

Name: Advik Hegde

Div/Roll No. : D15C/15

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

Step 1: Create our Elastic Beanstalk 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. The rest of the configuration settings are to be kept as default.

Application information [Info](#)

Application name

Maximum length of 100 characters.

► Application tags (optional)

Environment information [Info](#)

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

← ↻ 🔒 https://ap-south-1.console.aws.amazon.com/elasticbeanstalk/home?region=ap-south-1#/create-environment?applicationN...

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

Platform branch

Platform version

Application code [Info](#)

☒ Sample application

☐ Existing version

Application versions that you have uploaded.

☐ Upload your code

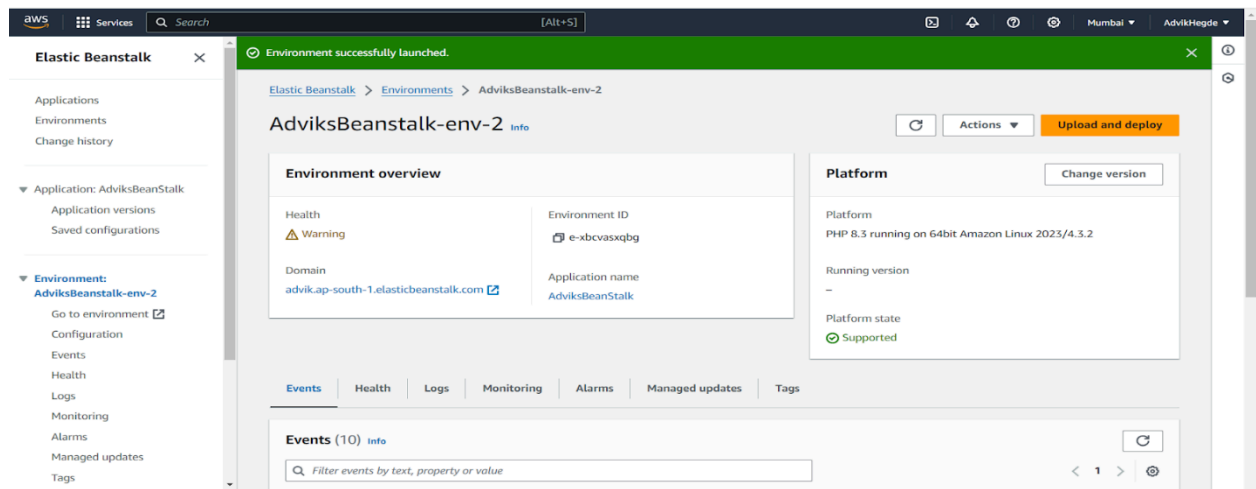
Upload a source bundle from your computer or copy one from Amazon S3.

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:

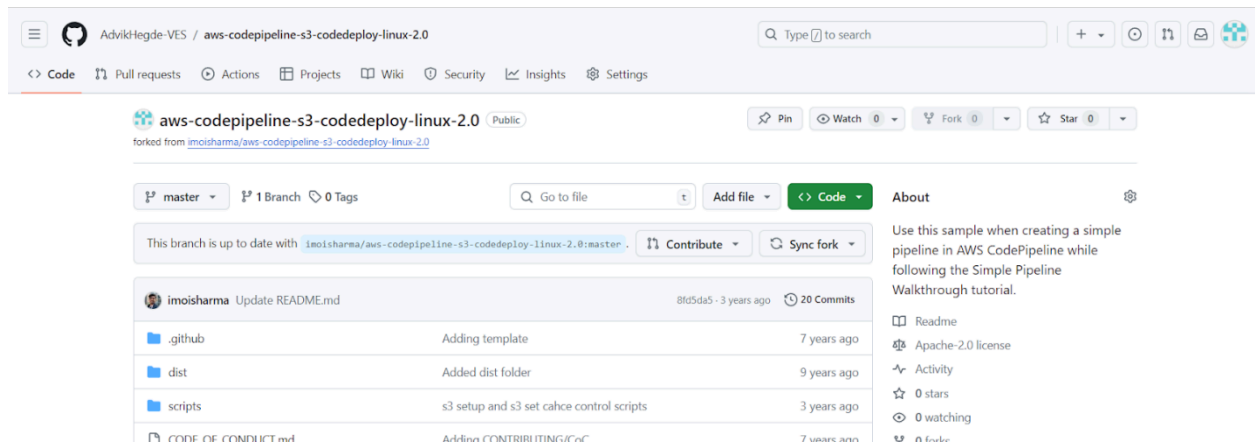
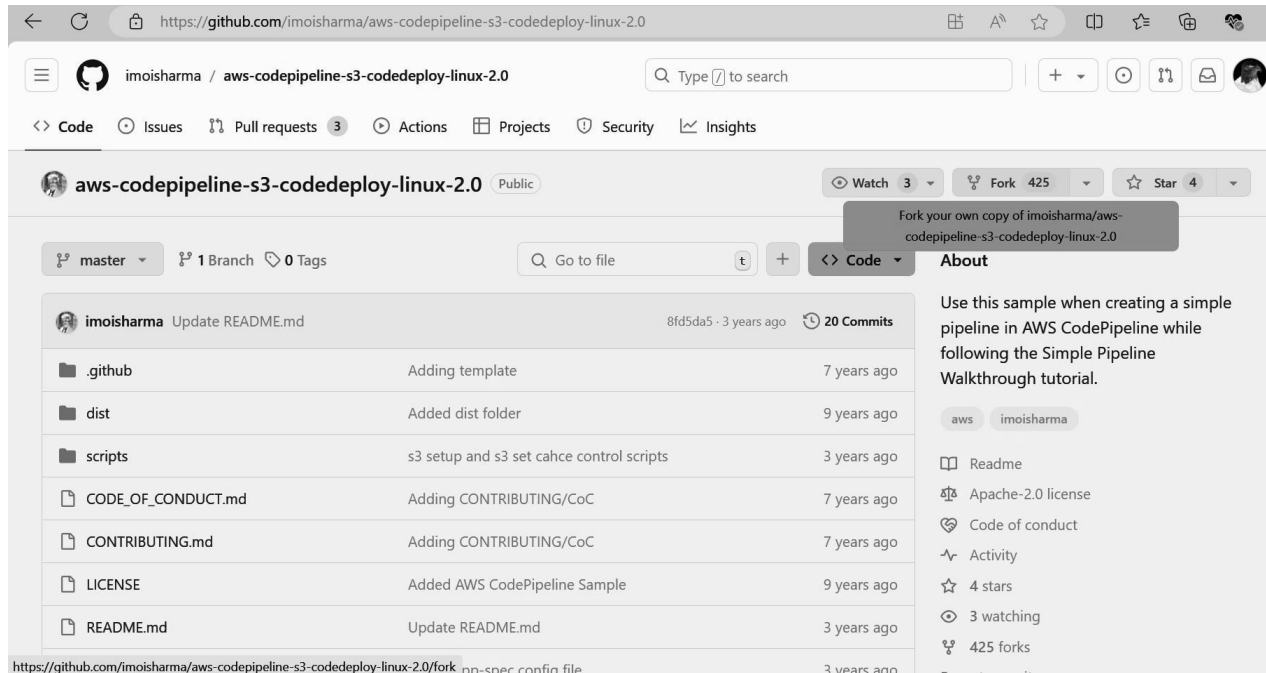
1. AWSElasticBeanStalkWebTier
2. AWSElasticBeanStalkWorkerTier
3. AWSElasticBeanStalkMulticontainerDocker

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.



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.

The screenshot shows the AWS CodePipeline console interface. The top navigation bar includes the AWS logo, 'Services', a search bar, and user information. The breadcrumb trail indicates the path: Developer Tools > CodePipeline > Pipelines > Create new pipeline. The left sidebar lists the steps of the pipeline creation process: Step 1: Choose pipeline settings (selected), Step 2: Add source stage, Step 3: Add build stage, Step 4: Add deploy stage, Step 5: Review. The main content area is titled 'Choose pipeline settings' and 'Step 1 of 5'. It contains the following sections:

- Pipeline settings**
 - Pipeline name**: A text input field containing 'AdviksPipeline'. A message below states: 'Enter the pipeline name. You cannot edit the pipeline name after it is created. No more than 100 characters'.
 - Pipeline type**: A message box states: 'You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.'
- Execution mode**
 - Choose the execution mode for your pipeline. This determines how the pipeline is run.
 - ☐ Superseded: A more recent execution can overtake an older one. This is the default.
 - ☒ Queued (Pipeline type V2 required): Executions are processed one by one in the order that they are queued.
 - ☐ Parallel (Pipeline type V2 required): Executions don't wait for other runs to complete before starting or finishing.

And click on next ...

The screenshot shows the AWS CodePipeline console interface at the 'Add source stage' step. The breadcrumb trail is: Developer Tools > CodePipeline > Pipelines > Create new pipeline. The left sidebar shows the steps: Step 2: Add source stage (selected), Step 3: Add build stage, Step 4: Add deploy stage, Step 5: Review. The main content area is titled 'Add source stage' and 'Step 2 of 5'. It contains the following sections:

- Source**
 - Source provider**: A dropdown menu showing 'GitHub (Version 2)'. A message below states: 'This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.'
 - New GitHub version 2 (app-based) action**: A message box states: '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**: A text input field with a search icon, followed by 'or' and a 'Connecting' button.
 - Repository name**: A text input field with a search icon. A message below states: '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**: A text input field with a search icon. A message below states: 'Default branch will be used only when pipeline execution starts from a different source or manually started.'

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.

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.



arn:aws:codeconnections:ap-south-1:221082173765:connection/479335fc-9 X

or

Connect to GitHub



Ready to connect

Your GitHub connection is ready for use.

Repository name

Choose a repository in your GitHub account.



AdvikHegde-VES/aws-codepipeline-s3-codedeploy-linux-2.0 X

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.



master| X

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



Services



More ▾



[Developer Tools](#) > ... > Create connection



Create a connection [Info](#)

Create GitHub App connection [Info](#)

Connection name

► **Tags - optional**

Connect to GitHub



CloudShell

[Feedback](#)

[Privacy](#)

[Terms](#)

[Cookie preferences](#)

© 2024, Amazon Web Services, Inc. or its affiliates.

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.

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.

🔍 arn:aws:codeconnections:ap-south-1:221082173765:connection/479335fc-9 ✕

or

Connect to GitHub



Ready to connect

Your GitHub connection is ready for use.

Repository name

Choose a repository in your GitHub account.

🔍 AdvikHegde-VES/aws-codepipeline-s3-codedeploy-linux-2.0 ✕

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.

🔍 master| ✕

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.

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

Asia Pacific (Mumbai)

Input artifacts
Choose an input artifact for this action. [Learn more](#)

No more than 100 characters

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.

AdviksBeanStalk

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.

AdviksBeanstalk-env-2

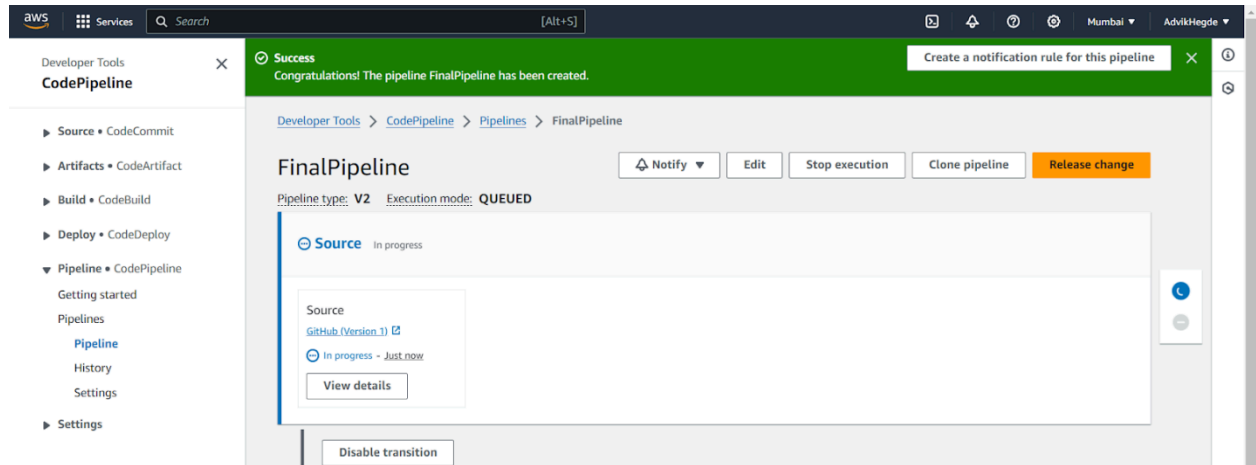
☐ Configure automatic rollback on stage failure

Cancel

Previous

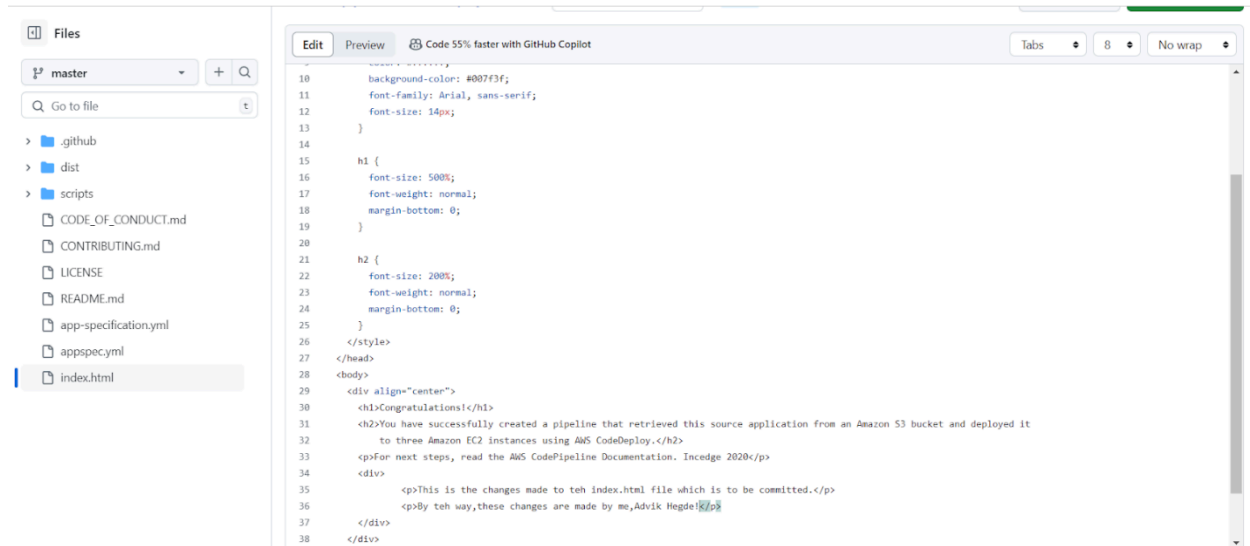
Next

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.



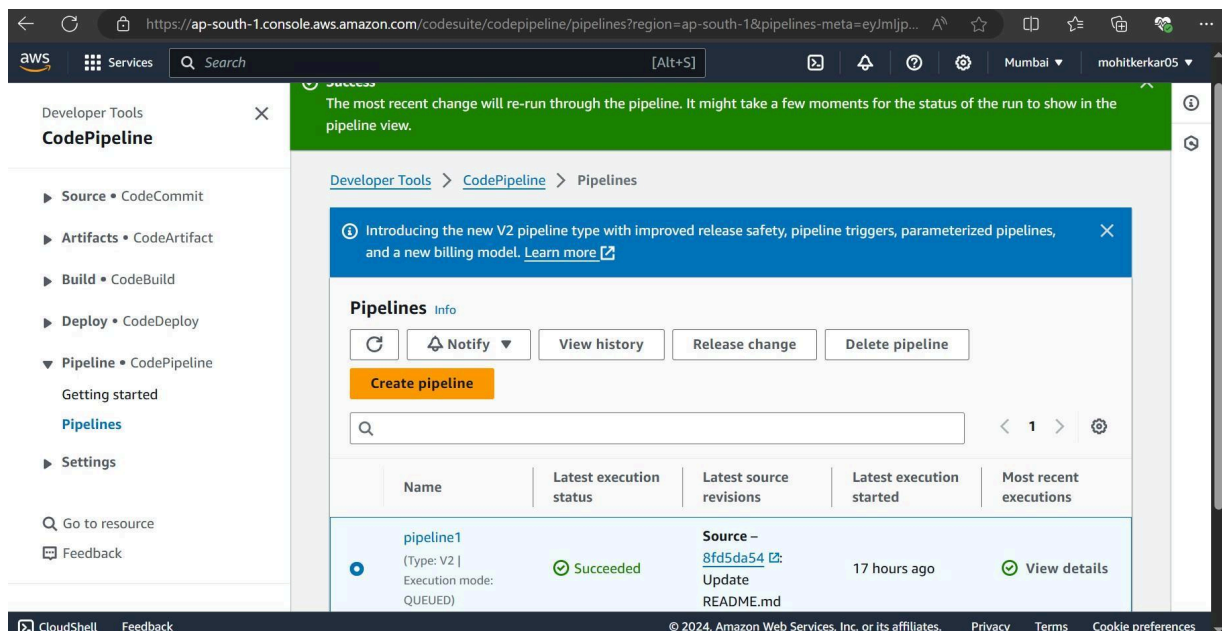
Step 7: Committing changes to your GitHub code

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.

Success
Congratulations! The pipeline FinalPipeline has been created.

Success
The most recent change will re-run through the pipeline. It might take a few moments for the status of the run to show in the pipeline view.

Developer Tools > CodePipeline > Pipelines > FinalPipeline

FinalPipeline

Pipeline type: V2 Execution mode: QUEUED

Notify Edit Stop execution Clone pipeline Release change

Source Succeeded
Pipeline execution ID: 0f9bb793-f278-42d7-8c0c-146557e9b8fb

Source
GitHub (Version 1)
Succeeded - Just now
3cf895ae
View details

3cf895ae Source: Updating the index.html file to contain a div and paragraph elements that are ad ***

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.

Congratulations!

You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.

For next steps, read the AWS CodePipeline Documentation. Incedge 2020
This is the changes made to teh index.html file which is to be committed.
By teh way,these changes are made by me,Advik Hegde!

