

**Lab Manual- AWS Elastic Beanstalk sample Project for CodeBuild and Code Pipeline**

**Prepared for**:

**Date:** 18th Nov 2018

**Prepared by:** Bipin Sinhaa Document Name: Lab Manual **Document Number** DevOpsLab401

**Contributor:**

Shruti Sinhaa

Contents

[1. OBJECTIVE 3](#_Toc83150565)

[2. PRE-REQUISISTE 3](#_Toc83150566)

[3. Setup Source File 3](#_Toc83150567)

[4. Setup Codebuild 6](#_Toc83150568)

[5. Setup Elastic Beanstalk 10](#_Toc83150569)

[6. Setup CodePipeline 13](#_Toc83150570)

# **OBJECTIVE**

This sample uses project use AWS CodeBuild with Maven to produce a single WAR file named ROOT.war as the build output. This sample then deploys the WAR file to the instances in an AWS Elastic Beanstalk environment.

# **PRE-REQUISISTE**

▪ Prior knowledge of Maven

▪ Accounts in AWS

▪ A local Computer with 4 CPU, 16 GB RAM, 200 GB disk space

# **Setup Source File**

In this section, you use Maven to produce the source code. Later, you use CodeBuild to build a WAR file based on this source code.

* Download and install Maven
* Switch to an empty directory on your local computer or instance, and then run this Maven command

mvn archetype:**generate** "-DgroupId=**com.mycompany.app**" "-DartifactId=**ROOT**" "-DarchetypeArtifactId=**maven-archetype-webapp**" "-DinteractiveMode=false"

Text

Description automatically generatedText

Description automatically generated

* If successful, this directory structure and files are created.

**Tree**

Text

Description automatically generated

* Create a subdirectory named **.ebextensions** in the ROOT directory. In the **.ebextensions** subdirectory, create a file named **fix-path.config** with this content.

**container\_commands:**

**fix\_path:**

**command: "unzip ROOT.war 2>&1 > /var/log/my\_last\_deploy.log"**

Text

Description automatically generated

* Create a file named **buildspec.yml** with the following contents. Store the file in the **ROOT** directory.

**version: 0.2**

**phases:**

**install:**

**runtime-versions:**

**java: corretto11**

**post\_build:**

**commands:**

**- mvn package**

**- mv target/ROOT.war ROOT.war**

**artifacts:**

**files:**

**- ROOT.war**

**- .ebextensions/\*\*/\***

A screenshot of a computer

Description automatically generated with medium confidence

* Now Create Two S3 Bucket with **versioning enable** in same region .

1. **CodebuildInput**
2. **CodeBuildOutput**

* Create a Folder in Inputbucket name **ddemo**
* Upload the contents of the ROOT directory to an S3 input bucket under the folder **ddemo**

Note :

* Do not upload ROOT, just the directories and files in ROOT.
* If you are using an S3 input bucket, it must be versioned. Be sure to create a ZIP file that contains the directory structure and files, and then upload it to the input bucket. Do not add ROOT to the ZIP file, just the directories and files in ROOT

Graphical user interface, application

Description automatically generated

# **Setup Codebuild**

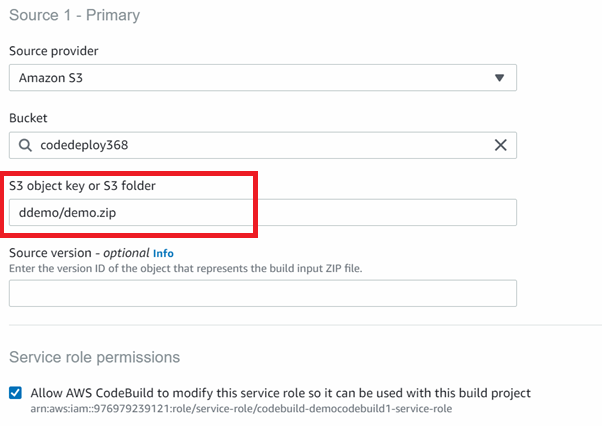
1. Open the AWS CodeBuild console .Use the AWS region selector to choose an AWS Region where CodeBuild is supported. This must be the same Region where your S3 output bucket is stored.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated



* For **Environment**:
  + For **Environment image**, choose **Managed image**.
  + For **Operating system**, choose **Amazon Linux 2**.
  + For **Runtime(s)**, choose **Standard**.
  + For **Image**, choose **aws/codebuild/amazonlinux2-x86\_64-standard:3.0**.

Graphical user interface, text, application, email

Description automatically generated

* For **Artifacts**:
  + For **Type**, choose **Amazon S3**.
  + For **Bucket name**, enter the name of an S3 bucket.
  + For **Name**, enter a build output file name that's easy for you to remember. Include the .zip extension.
  + For **Artifacts packaging**, choose **Zip**.

Graphical user interface, text, application, email

Description automatically generated

* Then Create Project.
* Once Project Create Click **Start Build**

Graphical user interface, application

Description automatically generated

Text

Description automatically generated

* Check s3 output directory

Graphical user interface, application

Description automatically generated

# **Setup Elastic Beanstalk**

* Create an Elastic Beanstalk application.
* Create an Elastic Beanstalk environment for this application. Leave all settings at their default values, except for these settings.
* For **Platform**, choose **Tomcat**.
  + For **Application code**, choose **Upload your code**, and then choose **Upload**. For **Source code origin**, choose **Public S3 URL**, and then enter the full URL to the build output ZIP file in the output bucket. Choose **Upload**.

Graphical user interface, application, Teams

Description automatically generated

Application

Description automatically generated

Graphical user interface, application, email, Teams

Description automatically generated

Text

Description automatically generated

* Once After Elastic Beanstalk deploys the build output to the environment, you can see the results in a web browser. Go to the environment URL for the instance

Graphical user interface, application

Description automatically generated

* example, http://*my-environment-name*.*random-string*.*region-ID*.elasticbeanstalk.com). The web browser should display the text Hello World!.

Graphical user interface, text, application

Description automatically generated

# **Setup CodePipeline**

[ for this step create new Elastic Beanstalk with same setting ]

In this step, you use the AWS CodePipeline console to create a pipeline. After you create and run the pipeline, CodePipeline uses CodeBuild to build the source code. CodePipeline then uses Elastic Beanstalk to deploy the build output to the environment.

* Create an Elastic Beanstalk application.
* Create an Elastic Beanstalk environment for this application. Leave all settings at their default values, except for these settings.

Graphical user interface, text, application, email

Description automatically generated

‘Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* On **Add build stage**, for **Build provider**, choose **AWS CodeBuild**. For **Project name**, choose the build project you just created.

Graphical user interface, application

Description automatically generated

* On **Add deploy stage**, for **Deploy provider**, choose **AWS Elastic Beanstalk**.
  + For **Application name**, choose the Elastic Beanstalk application you just created.
  + For **Environment name**, choose the environment you just created.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

After the pipeline has run successfully, you can see the results in a web browser. Go to the environment URL for the instance (for example, http://*my-environment-name*.*random-string*.*region-ID*.elasticbeanstalk.com). The web browser should display the text Hello World!.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated