

Creating CodePipeline

- 1) Go to the following link for the sample code

https://github.com/awslabs/aws-codepipeline-s3-aws-codedeploy_linux

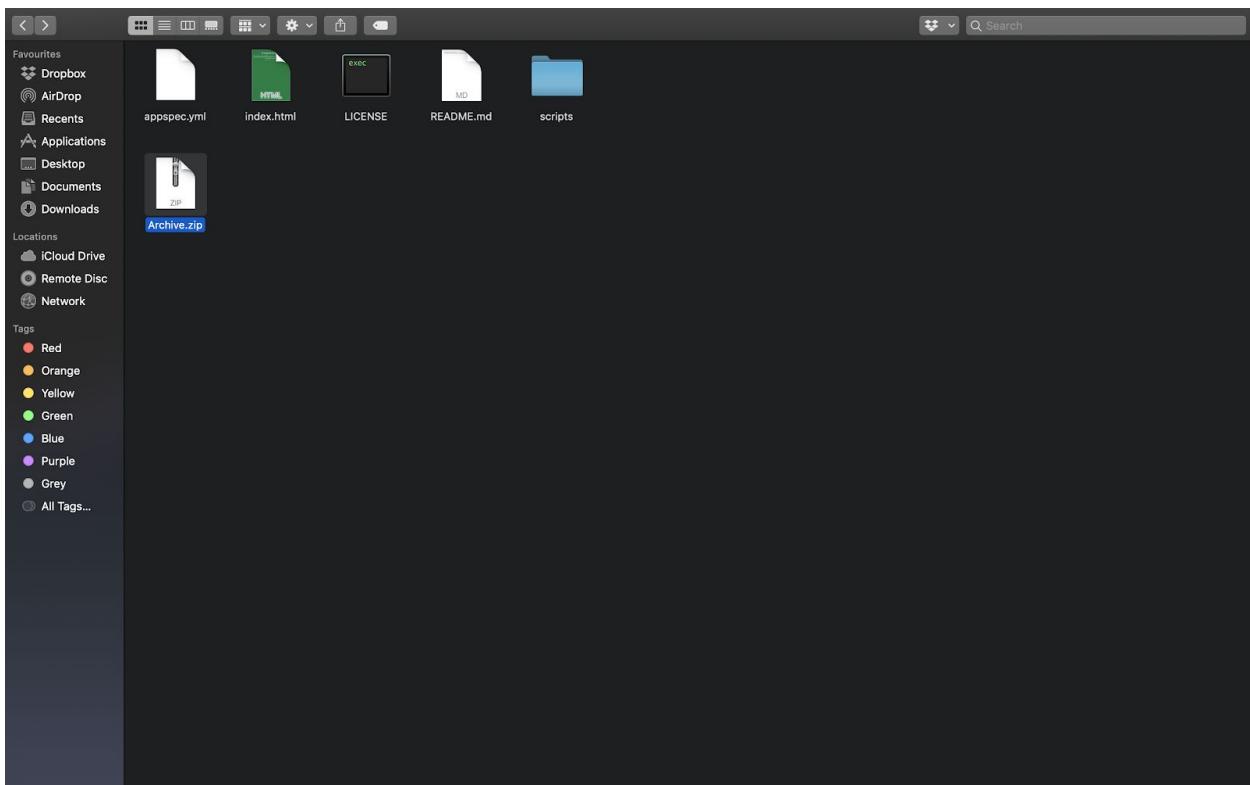
File	Description	Date
.github	Adding template	2 years ago
dist	Added dist folder	4 years ago
scripts	Added AWS CodePipeline Sample	4 years ago
CODE_OF_CONDUCT.md	Adding CONTRIBUTING/CoC	2 years ago
CONTRIBUTING.md	Adding CONTRIBUTING/CoC	2 years ago
LICENSE	Added AWS CodePipeline Sample	4 years ago
README.md	Initial commit	4 years ago
appspec.yml	Added AWS CodePipeline Sample	4 years ago
index.html	Added AWS CodePipeline Sample	4 years ago

aws-codepipeline-s3-aws-codedeploy_linux

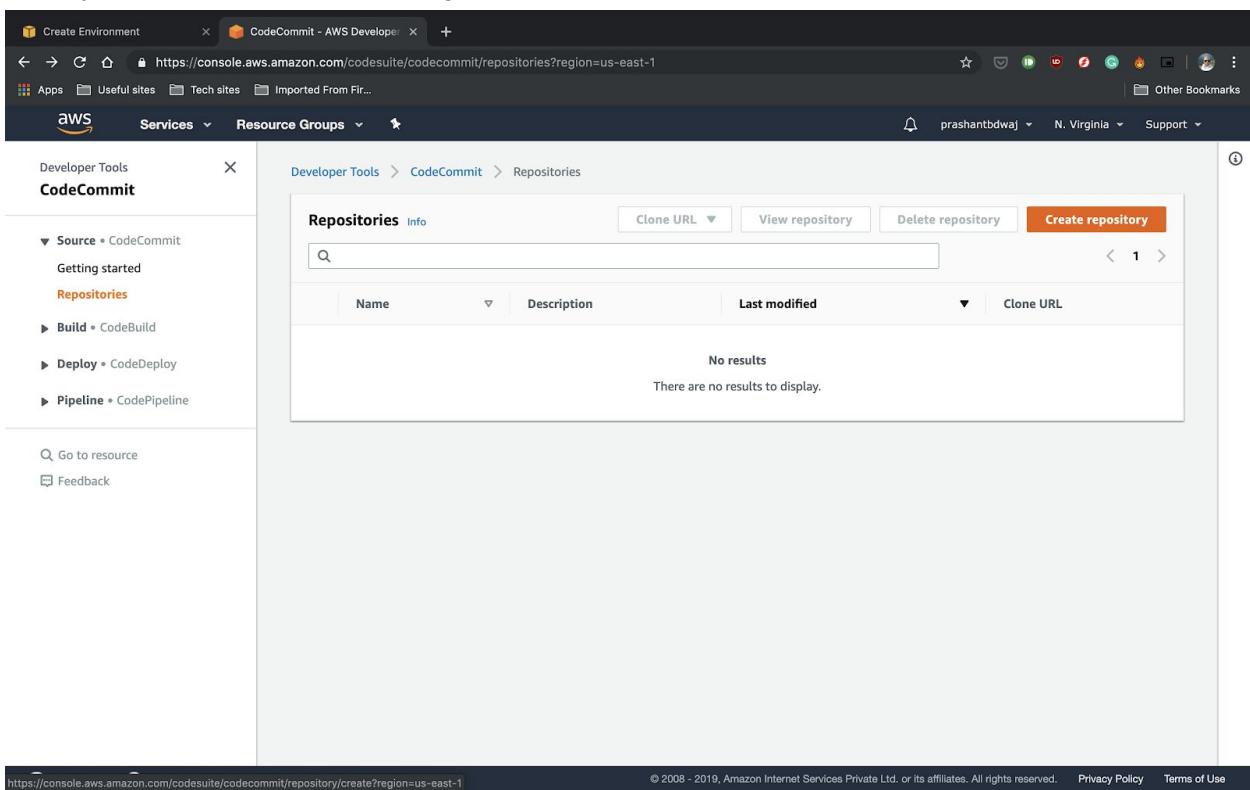
Use this sample when creating a simple pipeline in AWS CodePipeline while following the Simple Pipeline Walkthrough tutorial. <http://docs.aws.amazon.com/codepipeline/latest/userguide/getting-started-w.html>

- 2) Select the *dist* folder.
- 3) Select the file named *aws-codepipeline-s3-aws-codedeploy_linux.zip*.
- 4) Select **View Raw**.
- 5) Save the file to your computer.

- 6) Unzip the file, open the folder created, select all the files inside and zip them again.



- 7) Open your AWS console and navigate to Code Commit.



The screenshot shows the AWS CodeCommit interface. On the left, there's a sidebar with 'Developer Tools' and 'CodeCommit' selected. Under 'CodeCommit', there are sections for 'Source', 'Build', 'Deploy', and 'Pipeline'. Below that, there are links to 'Go to resource' and 'Feedback'. The main content area shows the 'Repositories' list. At the top of the list area, there are buttons for 'Clone URL', 'View repository', 'Delete repository', and 'Create repository'. A search bar is also present. Below the buttons, there's a table with columns for 'Name', 'Description', and 'Last modified'. A message 'No results' and 'There are no results to display.' is centered in the list area. The bottom of the screen shows the browser's address bar with the URL 'https://console.aws.amazon.com/codesuite/codecommit/repositories?region=us-east-1' and some footer text including '© 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.' and links for 'Privacy Policy' and 'Terms of Use'.

8) Click on **Create Repository**.

The screenshot shows the AWS CodeCommit 'Create repository' interface. On the left, there's a sidebar with navigation links for Source (CodeCommit), Build (CodeBuild), Deploy (CodeDeploy), and Pipeline (CodePipeline). The main area is titled 'Create repository' and contains fields for 'Repository name' (set to 'testRepo') and 'Description - optional' (set to 'Test Repository for Code Pipeline'). Below these fields is an 'Add tag' button. At the bottom right are 'Cancel' and 'Create' buttons, with 'Create' being the active one.

9) Enter a repository name and description and click on **Create**.

The screenshot shows the 'Connection steps' page for the 'testRepo' repository. The sidebar on the left shows the 'Code' section selected under 'Source'. The main content area is titled 'Connection steps' and includes a note about using a root account. It has tabs for 'HTTPS' (selected) and 'SSH'. Under 'Step 1: Prerequisites', it says you must use a Git client that supports Git version 1.7.9 or later. Under 'Step 2: Set up the AWS CLI Credential Helper', it says to set up the AWS CLI Credential Helper. There's also an 'Additional details' section with a link to documentation. At the bottom, there's a file upload section with 'testRepo' and 'Info' buttons, and an 'Add file' button.

10) Select add file , then click on upload file

The screenshot shows the AWS CodeCommit console at the URL <https://console.aws.amazon.com/codesuite/codecommit/repositories/testRepo/setup?region=us-east-1>. The left sidebar has a 'CodeCommit' section expanded, showing 'Source' (selected), 'Getting started', 'Repositories', 'Code' (selected), 'Pull requests', 'Commits', 'Branches', 'Git tags', and 'Settings'. Below these are sections for 'Build', 'Deploy', and 'Pipeline'. At the bottom of the sidebar are 'Go to resource' and 'Feedback' links. The main content area is titled 'Step 1: Prerequisites' and 'Step 2: Set up the AWS CLI Credential Helper'. It includes instructions for Git clients, IAM users, and AWS CLI setup. A 'testRepo' section shows an 'Empty repository' message with a 'Create file' button. On the right, there are buttons for 'Add file', 'Create file', and 'Upload file'. The top navigation bar includes links for 'Create Environment', 'AWS Developer', and 'Other Bookmarks', along with user information for 'prashantbdwaj'.

11) Click on **Choose file**

The screenshot shows the AWS CodeCommit interface for uploading files. The left sidebar shows navigation options like 'Source', 'Code', 'Build', 'Deploy', and 'Pipeline'. The main area is titled 'Upload a file' under 'testRepo'. It has sections for 'Upload file' (with a 'Choose file' button), 'Commit changes to master' (with fields for 'Author name' and 'Email address'), and a 'Commit message - optional' field. The URL in the browser is https://console.aws.amazon.com/codesuite/codecommit/repositories/testRepo/setup/upload.

12) Select the zip file downloaded in step 5.

- 13) Add the author name, email address and commit message and click on **Commit Changes**.

The screenshot shows the AWS CodeCommit interface. On the left, a sidebar menu includes 'Source', 'Code', 'Build', 'Deploy', and 'Pipeline'. The 'Code' section is selected. In the main area, a 'Upload a file' dialog is open. It displays a table with one row for 'aws-codepipeline-s3-aws-codedeploy_linux.zip'. Below the table, there's a section for committing changes to the 'master' branch. The 'Author name' field contains 'GL Test', the 'Email address' field contains 'testemail.com', and the 'Commit message' field contains 'Testcommit'. At the bottom right of the dialog is a large orange 'Commit changes' button.

- 14) Confirm that the file has been uploaded to Code Commit.

The screenshot shows the AWS CodeCommit interface again. The left sidebar shows the 'Code' section is selected. The main area displays the 'testRepo' repository. In the top right, there are dropdowns for 'master' and 'Create pull request', and a 'Clone URL' button. Below these, the 'Info' section shows a table with one row for 'aws-codepipeline-s3-aws-codedeploy_linux.zip'. The file is listed under the 'Name' column. At the bottom right of the page is a footer with links for 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

- 15) Navigate to Elastic Beanstalk using the **Services** menu at the top of the screen.
- 16) Click on “Create New Application” on the top right side

Welcome to AWS Elastic Beanstalk

With Elastic Beanstalk, you can **deploy**, **monitor**, and **scale** an application quickly and easily. Let us do the heavy lifting so you can focus on your business.

To deploy your **existing web application**, create an [application source bundle](#) and then [create a new application](#). If you're using **Git** and would prefer to use it with our command line tool, please see [Getting Started with the EB CLI](#).

To deploy a **sample application**, click **Get started**, choose a name, select a platform and click **Create app**.

By launching the sample application, you allow AWS Elastic Beanstalk to administer AWS resources and necessary permissions on your behalf. [Learn more](#)

Get started

Get Started in Three Easy Steps

- Select a Platform
- Upload an Application or Use a Sample
- Run it!

<https://console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1>

- 17) Enter the fields as shown below and click on Create Environment and the bottom of the screen.

The screenshot shows the 'Create Environment' wizard on the AWS Elastic Beanstalk console. The application name is set to 'testApplication'. The environment name is 'Testapplication-env'. The domain is set to '.us-east-1.elasticbeanstalk.com'. The description field is empty. Under 'Base configuration', the 'Platform' section is selected, showing 'Preconfigured platform' (PHP) as the choice. The 'Application code' section is also visible, with 'Sample application' selected. Other options include 'Existing version' and 'Upload your code'.

Application name testApplication

Environment name Testapplication-env

Domain Leave blank for autogenerated value .us-east-1.elasticbeanstalk.com Check availability

Description

Base configuration

Platform Preconfigured platform Platforms published and maintained by AWS Elastic Beanstalk.
PHP

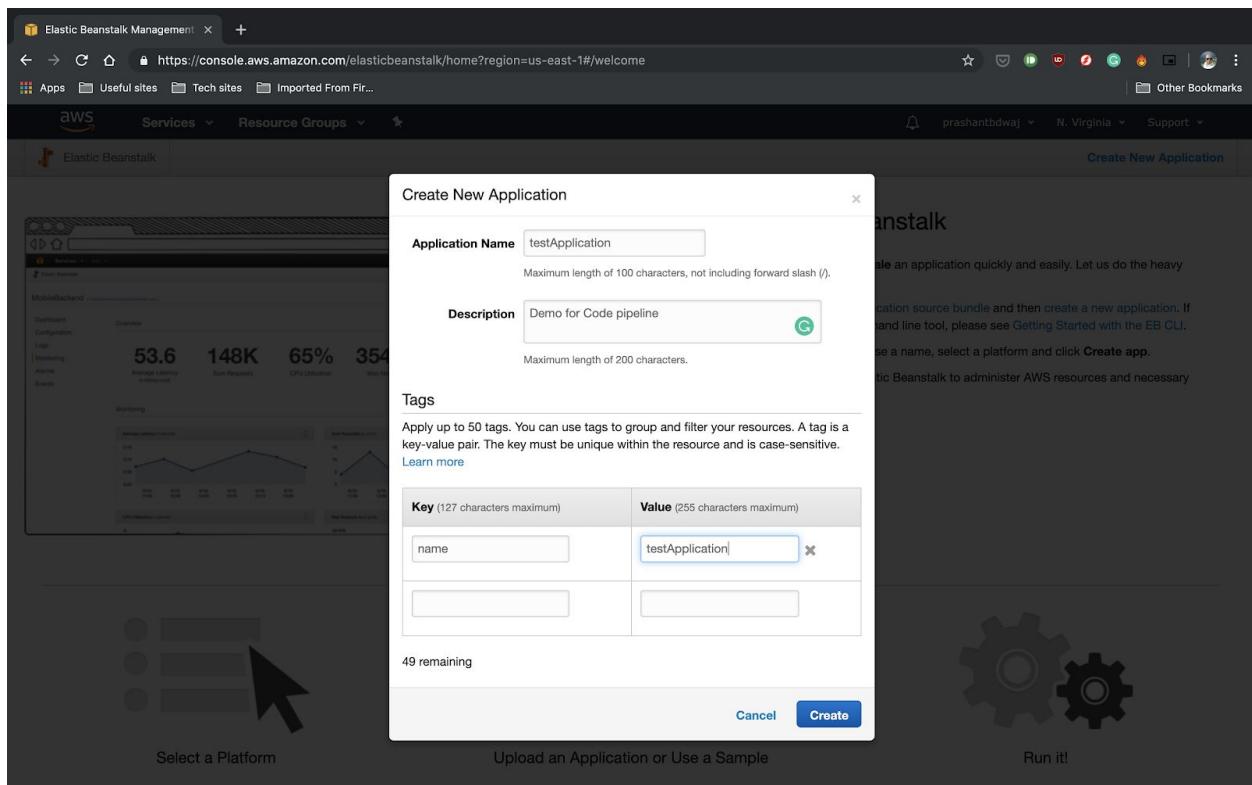
Custom platform Platforms created and owned by you. [Learn more](#)
-- Choose a custom platform --

Application code Sample application Get started right away with sample code.

Existing version Application versions that you have uploaded for testApplication.
-- Choose a version --

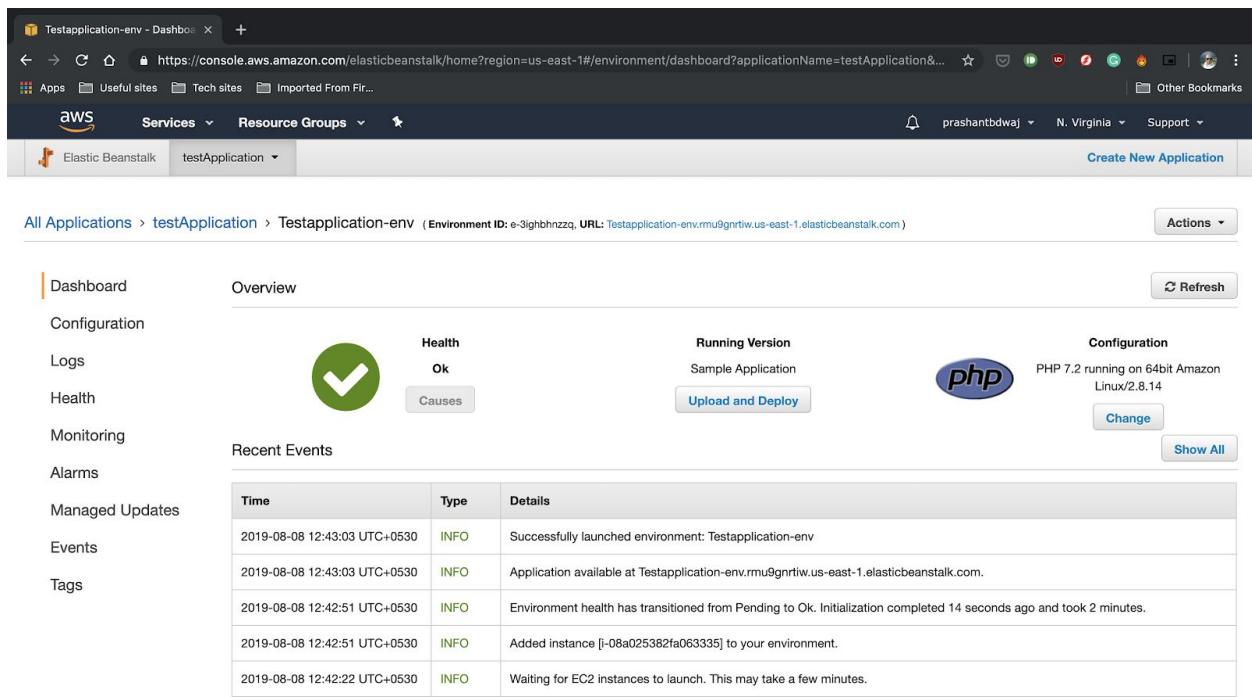
Upload your code Upload a source bundle from your computer or copy one from Amazon S3.
Upload ZIP or WAR

18) Enter an application name and description and click on “Create”.



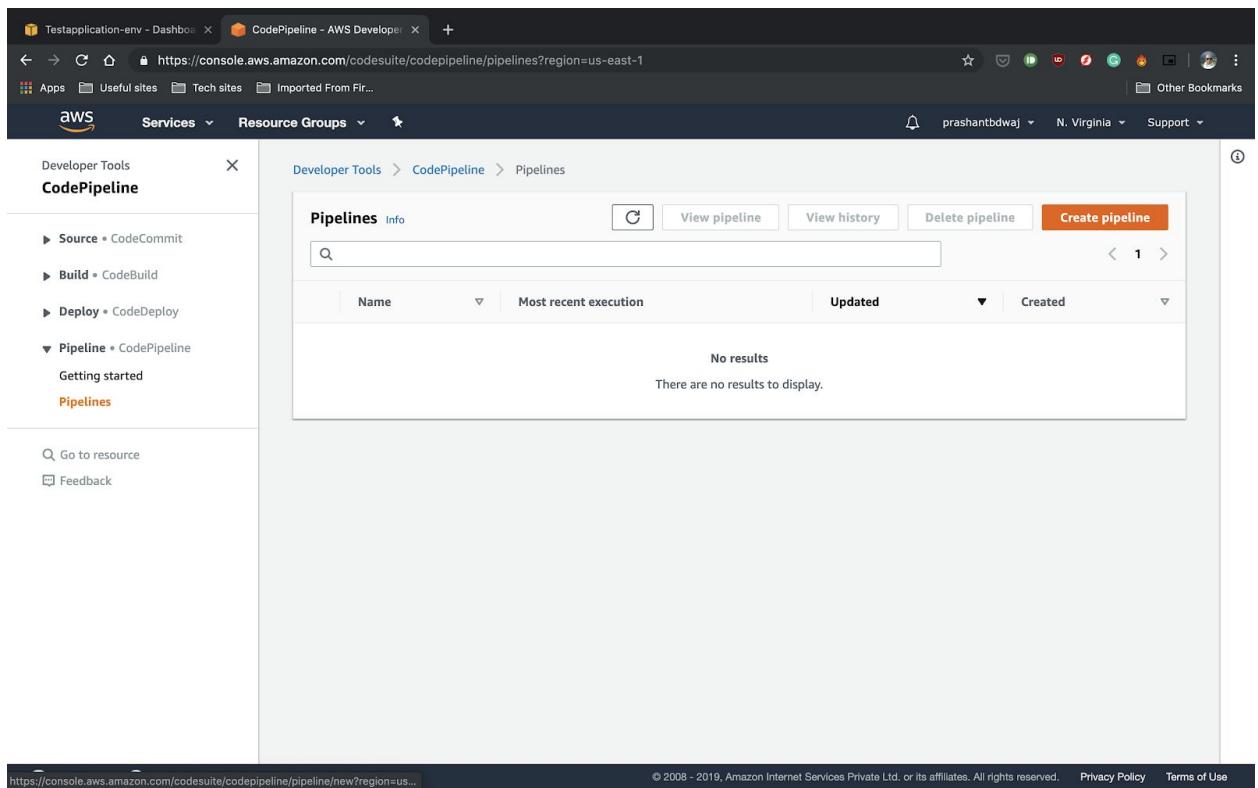
Elastic BeanStalk will typically take a few minutes to set up the environment

19) Once the environment is set up, you should see the screen below.



20) Navigate to Code Pipeline using the **Services** menu at the top of the screen.

21) Click on **Create Pipeline**.



22) Enter the fields as shown below.

The screenshot shows the 'Choose pipeline settings' step of the AWS CodePipeline creation wizard. The pipeline name is set to 'TestPipeline'. Under 'Service role', the 'New service role' option is selected and named 'pipelineTestRole'. The 'Advanced settings' section is collapsed. At the bottom right, there are 'Cancel' and 'Next' buttons.

The screenshot shows the 'Add source stage' step of the AWS CodePipeline creation wizard. The source provider is set to 'AWS CodeCommit', the repository is 'testRepo', and the branch is 'master'. Under 'Change detection options', the 'Amazon CloudWatch Events (recommended)' option is selected. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review

Add build stage

Build - optional

Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.
AWS CodeBuild

Region
US East - (N. Virginia)

Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.
Q or Create project

Cancel Previous **Skip build stage** Next

Click on **Skip Build Stage** here, since the sample code is already pre-built.

Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review

Add deploy stage

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

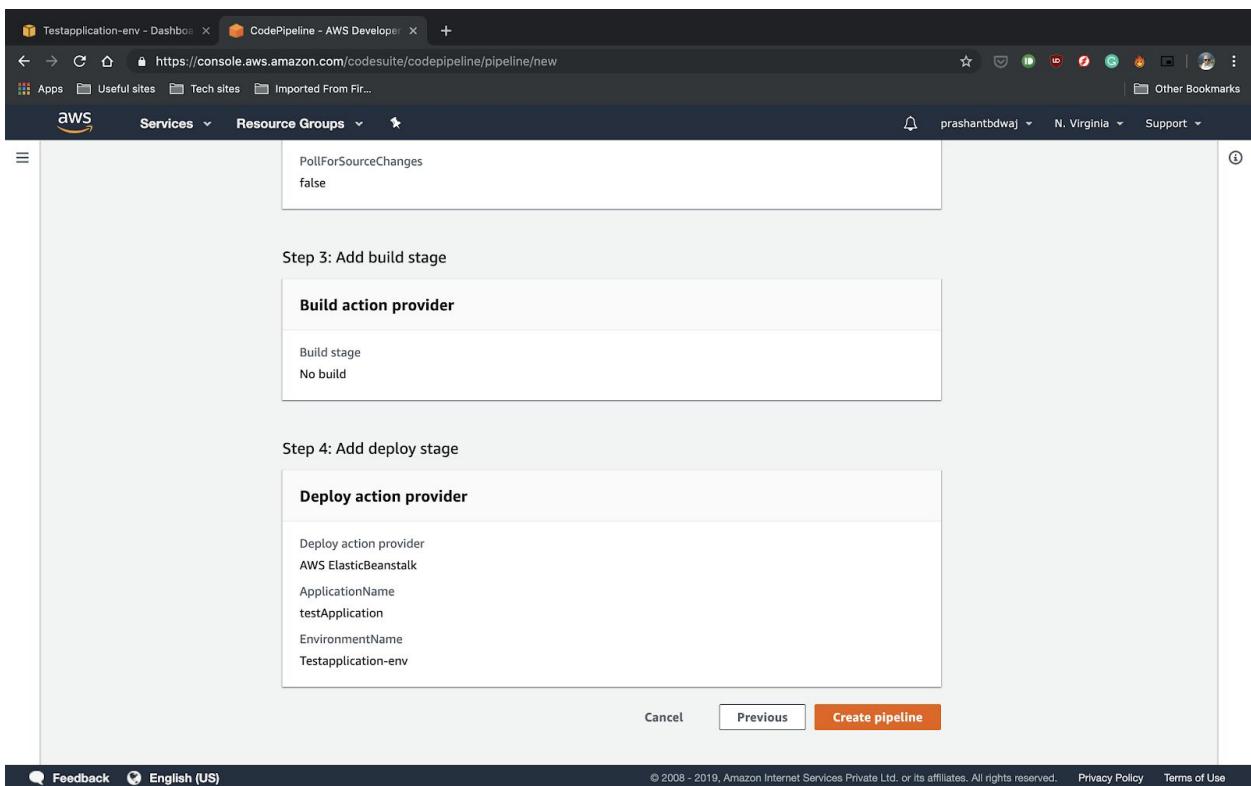
Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS Elastic Beanstalk

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.
testApplication

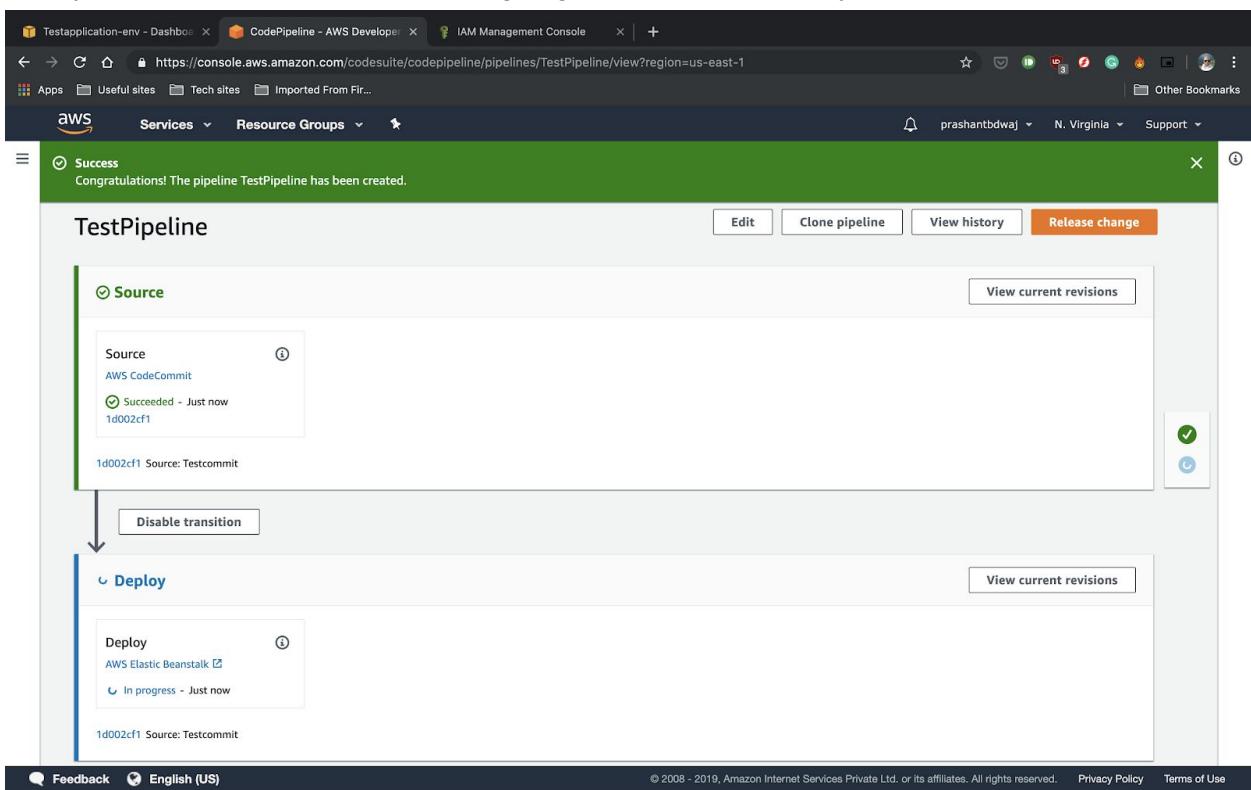
Environment name
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.
Testapplication-env

Cancel Previous **Next**



Click on **Create Pipeline** at the bottom of the screen.

- 23) You should see the screen below. Here, our source file has been processed while deployment via Elastic Beanstalk is ongoing. This process usually takes a few minutes.



24) Click on **AWS Elastic Beanstalk** under Deploy.

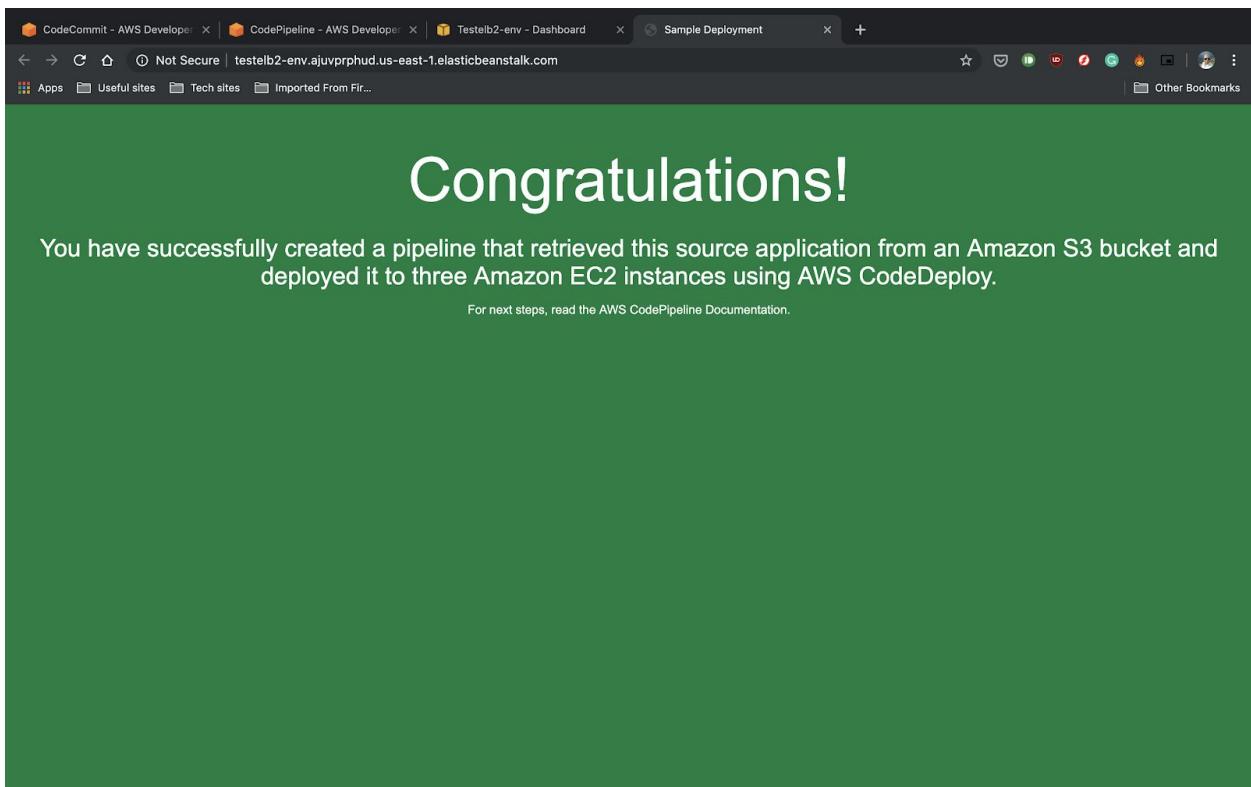
25) Click on the environment you have just deployed

The screenshot shows the AWS Elastic Beanstalk console with the URL <https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/application/overview?applicationName=TestELB2>. The page displays the 'All Applications > TestELB2' environment. On the left, there's a sidebar with 'Environments' selected, showing 'Testelb2-env'. The main content area shows the environment configuration: Environment tier: Web Server, Platform: PHP 7.2 running on 64bit Amazon Linux/2.8.14, Running versions: code-pipeline-1565340006883-604132cf720f9a936a13f4001ecc95ab1e84696e, Last modified: 2019-08-09 14:10:21 UTC+0530, URL: Testelb2-env.ajuvprphud.us-east-1.elasticbeanstalk.com, and Health status: Ok. At the bottom, there are links for Feedback, English (US), Privacy Policy, and Terms of Use.

26) Click on the URL at the top of the screen.

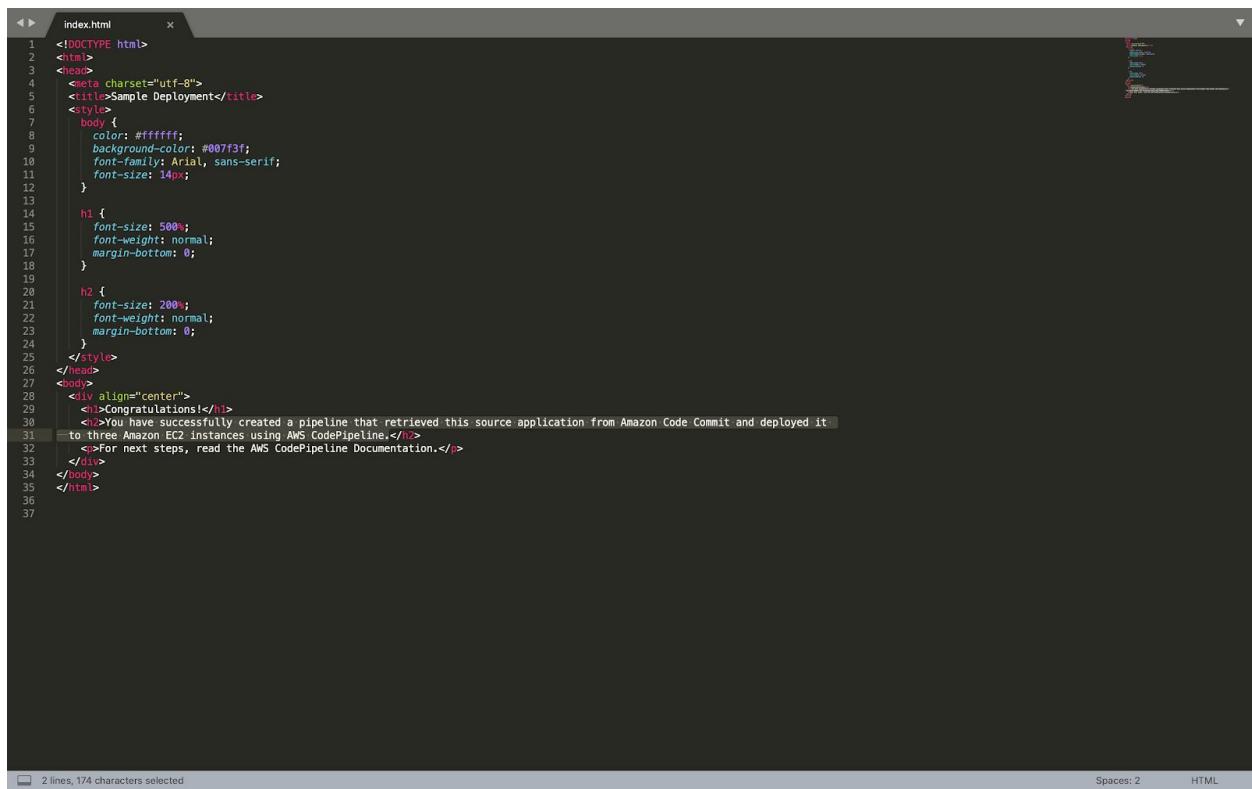
The screenshot shows the AWS Elastic Beanstalk environment dashboard for 'Testelb2-env'. The URL in the browser is <https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/environment/dashboard?applicationName=TestELB2&environmentName=Testelb2-env>. The dashboard includes sections for Overview (Health: Ok, Running Version: code-pipeline-1565340006883-604132cf720f9a936a13f4001ecc95ab1e84696e, Configuration: PHP 7.2 running on 64bit Amazon Linux/2.8.14), Recent Events (listing deployment logs from 2019-08-09 14:10:21 UTC+0530 to 2019-08-09 14:10:12 UTC+0530), and Managed Updates (listing events from 2019-08-09 14:11:59 UTC+0530 to 2019-08-09 14:10:59 UTC+0530). There are also 'Upload and Deploy' and 'Show All' buttons. The footer links to Feedback, English (US), Privacy Policy, and Terms of Use.

27) Congratulations! You have just deployed a webpage using AWS Code pipeline.



28) To perform an update to the app, find the file *index.html* in the files you had unzipped in step 6, open it in your preferred text editor and edit the highlighted lines as per your

needs.



```

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: 500px;
16      font-weight: normal;
17      margin-bottom: 0;
18    }
19
20    h2 {
21      font-size: 200px;
22      font-weight: normal;
23      margin-bottom: 0;
24    }
25  </style>
26 </head>
27 <body>
28   <div align="center">
29     <h1>Congratulations!</h1>
30     <p>You have successfully created a pipeline that retrieved this source application from Amazon Code Commit and deployed it to three Amazon EC2 instances using AWS CodePipeline.</p>
31     <p>For next steps, read the AWS CodePipeline Documentation.</p>
32   </div>
33 </body>
34 </html>
35
36
37

```

2 lines, 174 characters selected

Spaces: 2 HTML

- 29) Zip the files again with the same name, and reupload the file by repeating the steps 10-13.
- 30) Navigate to AWS CodePipeline and wait till the Source and Deploy section both turn green as before.

- 31) Refresh the webpage opened in step 27. If you still see the same page as before, wait a few minutes for Elastic Beanstalk to apply the changes.

