Advanced DevOps Lab
<u>Experiment 2</u>

Roll No.	24
Name	Iyer Sreekesh Subramanian
Class	D15-A
Subject	Advanced DevOps Lab

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

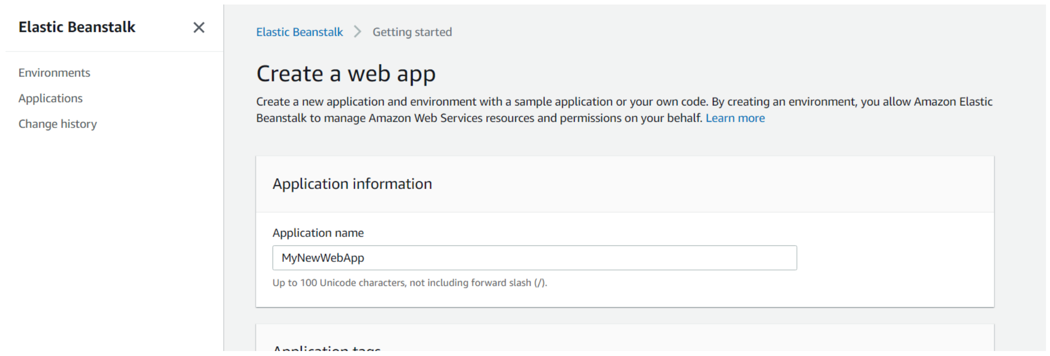
Theory:

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. 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 (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.

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.

1. Open up Elastic Beanstalk and name your web app.



Create a web app

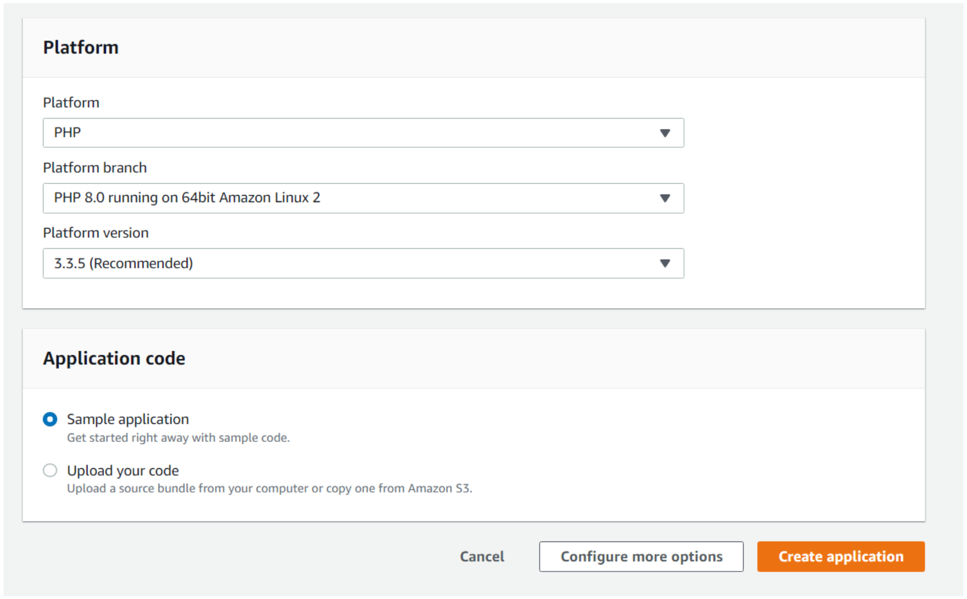
Create a new application and environment with a sample application or your own code. By creating an environment, you allow Amazon Elastic Beanstalk to manage Amazon Web Services resources and permissions on your behalf. Learn more

Application information

Application name: MyNewWebApp

Up to 100 Unicode characters, not including forward slash (/).

2. Choose PHP from the drop-down menu and then click Create Application.



Platform

Platform: PHP

Platform branch: PHP 8.0 running on 64bit Amazon Linux 2

Platform version: 3.3.5 (Recommended)

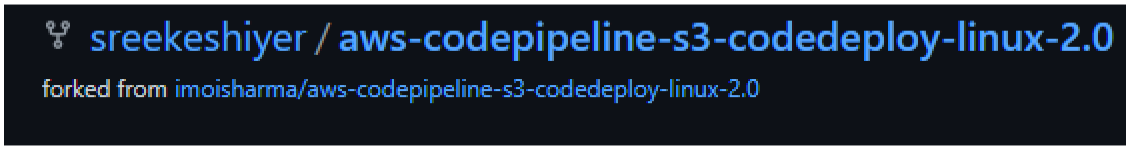
Application code

● Sample application
Get started right away with sample code.

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

3. Beanstalk creates a sample environment for you to deploy your application.
By default, it creates an EC2 instance, a security group, an Auto Scaling group, an Amazon S3 Bucket, Amazon CloudWatch alarms and a domain name for your application.

Step 2: Get a copy of your sample code

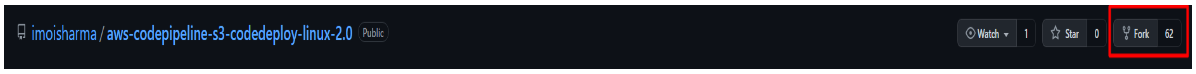


sreekeshiyer / aws-codepipeline-s3-codedeploy-linux-2.0
forked from imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0

In this step, we will get the sample code from [this] GitHub Repository to later host it. The pipeline takes code from the source and then performs actions on it.

For this experiment, as a source, we will use this forked GitHub repository. We can alternatively also use Amazon S3 and AWS CodeCommit.

Go to the repository shared above and simply fork it.

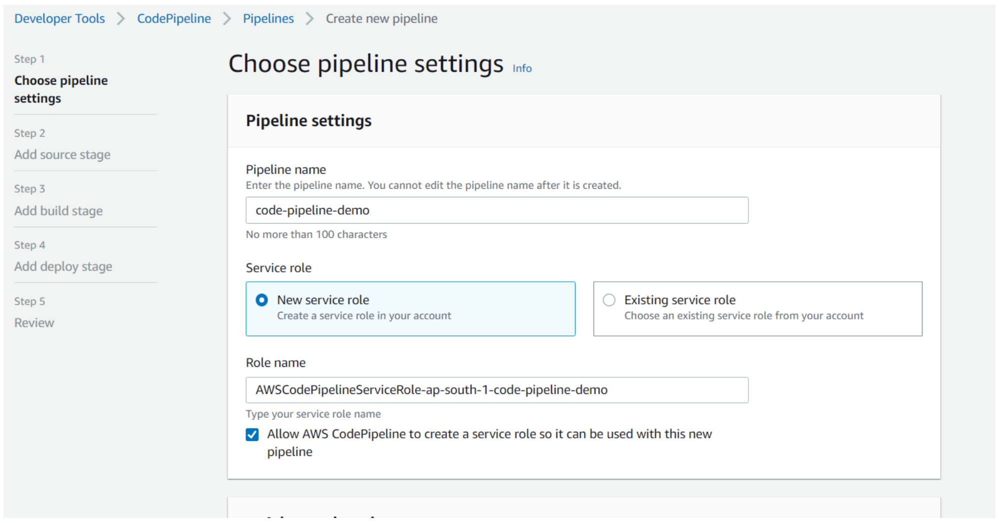


imoisharma / aws-codepipeline-s3-codedeploy-linux-2.0 Public

Step 3: Creating a CodePipeline

In this step, we'll create a simple pipeline that has its source and deployment information. In this case, however, we will skip the build stage where you get to plug in our preferred build provider.

1. Go to AWS Developer Tools -> CodePipeline and create a new Pipeline. Fill in the initial settings first.



Developer Tools > CodePipeline > Pipelines > Create new pipeline

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

Choose pipeline settings Info

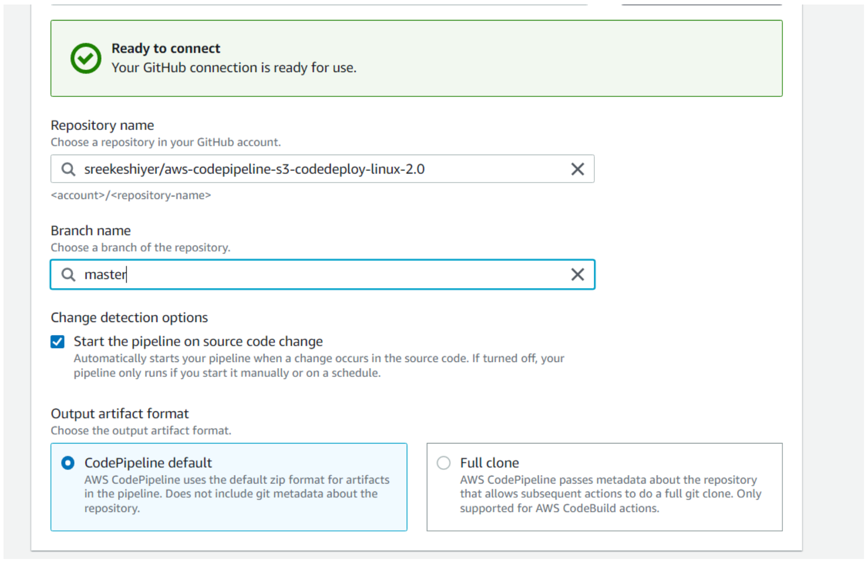
Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
code-pipeline-demo
No more than 100 characters

Service role
[✓ New service role — Create a service role in your account] Existing service role — Choose an existing service role from your account

Role name
AWSCodePipelineServiceRole-ap-south-1-code-pipeline-demo
Type your service role name
☑ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

2. In the source stage, choose GitHub v2 as the provider, then connect your GitHub account to AWS by creating a connection. You'd need your GitHub credentials and then you'd need to authorize and install AWS on the forked GitHub Repository.



Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.
sreekeshiyer/aws-codepipeline-s3-codedeploy-linux-2.0
<account>/<repository-name>

Branch name
Choose a branch of the repository.
master

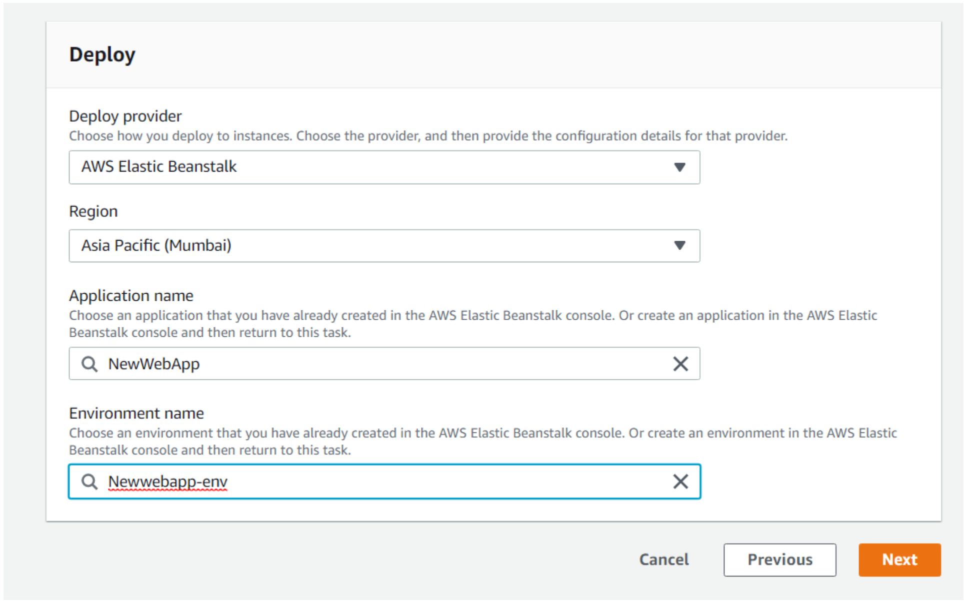
Change detection options
☑ Start the pipeline on source code change
Automatically starts your pipeline when a change occurs in the source code. If turned off, your pipeline only runs if you start it manually or on a schedule.

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.

3. Then, simply choose this forked repository and the branch which you will be able to find in the search box. After that, click Continue and skip the build stage. Proceed to the Deployment stage.

<u>Step 4</u>: Deployment

1. Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name. Click Next, Review and create the pipeline.



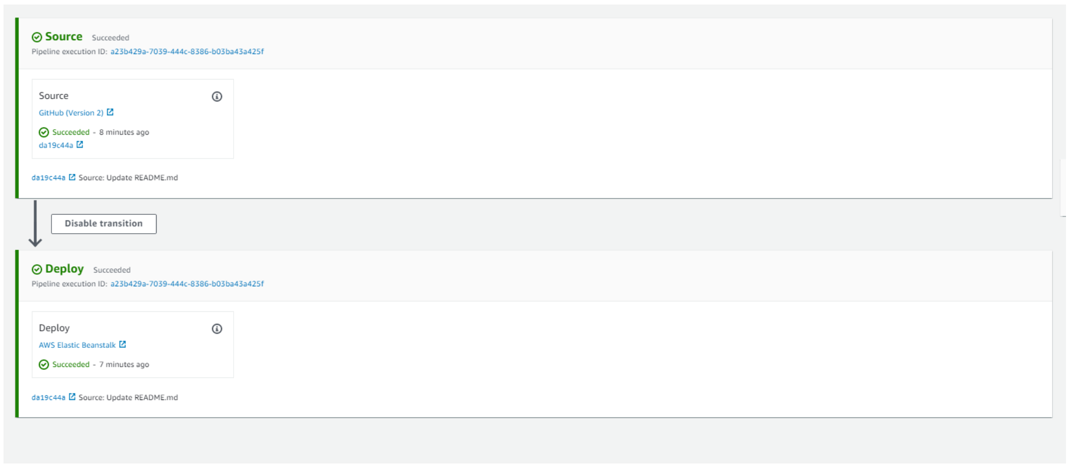
Deploy

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)

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.
NewWebApp

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.
Newwebapp-env



Source Succeeded
Pipeline execution ID: a23b429a-7039-444c-8386-b03ba43a425f

Source
GitHub (Version 2)
Succeeded - 8 minutes ago
da19c44a

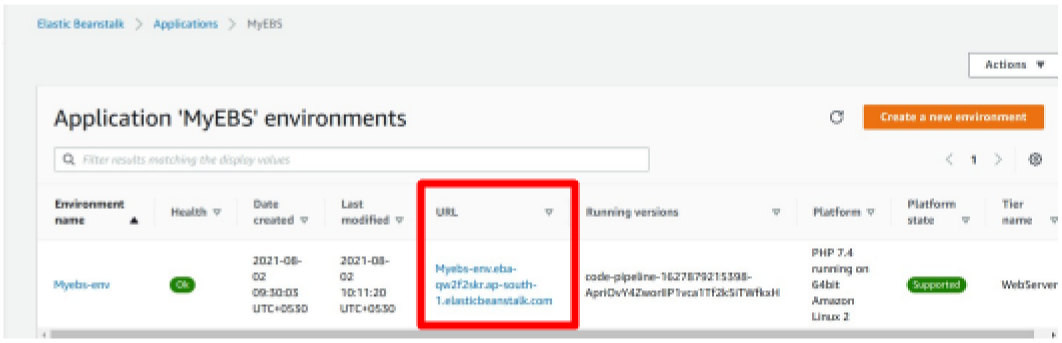
da19c44a Source: Update README.md

Deploy Succeeded
Pipeline execution ID: a23b429a-7039-444c-8386-b03ba43a425f

Deploy
AWS Elastic Beanstalk
Succeeded - 7 minutes ago

da19c44a Source: Update README.md

2. In a few minutes, we will have our pipeline created. Once we have the success message on the Deploy part, we can go ahead and check our URL provided in the EBS environment.

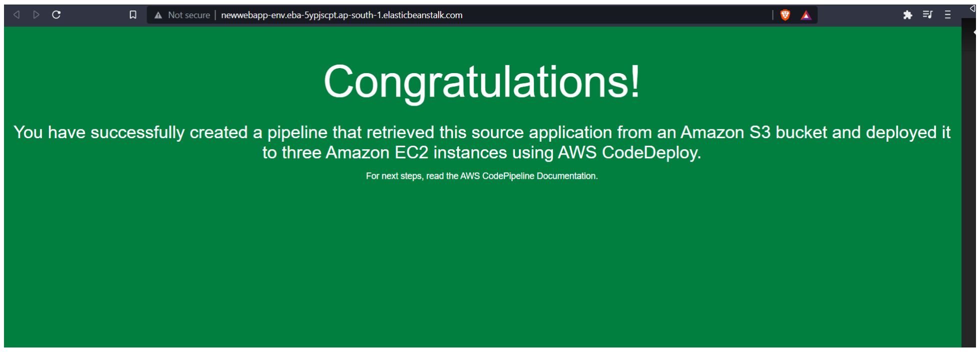


Elastic Beanstalk > Applications > MyEBS

Application 'MyEBS' environments

Environment name	Health	Date created	Last modified	URL	Running versions	Platform	Platform state	Tier name
Myebs-env	Ok	2021-08-02 09:30:05 UTC+0530	2021-08-02 10:11:20 UTC+0530	Myebs-env.eba-qw2f2skr.ap-south-1.elasticbeanstalk.com	code-pipeline-1627879215398-ApriDvY4ZworIP1vca1Tf2k5iTWfkxH	PHP 7.4 running on 64bit Amazon Linux 2	Supported	WebServer

This is the sample website we just created.



Not secure | newwebapp-env.eba-5ypjscpt.ap-south-1.elasticbeanstalk.com

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.

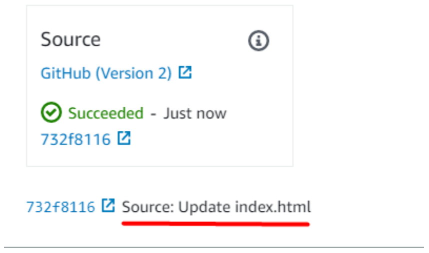
If you can see this, that means that you successfully created an automated software using CodePipeline.

<u>Step 5</u>: Committing changes to update app

1. In this step, we will update the code which we had and make a few changes to the HTML file (keep in mind, this is in our version of the forked repository).

2. In GitHub, open index.html. Then, make changes to either the heading tag or the paragraph tag. Commit these changes on the fly on GitHub.

3. When you commit these changes to your forked version, you'll notice the changes being made in real-time on the Source panel



Source

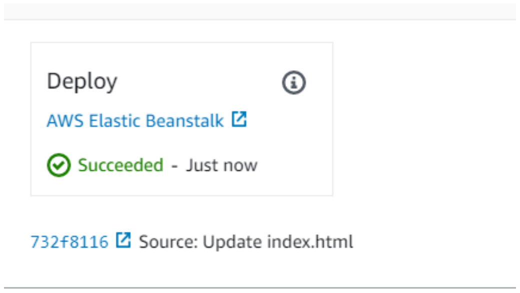
GitHub (Version 2)

Succeeded - Just now

732f8116

732f8116 Source: Update index.html

4. You can view the changes on the website using the same URL, once the deployment section shows success.



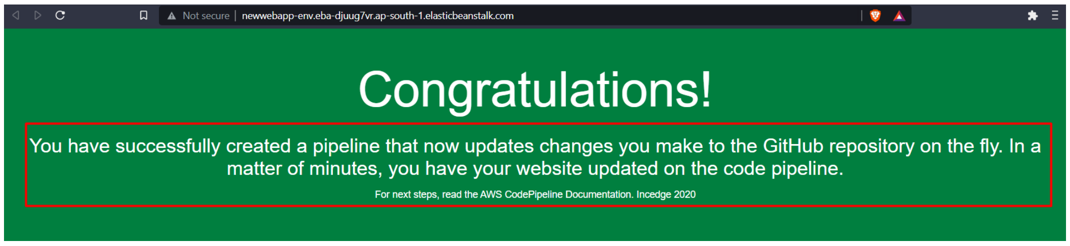
Deploy

AWS Elastic Beanstalk

Succeeded - Just now

732f8116 Source: Update index.html

5. Check the changes live on your website.



Not secure | newwebapp-env.eba-djuug7vr.ap-south-1.elasticbeanstalk.com

Congratulations!

You have successfully created a pipeline that now updates changes you make to the GitHub repository on the fly. In a matter of minutes, you have your website updated on the code pipeline.

For next steps, read the AWS CodePipeline Documentation. Incedge 2020

<u>Conclusion</u>:

In this experiment, we learned how to use AWS Elastic Beanstalk Environments to deploy our websites and create a CodePipeline under the AWS Development Console, source data to the Beanstalk using GitHub and finally, make real-time changes to the website just by pushing updates to GitHub.