



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 © 2026 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

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

Set Up a Continuous Deployment Pipeline Using AWS CodePipeline

AWS experience	Beginner
Minimum time to complete	30 minutes
Cost to complete	Free Tier eligible
Services used	AWS CodePipeline AWS Elastic Beanstalk
Last updated	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**.

The screenshot shows the AWS Elastic Beanstalk console. At the top, there's a search bar with the text "Search for services, features, blogs, docs, and more" and a search icon. To the right of the search bar is a dropdown menu for the region, currently set to "N. Virginia". Below the search bar is a navigation menu with "Elastic Beanstalk" selected. The main content area is titled "Compute" and features a large heading "Amazon Elastic Beanstalk End-to-end web application management." Below this heading is a "Get started" section with a "Create Application" button. To the right of the "Get started" section is a "Pricing" section. Below the "Pricing" section is a "How it works" section. The footer contains a "Feedback" link, a link for language selection, and copyright information for Amazon Web Services, Inc. or its affiliates.

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.

The screenshot shows the AWS Elastic Beanstalk console interface for creating a new web application. The page title is "Create a web app" and it includes a sub-header "Getting started". The main content is divided into several sections:

- Application information:** The "Application name" field is filled with "Deployment Tutorial". A note below states: "Up to 100 Unicode characters, not including forward slash (/)." The text "Deployment Tutorial" is highlighted with a red box.
- Application tags:** A section for adding tags, with a note: "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](#)". It includes input fields for "Key" and "Value", and buttons for "Add tag" and "Remove tag". A "50 remaining" indicator is shown.
- Platform:** A section for selecting the platform. The "Platform" dropdown is set to "PHP" (highlighted with a red box). The "Platform branch" is set to "PHP 8.1 running on 64bit Amazon Linux 2". The "Platform version" is set to "3.4.0 (Recommended)".
- Application code:** A section for selecting the code source. The "Sample application" radio button is selected, with the sub-text "Get started right away with sample code." The "Upload your code" radio button is unselected, with the sub-text "Upload a source bundle from your computer or copy one from Amazon S3."

At the bottom of the form, there are three buttons: "Cancel", "Configure more options", and "Create application" (highlighted with a red box).

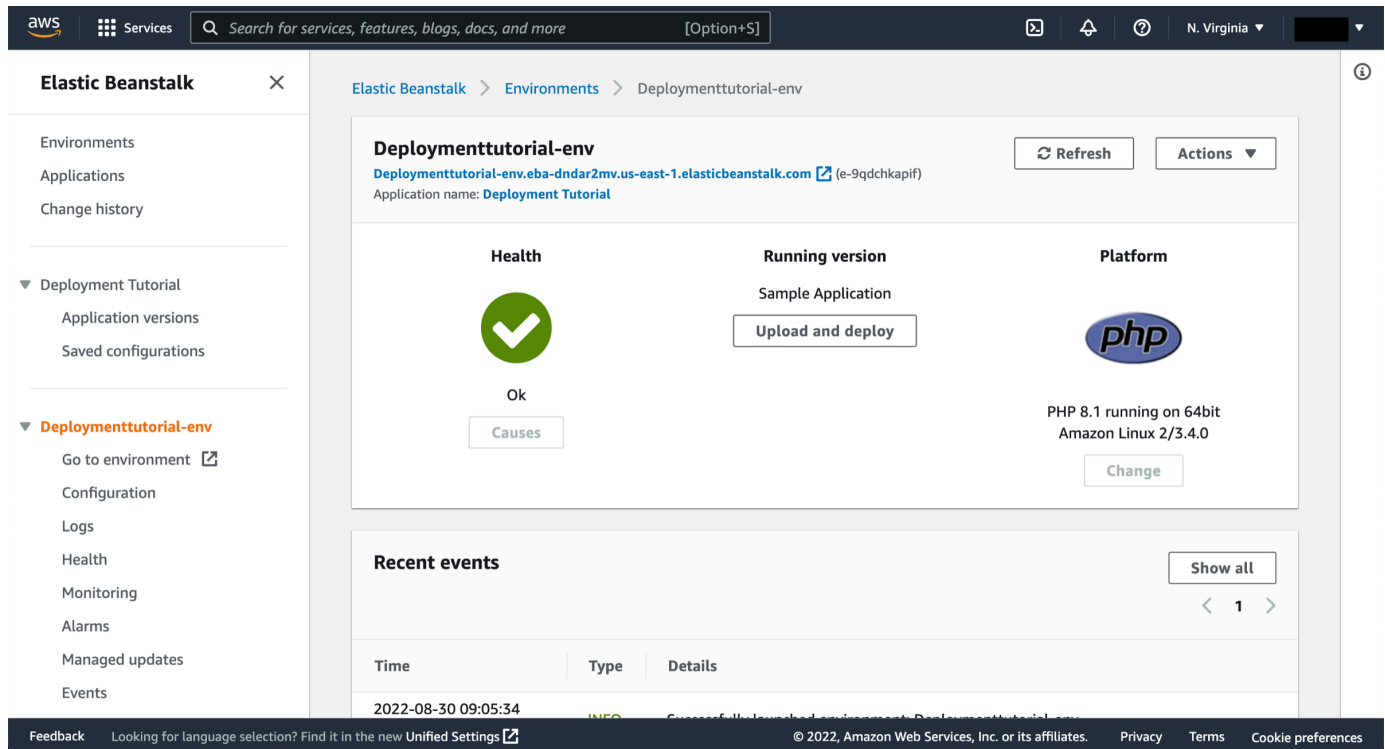
The footer of the console shows: "Feedback", "Looking for language selection? Find it in the new Unified Settings", "© 2022, Amazon Web Services, Inc. or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

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

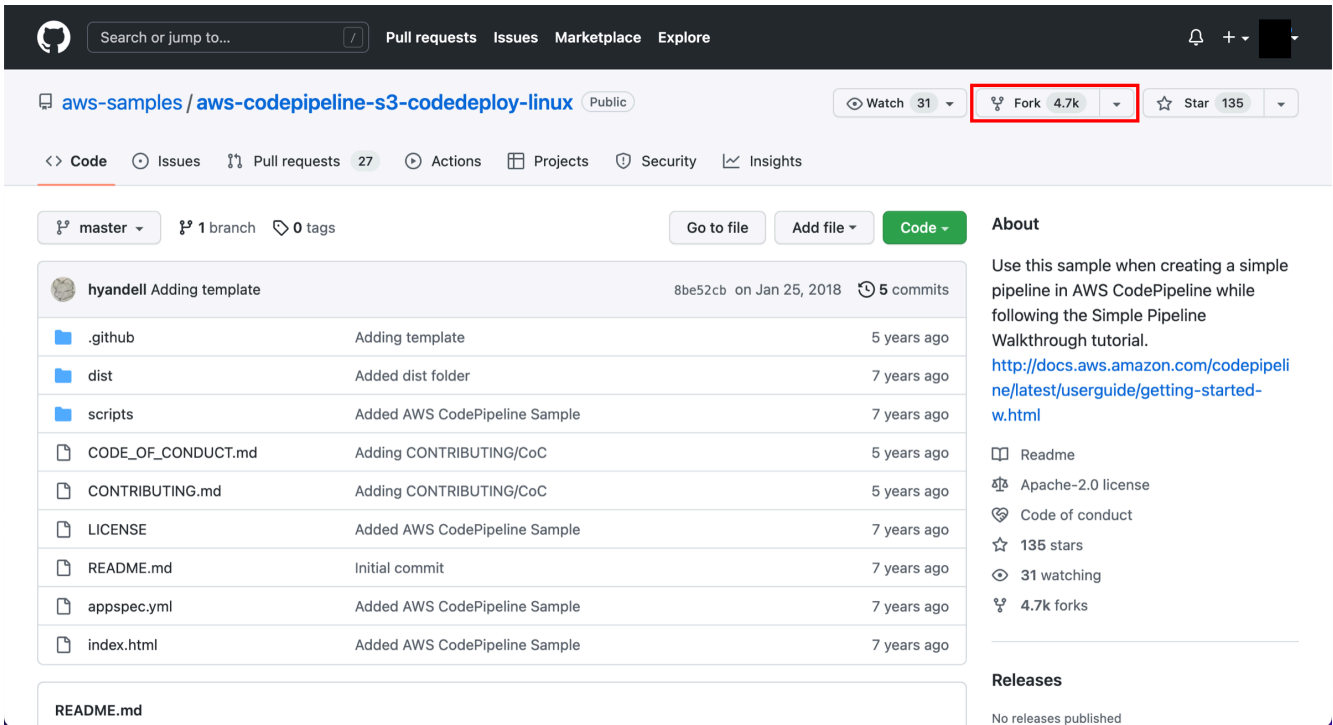
Use this procedure if you would like to use your GitHub account as your source.

- 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**.



The screenshot shows the GitHub repository page for `aws-samples/aws-codepipeline-s3-codedeploy-linux`. The `Fork` button is highlighted with a red box. The repository has 4.7k forks, 135 stars, and 31 watchers. The file list includes `.github`, `dist`, `scripts`, `CODE_OF_CONDUCT.md`, `CONTRIBUTING.md`, `LICENSE`, `README.md`, `appspec.yml`, and `index.html`. The `dist` folder is highlighted in blue.

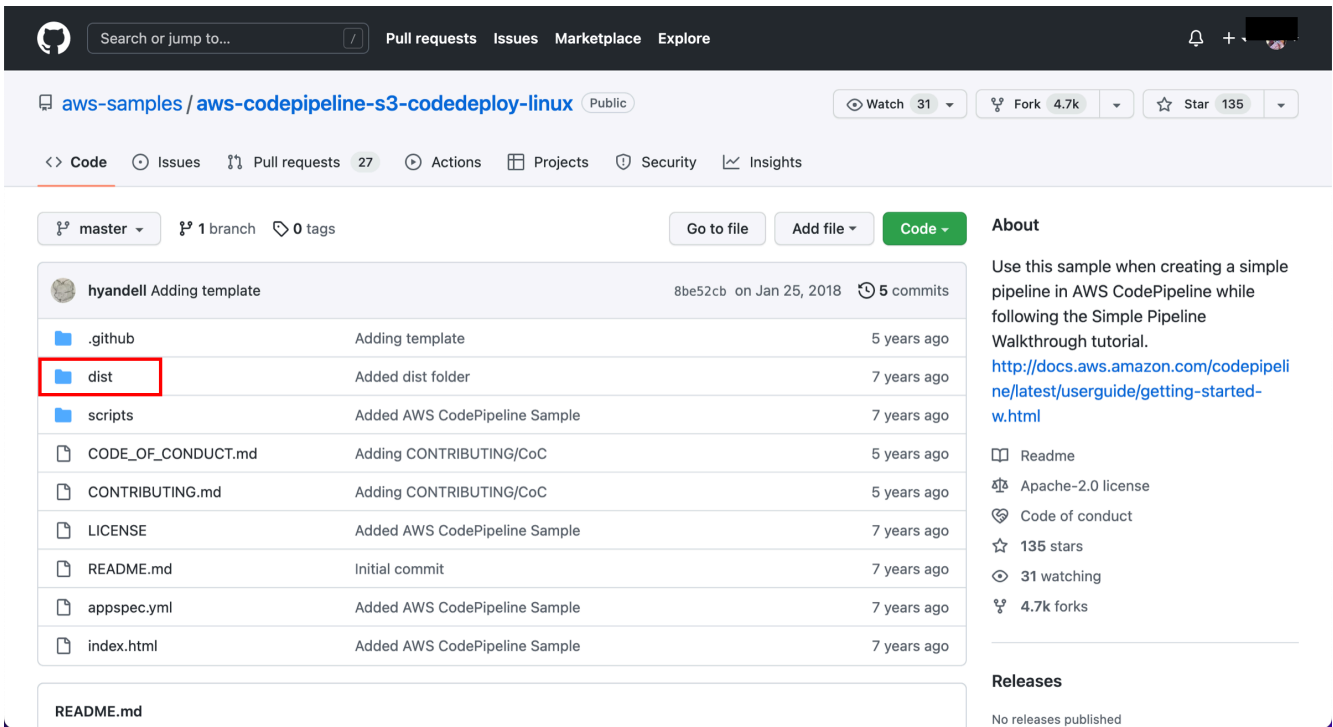
Amazon S3

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

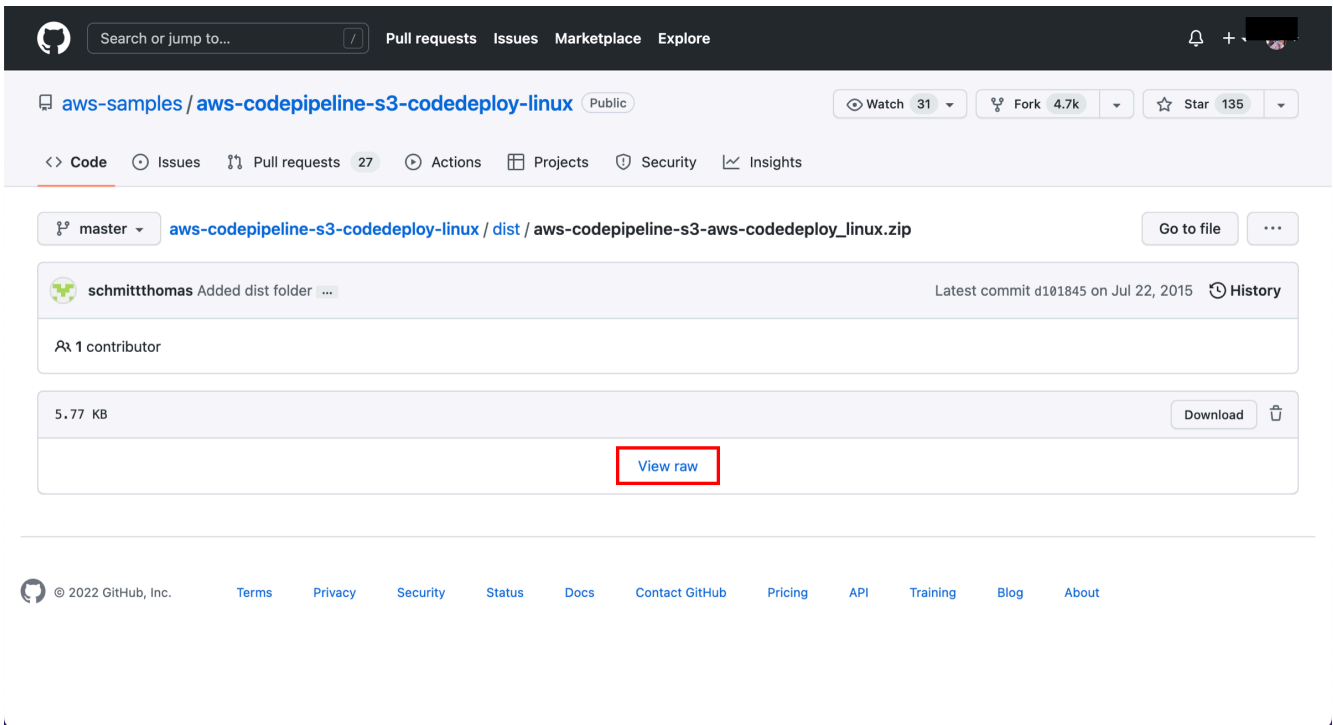
Releases

No releases published

2. Download the files

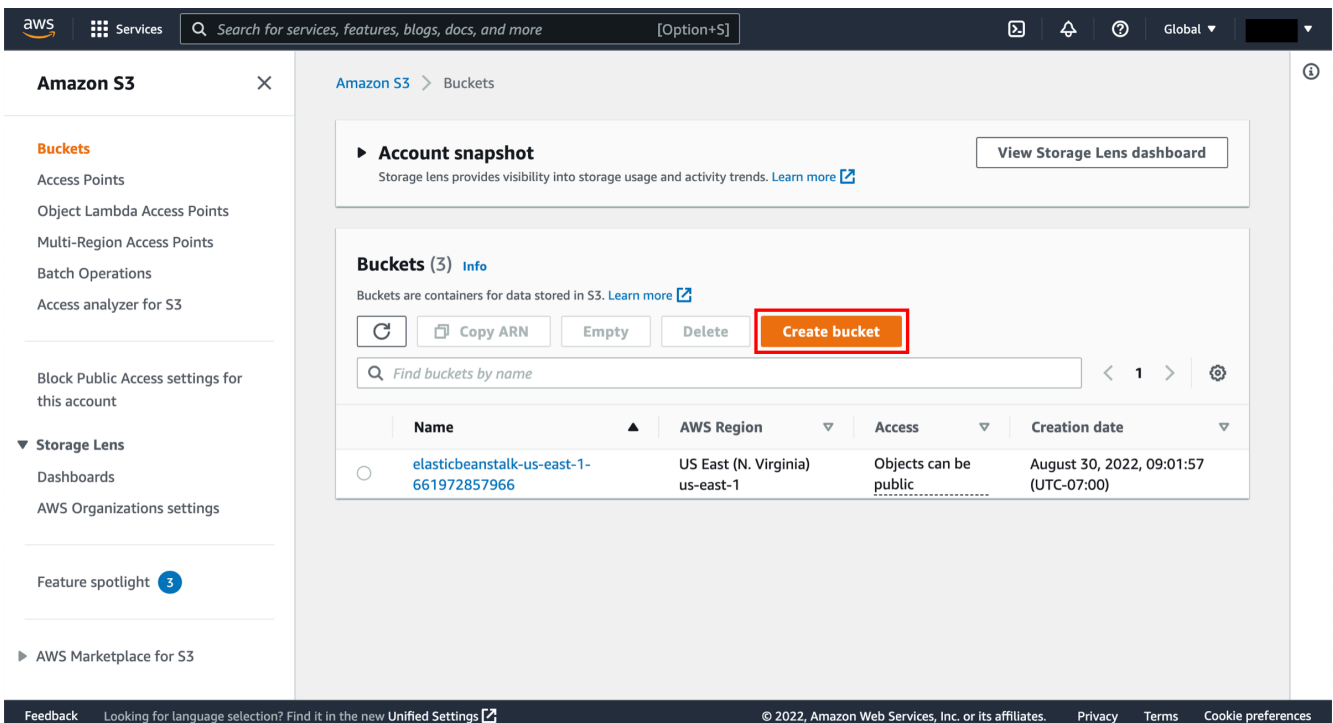
Save the source files to your computer:

- 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
awspipeline-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](#)

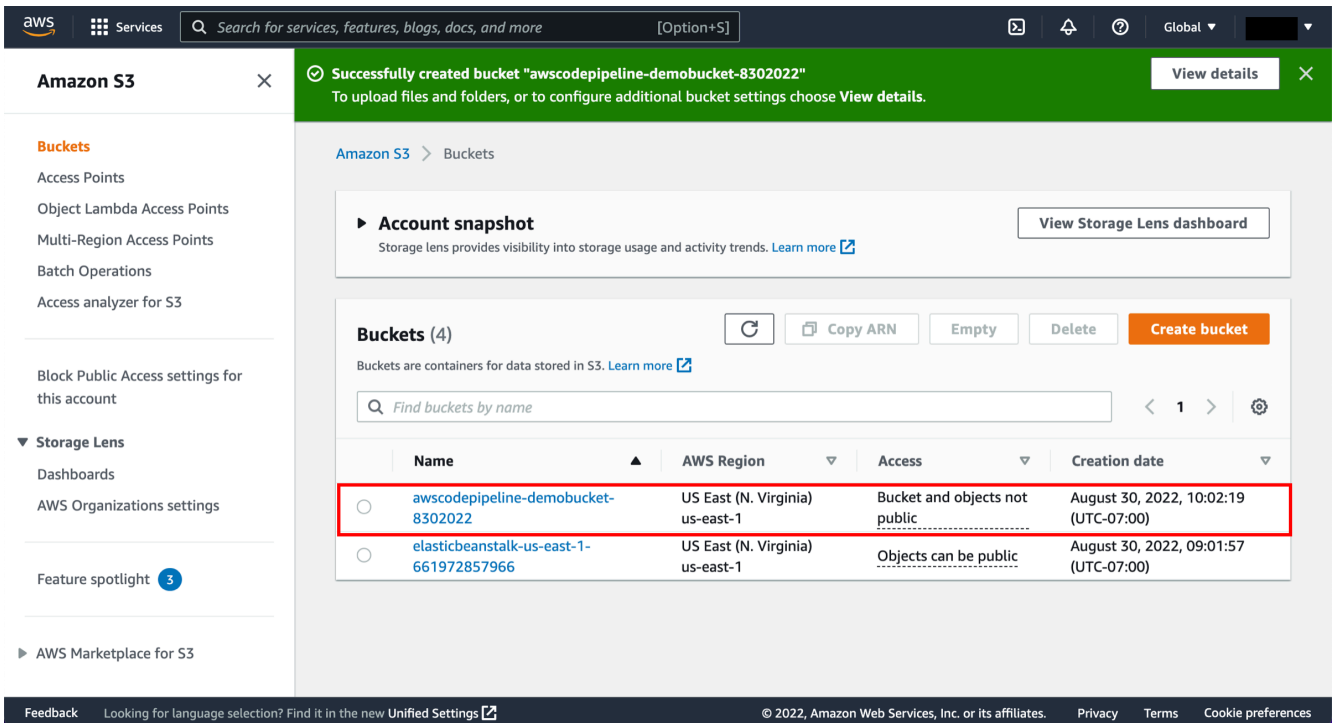
No tags associated with this bucket.
[Add tag](#)

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

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

6. Open the bucket

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

Amazon S3

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

Amazon S3 > Buckets

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

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 > 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](#)

Create folder **Upload** Copy S3 URI Copy URL Download Open Delete Actions

Find objects by prefix Show versions

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

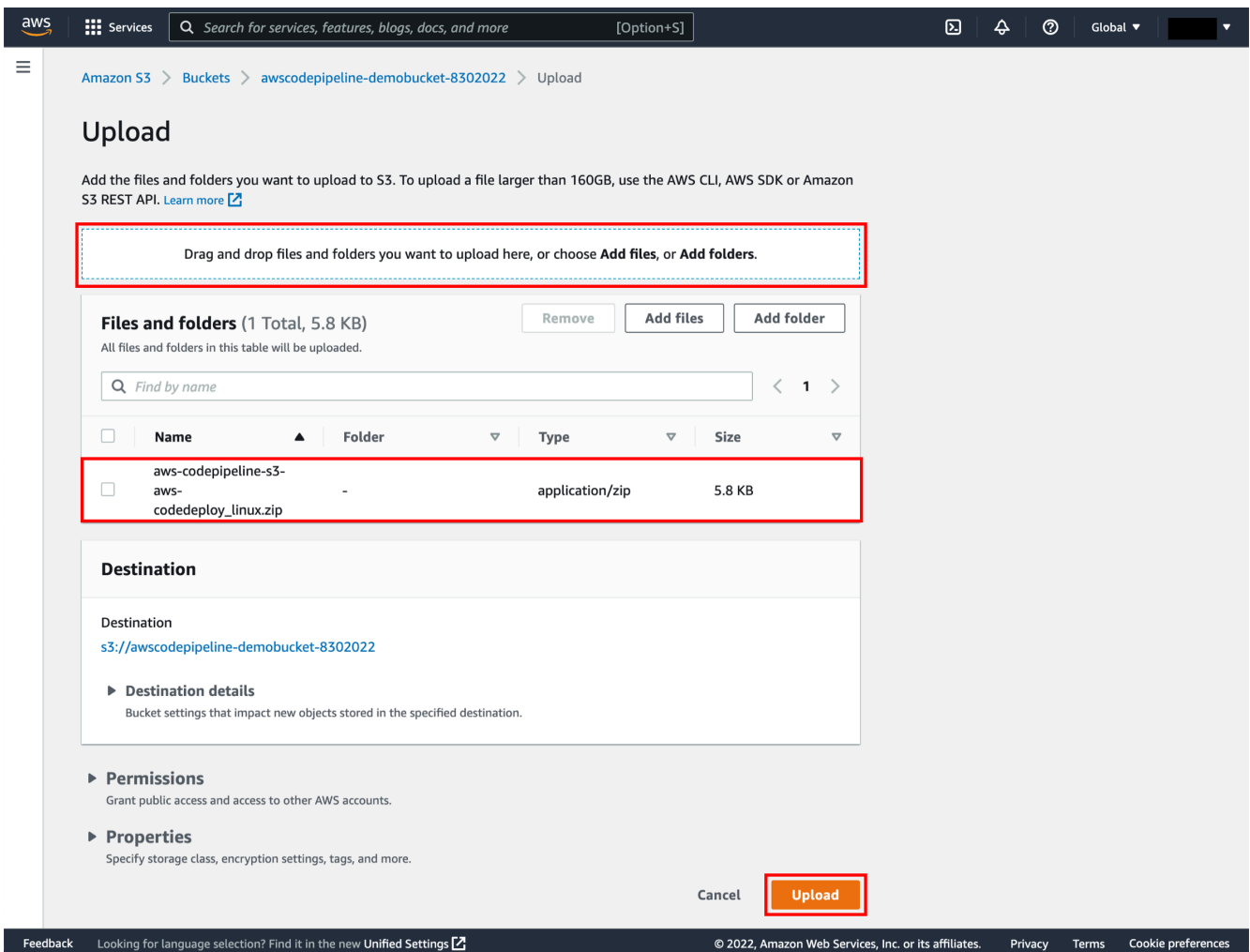
https://us-east-1.console.aws.amazon.com/s3# Find it in the new Unified Settings

© 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**.



The screenshot shows the AWS S3 console's 'Upload' page. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and utility icons. Below that, the breadcrumb path is 'Amazon S3 > Buckets > awscodepipeline-demobucket-8302022 > Upload'. The main heading is 'Upload'. A message says: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)'. Below this is a dashed box containing the text: 'Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.' Underneath is a section titled 'Files and folders (1 Total, 5.8 KB)' with 'Remove', 'Add files', and 'Add folder' buttons. A search bar 'Find by name' is present. A table lists the file:

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	aws-codepipeline-s3-aws-codedeploy_linux.zip	-	application/zip	5.8 KB

Below the table is the 'Destination' section, showing 'Destination: s3://awscodepipeline-demobucket-8302022' and 'Destination details'. At the bottom, there are sections for 'Permissions' and 'Properties'. The 'Upload' button is highlighted with a red box.

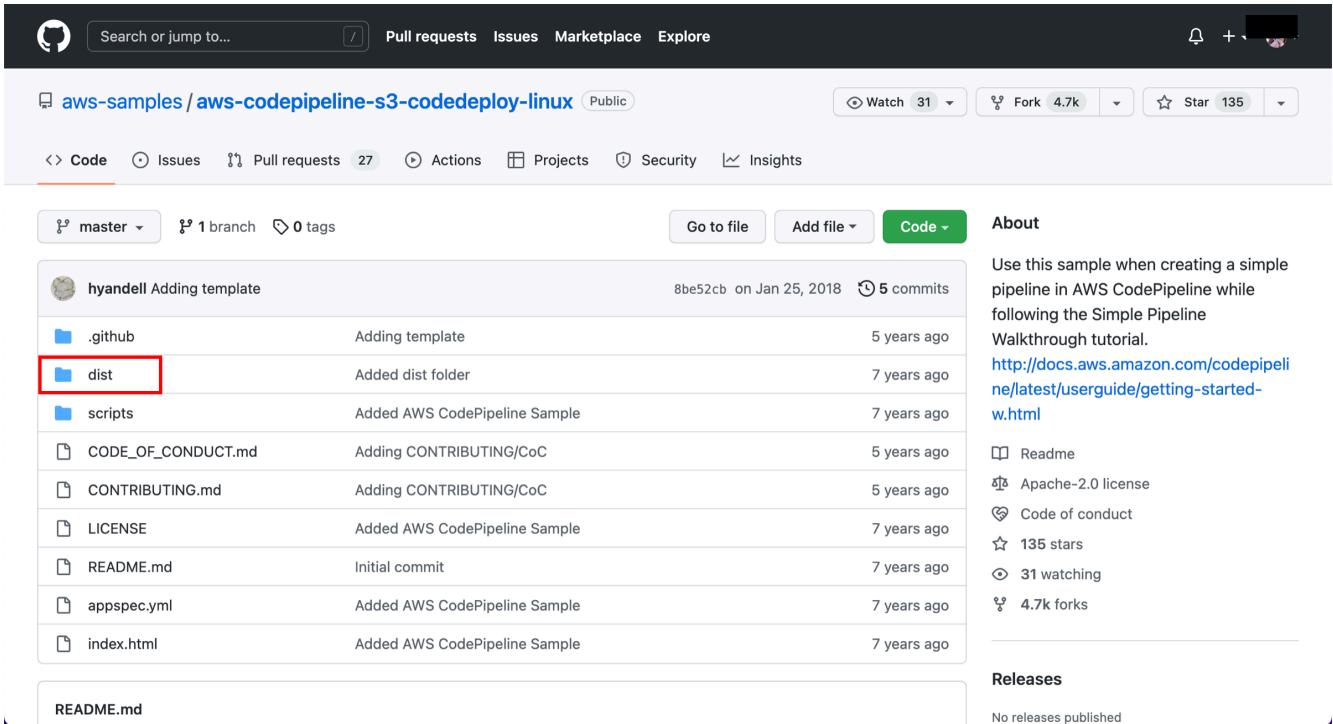
AWS CodeCommit

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.



The screenshot shows the GitHub repository page for `aws-samples/aws-codepipeline-s3-codedeploy-linux`. The repository is public and has 31 watchers, 4.7k forks, and 135 stars. The file list shows the following files and folders:

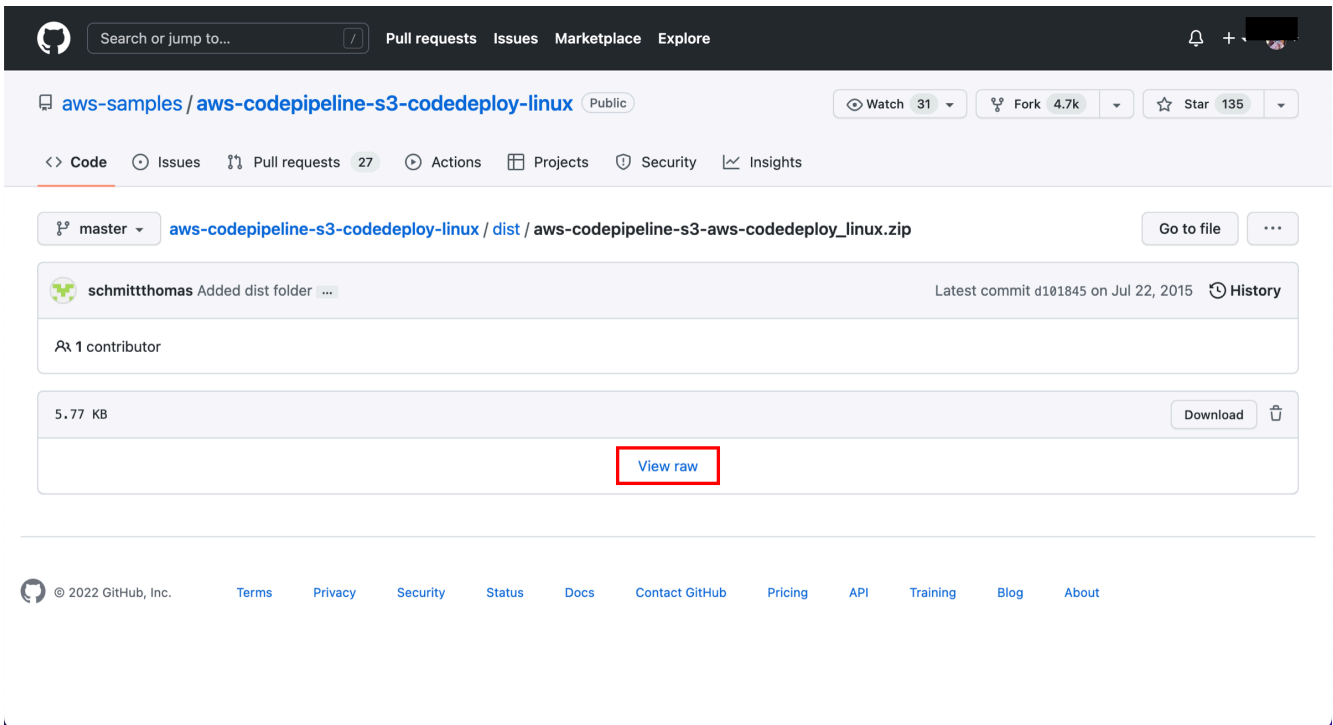
File/Folder	Description	Commit Date
<code>.github</code>	Adding template	5 years ago
<code>dist</code>	Added dist folder	7 years ago
<code>scripts</code>	Added AWS CodePipeline Sample	7 years ago
<code>CODE_OF_CONDUCT.md</code>	Adding CONTRIBUTING/CoC	5 years ago
<code>CONTRIBUTING.md</code>	Adding CONTRIBUTING/CoC	5 years ago
<code>LICENSE</code>	Added AWS CodePipeline Sample	7 years ago
<code>README.md</code>	Initial commit	7 years ago
<code>appspec.yml</code>	Added AWS CodePipeline Sample	7 years ago
<code>index.html</code>	Added AWS CodePipeline Sample	7 years ago

The `dist` folder is highlighted with a red box. The right sidebar shows the repository's description, license (Apache-2.0), and other metadata.

2. Download the files

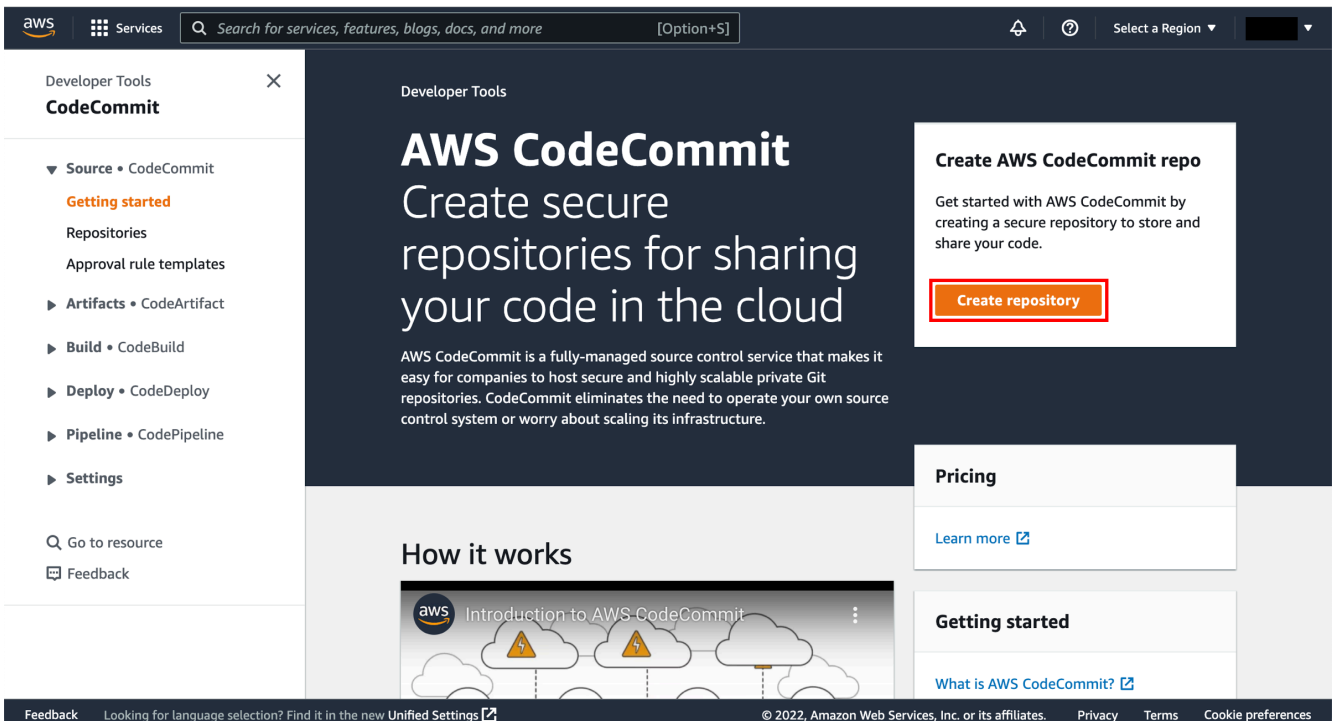
Save the source files to your computer:

- 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**.

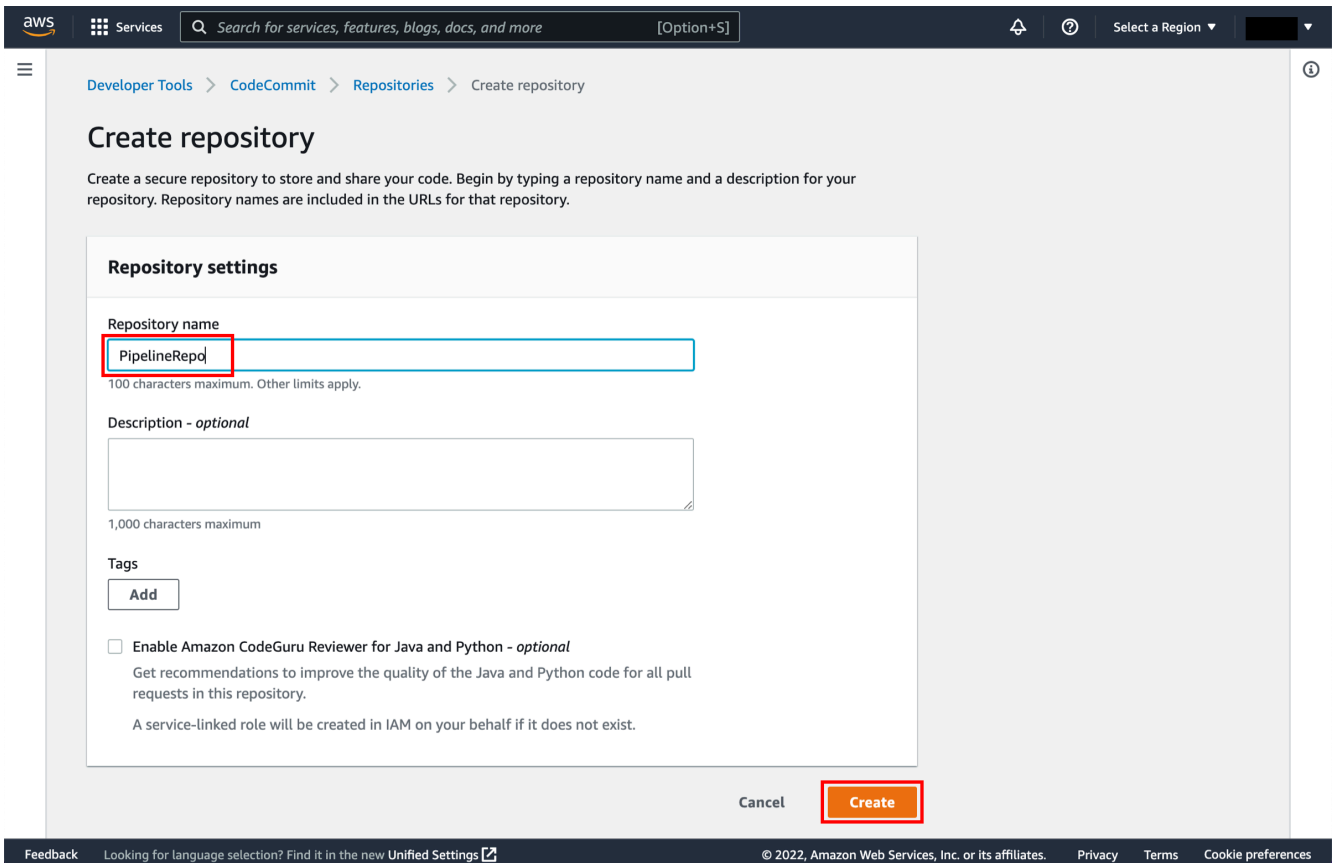


4. Configure repository settings

On the **Create repository** page:

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

Choose **Create**.



The screenshot shows the AWS CodeCommit 'Create repository' page. The breadcrumb navigation is 'Developer Tools > CodeCommit > Repositories > Create repository'. The page title is 'Create repository'. Below the title, there is a brief instruction: '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', highlighted with a red box. Below it, a note says '100 characters maximum. Other limits apply.'
- Description - optional:** A text area for a description, with a note '1,000 characters maximum' below it.
- Tags:** A section with an 'Add' button.
- Enable Amazon CodeGuru Reviewer for Java and Python - optional:** A checkbox that is currently unchecked. 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 form, there are two buttons: 'Cancel' and 'Create', with the 'Create' button highlighted by a red box.

The footer of the page includes: 'Feedback', 'Looking for language selection? Find it in the new Unified Settings', '© 2022, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and '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
Empty repository		
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.		
Create 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
Upload file		
Choose a file to upload.		
Choose file		

Commit changes to main

Author name

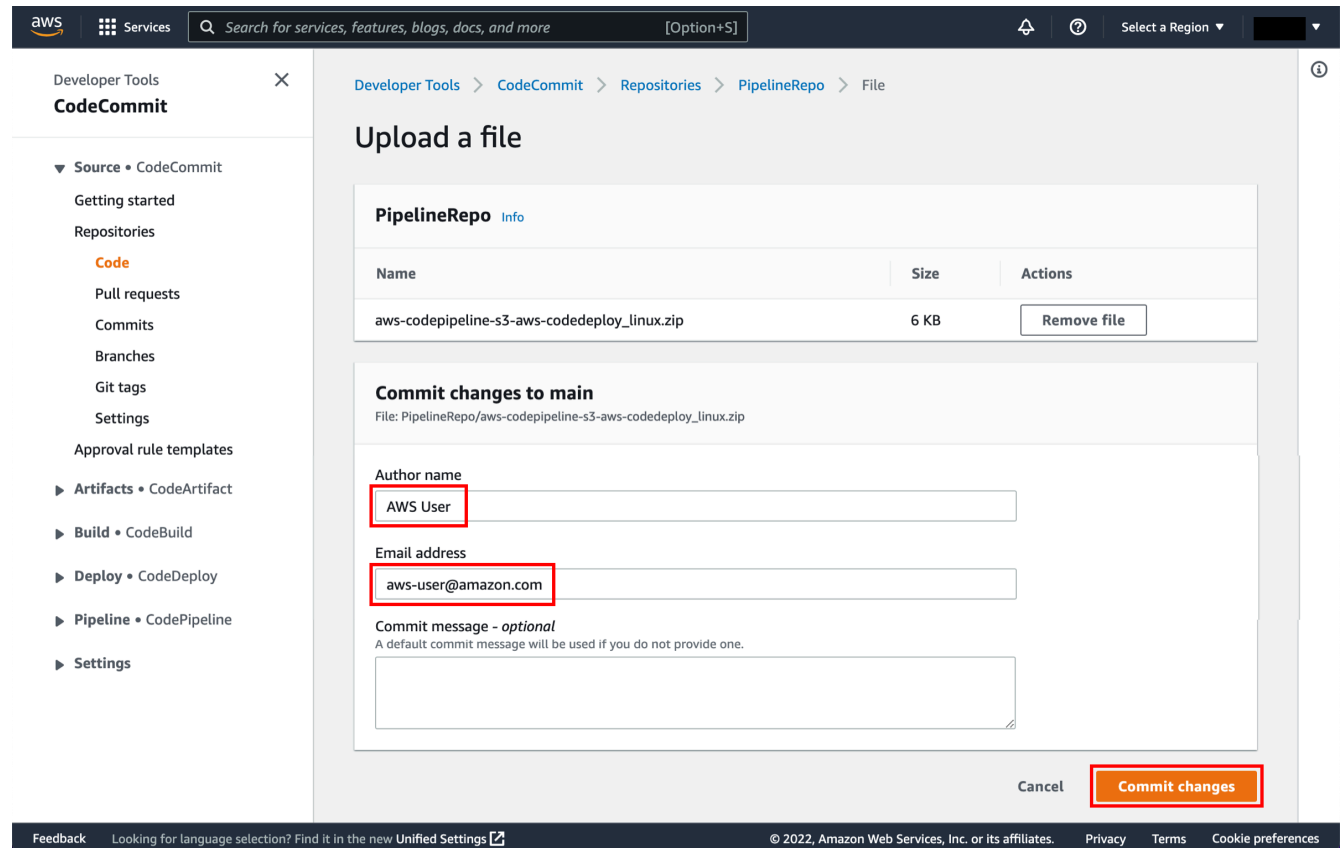
Email address

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**.



The screenshot shows the AWS CodeCommit console interface for uploading a file. The breadcrumb navigation is Developer Tools > CodeCommit > Repositories > PipelineRepo > File. The main heading is 'Upload a file'. Below this, there is a 'PipelineRepo' section with an 'Info' link. A table lists the uploaded file:

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

Below the table is the 'Commit changes to main' section. The file path is 'File: PipelineRepo/aws-codepipeline-s3-aws-codedeploy_linux.zip'. There are three input fields:

- Author name:** 'AWS User' (highlighted with a red box)
- Email address:** 'aws-user@amazon.com' (highlighted with a red box)
- Commit message - optional:** A text area with a placeholder message: 'A default commit message will be used if you do not provide one.'

At the bottom right, there are two buttons: 'Cancel' and 'Commit changes' (highlighted with a red box).

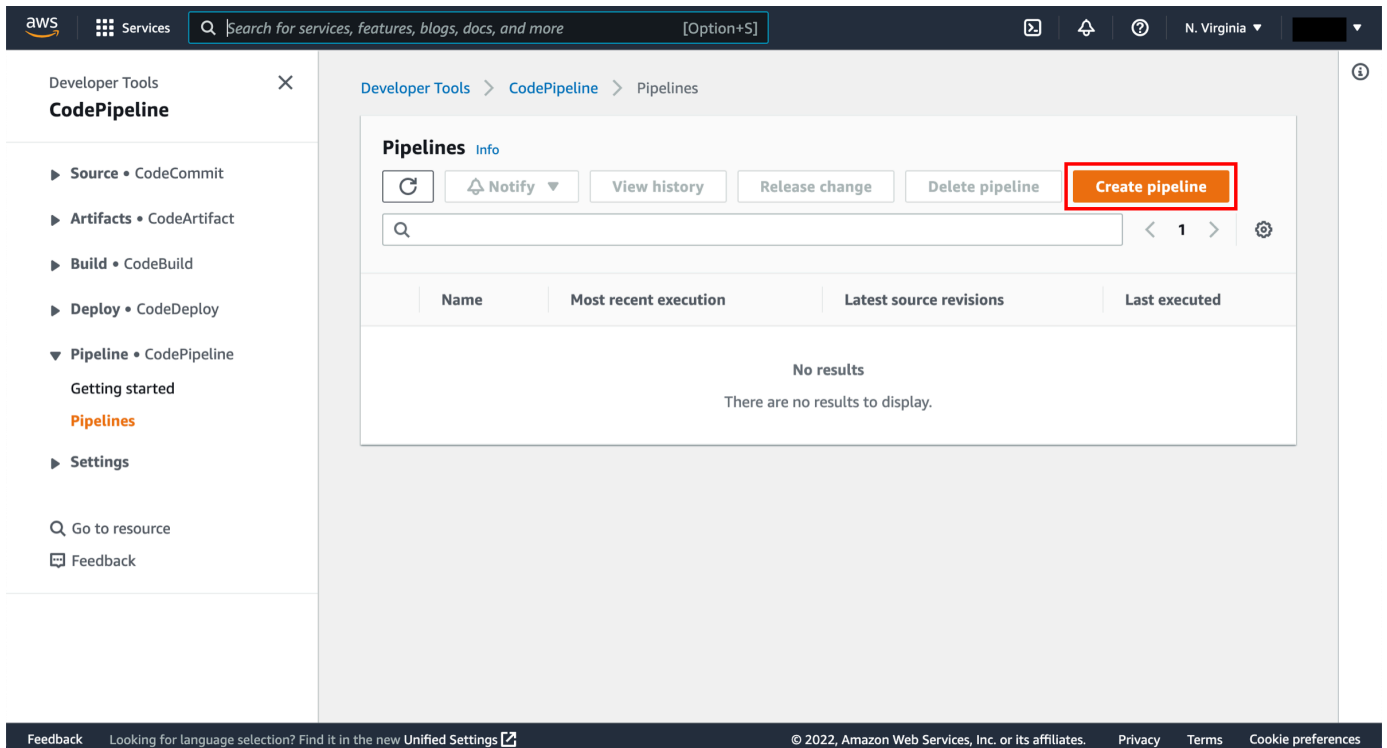
Step 3: Create your pipeline

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**.



The screenshot shows the AWS CodePipeline console interface. The top navigation bar includes the AWS logo, a search bar, and the region 'N. Virginia'. The left sidebar shows the 'Developer Tools' menu with 'CodePipeline' selected. The main content area displays the 'Pipelines' page, which includes a search bar, a 'Notify' dropdown, and buttons for 'View history', 'Release change', 'Delete pipeline', and 'Create pipeline'. The 'Create pipeline' button is highlighted with a red box. Below the buttons is a table with columns for 'Name', 'Most recent execution', 'Latest source revisions', and 'Last executed'. The table currently shows 'No results' with the message 'There are no results to display.'

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.

The screenshot shows the AWS CodePipeline console interface for creating a new pipeline. The breadcrumb navigation is 'Developer Tools > CodePipeline > Pipelines > Create new pipeline'. The main heading is 'Choose pipeline settings' with an 'Info' link. A left-hand sidebar lists five steps: 'Step 1: Choose pipeline settings' (active), '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 'Pipeline settings' and contains the following fields and options:

- Pipeline name:** A text input field containing 'DemoPipeline'. A red box highlights this field. A note below it says 'Enter the pipeline name. You cannot edit the pipeline name after it is created.' and 'No more than 100 characters'.
- Service role:** Two radio button options: 'New service role' (selected) with the subtext 'Create a service role in your account', and 'Existing service role' with the subtext 'Choose an existing service role from your account'.
- Role name:** A text input field containing 'AWSCodePipelineServiceRole-us-east-1-DemoPipeline'. A note below it says 'Type your service role name'.
- Permissions:** A checked checkbox labeled 'Allow AWS CodePipeline to create a service role so it can be used with this new pipeline'.
- Advanced settings:** A section with a right-pointing arrow and the text 'Advanced settings'.
- Navigation:** 'Cancel' and 'Next' buttons at the bottom right. The 'Next' button is highlighted with a red box.

The footer of the console shows 'Feedback', a link for language selection, '© 2022, Amazon Web Services, Inc. or its affiliates.', and links for 'Privacy', 'Terms', and '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

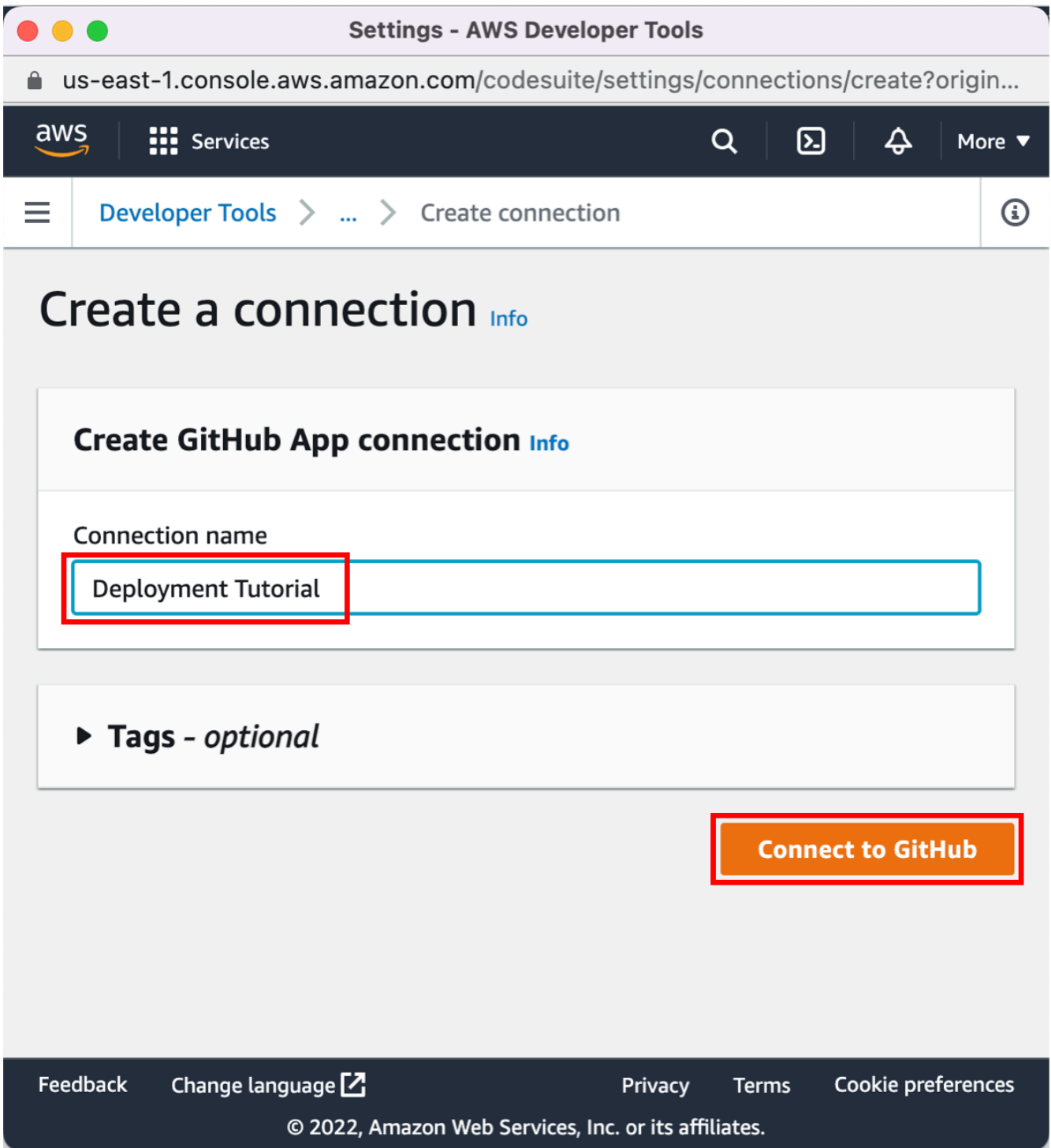
Select **GitHub (Version 2)** for the **Source provider**.

Choose **Connect to GitHub**.

The screenshot shows the AWS CodePipeline console interface for adding a source stage. The breadcrumb navigation is 'Developer Tools > CodePipeline > Pipelines > Create new pipeline'. The left sidebar shows a progress indicator for 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 main content area is titled 'Add source stage' with an 'Info' link. Under the 'Source' section, the 'Source provider' is set to 'GitHub (Version 2)'. A callout box titled 'New GitHub version 2 (app-based) action' explains that a connection must be created. Below this, the 'Connection' section has a search field and a 'Connect to GitHub' button. The 'Repository name' section has a search field and a placeholder '<account>/<repository-name>'. The footer contains 'Feedback', a link for language selection, '© 2022, Amazon Web Services, Inc. or its affiliates.', and links for 'Privacy', 'Terms', and '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

► **Tags - optional**

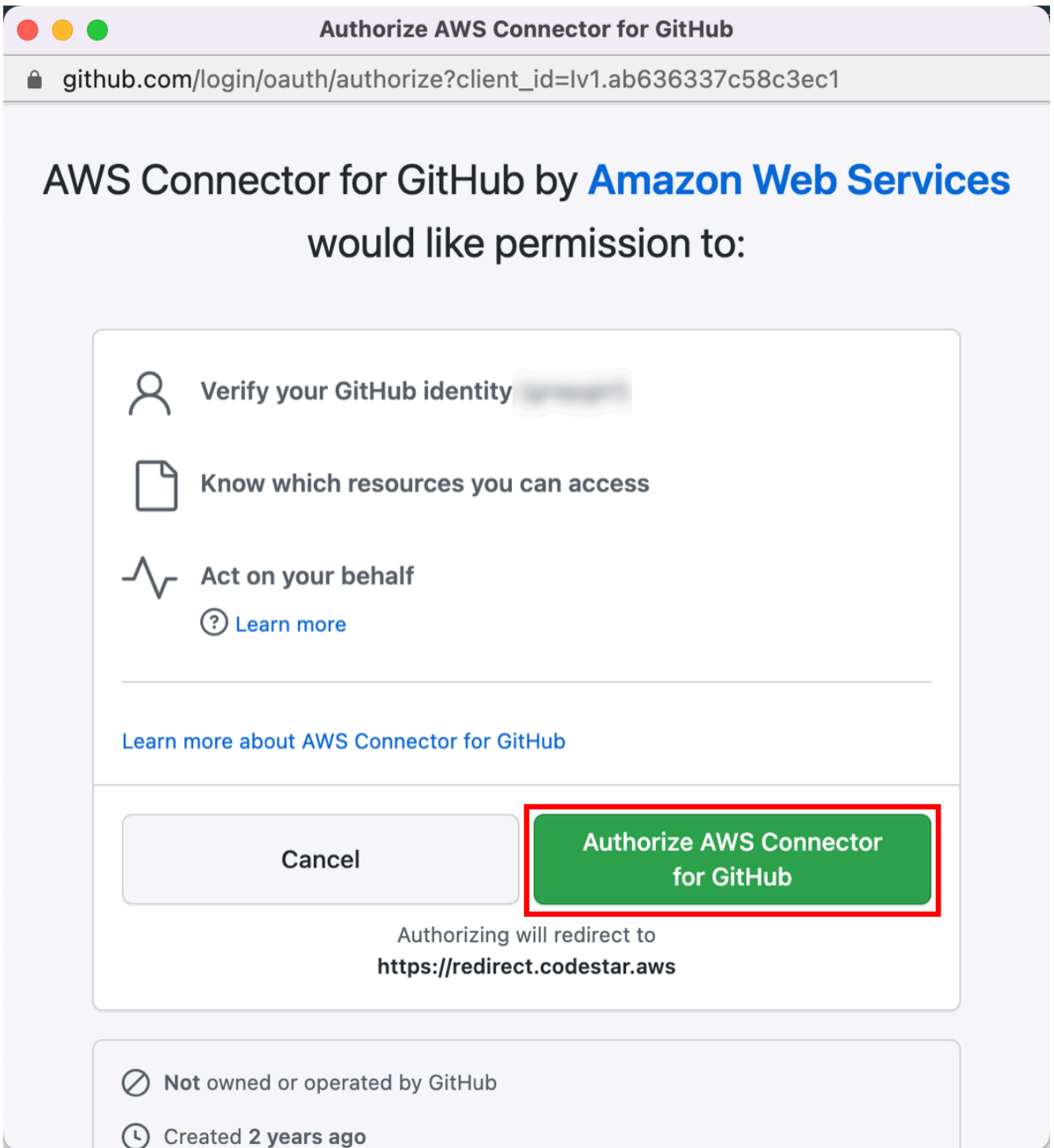
[Connect to GitHub](#)

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

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

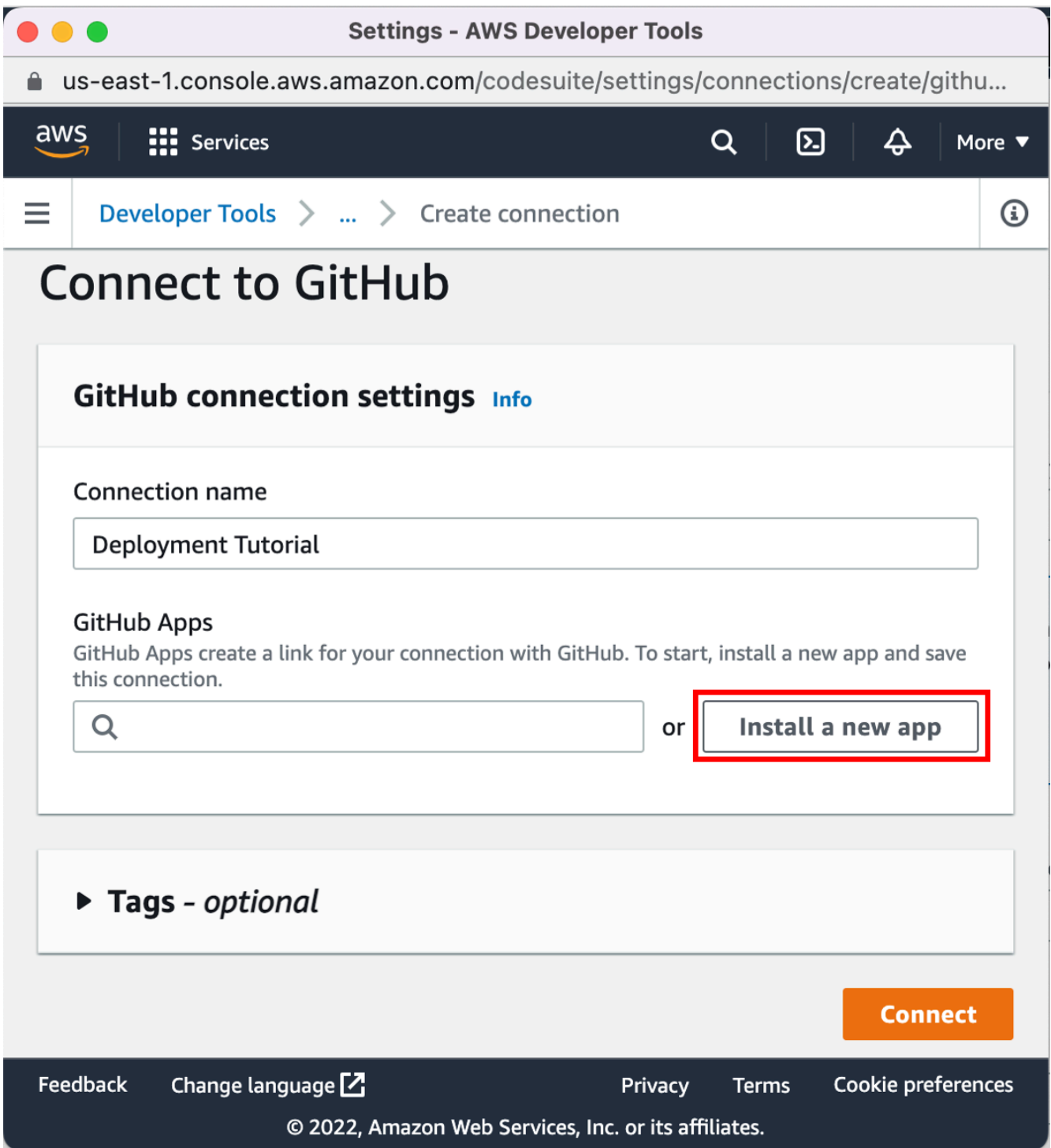
3. Grant permissions

Select **Authorize AWS Connector for GitHub**.



4. Install a new app

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



The screenshot shows the AWS Developer Tools console interface for creating a GitHub connection. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/codesuite/settings/connections/create/github...`. The page title is "Settings - AWS Developer Tools". The breadcrumb navigation shows "Developer Tools > ... > Create connection".

Connect to GitHub

GitHub connection settings [Info](#)

Connection name

GitHub Apps

GitHub Apps create a link for your connection with GitHub. To start, install a new app and save this connection.

 or **Install a new app**

► **Tags - optional**

Connect

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

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

5. Install the forked repository

On the **Install AWS Connector for GitHub** page, choose **Only select repositories** and select the **aws-codepipeline-s3-coddeploy-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

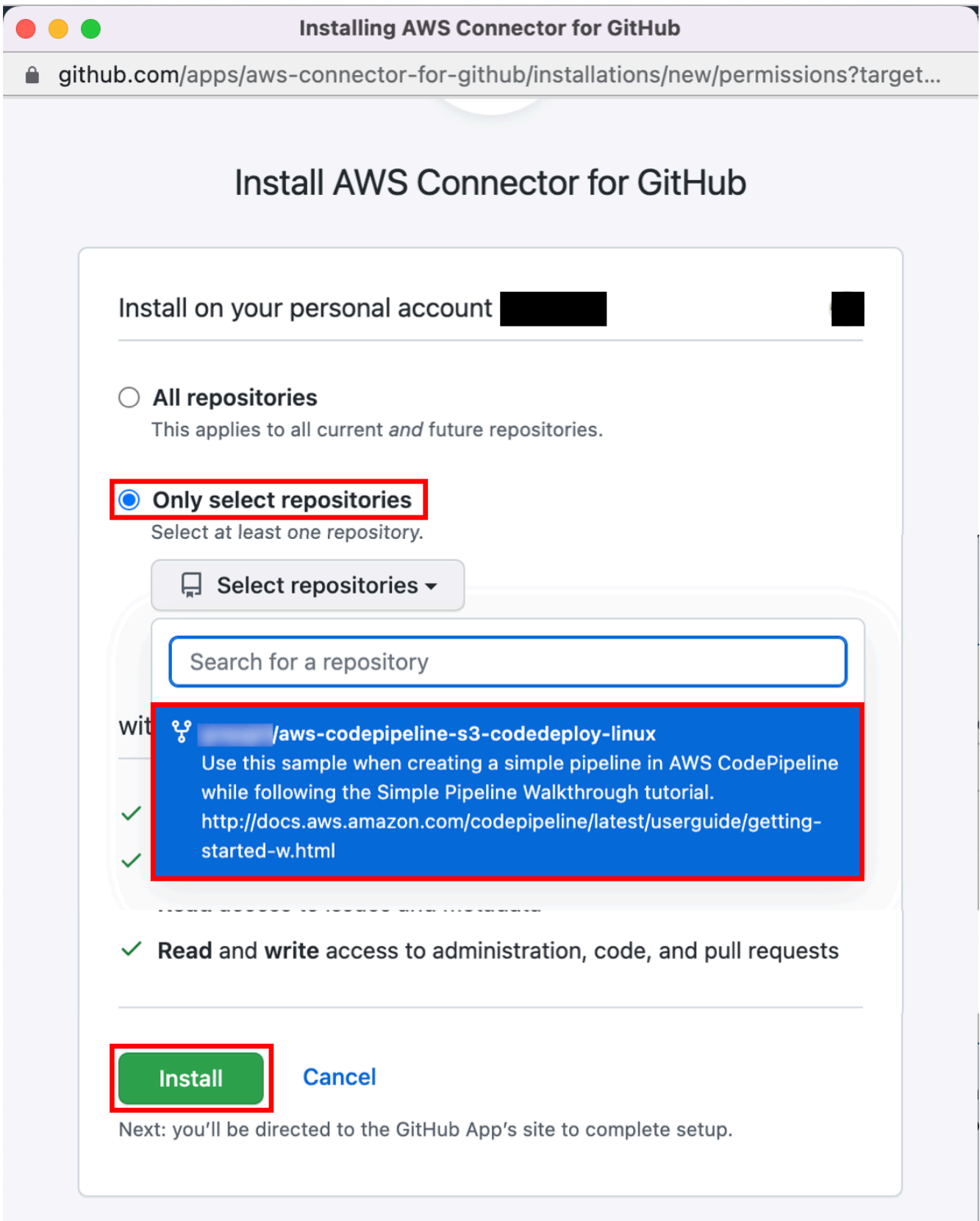
Install on your personal account [redacted]

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

Only select repositories
Select at least one repository.

Select repositories ▾

Search for a repository

wit  [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

✓
✓

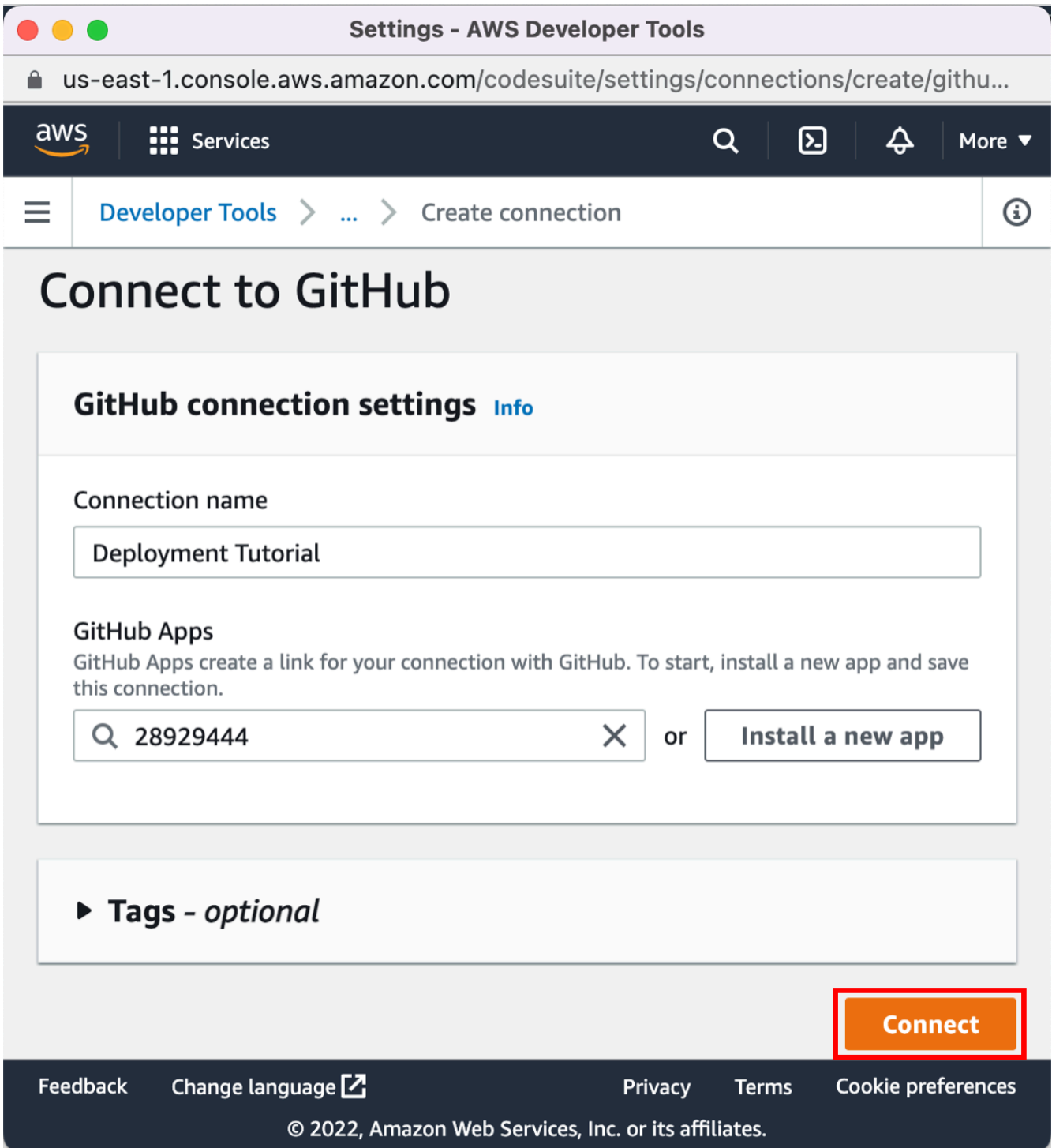
✓ Read and write access to administration, code, and pull requests

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 for creating a GitHub connection. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/codesuite/settings/connections/create/github...`. The page title is "Settings - AWS Developer Tools". The breadcrumb navigation shows "Developer Tools > ... > Create connection". The main heading is "Connect to GitHub". Below the heading is a section titled "GitHub connection settings" with an "Info" link. The "Connection name" field contains the text "Deployment Tutorial". The "GitHub Apps" section includes a description: "GitHub Apps create a link for your connection with GitHub. To start, install a new app and save this connection." Below this is a search box containing "28929444" and an "X" icon, followed by the word "or" and a button labeled "Install a new app". At the bottom right of the main content area, there is a prominent orange button labeled "Connect" which is highlighted with a red rectangular border. 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

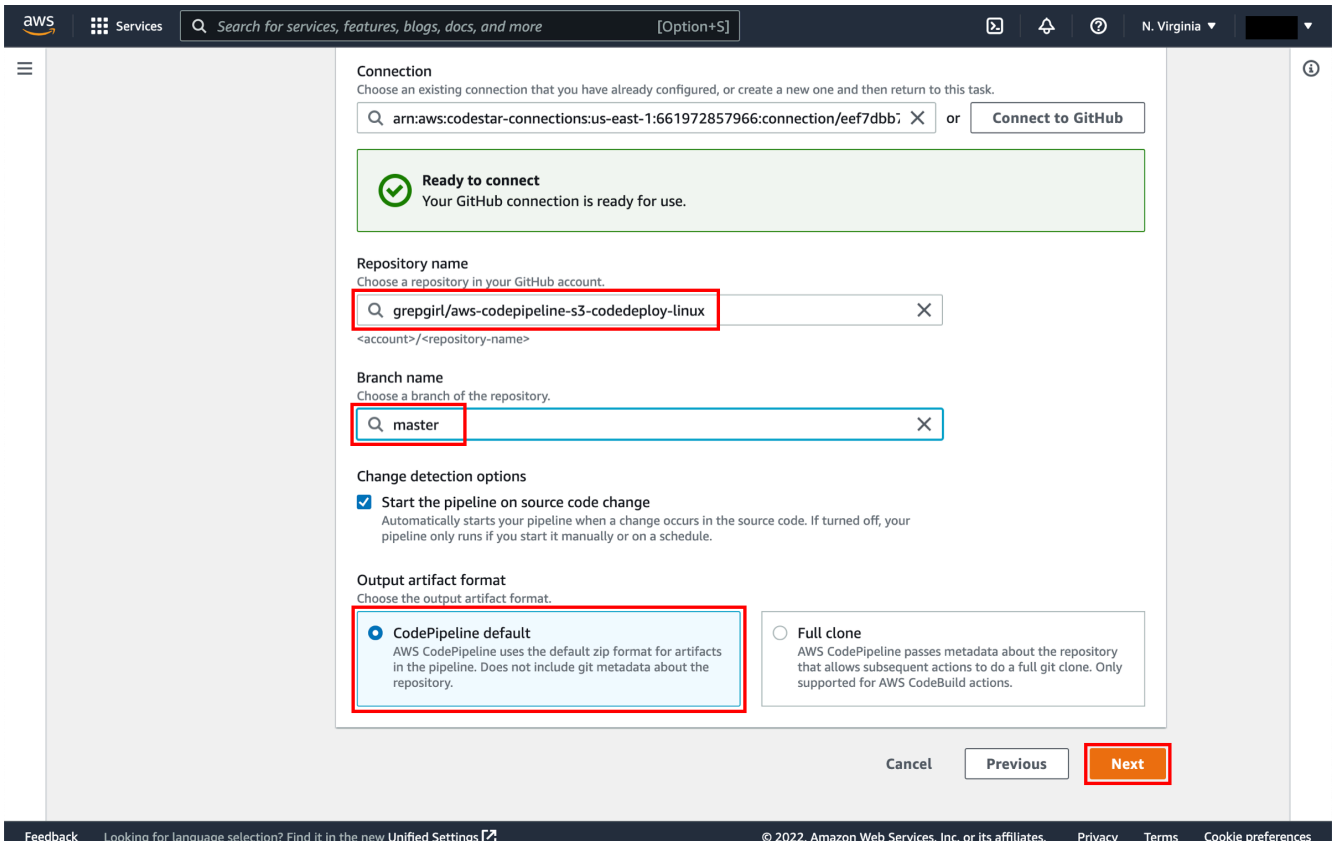
The **Add source** page will be updated to reflect GitHub is ready to connect. Specify the repository and branch:

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.

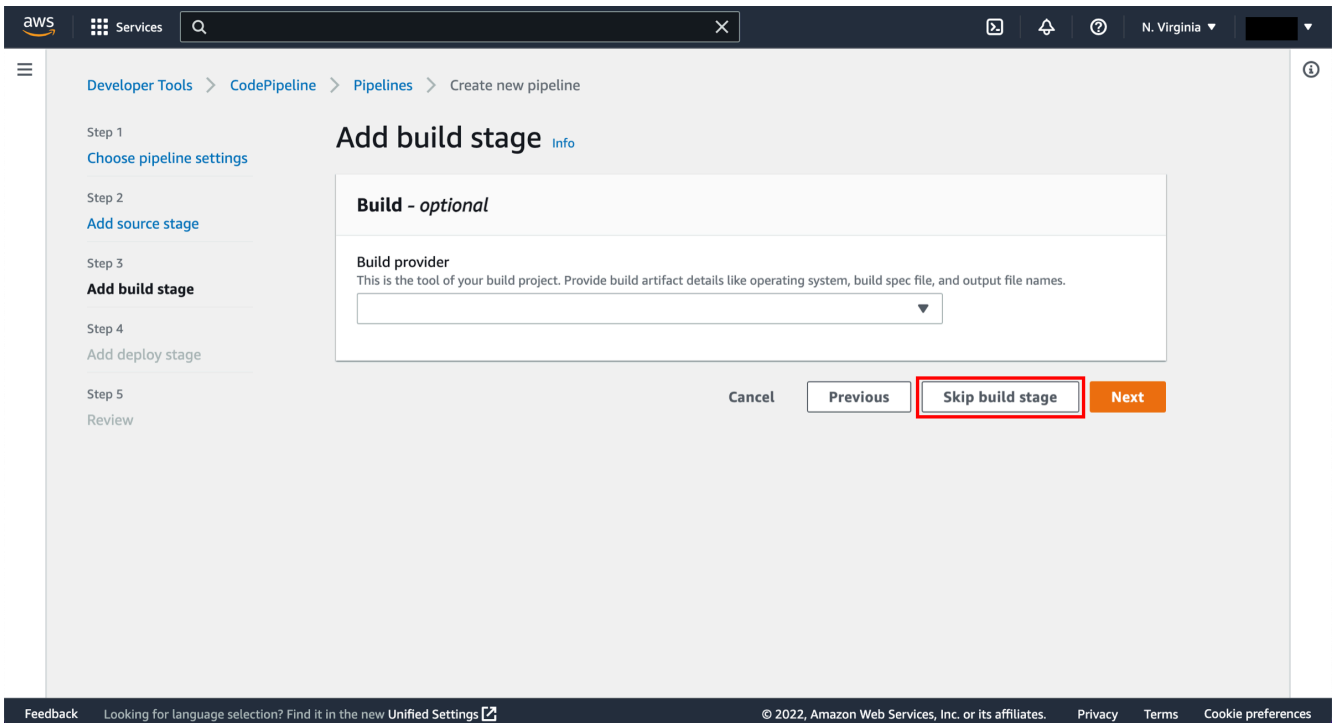


The screenshot shows the AWS CodePipeline console configuration page. The 'Connection' section is at the top, with a search bar containing an ARN and a 'Connect to GitHub' button. Below this is a green box indicating 'Ready to connect'. The 'Repository name' section has a dropdown menu with 'grepgirl/aws-codepipeline-s3-codedeploy-linux' selected. The 'Branch name' section has a dropdown menu with 'master' selected. The 'Change detection options' section has a checked box for 'Start the pipeline on source code change'. The 'Output artifact format' section has two radio buttons: 'CodePipeline default' (selected) and 'Full clone'. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons, with 'Next' being highlighted in orange.

8. Skip build stage

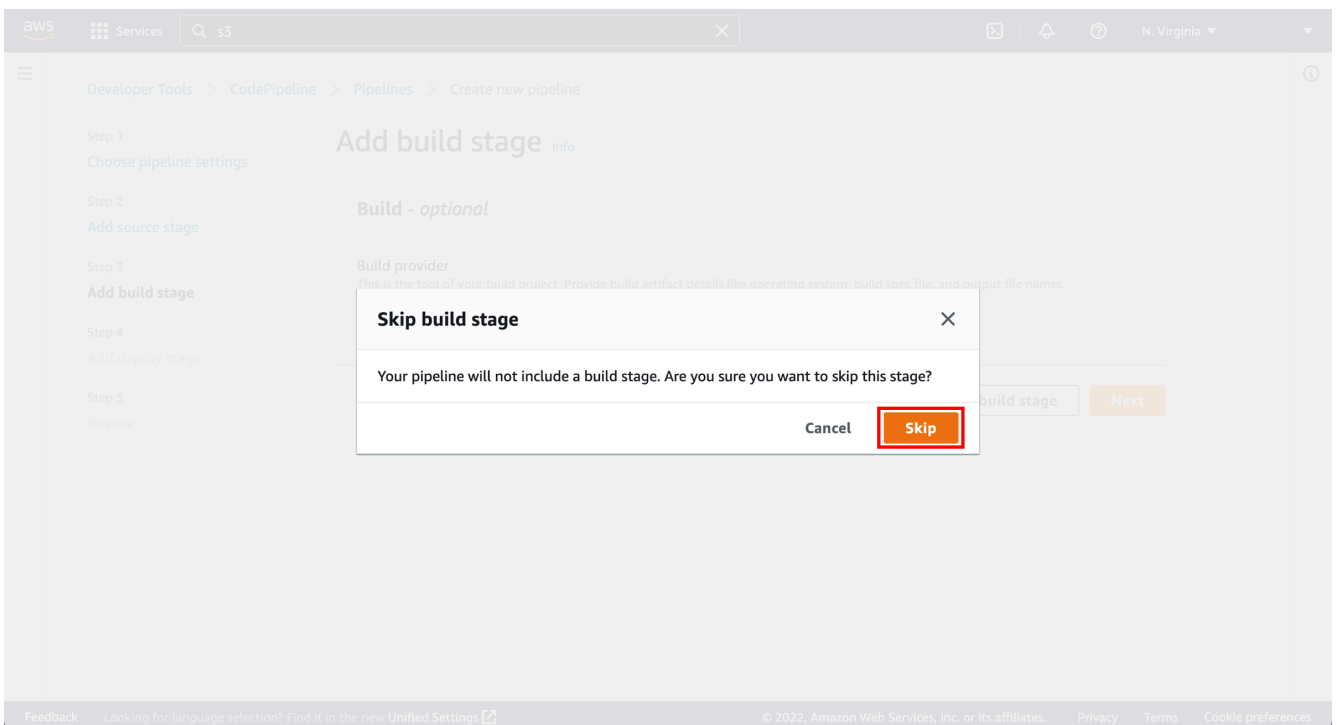
A true continuous deployment pipeline requires a build stage, where code is compiled and unit tested. CodePipeline lets you plug your preferred build provider into your pipeline. However, in this tutorial you will skip the build stage.

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**.

The screenshot shows the AWS CodePipeline console interface for adding a new deploy stage. The breadcrumb navigation is 'Developer Tools > CodePipeline > Pipelines > Create new pipeline'. The left sidebar shows a progress indicator with five steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage - currently active), and Step 5 (Review). The main content area is titled 'Add deploy stage' and includes an information box stating: '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 with the following fields: 'Deploy provider' (set to 'AWS Elastic Beanstalk'), 'Region' (set to 'US East (N. Virginia)'), 'Application name' (set to 'Deployment Tutorial'), and 'Environment name' (set to 'Deploymenttutorial-env'). At the bottom right of the configuration area are 'Cancel', 'Previous', and 'Next' buttons, with 'Next' highlighted in orange. The footer contains a feedback link, a language selection notice, and copyright information for Amazon Web Services, Inc. (© 2022).

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 is 'Developer Tools > CodePipeline > Pipelines > Create new pipeline'. The left sidebar shows a progress indicator with 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 main content area is titled 'Add source stage' and contains a 'Source' configuration form. The form has the following fields and options:

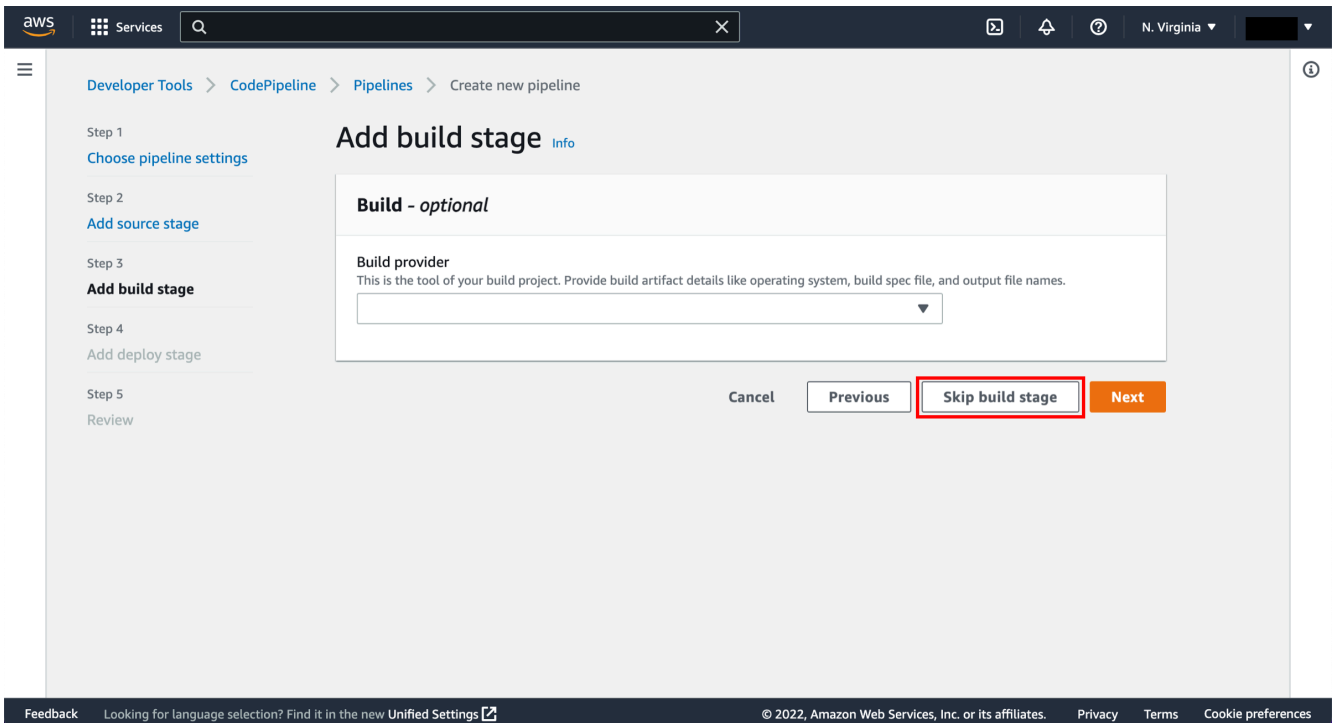
- Source provider:** A dropdown menu with 'Amazon S3' selected.
- Bucket:** A text input field containing 'awscodepipeline-demobucket-8302022'.
- S3 object key:** A text input field containing 'aws-codepipeline-s3-aws-codedeploy_linux.zip'.
- Change detection options:** Two radio button options: 'Amazon CloudWatch Events (recommended)' (selected) and 'AWS CodePipeline'.

At the bottom of the form are three buttons: 'Cancel', 'Previous', and 'Next'.

2. Skip build stage

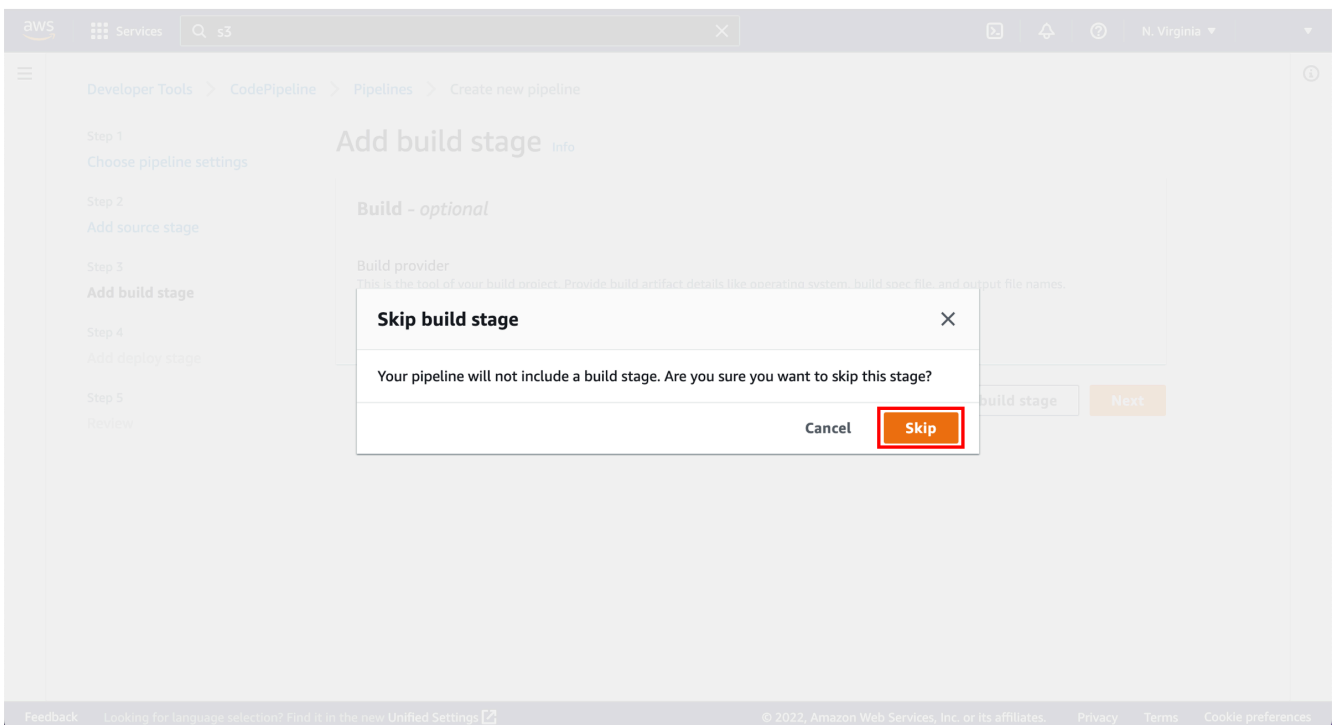
A true continuous deployment pipeline requires a build stage, where code is compiled and unit tested. CodePipeline lets you plug your preferred build provider into your pipeline. However, in this tutorial you will skip the build stage.

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 deploy stage. The breadcrumb navigation is 'Developer Tools > CodePipeline > Pipelines > Create new pipeline'. The left sidebar shows a step list: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage - selected), Step 5 (Review). The main content area is titled 'Add deploy stage' and includes an information box stating: '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 contains the following fields: 'Deploy provider' (set to 'AWS Elastic Beanstalk'), 'Region' (set to 'US East (N. Virginia)'), 'Application name' (set to 'Deployment Tutorial'), and 'Environment name' (set to 'Deploymenttutorial-env'). At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons, with 'Next' highlighted in orange. The footer contains 'Feedback', a language selection link, '© 2022, Amazon Web Services, Inc. or its affiliates.', and links for 'Privacy', 'Terms', and 'Cookie preferences'.

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 AWS CodePipeline console interface for configuring a new pipeline. The breadcrumb navigation is 'Developer Tools > CodePipeline > Pipelines > Create new pipeline'. The left sidebar shows the pipeline configuration 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 contains the following configuration options:

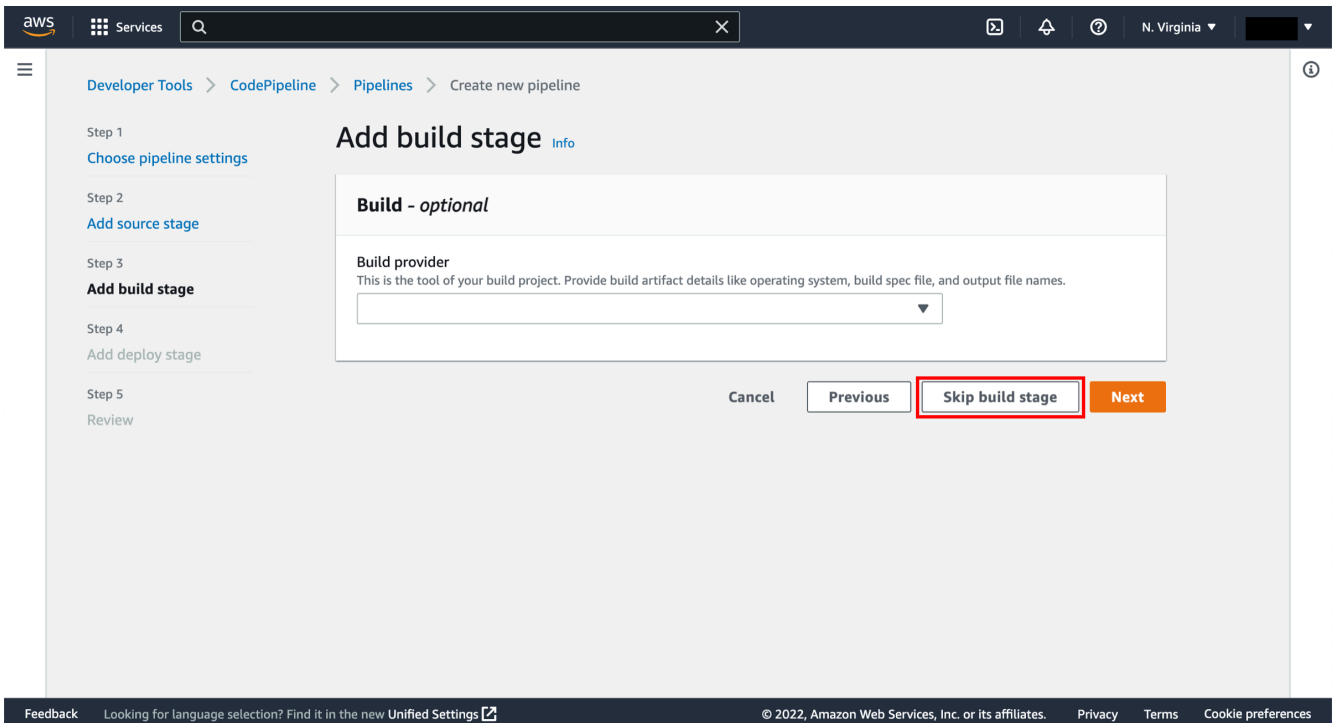
- Source provider:** AWS CodeCommit (selected)
- Repository name:** PipelineRepo (selected)
- Branch name:** main (selected)
- Change detection options:** Amazon CloudWatch Events (recommended) (selected), AWS CodePipeline (not selected)
- Output artifact format:** CodePipeline default (selected), Full clone (not selected)

At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'. The 'Next' button is highlighted in orange, indicating the next step in the process.

2. Skip build stage

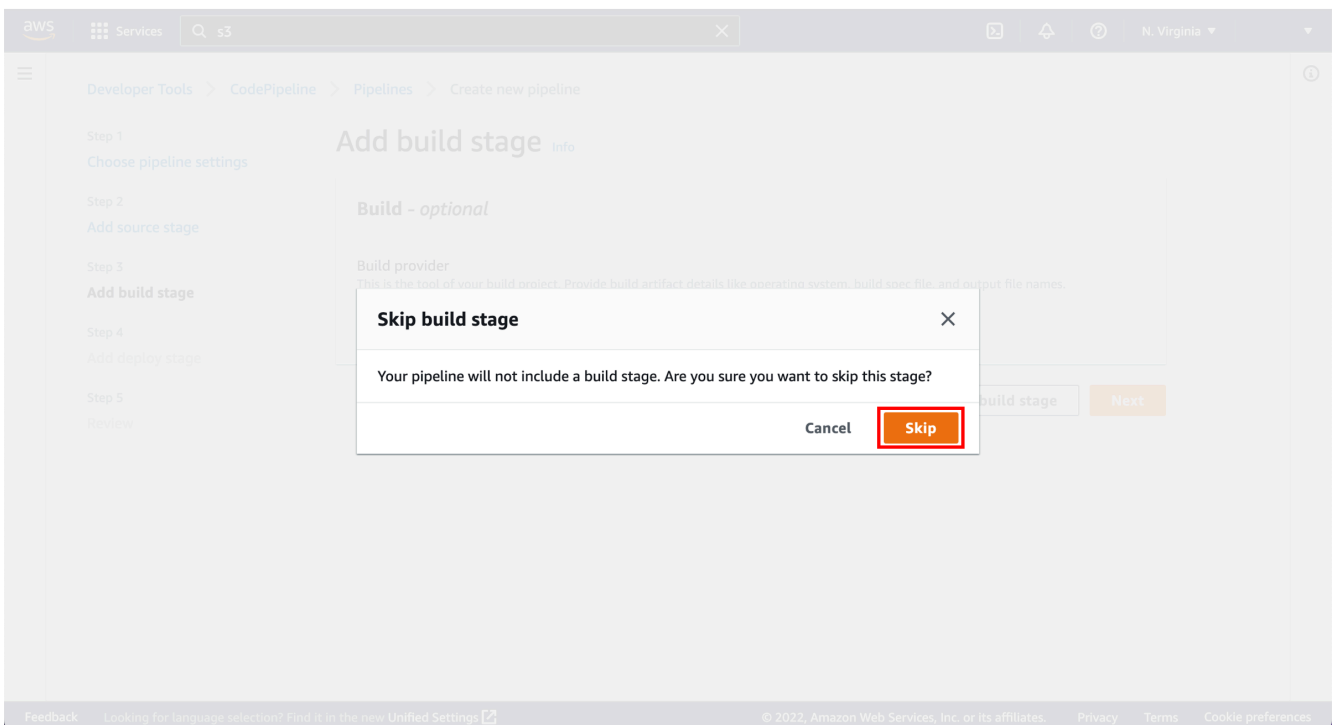
A true continuous deployment pipeline requires a build stage, where code is compiled and unit tested. CodePipeline lets you plug your preferred build provider into your pipeline. However, in this tutorial you will skip the build stage.

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 pipeline. The breadcrumb navigation is 'Developer Tools > CodePipeline > Pipelines > Create new pipeline'. The left sidebar shows a progress indicator with five steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage - currently active), and Step 5 (Review). The main content area is titled 'Add deploy stage' and includes an information box stating: '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' section contains the following fields: 'Deploy provider' (AWS Elastic Beanstalk), 'Region' (US East (N. Virginia)), 'Application name' (Deployment Tutorial), and 'Environment name' (Deploymenttutorial-env). At the bottom right, there 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', a search bar, and a region dropdown set to '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 main content area is titled 'Review' and contains four sections, each with a title and a list of configuration details:

- 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

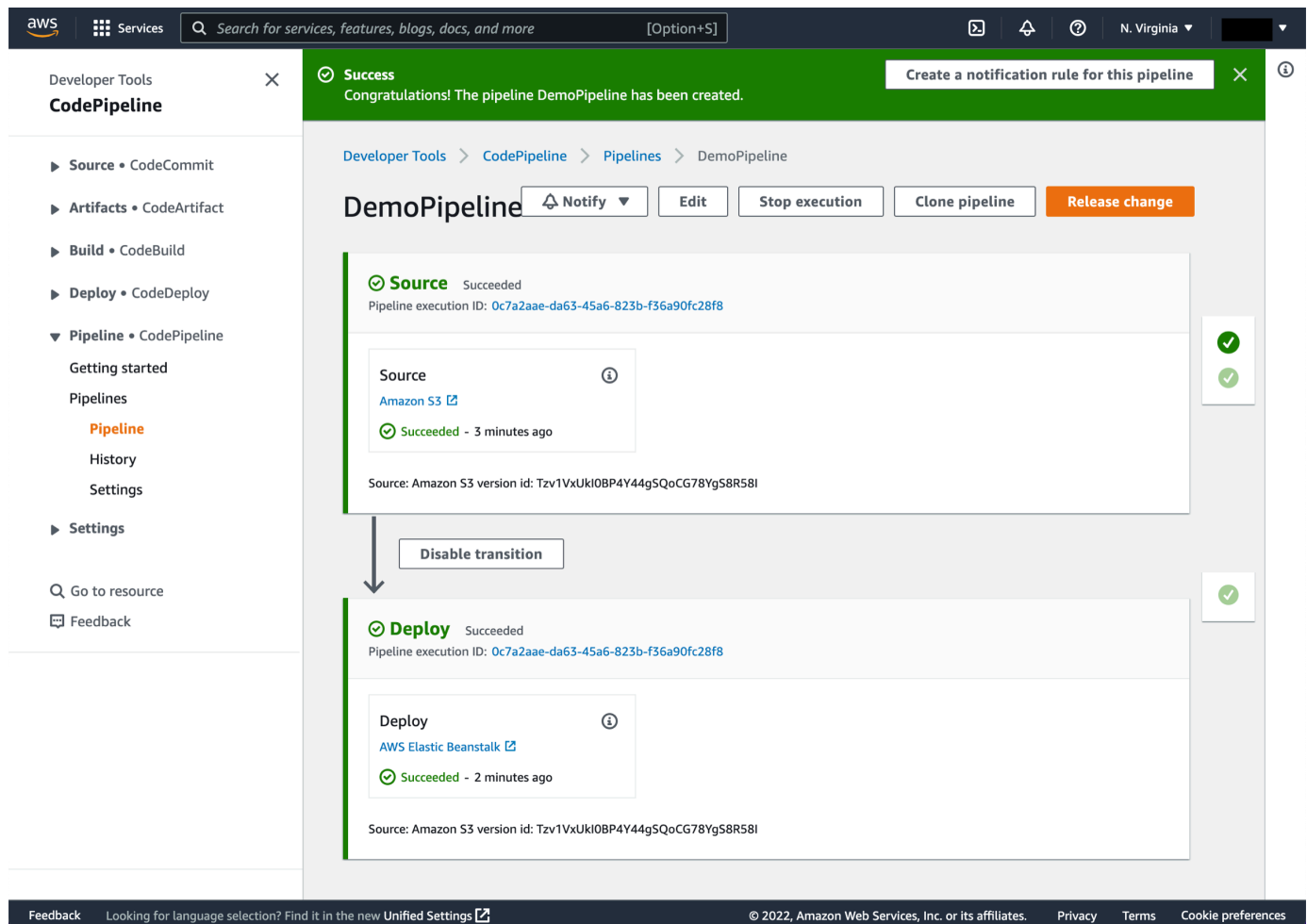
At the bottom right of the main content area, there are three buttons: 'Cancel', 'Previous', and 'Create pipeline' (which is highlighted with a red border).

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. 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 a success message: "Success Congratulations! The pipeline DemoPipeline has been created." Below this, the breadcrumb navigation shows "Developer Tools > CodePipeline > Pipelines > DemoPipeline". The main content area shows the pipeline "DemoPipeline" with a "Notify" dropdown, "Edit", "Stop execution", "Clone pipeline", and "Release change" buttons. The pipeline execution is shown as successful, with two stages: "Source" and "Deploy". Both stages are marked as "Succeeded" with green checkmarks. The "Source" stage used "Amazon S3" and "Succeeded - 3 minutes ago". The "Deploy" stage used "AWS Elastic Beanstalk" and "Succeeded - 2 minutes ago". A "Disable transition" button is visible between the stages. The footer contains "Feedback", "Looking for language selection? Find it in the new Unified Settings", and "© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

3. Select Elastic Beanstalk

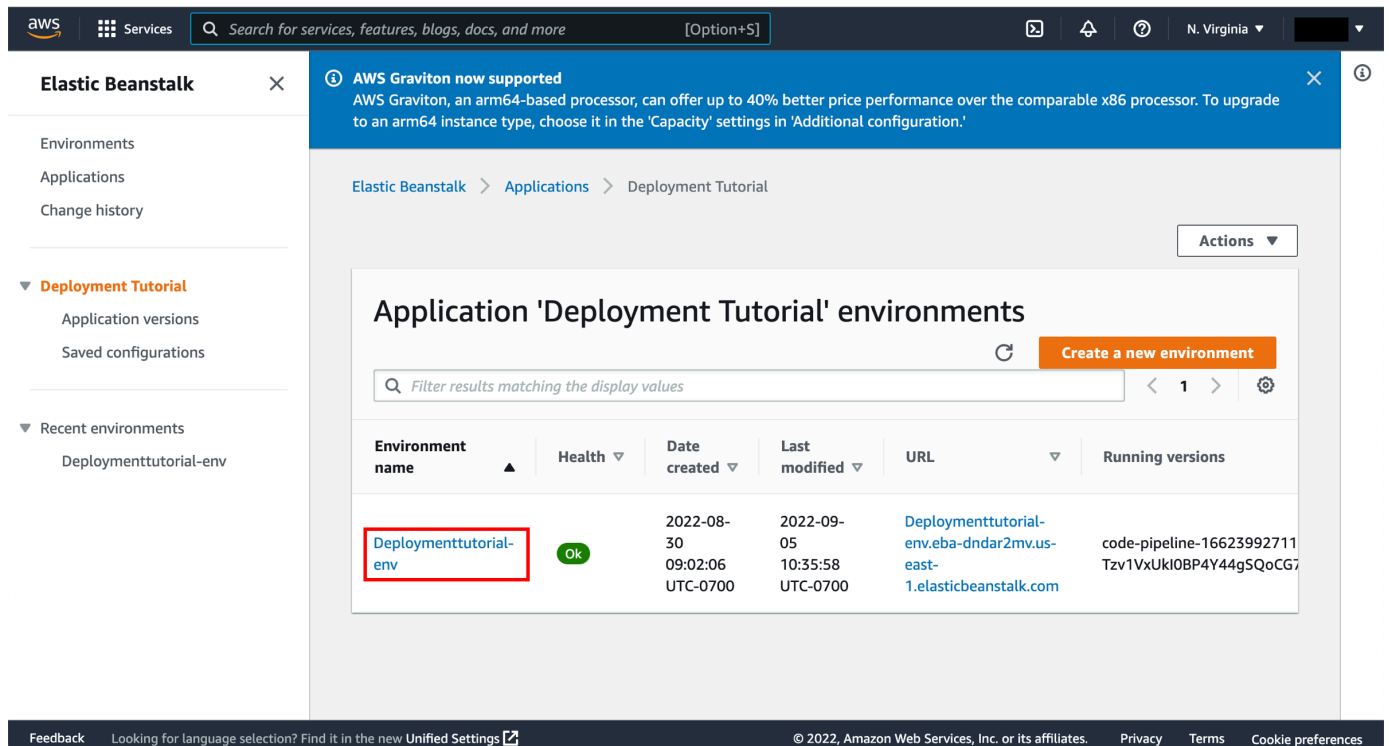
In the status area for the Beta stage, select **AWS Elastic Beanstalk**.

The screenshot displays the AWS CodePipeline console interface. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and the region 'N. Virginia'. The main content area shows the 'DemoPipeline' details. The pipeline consists of two stages: 'Source' and 'Deploy', both of which have succeeded. The 'Source' stage uses 'Amazon S3' as the provider, and the 'Deploy' stage uses 'AWS Elastic Beanstalk'. A 'Disable transition' button is visible between the stages. The left sidebar contains navigation options for 'Developer Tools' and 'CodePipeline', with 'Pipeline' selected under 'CodePipeline'. The bottom footer includes 'Feedback', a language selection prompt, and copyright information for Amazon Web Services, Inc. (© 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. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and the region 'N. Virginia'. A blue notification banner at the top right states 'AWS Graviton now supported'. The left sidebar shows the 'Elastic Beanstalk' menu with options for 'Environments', 'Applications', and 'Change history'. Under 'Deployment Tutorial', there are 'Application versions', 'Saved configurations', and 'Recent environments' (listing 'Deploymenttutorial-env'). The main content area is titled 'Application 'Deployment Tutorial' environments' and features a 'Create a new environment' button and a search filter. 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 10:35:58 UTC-0700	Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com	code-pipeline-16623992711 Tzv1VxUki0BP4Y44gSQoCG7

At the bottom of the console, there is a footer with 'Feedback', a link for language selection, and copyright information for Amazon Web Services, Inc. or its affiliates, along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

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 interface. At the top, there's a navigation bar with the AWS logo, 'Services' menu, a search bar, and regional settings for 'N. Virginia'. A blue notification banner at the top right states 'AWS Graviton now supported'. The main content area shows the 'Deploymenttutorial-env' environment details. A red box highlights the environment URL: `Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com`. Below this, the 'Health' section shows a green checkmark and 'Ok' status. The 'Running version' section shows the version ID 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 with a warning 'Different version recommended'. A 'Recent events' section is partially visible at the bottom.

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.

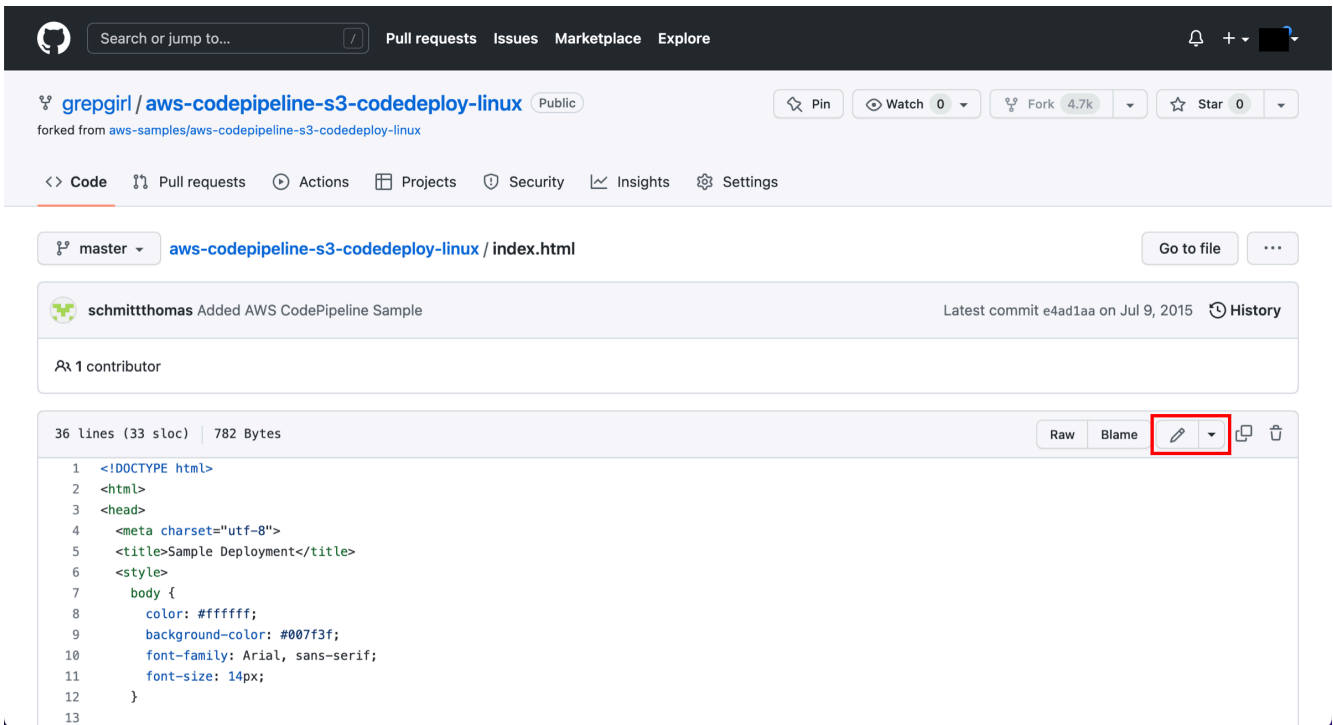
GitHub

1. Edit the code

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

Open **index.html**.

Select the **Edit icon**.



The screenshot shows a GitHub repository page for 'aws-codepipeline-s3-coddeploy-linux' by user 'greggirl'. The repository is public and has 0 stars and 0 forks. The file 'index.html' is selected, showing its content. The file was added by 'schmittthomas' on Jul 9, 2015. The code content is as follows:

```
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
```

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 <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 GitHub and deployed it to one Amazon EC2 instance using AWS E
31   <p>For next steps, read the AWS CodePipeline Documentation.</p>
32 </div>
33 </body>
34 </html>
35
```

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 E
31   <p>For next steps, read the AWS CodePipeline Documentation.</p>
32 </div>
33 </body>
34 </html>
35
```

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

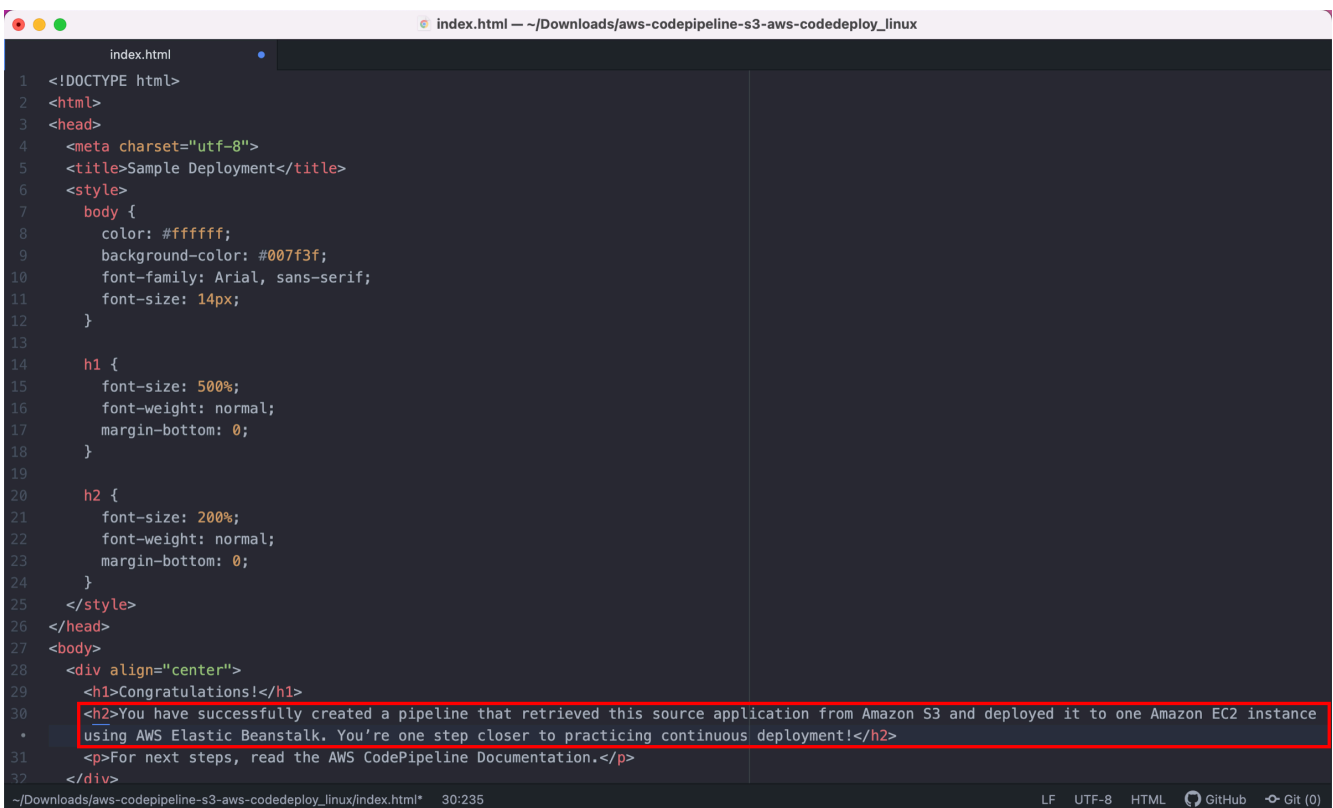
On your desktop, visit the zip file you downloaded called **aws-codepipeline-s3-aws-codedeploy_linux.zip**.

Edit the sample web app code:

- a. Extract index.html from the zip file and open it using your preferred text editor.
- b. 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!

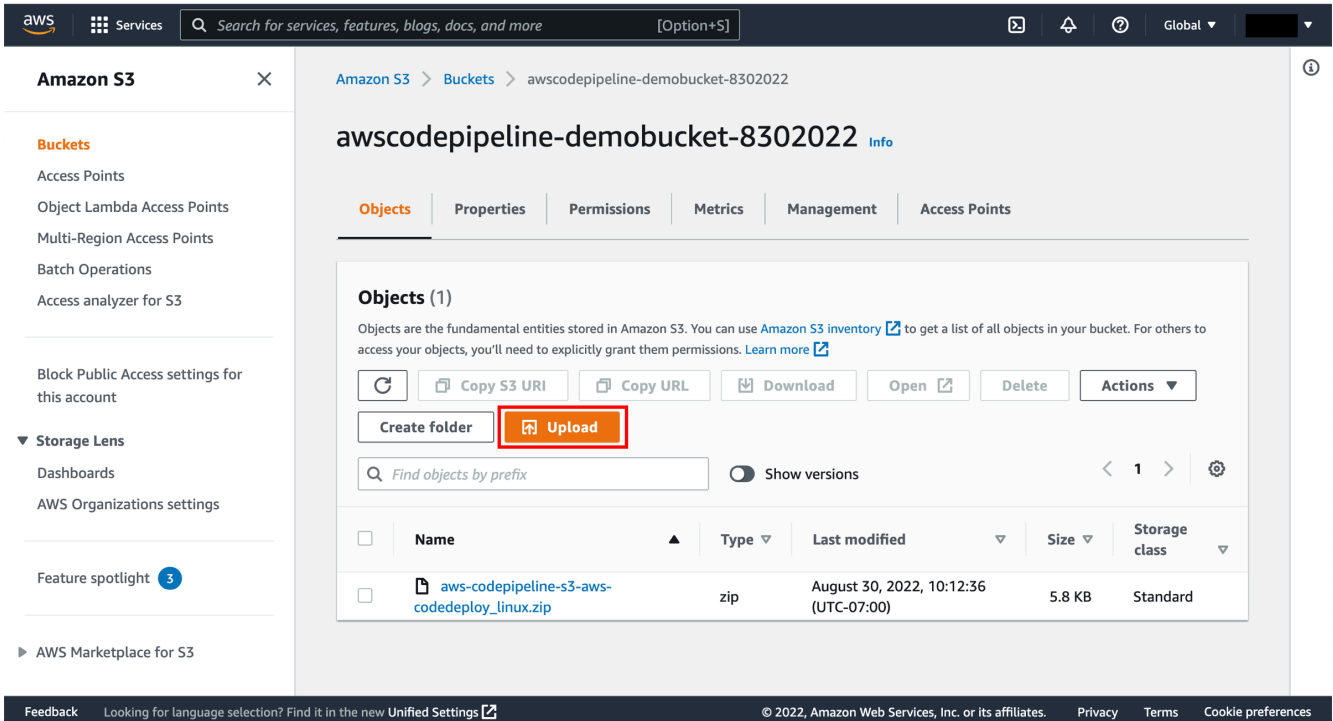
- c. Copy the updated index.html file back into **aws-codepipeline-s3-aws-codedeploy_linux.zip** and replace the older version of index.html.



```
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 <p>For next steps, read the AWS CodePipeline Documentation.</p>
33 </div>
```

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.

The screenshot shows the AWS S3 console's 'Upload' page. At the top, there's a search bar and navigation links. Below that, a message says 'Drag and drop files and folders you want to upload here, or choose Add files, or Add folders.' A table titled 'Files and folders (1 Total, 5.8 KB)' shows one file: 'aws-codepipeline-s3-aws-codedeploy_linux.zip' with a size of 5.8 KB. Below the table, the 'Destination' is set to 's3://awscodepipeline-demobucket-8302022'. At the bottom right, there are 'Cancel' and 'Upload' buttons, with the 'Upload' button highlighted by a red box.

AWS CodeCommit

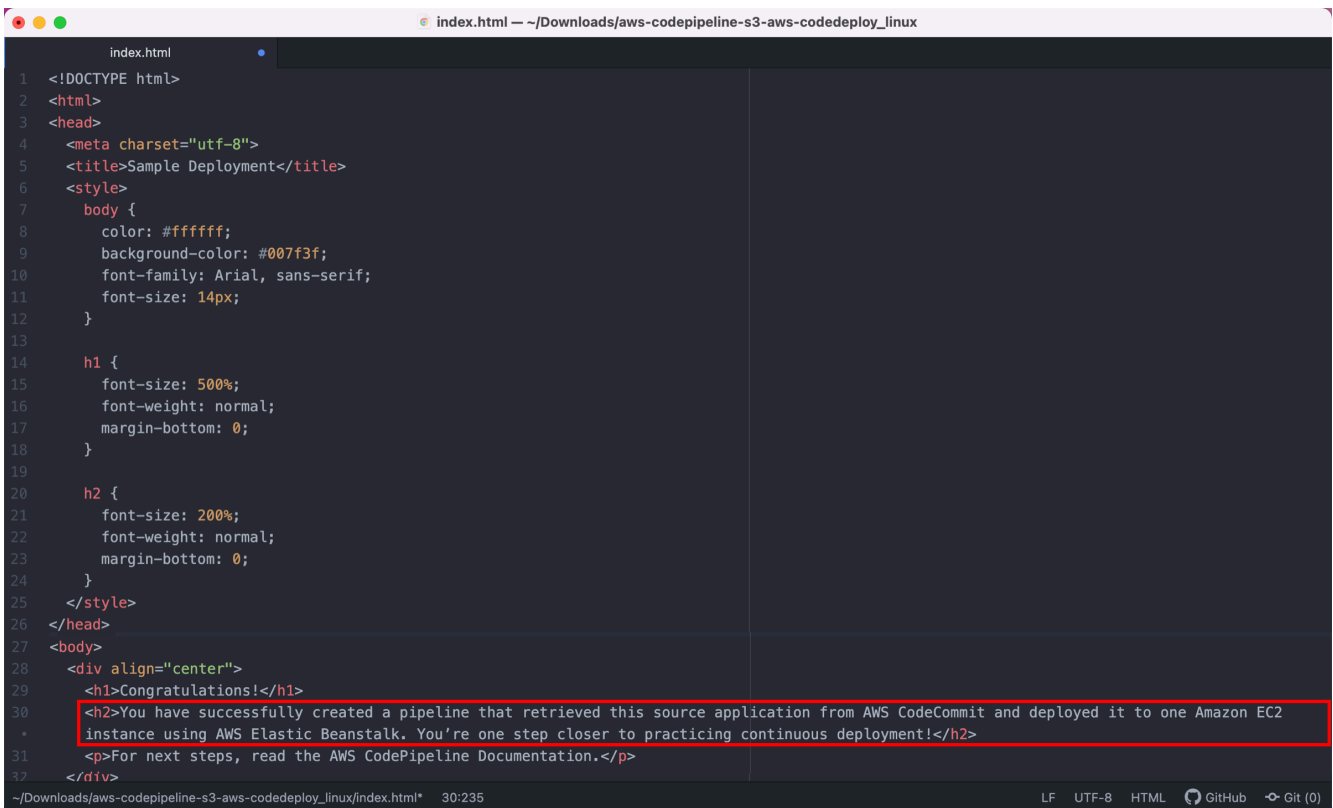
1. Edit the code

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:

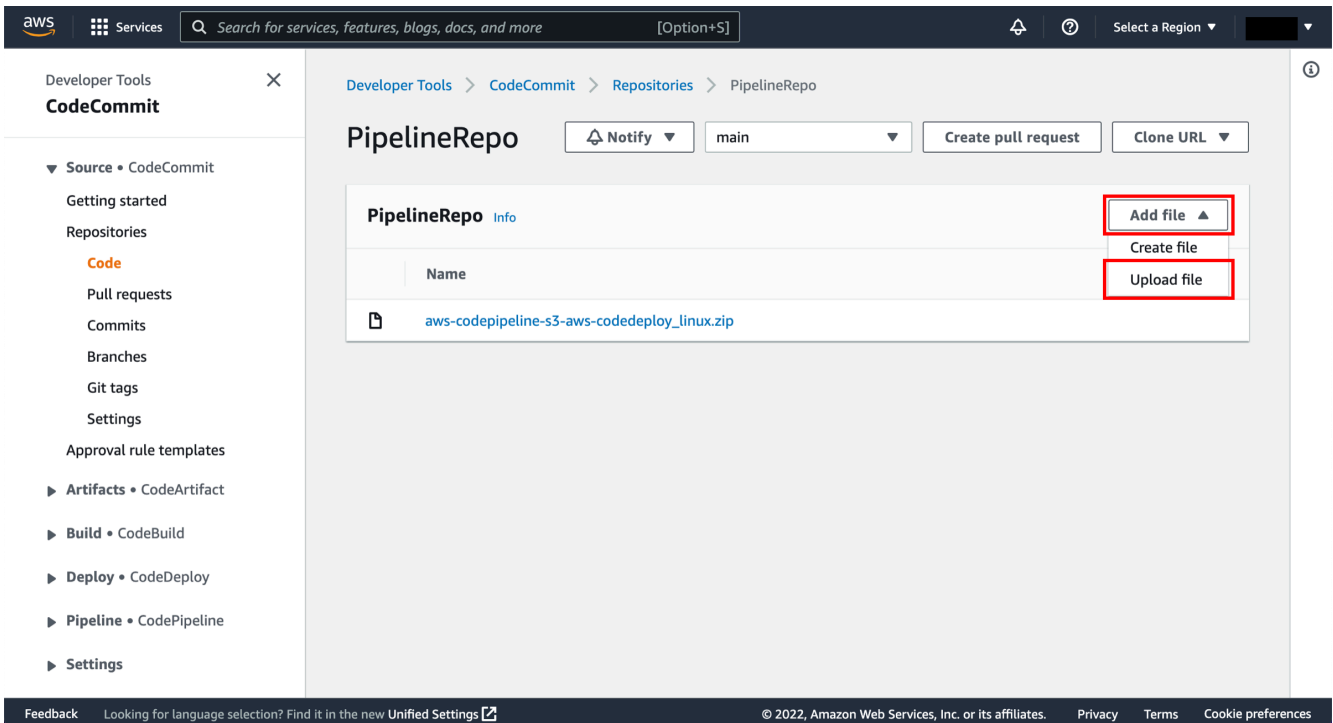
You have successfully created a pipeline that retrieved this source application from AWS CodeCommit and deployed it to one Amazon EC2 instance using AWS Elastic Beanstalk. You're one step closer to practicing continuous deployment!



```
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

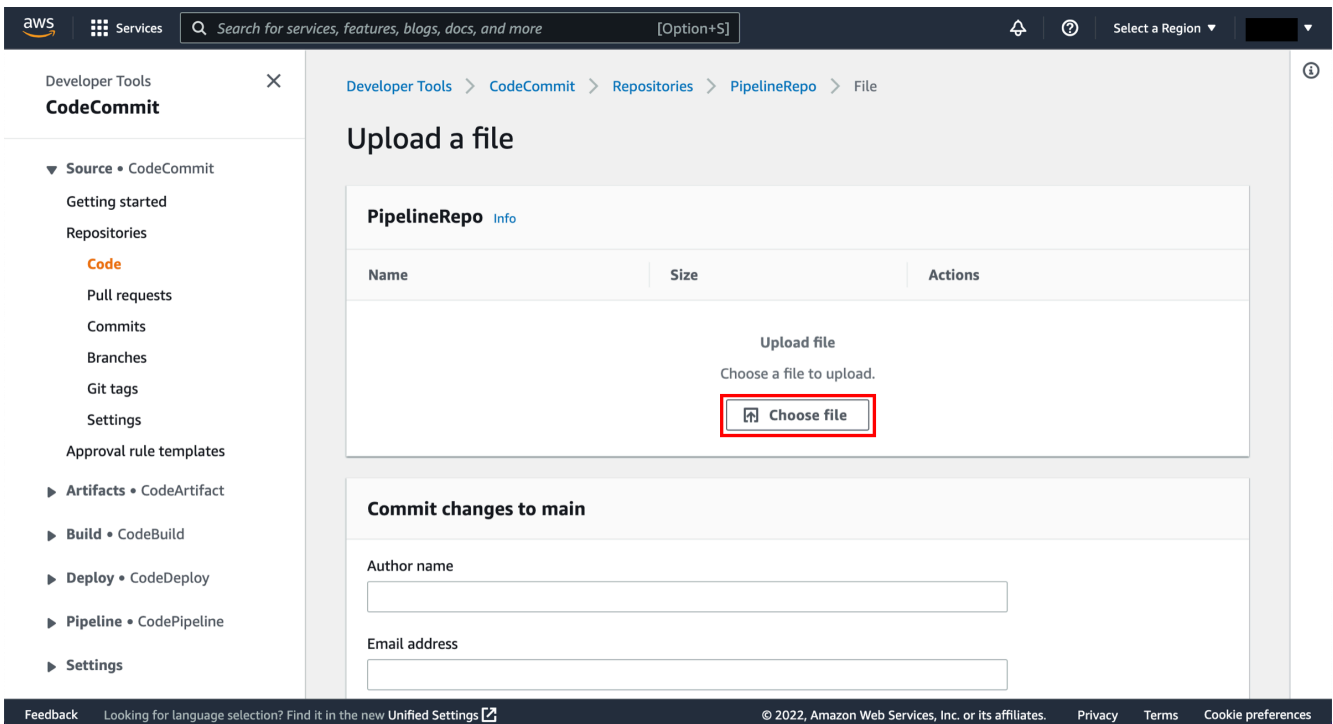
From the **CodeCommit PipelineRepo** page, choose **Add file** and select **Upload file**.



The screenshot shows the AWS CodeCommit PipelineRepo interface. The left sidebar contains a navigation menu with options like Source, Artifacts, Build, Deploy, Pipeline, and Settings. The main content area displays the PipelineRepo details for the 'main' branch. A table lists the repository files, with one file highlighted: 'aws-codepipeline-s3-aws-codedeploy_linux.zip'. To the right of the table, there are three buttons: 'Add file', 'Create file', and 'Upload file', all of which are highlighted with red boxes.

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.



The screenshot shows the 'Upload a file' page in the AWS CodeCommit PipelineRepo interface. The main content area features a table with columns for 'Name', 'Size', and 'Actions'. Below the table, there is a section titled 'Upload file' with the text 'Choose a file to upload.' and a button labeled 'Choose file', which is highlighted with a red box. Below this, there is a section titled 'Commit changes to main' with input fields for 'Author name' and 'Email address'.

4. Commit changes

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

The screenshot shows the AWS CodeCommit 'Upload a file' interface. The breadcrumb navigation is 'Developer Tools > CodeCommit > Repositories > PipelineRepo > File'. The file being uploaded is 'aws-codepipeline-s3-aws-codedeploy_linux.zip' (6 KB). The 'Commit changes to main' section has the following fields:

- Author name: AWS User
- Email address: aws-user@amazon.com
- Commit message - optional: (empty)

At the bottom right, there are 'Cancel' and 'Commit changes' buttons. The 'Commit changes' button is highlighted with a red border.

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 displays the AWS CodePipeline console interface. On the left, a navigation sidebar lists various developer tools and pipeline-related options, with 'Pipeline' selected. The main content area shows the 'DemoPipeline' execution details. At the top, there are buttons for 'Notify', 'Edit', 'Stop execution', 'Clone pipeline', and 'Release change'. Below this, the pipeline execution is shown as 'Succeeded' with a green checkmark. The pipeline execution ID is 'b3115f96-4ec4-4c70-90ce-228bb7747b82'. The pipeline consists of two stages: 'Source' and 'Deploy'. The 'Source' stage is also 'Succeeded' and used 'Amazon S3' as the provider. The 'Deploy' stage is 'Succeeded' and used 'AWS Elastic Beanstalk' as the provider. The 'AWS Elastic Beanstalk' link in the Deploy stage is highlighted with a red box. A 'Disable transition' button is visible between the two stages. The footer of the console shows 'Feedback', a language selection prompt, 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. The left sidebar is open to 'Elastic Beanstalk' > 'Applications' > 'Deployment Tutorial'. The main content area displays 'Application 'Deployment Tutorial' environments'. A table lists the 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 'URL' column contains the link to the application.

3. Select the URL

Select the URL to view the sample website again.

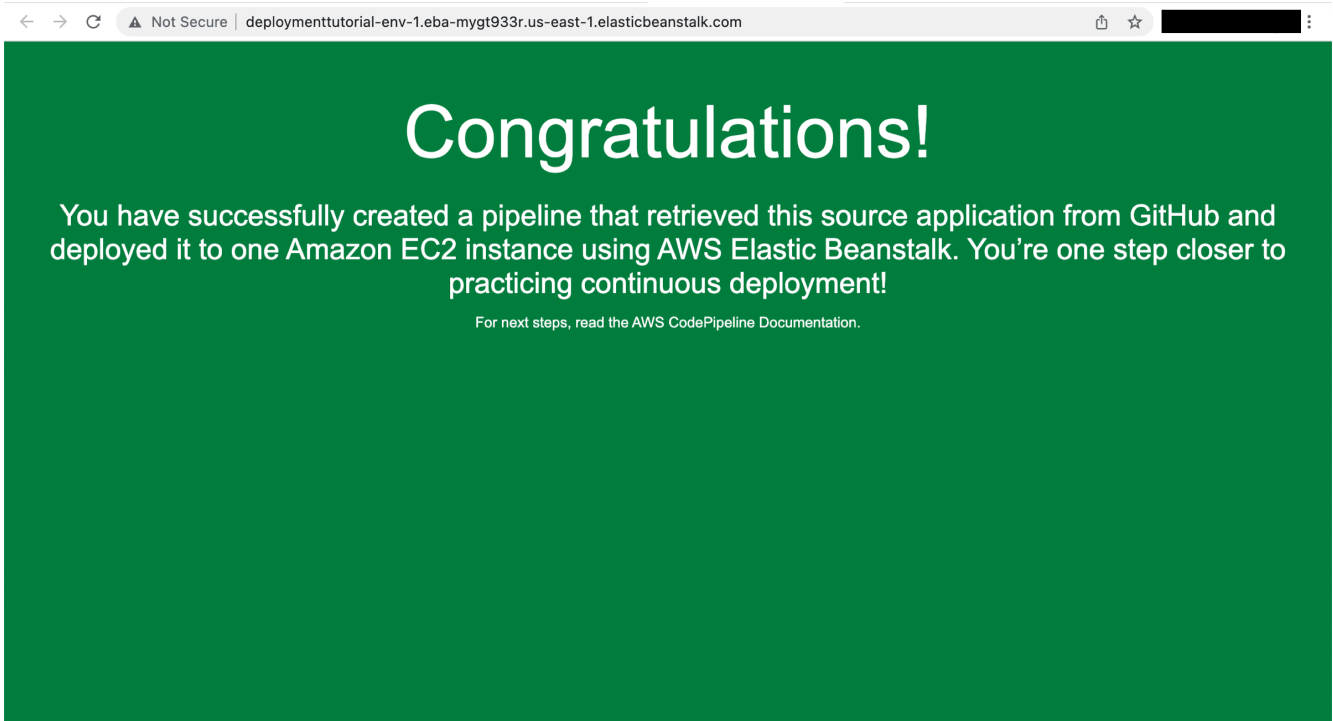
The screenshot shows the AWS Elastic Beanstalk console with the 'Deploymenttutorial-env' environment selected. The breadcrumb navigation is 'Elastic Beanstalk > Environments > Deploymenttutorial-env'. The environment details are displayed:

- Environment name:** Deploymenttutorial-env
- URL:** Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com (highlighted with a red box)
- Application name:** Deployment Tutorial
- Health:** Ok (with a green checkmark icon)
- Running version:** code-pipeline-1662399271185-Tzv1VxUki0BP4Y44gSQoCG78YgS8R58I
- Platform:** PHP 8.1 running on 64bit Amazon Linux 2/3.4.0

Buttons for 'Refresh', 'Upload and deploy', and 'Change' are visible. A warning message about AWS Graviton is shown at the top.

4. View the page

Confirm that the updated text appears on the webpage.

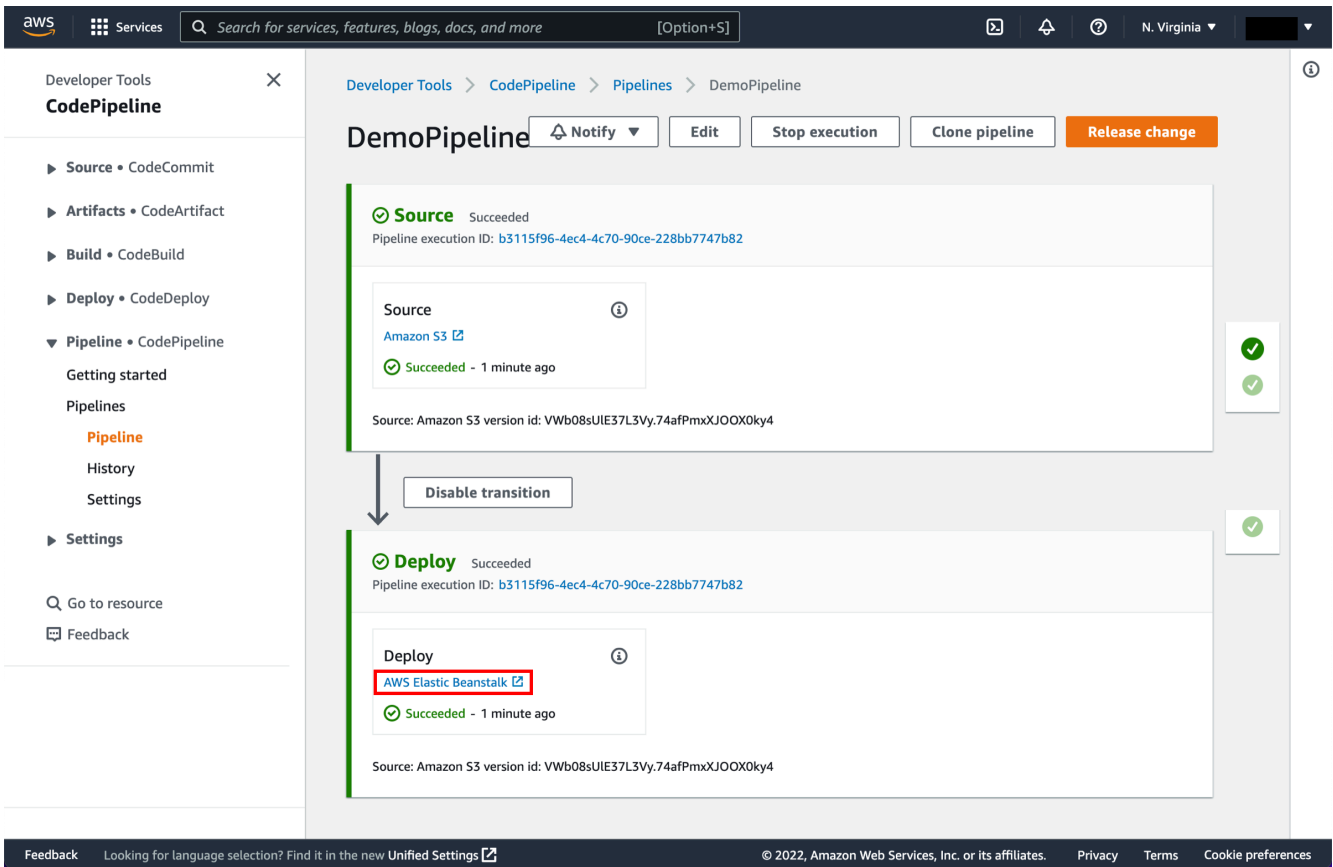


Amazon S3

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 displays the AWS CodePipeline console interface. On the left, a navigation pane shows the 'CodePipeline' section expanded, with 'Pipelines' selected. The main area shows the 'DemoPipeline' details. At the top, there are buttons for 'Notify', 'Edit', 'Stop execution', 'Clone pipeline', and 'Release change'. Below these, the pipeline execution is shown as 'Succeeded'. The 'Source' stage is detailed, showing it used 'Amazon S3' and succeeded '1 minute ago'. A 'Disable transition' button is visible between the stages. The 'Deploy' stage is also detailed, showing it used 'AWS Elastic Beanstalk' (highlighted with a red box) and succeeded '1 minute ago'. The source for both stages is identified as 'Amazon S3 version id: VWb08sUIE37L3Vy.74afPmxXJOOX0ky4'. The footer of the console includes a 'Feedback' link, a language selection prompt, 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. The left sidebar is expanded to 'Deployment Tutorial', showing 'Deploymenttutorial-env' under 'Recent environments'. The main content area displays 'Application 'Deployment Tutorial' environments' with a table of environments. The 'Deploymenttutorial-env' row is highlighted with a red box.

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 URL 'Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com' is highlighted with a red box. The environment details are shown below.

Deploymenttutorial-env
 Application name: **Deployment Tutorial**

Health: Ok

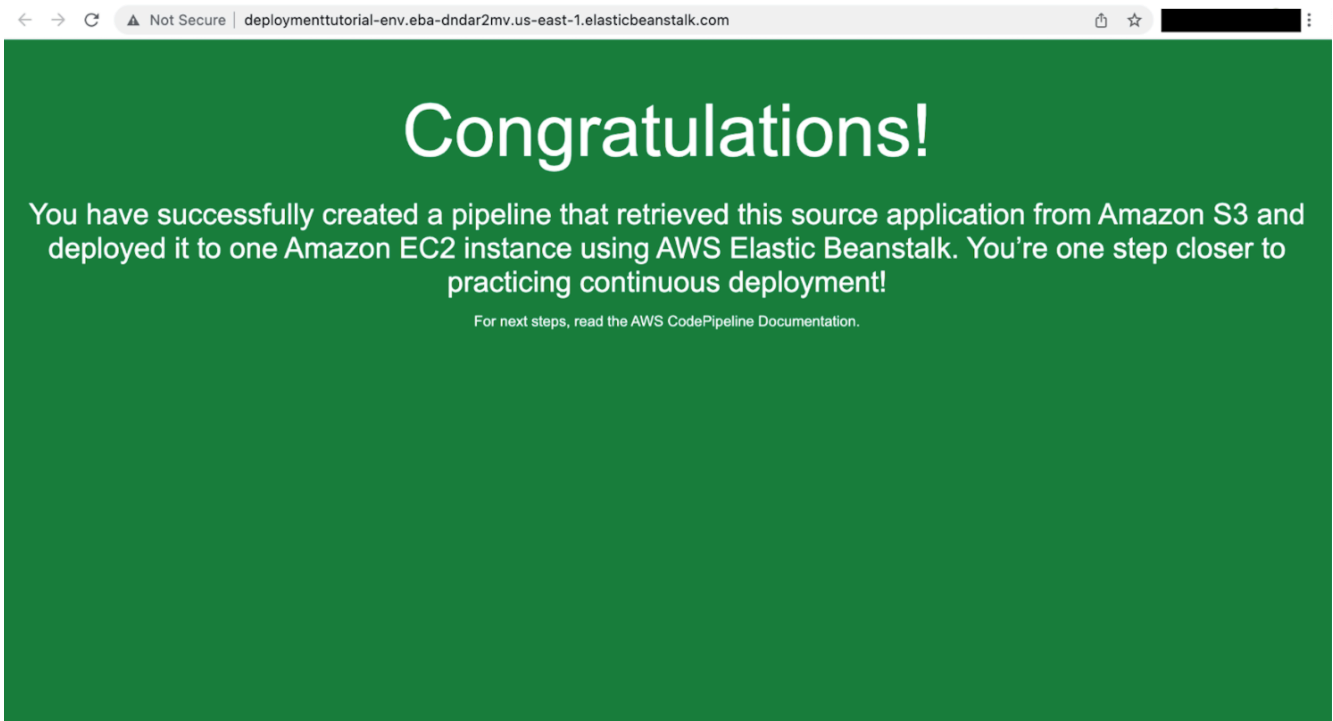
Running version: code-pipeline-1662399271185-Tzv1VxUki0BP4Y44gSQoCG78YgS8R58I

Platform: PHP 8.1 running on 64bit Amazon Linux 2/3.4.0

Recent events: [View]

4. View the page

Confirm that the updated text appears on the webpage.

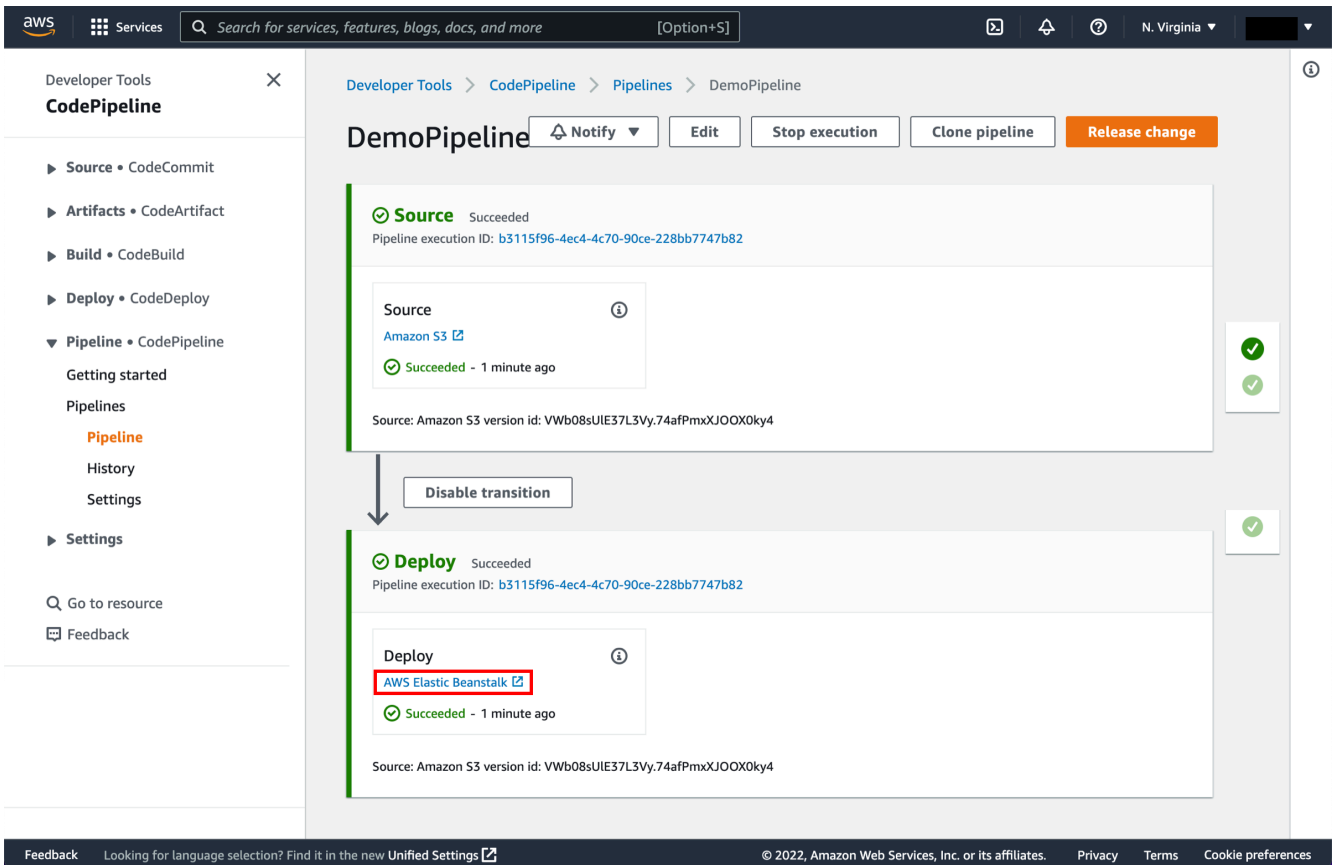


AWS CodeCommit

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 displays the AWS CodePipeline console interface. On the left, a navigation pane shows 'Developer Tools' with 'CodePipeline' selected. The main area shows the 'DemoPipeline' details. At the top, there are buttons for 'Notify', 'Edit', 'Stop execution', 'Clone pipeline', and 'Release change'. The pipeline execution is shown as 'Succeeded' with a green checkmark. The pipeline execution ID is 'b3115f96-4ec4-4c70-90ce-228bb7747b82'. The pipeline consists of two stages: 'Source' and 'Deploy'. The 'Source' stage is highlighted with a green bar on the left and shows 'Succeeded' with a green checkmark. The provider is 'Amazon S3' and it succeeded '1 minute ago'. The source is 'Amazon S3 version id: VWb08sUIE37L3Vy.74afPmxXJOOX0ky4'. The 'Deploy' stage is also highlighted with a green bar on the left and shows 'Succeeded' with a green checkmark. The provider is 'AWS Elastic Beanstalk' (highlighted with a red box) and it succeeded '1 minute ago'. The source is 'Amazon S3 version id: VWb08sUIE37L3Vy.74afPmxXJOOX0ky4'. A 'Disable transition' button is visible between the stages. The footer contains 'Feedback', 'Looking for language selection? Find it in the new Unified Settings', '© 2022, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

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. The left sidebar is open to 'Deployment Tutorial' and shows 'Deploymenttutorial-env' under 'Recent environments'. 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 11:15:30 UTC-0700	Deploymenttutorial-env.eba-dndar2mv.us-east-1.elasticbeanstalk.com	code-pipeline-16624017111 ABLhKicTNwtrXSA5t1hGb4Y

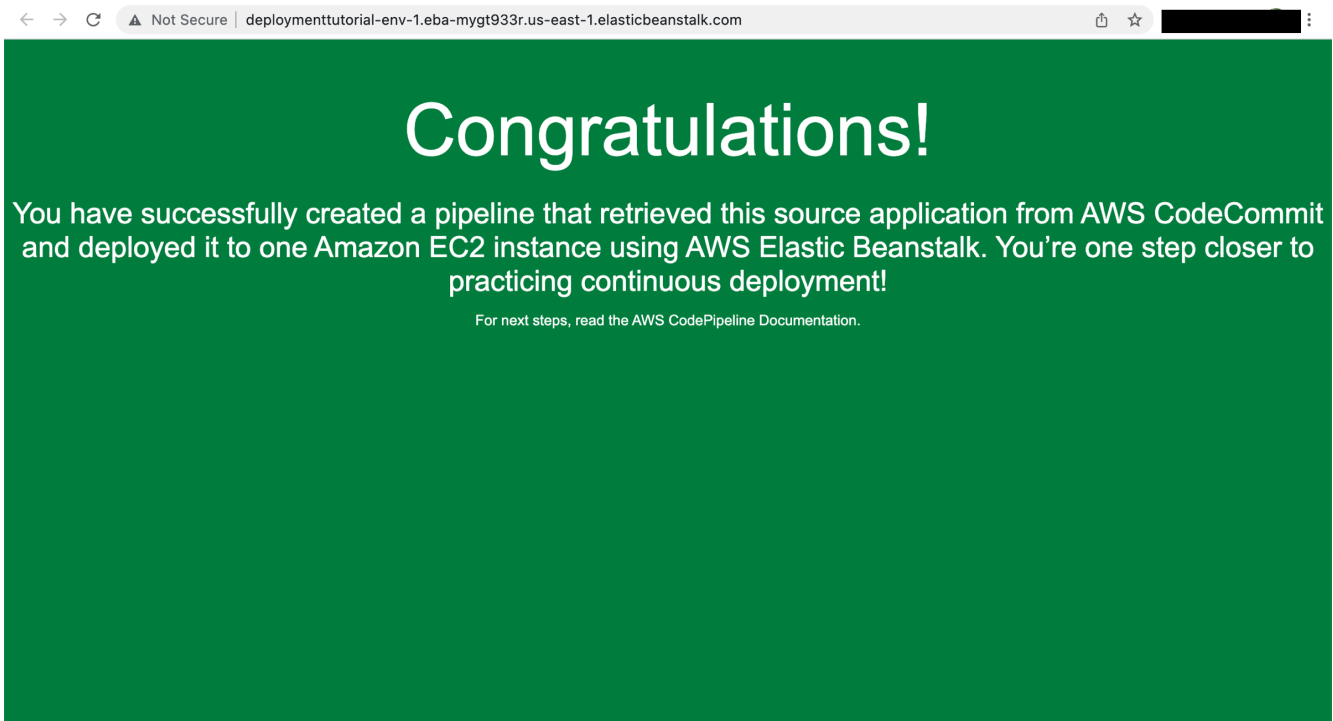
3. Select the URL

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 environment is in 'Ok' health. The running version is 'code-pipeline-1662399271185-Tzv1VxUki0BP4Y44gSQoCG78YgS8R58I'. The platform is 'PHP 8.1 running on 64bit Amazon Linux 2/3.4.0'. A warning message indicates 'Different version recommended'.

4. View the page

Confirm that the updated text appears on the webpage.

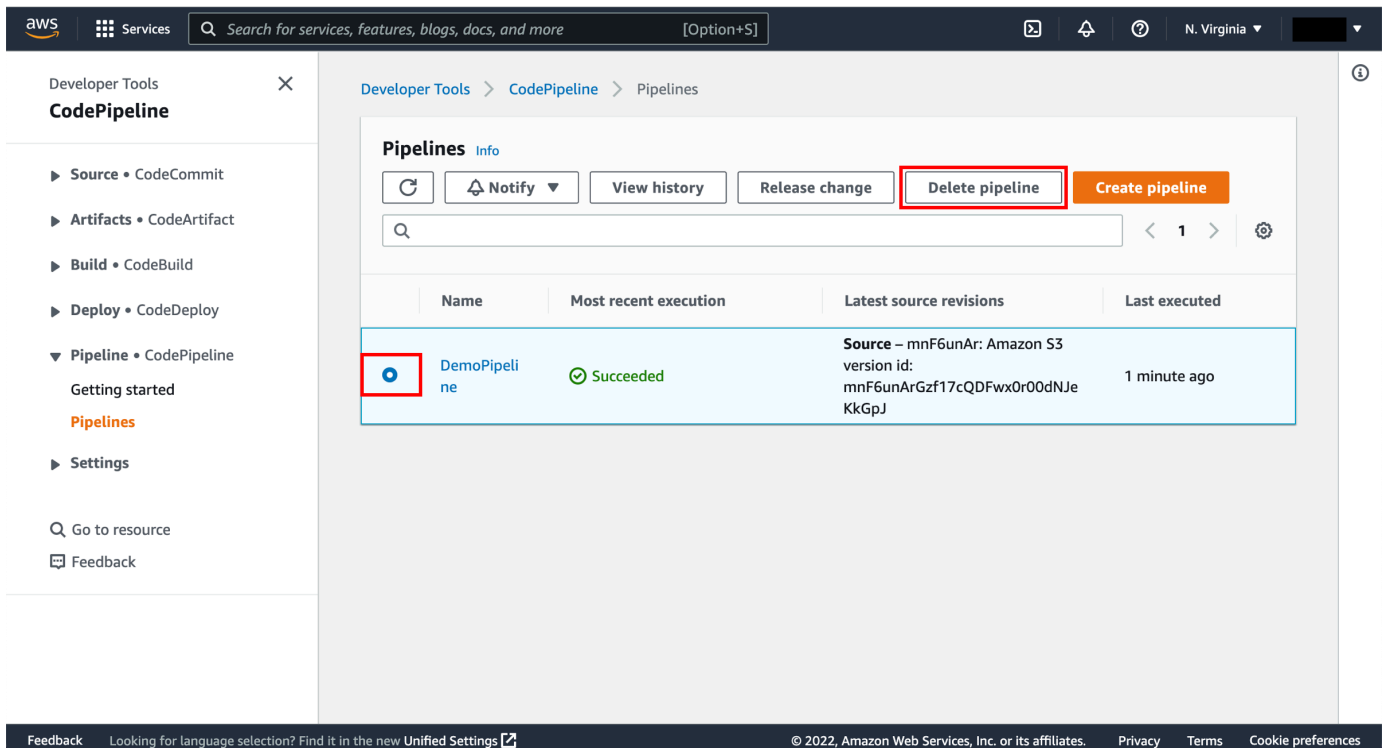


Clean up resources

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

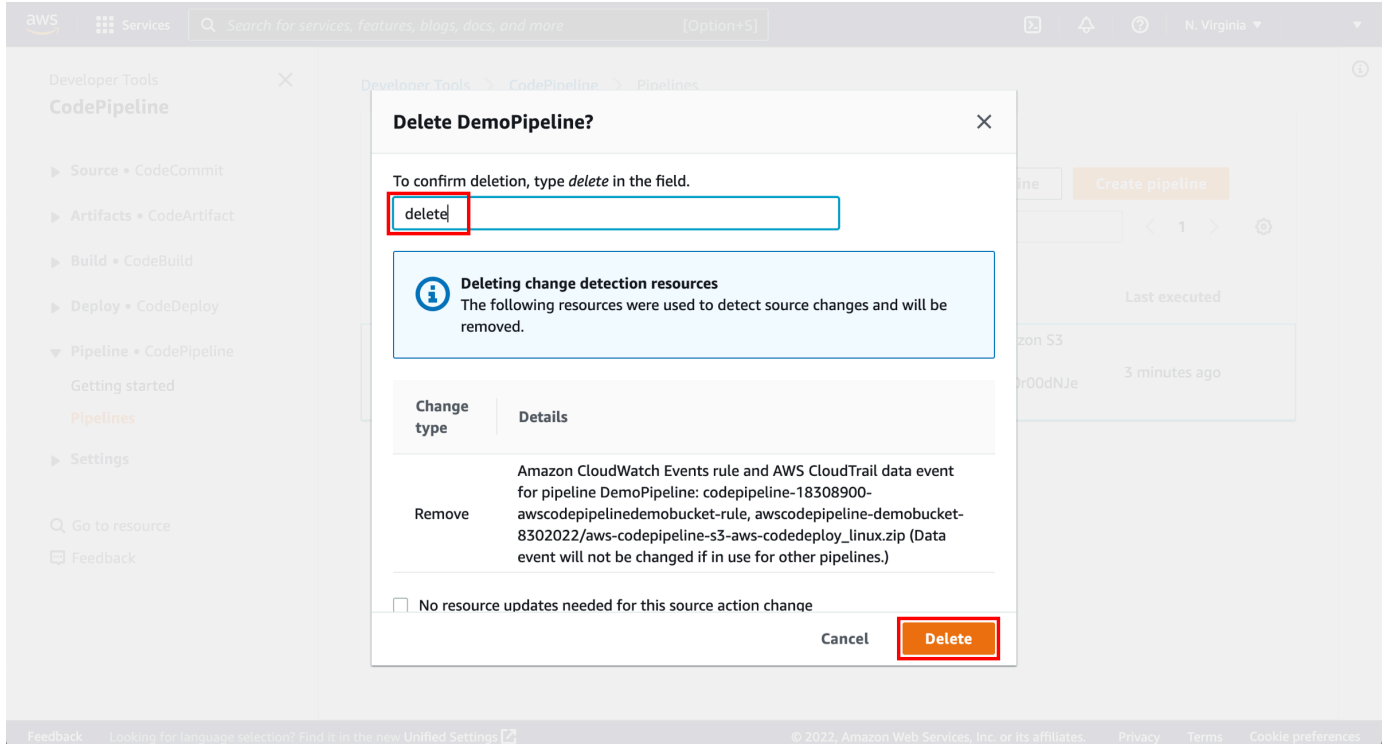
First, you will delete your pipeline. In the **Pipelines** view, select the pipeline radio button and select **Delete pipeline**.



The screenshot shows the AWS CodePipeline console interface. The left sidebar contains navigation options for Developer Tools, CodePipeline, Source (CodeCommit), Artifacts (CodeArtifact), Build (CodeBuild), Deploy (CodeDeploy), Pipeline (CodePipeline), and Settings. The main content area displays the 'Pipelines' page for 'DemoPipeline'. At the top, there are buttons for 'Refresh', 'Notify', 'View history', 'Release change', 'Delete pipeline' (highlighted with a red box), and 'Create pipeline'. Below these buttons is a search bar and a table of pipeline executions. The table has columns for Name, Most recent execution, Latest source revisions, and Last executed. The first row shows 'DemoPipeline' with a status of 'Succeeded' and a last executed time of '1 minute ago'. The 'Delete pipeline' button is highlighted with a red box.

2. Confirm deletion

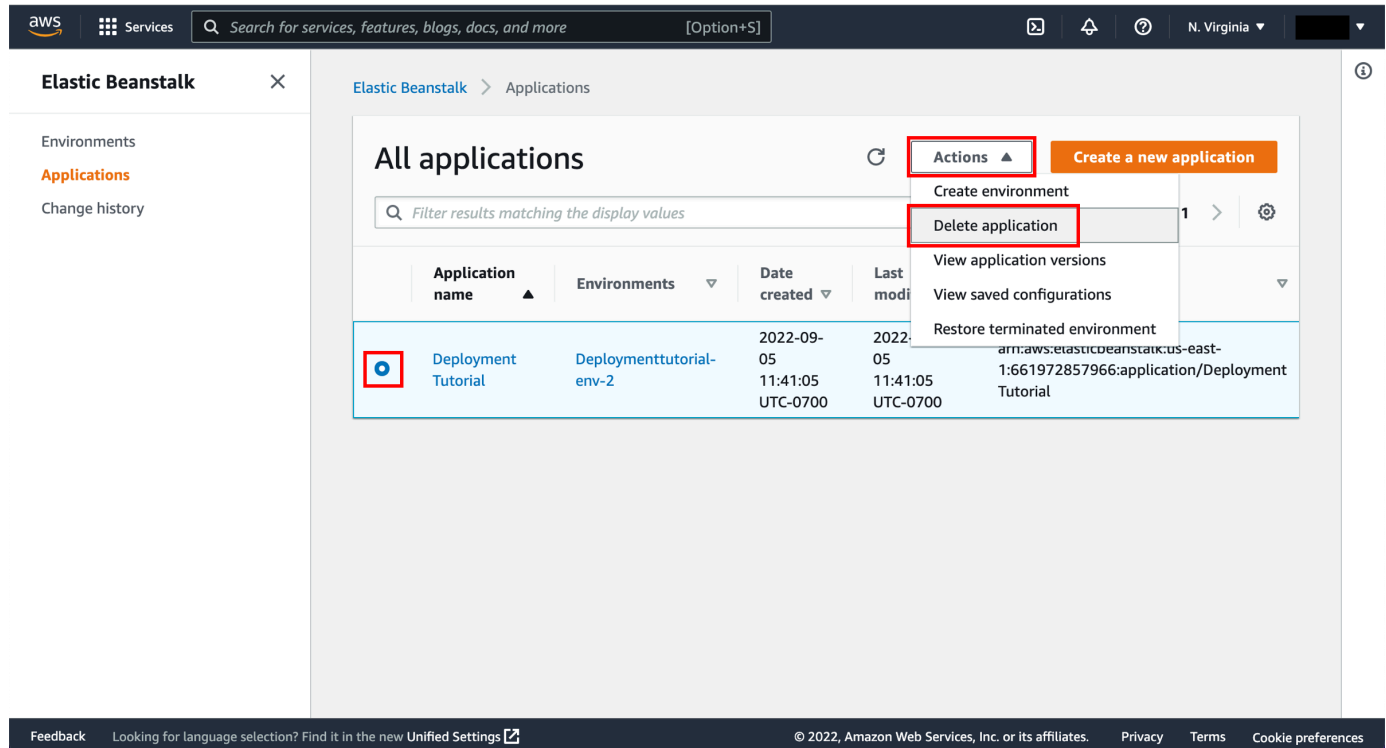
To confirm deletion, enter **delete** in the field and choose **Delete**.



The screenshot shows the AWS CodePipeline console with a 'Delete DemoPipeline?' dialog box open. The dialog box has a title bar with a close button. Below the title, it says 'To confirm deletion, type *delete* in the field.' A text input field contains the word 'delete', which is highlighted with a red box. Below the input field is a blue information box with an 'i' icon, stating 'Deleting change detection resources' and 'The following resources were used to detect source changes and will be removed.' Below this is a table with two columns: 'Change type' and 'Details'. The table has one row with 'Remove' in the 'Change type' column and '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.)' in the 'Details' column. At the bottom of the dialog, there is a checkbox labeled 'No resource updates needed for this source action change' which is unchecked, and two buttons: 'Cancel' and 'Delete' (highlighted with a red box).

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**.

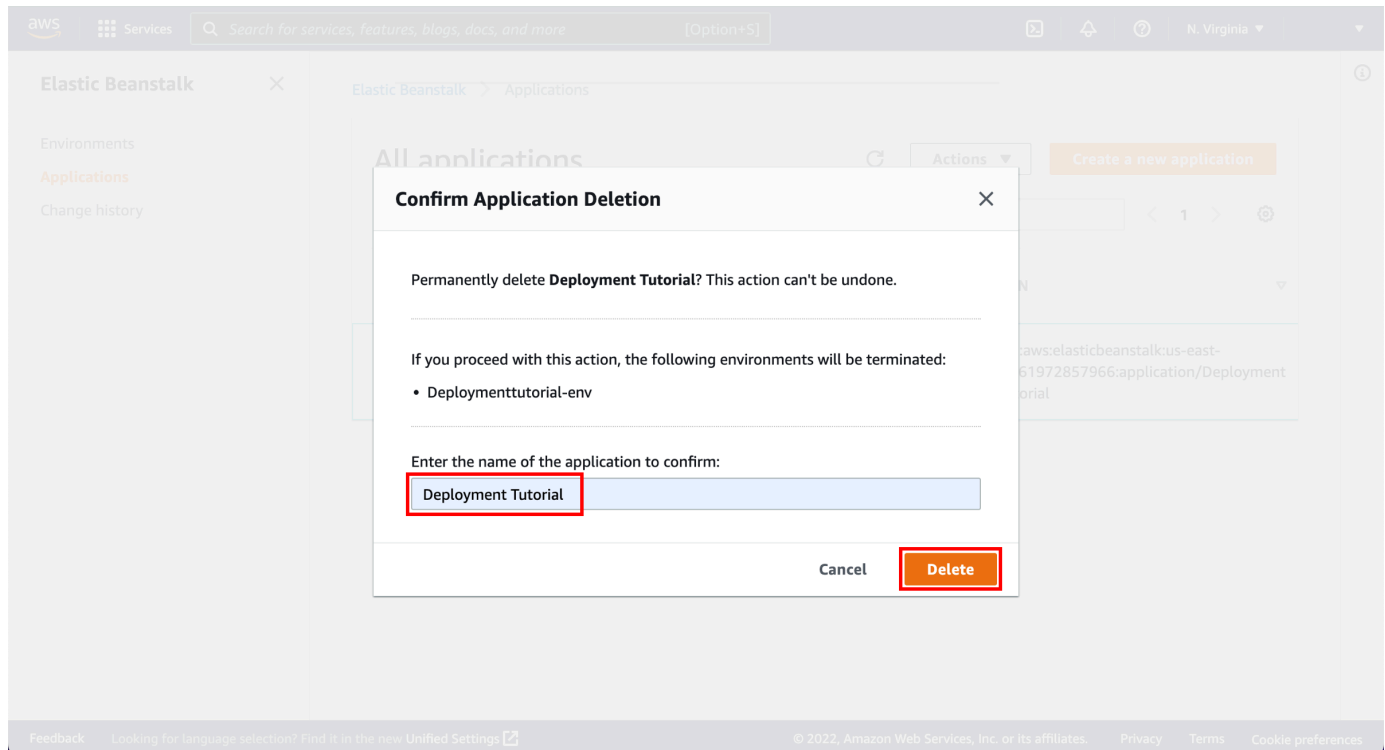


The screenshot shows the AWS Elastic Beanstalk console. The left sidebar has 'Elastic Beanstalk' selected, with 'Applications' highlighted. The main content area is titled 'All applications' and contains a table of applications. The 'Deployment Tutorial' application is selected, and the 'Actions' dropdown menu is open, showing 'Delete application' as the selected option.

Application name	Environments	Date created	Last modified	Actions
Deployment Tutorial	Deploymenttutorial-env-2	2022-09-05 11:41:05 UTC-0700	2022-09-05 11:41:05 UTC-0700	<ul style="list-style-type: none">Create environmentDelete applicationView application versionsView saved configurationsRestore terminated environment

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**.

Amazon S3 > Buckets

Account snapshot [View Storage Lens dashboard](#)

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

Buckets (5) [Info](#) [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
awscodpipeline-demobucket-8302022	US East (N. Virginia) us-east-1	Bucket and objects not public	August 30, 2022, 10:02:19 (UTC-07:00)

2. Confirm deletion

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

Amazon S3 > Buckets > awscodpipeline-demobucket-8302022 > Empty bucket

Empty bucket [Info](#)

Warning

- Emptying the bucket deletes all objects in the bucket and cannot be undone.
- Objects added to the bucket while the empty bucket action is in progress might be deleted.
- To prevent new objects from being added to this bucket while the empty bucket action is in progress, you might need to update your bucket policy to stop objects from being added to the bucket.

[Learn more](#)

Info If your bucket contains a large number of objects, creating a lifecycle rule to delete all objects in the bucket might be a more efficient way of emptying your bucket. [Learn more](#) [Go to lifecycle rule configuration](#)

Permanently delete all objects in bucket "awscodpipeline-demobucket-8302022"?

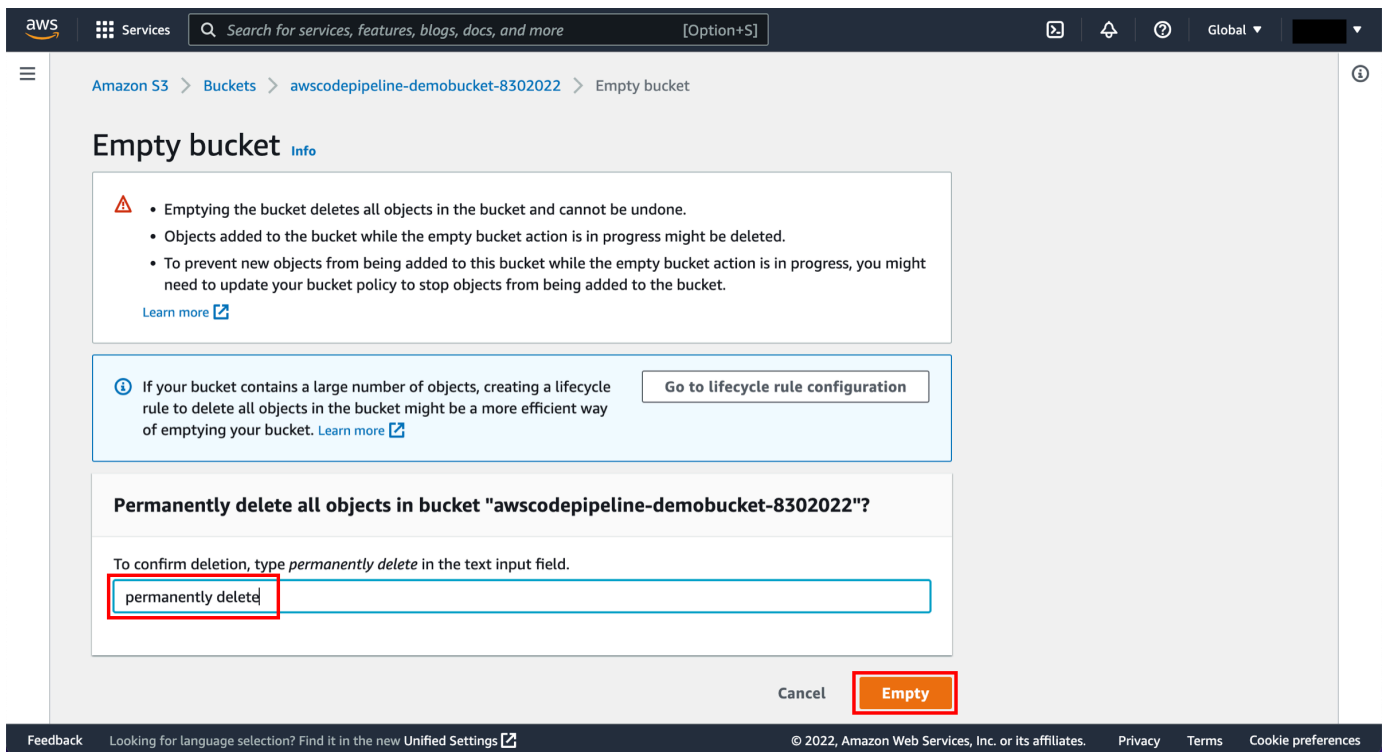
To confirm deletion, type *permanently delete* in the text input field.

permanently delete

[Cancel](#) [Empty](#)

3. Delete the bucket

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



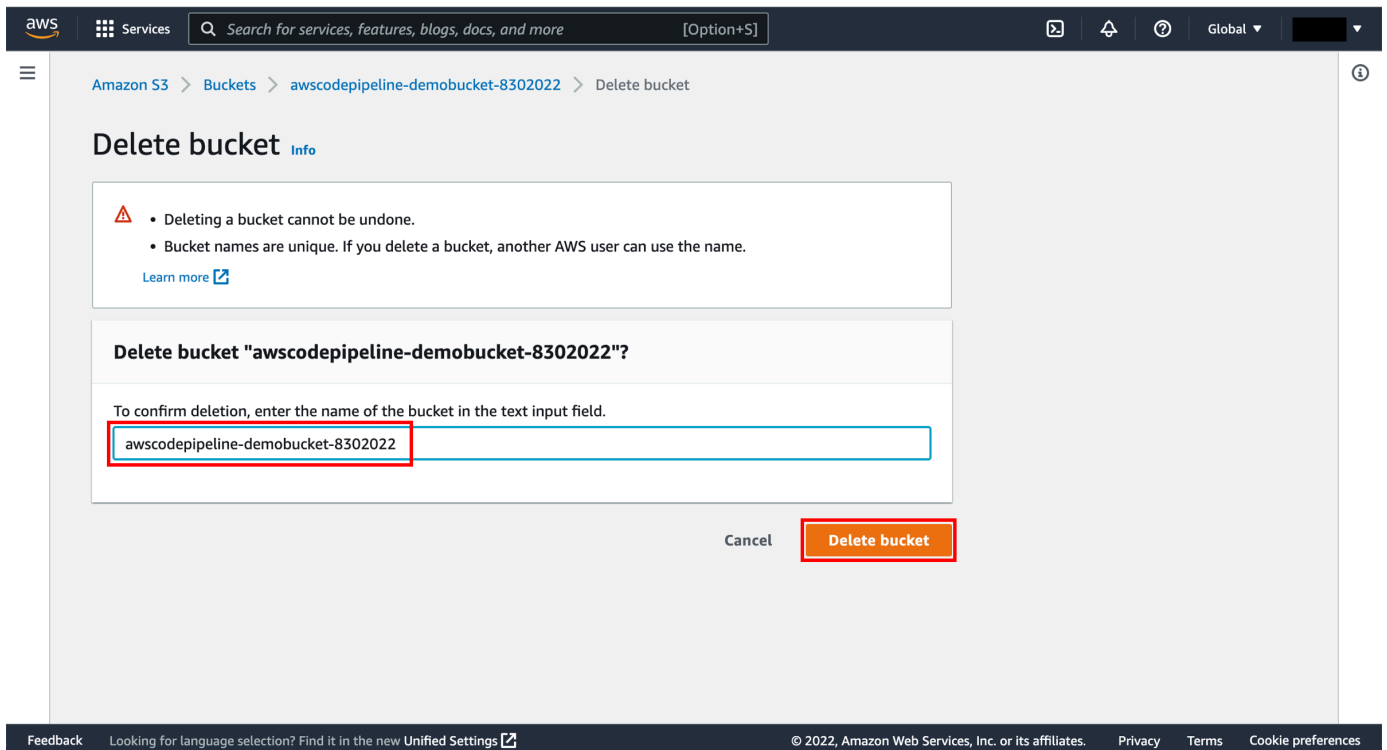
The screenshot shows the AWS console interface for the 'Empty bucket' action. The breadcrumb trail is 'Amazon S3 > Buckets > awscodepipeline-demobucket-8302022 > Empty bucket'. The main heading is 'Empty bucket' with an 'Info' link. A warning box contains the following text:

- Emptying the bucket deletes all objects in the bucket and cannot be undone.
- Objects added to the bucket while the empty bucket action is in progress might be deleted.
- To prevent new objects from being added to this bucket while the empty bucket action is in progress, you might need to update your bucket policy to stop objects from being added to the bucket.

A 'Learn more' link is provided below the warning box. Below this, an information box states: 'If your bucket contains a large number of objects, creating a lifecycle rule to delete all objects in the bucket might be a more efficient way of emptying your bucket.' A 'Go to lifecycle rule configuration' button is present. The main confirmation section is titled 'Permanently delete all objects in bucket "awscodepipeline-demobucket-8302022"?'. It instructs the user: 'To confirm deletion, type *permanently delete* in the text input field.' The text input field contains 'permanently delete' and is highlighted with a red box. At the bottom right, there are 'Cancel' and 'Empty' buttons, with the 'Empty' button also highlighted with a red box.

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 console interface. The left sidebar contains navigation options for Developer Tools, CodeCommit, and various services like CodeArtifact, CodeBuild, CodeDeploy, CodePipeline, and Settings. The main content area is titled 'Developer Tools > CodeCommit > Repositories'. At the top, there are buttons for 'Notify', 'Clone URL', 'View repository', 'Delete repository' (highlighted with a red box), and 'Create repository'. Below these is a search bar and a table of repositories. The table has columns for Name, Description, Last modified, and Clone URL. One repository, 'PipelineRepo', is listed with a description of '-' and a last modified time of '1 hour ago'. The 'PipelineRepo' row is highlighted with a red box. The bottom of the page 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. Confirm deletion

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

The screenshot shows the same AWS CodeCommit console interface as the previous image, but with a confirmation dialog box open. The dialog is titled 'Delete PipelineRepo?' and contains the following text: '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.' Below the text is a text input field with the word 'delete' typed in it, and a 'Cancel' button and a 'Delete' button (highlighted with a red box) at the bottom right. The background of the console is dimmed.

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!