Name: Hrishikesh Kumbhar

Div: D15A

Roll no: 32

Sub: Advanced DevOps

Experiment No: 2

Date: 31/07/2022

Experiment No. 2

Aim:- To build your application using AWS Codebuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample application

Theory:-

Continuous deployment allows you to deploy revisions to production environment automatically without explicitly approval from a developer, making the entire software release process automated.

We 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.

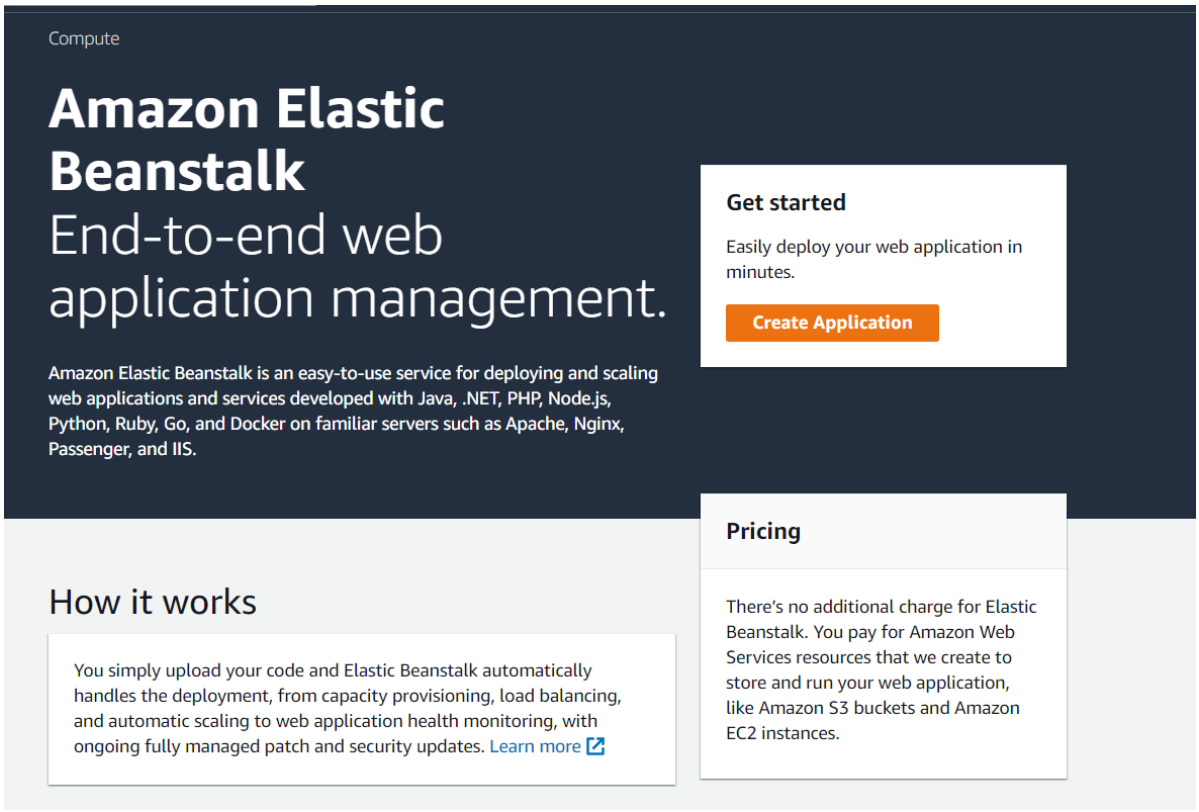
Steps:

Step1: 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) 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. Elastic Beanstalk lets you easily host web applications without needing to launch, configure, or operate virtual servers on your own. It automatically provisions and operates the infrastructure (e.g. virtual servers, load balancers, etc.) and provides the application stack (e.g. OS, language and framework, web and application server, etc.) for you.

2) Name your web app and choose PHP from the drop-down menu(or any other language you are interested in) and then click Create Application.



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

Cerberus

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

Application 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. Learn more

Key	Value	
Cerberus	CICD	Remove tag

Add tag

49 remaining

Platform

Platform

PHP

Platform branch

PHP 8.0 running on 64bit Amazon Linux 2

Platform version

3.3.15 (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.

Cancel | Configure more options | Create application

3) 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.

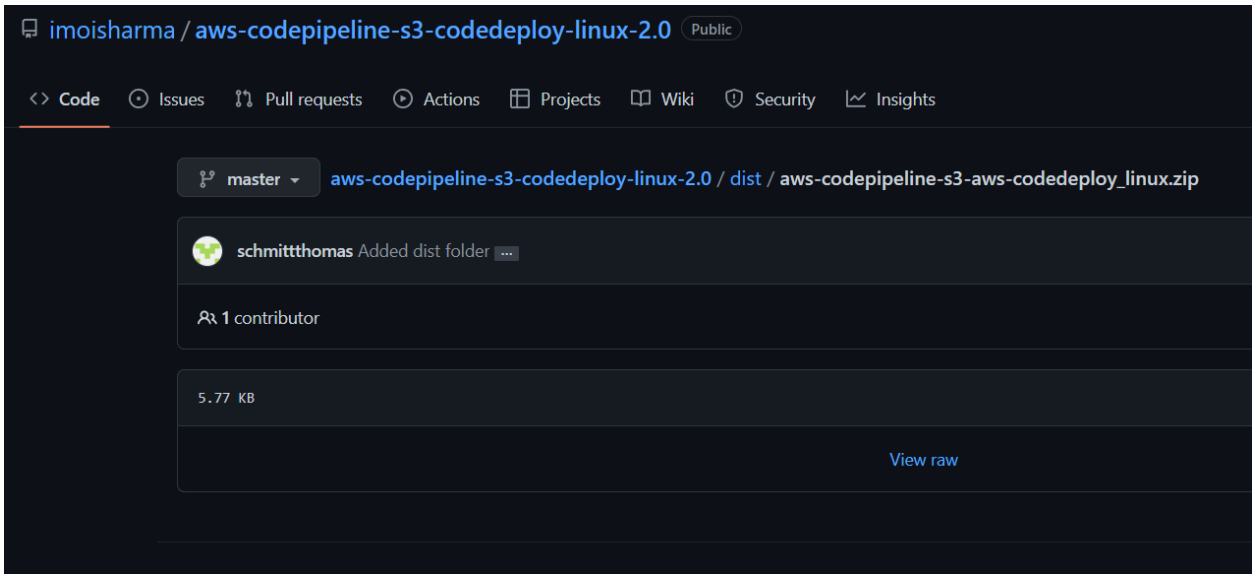
Note: This will take several minutes to complete.

Step2: 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 below:

a. 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/imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0
- Click the dist folder.

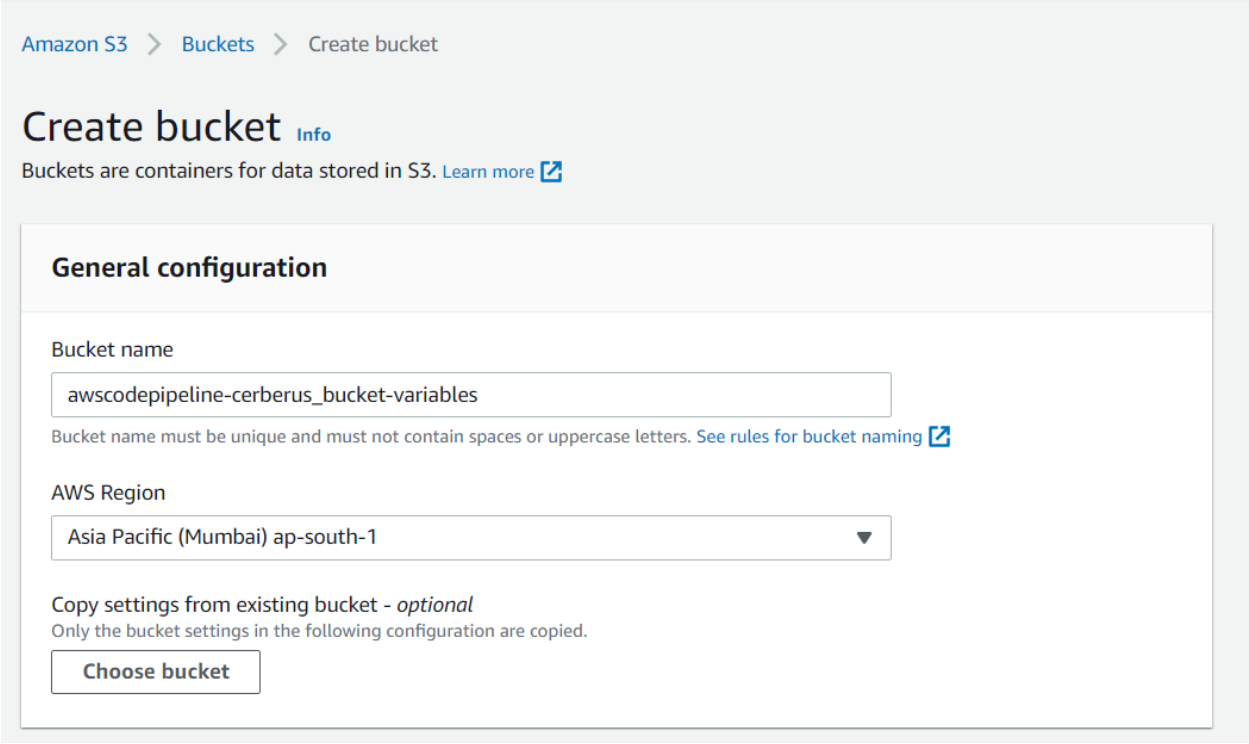


b. Save the source files to your computer:

- Click the file named aws-codepipeline-s3-aws-codedeploy_linux.zip
- Click View Raw.
- Save the sample file to your local computer.

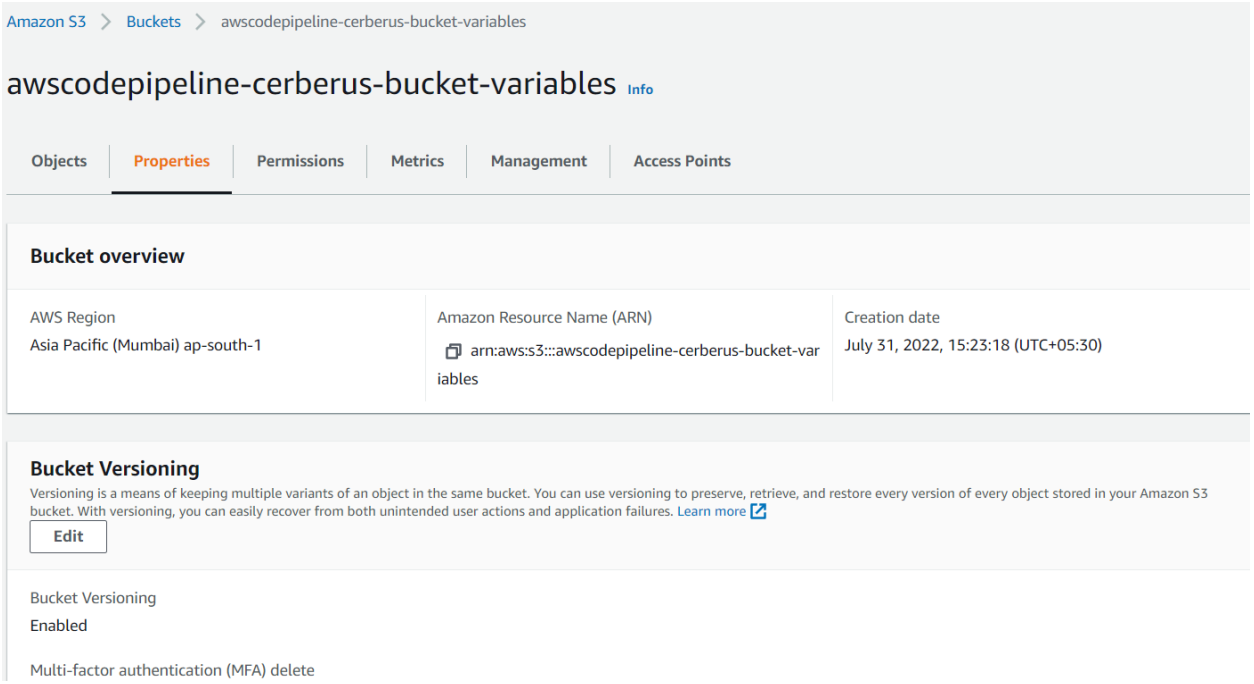
c. open the Amazon S3 console and create your Amazon S3 bucket:
Click Create Bucket

- Bucket Name: type a unique name for your bucket, such as awscodepipeline-cerberus-bucket-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 drop-down, select the region where you will create your pipeline, such as ap-South-1
- Click Create.



d.The console displays the newly created bucket, which is empty.

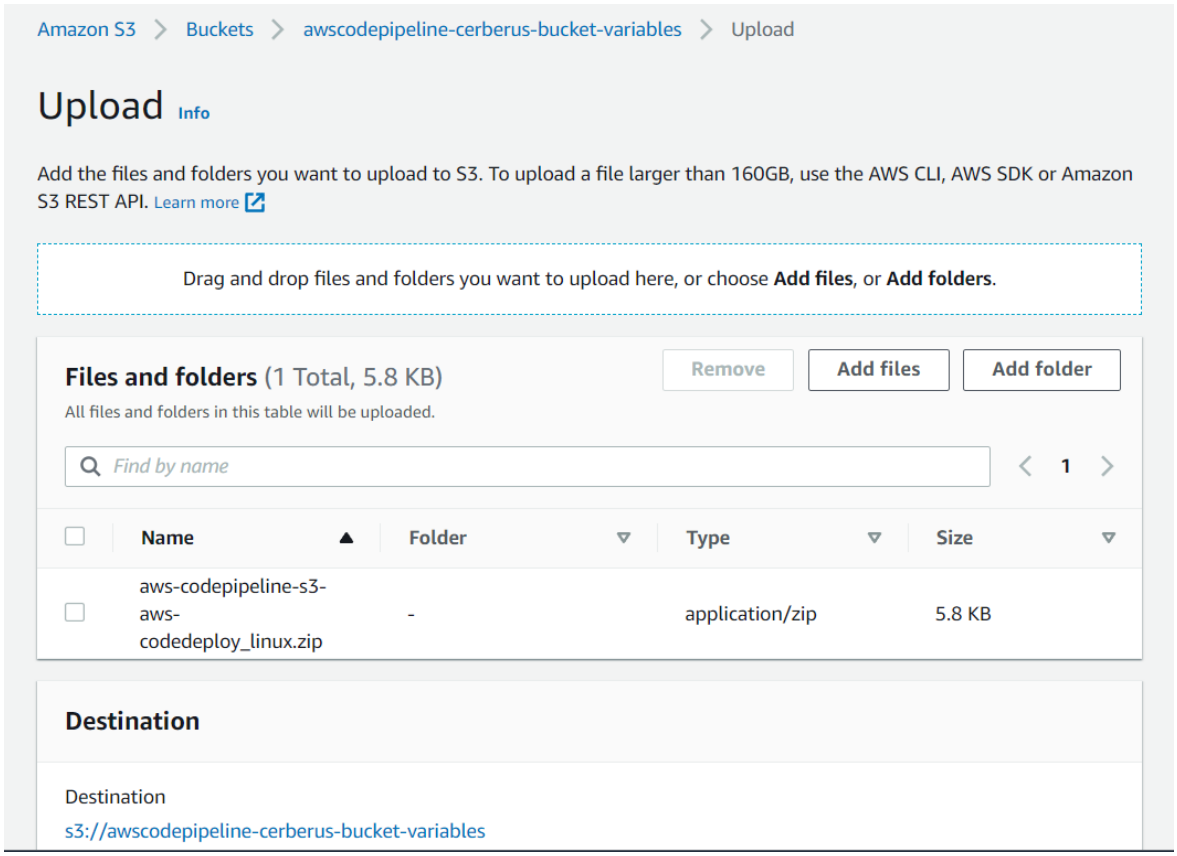
- Click Properties.
- Expand Versioning and select Enable Versioning. When versioning is enabled, Amazon S3 saves every version of every object in the bucket.

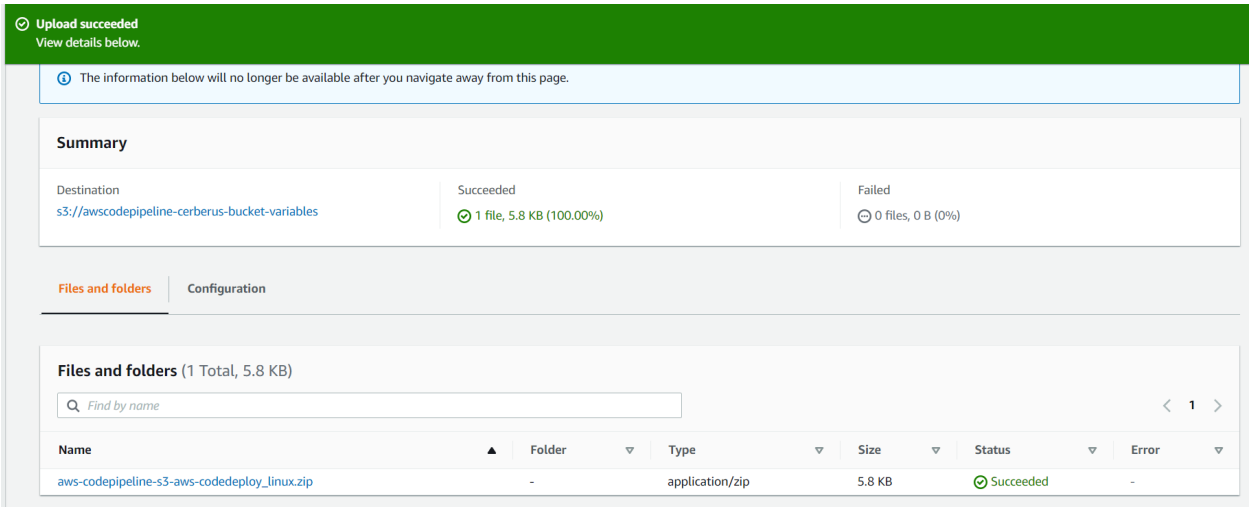


e. You will now upload the sample code to the Amazon S3 bucket:

- Click Upload.
- Follow the on-screen directions to upload the .zip file containing the sample code you downloaded from GitHub.

you can upload directly zip file here from
https://github.com/imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0





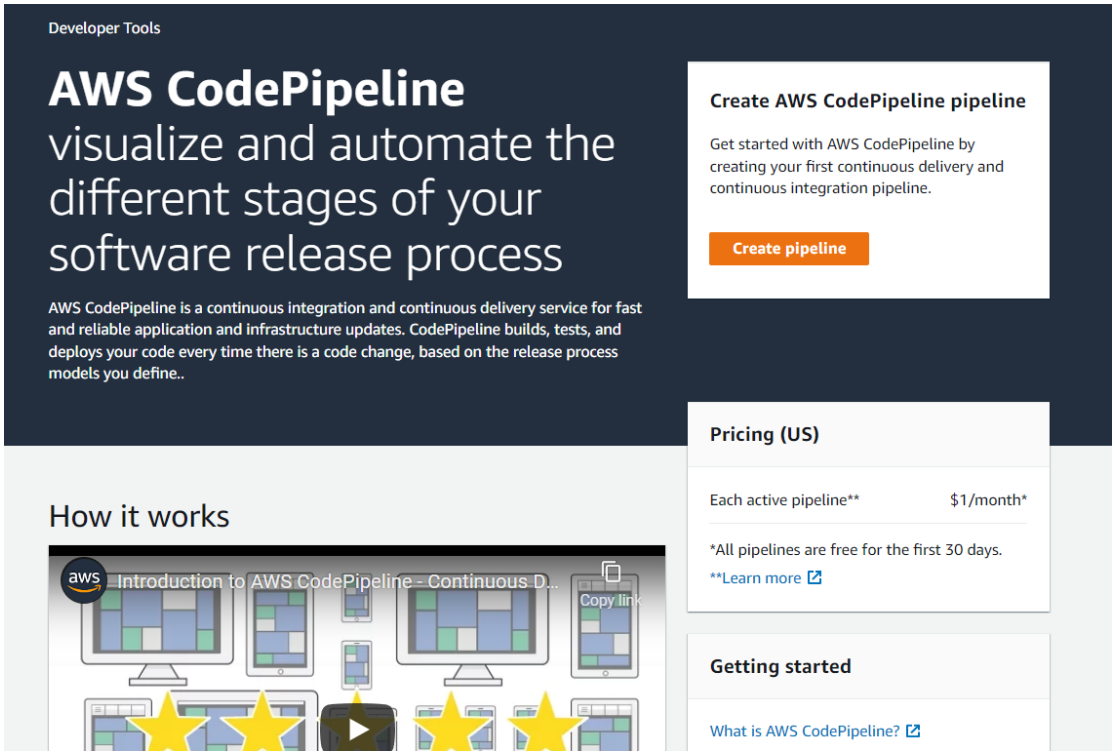
Step3: Create your Pipeline

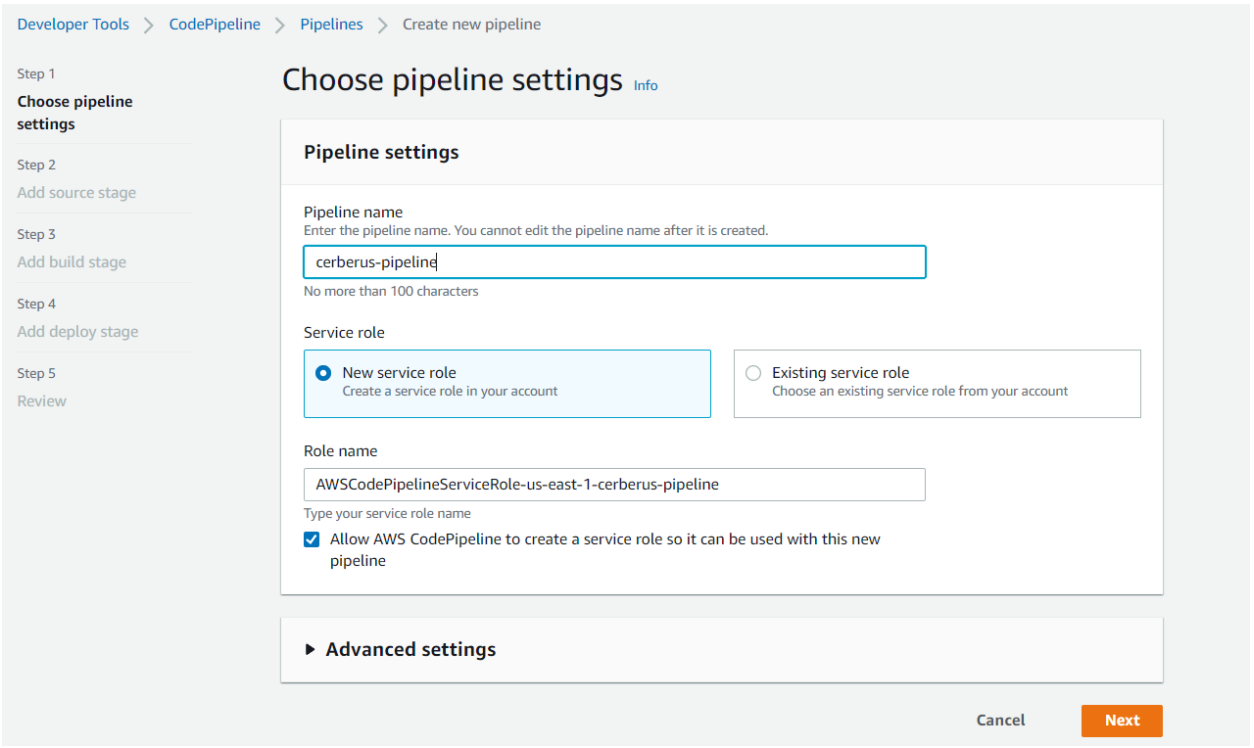
In this step, you will create and configure a simple pipeline with two actions: source and deploy. You will provide CodePipeline with the locations of your source repository and deployment environment.

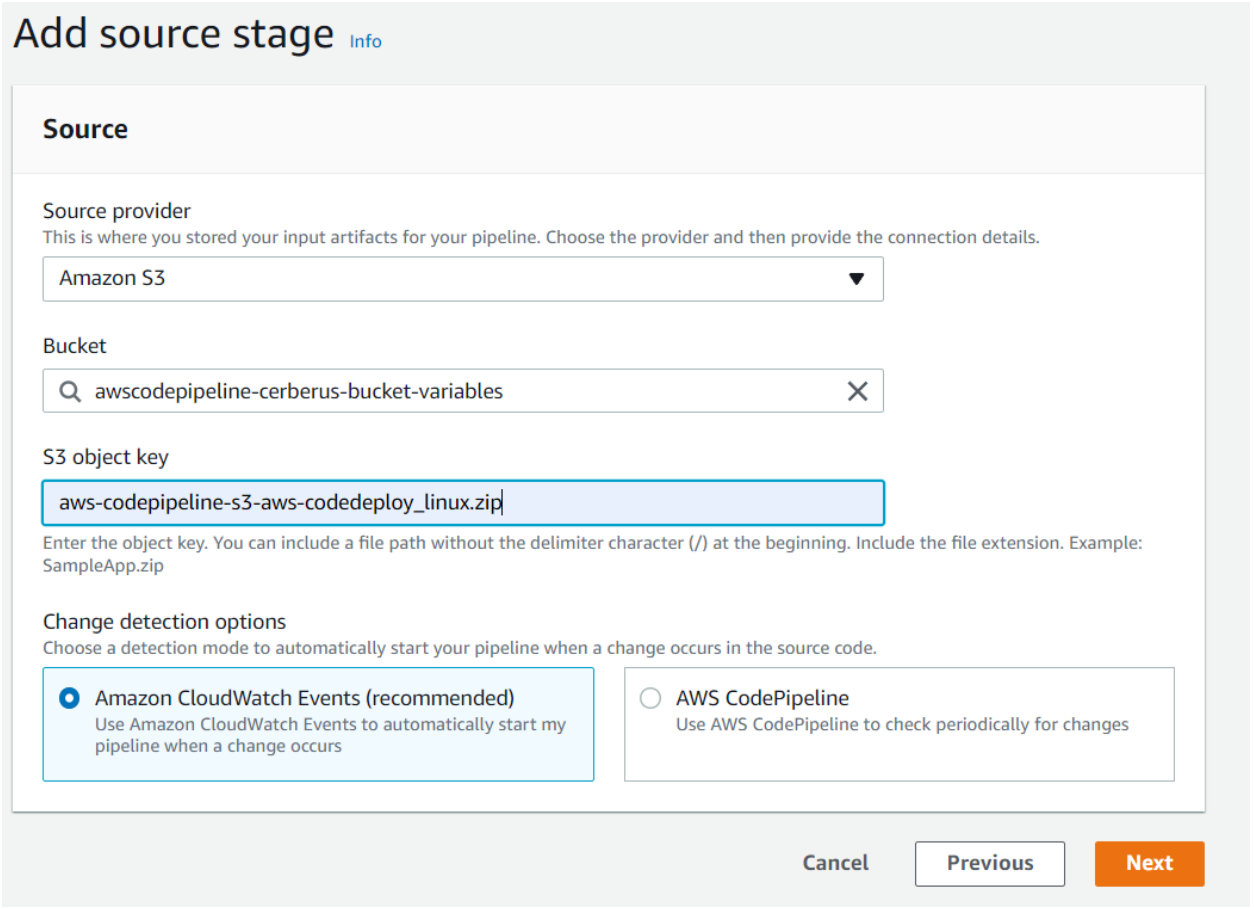
A true continuous deployment pipeline requires a build stage, where code is compiled and unit tested. CodePipeline lets you plug your preferred build provider into your pipeline. However, in this we will skip the build stage.

Goto Pipeline again and create it

Go to Developer tools -> CodePipeline and Click Create Pipeline




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.
cerberus-pipeline
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-us-east-1-cerberus-pipeline
Type your service role name
Allow AWS CodePipeline to create a service role so it can be used with this new pipeline
Advanced settings
Cancel
Next

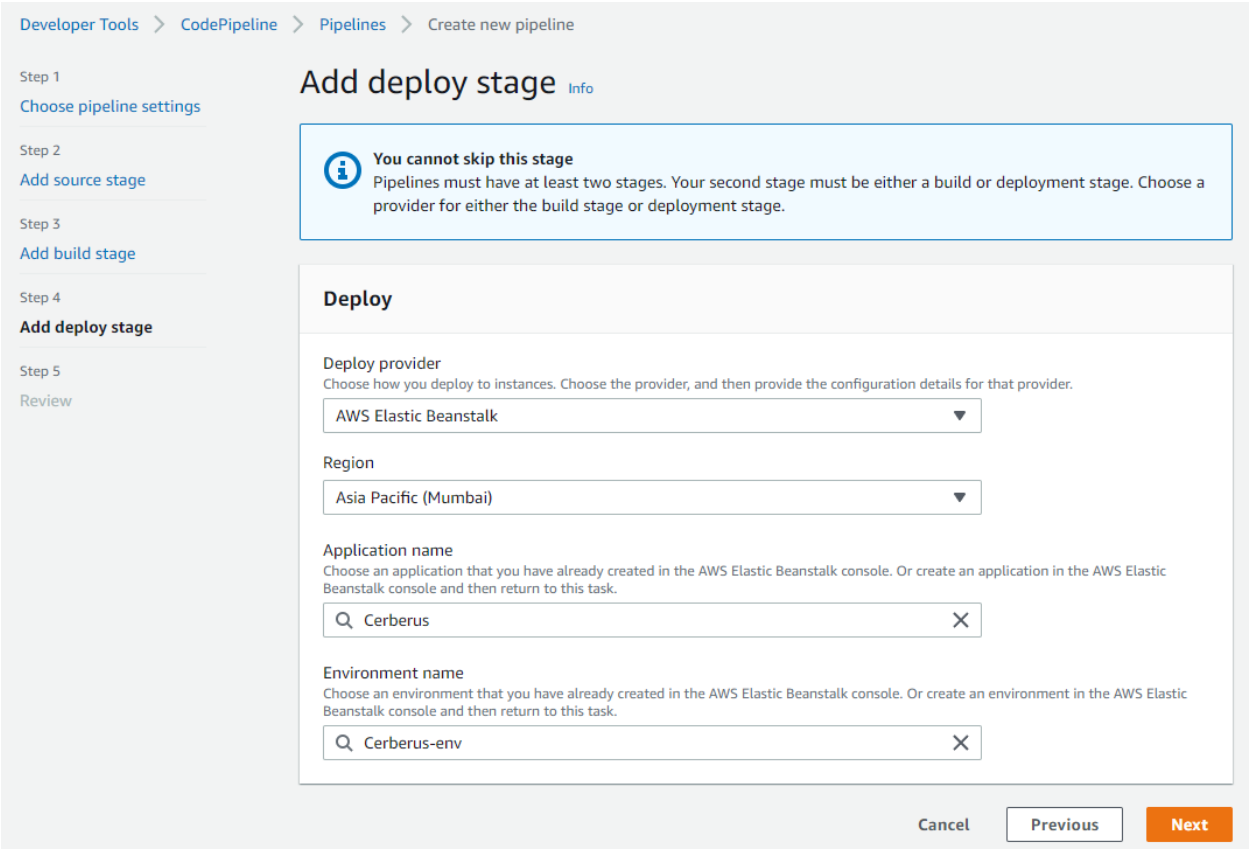

Add source stage Info
Source
Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
Amazon S3
Bucket
awscodepipeline-cerberus-bucket-variables
S3 object key
aws-codepipeline-s3-aws-codedeploy_linux.zip
Enter the object key. You can include a file path without the delimiter character (/) at the beginning. Include the file extension. Example: SampleApp.zip
Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.
Amazon CloudWatch Events (recommended)
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs
AWS CodePipeline
Use AWS CodePipeline to check periodically for changes
Cancel
Previous
Next

In above you can give zip file name in S3 object Key and choose bucket name which you created

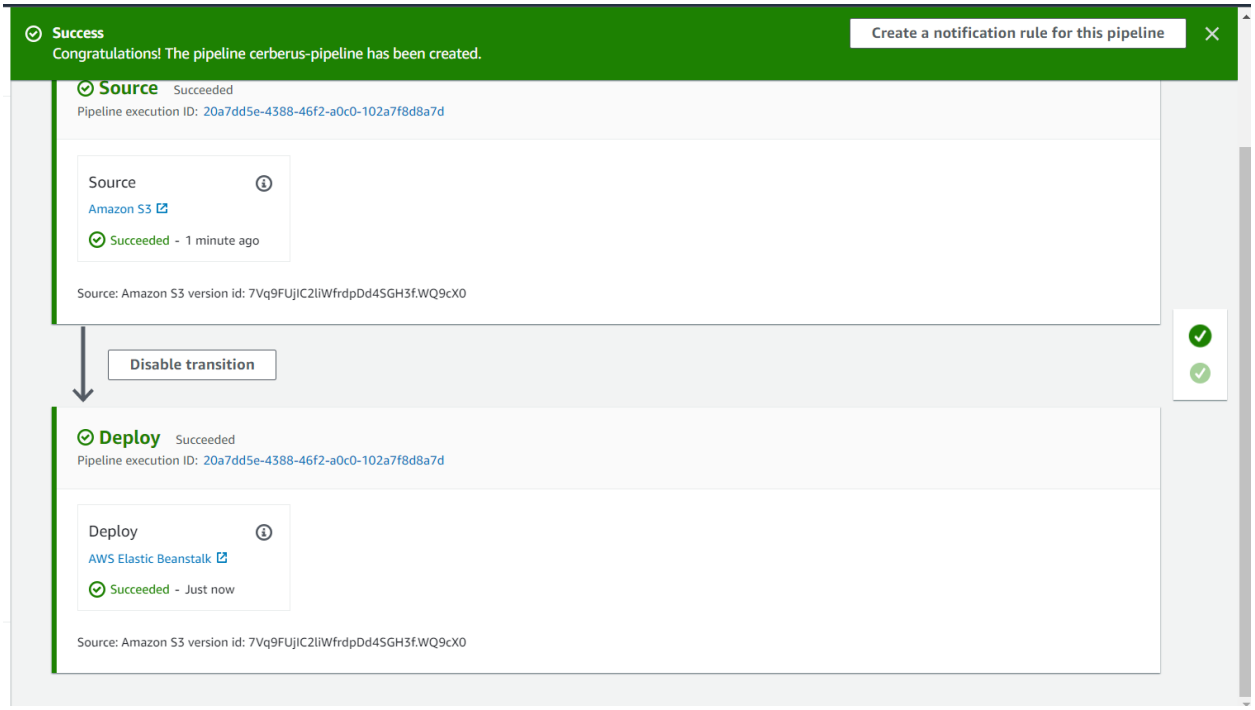
Skip the step 3: add build stage

In Step 4: Deploy Stage:

- Deployment provider: Click AWS Elastic Beanstalk.
- Application name: Cerberus.
- Environment name: Click Cerberus-env.
- Click Next step.



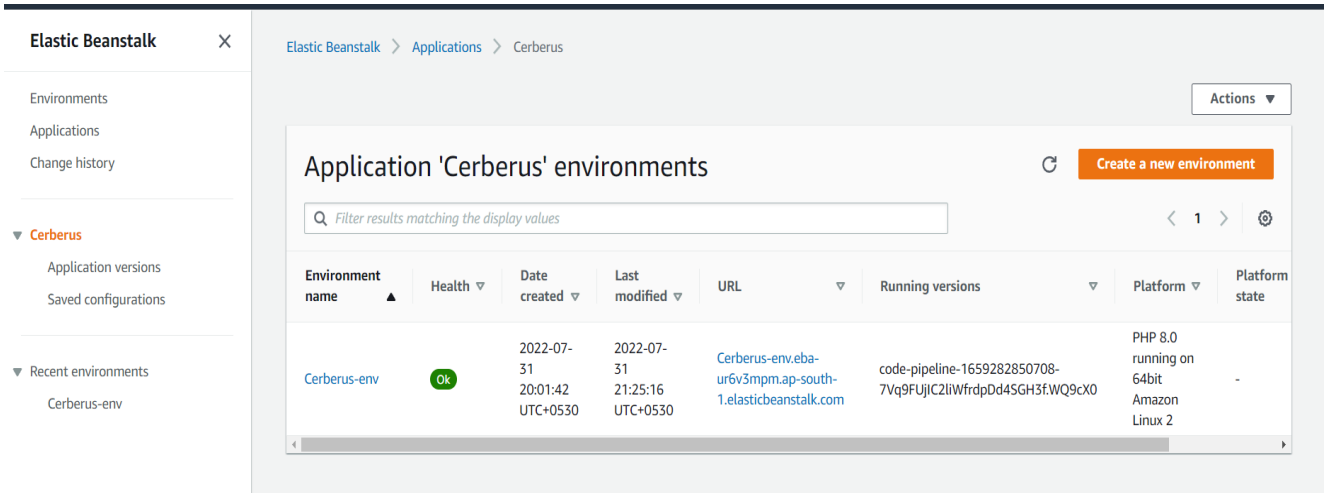
Go to review and click create pipeline



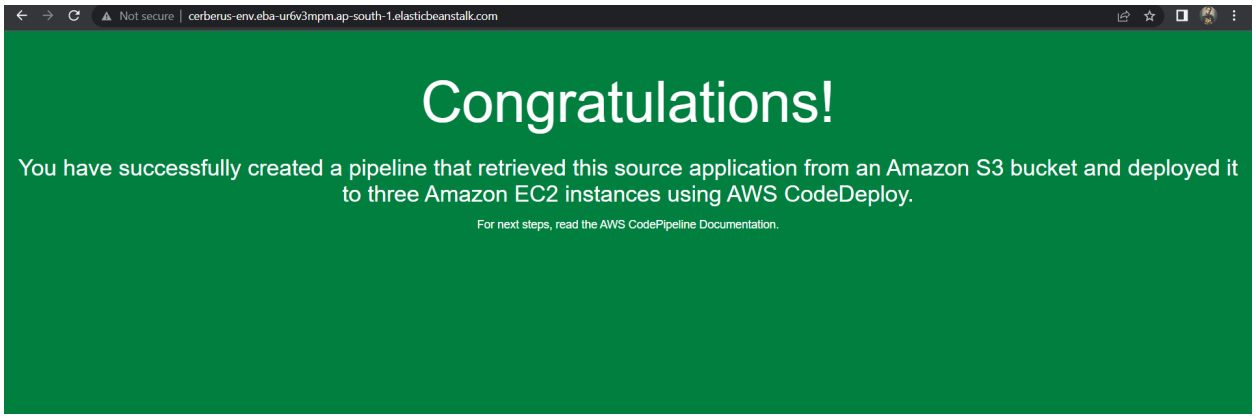
After your pipeline is created, the pipeline status page appears and the pipeline automatically starts to run. You can view progress as well as success and failure messages as the pipeline perform each
action.

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.

Now go to your Cerberus environment and click on the URL to view the sample website you deployed.



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.



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 via 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.

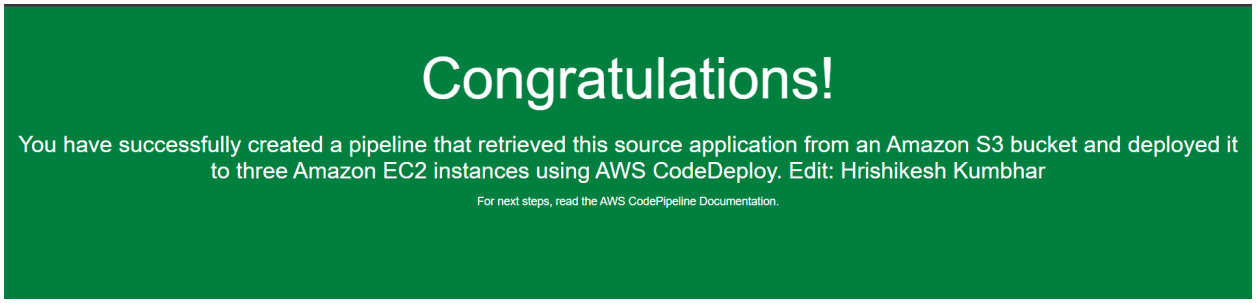
a. Visit your own copy of the repository that you forked in GitHub.
Open index.html Select the Edit icon

b. Update the webpage by copying and pasting the following text on line 30:

c. Commit the change to your repository.

d. 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, click AWS Elastic Beanstalk.

e. The AWS Elastic Beanstalk console opens with the details of the deployment. Select the environment you created earlier. And click the URL again from the Cerberus environment again.



Step 6: Clean up your 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.

a. First, you will delete your pipeline:
In the pipeline view, click Edit.
Click Delete. Type in the name of your pipeline and click Delete.

b. Second, delete your Elastic Beanstalk application:
Visit the Elastic Beanstalk console.
Click Actions. Then click Terminate Environment.

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.

Conclusion:-

Developed a pipeline that pulls application code from GitHub, Amazon S3, or AWS CodeCommit and deploys it to an Amazon EC2 instance run by AWS Elastic Beanstalk. Every time there is a code change, your pipeline will deploy your code automatically.