Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com', which is highlighted with a red box. The 'Health' section shows 'Ok' with a green checkmark. The 'Running version' section shows 'code-pipeline-1662399271185-Tzv1VxUki0BP4Y44gSQoCG78YgS8R58I' and an 'Upload and deploy' button. The 'Platform' section shows 'PHP 8.1 running on 64bit Amazon Linux 2/3.4.0' and a 'Change' button. A warning message indicates 'Different version recommended'.

**Deploymenttutorial-env**

Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com (e-9qdkhkapif)

Application name: Deployment Tutorial

**Health**

Ok

**Running version**

code-pipeline-1662399271185-Tzv1VxUki0BP4Y44gSQoCG78YgS8R58I

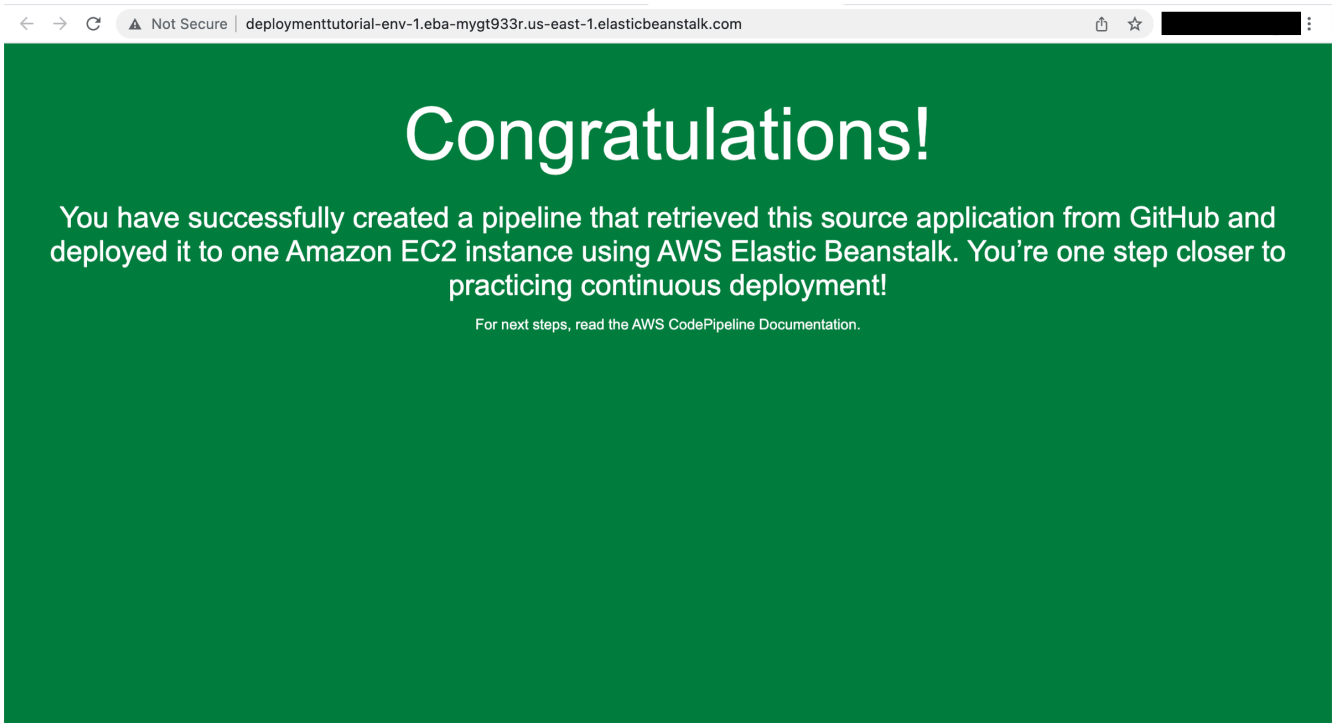
**Platform**

PHP 8.1 running on 64bit Amazon Linux 2/3.4.0

⚠ Different version recommended

### 4. View the page

Confirm that the updated text appears on the webpage.



## Amazon S3

1. Choose Elastic Beanstalk

Return to your pipeline in the CodePipeline console. In a few minutes, you should see the Source change to blue, indicating that the pipeline has detected the changes you made to your source repository. Once this occurs, it will automatically move the updated code to Elastic Beanstalk.

After the pipeline status displays **Succeeded**, in the status area for the Beta stage, choose **AWS Elastic Beanstalk**.

The screenshot shows the AWS CodePipeline console for a pipeline named 'DemoPipeline'. The left sidebar contains navigation links for Developer Tools, CodePipeline, and various stages like Source, Artifacts, Build, Deploy, Pipeline, and Settings. The main area displays the pipeline's execution history. The 'Source' stage is marked as 'Succeeded' and used 'Amazon S3' as the provider. The 'Deploy' stage is also marked as 'Succeeded' and used 'AWS Elastic Beanstalk' as the provider. A red box highlights the 'AWS Elastic Beanstalk' link in the Deploy stage. The pipeline execution ID is 'b3115f96-4ec4-4c70-90ce-228bb7747b82'. The bottom of the console shows a footer with 'Feedback', 'Looking for language selection? Find it in the new Unified Settings', and copyright information for Amazon Web Services, Inc. or its affiliates.

## 2. Select the environment

The AWS Elastic Beanstalk console opens with the details of the deployment. Select the environment you created earlier, called **Deploymenttutorial-env**.

The screenshot shows the AWS Elastic Beanstalk console. On the left, the 'Elastic Beanstalk' sidebar is visible with options like 'Environments', 'Applications', and 'Change history'. The main content area is titled 'Application 'Deployment Tutorial' environments'. It features a search bar and a 'Create a new environment' button. Below this is a table of environments:

Environment name	Health	Date created	Last modified	URL	Running versions
Deploymenttutorial-env	Ok	2022-08-30 09:02:06 UTC-0700	2022-09-05 11:15:30 UTC-0700	Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com	code-pipeline-16624017111 ABLhKicTNwtrXSA5t1hGb4Y

The 'Deploymenttutorial-env' environment name is highlighted with a red box. The bottom of the console shows a footer with 'Feedback', 'Looking for language selection? Find it in the new Unified Settings', and copyright information for Amazon Web Services.

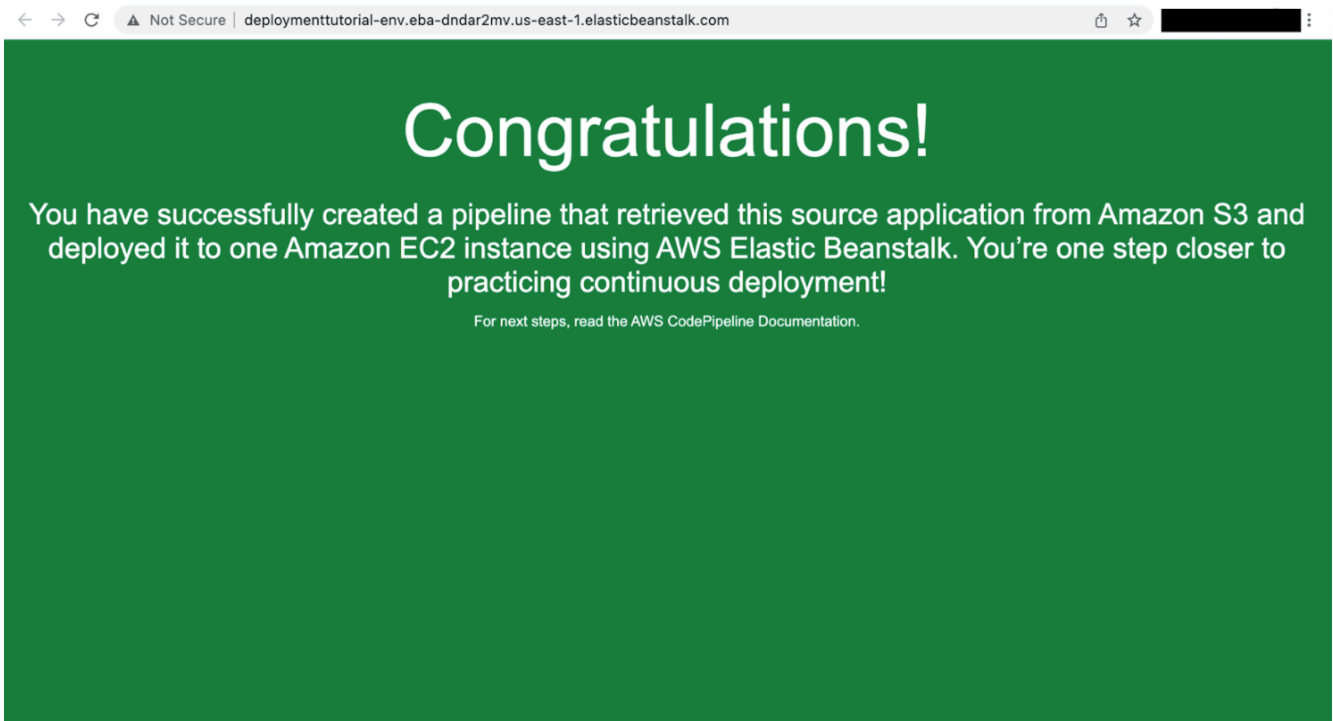
### 3. Select the URL

Select the URL to view the sample website again.

The screenshot shows the details of the 'Deploymenttutorial-env' environment. At the top, there's a blue banner about 'AWS Graviton now supported'. Below that, the breadcrumb navigation shows 'Elastic Beanstalk > Environments > Deploymenttutorial-env'. The main content area is titled 'Deploymenttutorial-env' and shows the URL 'Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com' highlighted with a red box. Below the URL, there's a 'Health' section with a green checkmark and 'Ok' status. The 'Running version' section shows 'code-pipeline-1662399271185-Tzv1VxUki0BP4Y44gSQoCG78YgS8R58I' and an 'Upload and deploy' button. The 'Platform' section shows 'PHP 8.1 running on 64bit Amazon Linux 2/3.4.0' and a 'Change' button. The bottom of the console shows a footer with 'Feedback', 'Looking for language selection? Find it in the new Unified Settings', and copyright information for Amazon Web Services.

### 4. View the page

Confirm that the updated text appears on the webpage.



## AWS CodeCommit

### 1. Choose Elastic Beanstalk

Return to your pipeline in the CodePipeline console. In a few minutes, you should see the Source change to blue, indicating that the pipeline has detected the changes you made to your source repository. Once this occurs, it will automatically move the updated code to Elastic Beanstalk.

After the pipeline status displays **Succeeded**, in the status area for the Beta stage, choose **AWS Elastic Beanstalk**.

The screenshot shows the AWS CodePipeline console for a pipeline named 'DemoPipeline'. The left sidebar contains navigation links for Developer Tools, CodePipeline, Source (CodeCommit), Artifacts (CodeArtifact), Build (CodeBuild), Deploy (CodeDeploy), Pipeline (CodePipeline), Getting started, Pipelines, Pipeline (highlighted), History, Settings, and Settings. The main area displays the pipeline execution details. The 'Source' stage is marked as 'Succeeded' and used 'Amazon S3' as the provider. The 'Deploy' stage is also marked as 'Succeeded' and used 'AWS Elastic Beanstalk' as the provider. The 'AWS Elastic Beanstalk' link in the Deploy stage is highlighted with a red box. The pipeline execution ID is 'b3115f96-4ec4-4c70-90ce-228bb7747b82'. The bottom of the console shows a footer with 'Feedback', 'Looking for language selection? Find it in the new Unified Settings', and copyright information for Amazon Web Services, Inc. or its affiliates.

## 2. Select the environment

The AWS Elastic Beanstalk console opens with the details of the deployment. Select the environment you created earlier, called **Deploymenttutortial-env**.

The screenshot shows the AWS Elastic Beanstalk console. In the left sidebar, under 'Elastic Beanstalk', the 'Deployment Tutorial' is selected. The main content area displays 'Application 'Deployment Tutorial' environments'. A table lists the environments, with 'Deploymenttutorial-env' highlighted by a red box. The table columns are: Environment name, Health, Date created, Last modified, URL, and Running versions.

Environment name	Health	Date created	Last modified	URL	Running versions
Deploymenttutorial-env	Ok	2022-08-30 09:02:06 UTC-0700	2022-09-05 11:15:30 UTC-0700	Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com	code-pipeline-16624017111 ABLhKicTNwtrXSA5t1hGb4Y

### 3. Select the URL

Select the URL to view the sample website again.

The screenshot shows the details for the 'Deploymenttutorial-env' environment. The URL 'Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com' is highlighted with a red box. The console displays the environment's health, running version, and platform.

**Deploymenttutorial-env**  
 Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com (e-9qdkhkapif)  
 Application name: Deployment Tutorial

**Health**  
 Ok  
 Causes

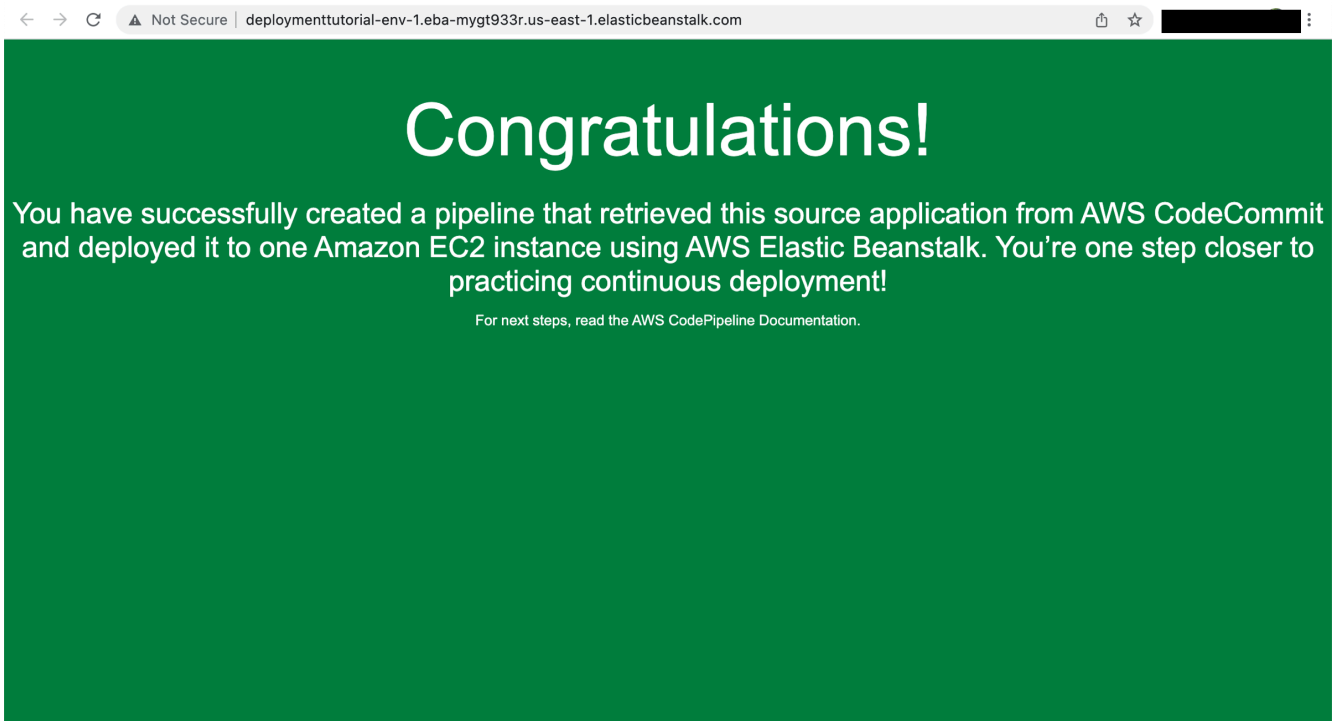
**Running version**  
 code-pipeline-1662399271185-Tzv1VxUki0BP4Y44gSQoCG78YgS8R58I  
 Upload and deploy

**Platform**  
 PHP 8.1 running on 64bit Amazon Linux 2/3.4.0  
 ⚠ Different version recommended  
 Change

### 4. View the page

Confirm that the updated text appears on the webpage.





## Clean up resources

To avoid future charges, you will delete all the resources you launched throughout this tutorial, which includes the pipeline, the Elastic Beanstalk application, and the source you set up to host the code.

### 1. Delete the pipeline

First, you will delete your pipeline. In the **Pipelines** view, select the pipeline radio button and select **Delete pipeline**.

The screenshot shows the AWS CodePipeline console. In the left-hand navigation pane, the 'Pipelines' link is highlighted. The main content area displays a list of pipelines. The first pipeline, 'DemoPipeline', is highlighted with a blue row. The 'Delete pipeline' button in the top right of the pipeline list is highlighted with a red rectangle. Below the pipeline list, there is a table with columns: Name, Most recent execution, Latest source revisions, and Last executed. The 'DemoPipeline' row shows a 'Succeeded' status and a source action using Amazon S3.

Name	Most recent execution	Latest source revisions	Last executed
DemoPipeline	Succeeded	Source – mnF6unAr: Amazon S3 version id: mnF6unArGzf17cQDFwx0r00dNJeKkGpJ	1 minute ago

## 2. Confirm deletion

To confirm deletion, enter **delete** in the field and choose **Delete**.

The screenshot shows the 'Delete DemoPipeline?' confirmation dialog. The dialog has a text input field with the word 'delete' entered. Below the input field, there is a section titled 'Deleting change detection resources' which lists the resources that will be removed. At the bottom of the dialog, there is a 'Delete' button highlighted in red.

**Delete DemoPipeline?**

To confirm deletion, type *delete* in the field.

**Deleting change detection resources**  
The following resources were used to detect source changes and will be removed.

Change type	Details
Remove	Amazon CloudWatch Events rule and AWS CloudTrail data event for pipeline DemoPipeline: codepipeline-18308900-awscodepipelinedemobucket-rule, awscodepipeline-demobucket-8302022/aws-codepipeline-s3-aws-codedeploy_linux.zip (Data event will not be changed if in use for other pipelines.)

☐ No resource updates needed for this source action change

Cancel **Delete**

## 3. Delete the Beanstalk application

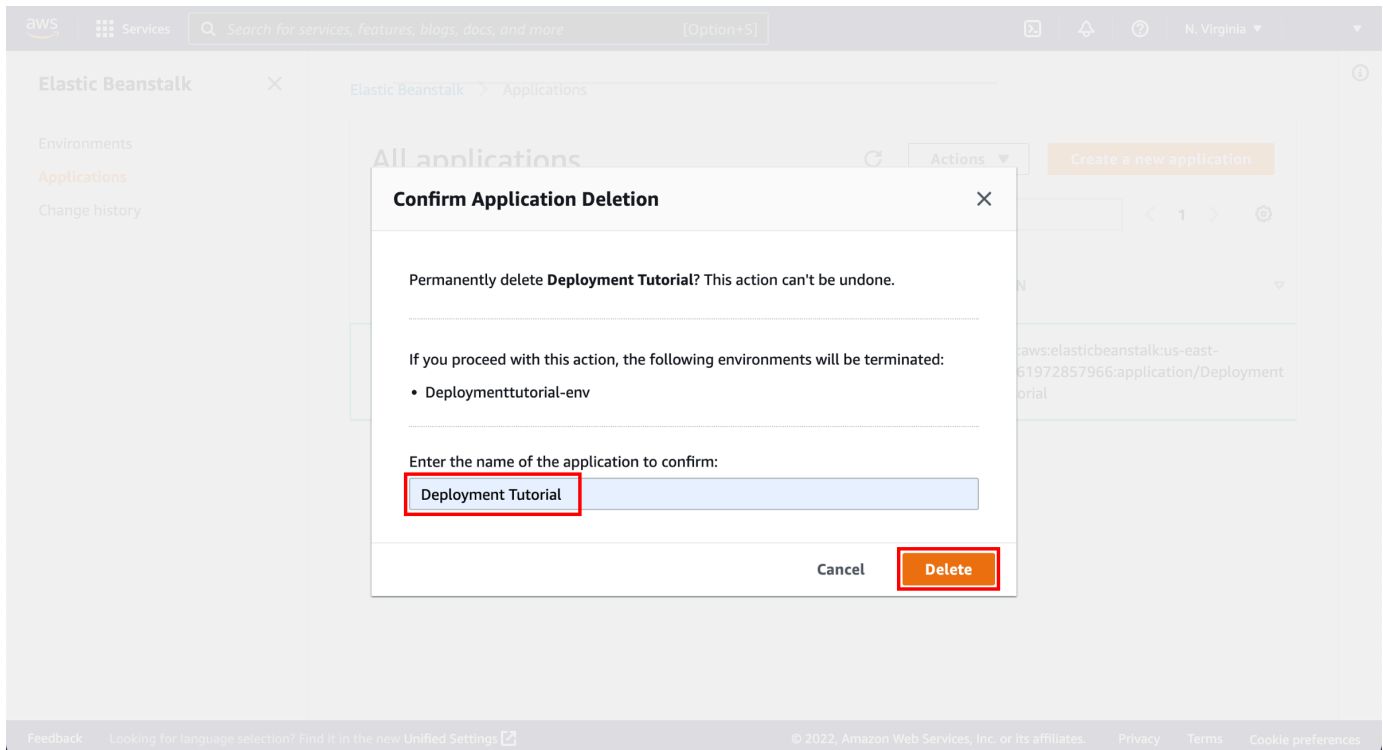
Second, delete your Elastic Beanstalk application. Visit the Elastic Beanstalk **Applications** page. Select the radio button for the **Deployment Tutorial**. Select **Actions** and **Delete application**.

The screenshot shows the AWS Elastic Beanstalk console. On the left, the 'Elastic Beanstalk' sidebar is visible with 'Applications' selected. The main content area is titled 'All applications'. A search bar is present. Below it, a table lists applications. The first application, 'Deployment Tutorial', is selected with a radio button. The 'Actions' dropdown menu is open for this application, showing options like 'Create environment', 'Delete application', 'View application versions', 'View saved configurations', and 'Restore terminated environment'. The 'Delete application' option is highlighted. The table row for 'Deployment Tutorial' shows the environment 'Deploymenttutorial-env-2', creation time '2022-09-05 11:41:05 UTC-0700', and last modified time '2022-09-05 11:41:05 UTC-0700'. The application ID is 'arn:aws:elasticbeanstalk:us-east-1:661972857966:application/Deployment Tutorial'.

Application name	Environments	Date created	Last modified
Deployment Tutorial	Deploymenttutorial-env-2	2022-09-05 11:41:05 UTC-0700	2022-09-05 11:41:05 UTC-0700

#### 4. Confirm deletion

In the **Confirm Application Deletion** window, enter the name of the application to be deleted and choose **Delete**.



## (Optional) Delete Amazon S3 resources

If you used Amazon S3 as your source, you can delete the resources to avoid future charges.

### 1. Empty the bucket contents

Visit the S3 console. First, we will empty the S3 bucket. Select the radio button next to the **awscodepipeline** bucket and choose **Empty**.

The screenshot shows the AWS Management Console interface for the 'Buckets' page under 'Amazon S3'. At the top, there's a search bar and navigation links. Below, an 'Account snapshot' section is visible. The main section is titled 'Buckets (5)' and includes a search bar, a refresh button, and action buttons: 'Copy ARN', 'Empty' (highlighted with a red box), 'Delete', and 'Create bucket'. A table lists the buckets with columns for Name, AWS Region, Access, and Creation date. The first bucket, 'awscodepipeline-demobucket-8302022', is highlighted with a red box around its selection icon. The footer contains feedback links and copyright information.

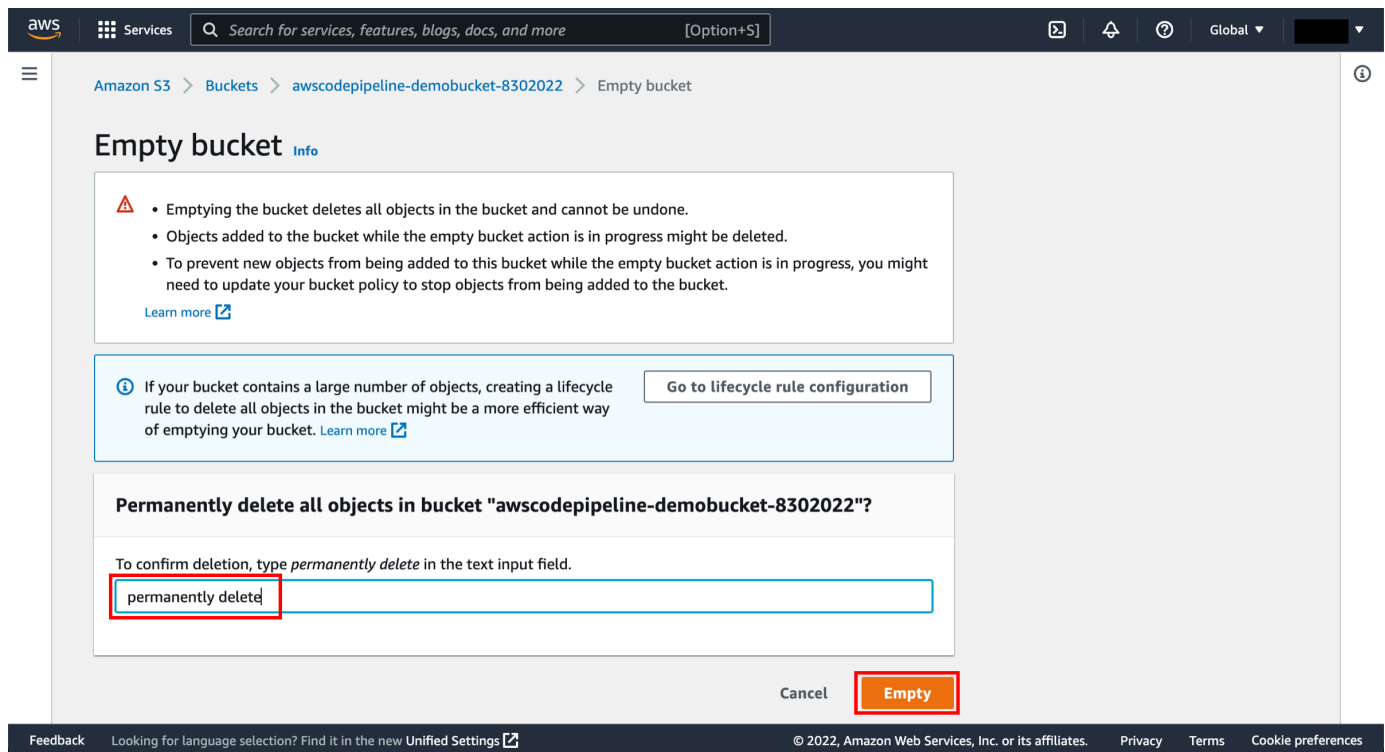
## 2. Confirm deletion

When a confirmation message appears, enter **permanently delete** in the text input field and choose **Empty**.

The screenshot shows the 'Empty bucket' confirmation dialog in the AWS Management Console. The breadcrumb trail indicates the path: 'Amazon S3 > Buckets > awscodepipeline-demobucket-8302022 > Empty bucket'. The dialog title is 'Empty bucket'. It contains a warning section with a triangle icon and a list of bullet points explaining the consequences of emptying the bucket. Below this is an information section with a circle icon and a button 'Go to lifecycle rule configuration'. The main part of the dialog is a confirmation prompt: 'Permanently delete all objects in bucket "awscodepipeline-demobucket-8302022"?'. It instructs the user to type 'permanently delete' in the text input field, which is shown with the text already entered and highlighted by a red box. At the bottom, there are 'Cancel' and 'Empty' buttons, with the 'Empty' button highlighted by a red box. The footer is identical to the previous screenshot.

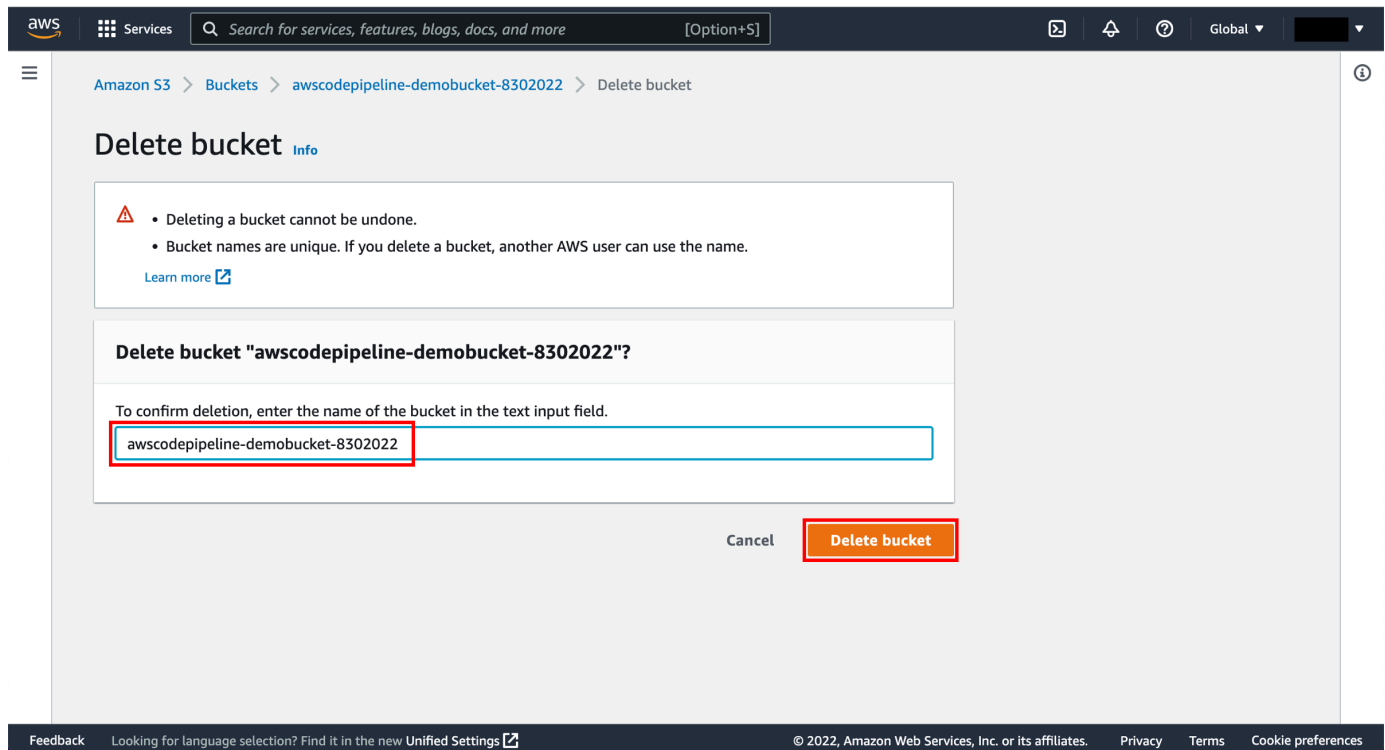
### 3. Delete the bucket

Now we will delete the bucket. Select the radio button next to the **awscodepipeline** bucket and choose **Delete**.



### 4. Confirm deletion

When a confirmation message appears, enter the bucket name and then choose **Delete bucket**.



## (Optional) Delete AWS CodeCommit resources

If you used AWS CodeCommit as your source, you can delete the resources to avoid future charges.

### 1. Delete the repository

Open the [AWS CodeCommit repository](#). Select the radio button next to the repository you created and choose **Delete repository**.

The screenshot shows the AWS CodeCommit 'Repositories' page. On the left is a navigation sidebar with 'CodeCommit' selected. The main content area shows a list of repositories. The 'Delete repository' button is highlighted with a red box. Below it, a table lists repositories, with 'PipelineRepo' highlighted by a blue selection bar and a red circle around its icon.

Name	Description	Last modified	Clone URL
PipelineRepo	-	1 hour ago	<a href="#">HTTPS</a> <a href="#">SSH</a> <a href="#">HTTPS (GRC)</a>

## 2. Confirm deletion

A confirmation window will pop up. Enter **delete** and choose **Delete**.

The screenshot shows a confirmation dialog titled 'Delete PipelineRepo?'. It contains a warning icon and text stating that deleting the repository is irreversible and will affect all branches, triggers, comments, pull requests, and history. Below the text is a field for confirmation, where the word 'delete' has been entered and is highlighted with a red box. 'Cancel' and 'Delete' buttons are at the bottom right.

**Delete PipelineRepo?**

**Are you sure you want to delete the repository PipelineRepo?**  
This will delete the repository in AWS CodeCommit, including all branches, triggers, comments, pull requests, and history. Deleting the repository cannot be undone.  
Users will no longer be able to connect to the repository in AWS CodeCommit, but they will still have access to their local repositories.

To confirm deletion, type *delete* in the field.

delete

Cancel Delete



## Conclusion

Congratulations! You have successfully created an automated software release pipeline using AWS CodePipeline. Using CodePipeline, you created a pipeline that uses GitHub, Amazon S3, or AWS CodeCommit as the source location for application code and then deploys the code to an Amazon EC2 instance managed by AWS Elastic Beanstalk. Your pipeline will automatically deploy your code every time there is a code change. You are one step closer to practicing continuous deployment!