SUB: ADVANCED DEVOPS LAB SEM: V R2019

BHARATI VIDYAPEETH

COLLEGE OF ENGINEERING

DEPARTMENT OF INFORMATION TECHNOLOGY

ACADEMIC YEAR: 2021-2022

COURSE NAME: Advance DevOps Lab

COURSE CODE	**ITL504**
EXPERIMENT NO.	**02**
EXPERIMENT TITLE.	**To build your application using AWS Codebuild and deploy on S3 / SEBS using AWS Codepipeline, deploy sample application on EC2 instance using AWS CodeDeploy**
NAME OF STUDENT	**SNEHAL RAI**
ROLL NO.	**58**
CLASS	**TE-IT**
SEMESTER	**V**
GIVEN DATE	**29/07/2021**
SUBMISSION DATE	**4/08/2021**
CORRECTION DATE	
REMARK	

TIMELY SUBMISSION		PRESENTATION		UNDERSTANDING		TOTAL MARKS	
	04		04		07		15
NAME& SIGN. OF FACULTY		Dr. S. M. Patil					

SUB: ADVANCED DEVOPS LAB SEM: V R2019

EXPERIMENT 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 CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers. CodeBuild scales continuously and processes multiple builds concurrently, so your builds are not left waiting in a queue. You can get started quickly by using prepackaged build environments, or you can create custom build environments that use your own build tools. With CodeBuild, you are charged by the minute for the compute resources you use.

Benefits:

FULLY MANAGED BUILD SERVICE

AWS CodeBuild eliminates the need to set up, patch, update, and manage your own build servers and software. There is no software to install or manage.

EXTENSIBLE

You can bring your own build tools and programming runtimes to use with AWS CodeBuild by creating customized build environments in addition to the prepackaged build tools and runtimes supported by CodeBuild.

CONTINUOUS SCALING

AWS CodeBuild scales up and down automatically to meet your build volume. It immediately processes each build you submit and can run separate builds concurrently, which means your builds are not left waiting in a queue.

ENABLES CONTINUOUS INTEGRATION AND DELIVERY

AWS CodeBuild belongs to a family of AWS Code Services, which you can use to create complete, automated software release workflows for continuous integration and delivery (CI/CD). You can also integrate CodeBuild into your existing CI/CD workflow. For example, you can use CodeBuild as a worker node for your existing Jenkins server setup for distributed builds.

PAY AS YOU GO

With AWS CodeBuild, you are charged based on the number of minutes it takes to complete your build. This means you no longer have to worry about paying for idle build server capacity.

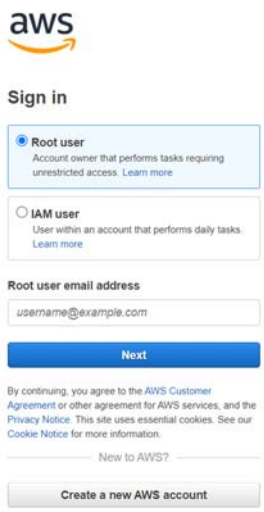
SECURE

With AWS CodeBuild, your build artifacts are encrypted with customer-specific keys that are managed by the AWS Key Management Service (KMS). CodeBuild is integrated with AWS Identity and Access Management (IAM), so you can assign user-specific permissions to your build projects.

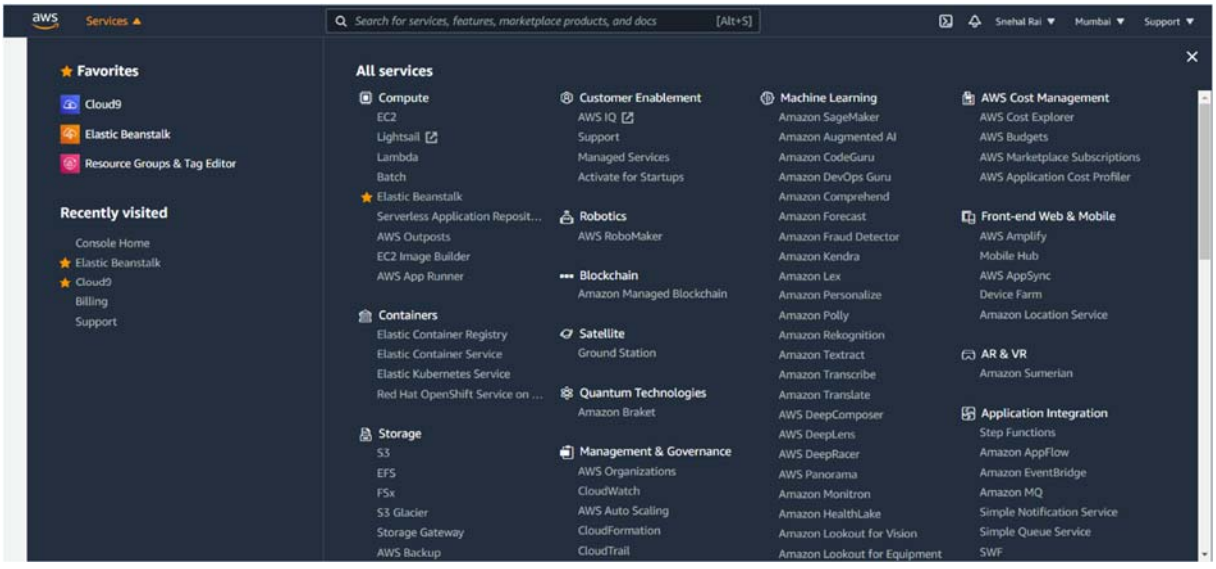
SUB: ADVANCED DEVOPS LAB SEM: V R2019

Steps:

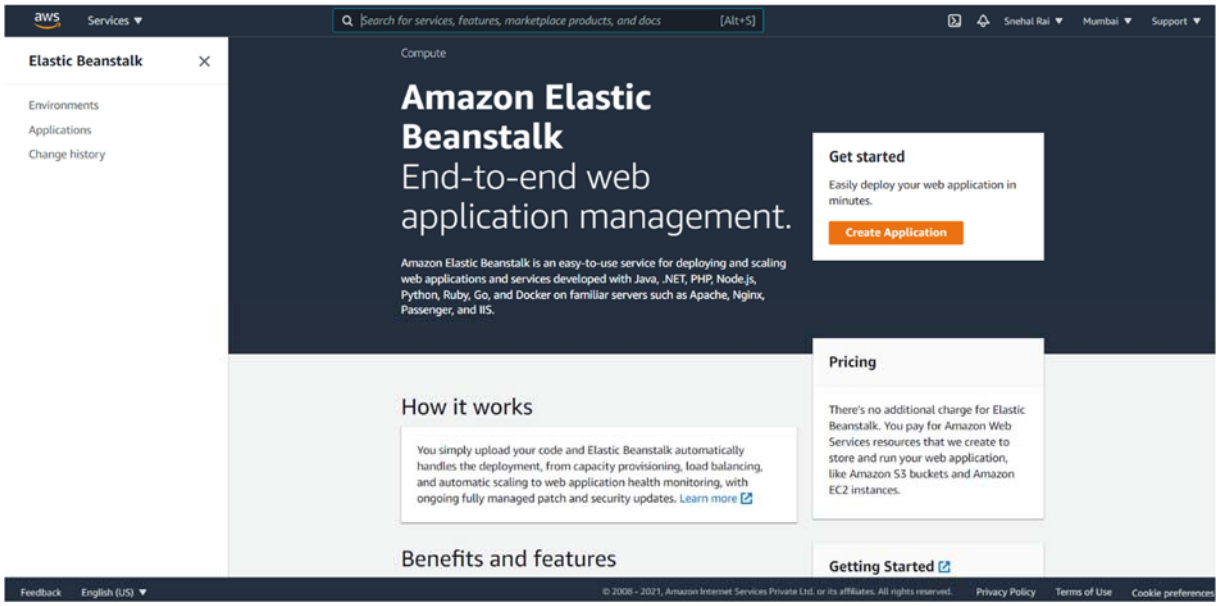
1. **Login with your AWS account.**



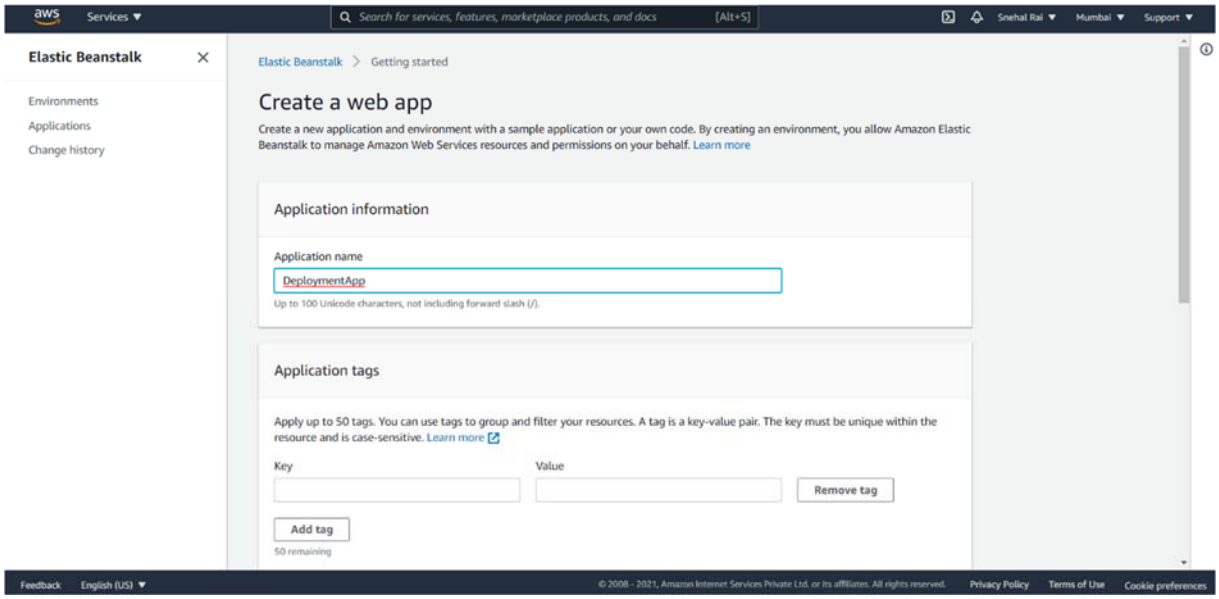
2. **Navigate to Elastic Beanstalk service from Developer tools section as below:**



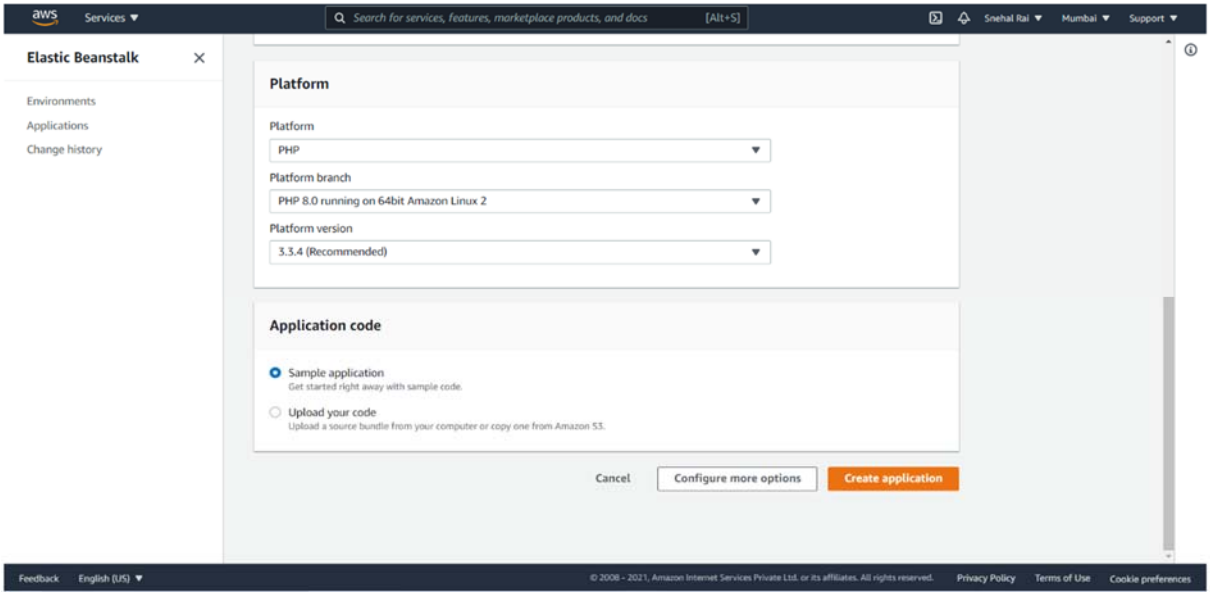
3. **Click on Create Application:**



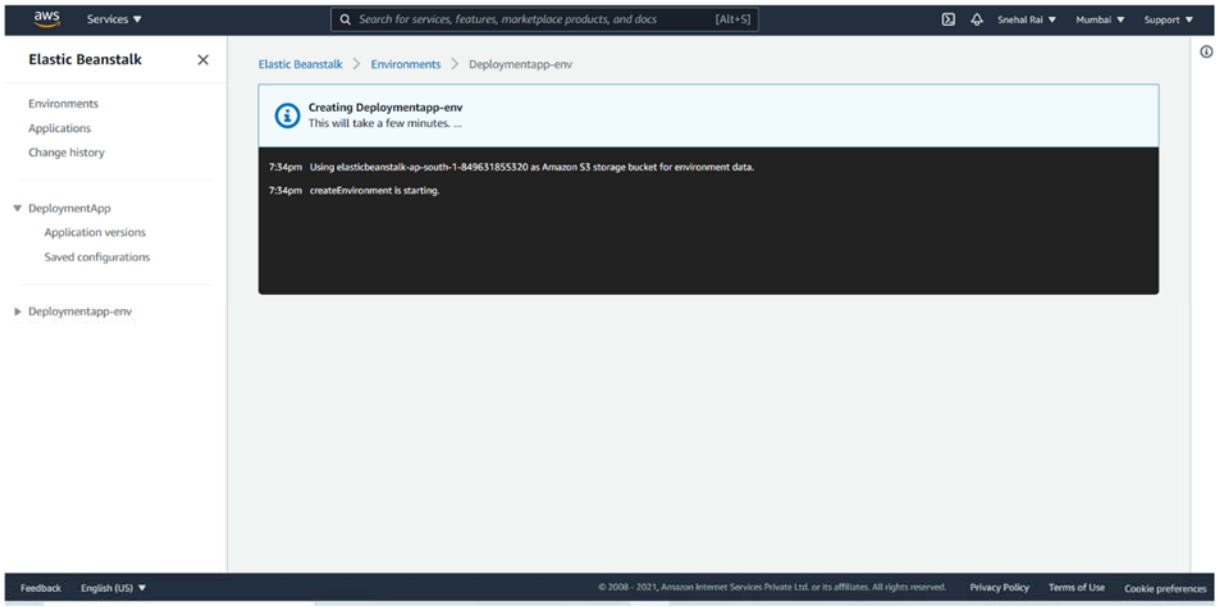
4. **Provide name for the web app .**

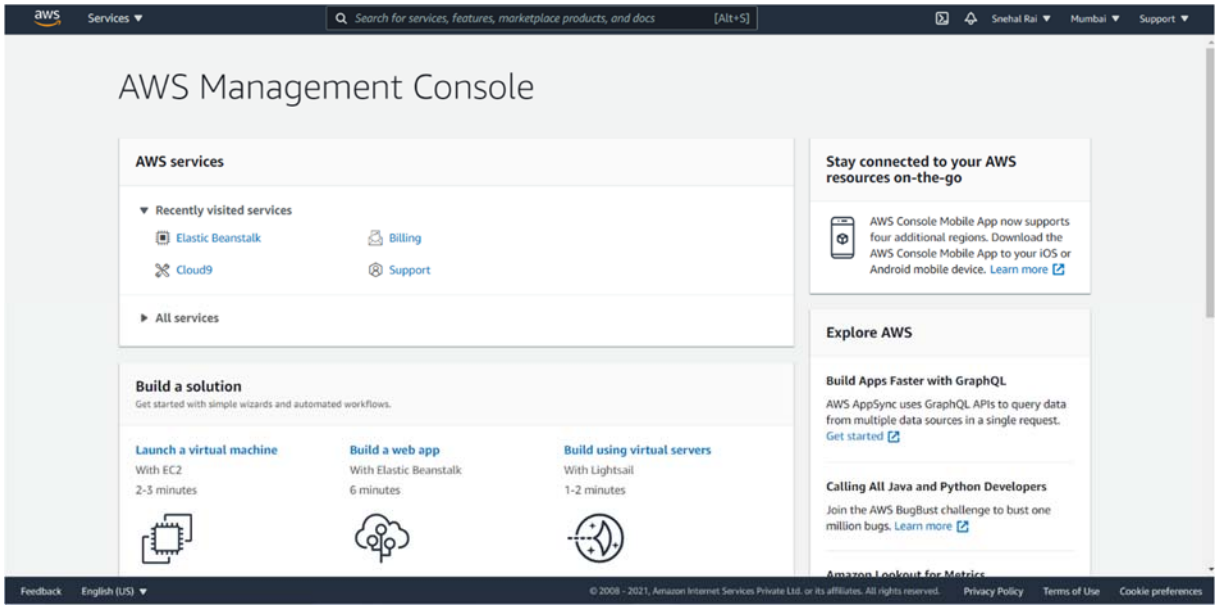


5. **Select PHP as platform and click on create application:**

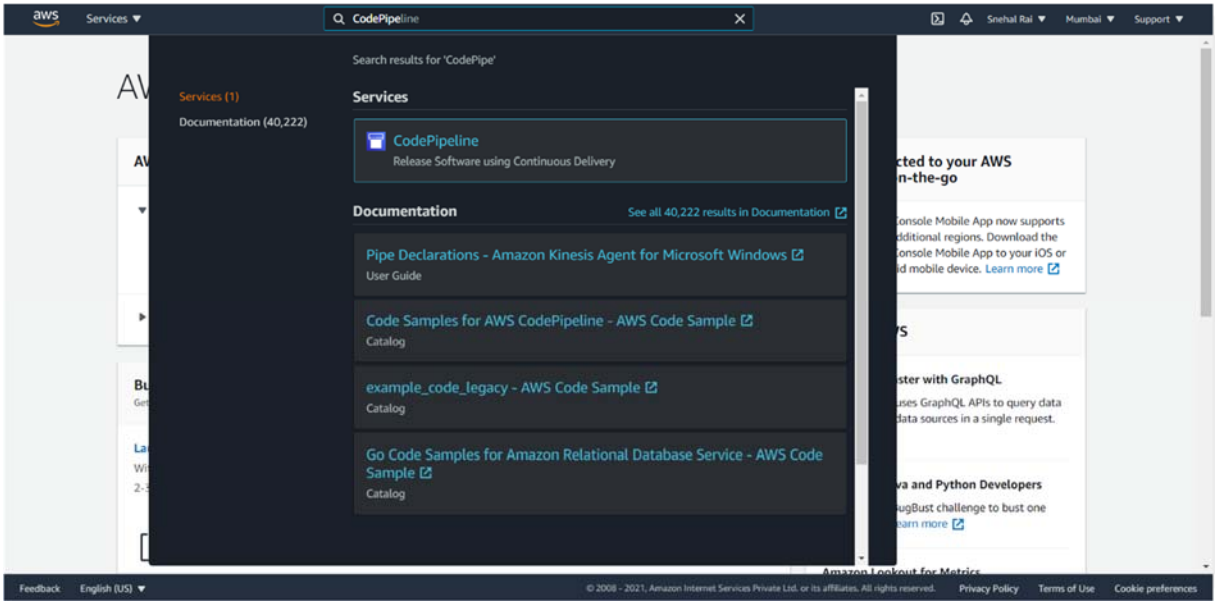


6. **It will take few minutes for the creation of web app environment, till that time open amazon console in new window:**

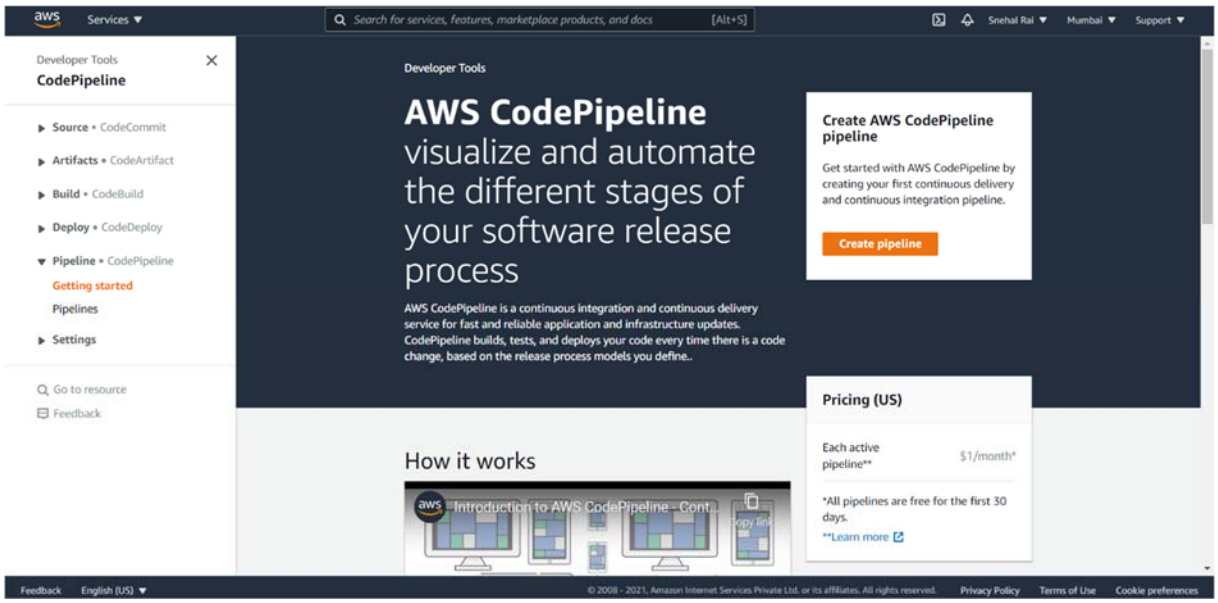




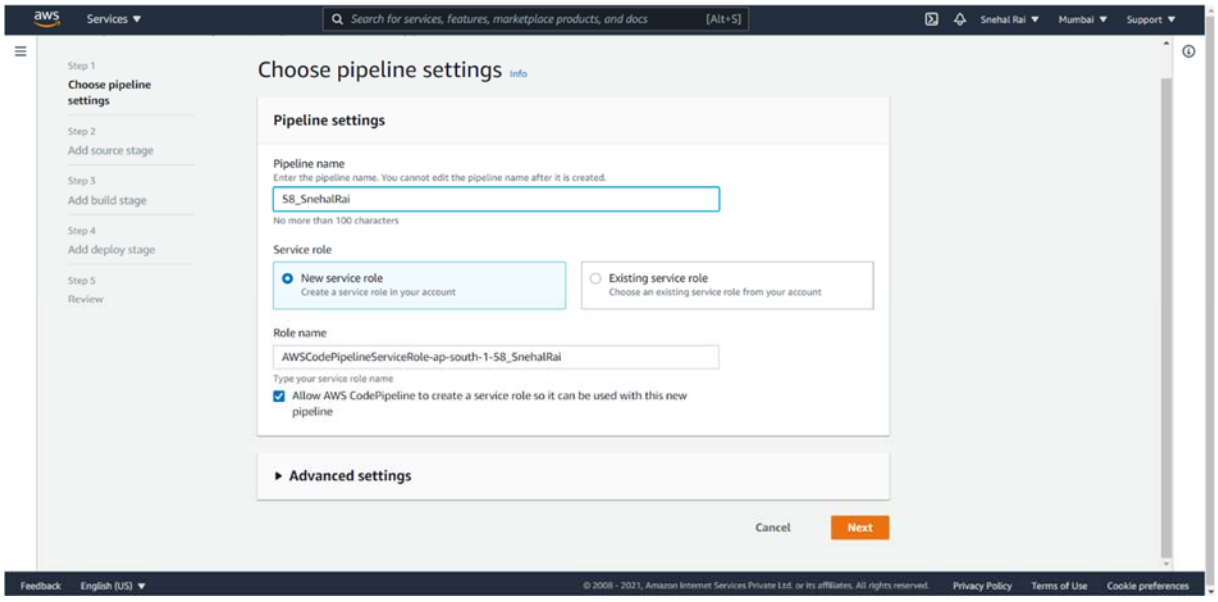
7. **Search for CodePipeline in services search box.**



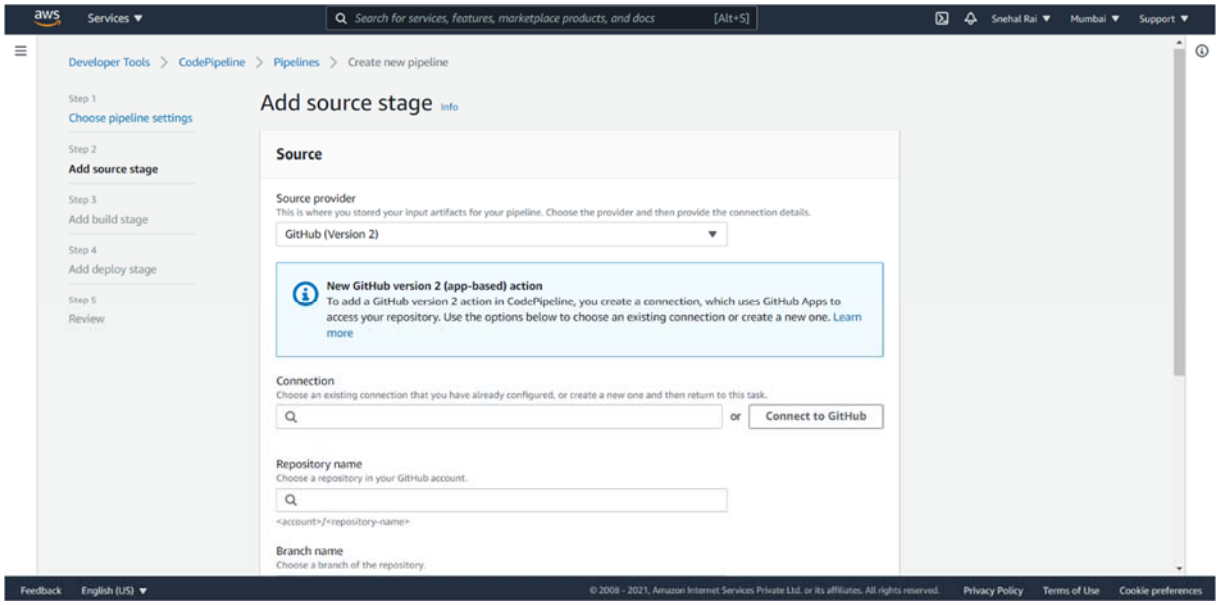
8. **Click on create pipeline.**



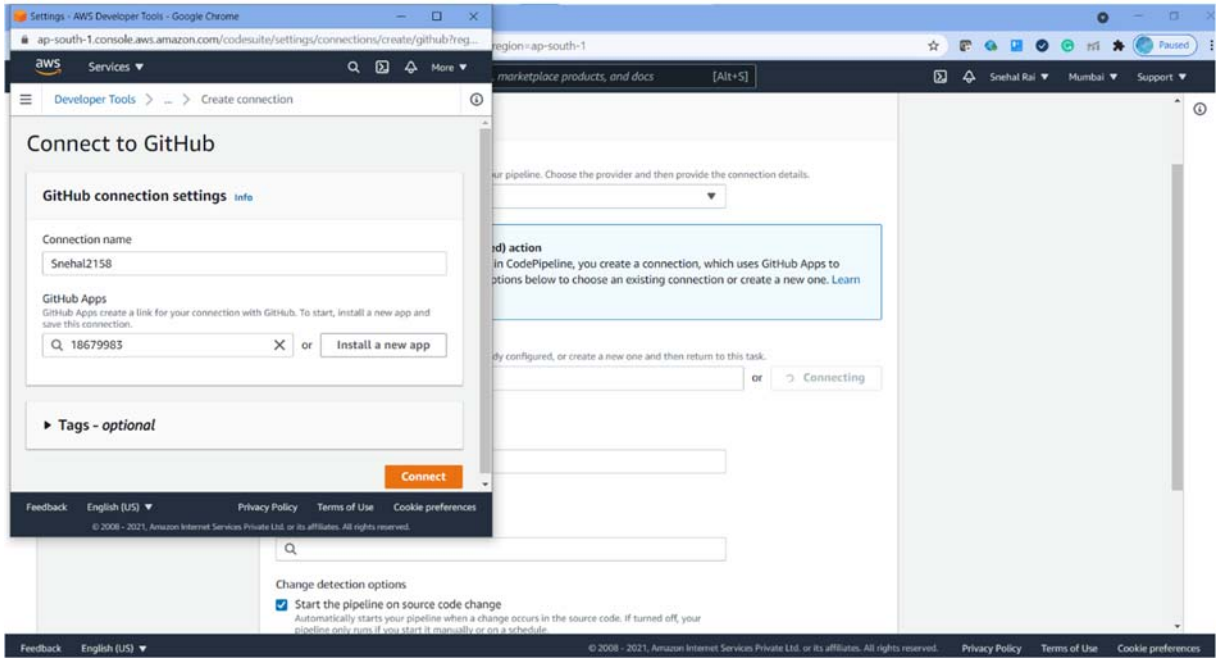
9. **Provide name for Pipeline and click on next**



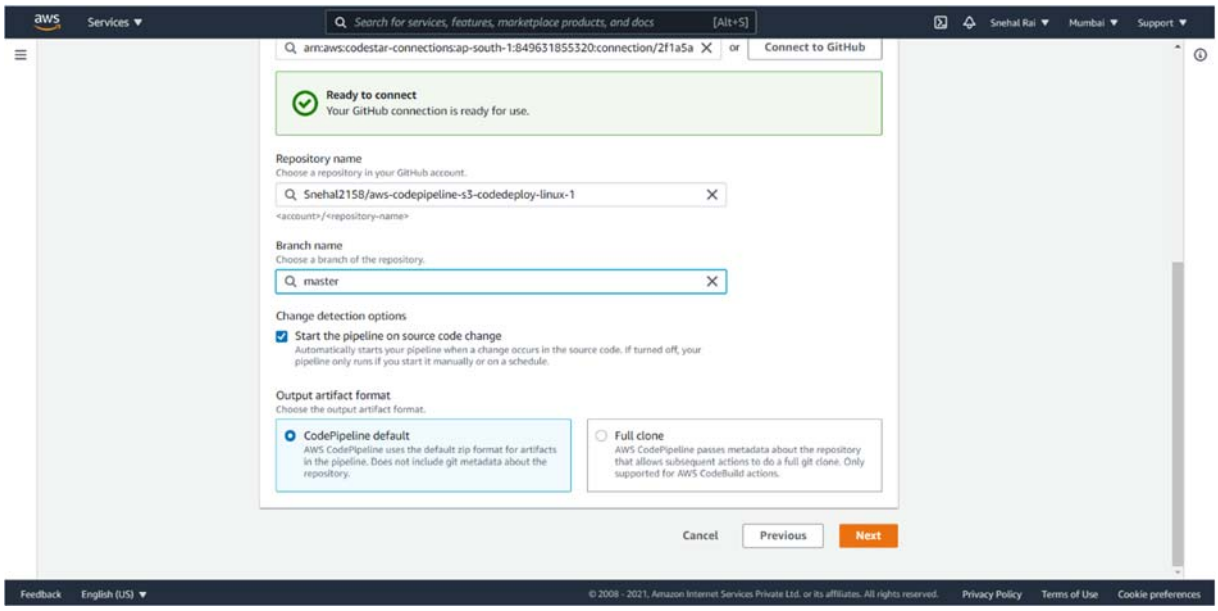
10. Select GitHub(Version 2) as the source provider.



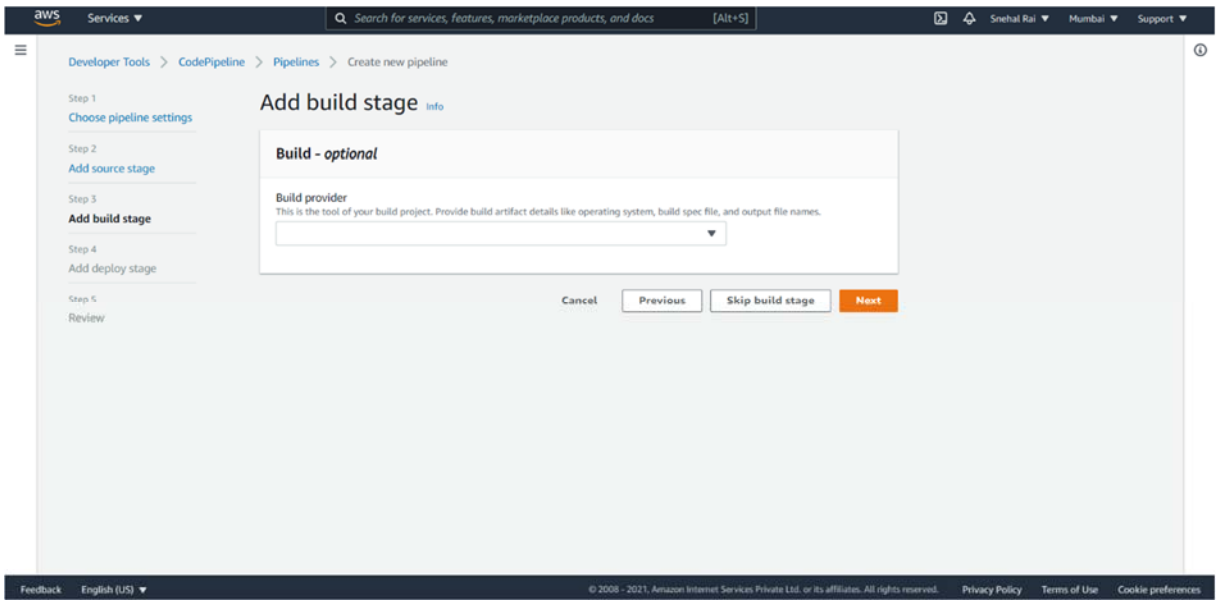
11. Connect your GitHub account to AWS .



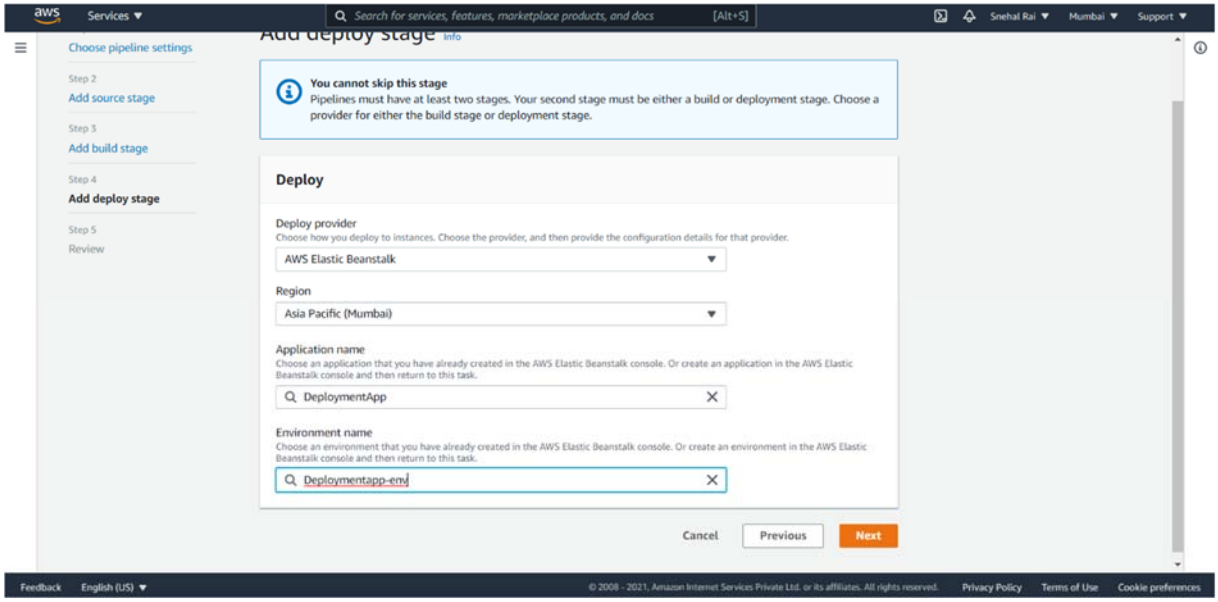
12. Once your GitHub connection is set-up, provide a repository name and select the default master as branch name and click on next.



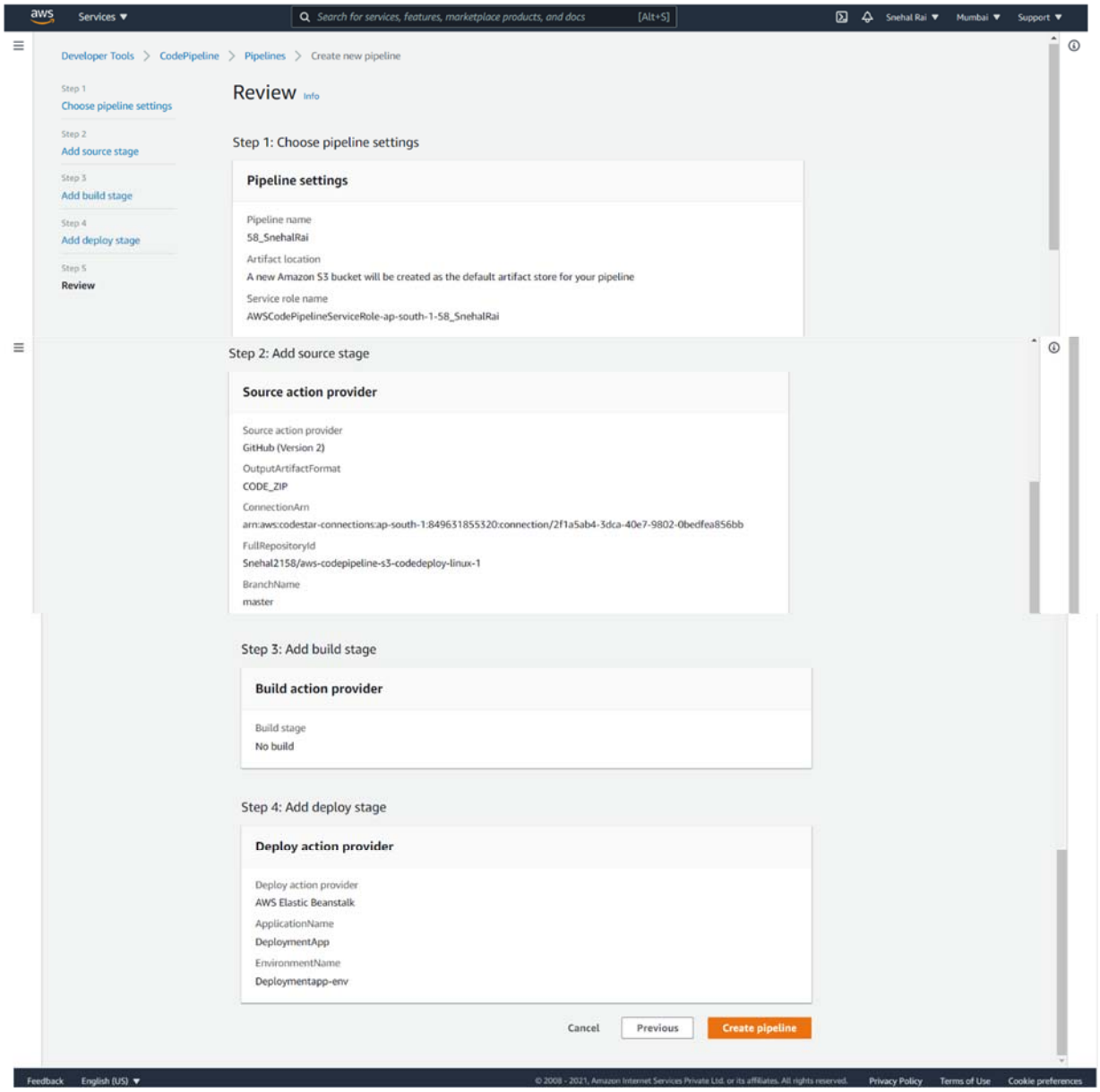
13. Click on Skip build stage:



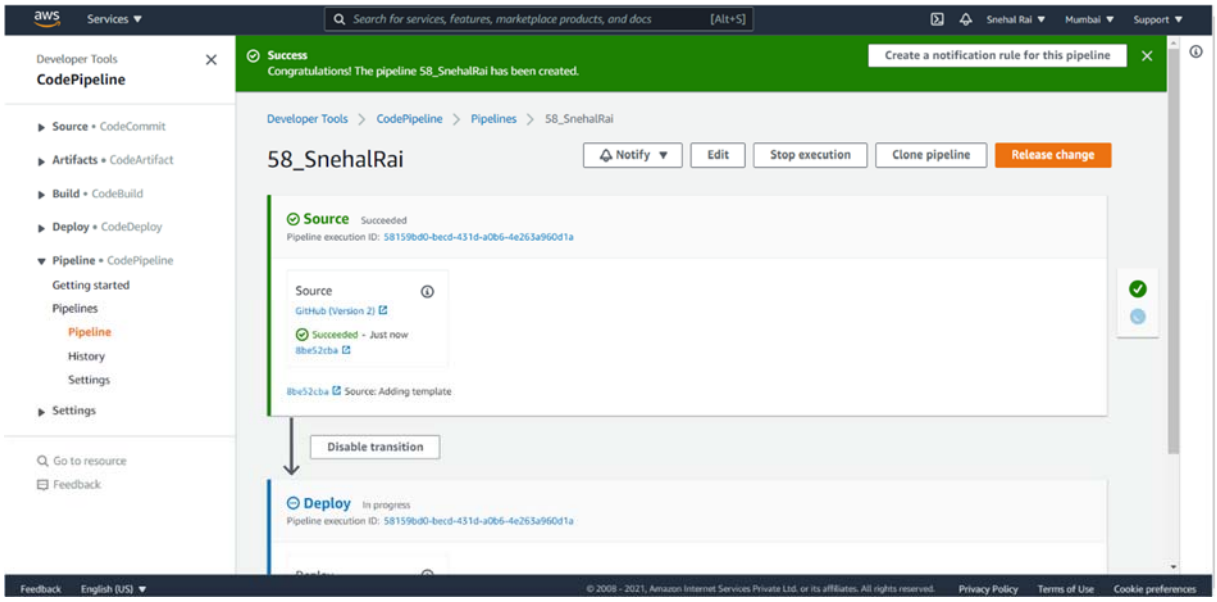
14. Select AWS Elastic Beanstalk as deploy provider and type the application name you provided before and click on next.

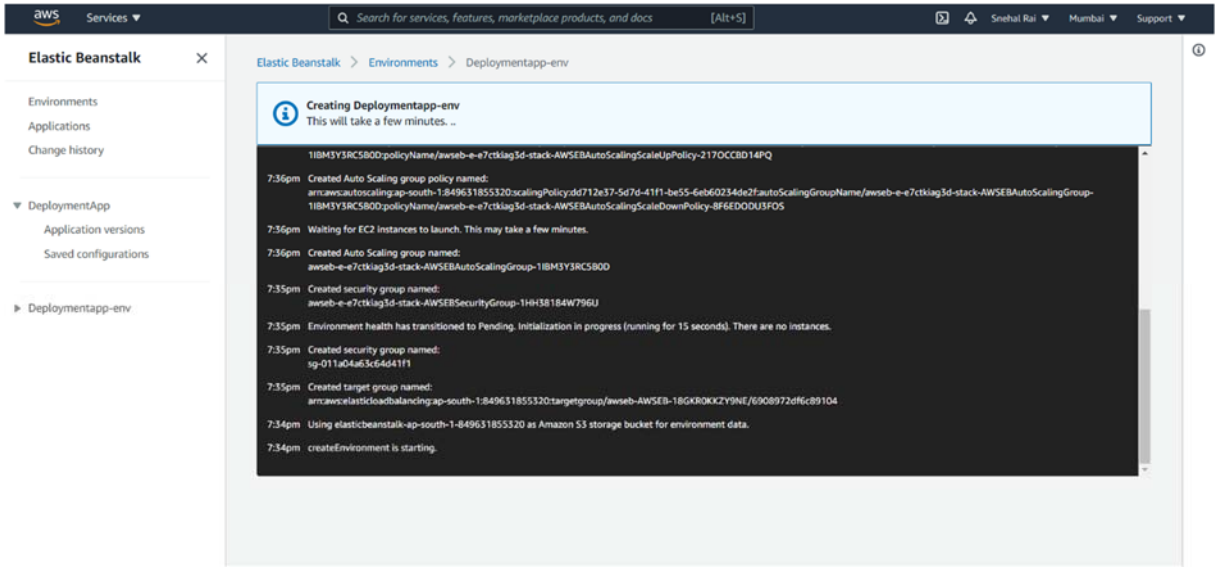


15. Review the settings and click on create Pipeline:

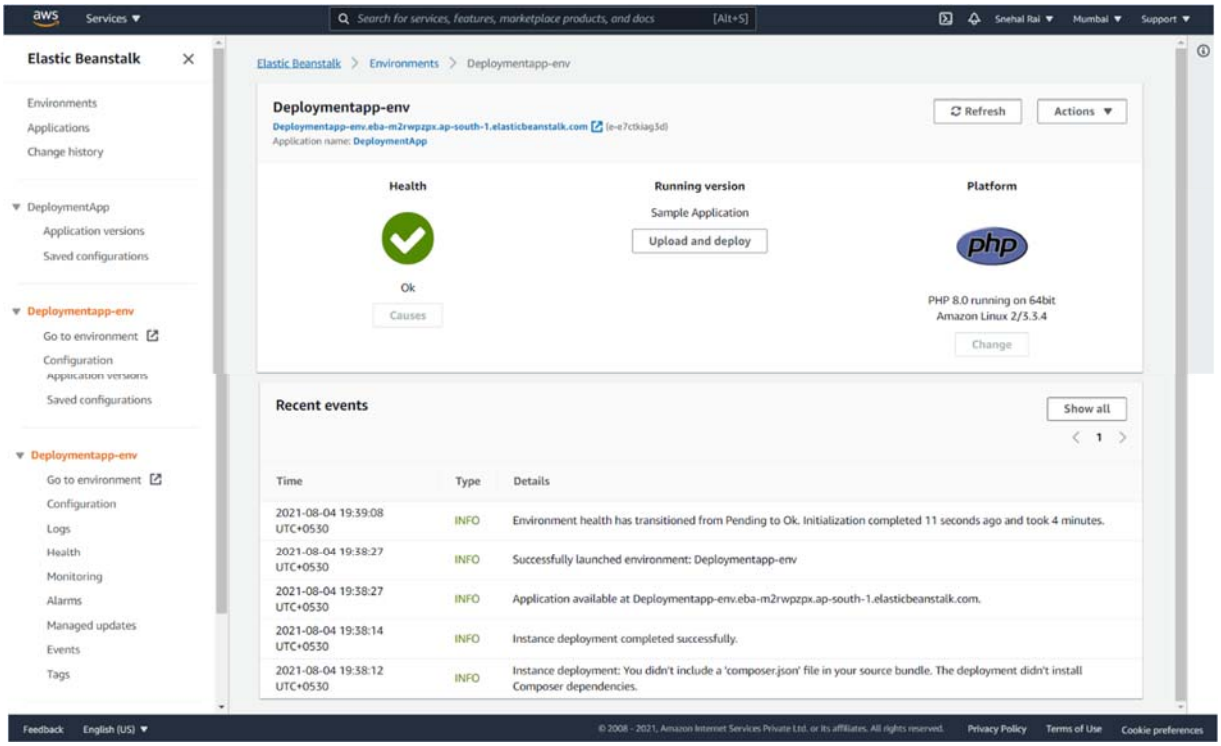


16. **After the successful creation of pipeline, go back to the environment window to check it's status.**

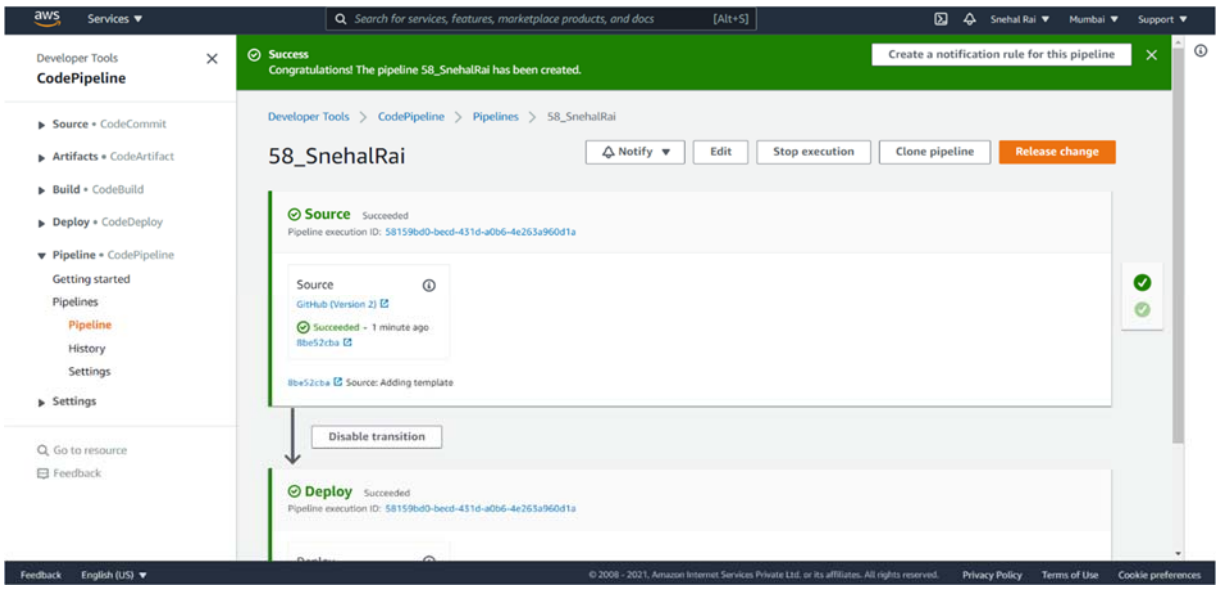


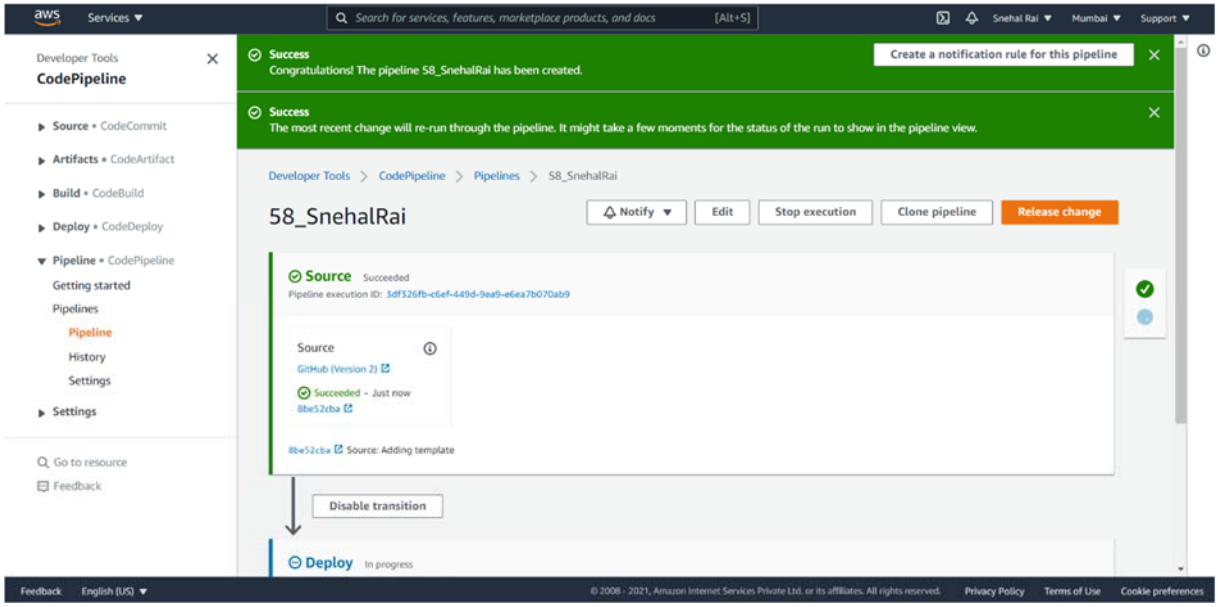


17. Click on beanstalk and and check it's running version and recent events.

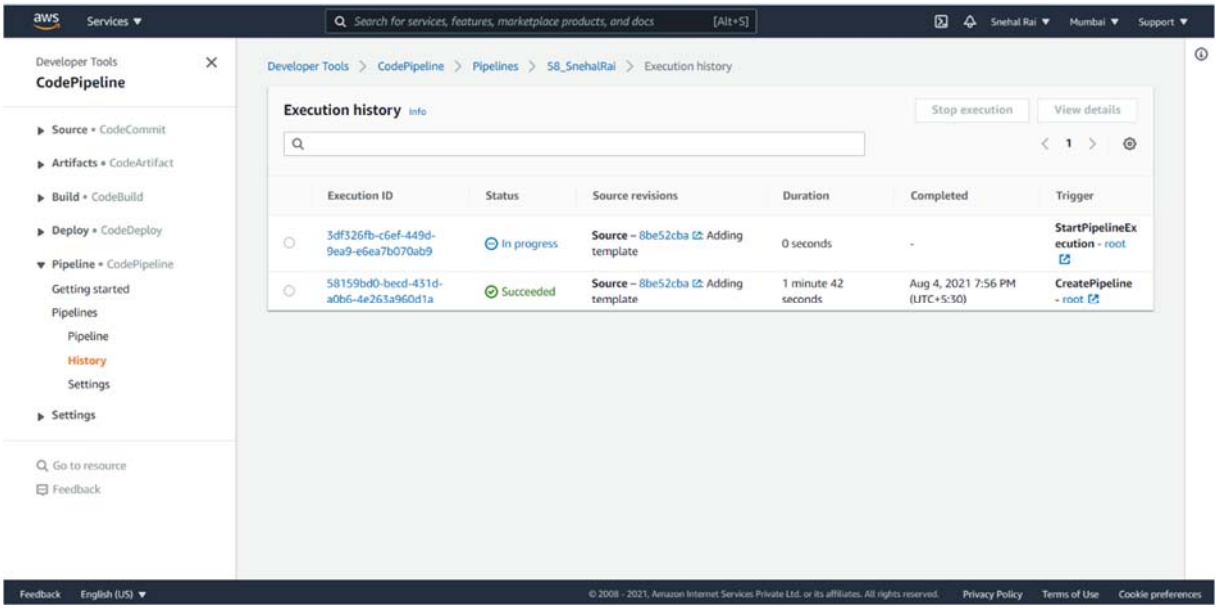


18. Come back to CodePipeline window and click on release change.

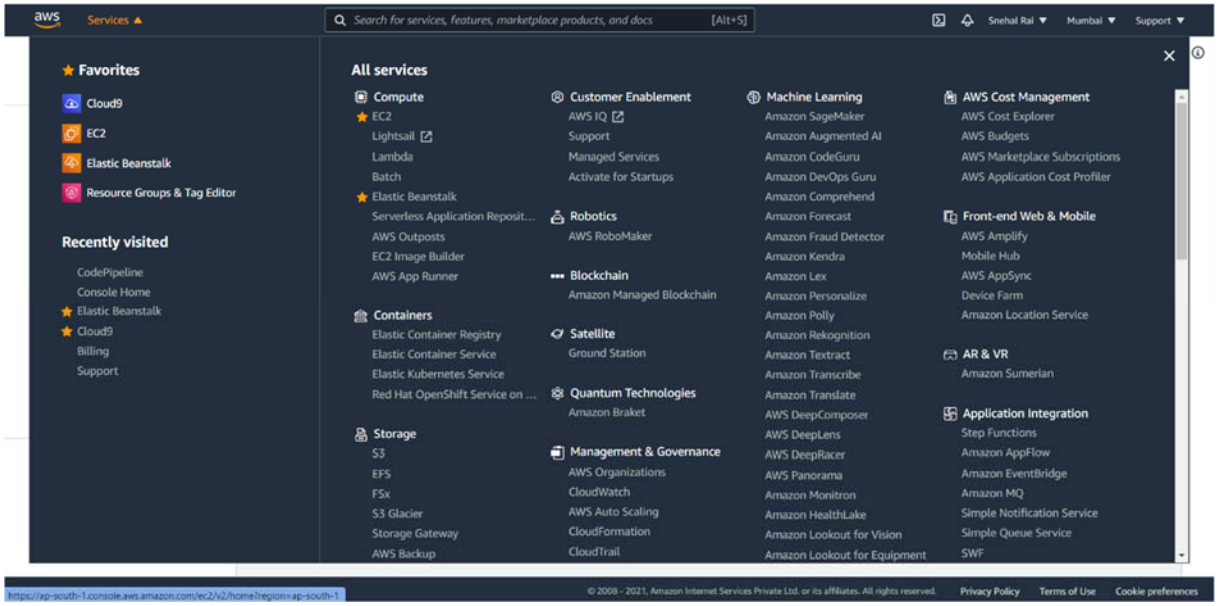




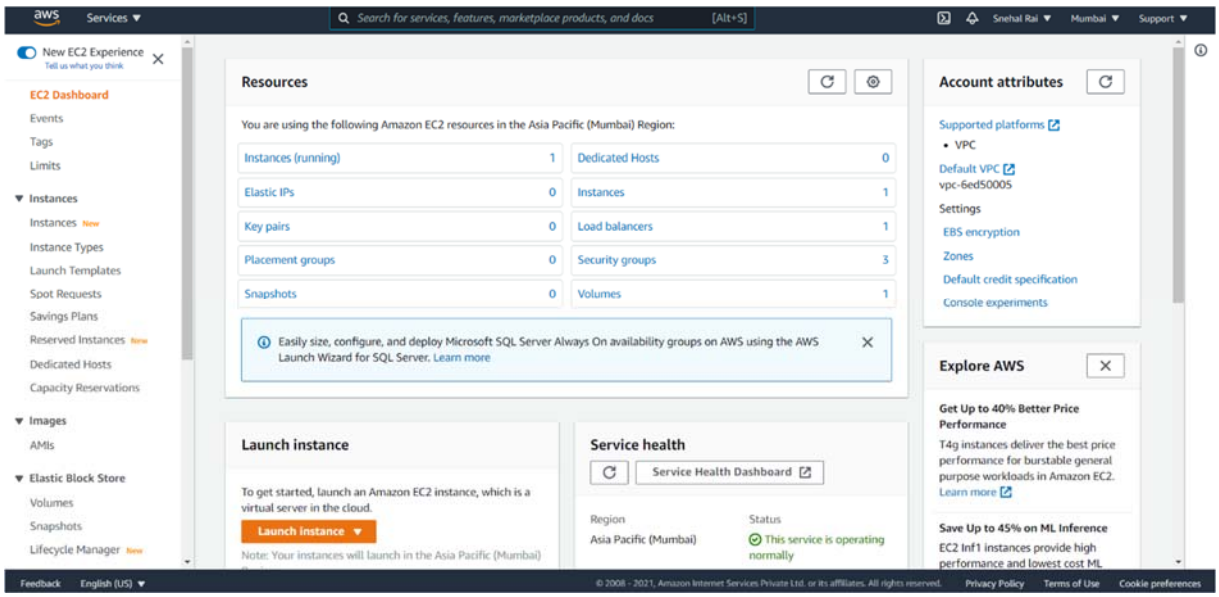
19. Click on History and view the execution history.



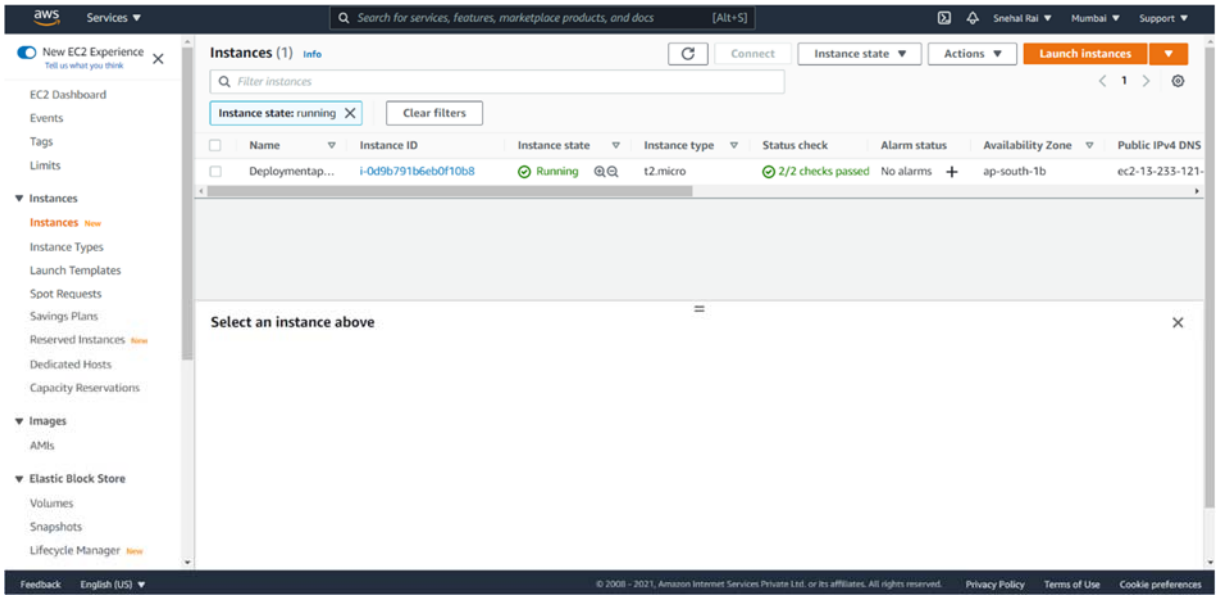
20. Click on Services and select EC2.



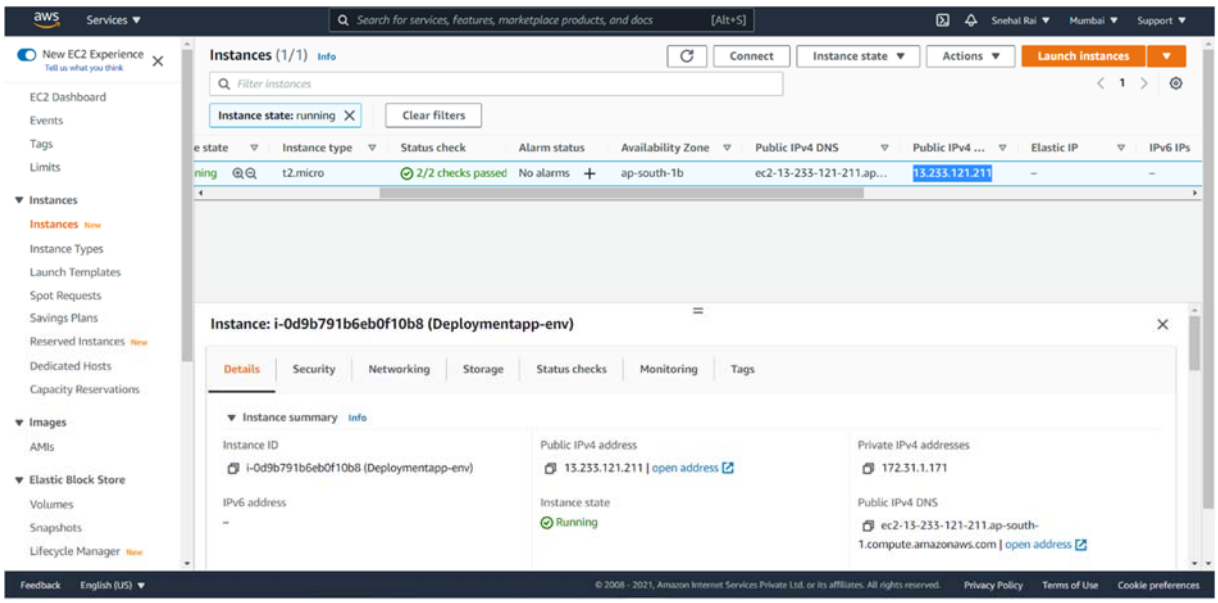
21. From the Resources, click on instances(running).



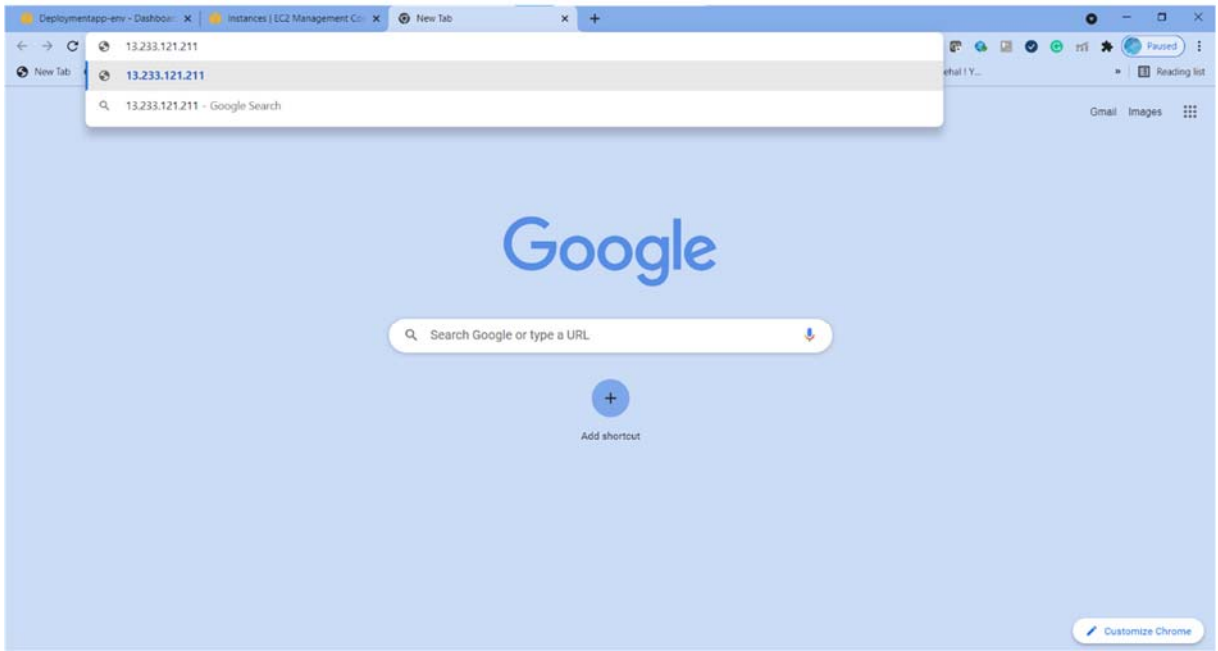
22. Select the running instance.



23. Copy the Public IPv4 address:

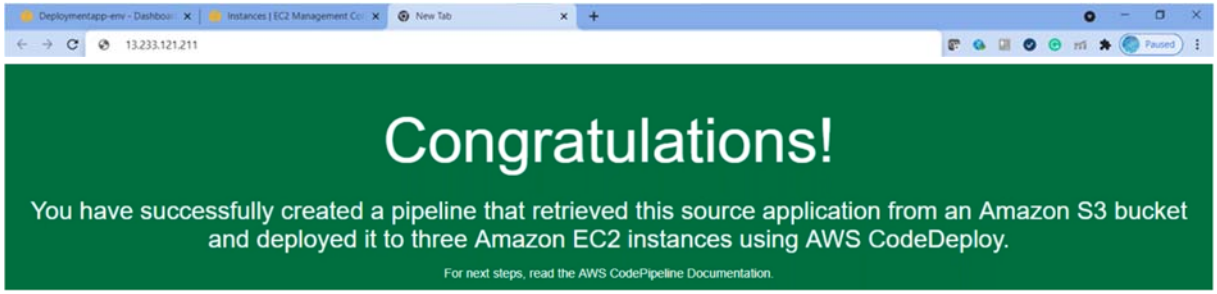


24. Paste the copied IPv4 address in new window.



SUB: ADVANCED DEVOPS LAB SEM: V R2019

25. Successful creation of an automated software release pipeline using AWS CodePipeline



Conclusion: There was successful creation of an automated software release pipeline using AWS CodePipeline. Using CodePipeline, I created a pipeline that uses GitHub, Amazon S3, or AWS CodeCommit as the source location for application code and then deploys the code to an Amazon EC2 instance managed by AWS Elastic Beanstalk. The pipeline will automatically deploy my code every time there is a code change. Continuous deployment allows me to deploy revisions to a production environment automatically without explicit approval from a developer, making the entire software release process automated.