Name: Alok Yadav Div: D15C Roll No.: 59

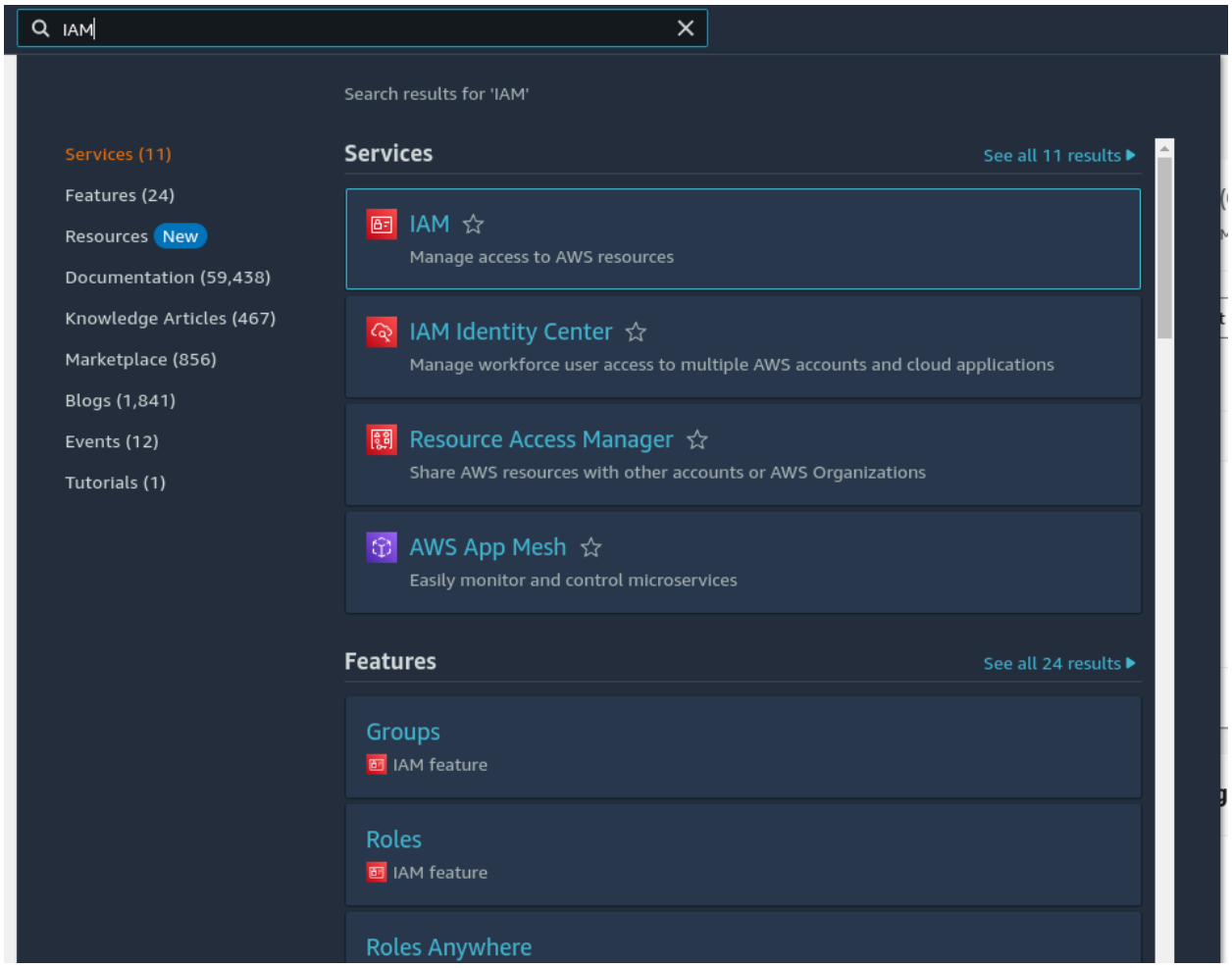
Experiment No. 2

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.

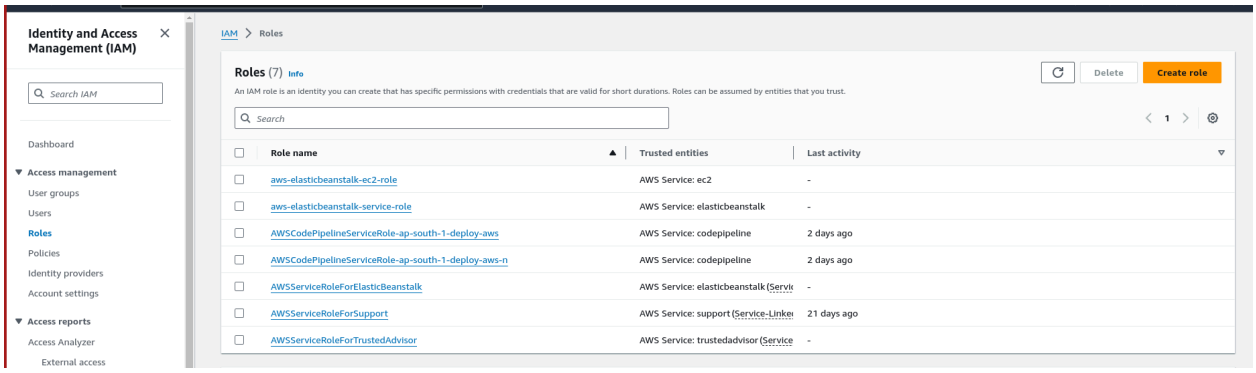
Steps:

Initially we will create a new Role in IAM

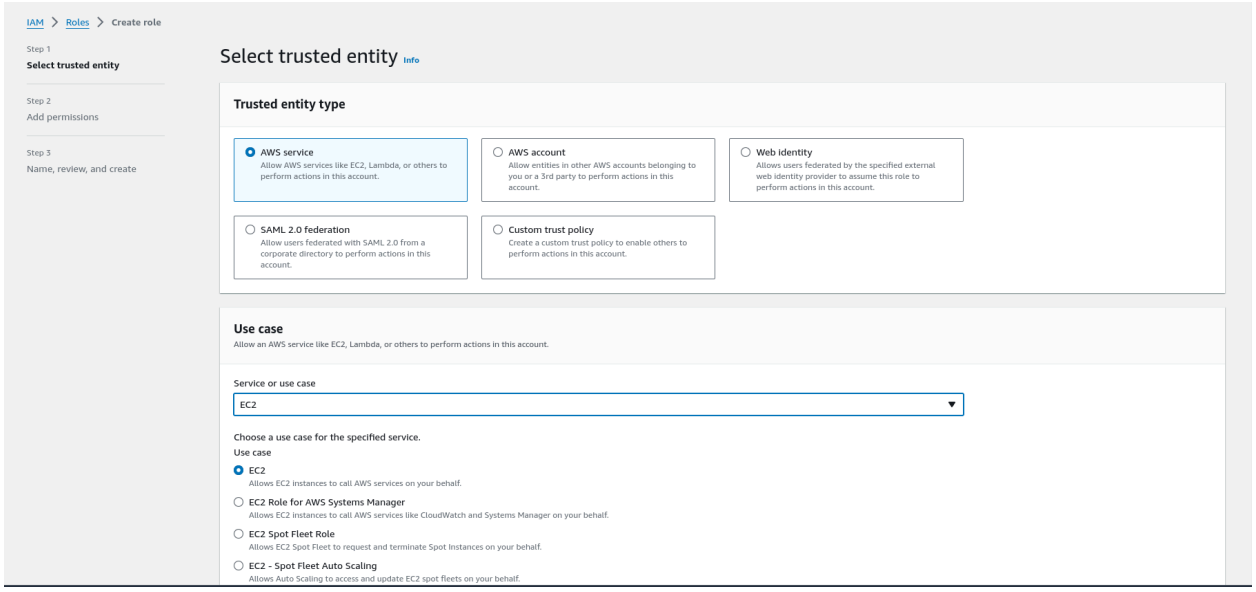
1. Search for IAM in search box

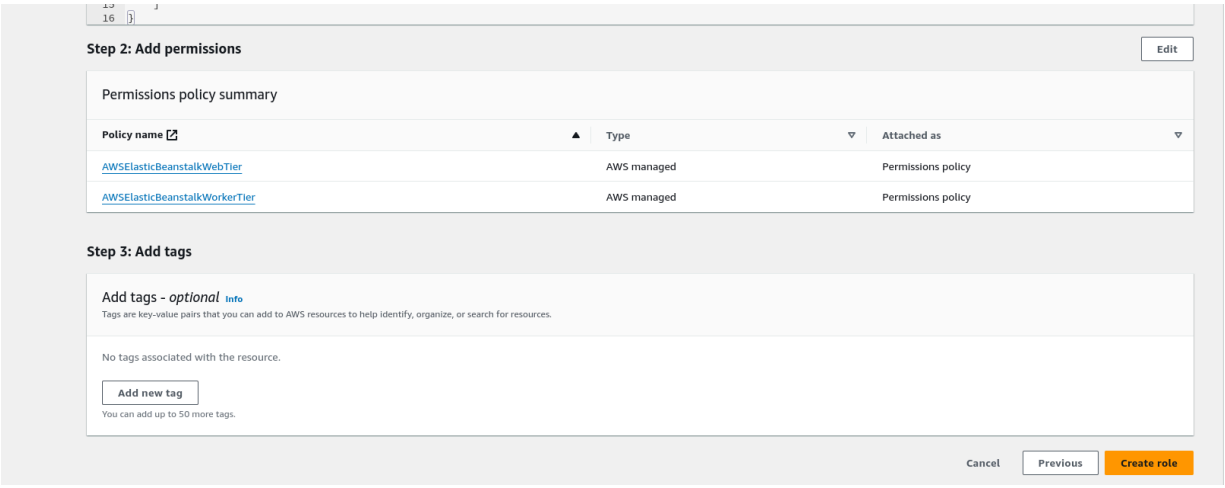


2. Go to Roles, and select AWSElasticBeanstalkWebtier and AWSElasticBeanstalkWorkertier, and create a role

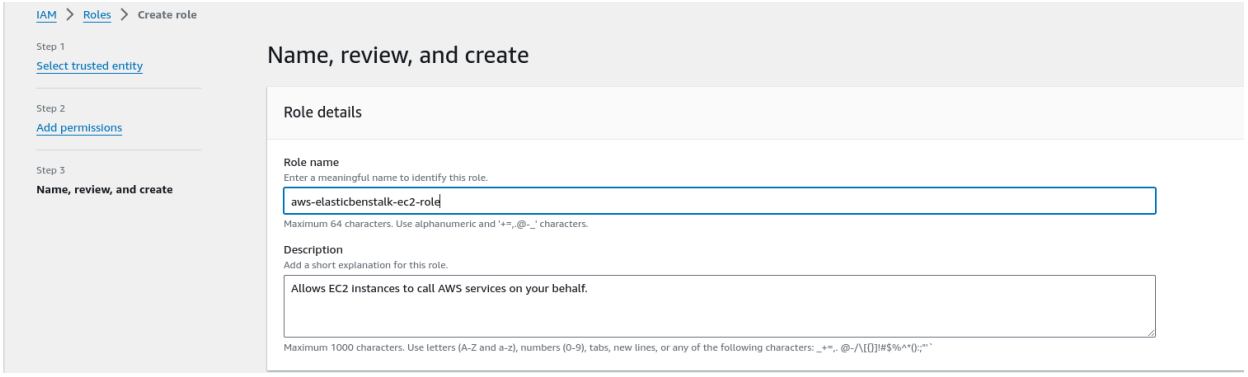


3. Select entity type as **AWS Service**

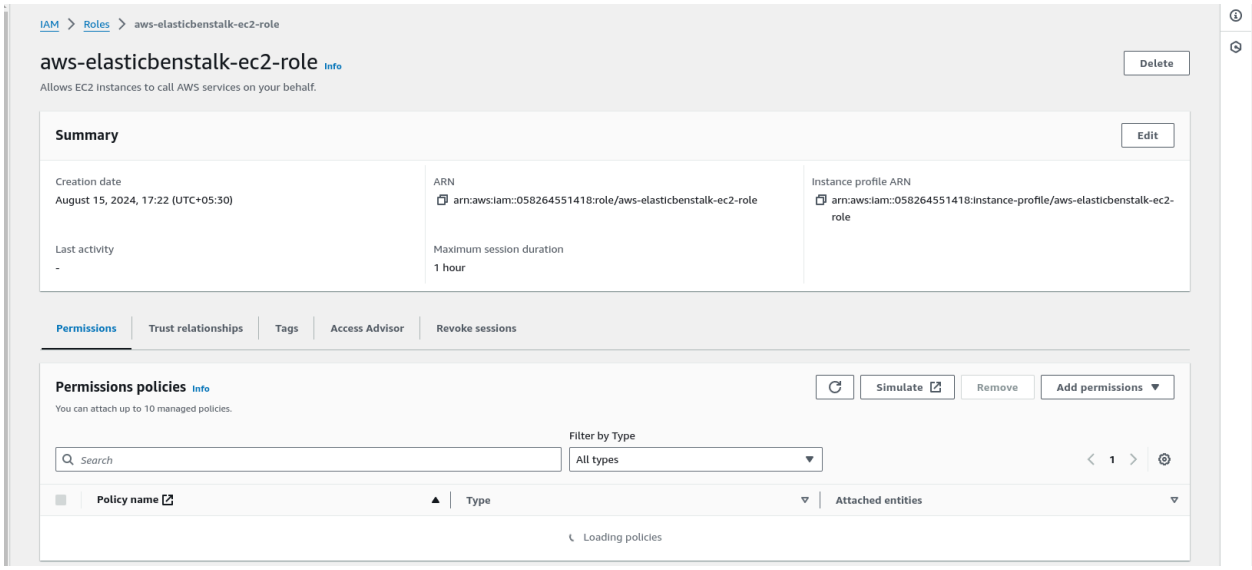




4. Give name to newly created role such as **aws-elasticbenstalk-ec2-role**

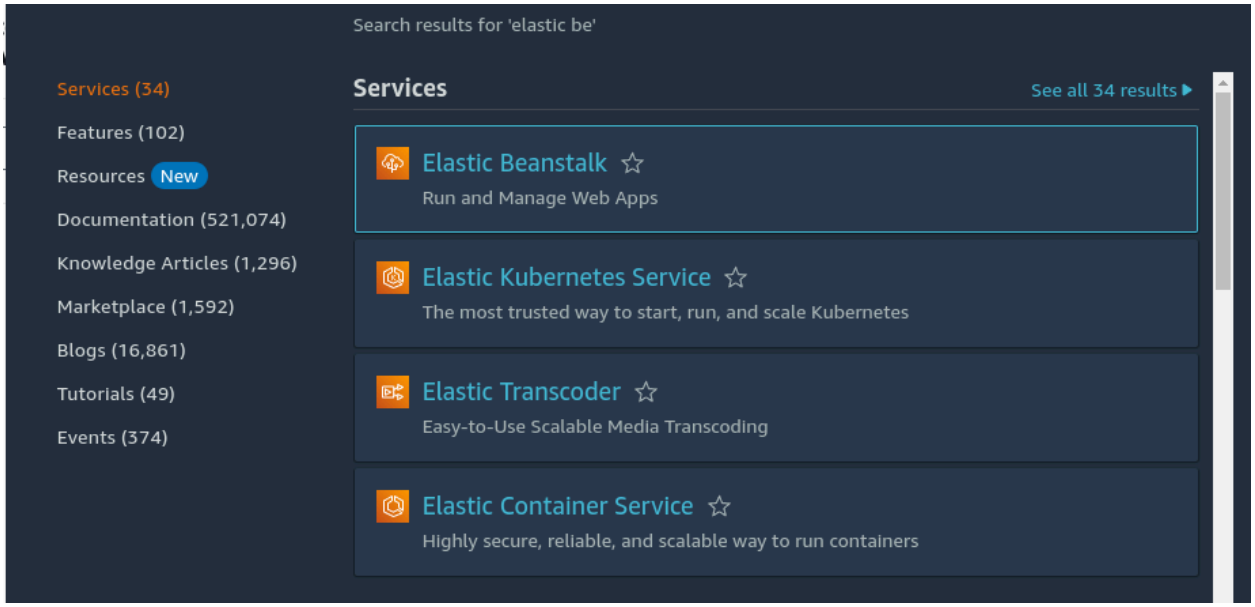


5. Summary of newly created role

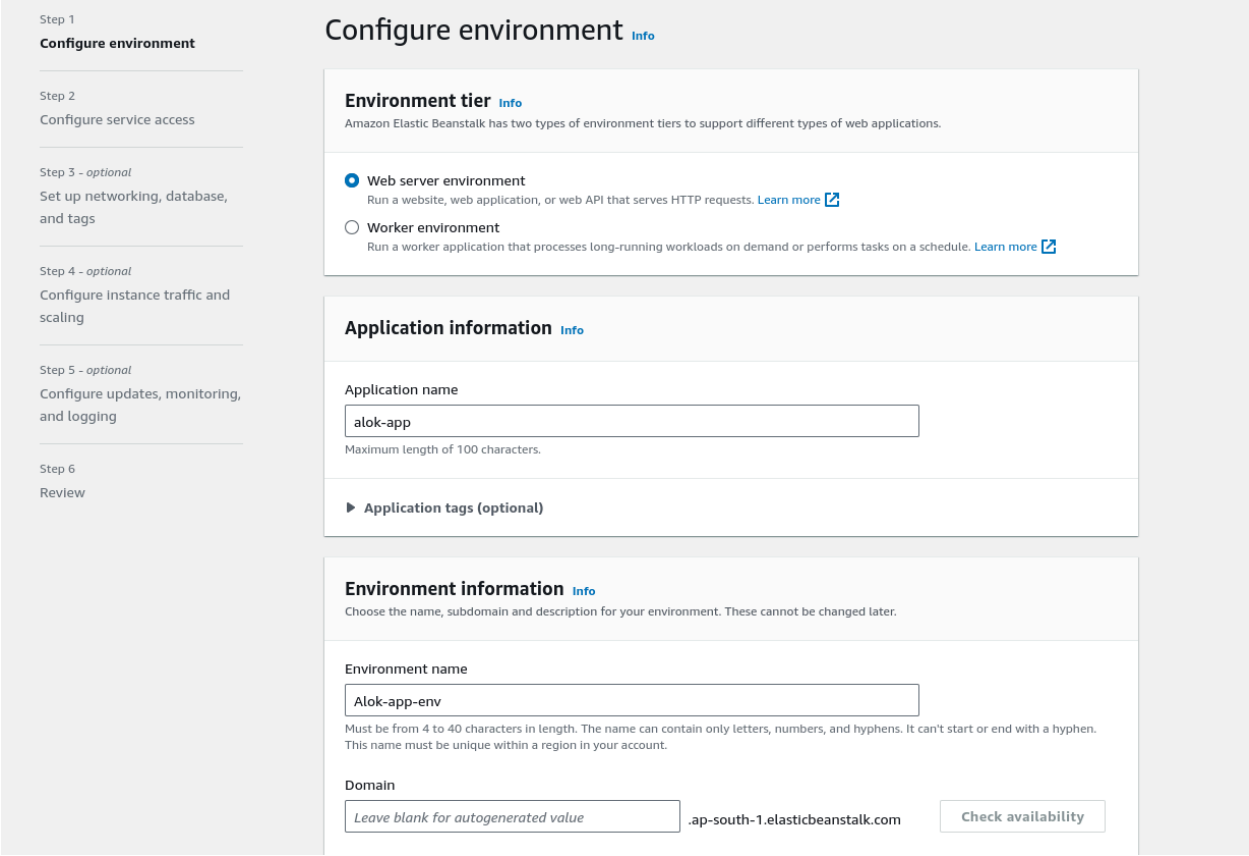


Now we have to create Elastic Beanstalk Environment

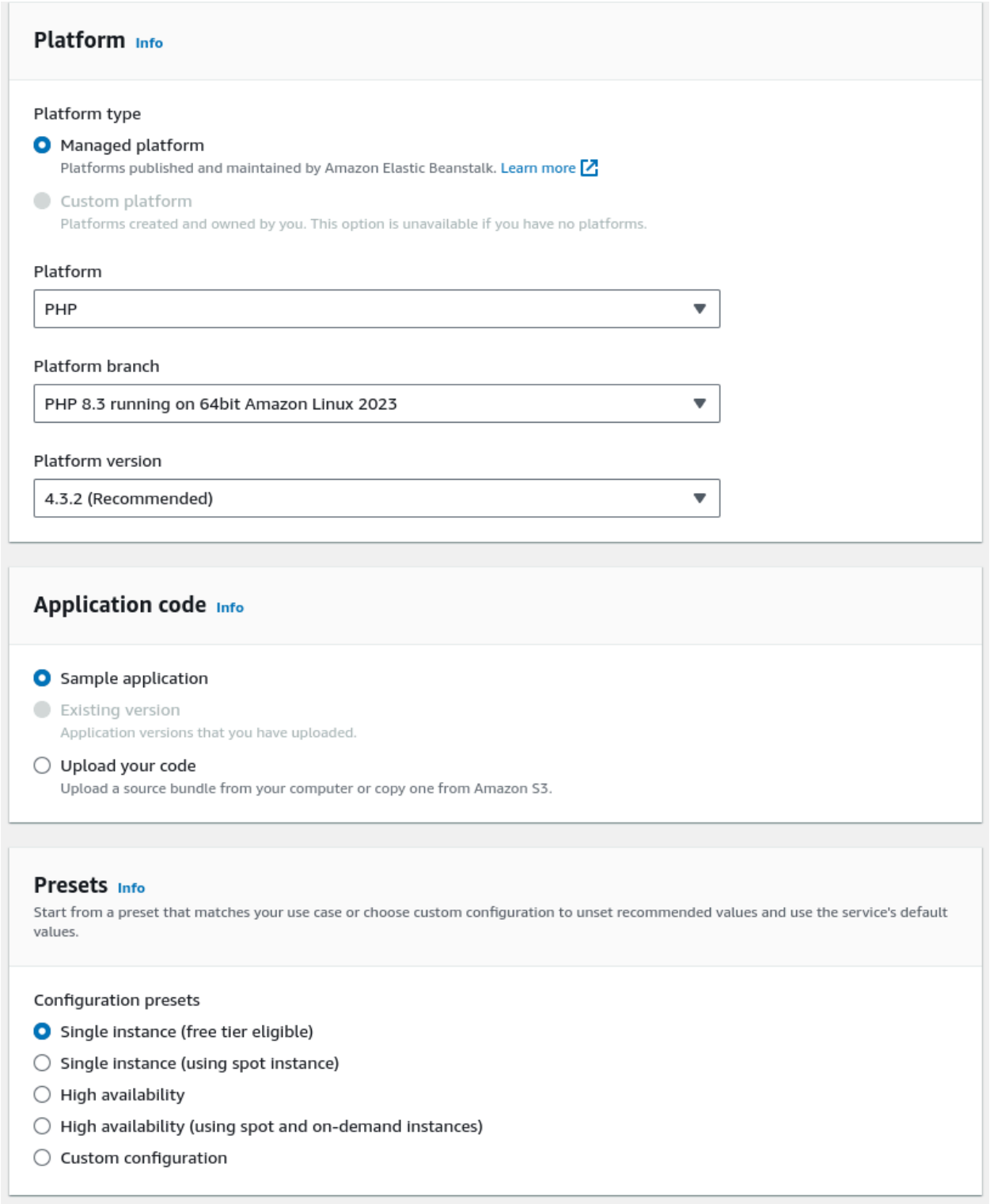
6. Search **Elastic Beanstalk** and proceed with it



7. Create a new Environment and name it.

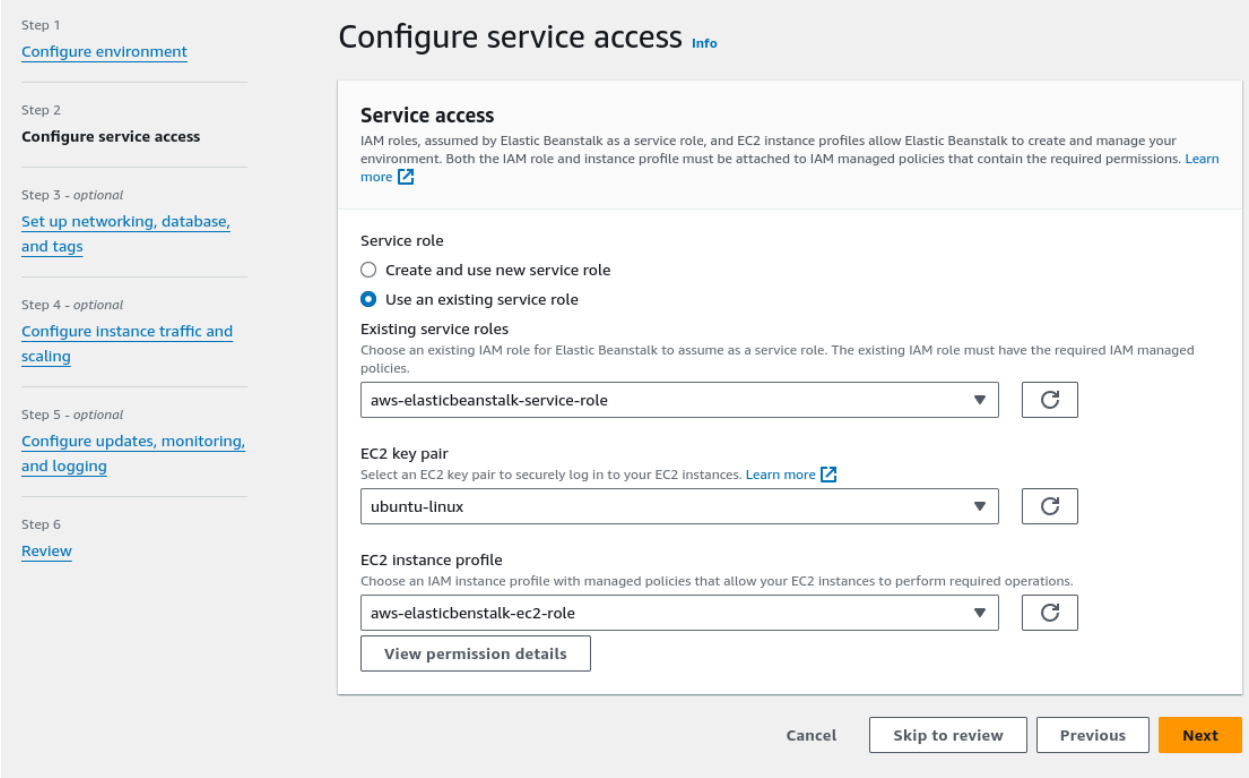


8. Select Platform as PHP, Application code as **Sample Application**, presets **Single Instance**



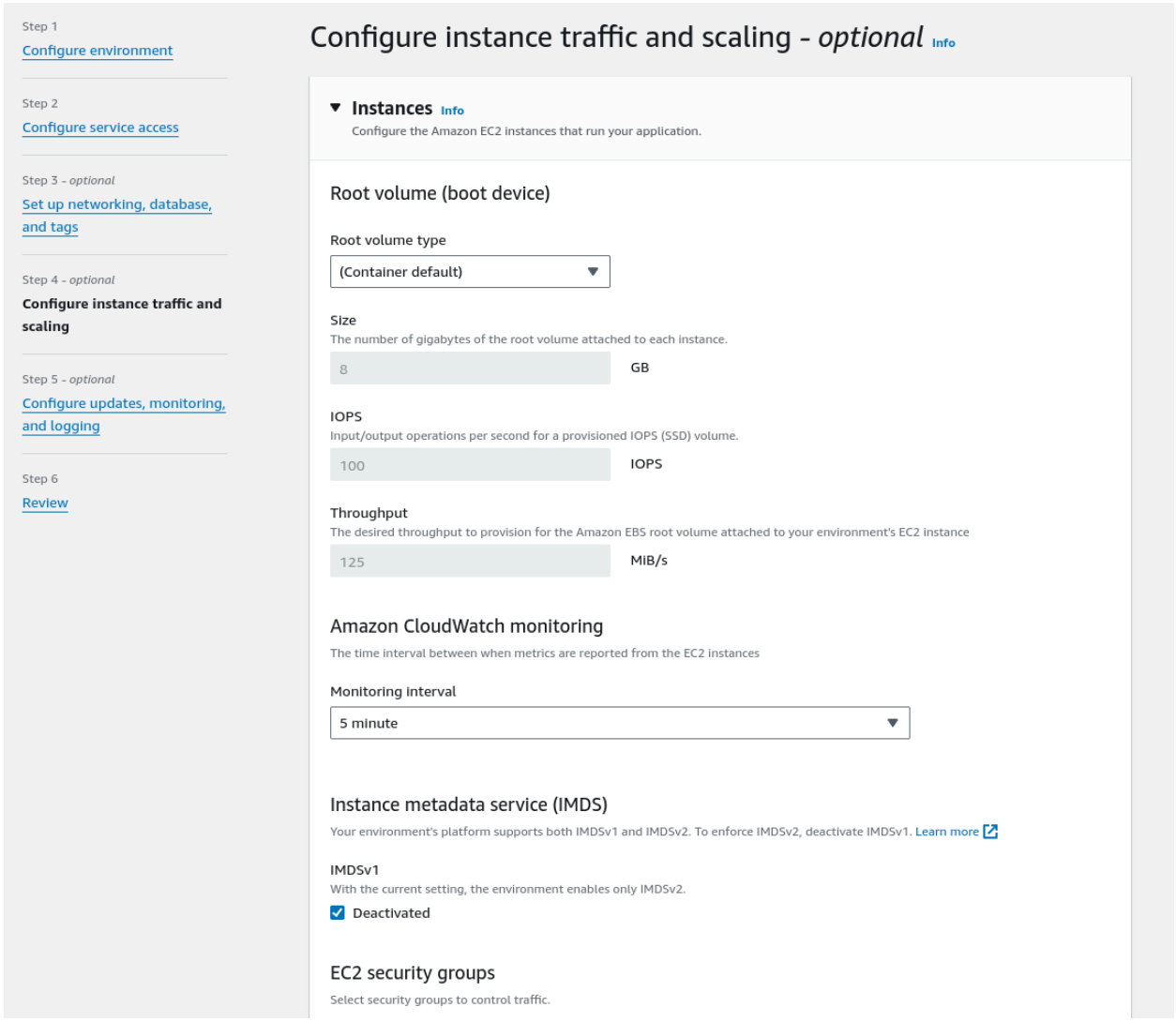
9. Under Service access settings, select **Use an existing service role**. Name service role as **aws-elasticbeanstalk-service-role** and EC2

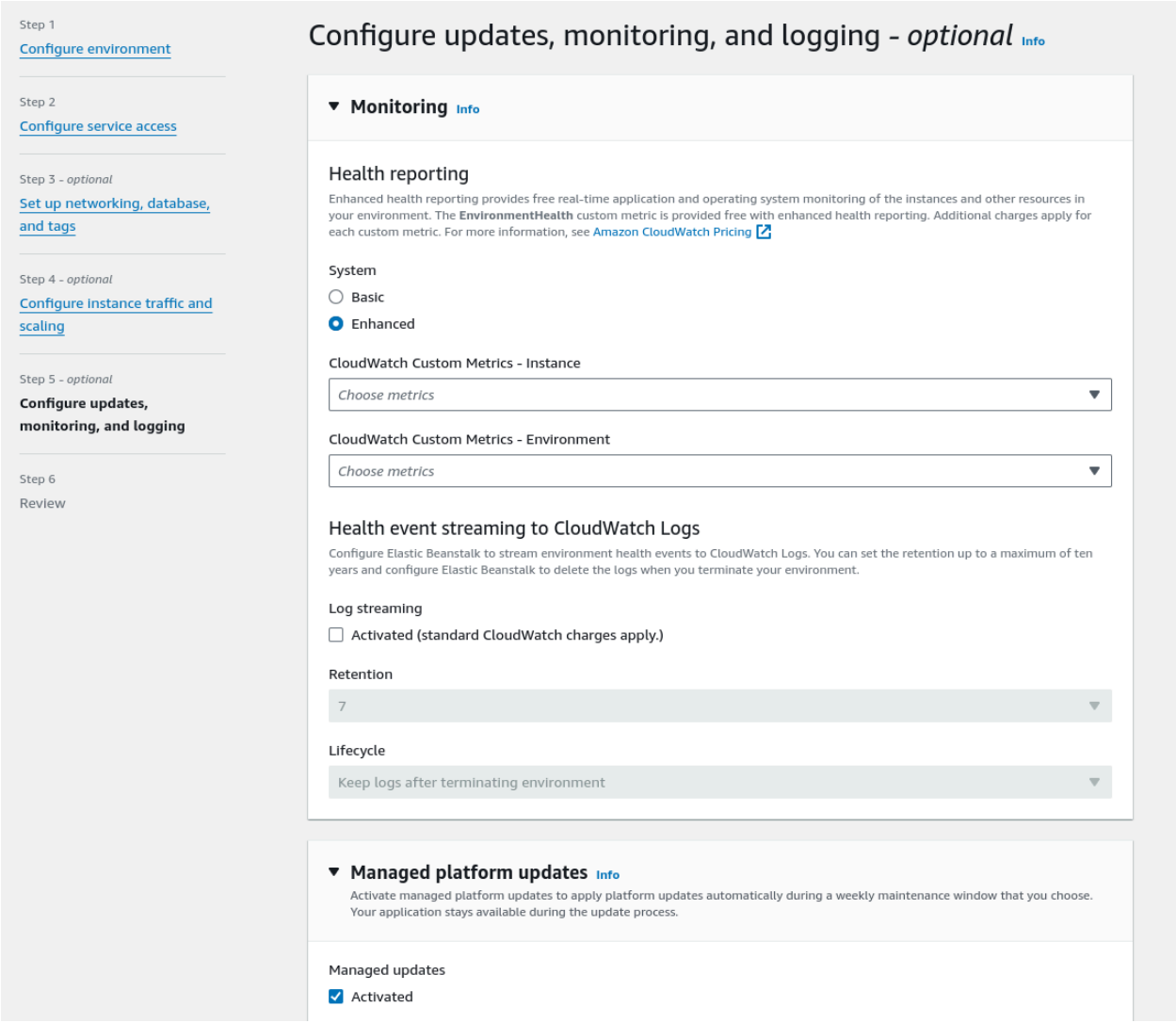
instance profile as **aws-elasticbenstalk-ec2-role** which we had created earlier



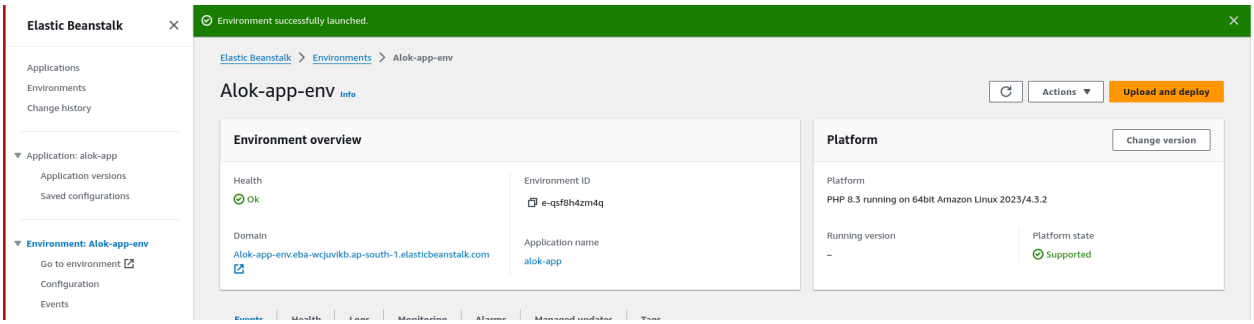
10. Keep below settings as it as

Name: Alok Yadav Div: D15C Roll No.: 59



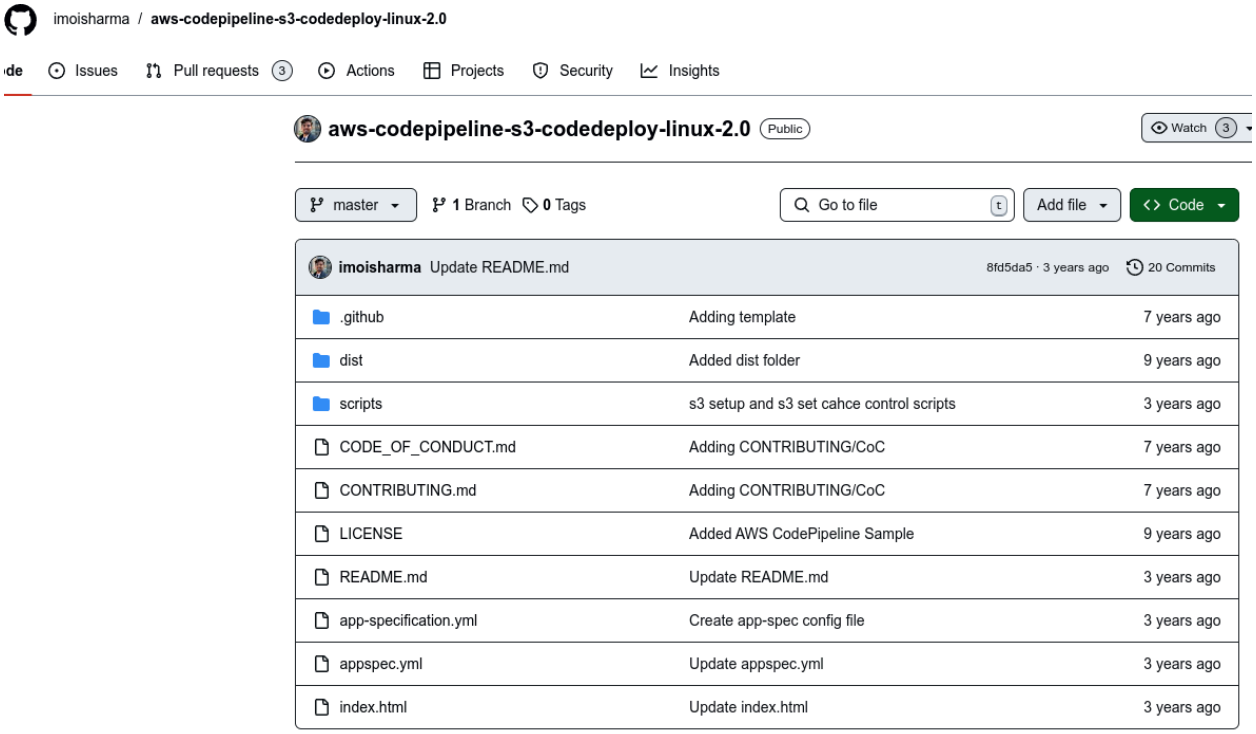


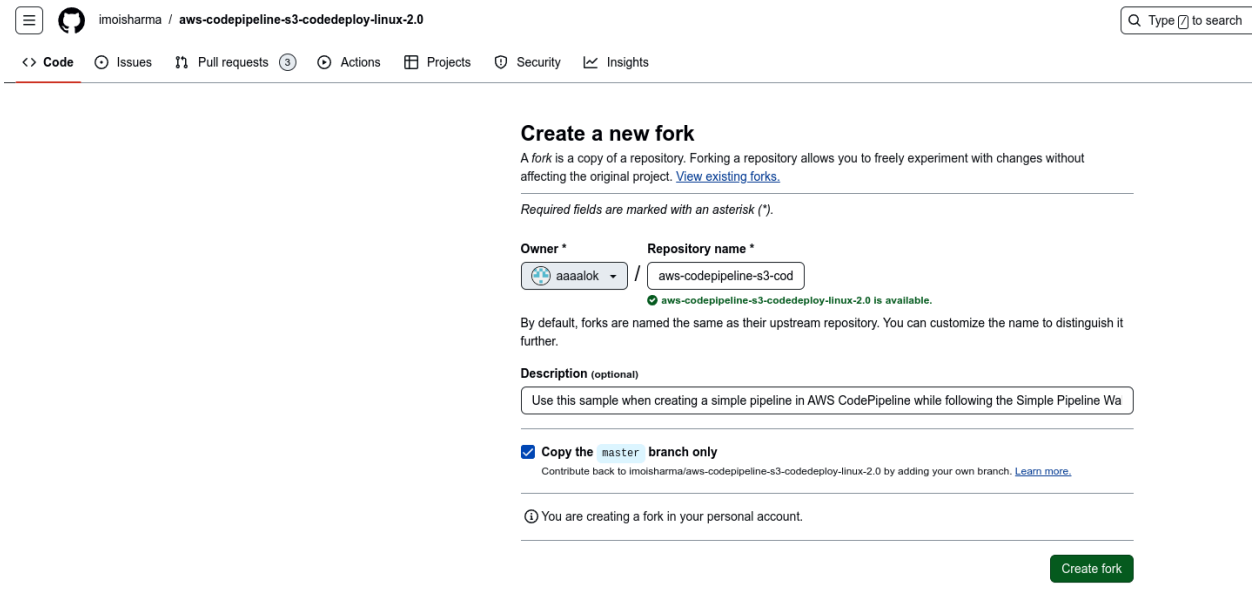
11. Our environment is created successfully



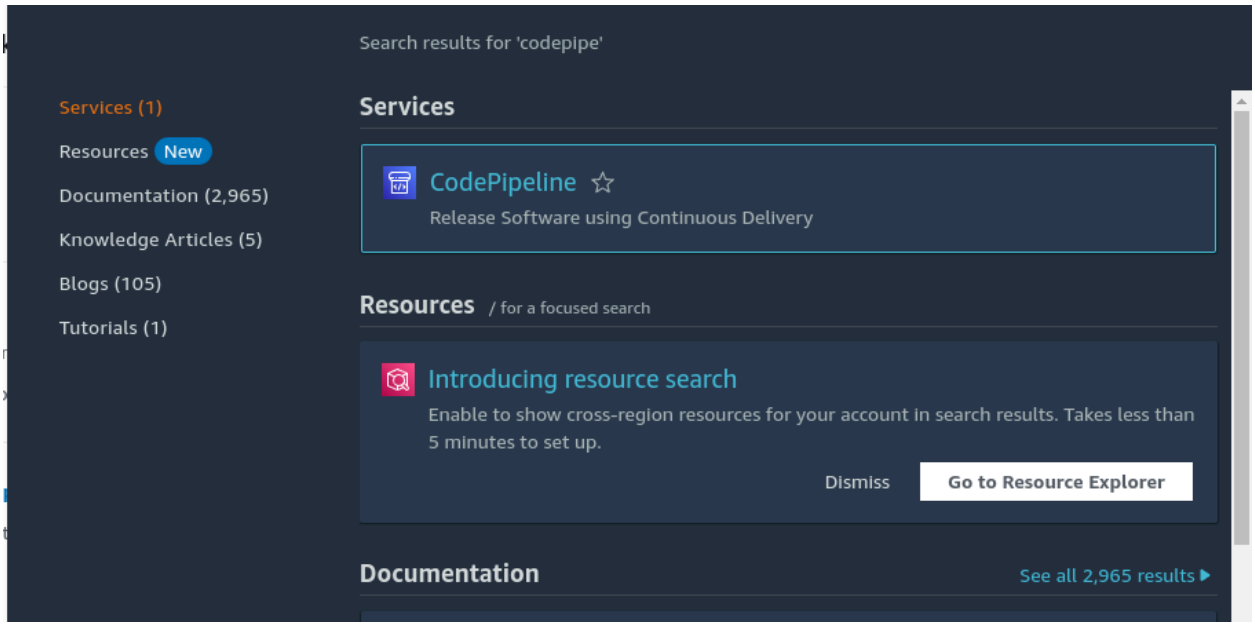
12. Now we would deploy a codepipeline. For below repo in your github

Fork this repository in your github.

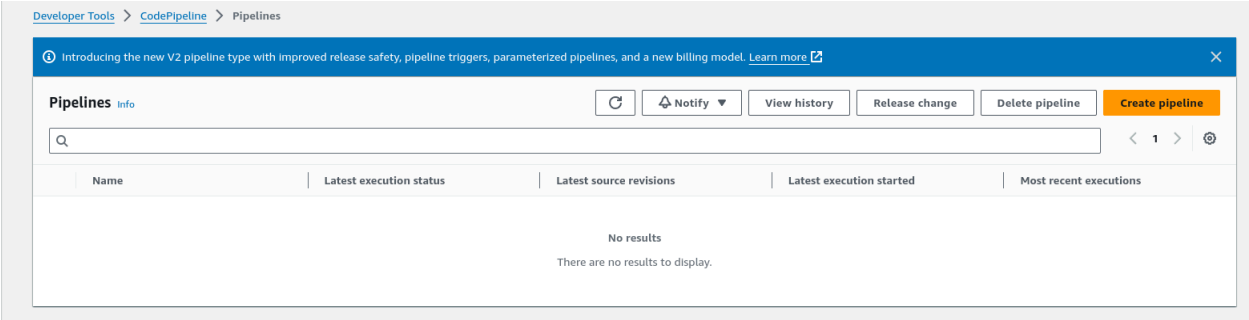




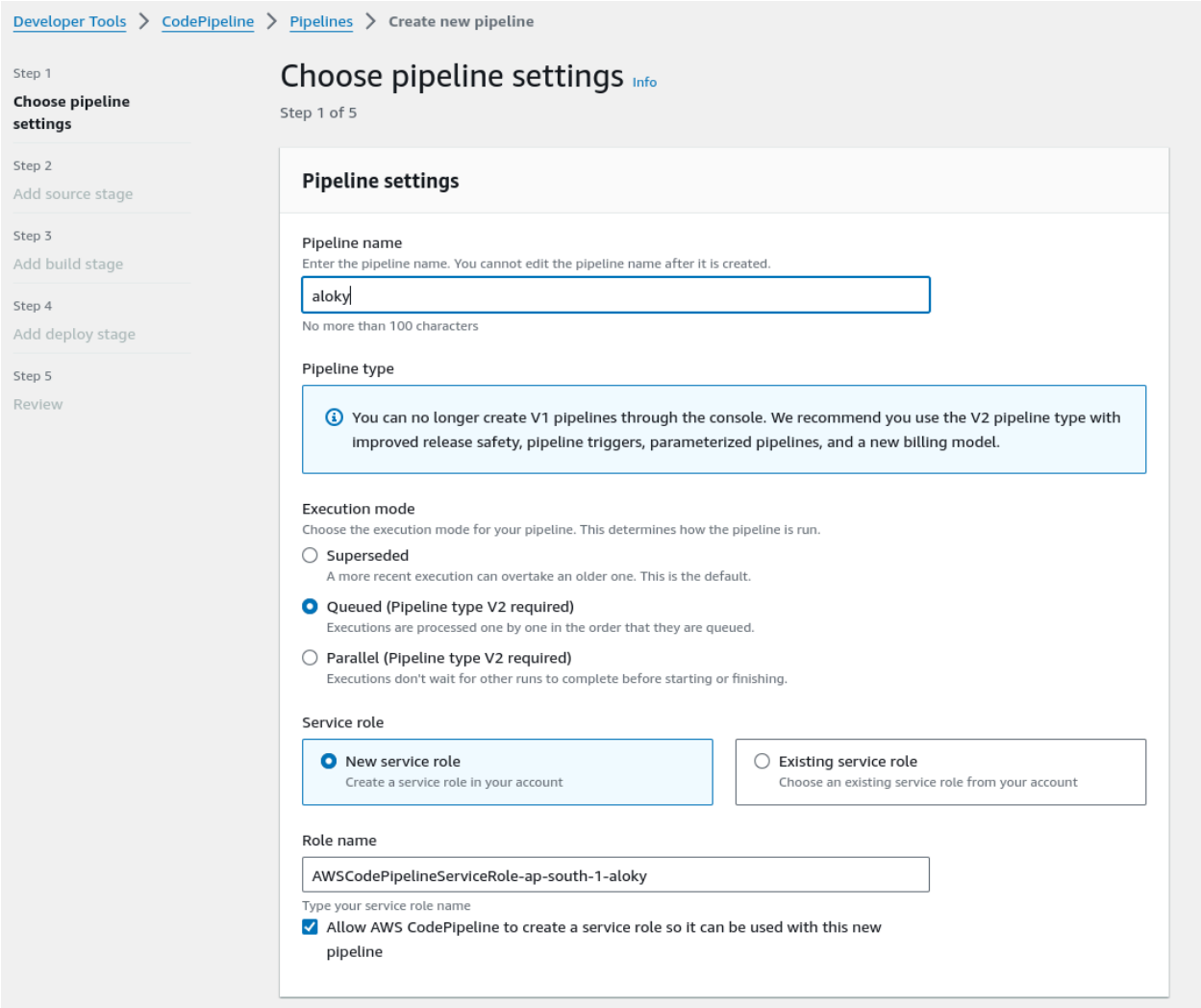
13. After forking, Now we would create a CodePipeline. Goto CodePipeline under services section



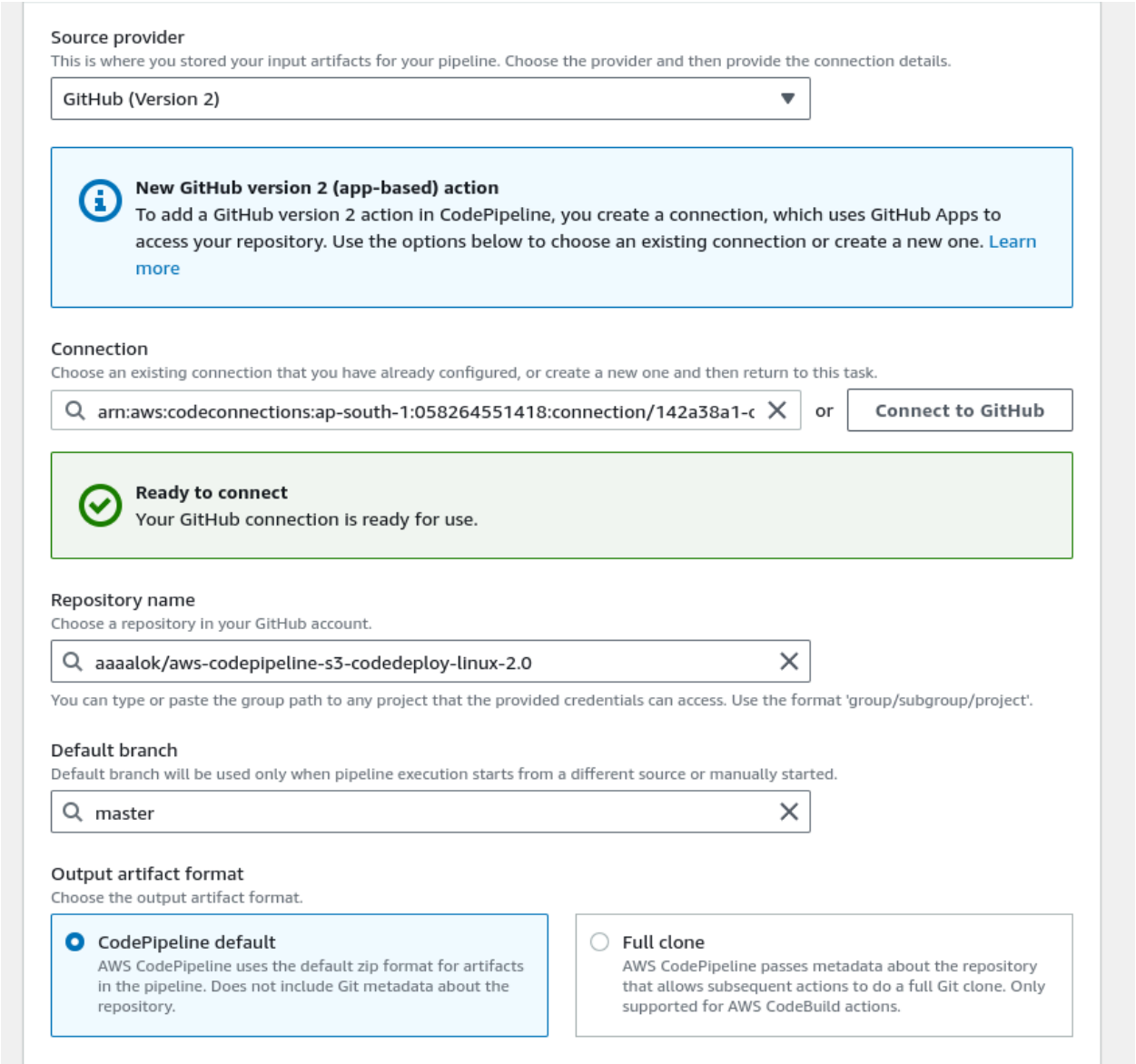
14. Create a new PipeLine



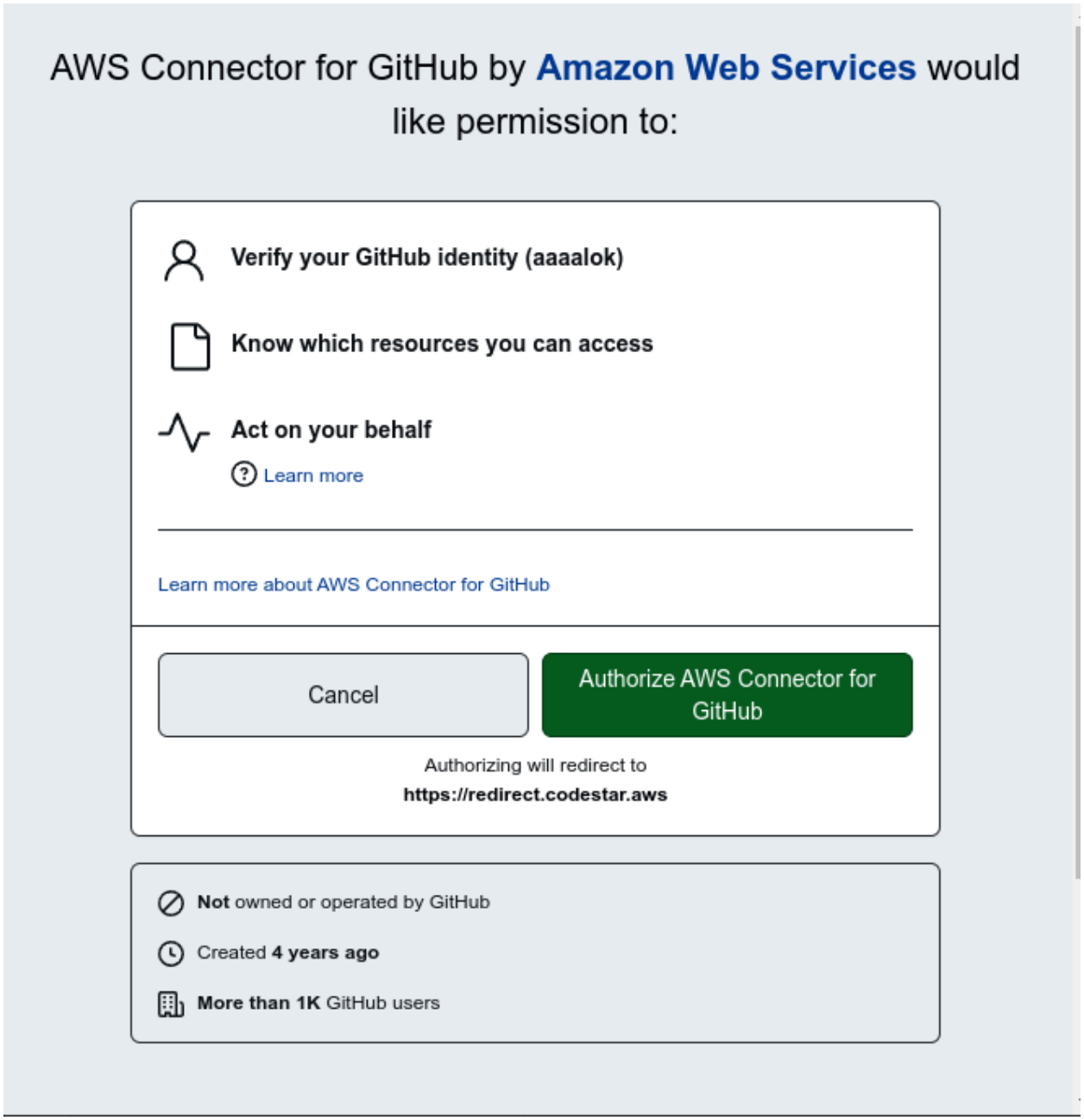
15. Name the pipeline leaving rest settings to its default



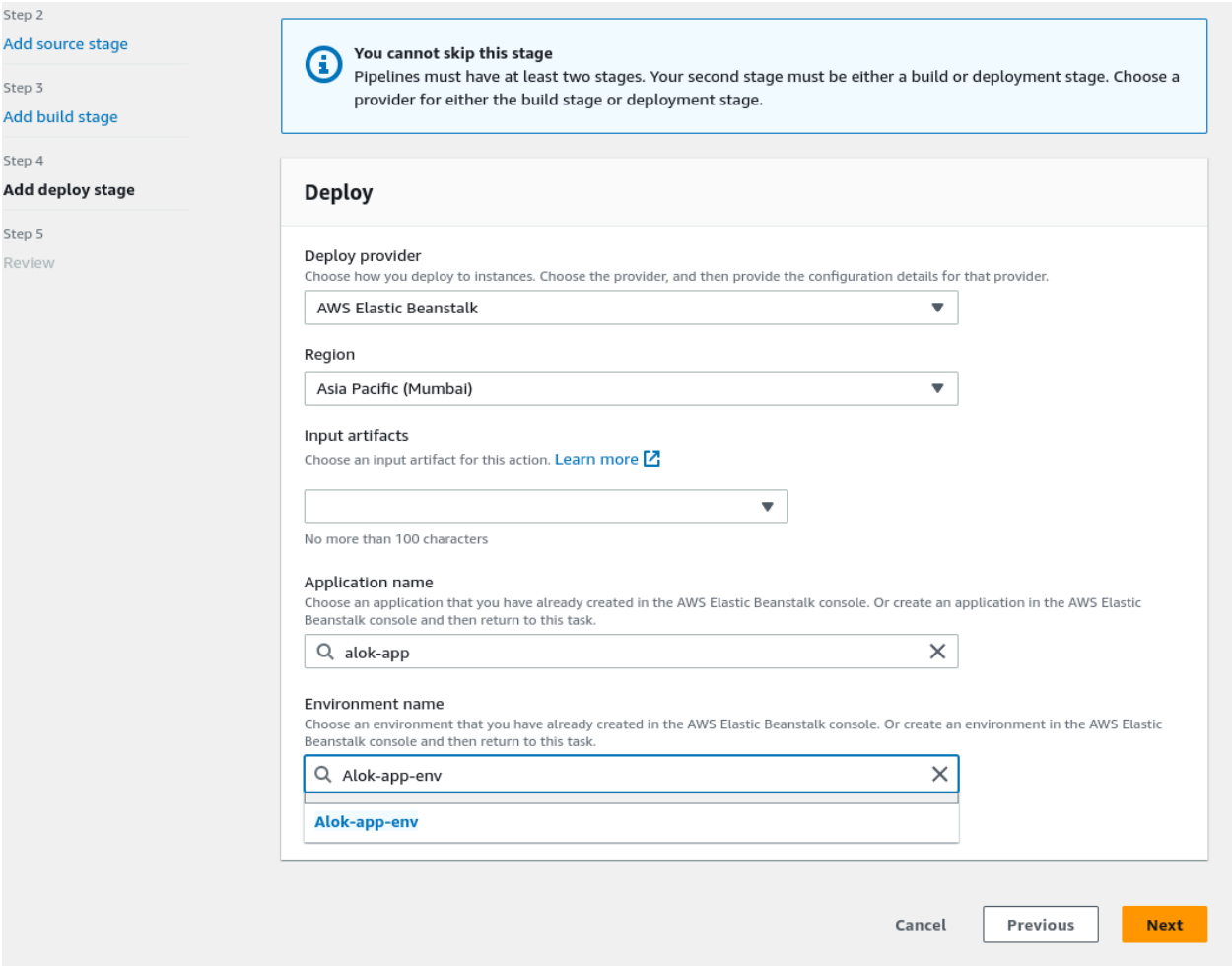
16. Under Source stage, select **Source Provider** as **GitHub (Version 2)**



17. Connect your GitHub account to AWS for it to build and deploy and track changes on repo

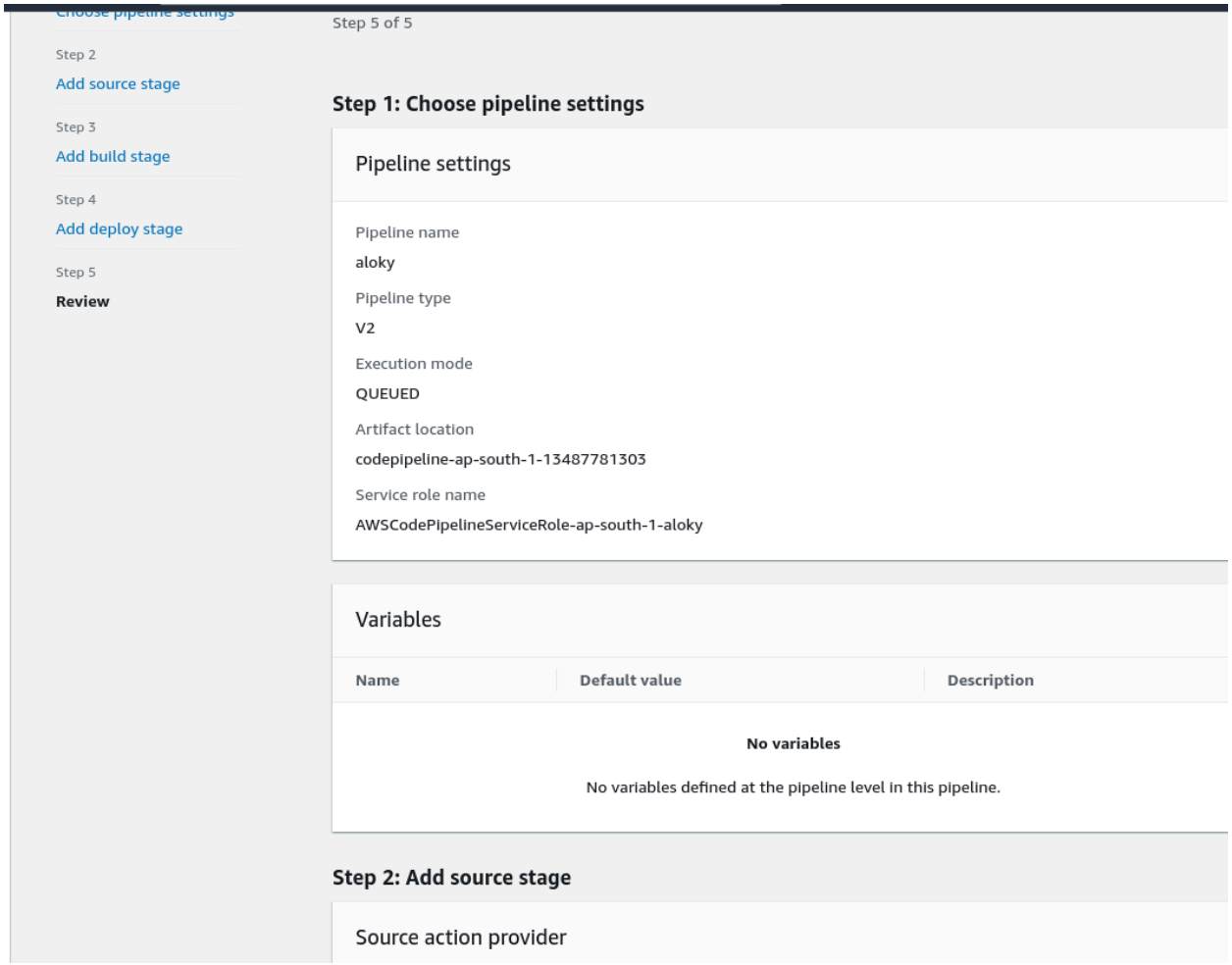


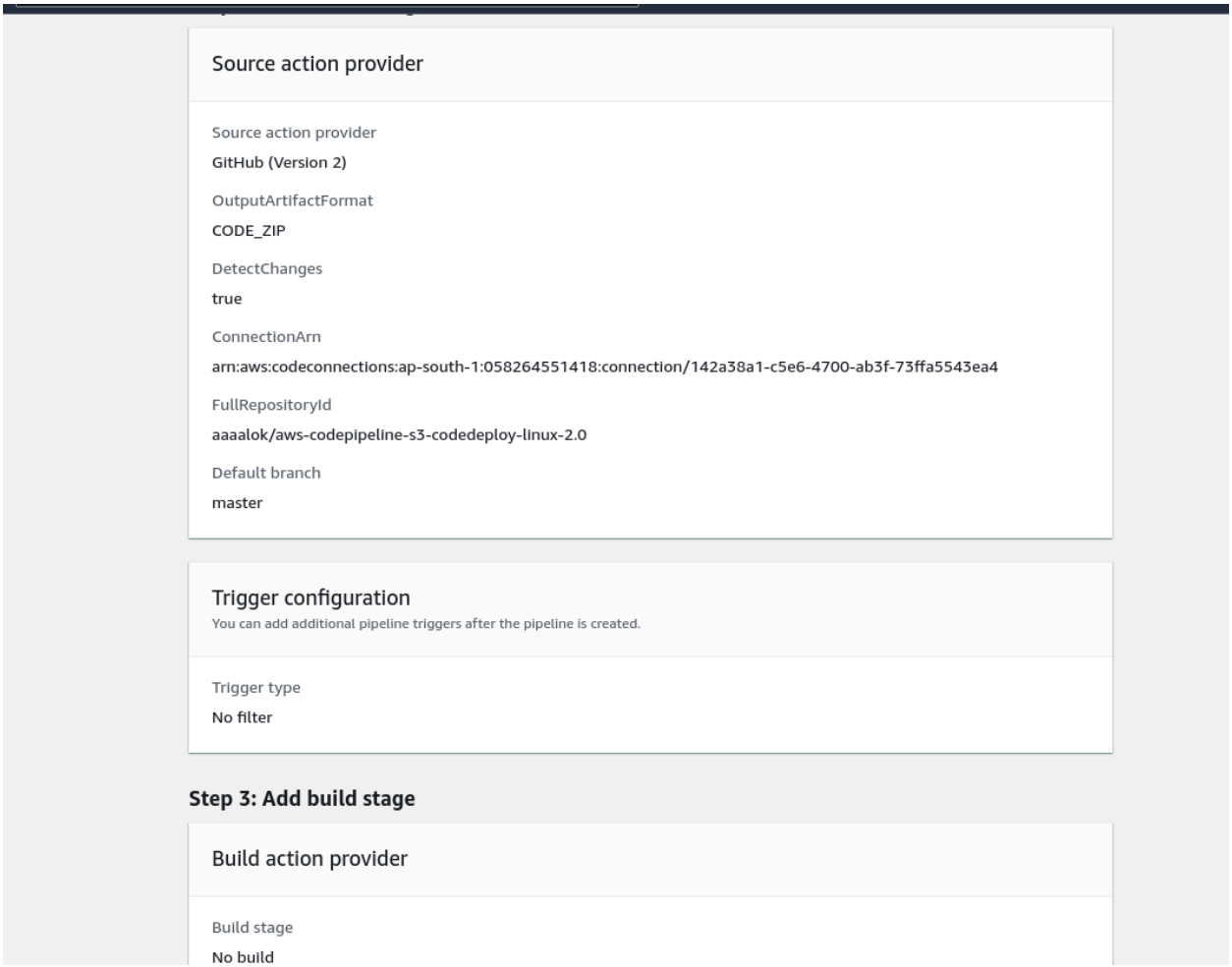
18. As of Now Skip **Build stage** under **Deploy Stage** enter AWS Elastic Beanstalk as Deploy Provider



19. Review the summary of Pipeline created and click on **Create Pipeline**

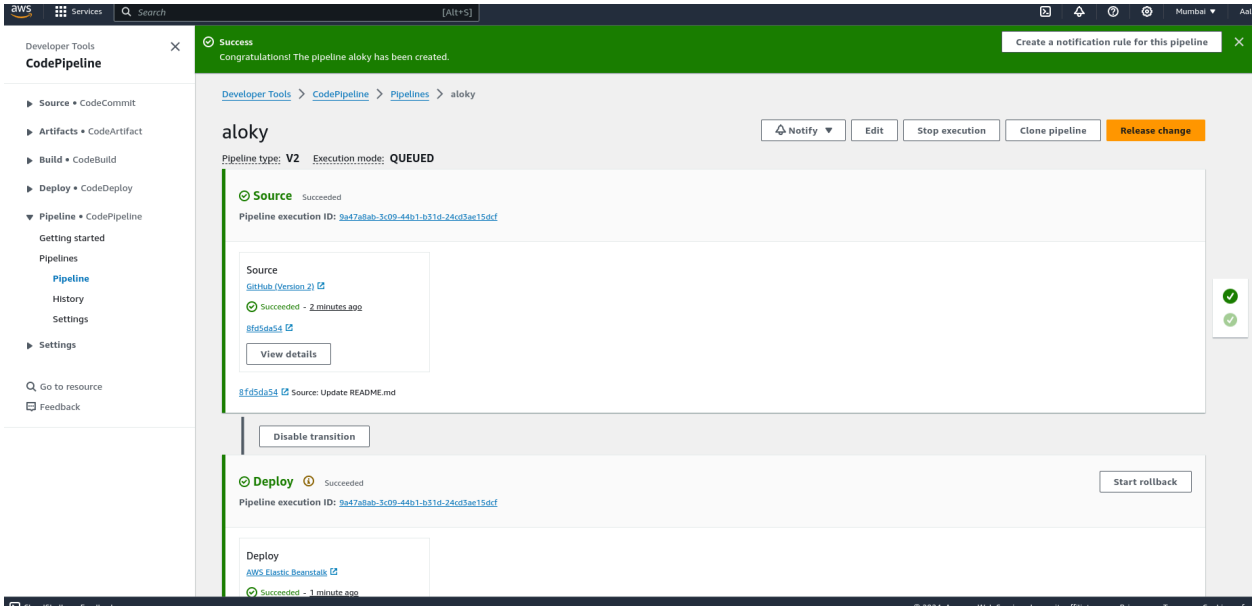
Name: Alok Yadav Div: D15C Roll No.: 59



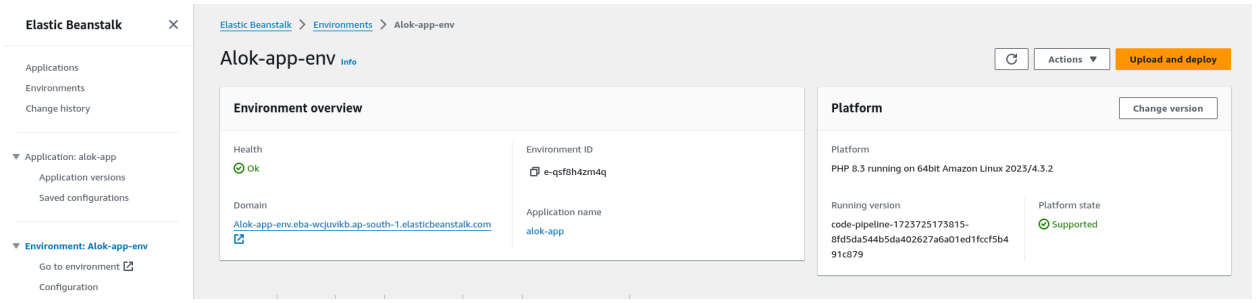


20. Our pipeline will be created and deployed in few minutes

Name: Alok Yadav Div: D15C Roll No.: 59

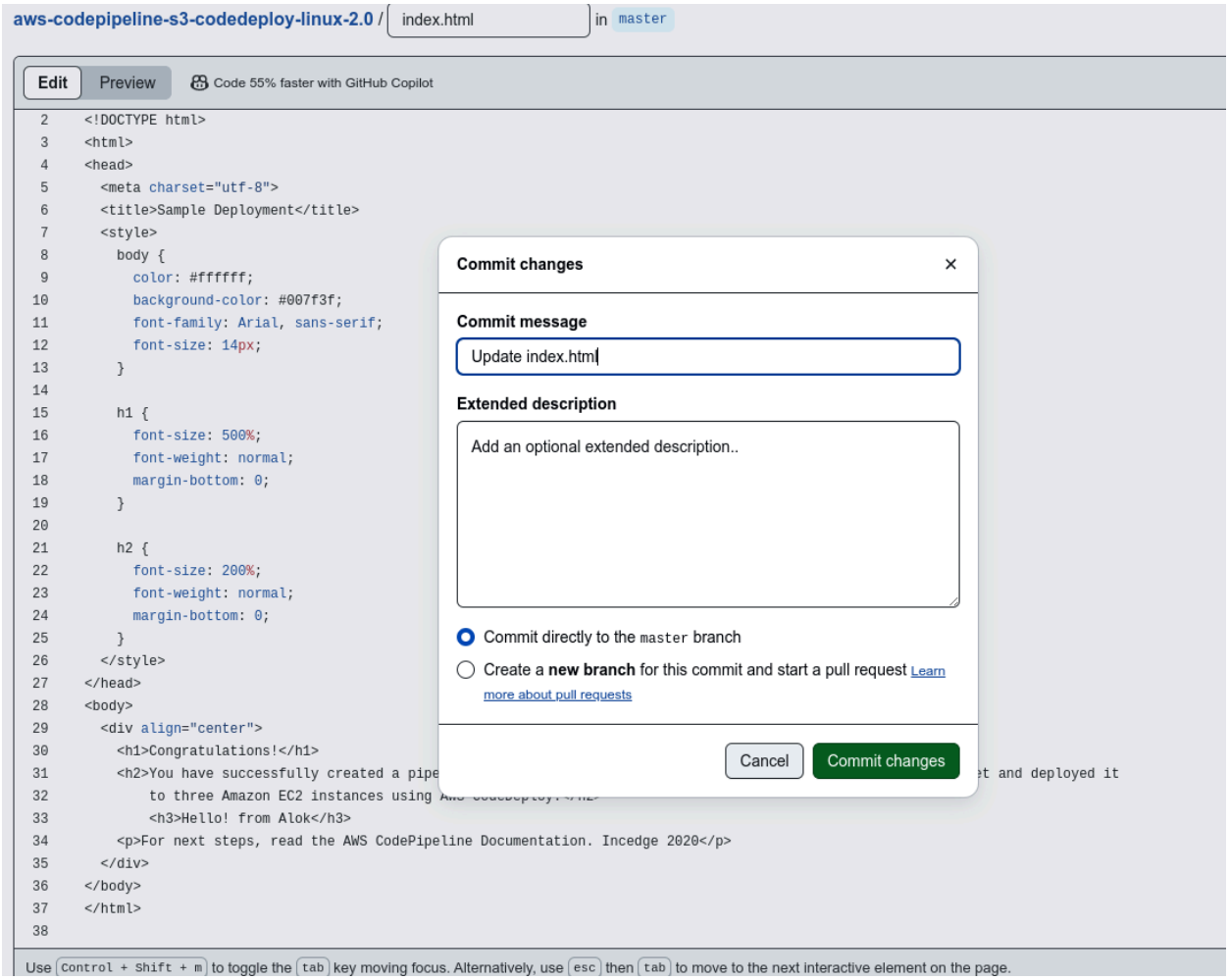


21. Go to Elastic Beanstalk environment that we had created previously and click on the URL given

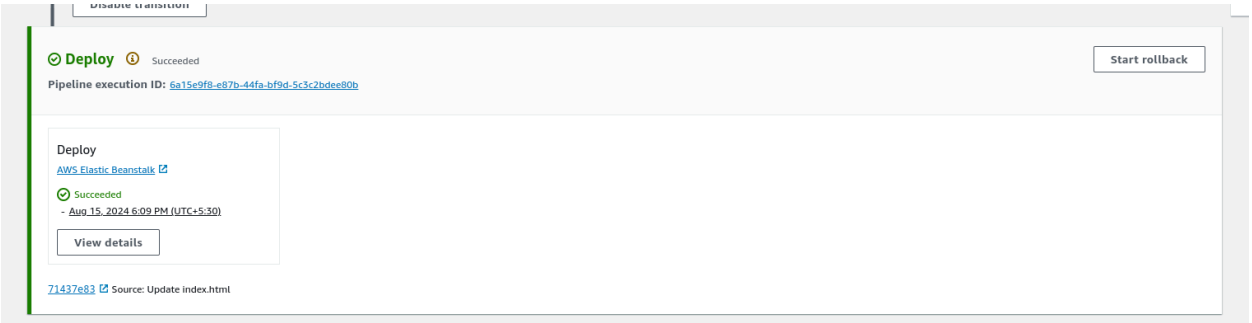


22. Our pipeline is successfully deployed

23. Now to test our pipeline, we would do some changes in files in the repository



24. Deploy process is automatically started on detecting change in repository



25. A new text **Hello! from Alok** is shown in the website.