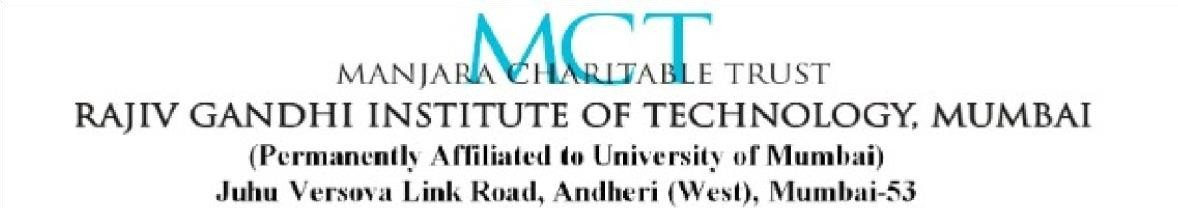
****

**Subject: Advance DevOps Lab(ADL) TE IT 563**

**Academic Year:-2022-23**

**PRACTICAL NO :- 02**

**Aim**

To build your application using AWS CodeBuild and Deploy on S3/ SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy.

**Theory**

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 (Amazon 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.

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.

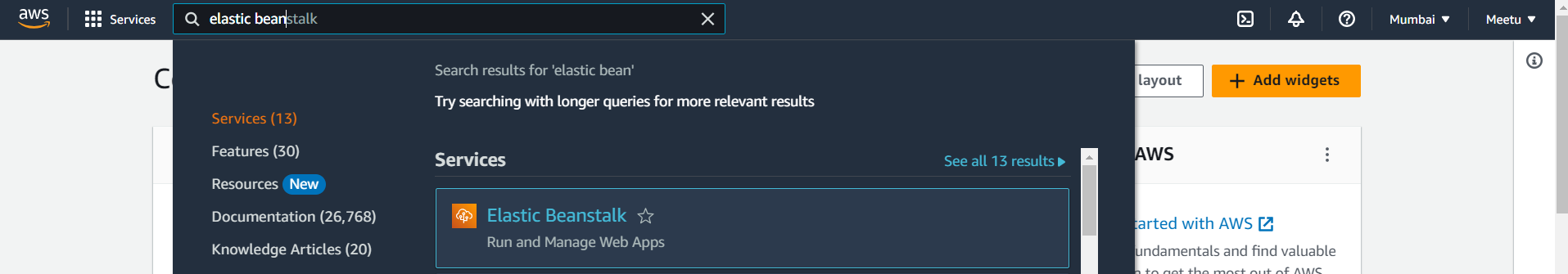
AWS ElasticBeankStalk:-

AWS Elastic Beanstalk is an orchestration service offered by Amazon Web Services for deploying applications which orchestrates various AWS services, including EC2, S3, Simple Notification Service, CloudWatch, autoscaling, and Elastic Load Balancers.

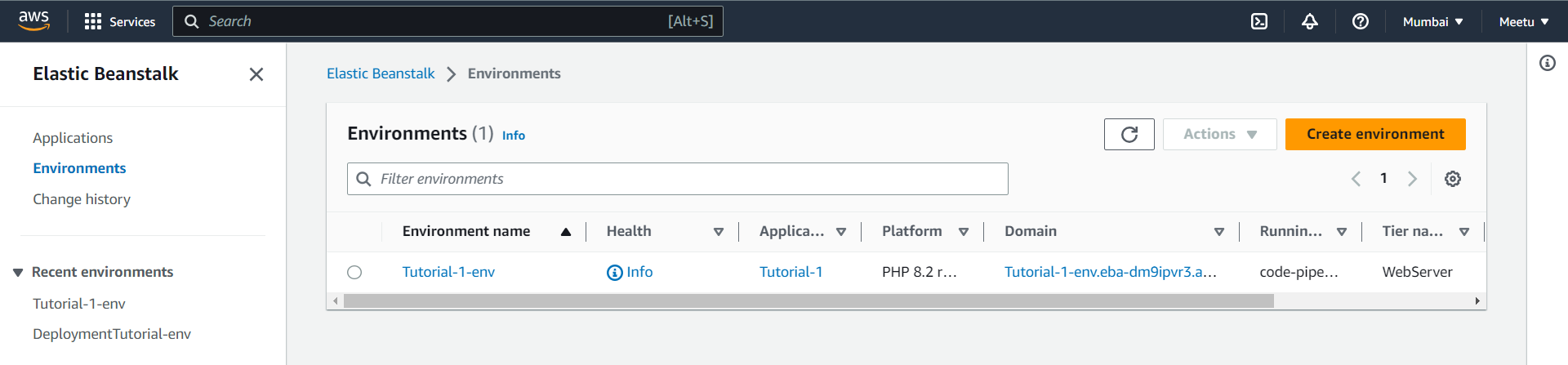
Elastic Beanstalk is a service for deploying and scaling web applications and services. Upload your code and Elastic Beanstalk automatically handles the deployment—from capacity provisioning, load balancing, and auto scaling to application health monitoring.

Now we will see how to build your application using AWS CodeBuild and Deploy on S3

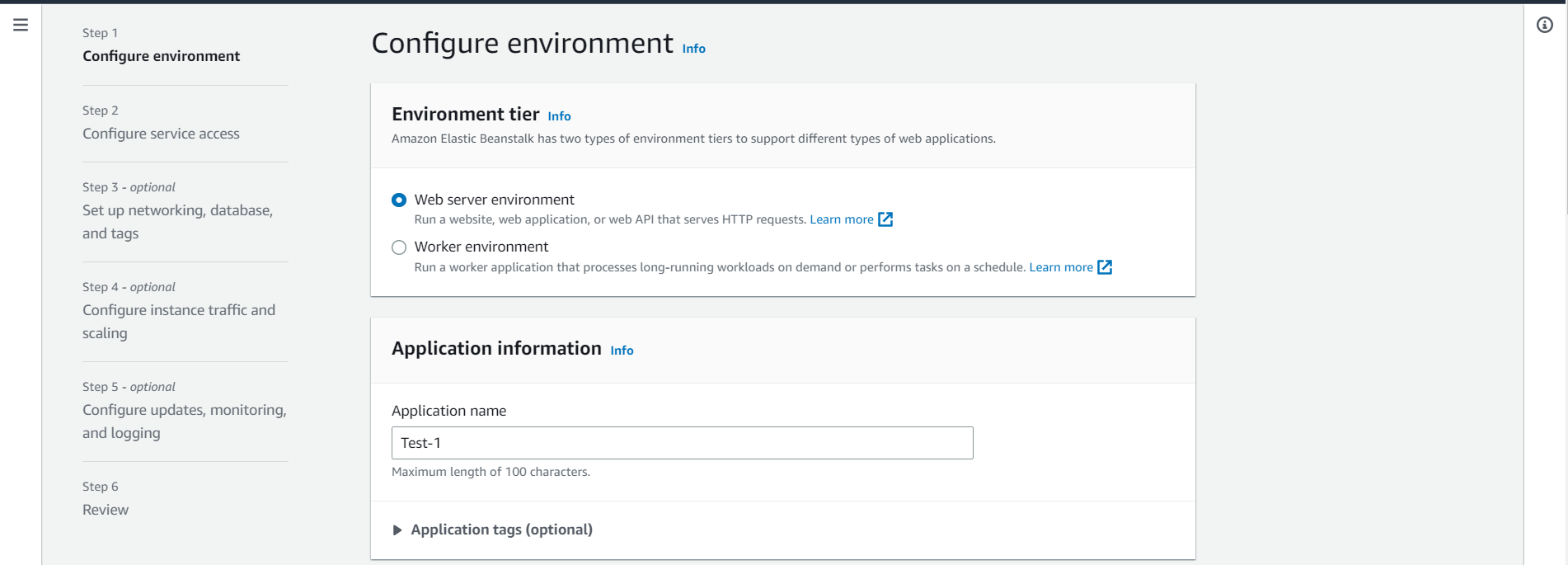
**Step 1** Go to your aws account and search for elastic bean stalk in the search bar

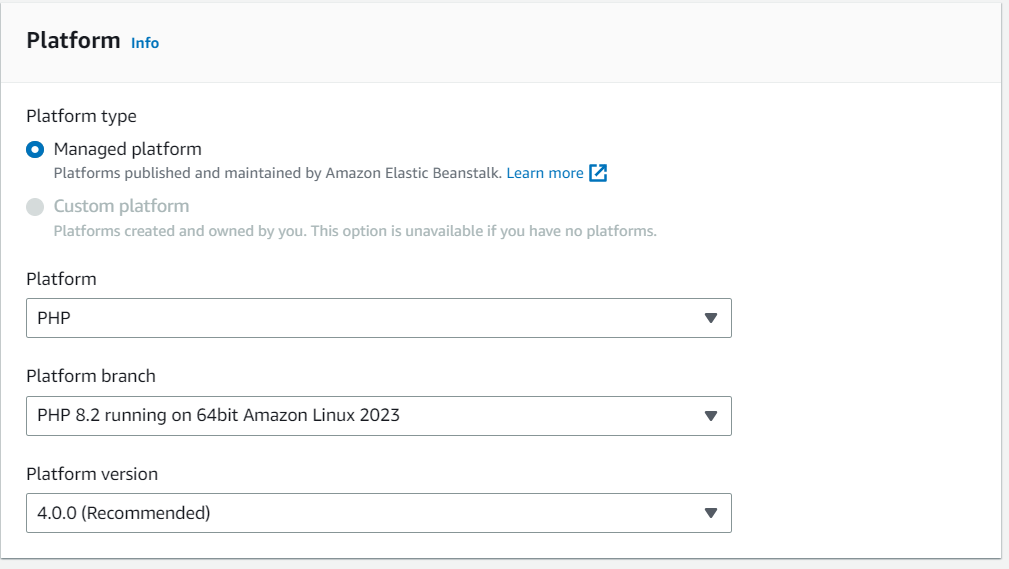


**Step 2** Select Elastic beanstalk and click on create environment

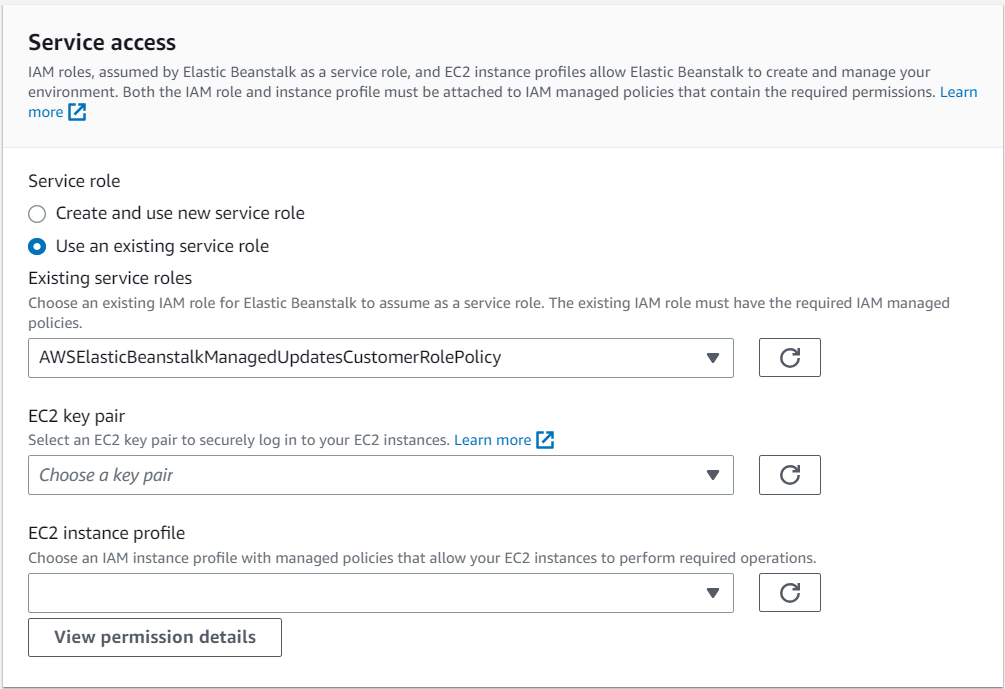


Fill in the details for the environment as follow and rest will be default





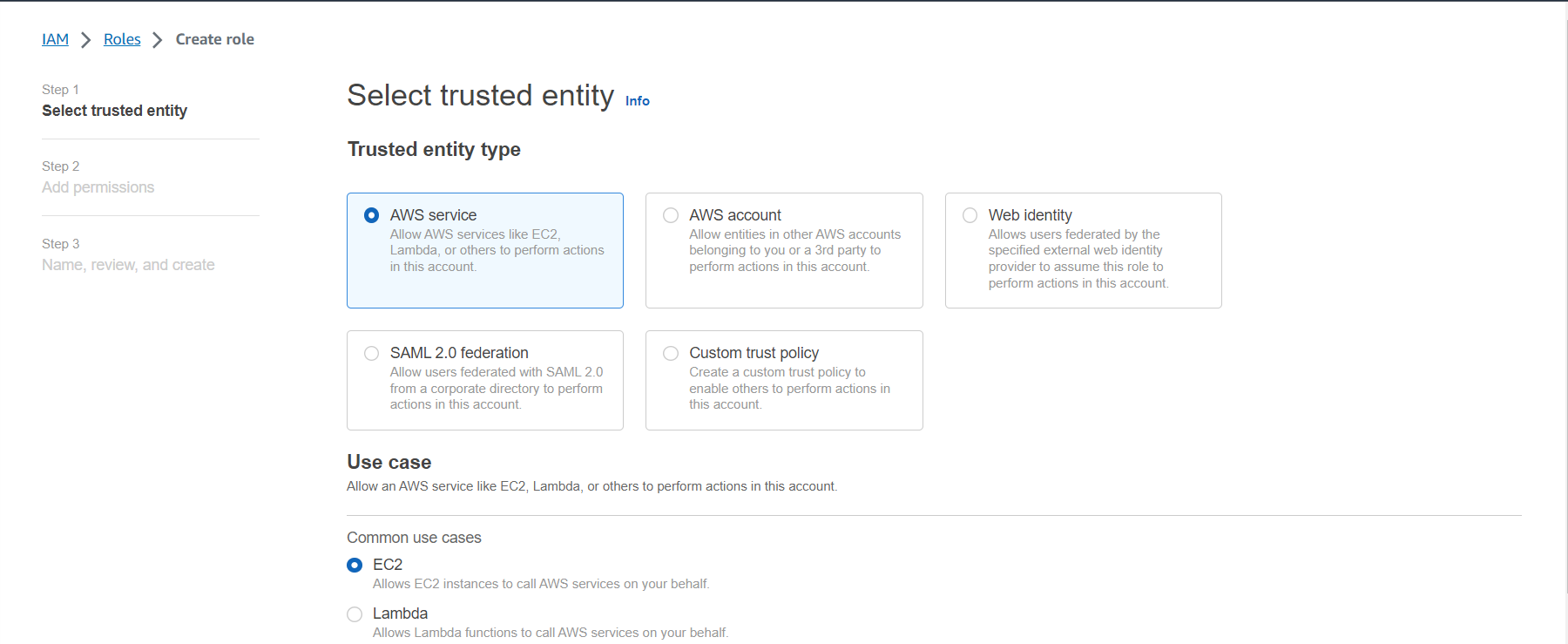
**Step 3** Now click next and in the Service Access we will select a service role



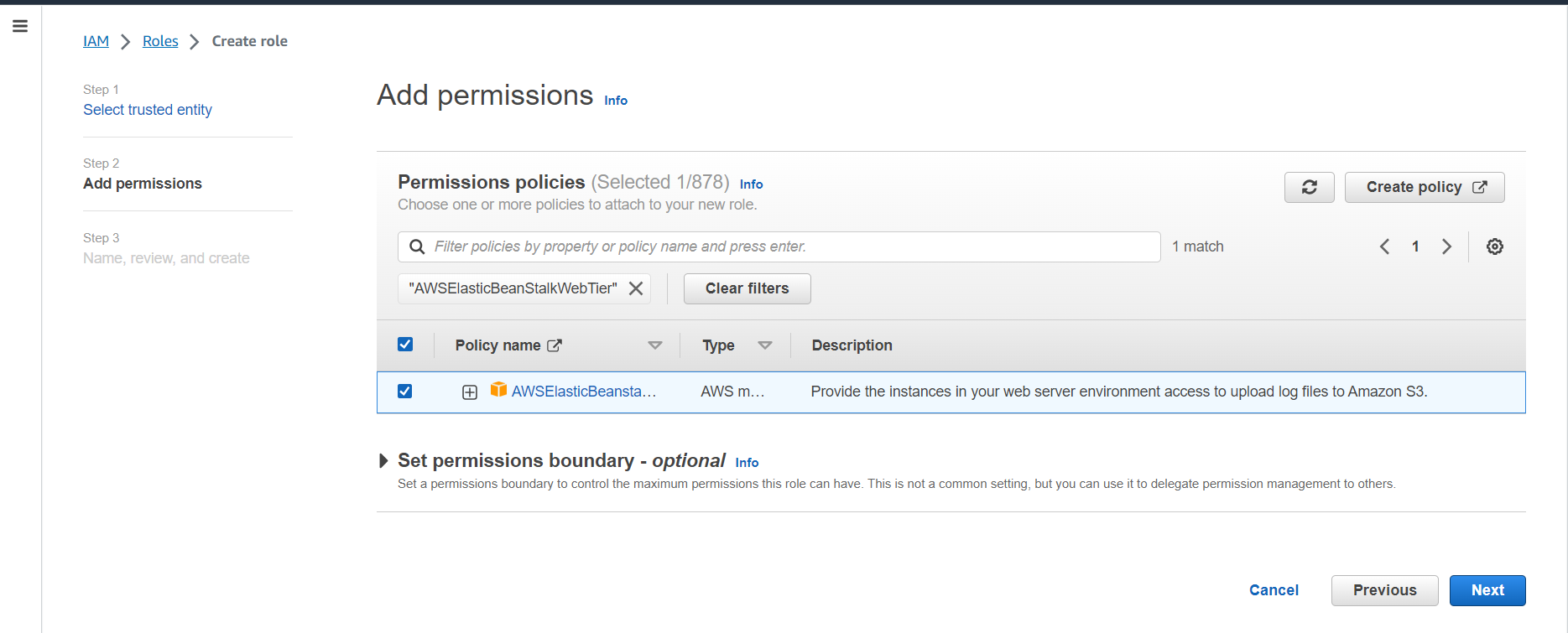
Now we will also need a instance profile associated with it , So to create it open IAM in new tab.

**Step 4** Open IAM and click on roles -> Create roles

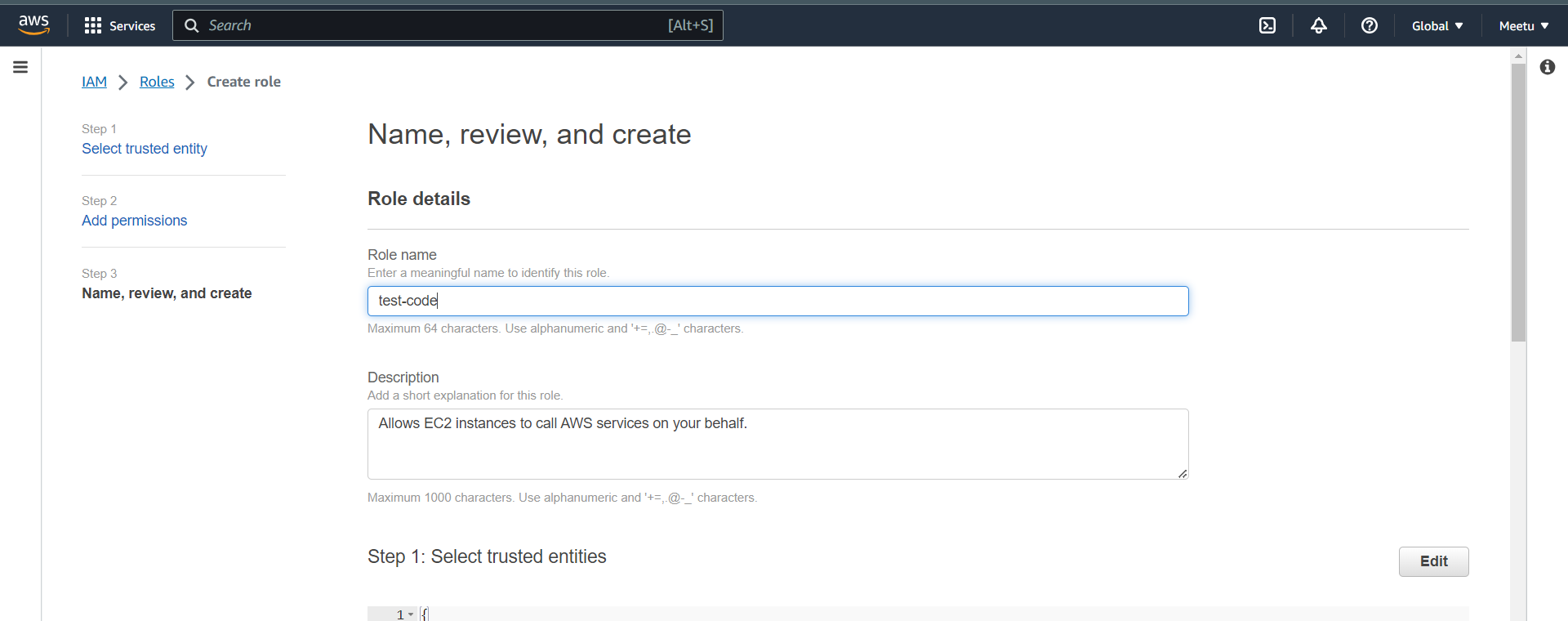
Select the following option and click next to add permissions



Select the following permission policy

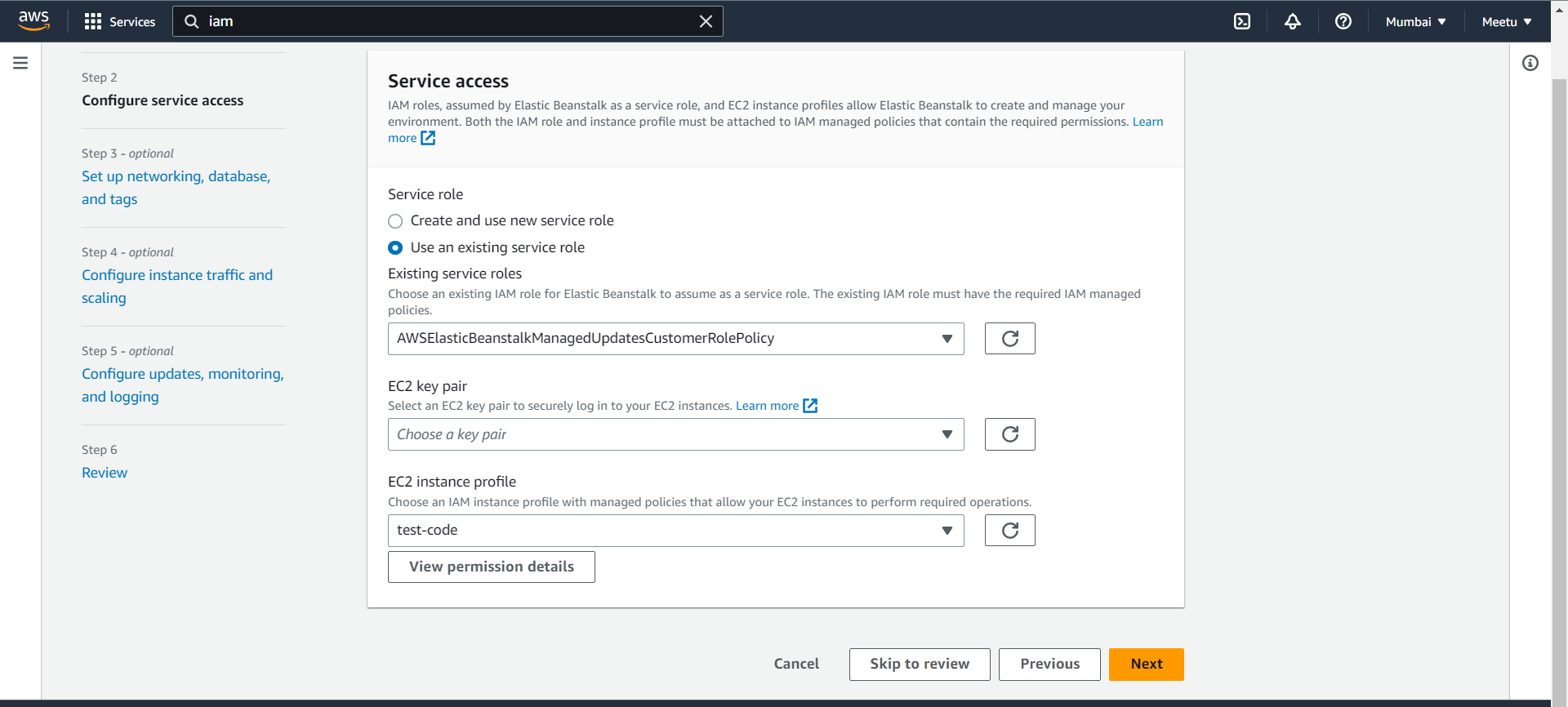


Click next and go to Name,Review and Create

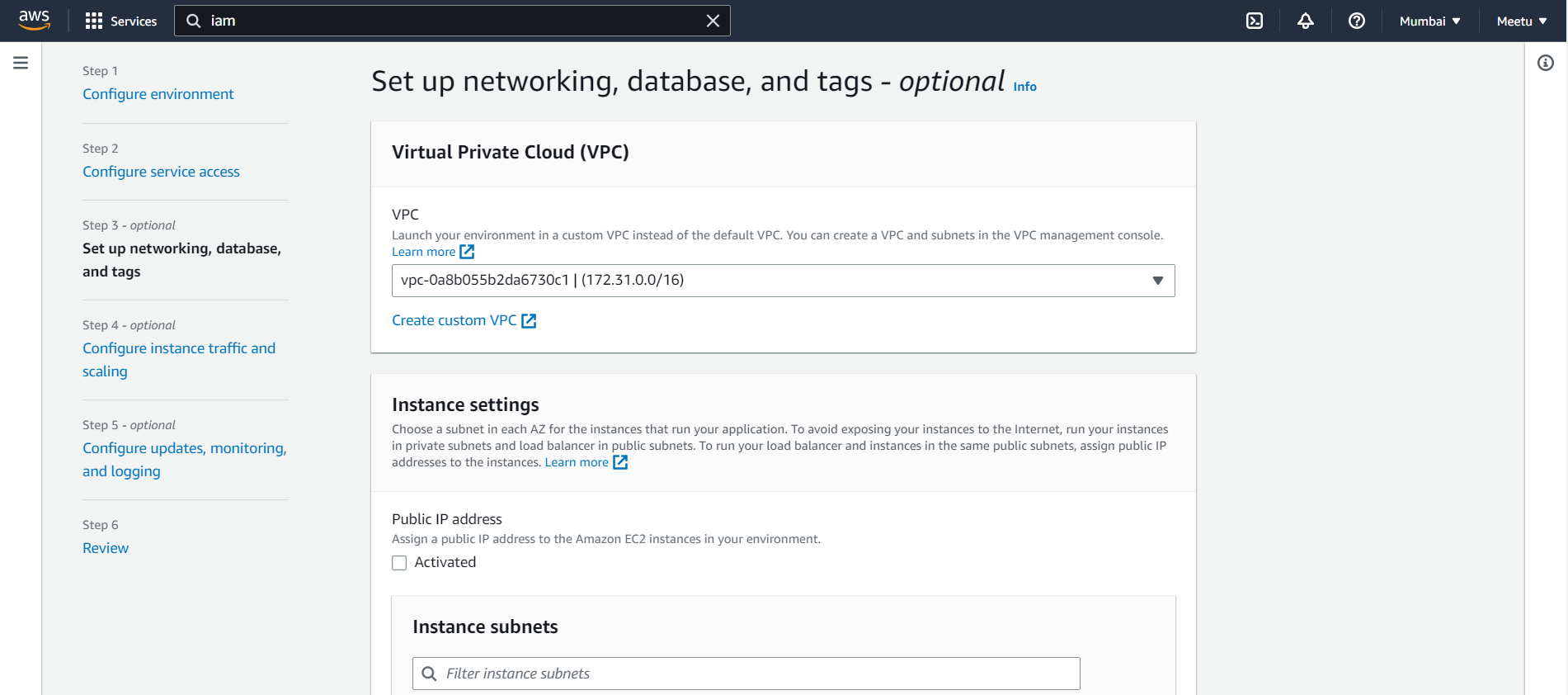


And click on create role

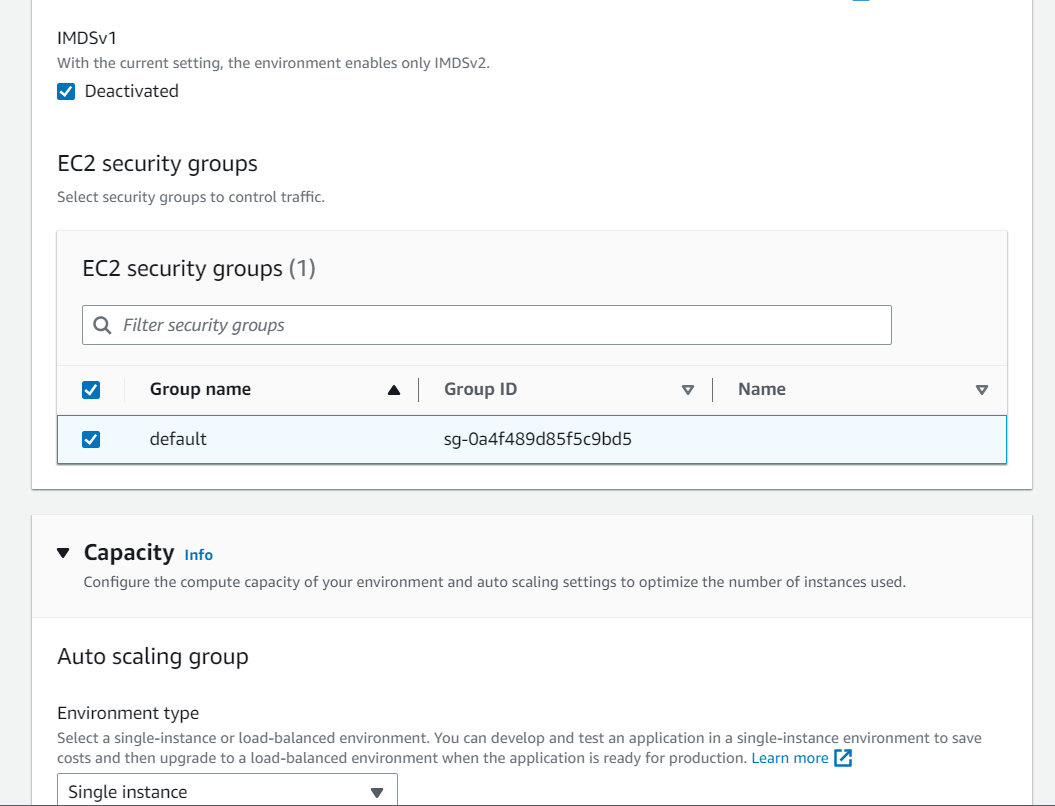
Now go back to elastic bean stalk and select the name of the instance profile you just created



**Step 5** Set up networking, database, and tags

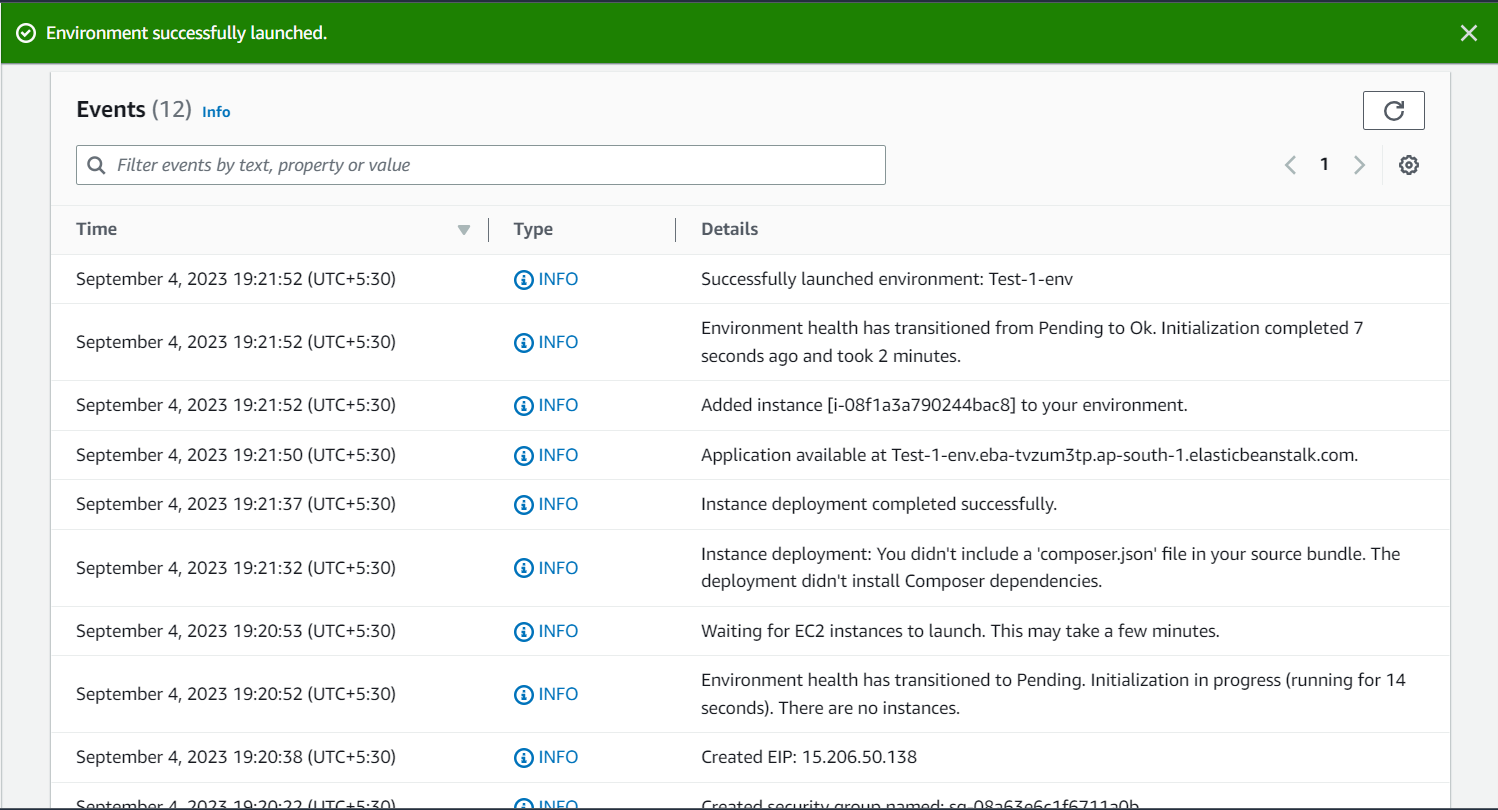


**Step 6** Configure instance traffic and scaling(rest settings are default)



**Step 7** Rest everything is default and click on create a environment

**Step 8** It will take a few time to create the environment, Once all the events are executed without error the environment is successfully launched

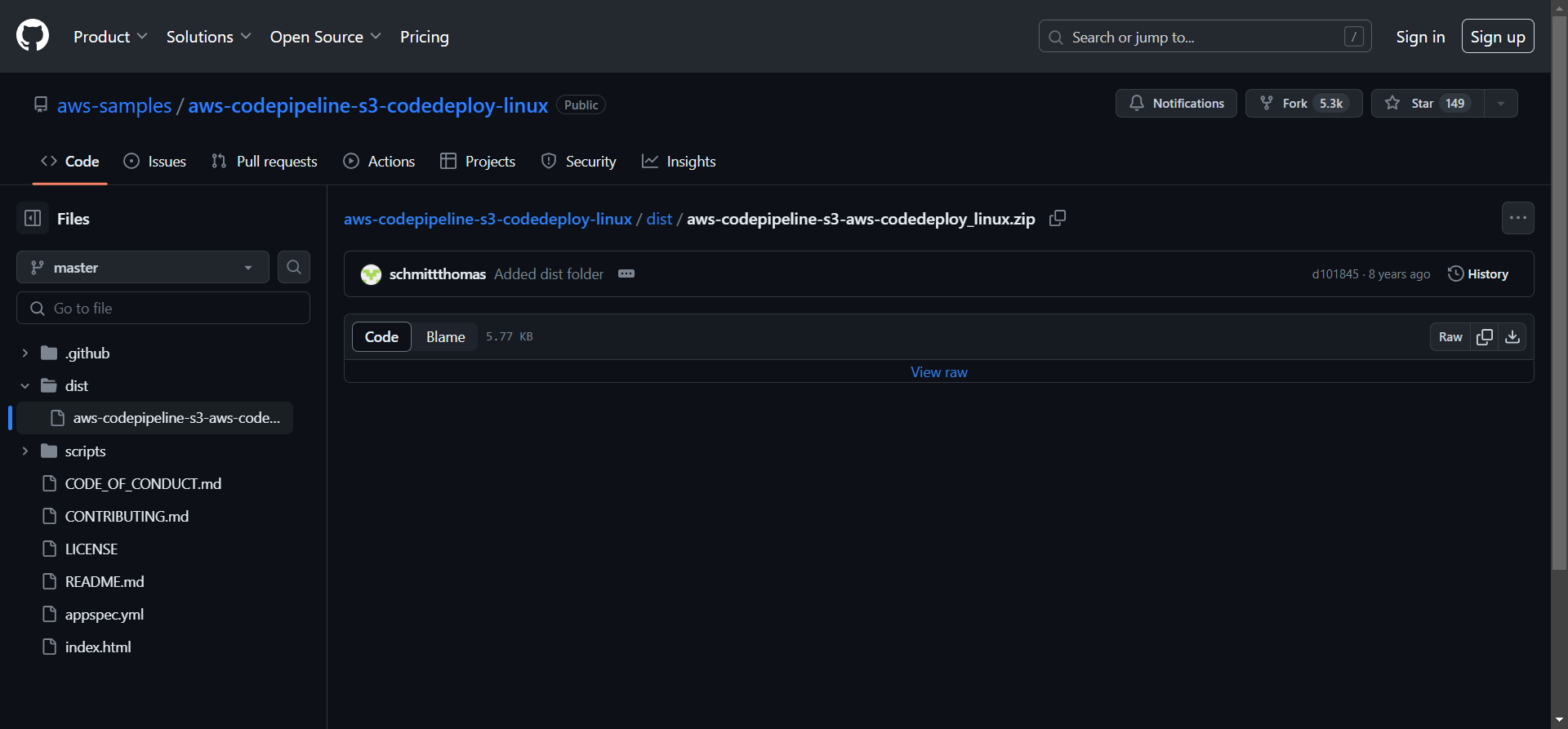




**Step 9** Now download a sample code file

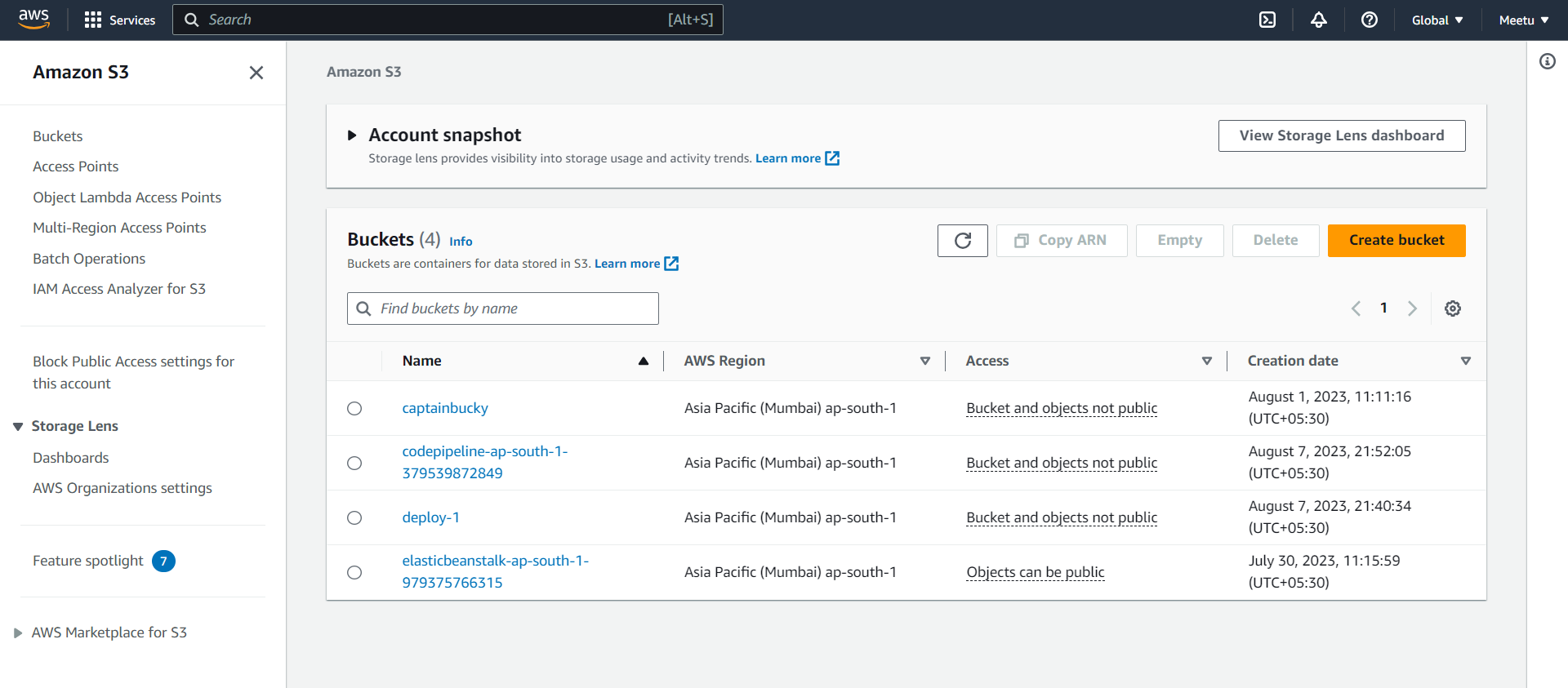
From this link :- <https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux>

Go to dist folder -> Click on the link -> and click on view raw



Now the file will be downloaded

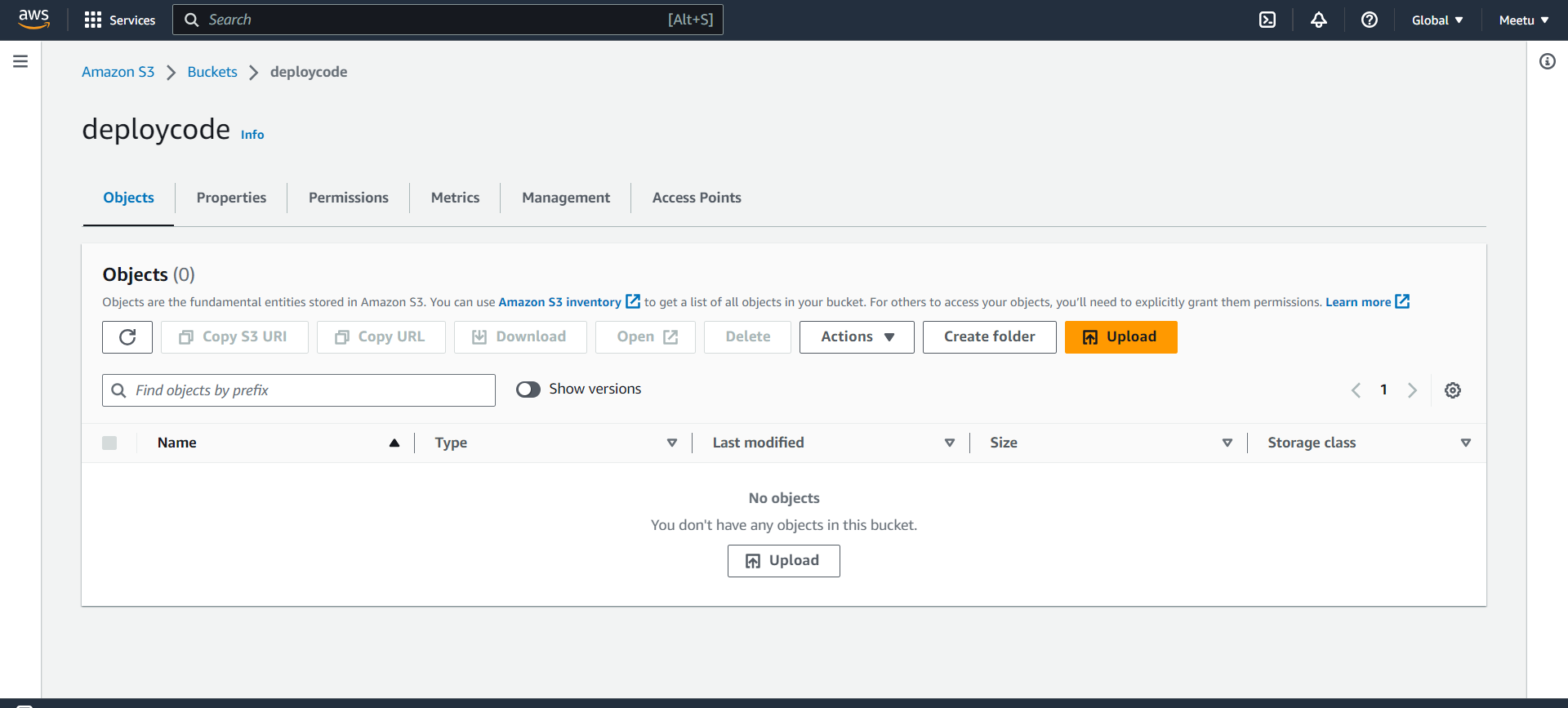
**Step 10** Go to Amazon S3 to create a bucket to upload your file ***Click*** on create a bucket



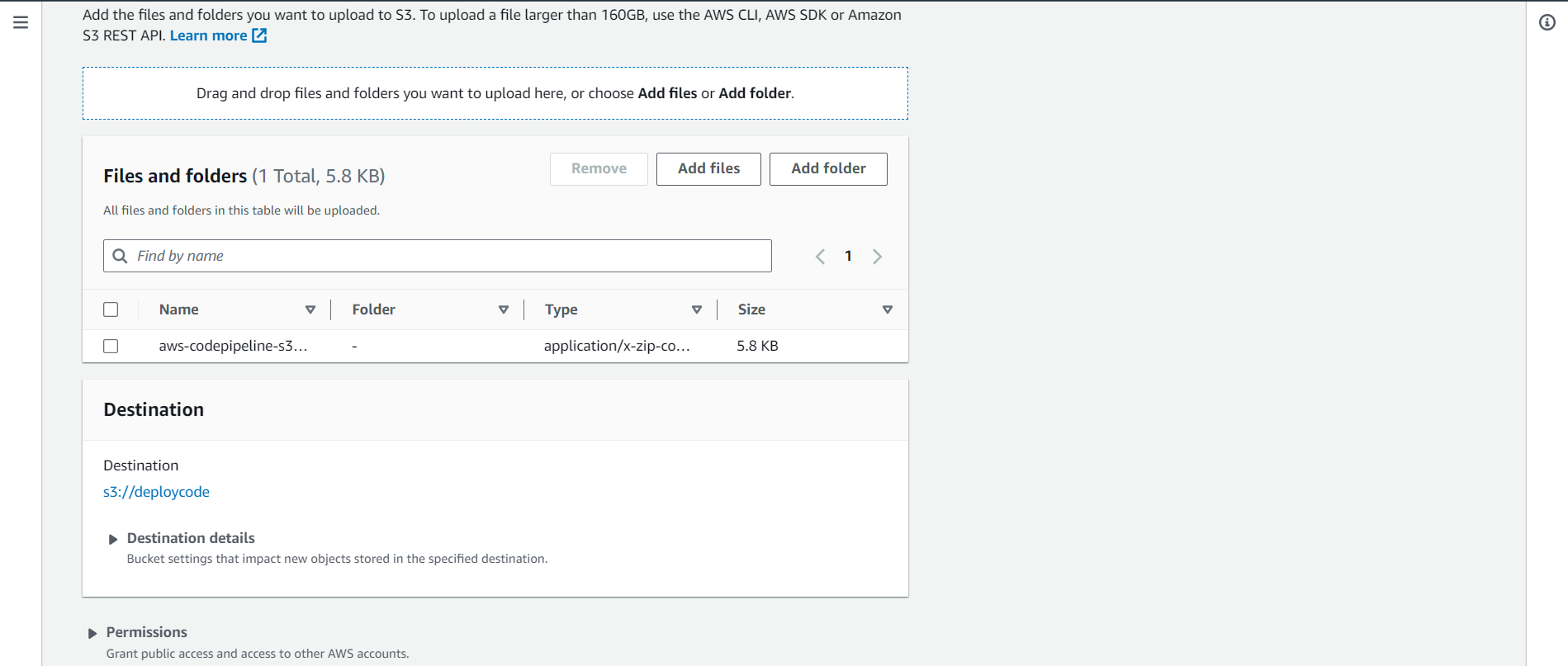
Now give name to your bucket and enable versioning and click on create



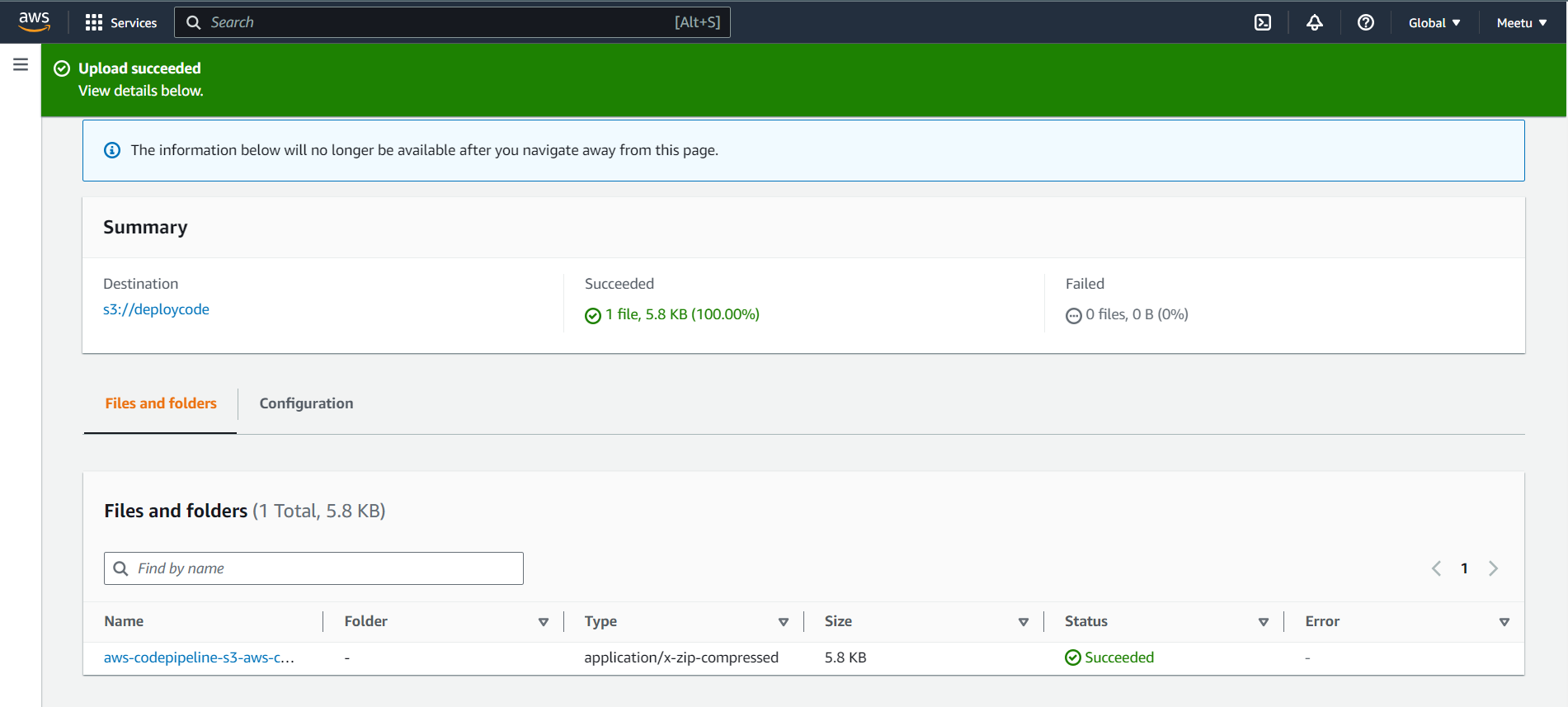
Then click on the name of your bucket and click on upload



Now add your file and click on upload

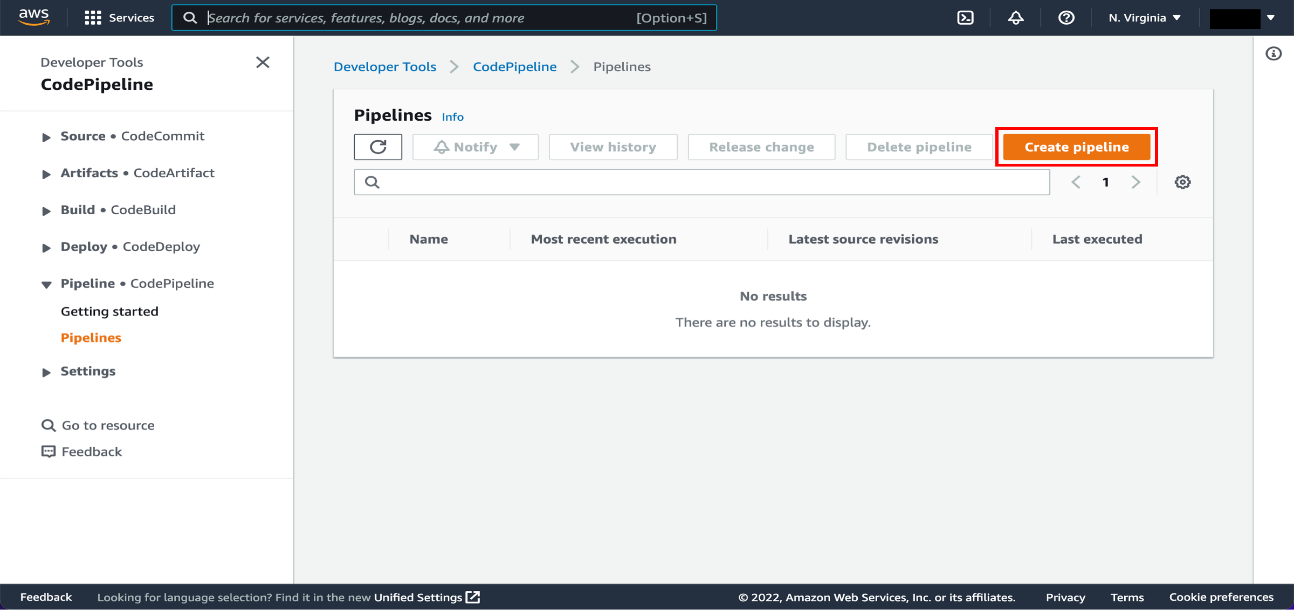


Hence it is successfully uploaded



**Step 11** Open the [AWS CodePipeline console.](http://console.aws.amazon.com/codepipeline)

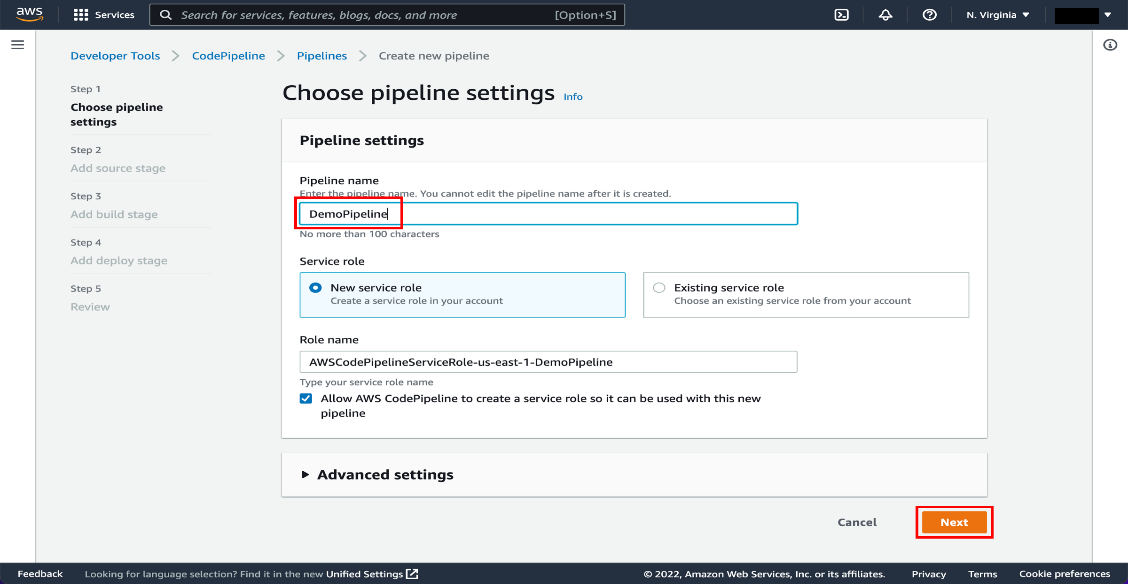
* On the Welcome page, choose Create pipeline.



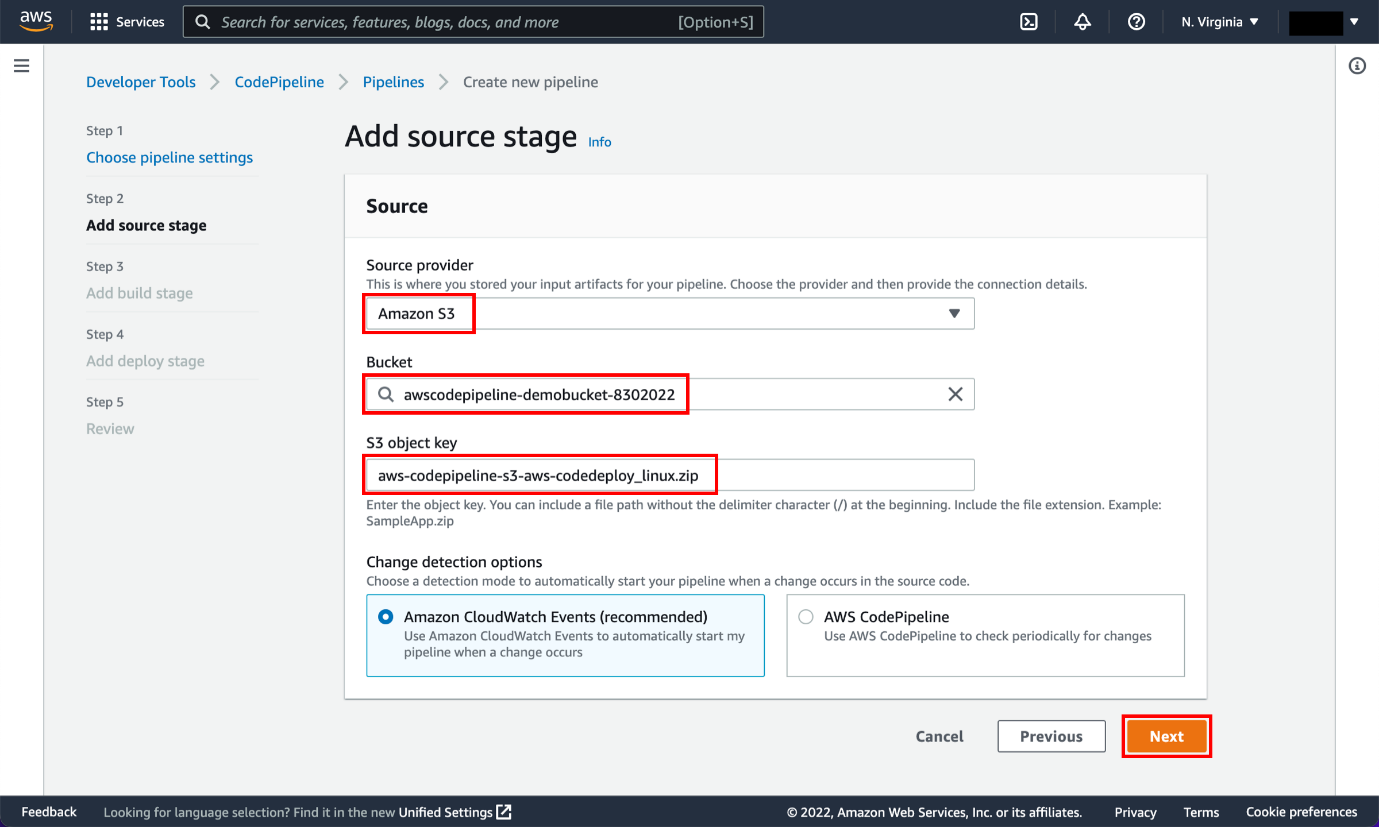
On the Step 1: Choose pipeline settings page:

* Pipeline name: Enter the name for your pipeline, *Demo Pipeline*.
* Choose Next.

Note: After you create a pipeline, you cannot change its name.

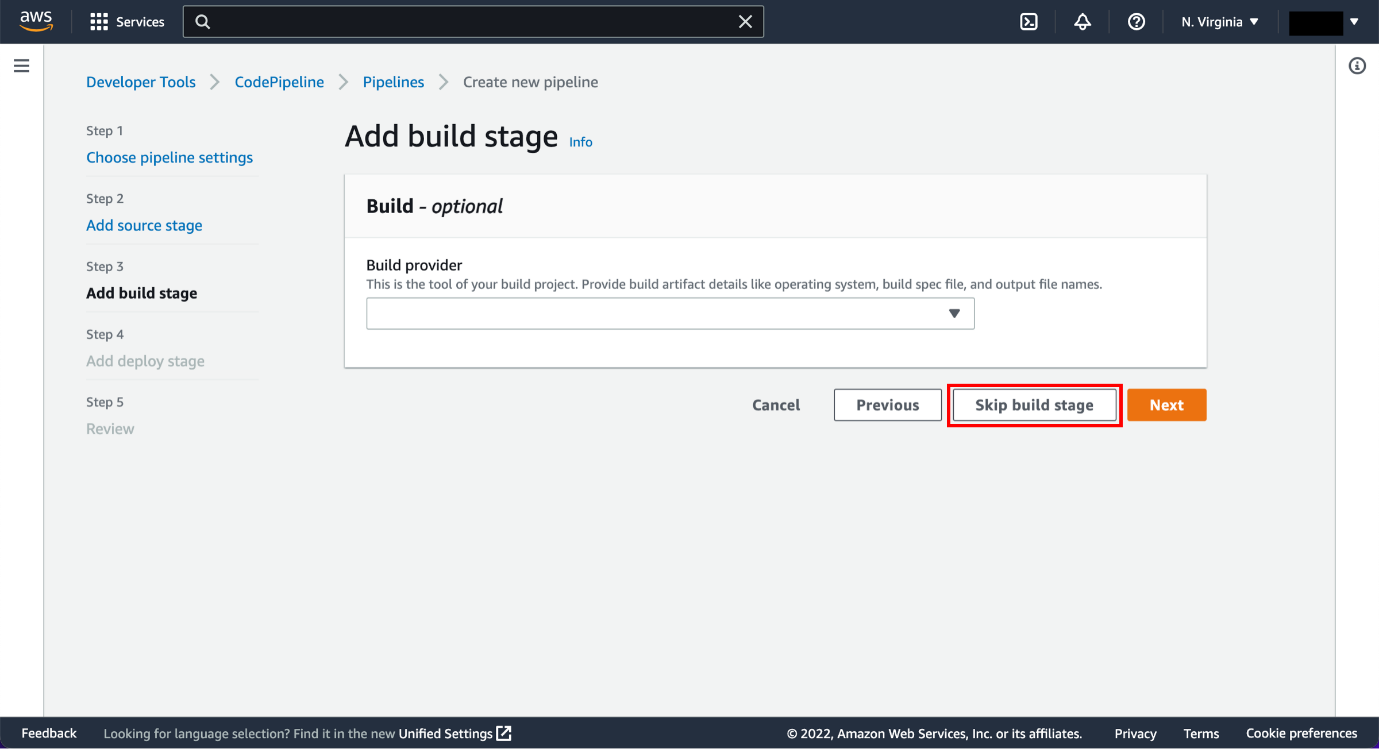


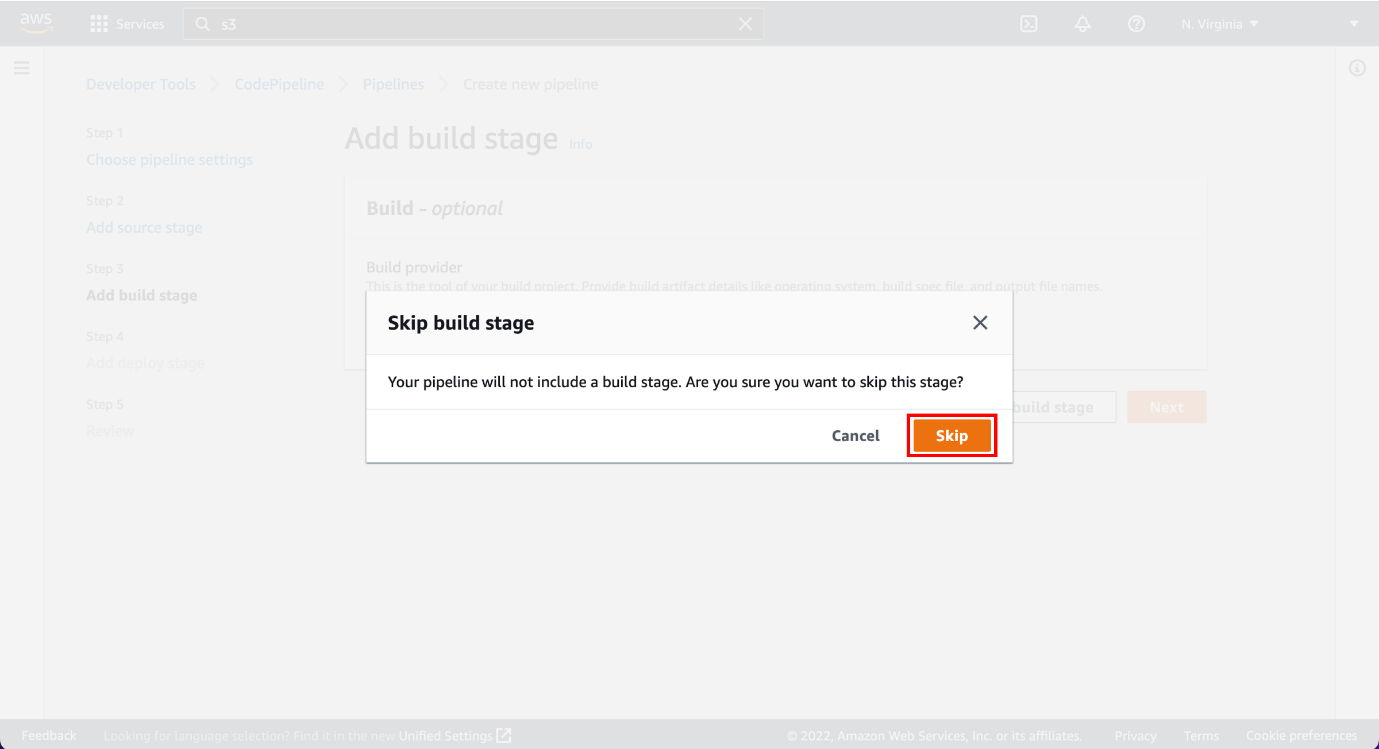
On the Step 2: Add source stage page, select the location of the source you selected and follow the steps below



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

* In Step 3: Add build stage, choose Skip build stage.



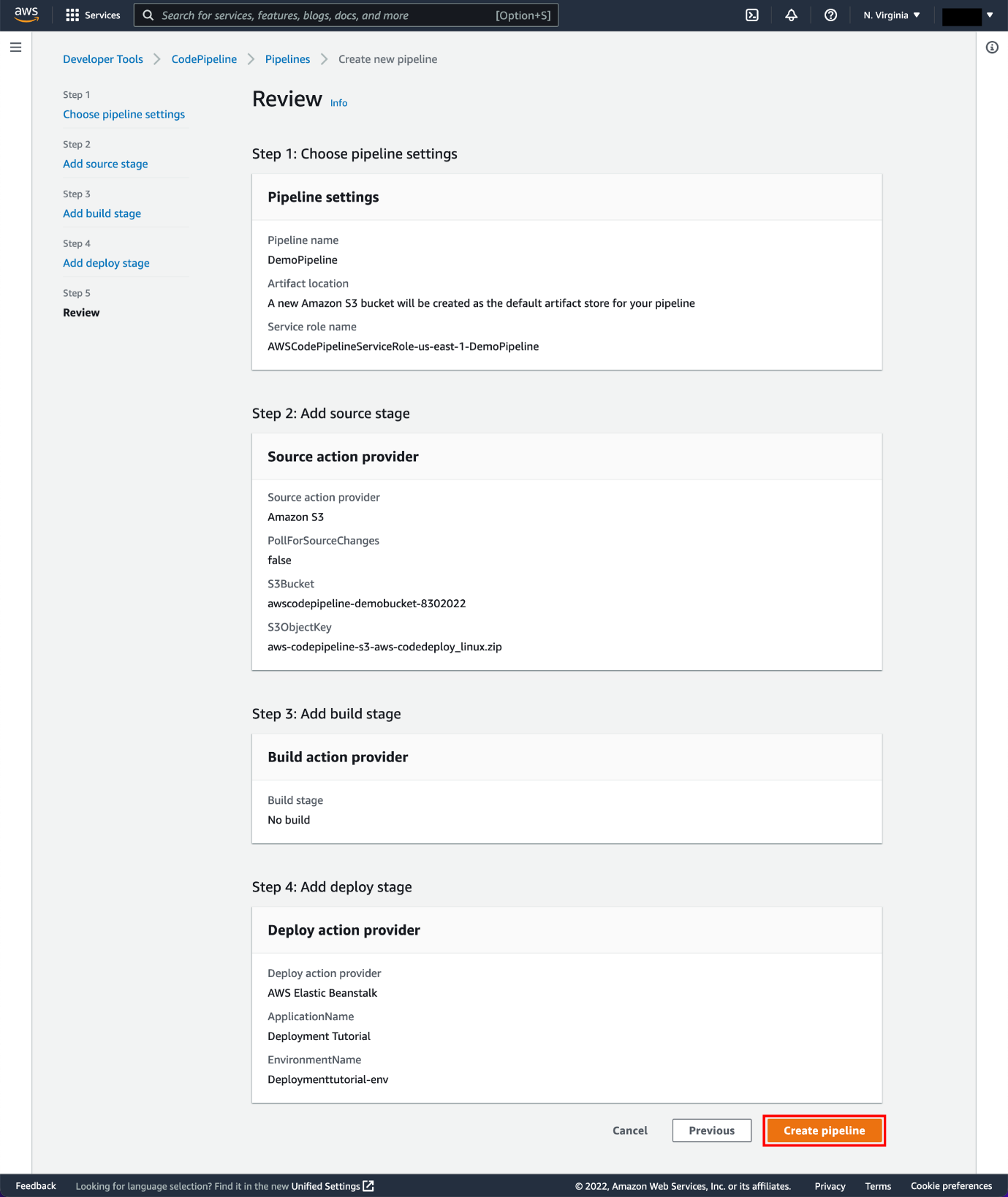


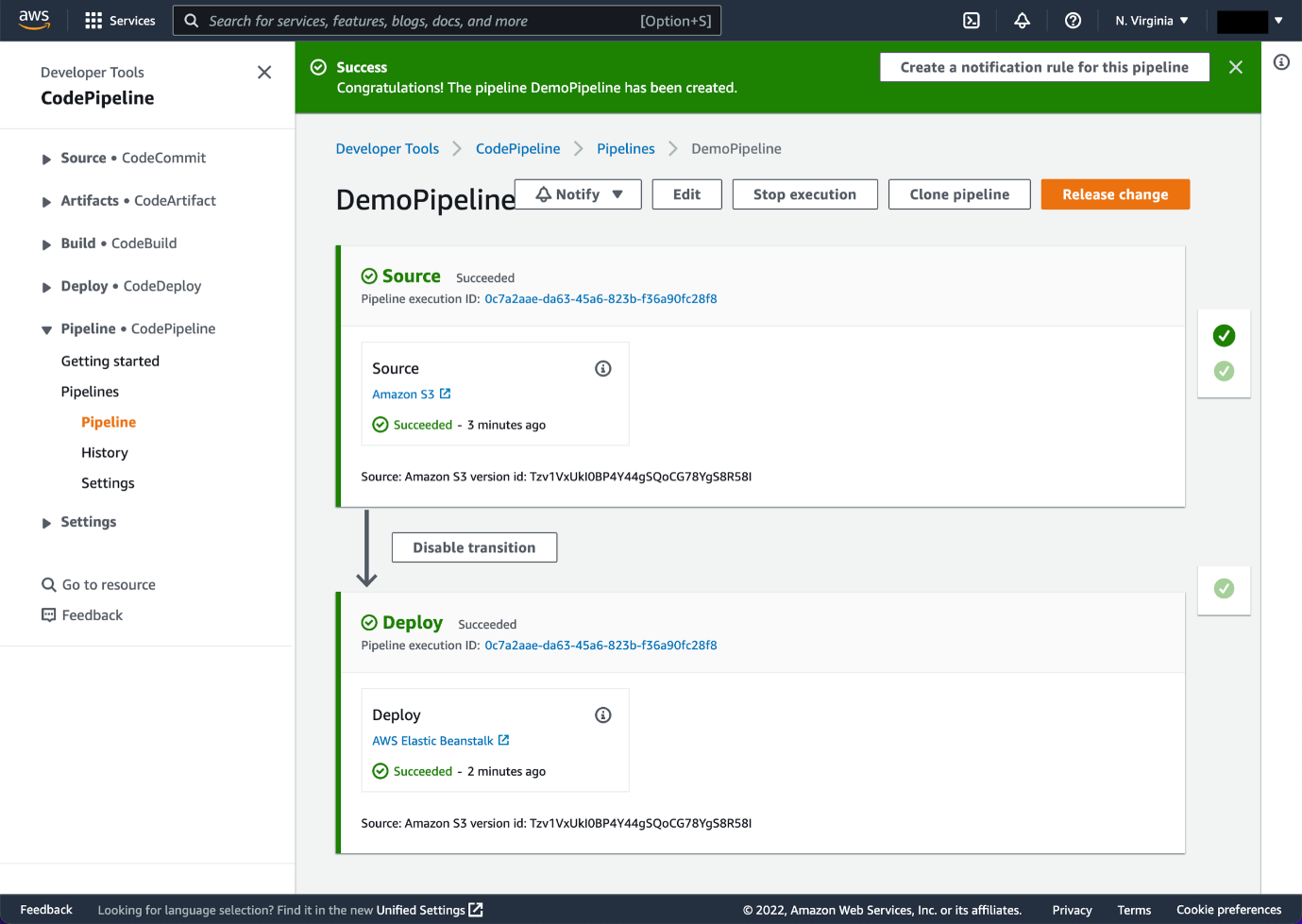
 In the*Step 4: Add deploy stage* page:

* Deploy provider: Select AWS Elastic Beanstalk.
* Region: Retain the default region.
* Application name: Select Deployment Tutorial.
* Environment name: Select Deploymenttutorial-env.
* Click Next.

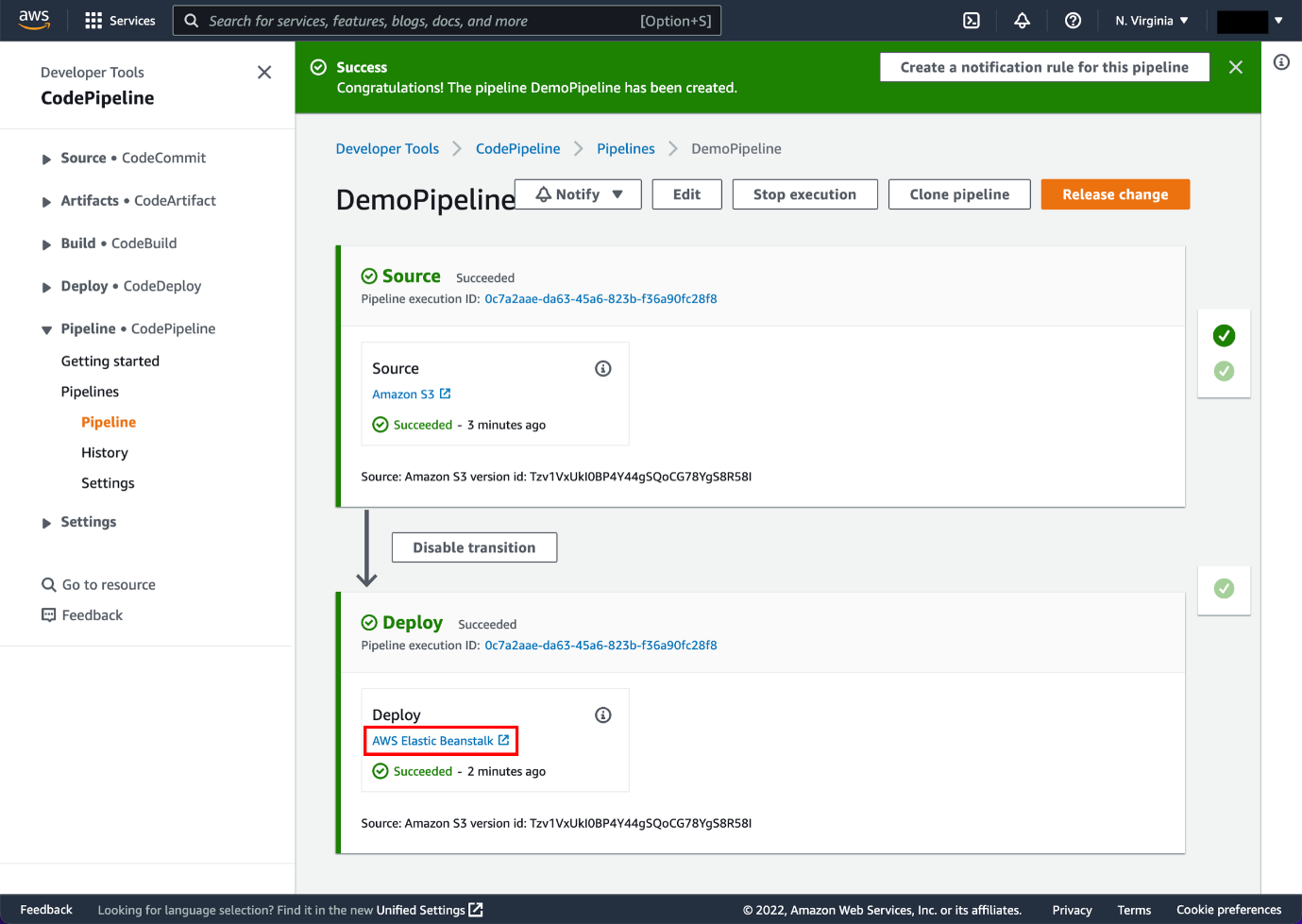


Review page, review the information and choose Create pipeline.



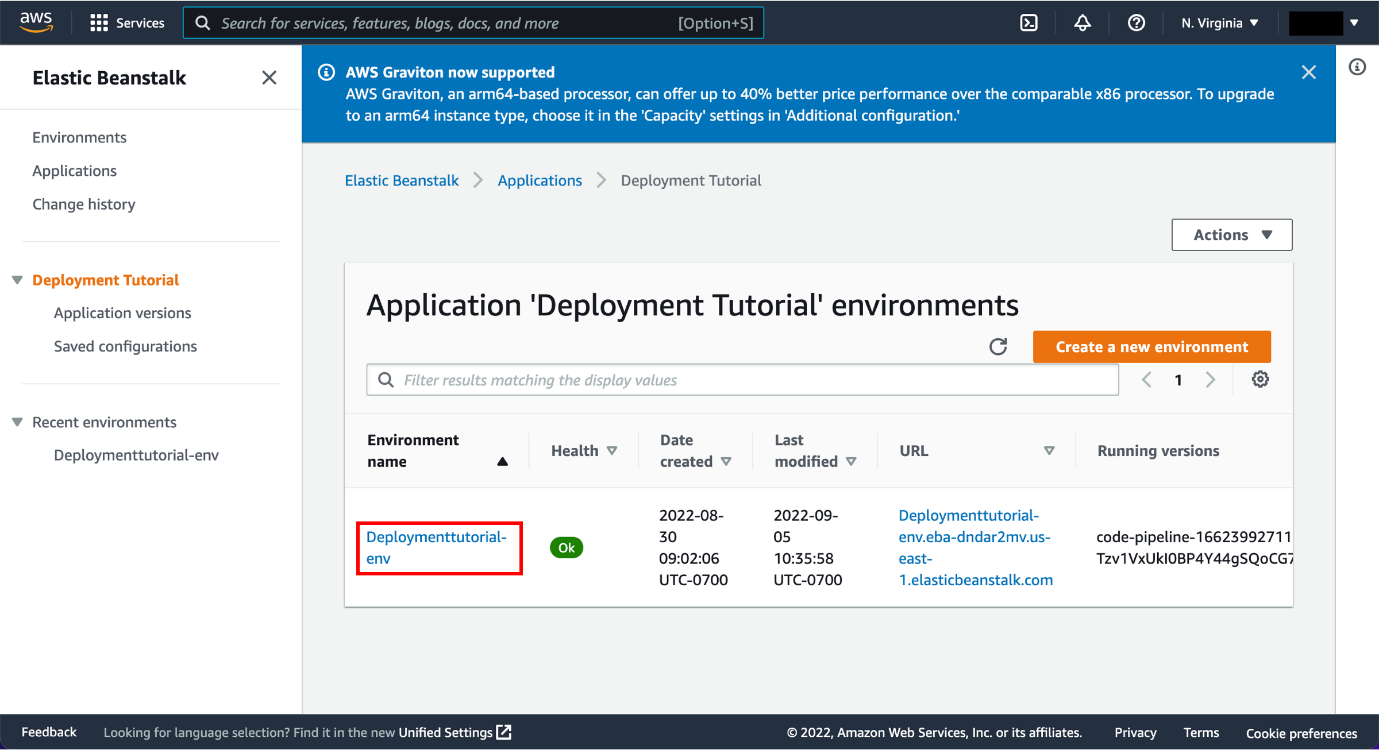


In the status area for the Beta stage, select AWS Elastic Beanstalk

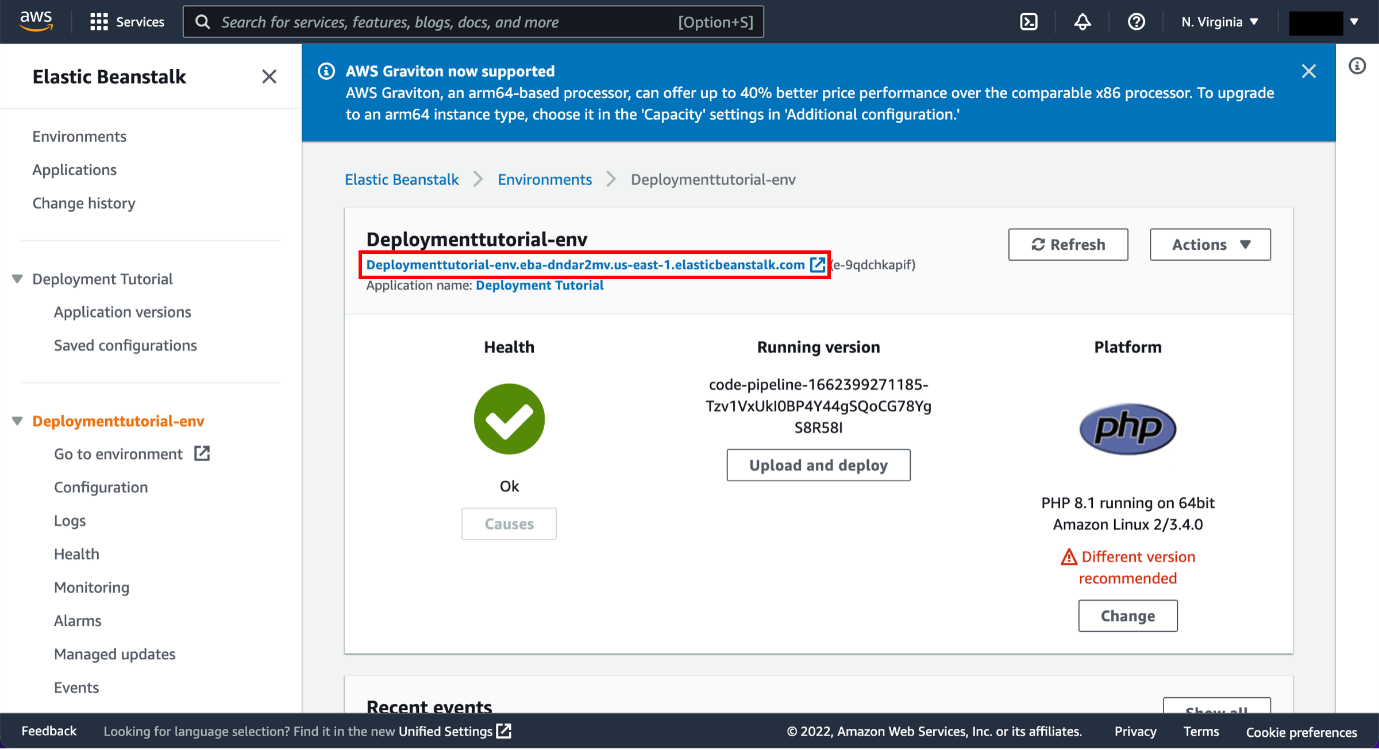


The AWS Elastic Beanstalk console opens with the details of the deployment.

* Select the environment you created earlier, called *Default-Environment test1l-env.*



Select the URL that appears in the upper-right part of the page to view the sample website you deployed.





**Conclusion**

Hence we learnt about 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 (Amazon S3) bucket, or an AWS CodeCommit repository as the source location for the sample app’s code and also about AWS Elastic Beanstalk is an orchestration service offered by Amazon Web Services for deploying applications which orchestrates various AWS services, including EC2, S3, Simple Notification Service, CloudWatch, autoscaling, and Elastic Load Balancers.