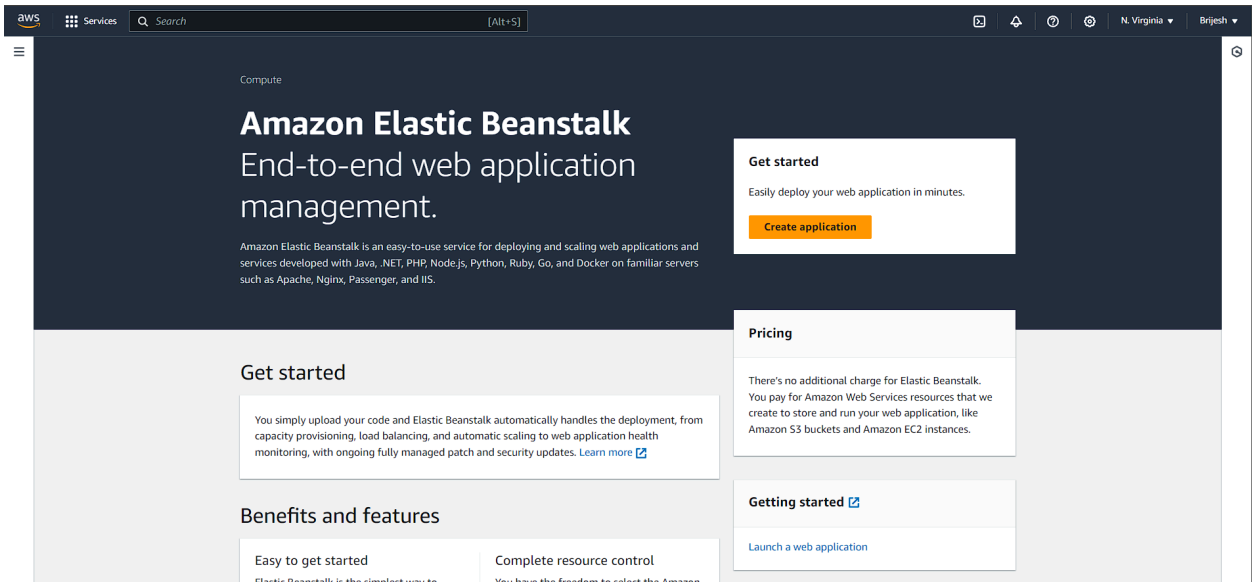
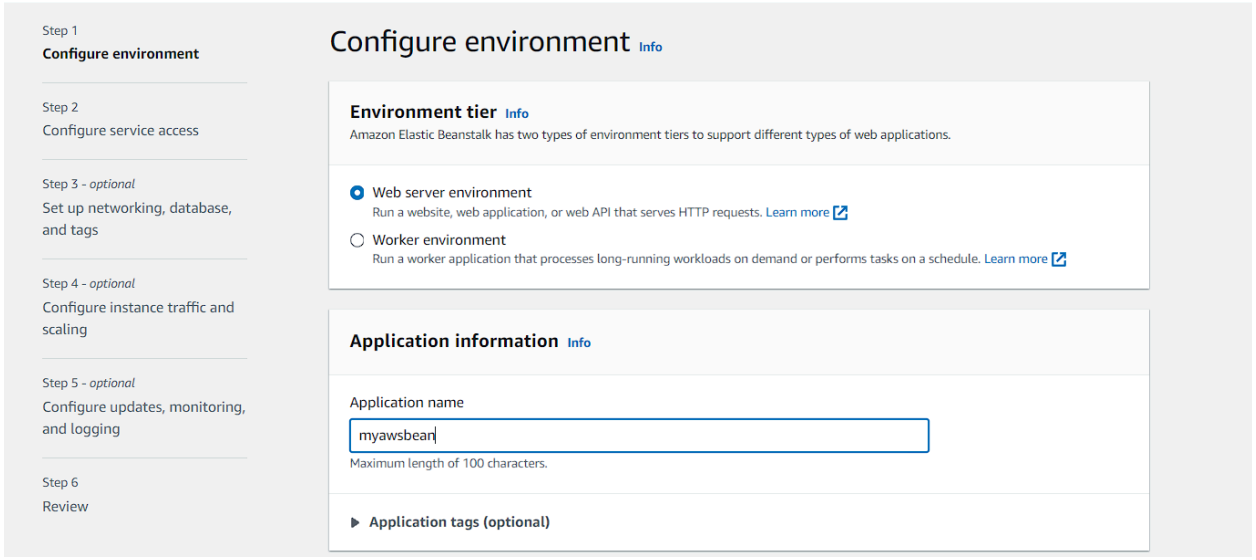
Aim : To Build Your Application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline, deploy Sample Application on EC2 instance using AWS CodeDeploy

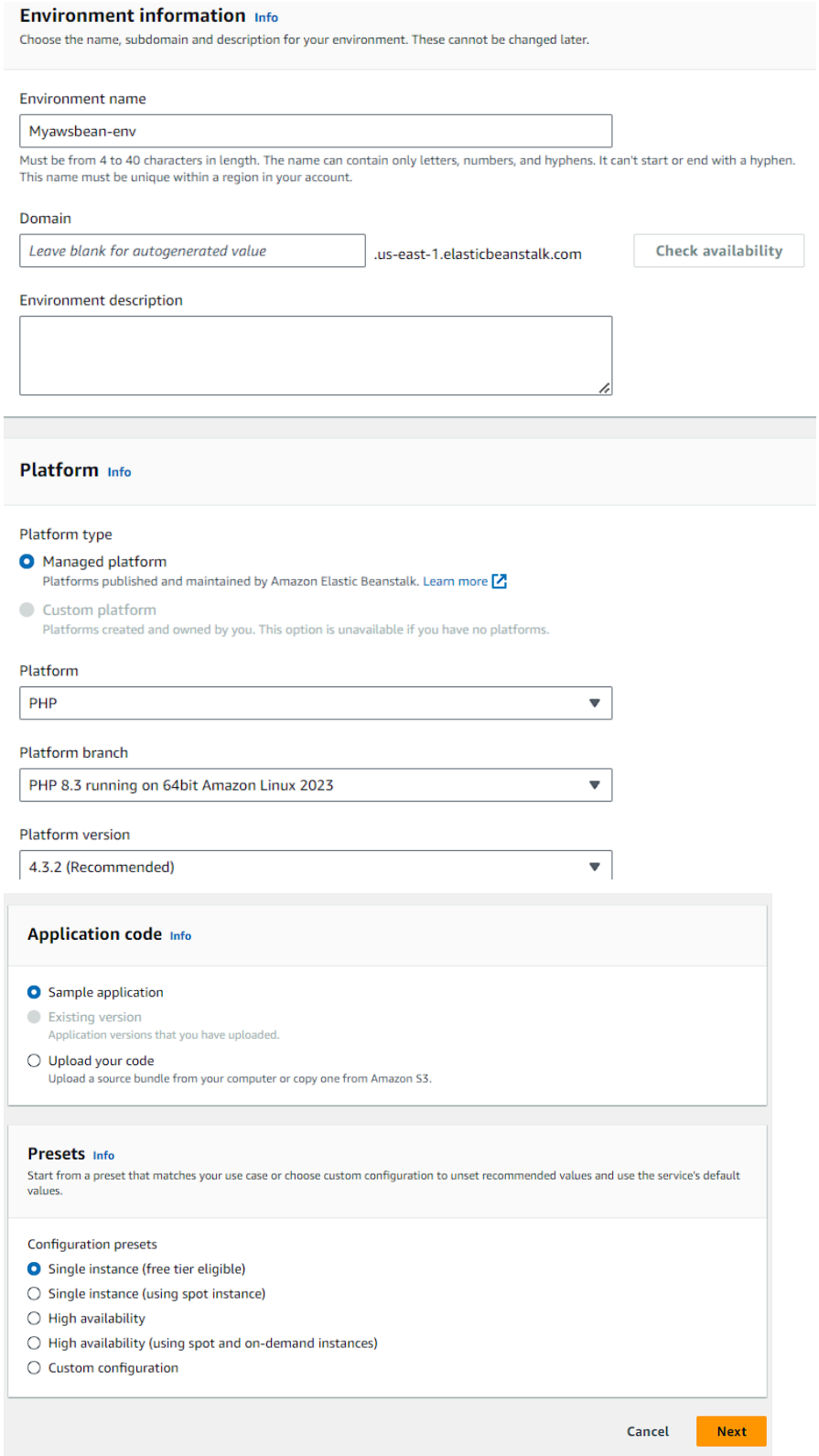
1) Open the aws console and then search Elastic Beanst



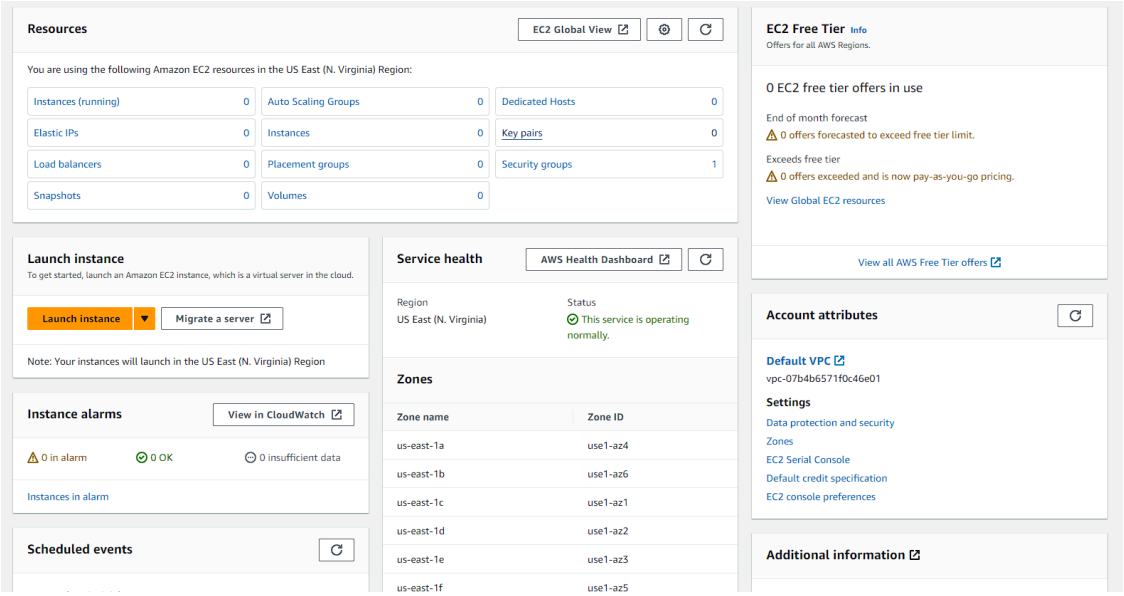
2) Click on create application and configure the environment by adding your application name.



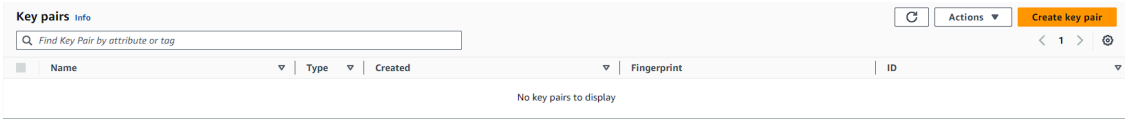
3) Select the environment as PHP and other options as default and click on next.



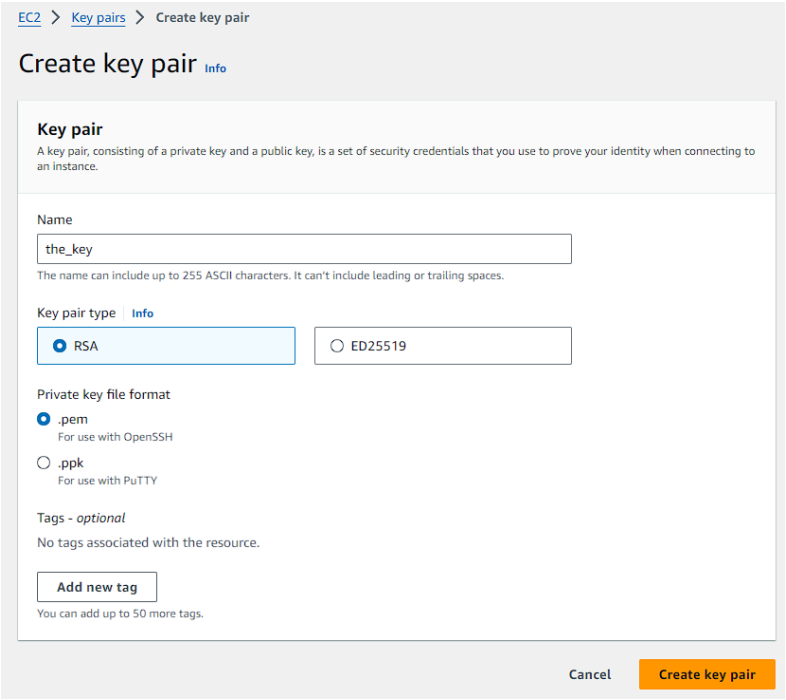
4) After clicking on Next for creating Elastic Beanstalk, we need key-pair that will be require for deployment. Go to EC2 Instance and click on Key Pairs.



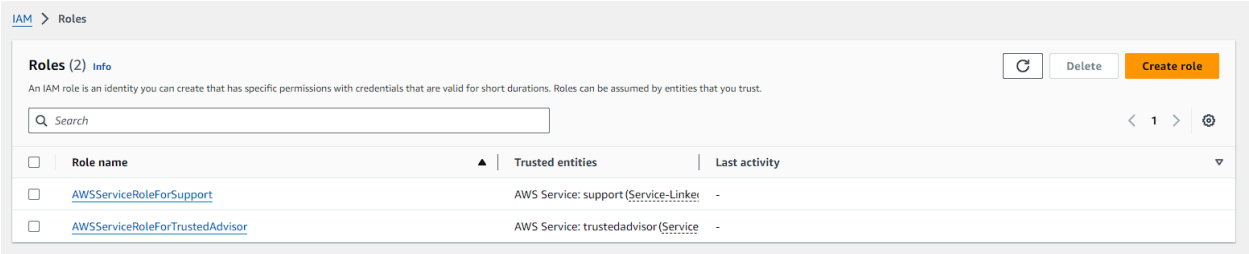
5) Then click on Create key pair



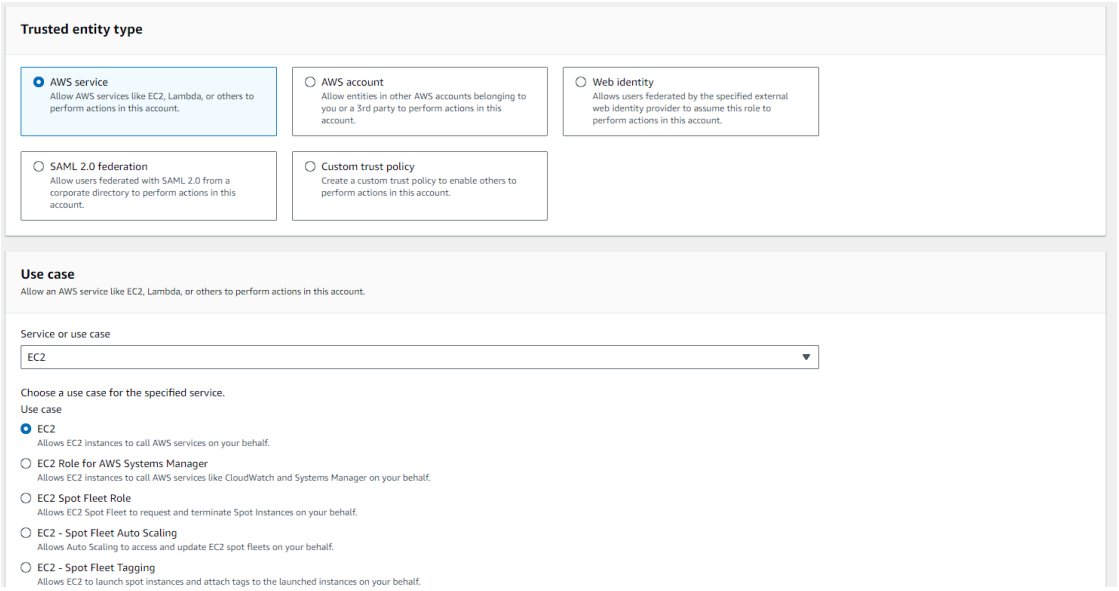
6) Input the name of the key-pair and select pem as file format and click on Create key pair.



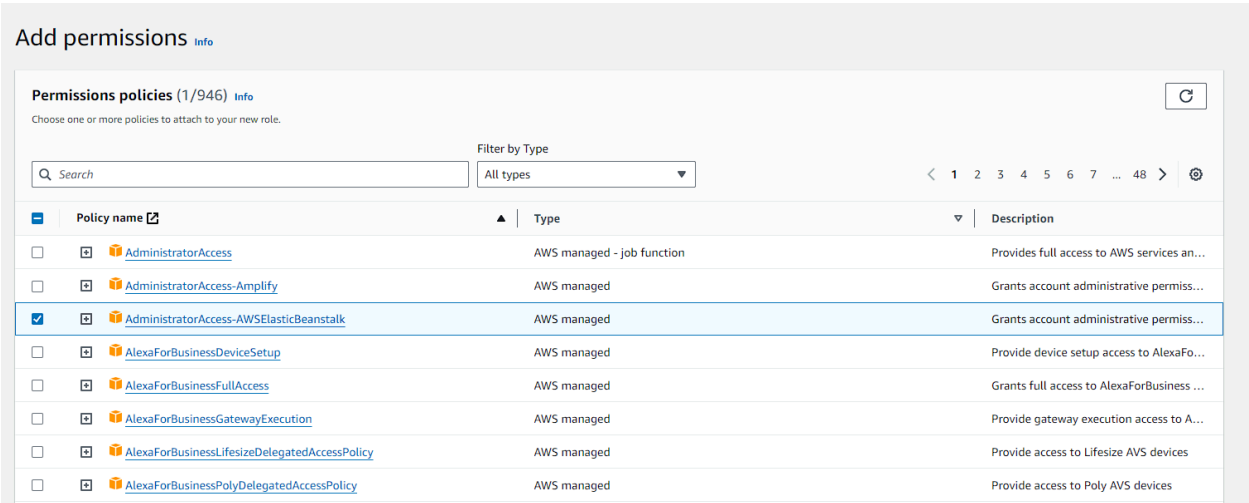
7) After creating key pair, open new tab and go to IAM to create a role that will be used to build Codepipeline. Click on Create role.



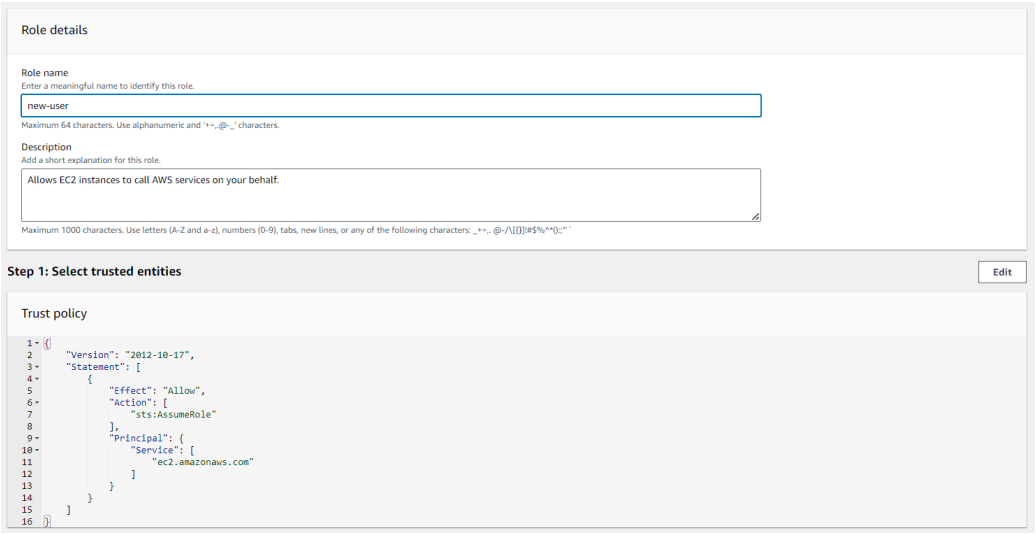
8) Select AWS service as Trusted Entity type and EC2 as service.



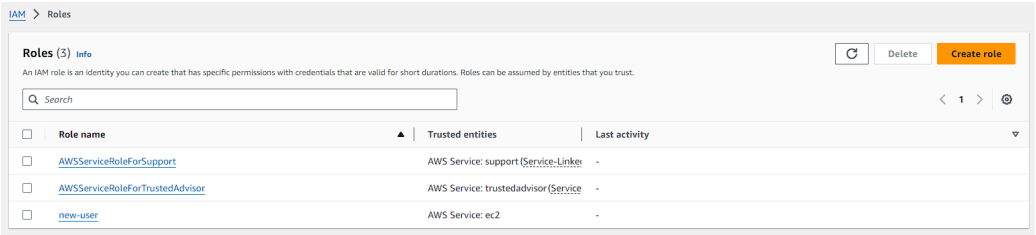
9) Choose AdministratorAccess-AWSElasticBeanstalk as Policy and click on Next.



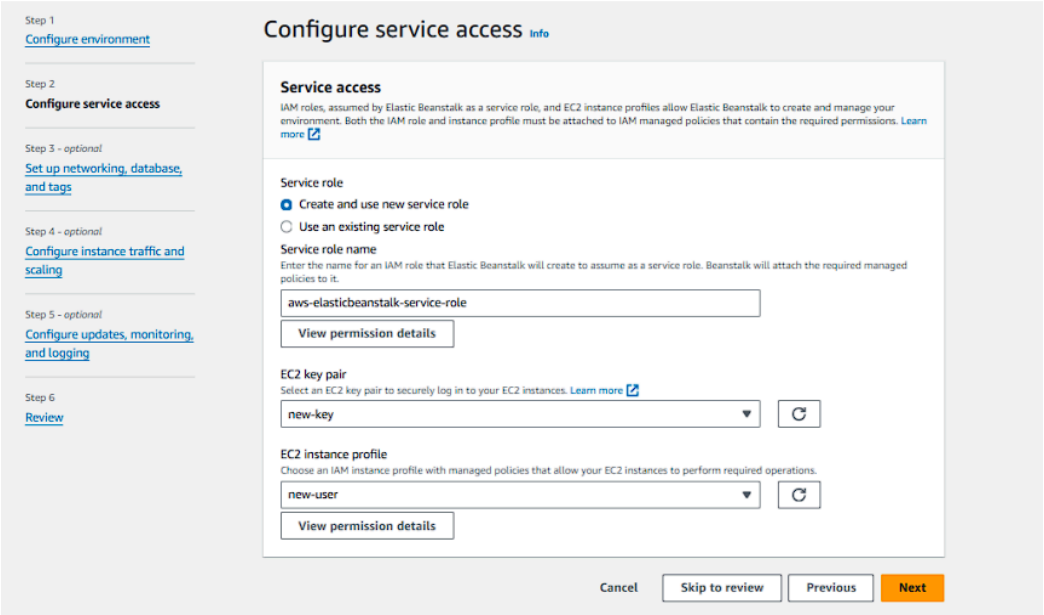
10) Name the role and keep other as default.

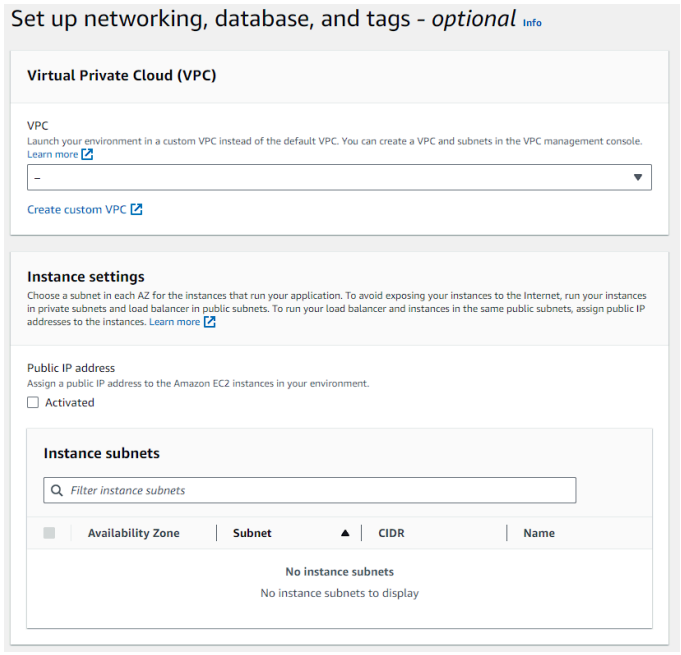


11) The role is created successfully.

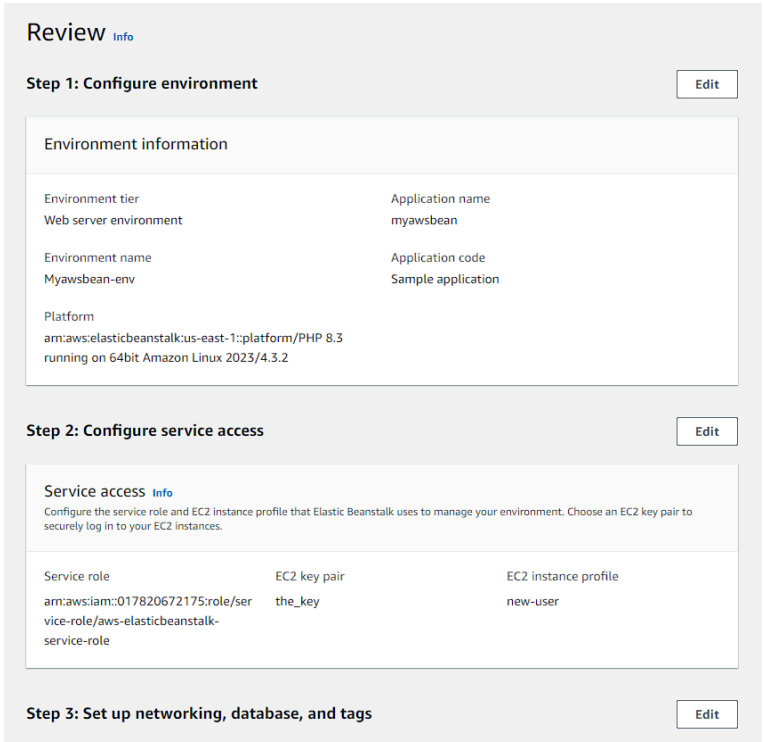


12) Now move to the tab where Elastic Beanstalk was opened and from the drop down menu select the newly created key pair and instance profile. Now let everything be default.

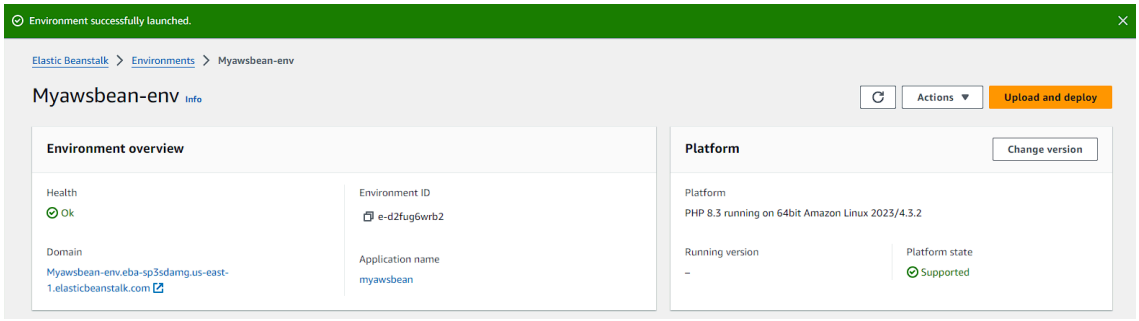




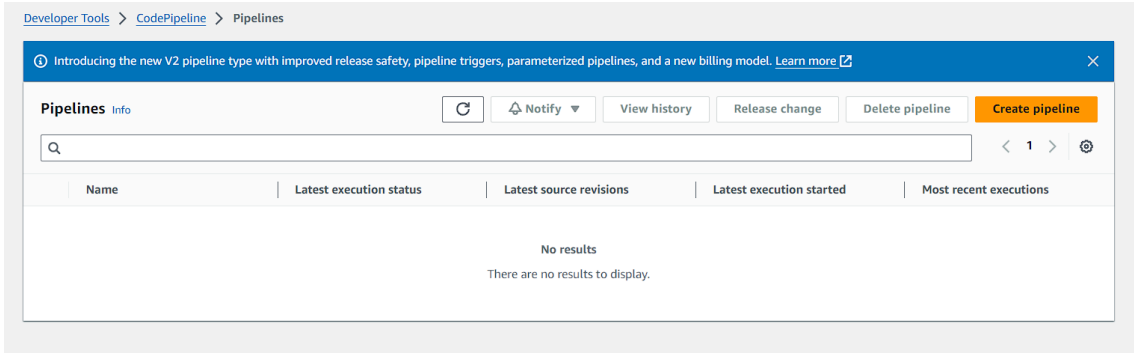
13) Review the changes and click on Create.



14) Your sample environment is created 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.



15) Now, we need to make a CodePipeline. Go to CodePipeline and click on Create Pipeline.



16) Name the pipeline and select the service role as below and click on Next.

Choose pipeline settings Info

Step 1 of 5

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.

mypipeline

No more than 100 characters

Pipeline type

You can no longer create V1 pipelines through the console. We recommend you use the V2 pipeline type with improved release safety, pipeline triggers, parameterized pipelines, and a new billing model.

Execution mode
Choose the execution mode for your pipeline. This determines how the pipeline is run.

- Superseded
  A more recent execution can overtake an older one. This is the default.
- Queued (Pipeline type V2 required)
  Executions are processed one by one in the order that they are queued.
- Parallel (Pipeline type V2 required)
  Executions don't wait for other runs to complete before starting or finishing.

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-mypipeline

Type your service role name

Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Variables

You can add variables at the pipeline level. You can choose to assign the value when you start the pipeline. Choosing this option requires pipeline type V2. Learn more

No variables defined at the pipeline level in this pipeline.

Add variable

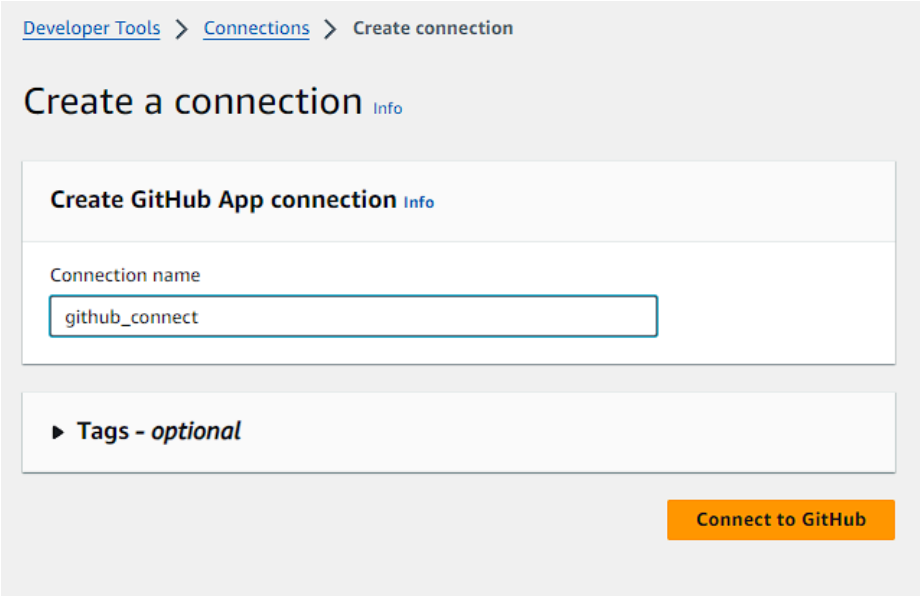
You can add up to 50 variables.

The first pipeline execution will fail if variables have no default values.

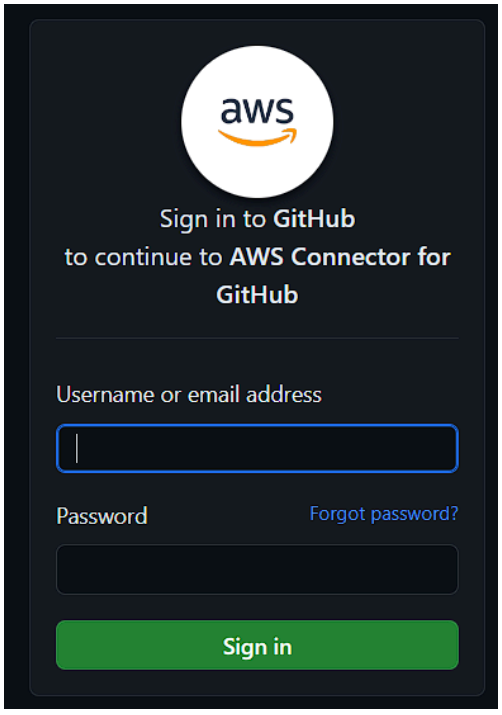
Advanced settings

Cancel Next

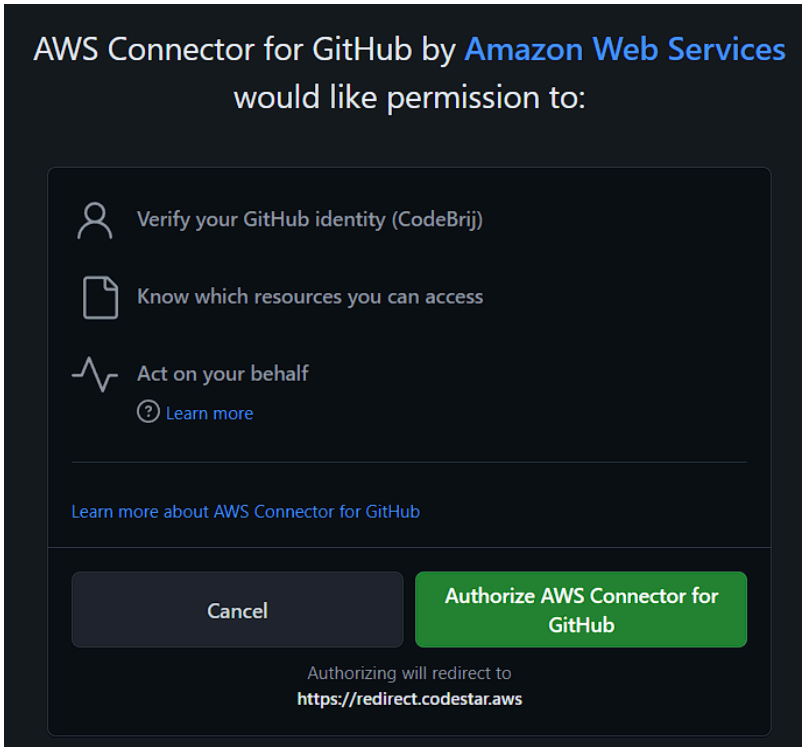
17) In the source stage select Github v2 as the provider and then connect your github connect so that the pipeline can access the forked source code. Name the connection.



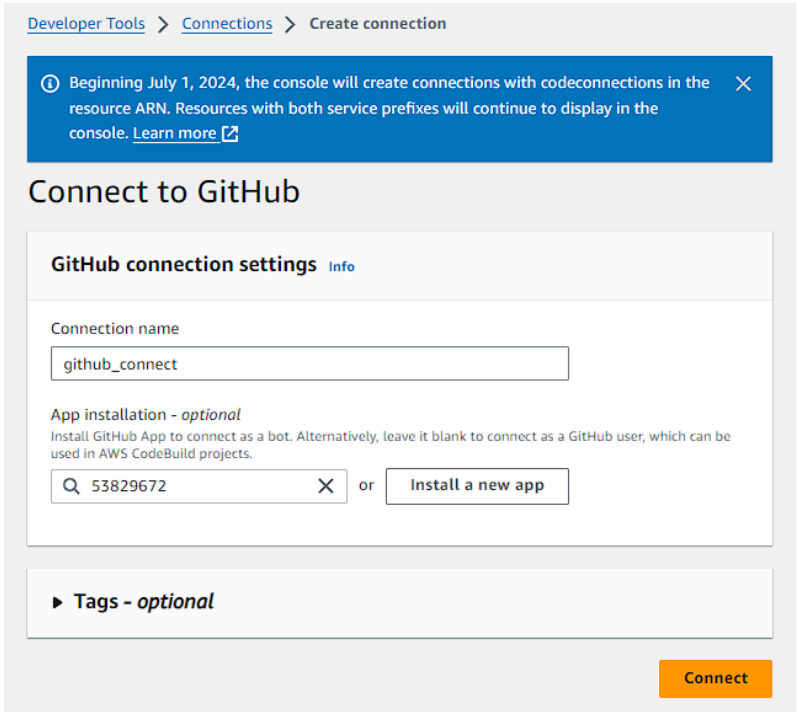
18) Signin to GitHub to connect with AWS.



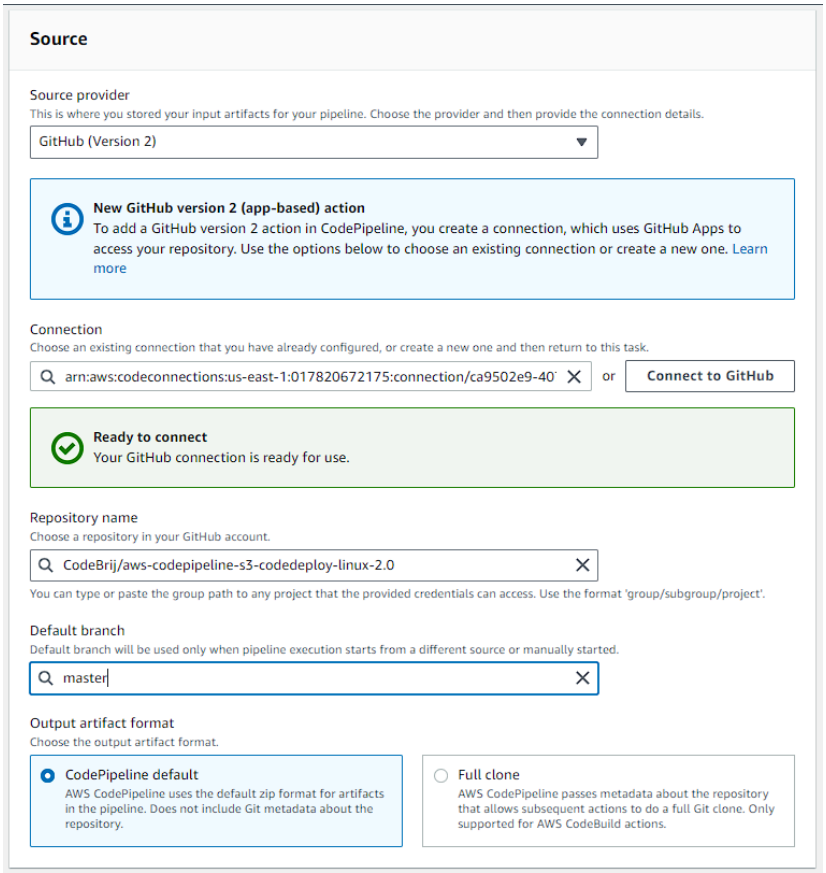
19) Authorize AWS Connector for GitHub.



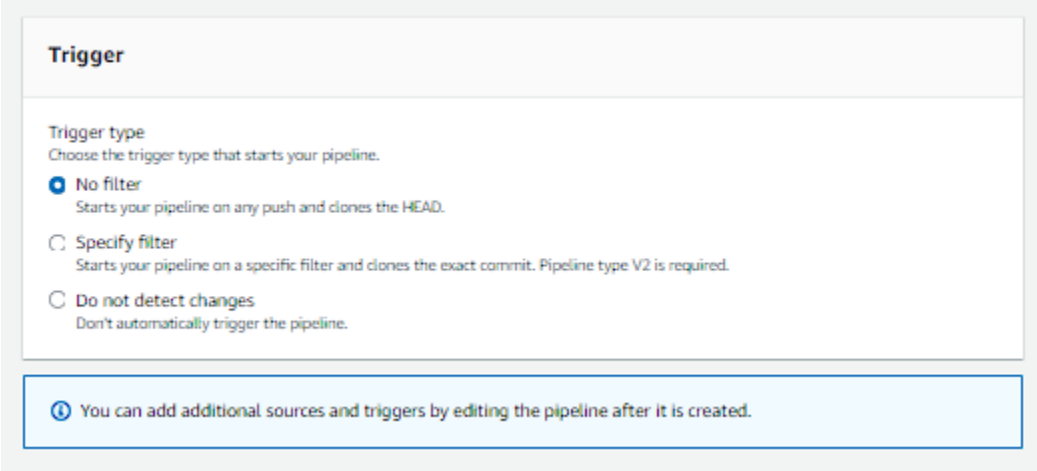
20) We need to install the GitHub connector.



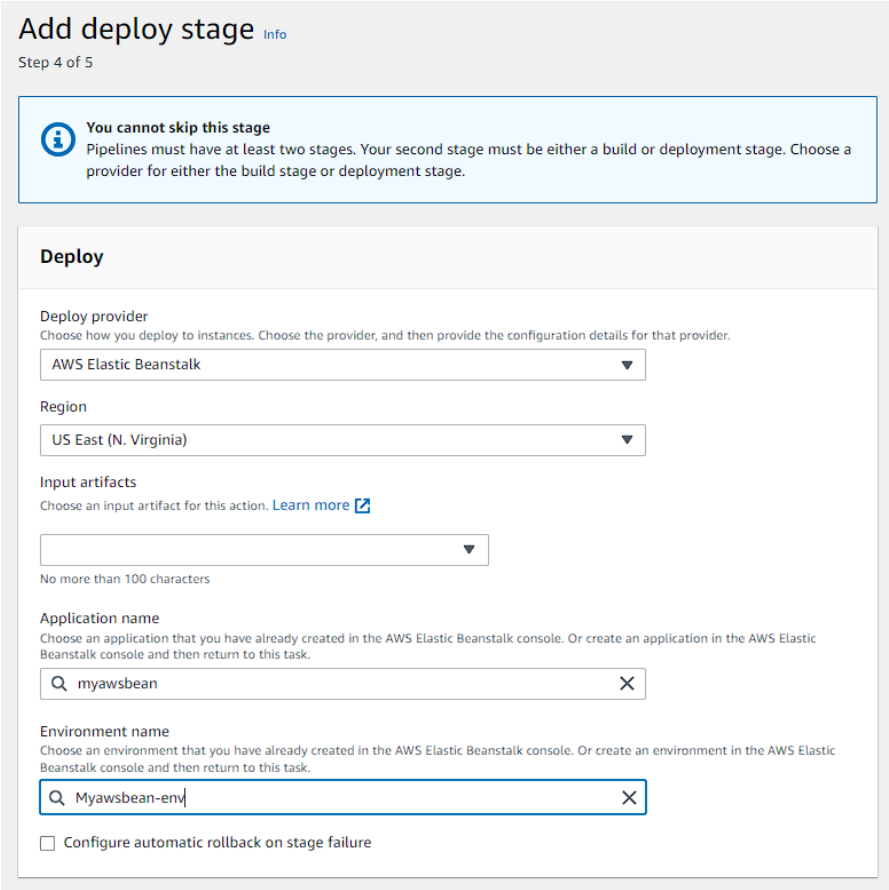
21) Now, select the repository and the branch to be deployed.



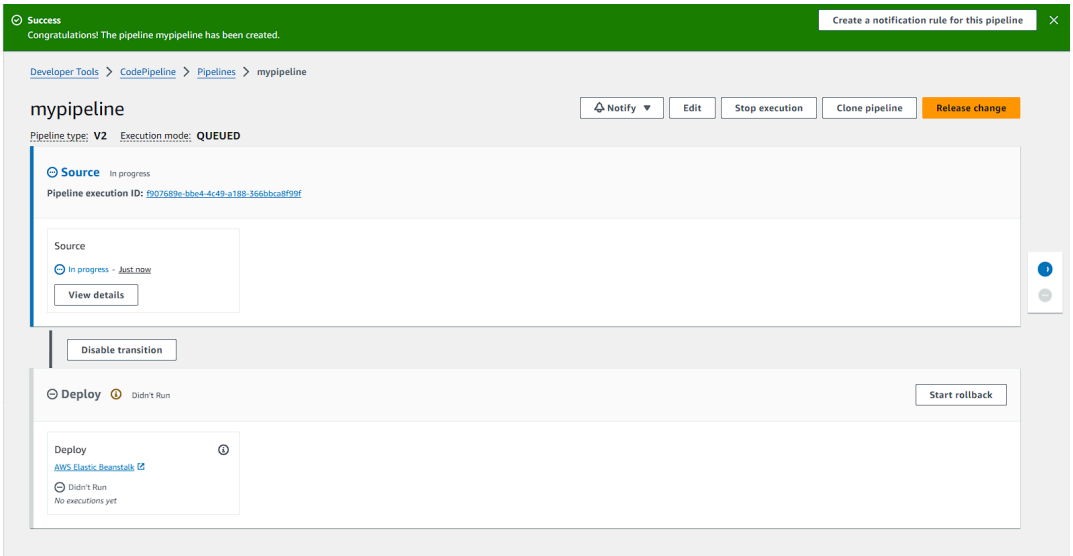
22) Select No fliter in Trigger.



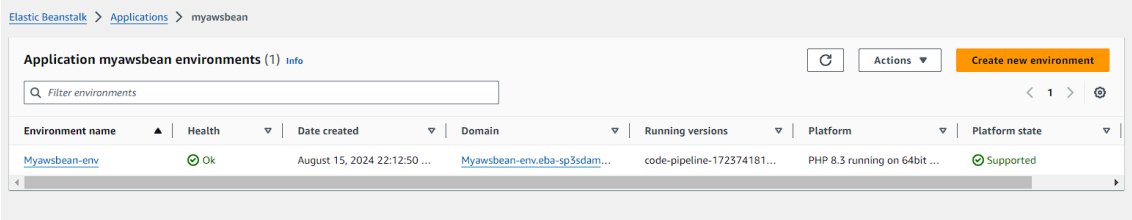
23) In deploy stage add application name as environment name. Then review the settings and click on Create pipeline.



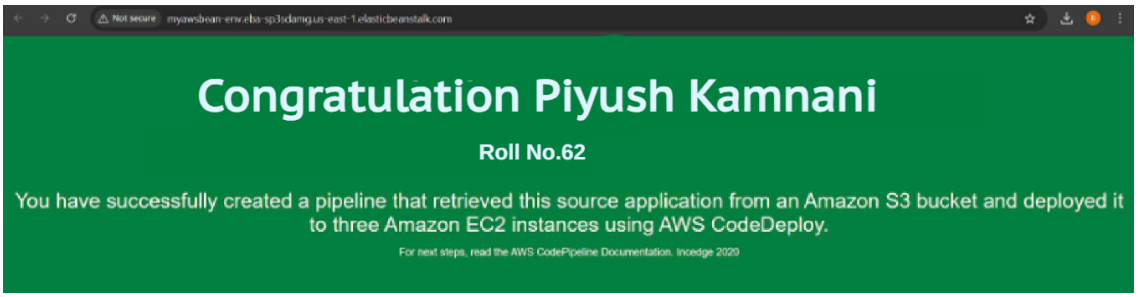
24) The pipeline is ready and the provided repo is deployed successfully.



25) Go to Elastic Beanstalk and from DOmain open the hosted site.



26) Hosted site from the github repository.



27) Make some changes in the index.html and reload.