

## Exp 02: To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

### Step 1: Create our ElasticBeanstalk Environment

Login into your AWS account and navigate to services. Search for Elastic Beanstalk service and click on create application. Give your application a suitable name. For the platform, select PHP. Rest of the configuration settings are to be kept as default.

The screenshot displays the AWS Elastic Beanstalk console in the 'ap-south-1' region. The page title is 'Create new application'. The 'Application information' section contains a text input for 'Application name' with the value 'beanstalk4' and a description field. The 'Tags' section has a note about applying up to 50 tags. The 'Custom platform' section is active, showing a 'Platform' dropdown set to 'PHP', a 'Platform branch' dropdown set to 'PHP 8.3 running on 64bit Amazon Linux 2023', and a 'Platform version' dropdown set to '4.3.1 (Recommended)'. The 'Application code' section has three radio buttons: 'Sample application' (selected), 'Existing version', and 'Upload your code'.

https://ap-south-1.console.aws.amazon.com/elasticbeanstalk/home?region=ap-south-1#/create-application

aws Services Search [Alt+S] Mumbai mohitkerkar05

Elastic Beanstalk > Create application

### Create new application Info

**Application information**

Application name  
beanstalk4  
Maximum length of 100 characters.

Description

**Tags**

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

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

https://ap-south-1.console.aws.amazon.com/elasticbeanstalk/home?region=ap-south-1#/create-environment?applicationName=beanstalk4

aws Services Search [Alt+S] Mumbai mohitkerkar05

☒ Custom platform  
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform  
PHP

Platform branch  
PHP 8.3 running on 64bit Amazon Linux 2023

Platform version  
4.3.1 (Recommended)

**Application code Info**


☒ Sample application  
☐ Existing version  
Application versions that you have uploaded.  
☐ Upload your code  
Upload a source bundle from your computer or copy one from Amazon S3.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Now, while creating the environment, we are asked to provide an IAM role with the necessary EC2 permissions. We are supposed to make sure that we have made an existing IAM role with the following set of permissions:



1. AWSElasticBeanStalkWebTier
2. AWSElasticBeanStalkWorkerTier
3. AWSElasticBeanStalkMulticontainerDocker

We can skip the steps to follow after the initial few steps mentioned above and move straight to review the settings of our environment. After reviewing everything properly, our environment can successfully be created.






[Elastic Beanstalk](#) > [Environments](#) > Laukikapp-env


## Laukikapp-env Info

 **Actions**  **Upload and deploy**

### Environment overview

Health  <b>Warning</b>	Environment ID  e-kmfuu3urih
Domain Laukikapp-env.eba-mmfhaxpm.eu-north-1.elasticbeanstalk.com 	Application name laukik_app

### Platform Change version

Platform PHP 8.3 running on 64bit Amazon Linux 2023/4.3.2
Running version -
Platform state  <b>Supported</b>

## Source

**Source provider**  
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (Version 2) 

 **New GitHub version 2 (app-based) action**  
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. [Learn more](#)

**Connection**  
Choose an existing connection that you have already configured, or create a new one and then return to this task.

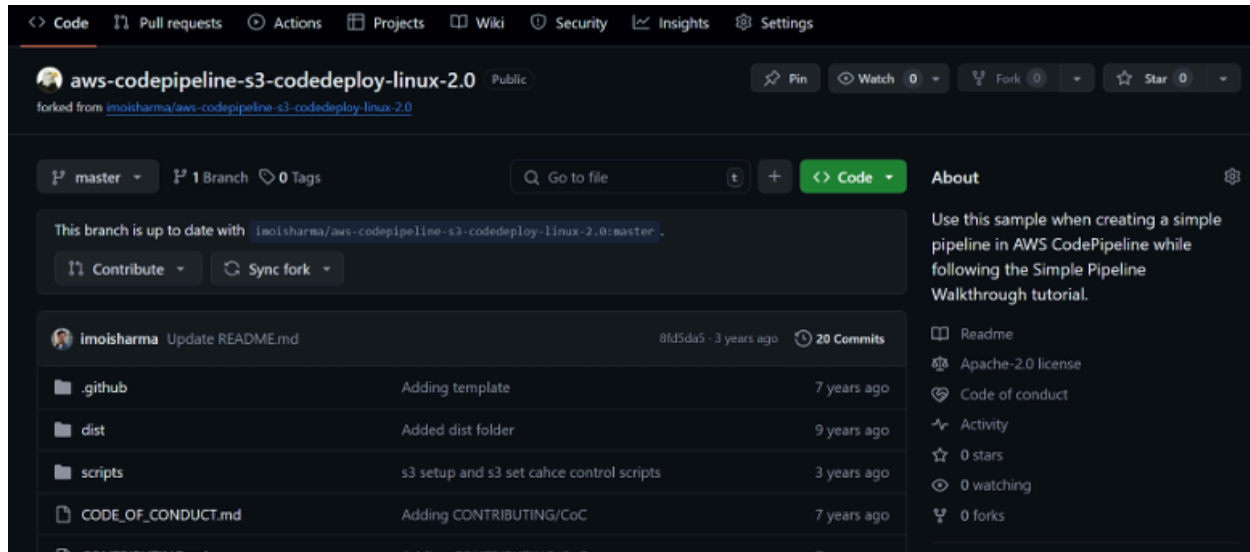
 or 

Connecting

Step 2: Fork the required repository onto our github account

The repository to be forked is- imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0

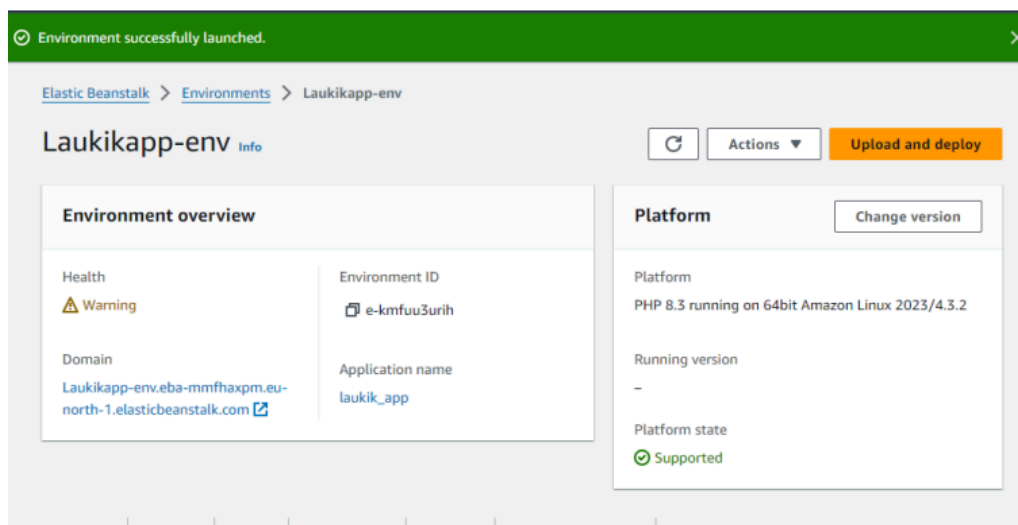
This step is necessary for the execution of the steps to follow. It will be helpful in the creation of a pipeline.



Step 3: Creation of the Pipeline

Navigate to Codepipeline inside Developer Tools. Give a suitable name to the pipeline you want to create.

And click on next ...



#### Step 4: Github connection

In this step, we are supposed to create a github connection and add our existing repository over


here i.e the one we forked earlier

We are supposed to enter our github username so as to proceed towards making the connection

**Source**

**Source provider**  
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (Version 2)

 **New GitHub version 2 (app-based) action**  
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. [Learn more](#)

**Connection**  
Choose an existing connection that you have already configured, or create a new one and then return to this task.


or Connecting

Now to finalize our connection, we are to install an application which connects AWS to our github account and repository.

Post the establishment of the connection, this is the message that is displayed. We can further select the branch of our repository that we want to connect.

**Connection**  
Choose an existing connection that you have already configured, or create a new one and then return to this task.

`arn:aws:codeconnections:eu-north-1:010928207735:connection/59473412-4` X or Connect to GitHub

 **Ready to connect**  
Your GitHub connection is ready for use.

**Repository name**  
Choose a repository in your GitHub account.

You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

**Default branch**  
Default branch will be used only when pipeline execution starts from a different source or manually started.

**Output artifact format**  
Choose the output artifact format.

☒ **CodePipeline default**  
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

☐ **Full clone**  
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

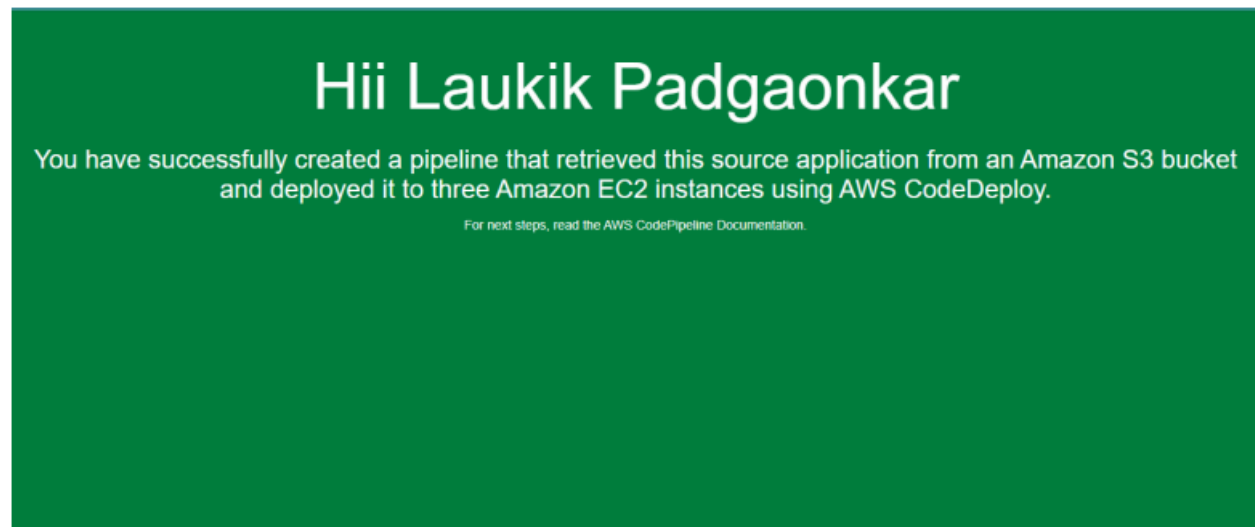
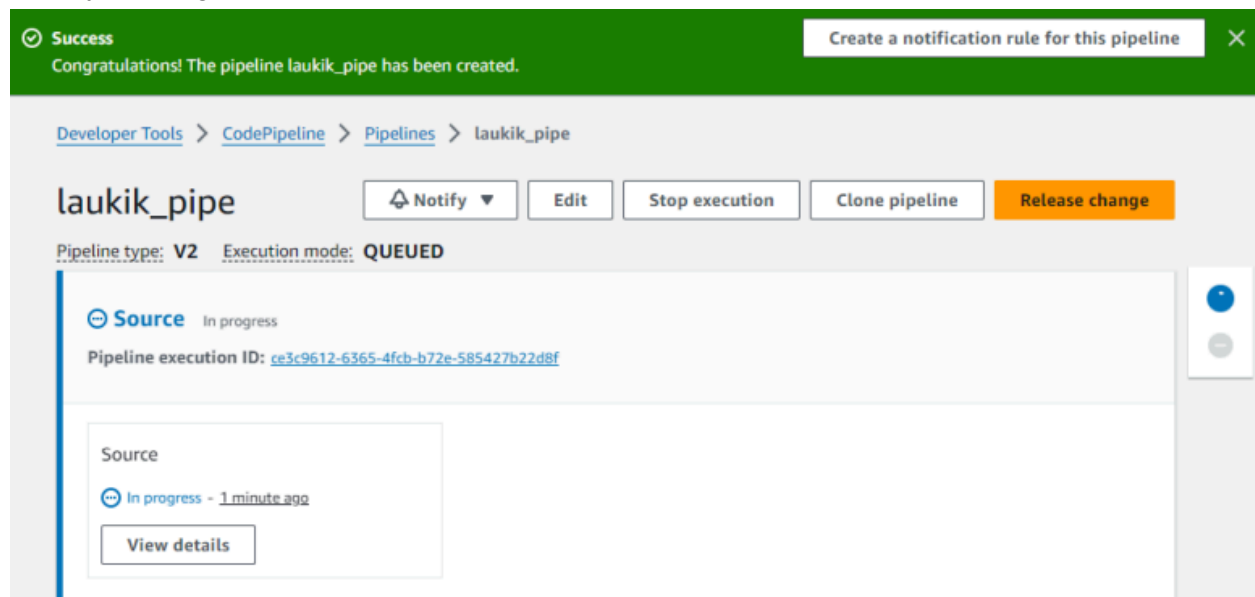
**Step 5: Deployment stage:**

We are expected to skip the build stage and move towards the deployment step. In the deployment step we are supposed to choose the Elastic Beanstalk application and the environment that we created earlier and proceed with our pipeline creation

The screenshot shows the AWS CodePipeline console in the 'ap-south-1' region. The breadcrumb navigation is 'Developer Tools > CodePipeline > Pipelines > Create new pipeline'. The left sidebar shows the steps of the pipeline: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main content area is titled 'Add deploy stage' and is labeled 'Step 4 of 5'. A blue information box states: 'You cannot skip this stage. Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.' Below this, the 'Deploy' section is active. It includes a 'Deploy provider' dropdown set to 'AWS Elastic Beanstalk', a 'Region' dropdown set to 'Asia Pacific (Mumbai)', and an 'Input artifacts' section with a 'Learn more' link.

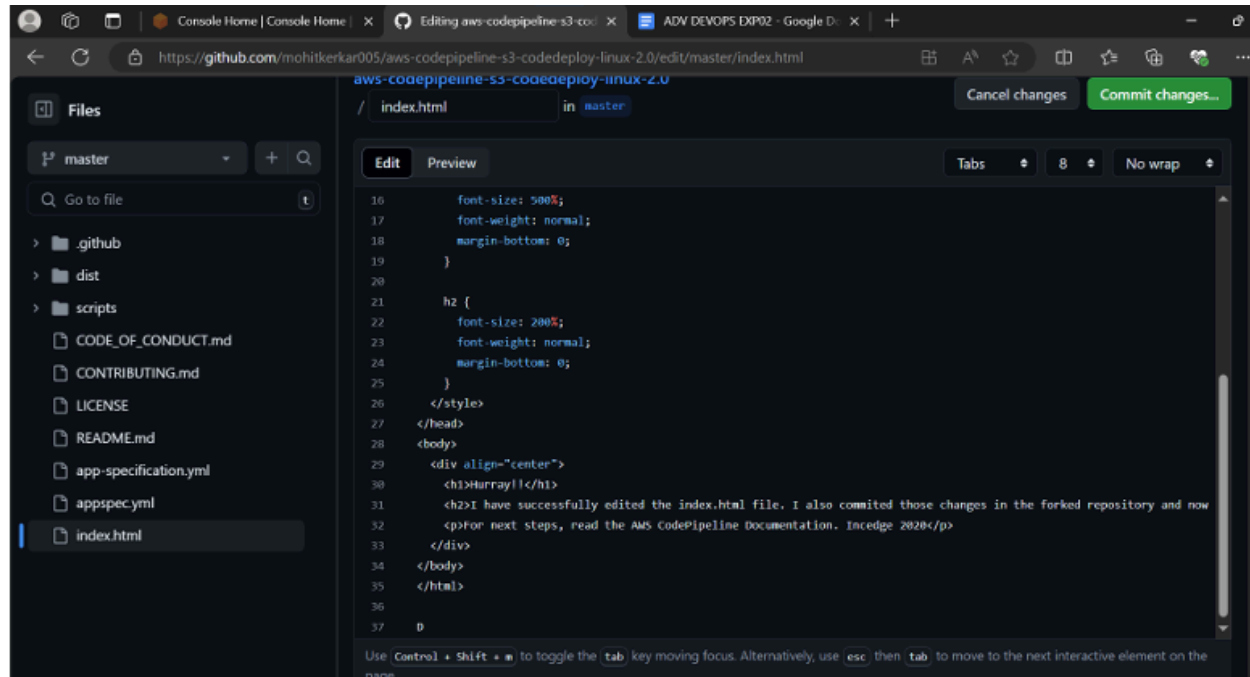
This screenshot shows the 'Input artifacts' and 'Application name' sections of the 'Add deploy stage' configuration. The 'Region' dropdown is set to 'Asia Pacific (Mumbai)'. The 'Input artifacts' section has a dropdown menu and a note 'No more than 100 characters'. The 'Application name' section has a text input field containing 'beanstalk4' and a 'Learn more' link. Below this, the 'Environment name' section has a dropdown menu showing 'Beanstalk4-env-1' and a note 'environment in the AWS Elastic'. There is also a checkbox for 'Configure automatic rollback on stage failure'. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

Step 6: Post deployment stage: When all the stages run successfully, this is what is displayed onto the screen. It shows us that our application and our environment have successfully been deployed using a dedicated pipeline created



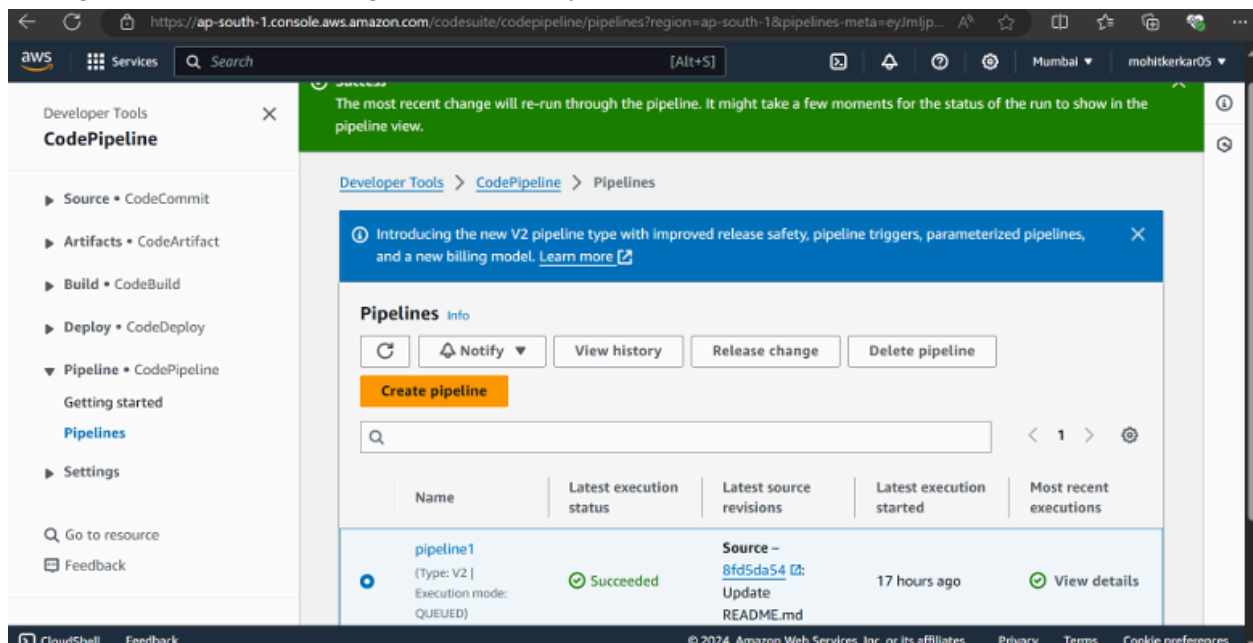
### Step 7: Committing changes to your github code

Now, we will go to our forked repository and make some changes to the index.html file. On making the desired changes, we are supposed to commit those changes on our forked repository. Write a good commit message so as to recognize it when it appears on the pipeline.

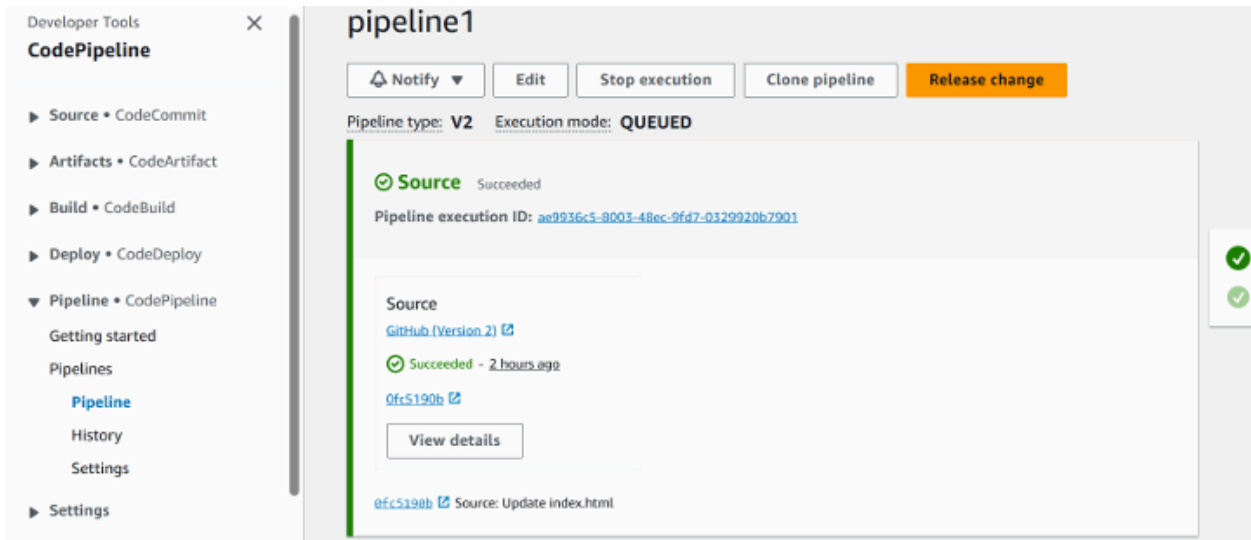


### Step 8: Apply the newly made changes in index.html onto our pipeline

Come back to the Codepipeline section and select the pipeline through which we successfully created and deployed our application. Click on the release change option to apply the latest changes/commits from our github repository to our pipeline



Once the changes have been applied, we see the commit message that we wrote for the latest commit on our repository being reflected on our pipeline. Over here, it would be seen somewhere near the bottom of the image that is attached. "Update index.html" was the latest commit message in the github repository



Step 9: Open the Domain of our Elastic Beanstalk environment

Now, we navigate back to our Elastic Beanstalk environment and open the environment domain of our deployed application

The text in this image is clearly distinguishable from the earlier website's text meaning that the changes that we made to our code in index.html has successfully been applied to the website that we deployed

