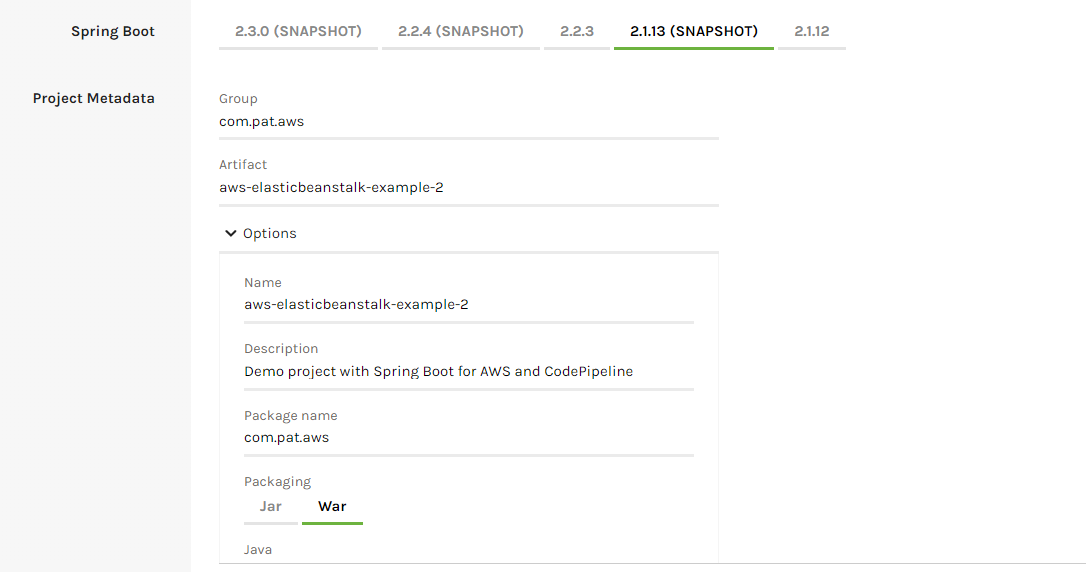
**CICD using AWS CodePipeline and Elastic Bean Stalk**

**Step 1:**

**Creating spring boot demo code template:**

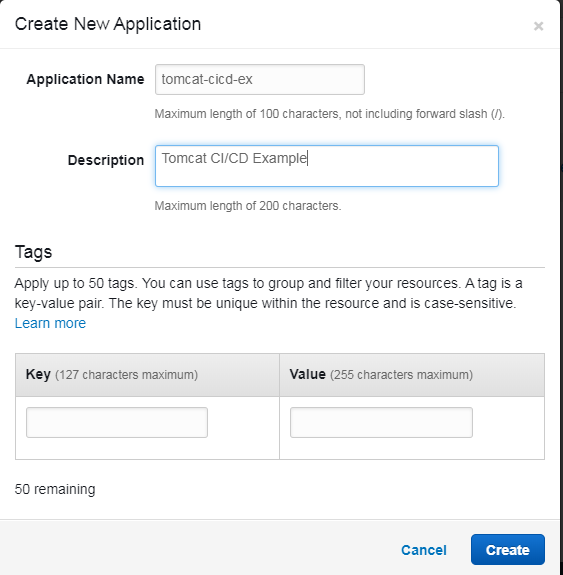


**Taking local Build**

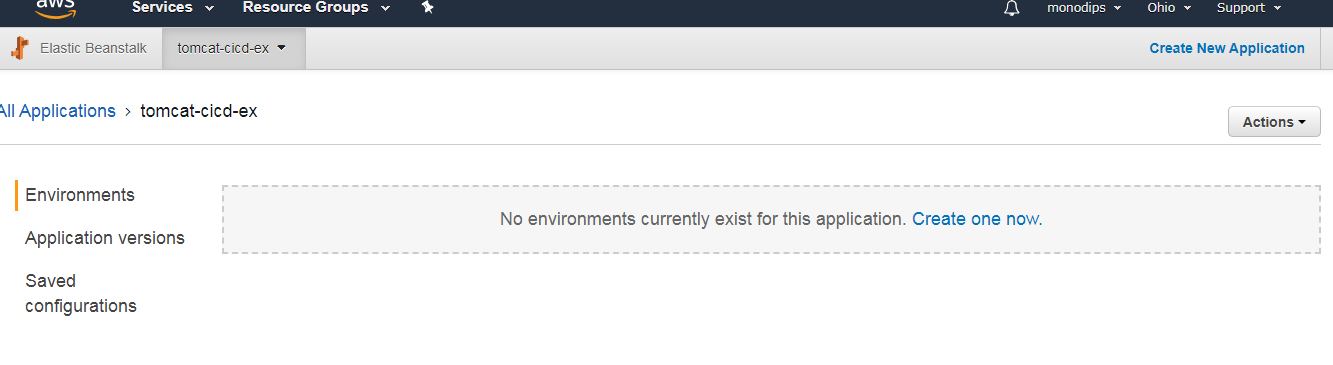
**Elastic Beanstalk:**

**Create Application in Beanstalk:**

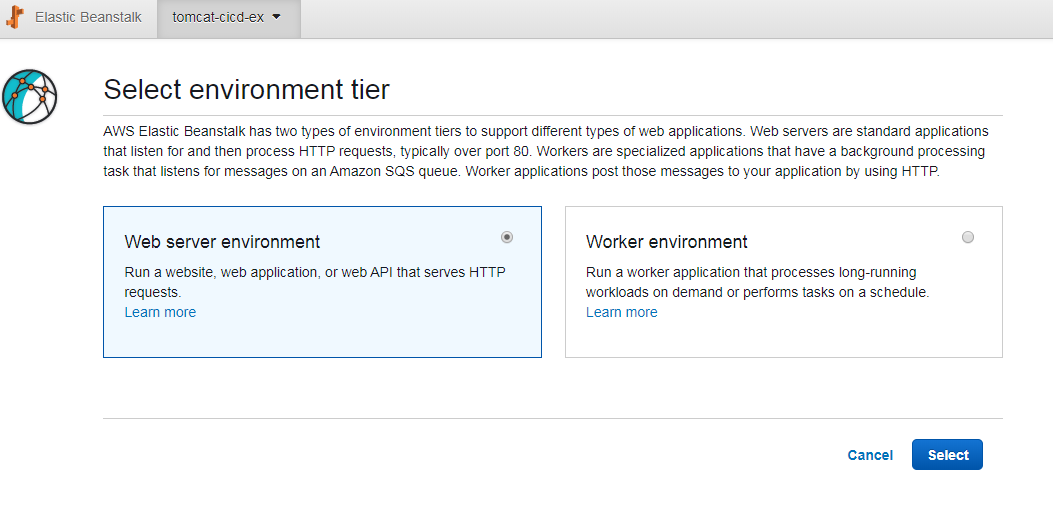
1. Click on Create New Application.



1. Click on Create.



1. Under Application, create new Environment.



1. Create Web server environment.

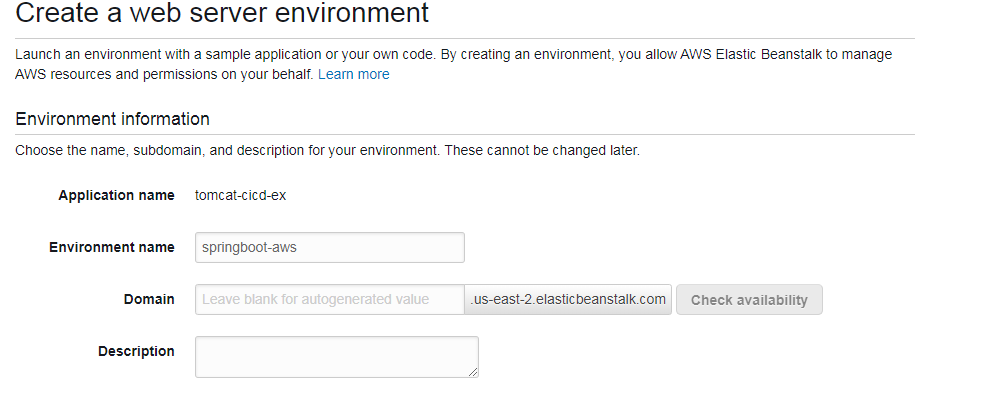
**With Below Details:**

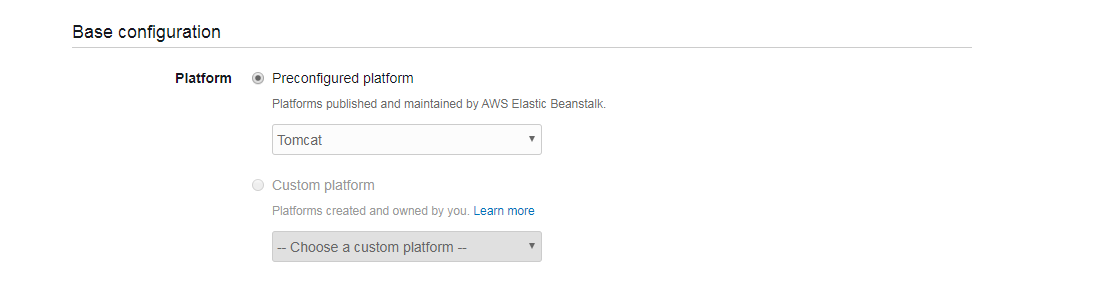
**Environment name:** springboot-aws

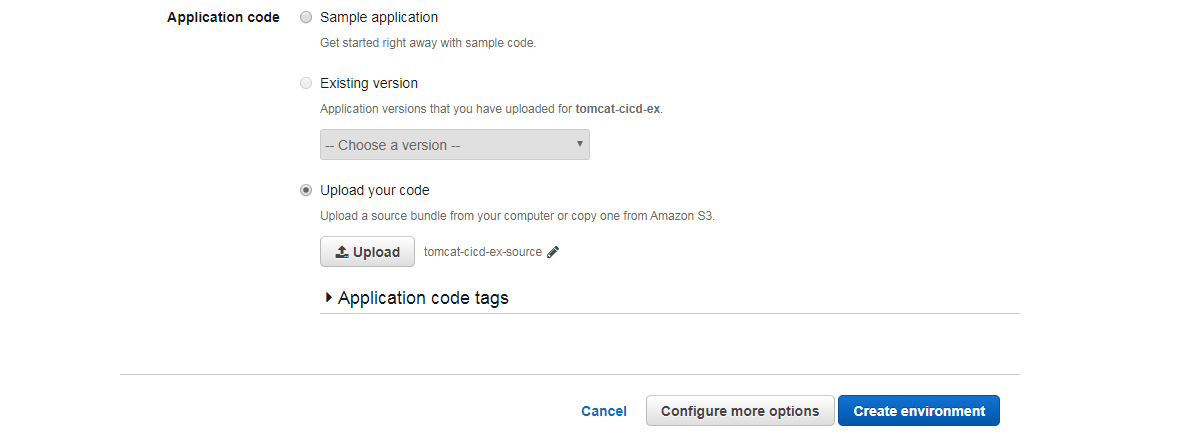
**Platform:** Tomcat

**Application code :** upload your code

Then uploading war file which is war file of spring boot application.

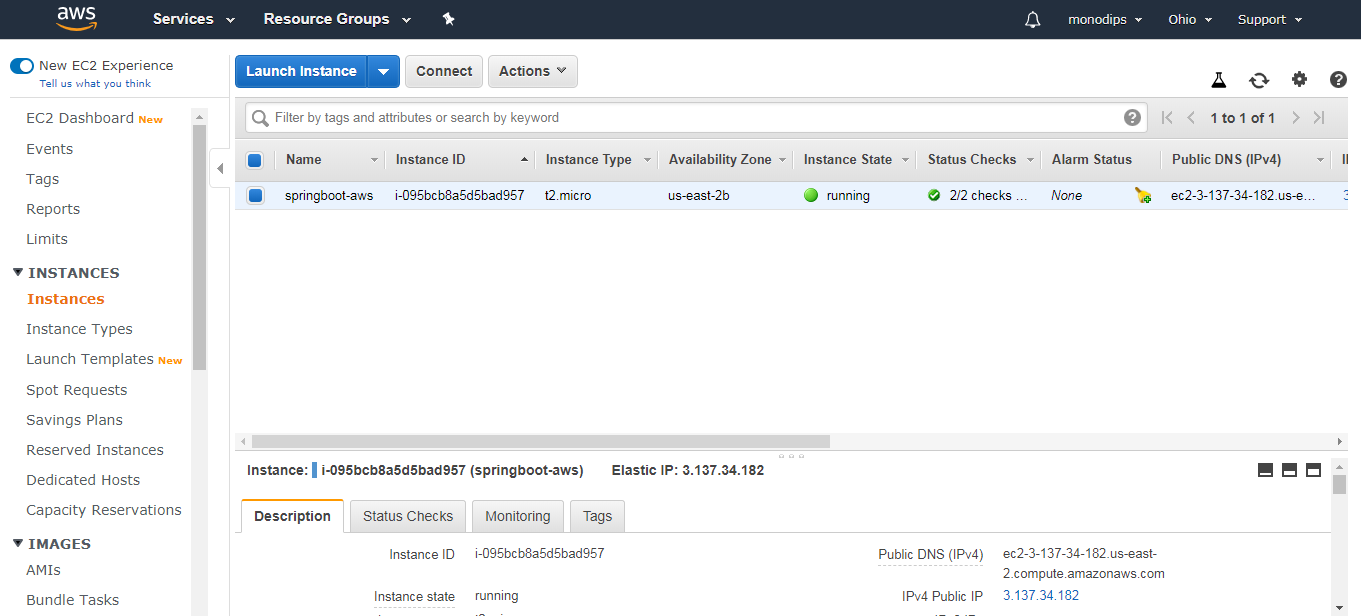






1. Click on Create environment.

After clicking on Create environment, in Backend one ec2 instance will be created and the application will be deployed on Tomcat.



**Code push to github from Local:**

1. Login to GitHub Website.

<https://github.com/login>

1. Create new Repository to Github
2. Now Open cmd:

Now go to the Terminal on your computer system. Use cd to navigate to the local project directory that you want to publish on GitHub.

CD to navigate to your directory

1. Initialize Local Directory

Now we will intialize our project. Use the below command to initialise the local directory as Git repository.

**Command**: git init

1. Add Local repository

Add all the files in the local directory to staging using the command below.

**Command**: git add .

This command stages all the files in the directory, ready for commit.

1. Commit Repository

You can now commit the staged files using the command below. It is explanatory and helpful.

Command: git commit –m “First Commit”

### Add Remote Repository url

Now, copy the remote repository URL provided by github to you when you published your repository on GitHub.

Now we will add the copied URL for your GitHub repository as remote repository using the code below.

**Command:** git remote add origin <https://github.com/gpartha123/aws-beanstalk-example.git>

This command add our GitHub repository as a remote that you can then push your local repository changes.

1. Push Local Repository to github.

In the last step, use the below command line in your terminal to push the local repository to GitHub. It will upload the file or project on github.

**Command:** git push origin master

[Note: For first commit before push, use below command:

git pull –rebase origin master]

**Adding Maven War plugin in POM.xml**

Details: <https://maven.apache.org/plugins/maven-war-plugin/>

POM Part:

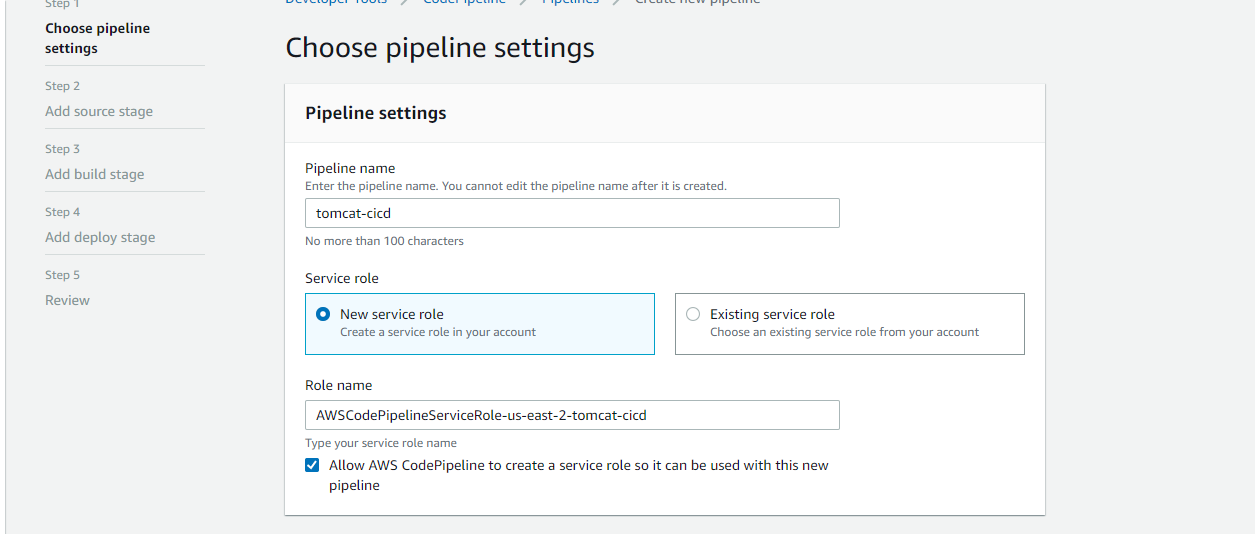


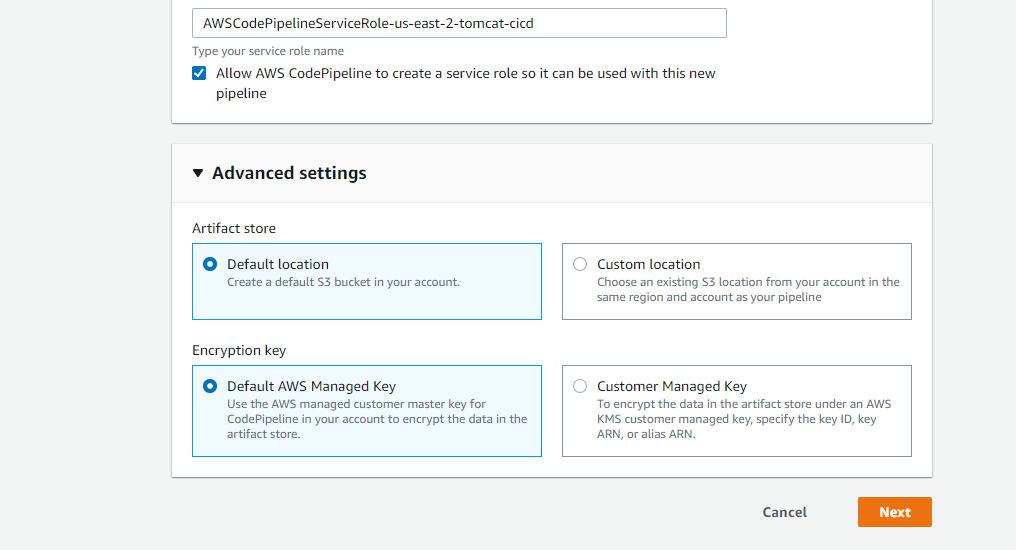
**Creating buildspec.yml:**

****

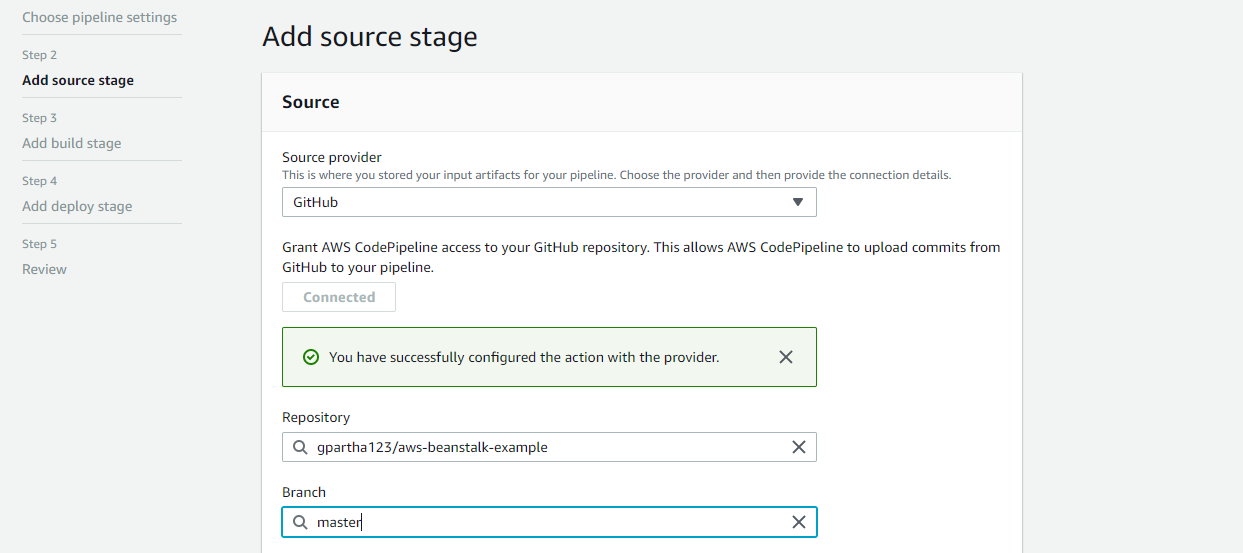
**AWS Code Pipe Line:**

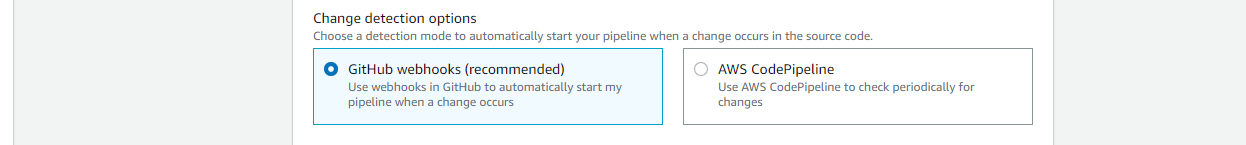
1. **Create Code Pipeline:**



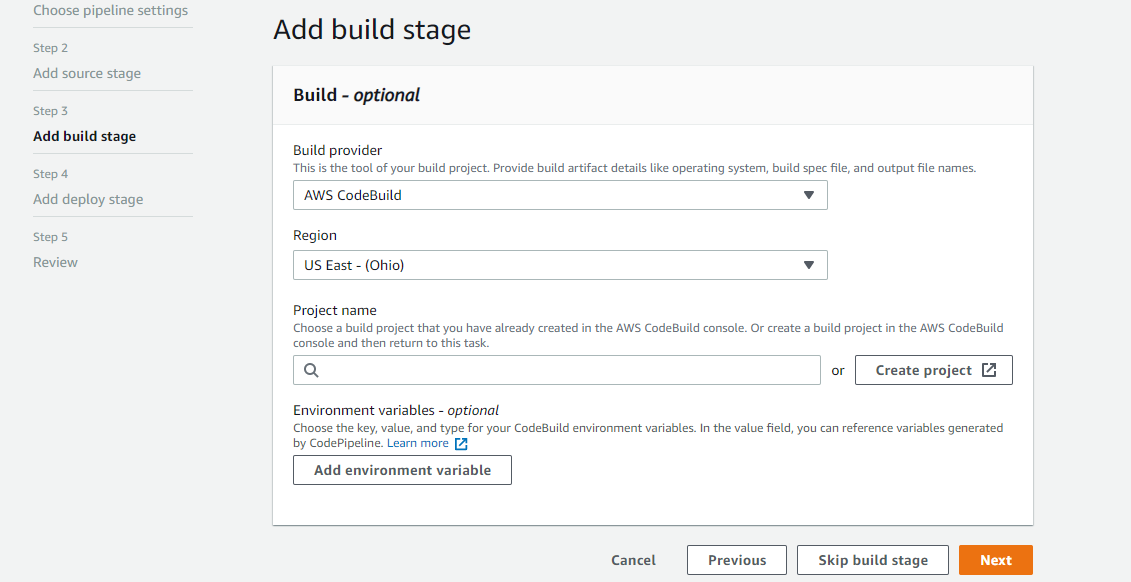


**Click on Next and configuring source stage:**

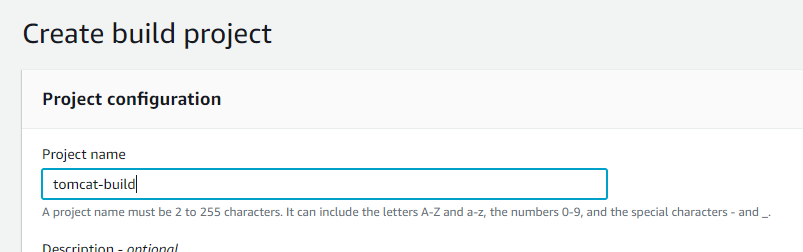




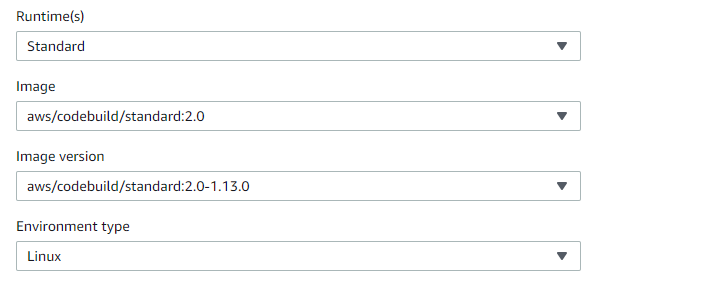
**Click on Next and configuring build stage:**



**Click on Create project to create build project:**



**Setup environment of container where build will happen:**



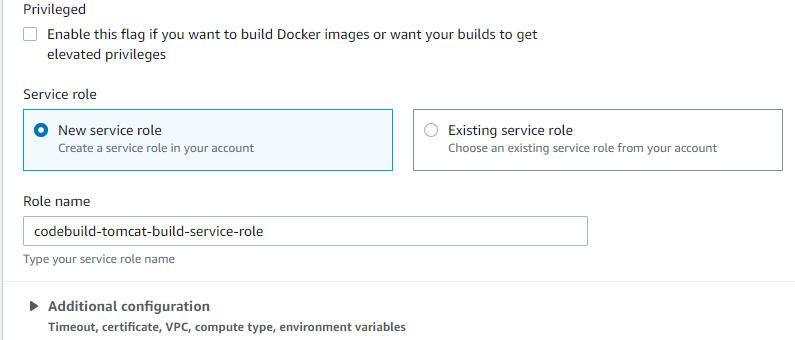
Add below runtime details in buildspec.yml:

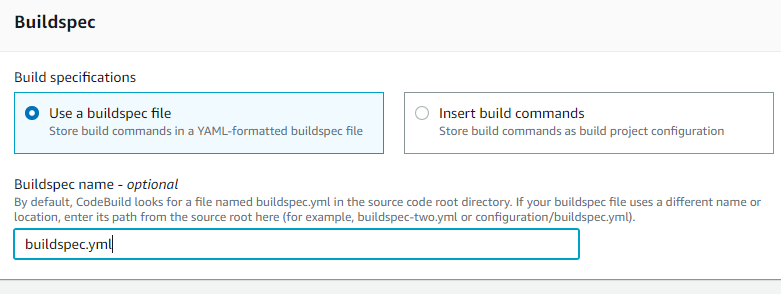
install:

runtime-versions:

java: openjdk8

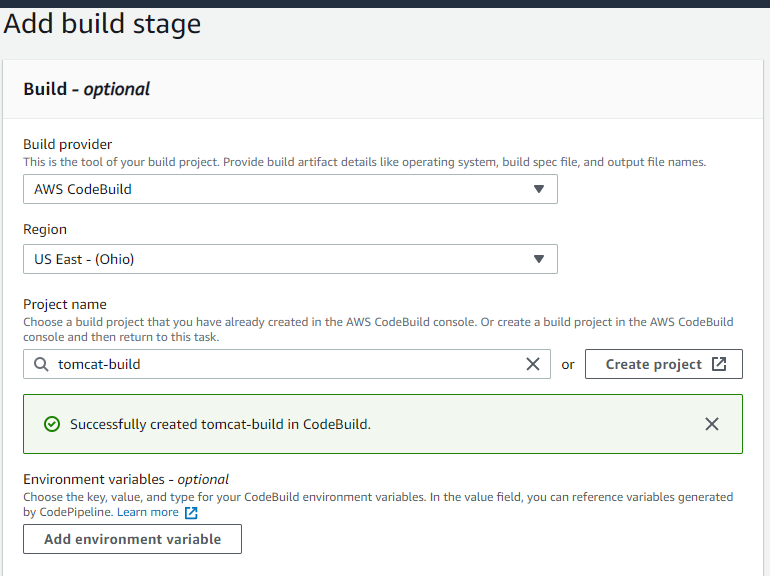
android: 29



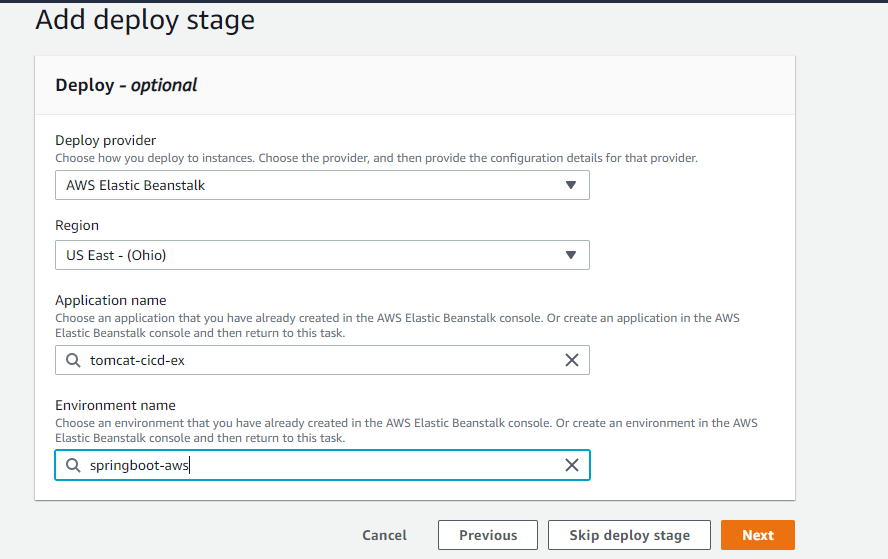


**Now Click on Continue to CodeCommit.’**

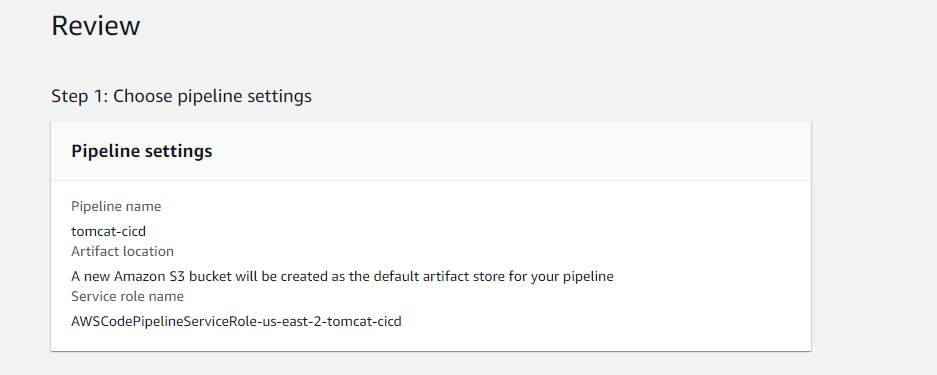
**After coming back to Add build stage:**

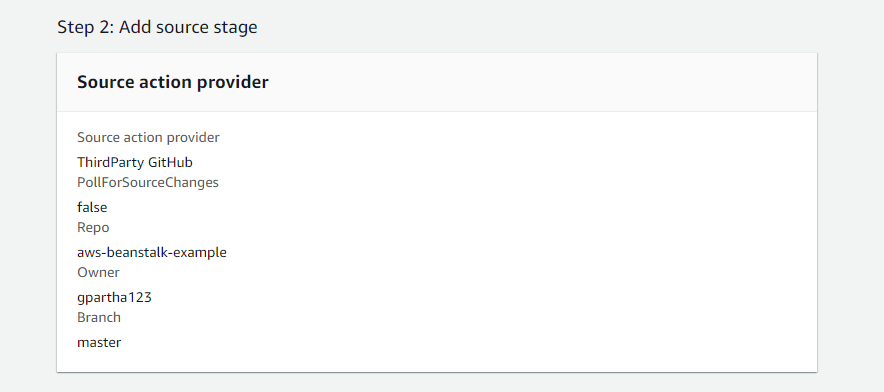


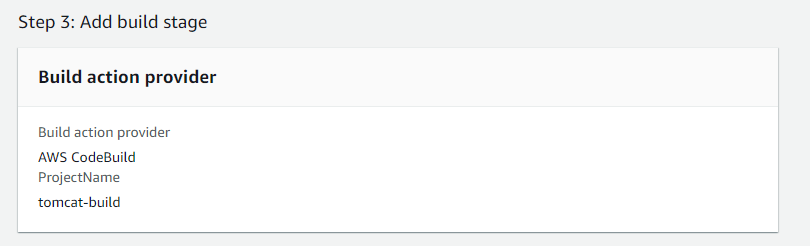
**Click on Next and configure deploy stage.**

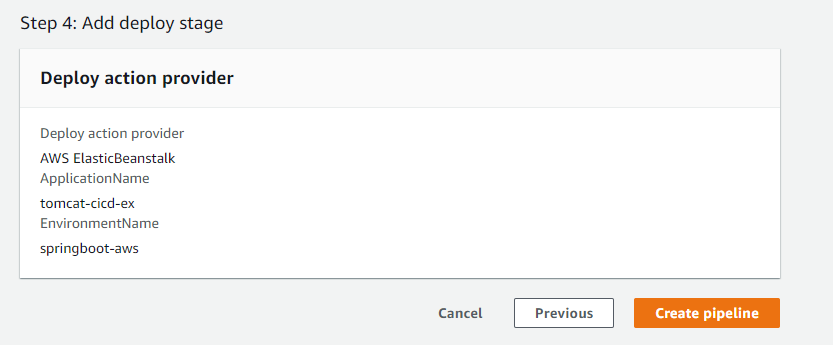


**Click on Next and check Review.**









**Click on Create pipeline.**

**Now time to push pom.xml change with maven war plugin and buildspec.yml in github:**

**Staging:**

**Command:** git add pom.xml

git add buildspec.yml

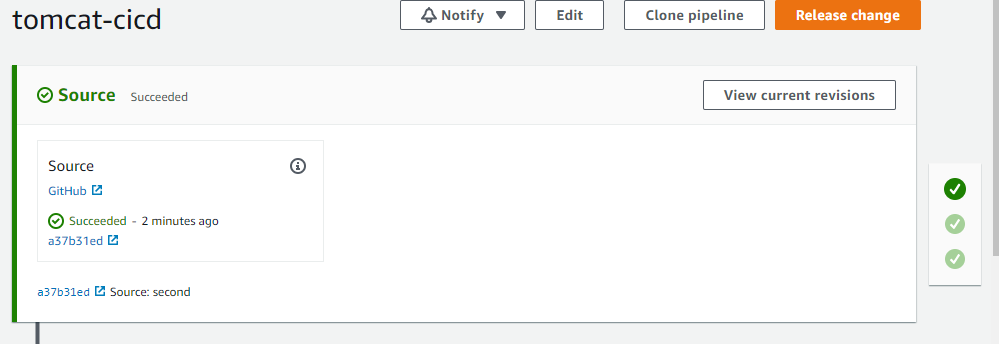
**Commit:**

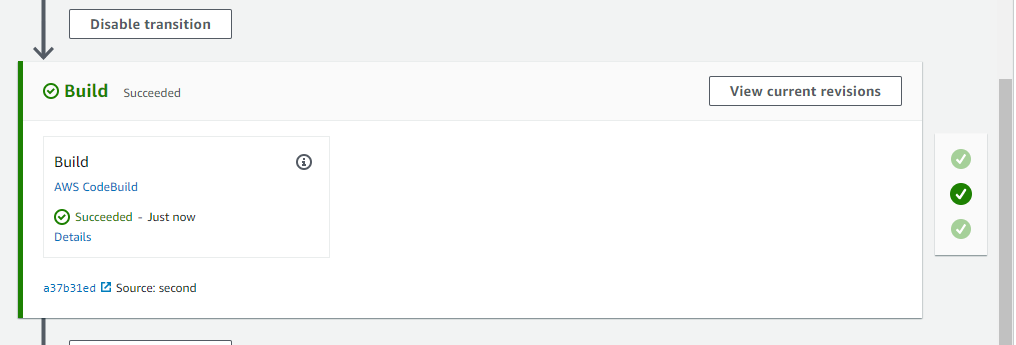
**Command:** git commit –m “second”

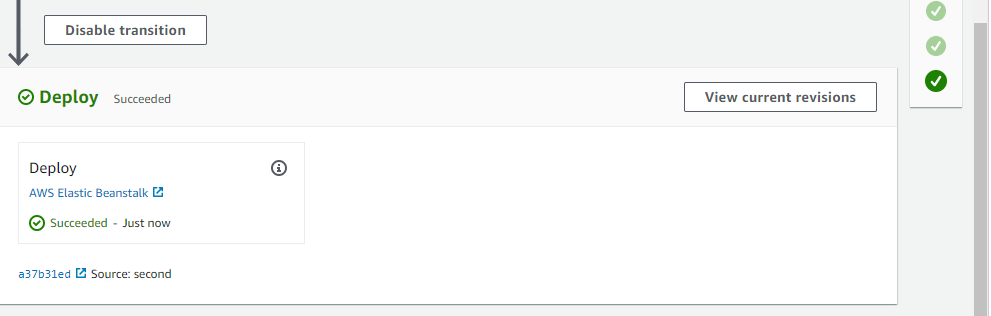
**Push**

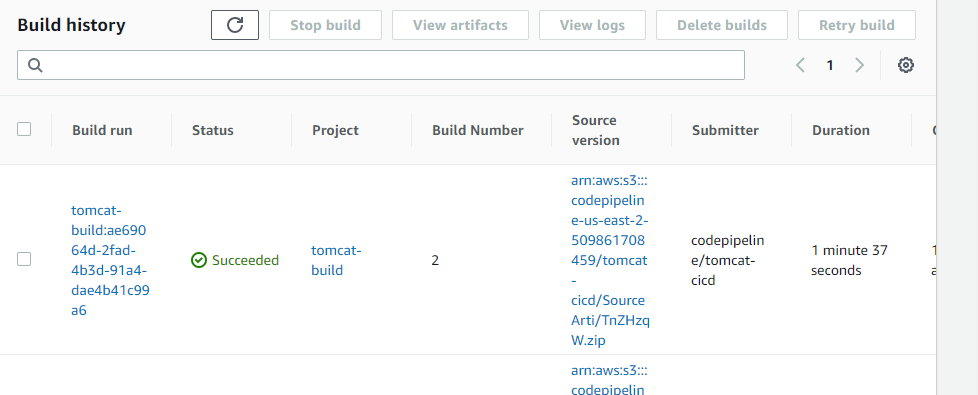
**Command:** git push origin master

After committing latest change, automatic code pipeline process will start including commit, build, deploy.

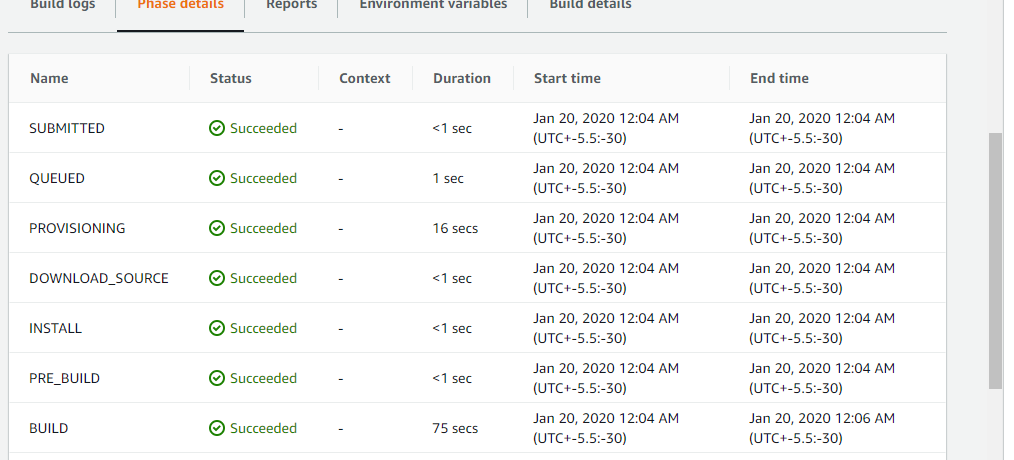


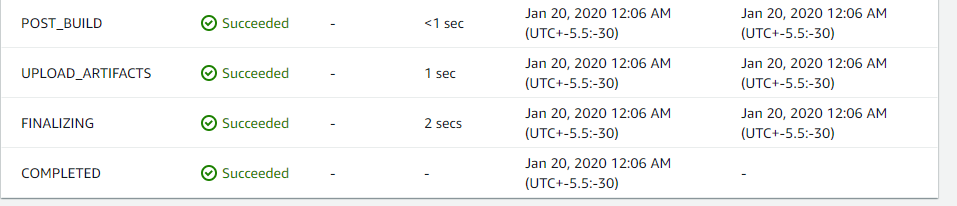




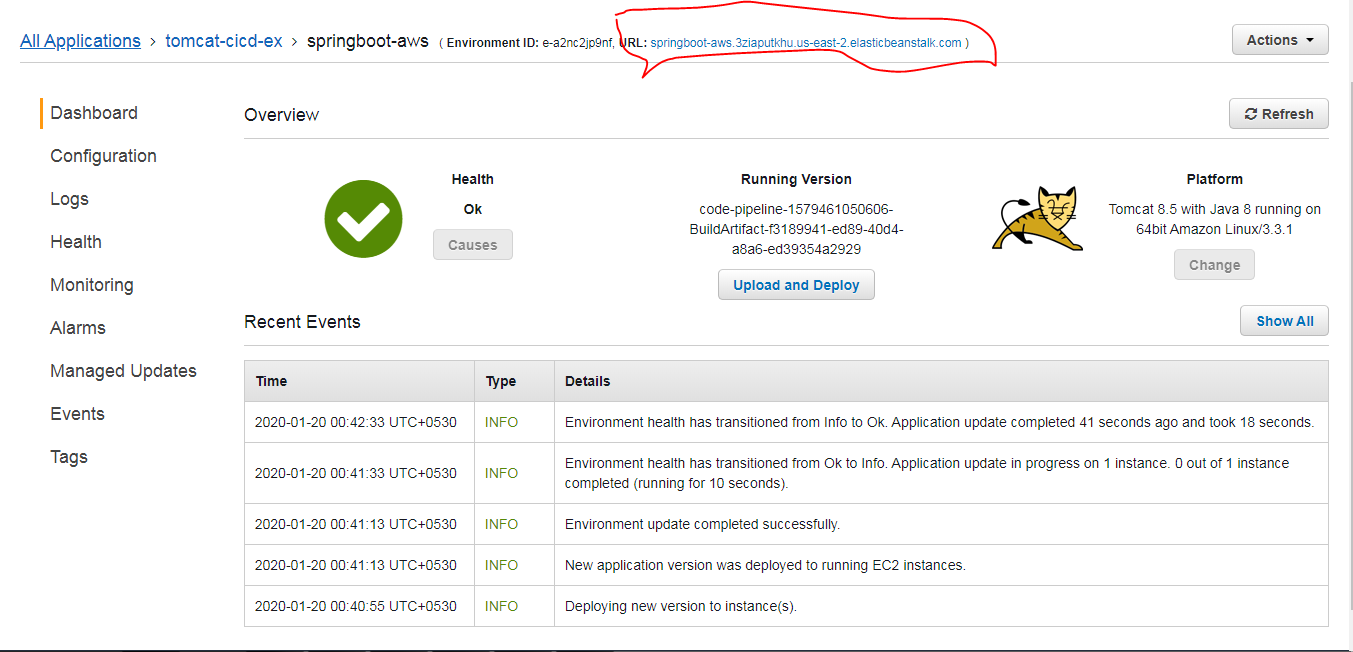


**PHASE DETAILS:**





**Finding the URL from Beanstalk:**



**In WebPage:**

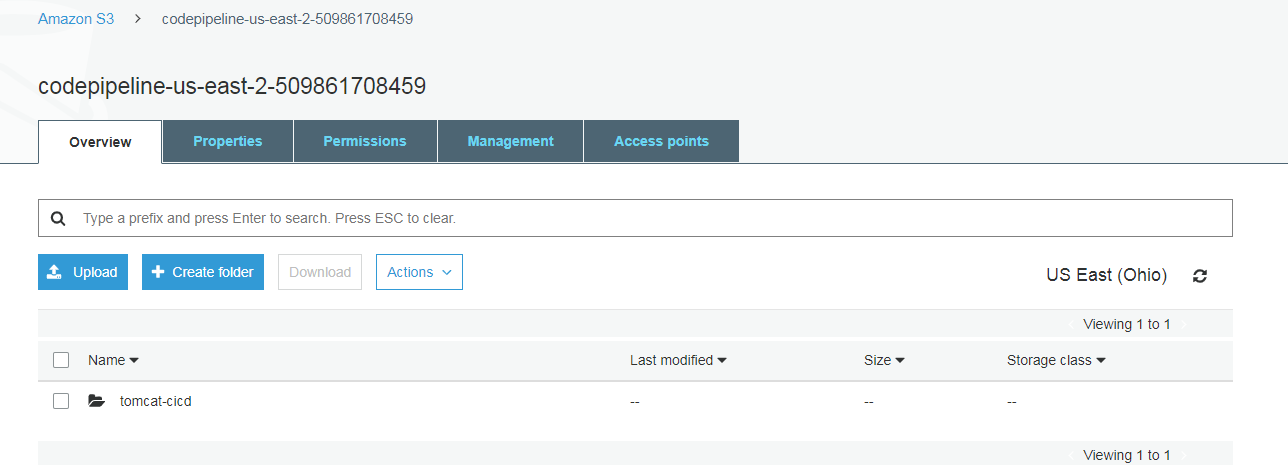
<http://springboot-aws.3ziaputkhu.us-east-2.elasticbeanstalk.com/actuator/health>

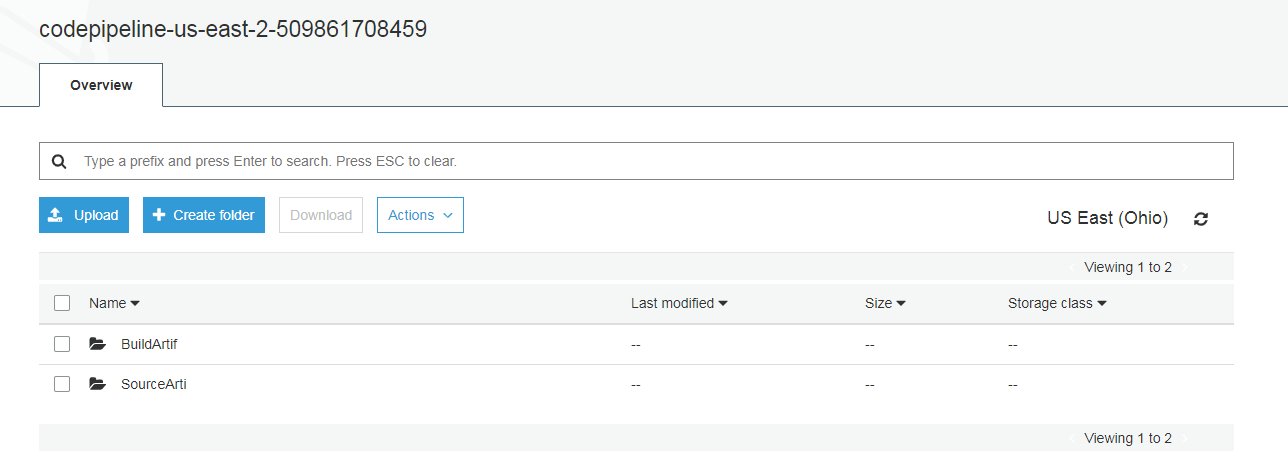
{"status":"UP"}

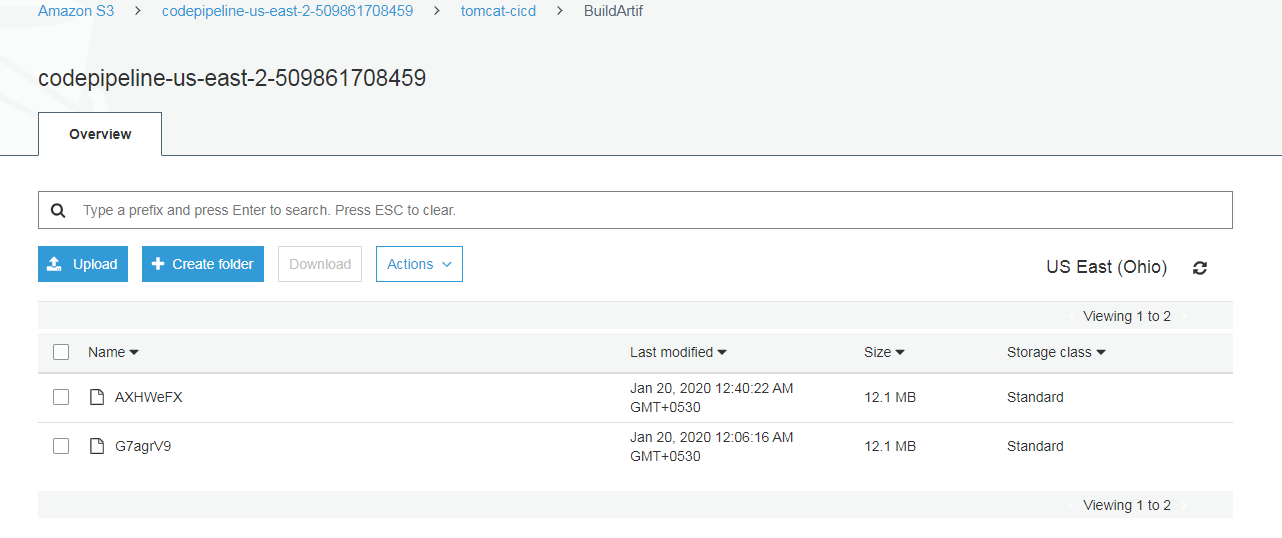
<http://springboot-aws.3ziaputkhu.us-east-2.elasticbeanstalk.com/hello>

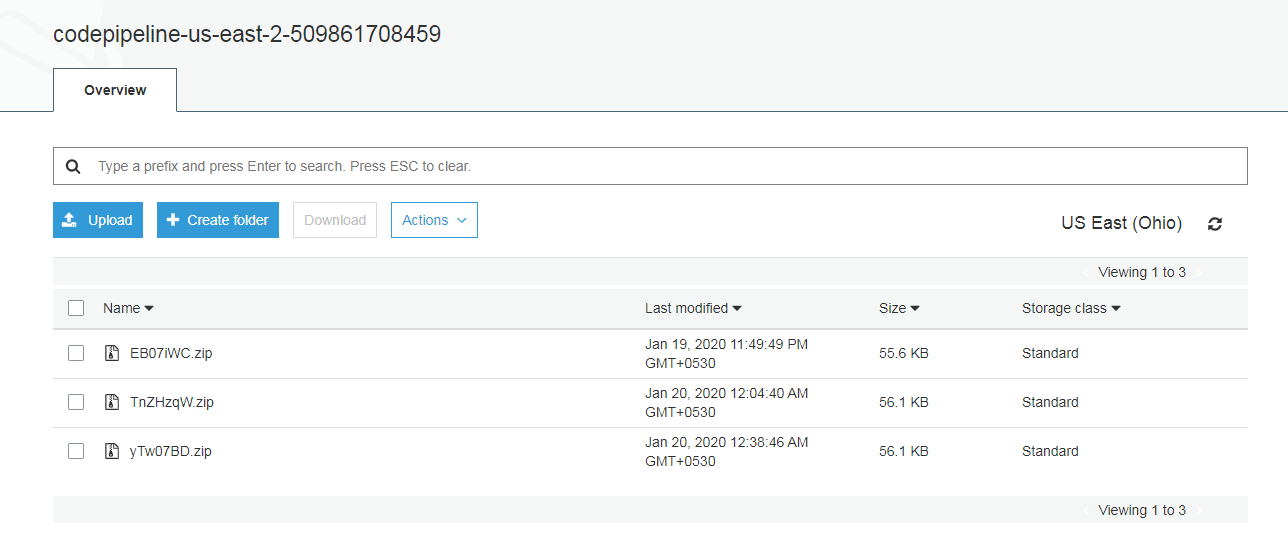
Hello Welcome

**Check AWS S3 to check the details because AWS S3 is being used by AWS to keep build details including output artifact and source code etc:**









**Checking codebuild details available in CloudWatch log:**

