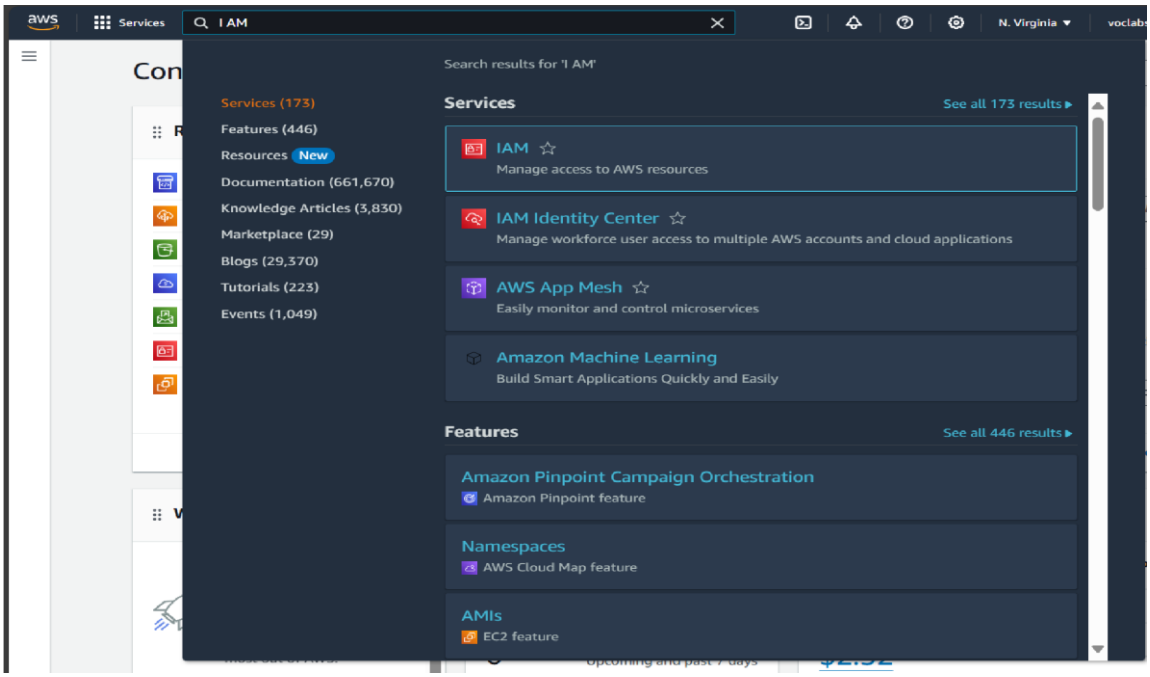
Komal Sabale
D15C-45

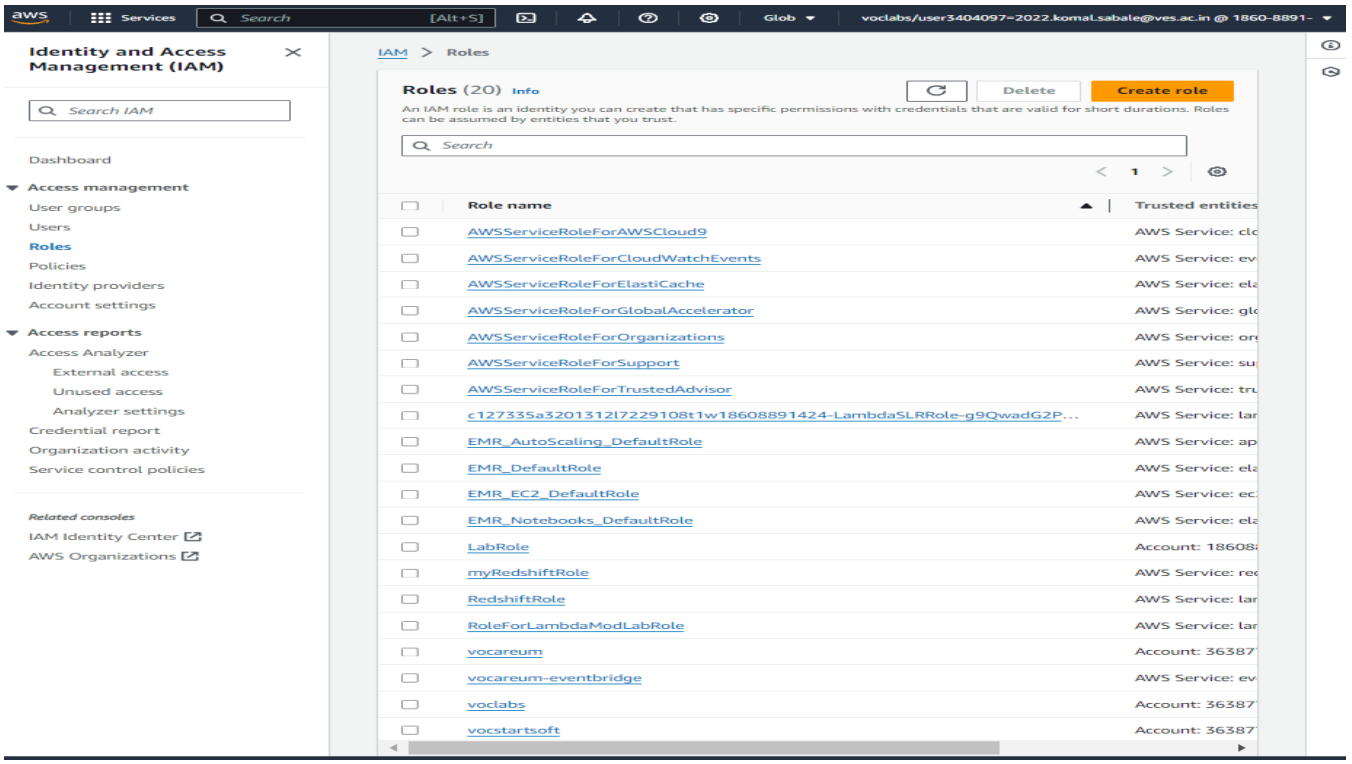
Experiment No: 2

Step1:- Creation of role:-

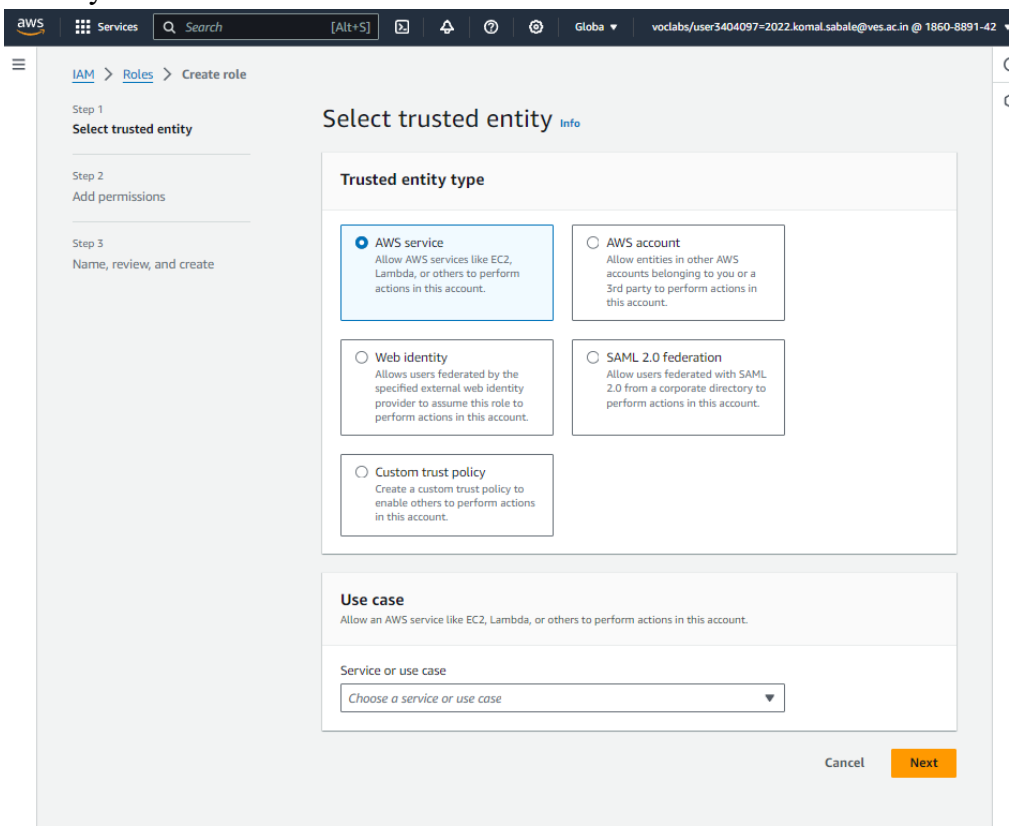
1. Login to your AWS account and search for IAM



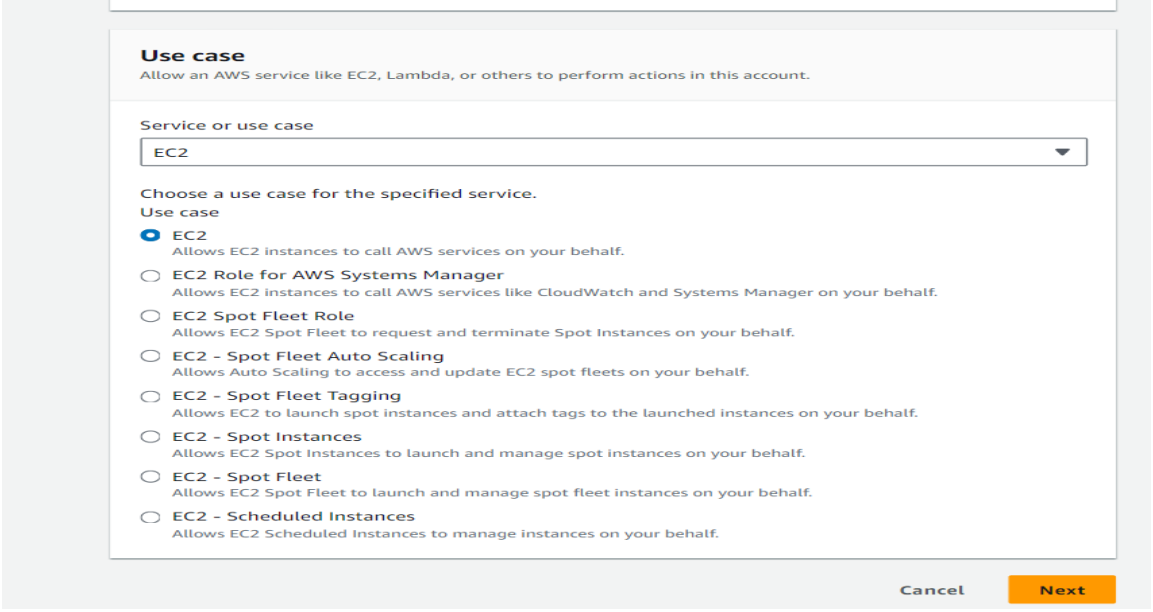
2. Then go into the role section and click on create role.



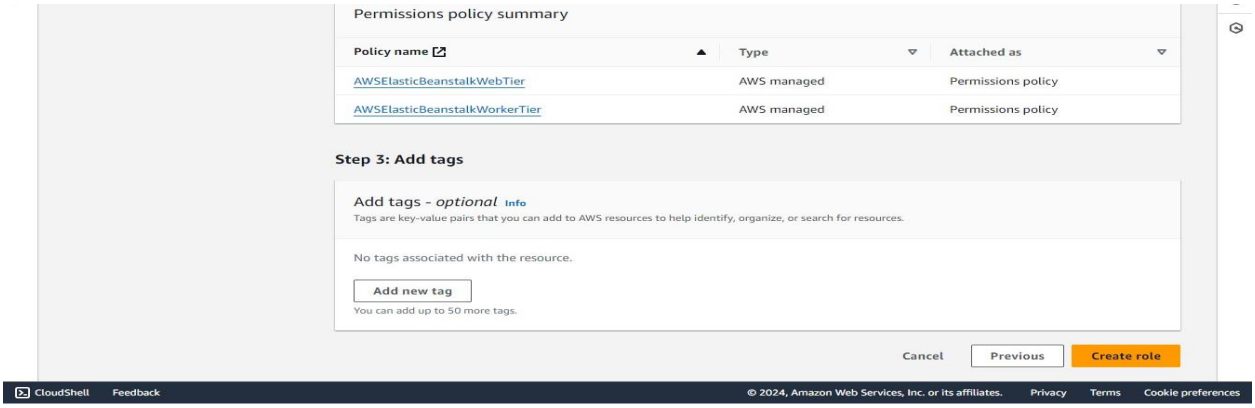
3. Then select a trusted entity as AWS service.



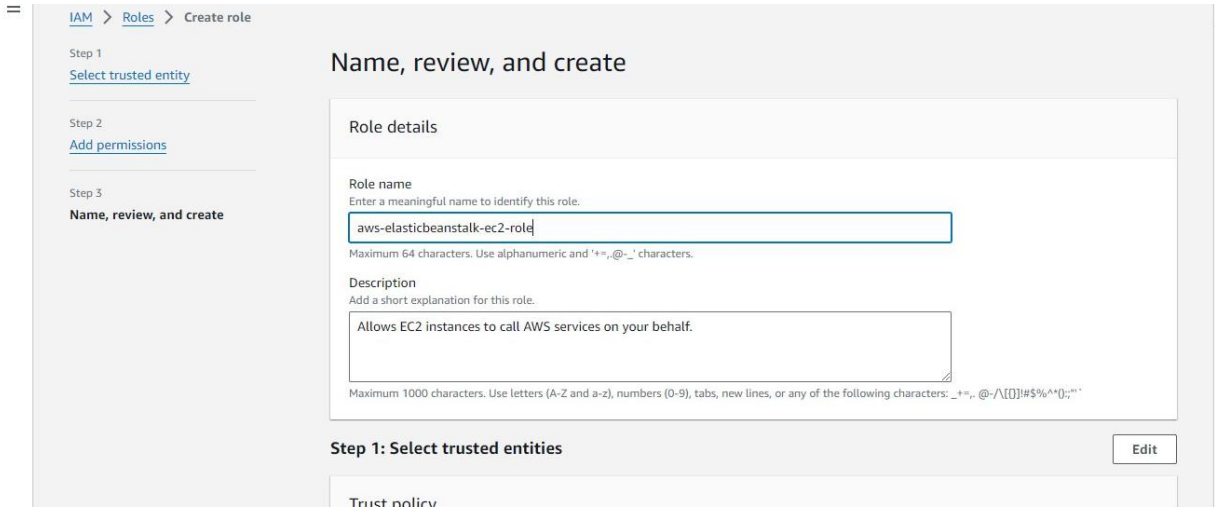
4. Select use case as EC2.



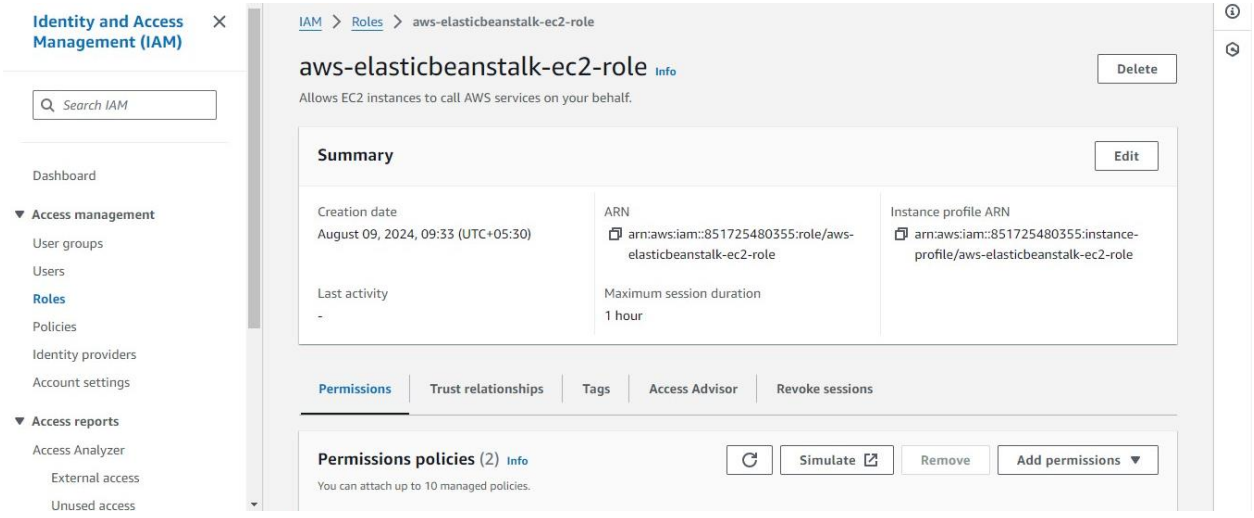
5. Select permissions as AWS Elastic Benstalk Web Tier.and AWS elastic Benstalk worker tier.



6. Give a name to Role. Here I have given my role name as aws -elasticbenstalk -ec2 role.



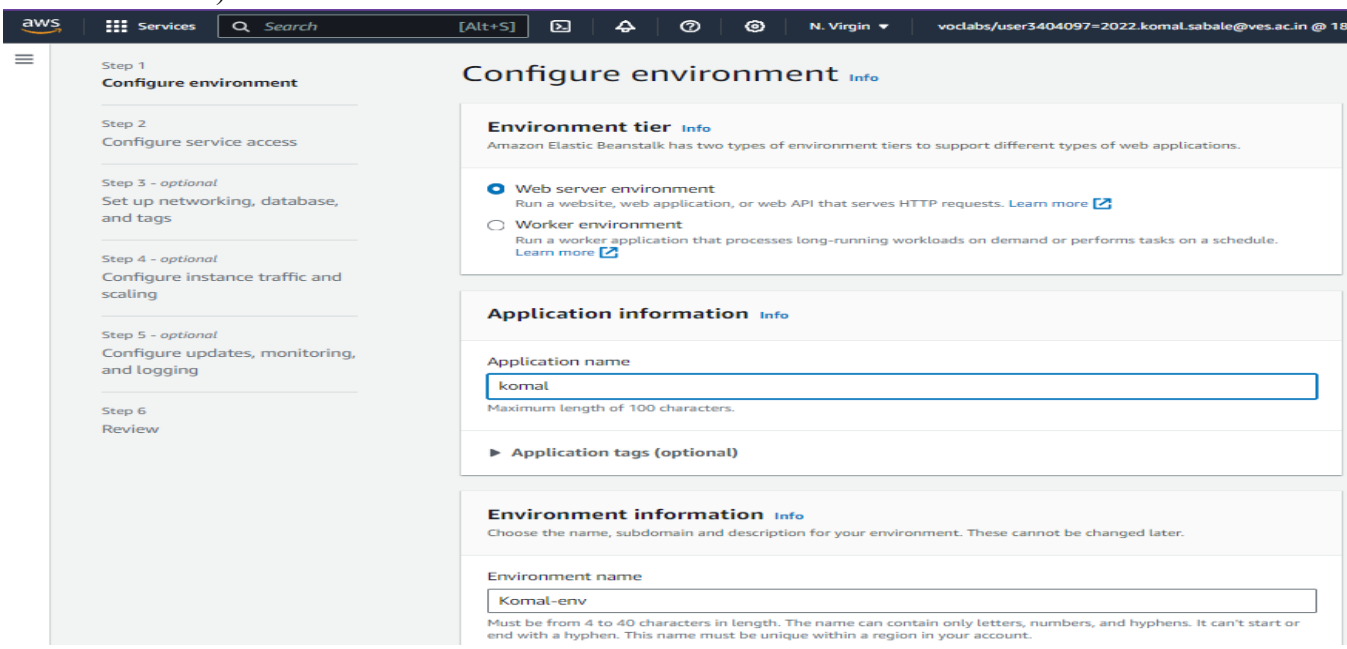
7. Then the role gets created.



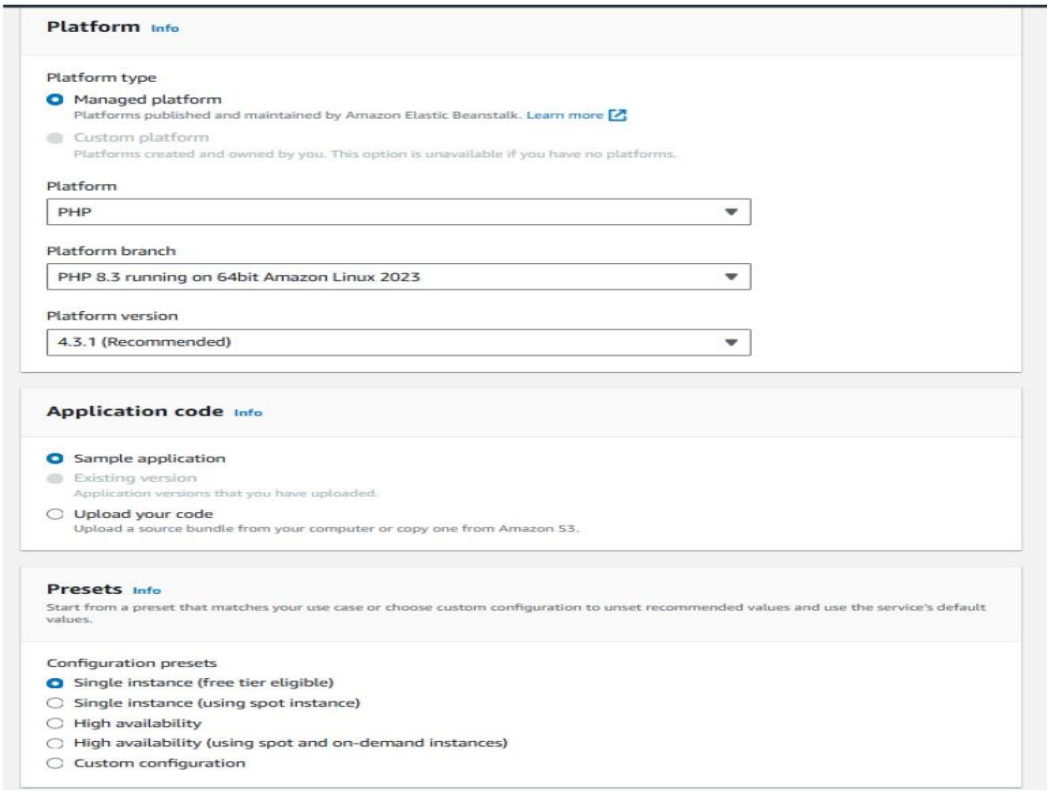
Komal Sabale
D15C-45

Step2:- Creation Elastic Beanstalk Environment 1

1. Search and Open up Elastic Beanstalk and name your web app. (here I have given name komal)

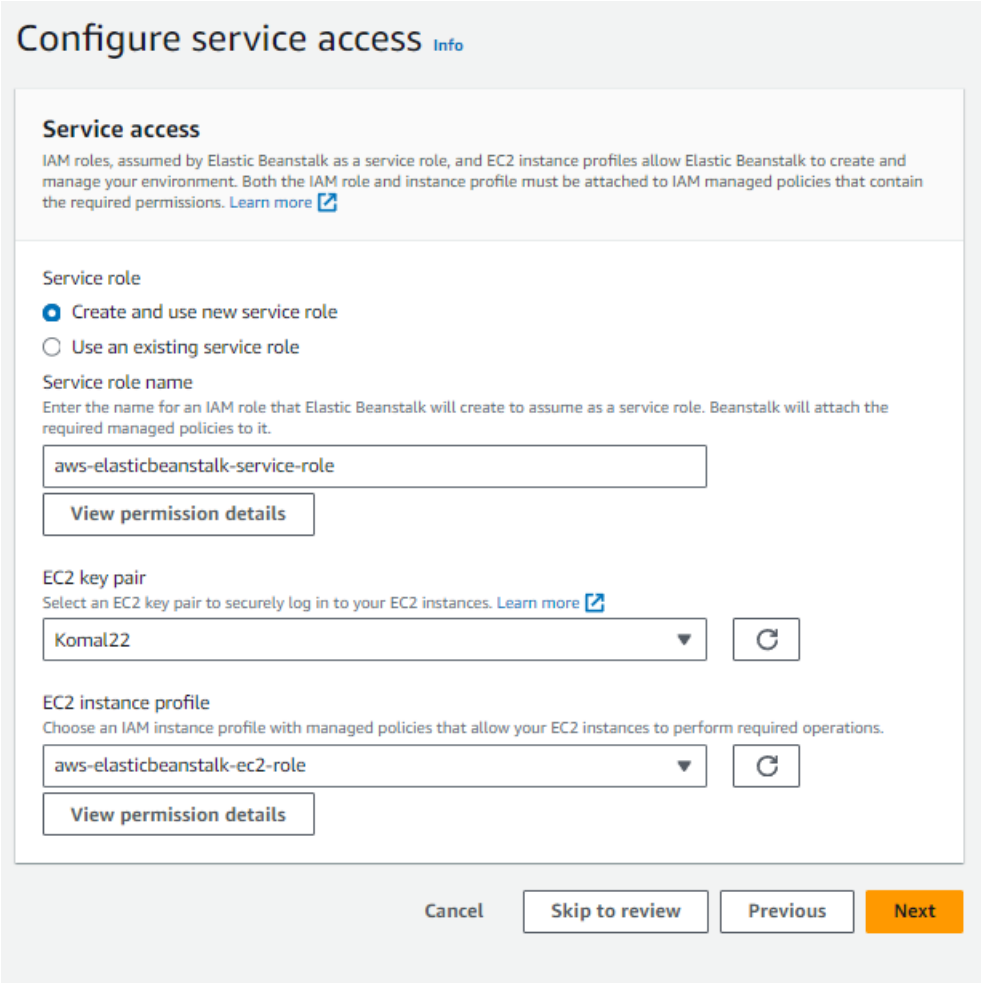


2. Select platform as PHP.



Komal Sabale
D15C-45

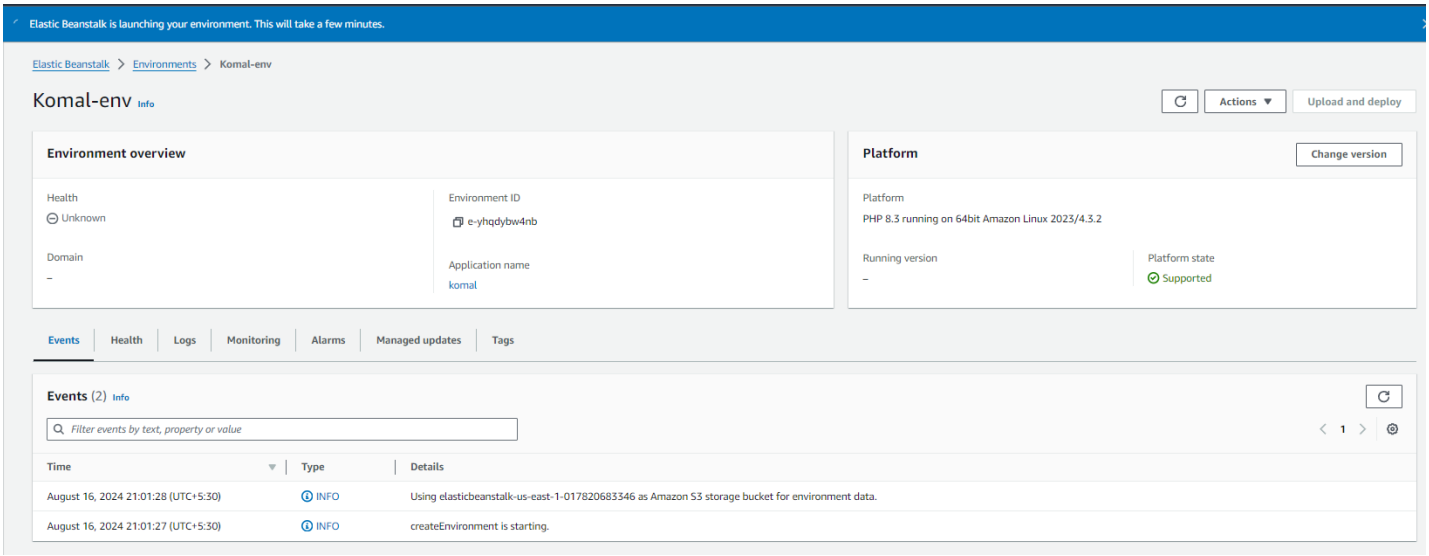
3. After clicking on next u need to select the use existing role. Then you will see the existing role select it like here it is aws-elasticbeanstalk-service-role. Keep other serttings as it is.



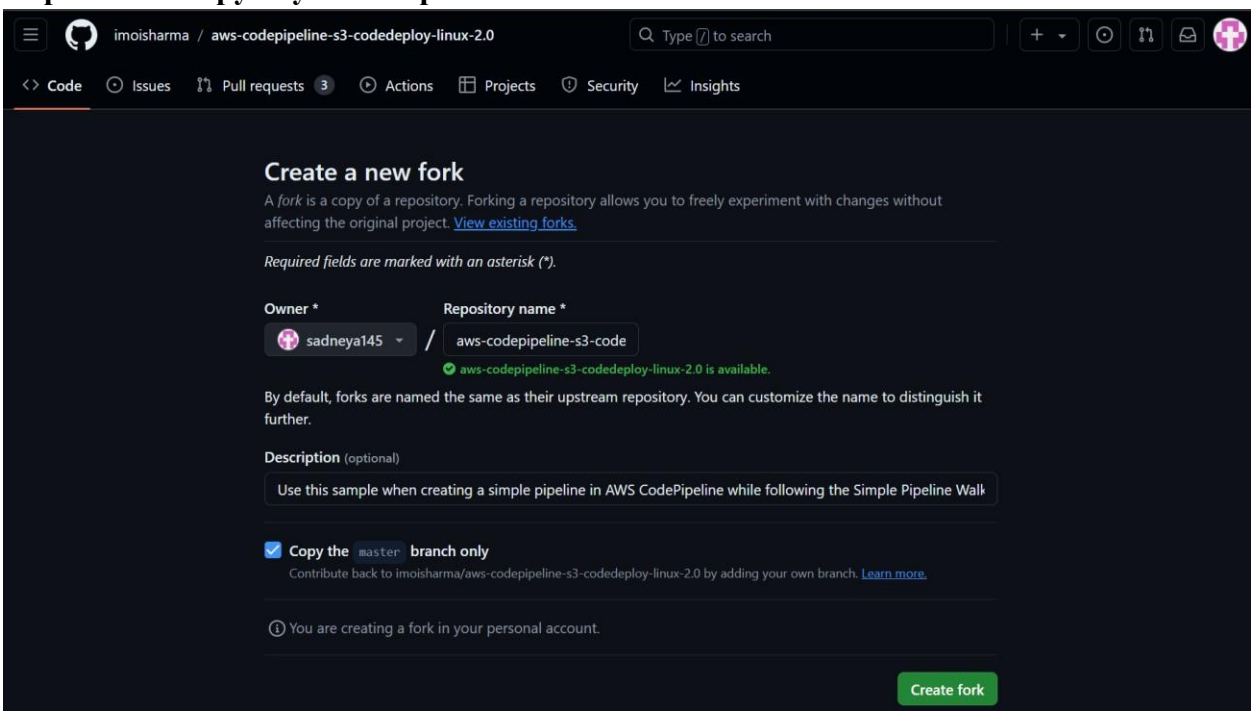
Komal Sabale
D15C-45

Keep Set up networking, database, and tags, Configure instance traffic and scaling, Configure updates, monitoring, and logging all these default.

4. Beanstalk creates a sample environment 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.



Step 3: Get a copy of your sample code

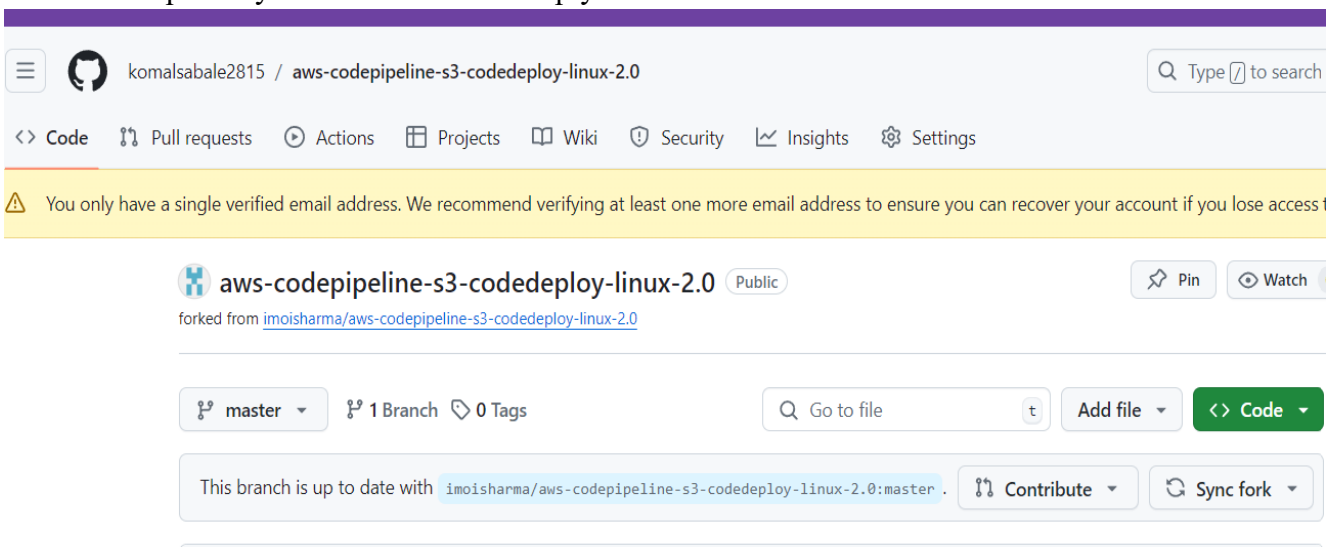


Komal Sabale
D15C-45

In this step, we will get the sample code from this GitHub Repository to later host it. The pipeline takes code from the source and then performs actions on it.

For this experiment, as a source, we will use this forked GitHub repository. We can alternatively also use Amazon S3 and AWS CodeCommit.

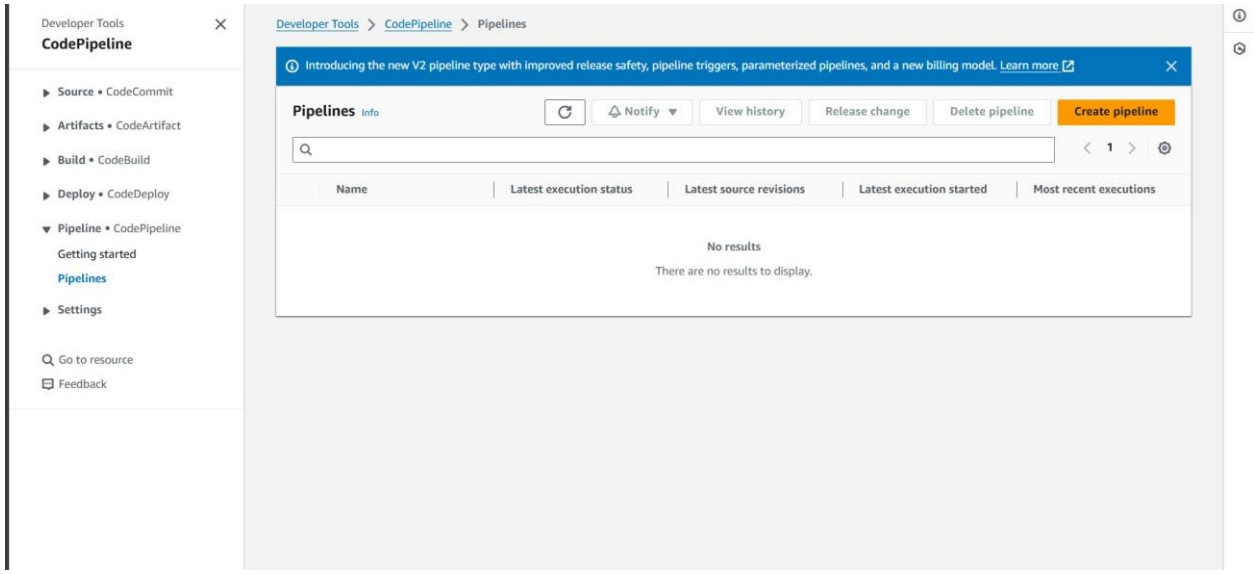
Go to the repository shared above and simply fork it.



Step 4: Creating a CodePipeline

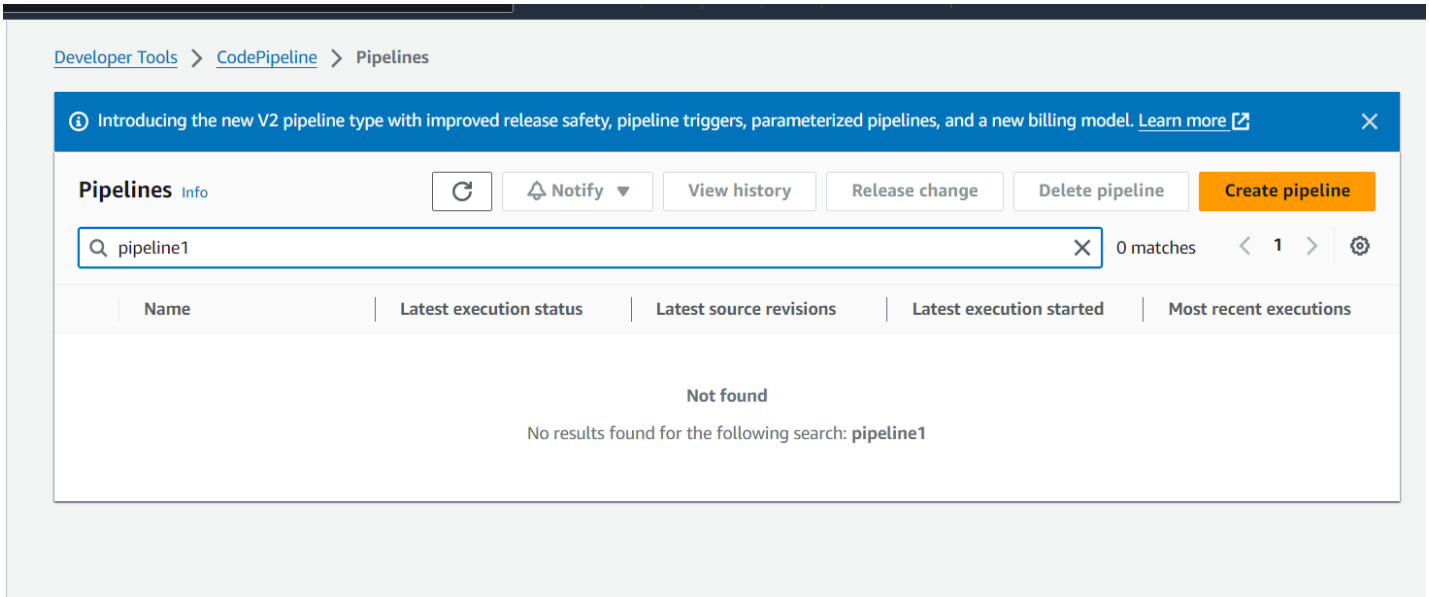
In this step, we'll create a simple pipeline that has its source and deployment information. In this case, however, we will skip the build stage where you get to plug in our preferred build provider.

1. Search CodePipeline in the search bar and click on create a new Pipeline.



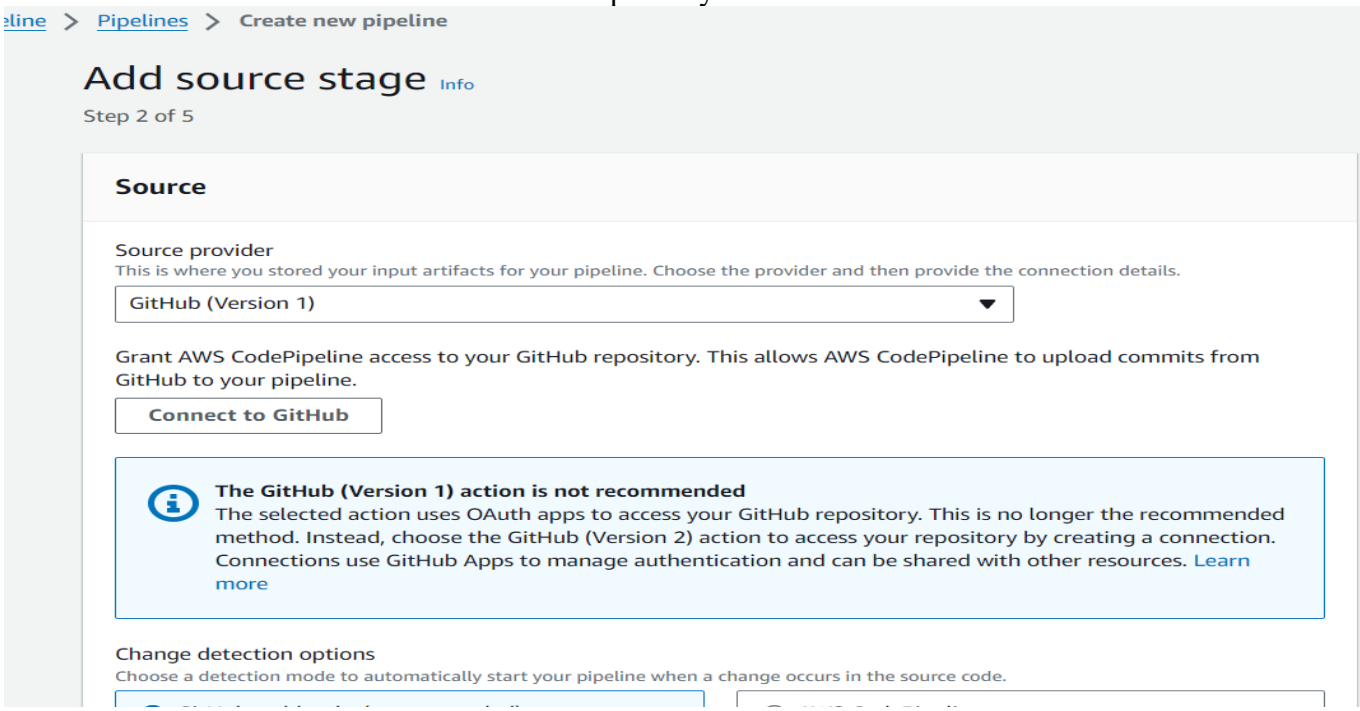
Komal Sabale
D15C-45

Give a name to your pipeline. Here I have given name as pipeline1.

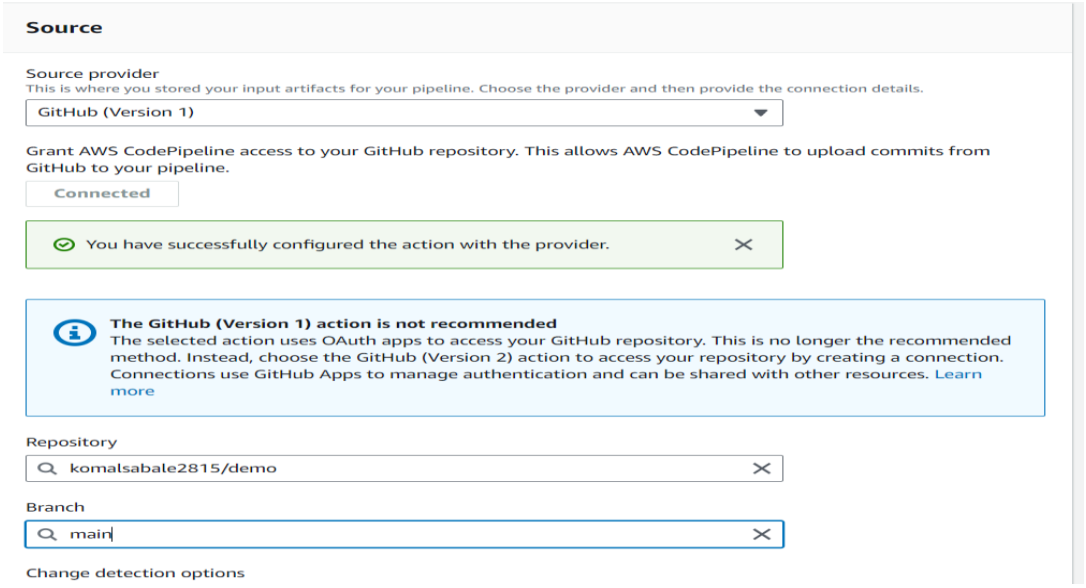


Komal Sabale
D15C-45

2. In the source stage, choose GitHub v2 as the provider, then connect your GitHub account to AWS by creating a connection. You'd need your GitHub credentials and then you'd need to authorize and install AWS on the forked GitHub Repository.

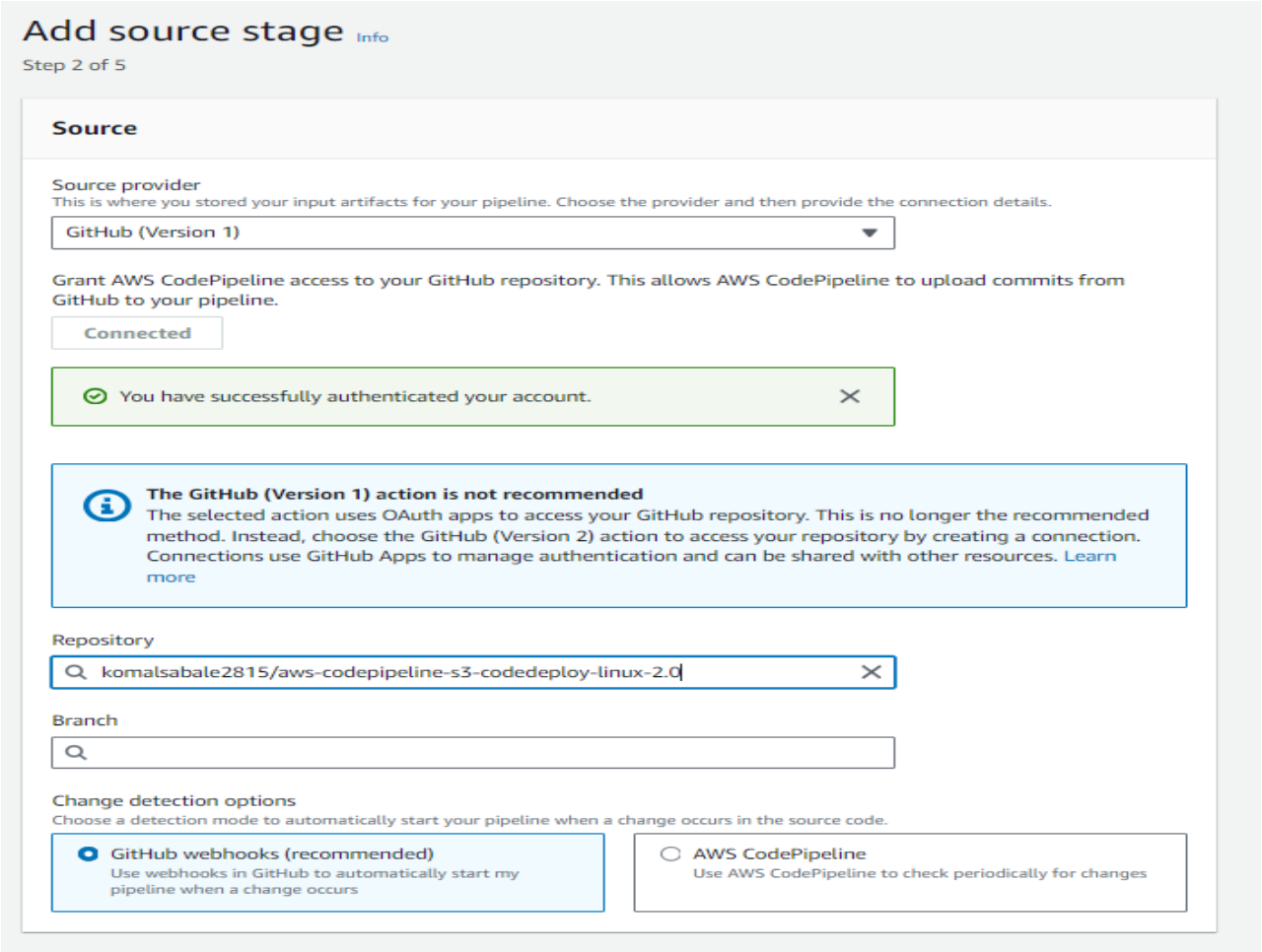


3.



Komal Sabale
D15C-45

4. select the forked repository then select the master branch.



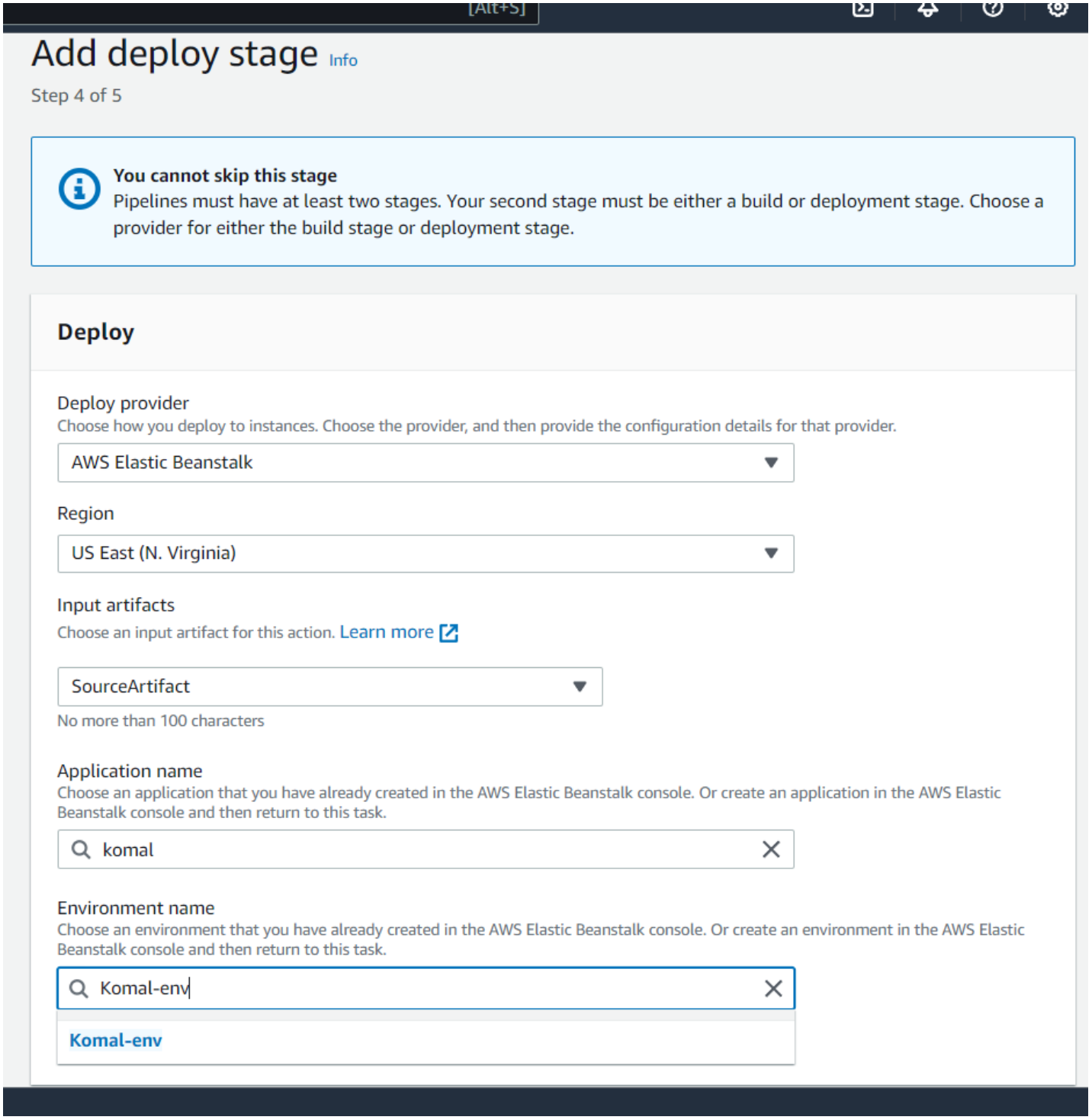
5. Then select trigger type none.

After that, click Continue and skip the build stage. Proceed to the Deployment stage.

Step 5: Deployment

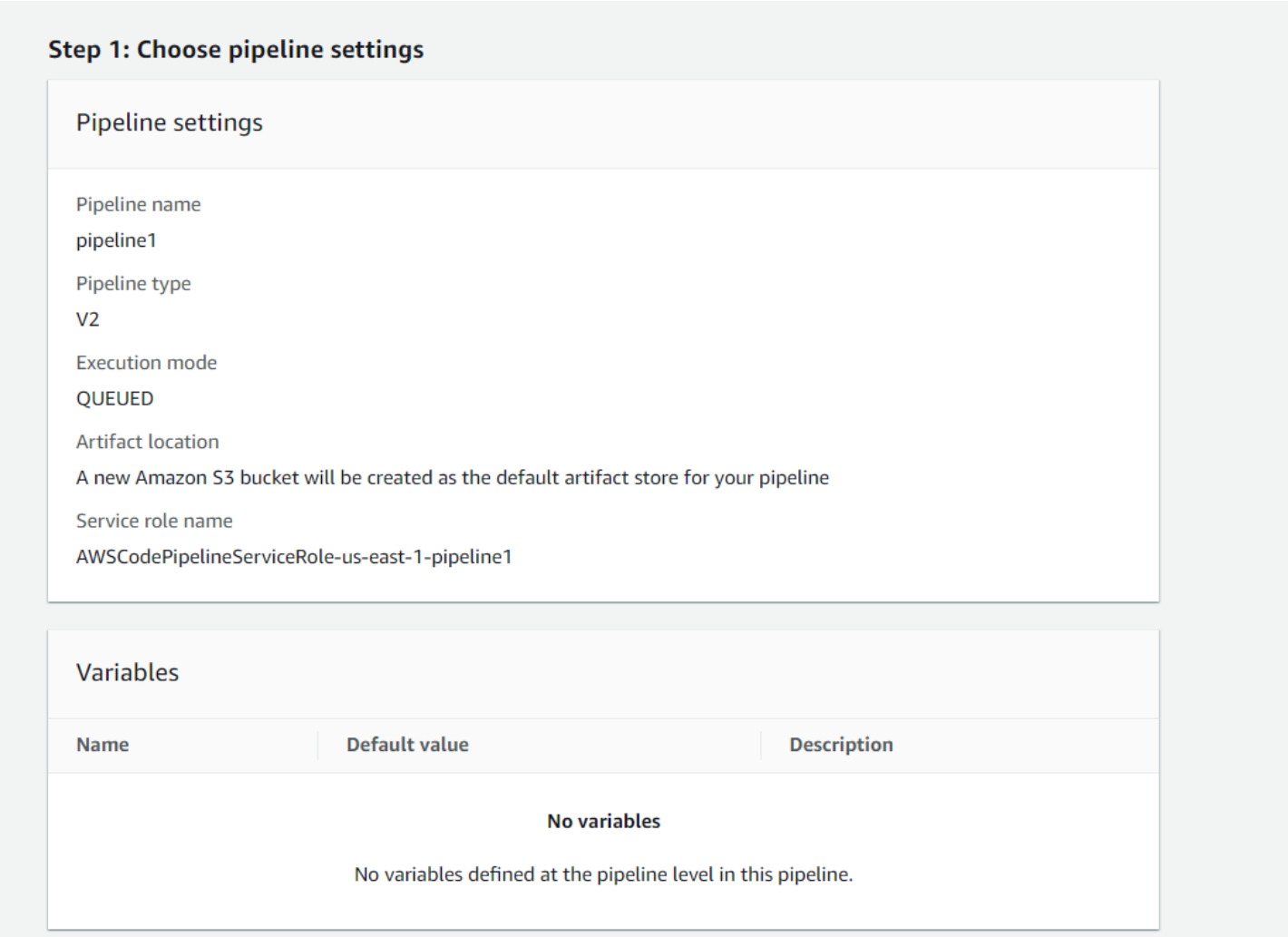
1. Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name.

Komal Sabale
D15C-45

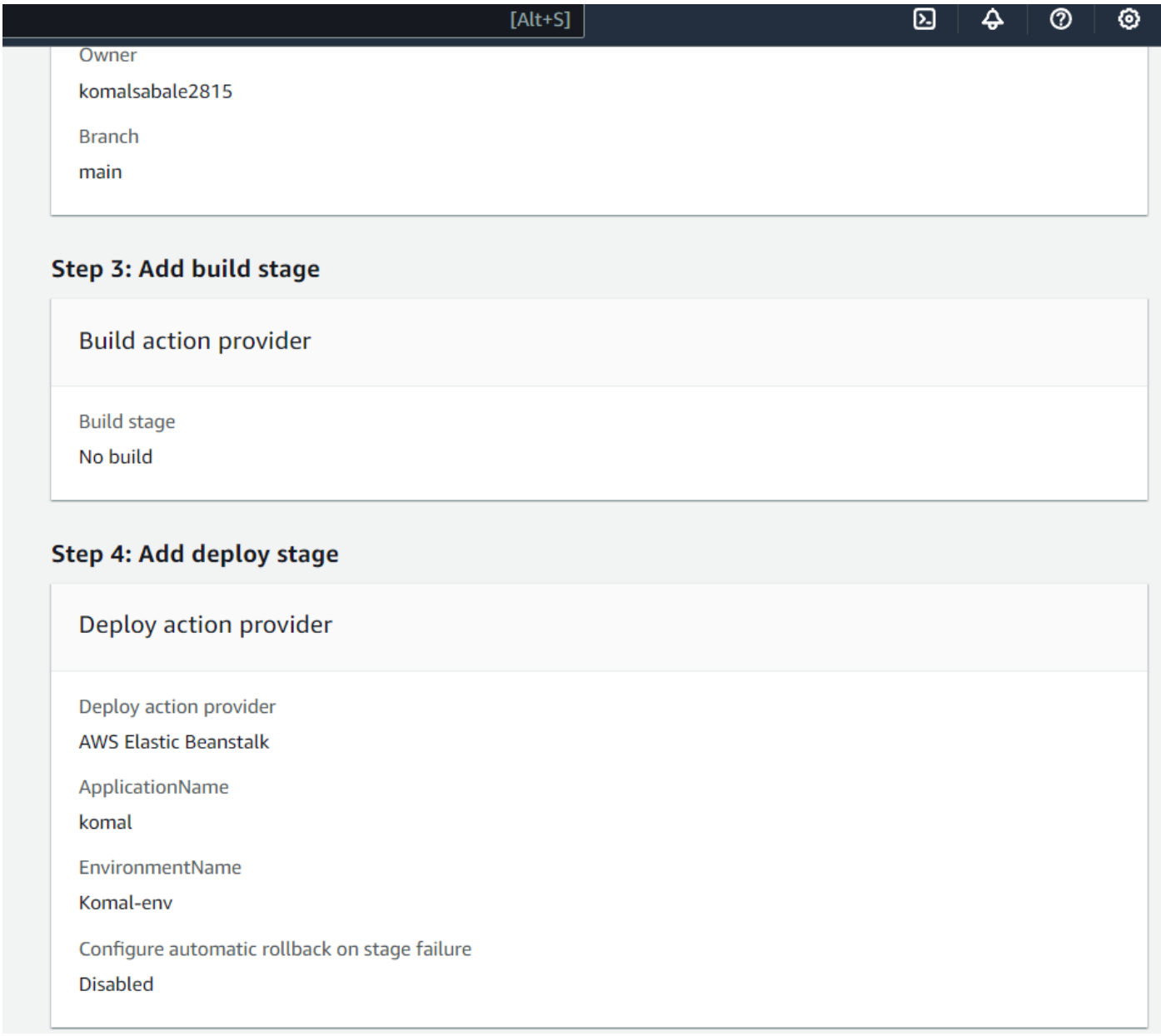


Komal Sabale
D15C-45

2.Click Next, Review and create the pipeline.

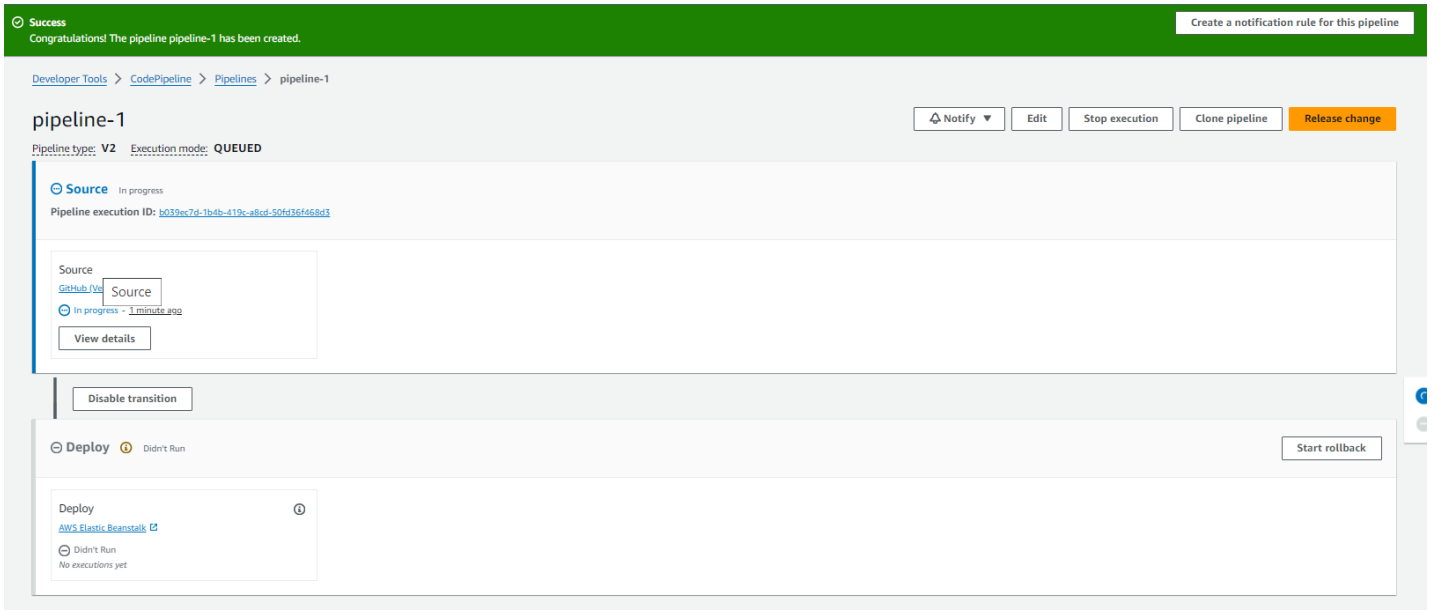


Komal Sabale
D15C-45

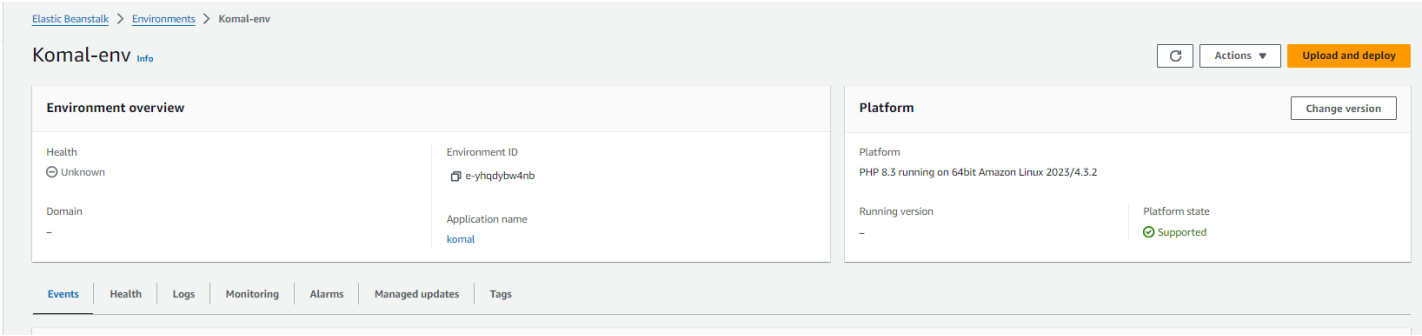


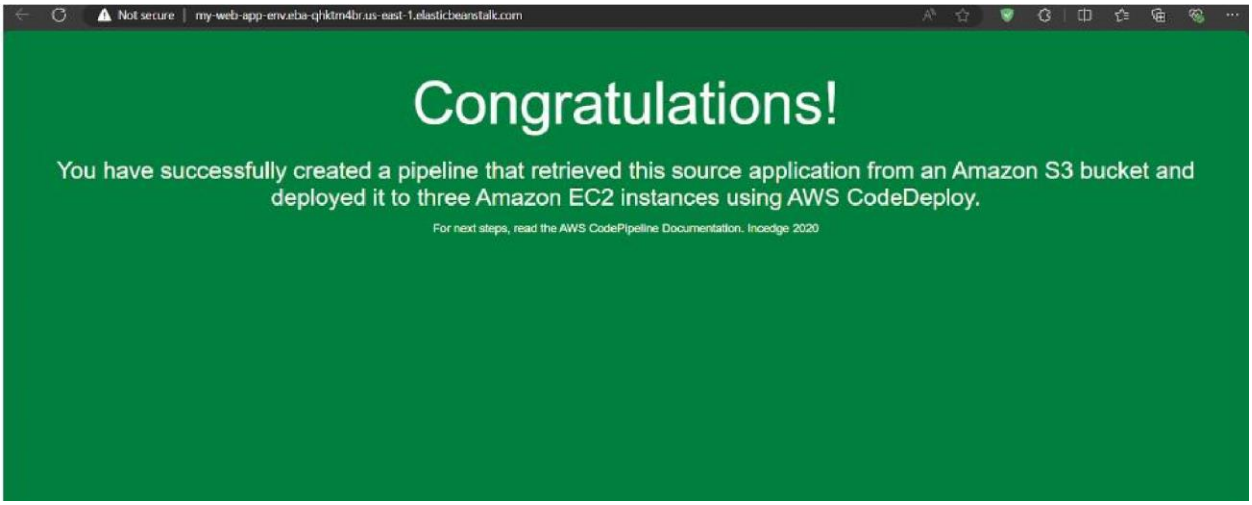
Komal Sabale
D15C-45

3. Then it will give you this result on screen. i.e. deployed successfully.



4. In a few minutes the website will get hosted successfully. Then click on the url present over the environment created on Elastic Benstalk.



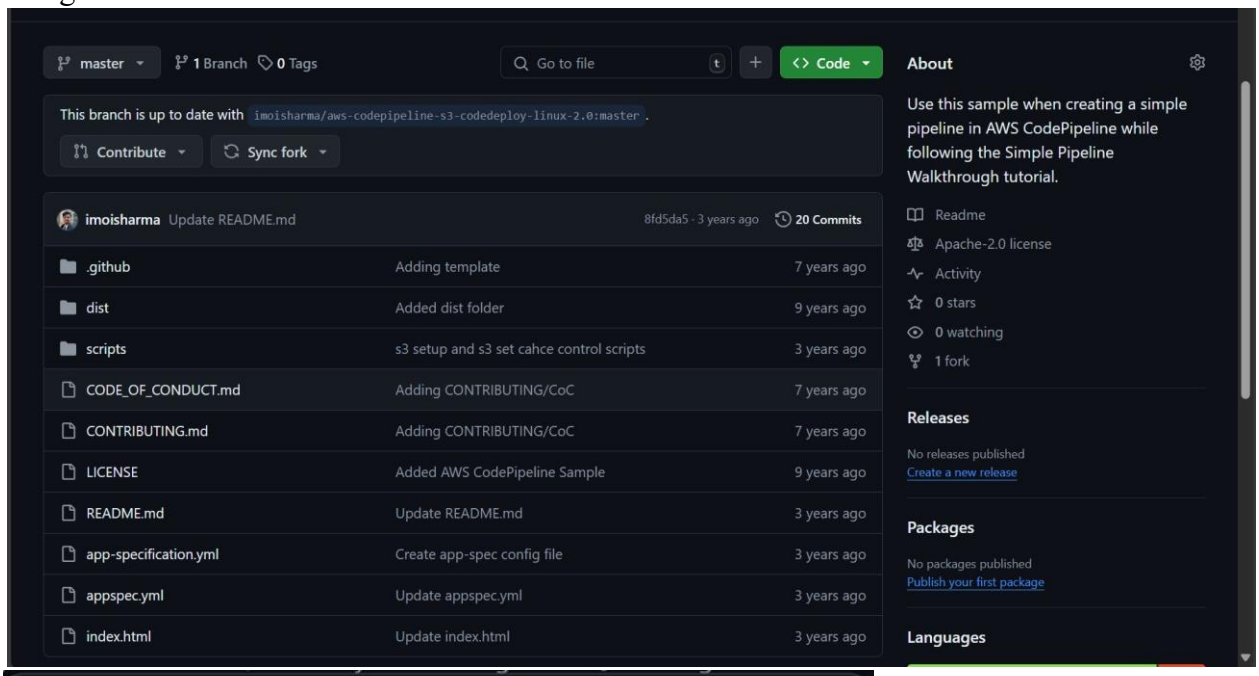


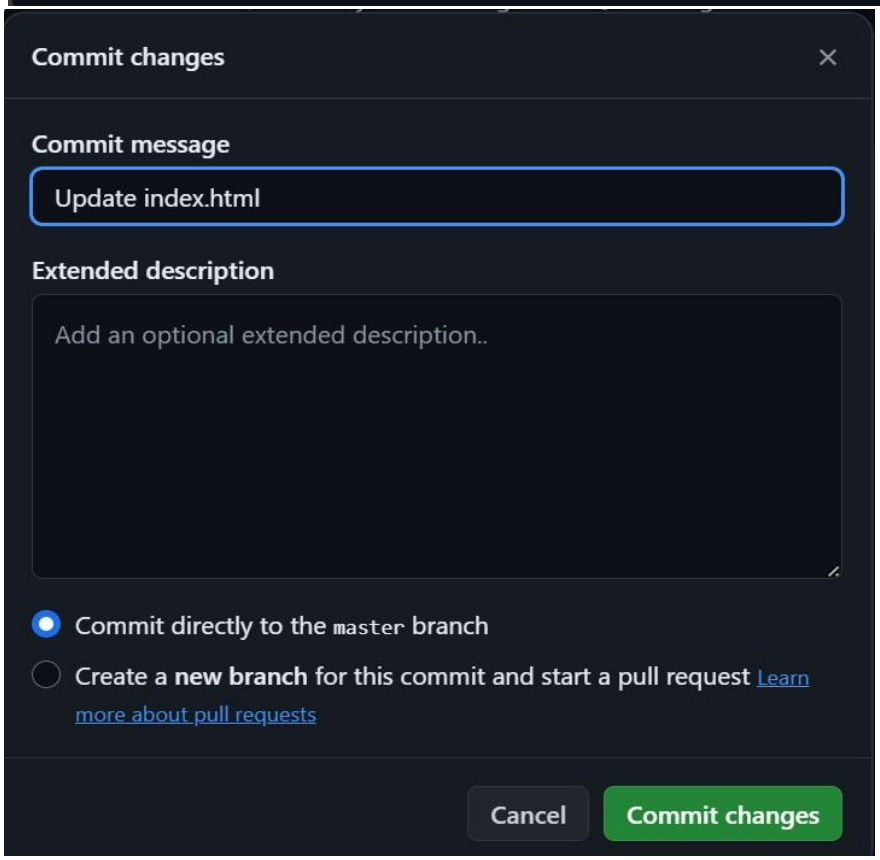
If you can see this, that means that you successfully created an automated software using CodePipeline.

Komal Sabale
D15C-45

Step 6: Committing changes to update app

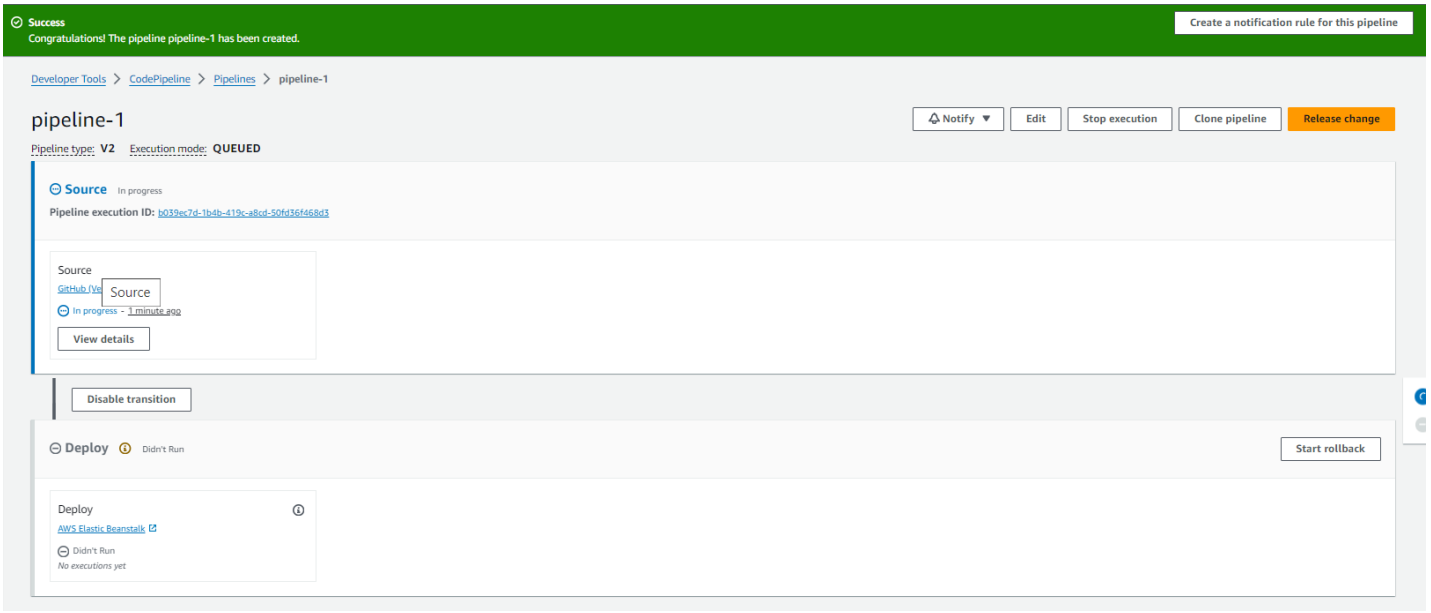
1. In this we make some changes in the file.Open github.com then open the forked repository. Then update the changes in the index.html file and finally commit those changes.





2. Then again start the deployment of the pipeline.

Komal Sabale
D15C-45



3. Check the changes in the website , here I have added a message in h3 tag.