Name: Harsh Dalvi
Roll No: 13
Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

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. **(LO1)**

Theory:

AWS CodePipeline:

AWS CodePipeline is a fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates. CodePipeline enables you to model, visualize, and automates the build, test, and deploy phases of your release process every time there is a code change, based on the release model you define. This enables you to rapidly and reliably deliver features and updates. You can integrate partner tools and your own custom tools into any stage of the release process to form an end-to-end continuous delivery solution. For example you can easily integrate AWS CodePipeline with third-party services such as GitHub or with your own custom plugin. With AWS CodePipeline, you only pay for what you use. There are no upfront fees or long-term commitments.

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.

Amazon S3 or Amazon Simple Storage Service is a service offered by Amazon Web Services that provides object storage through a web service interface. To store your data in Amazon S3, you first create a bucket and specify a bucket name and AWS Region. Then, you upload your data to that bucket as objects in Amazon S3. Each object has a key (or key name), which is the unique identifier for the object within the bucket.

Benefits of CodePipeline:

1. **Rapid delivery:** AWS CodePipeline automates your software release process, allowing you to rapidly release new features to your users. With CodePipeline, you can quickly iterate on feedback and get new features to your users faster.Automating your build, test, and release process allows you to quickly and easily test each code change and catch bugs while they are small and simple to fix. You can ensure the quality of your application or infrastructure code by running each change through your staging and release process.

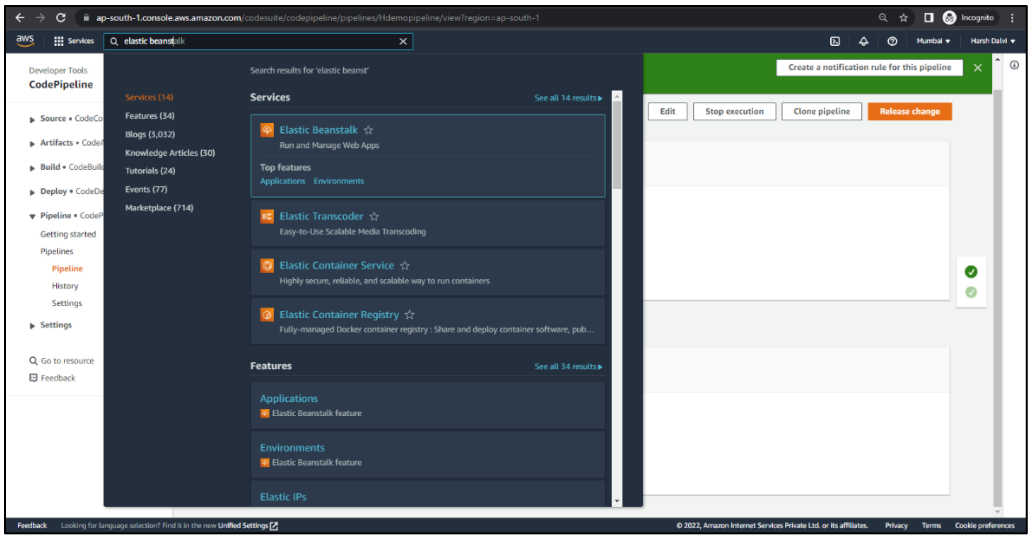
2. **Configurable workflow:** AWS CodePipeline allows you to model the different stages of your software release process using the console interface, the AWS CLI, AWS CloudFormation, or the AWS SDKs. You can easily specify the tests to run and customize the steps to deploy your application and its dependencies.

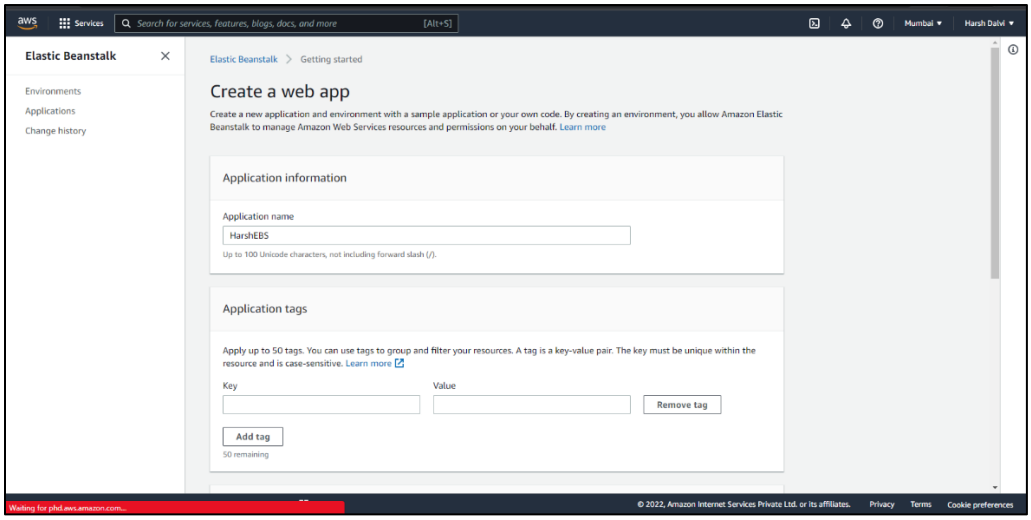
3. **Get started fast:** With AWS CodePipeline, you can immediately begin to model your software release process. There are no servers to provision or set up. CodePipeline is a fully managed continuous delivery service that connects to your existing tools and systems.

4. **Easy to integrate:** AWS CodePipeline can easily be extended to adapt to your specific needs. You can use our pre-built plugins or your own custom plugins in any step of your release process. For example, you can pull your source code from GitHub, use your on-premises Jenkins build server, run load tests using a third-party service, or pass on deployment information to your custom operations dashboard.

Using GitHub

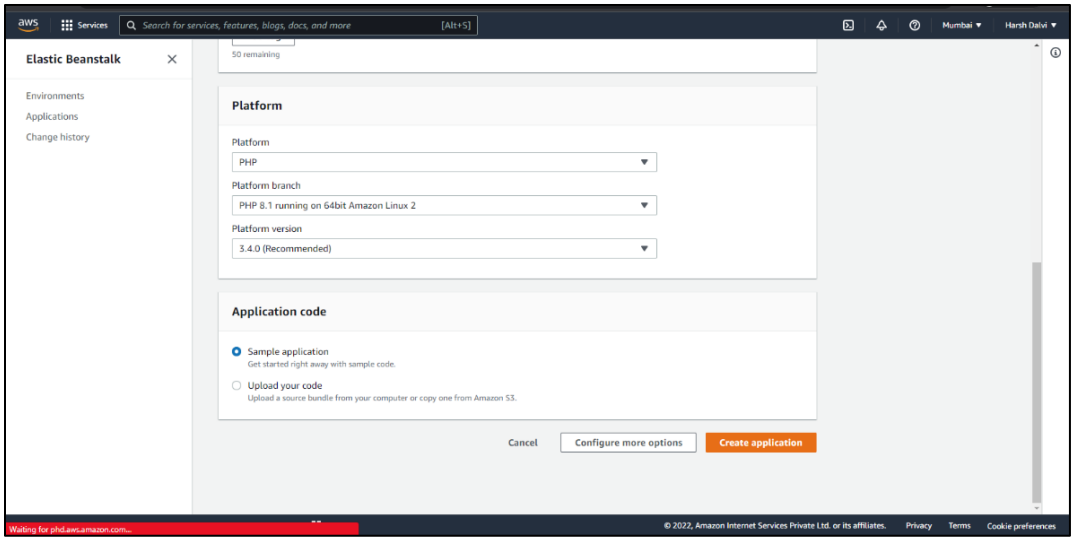
1) Create a deployment environment.

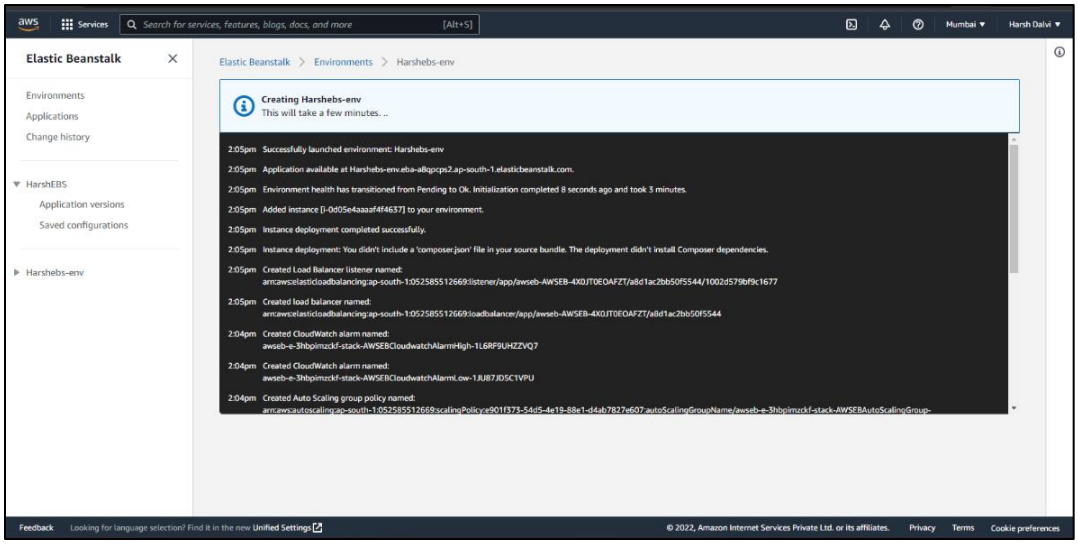


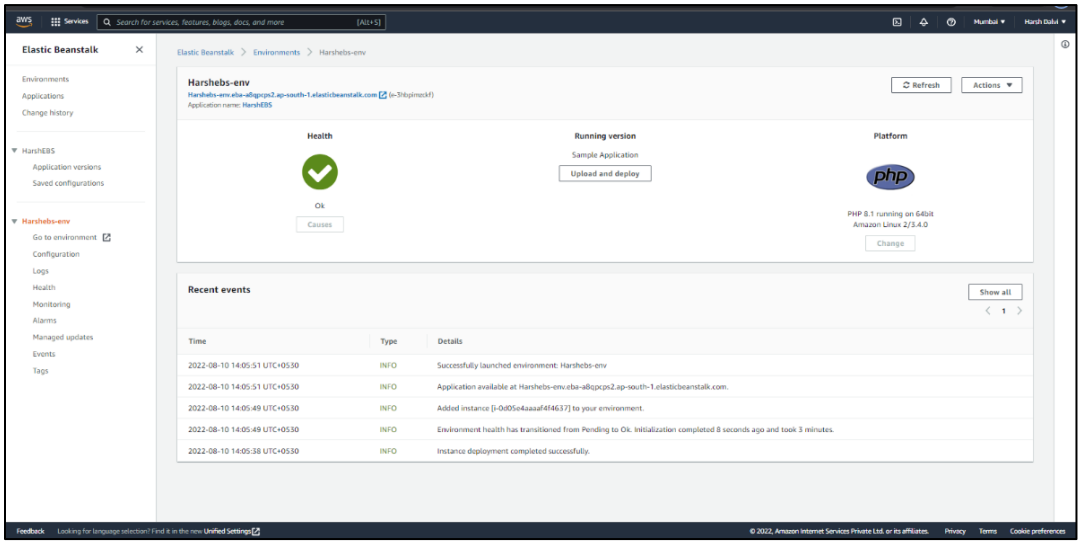


Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B





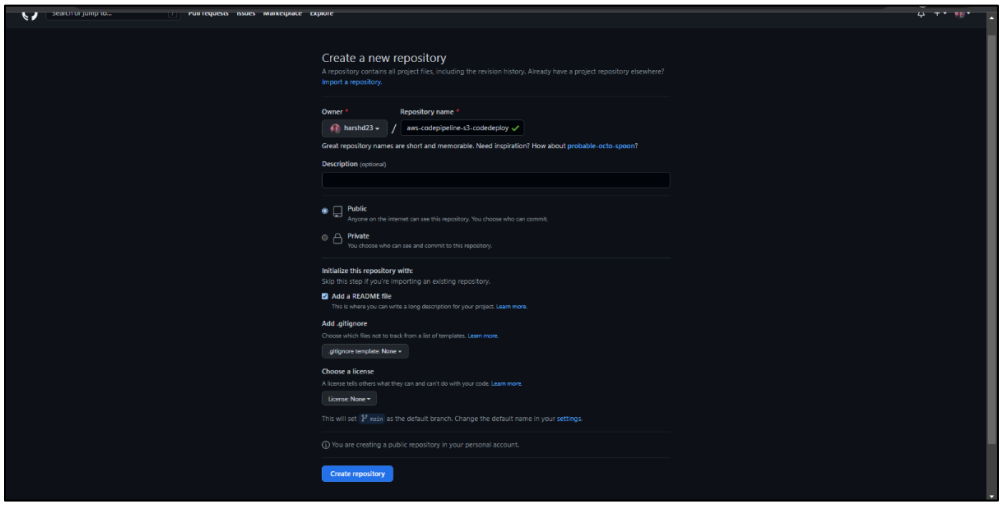


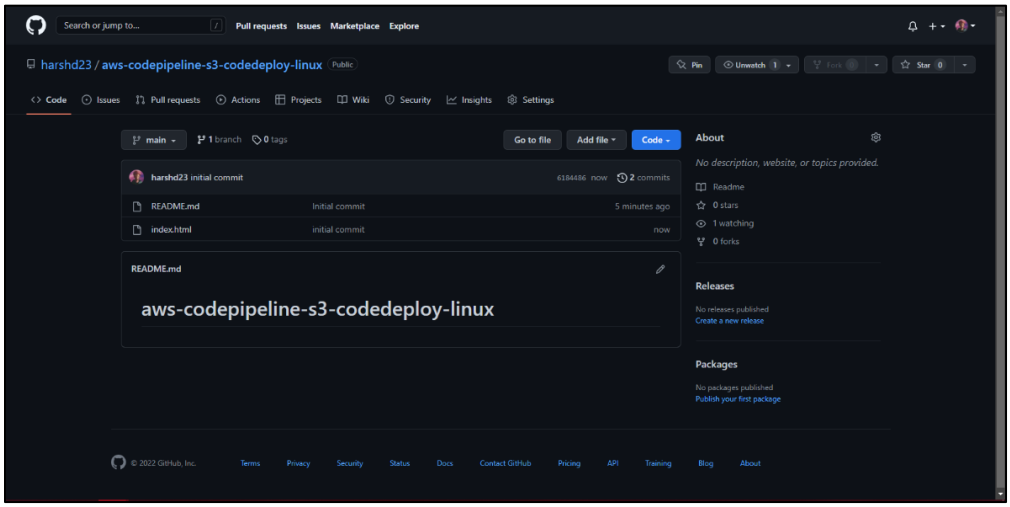
Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

2) Get a copy of a sample code.

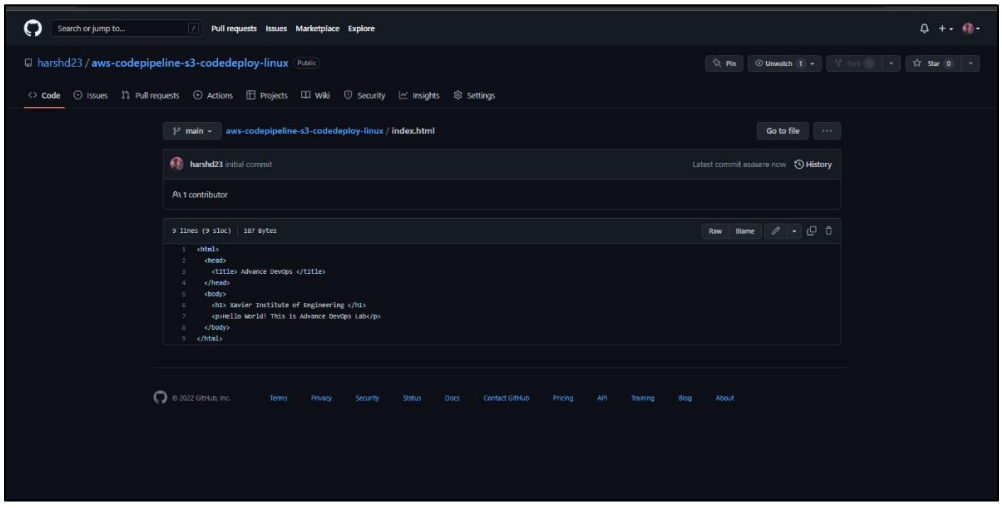
a) Go to Github and create new repository.





b) Create a new file with name index.html and write a sample html code.

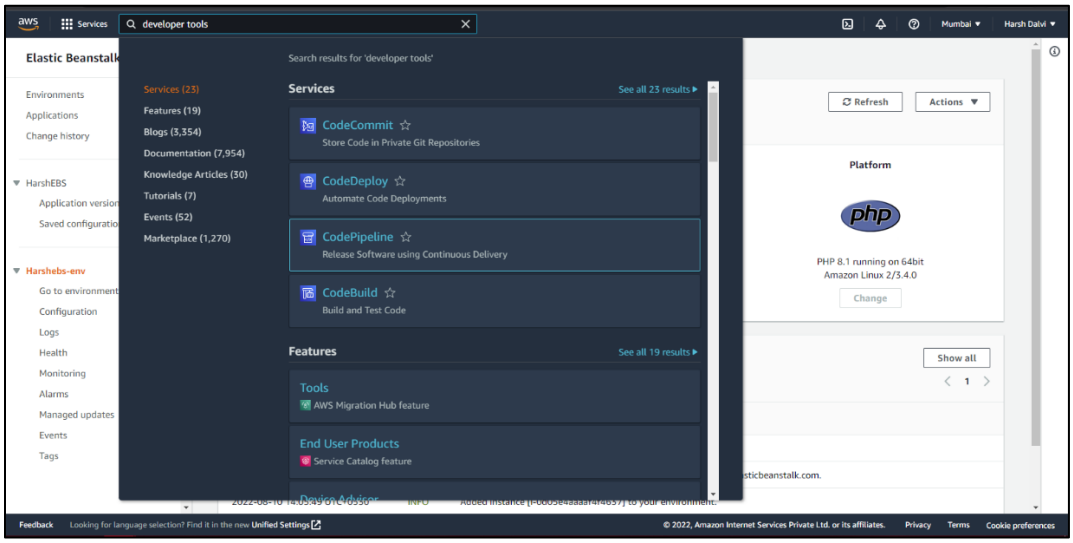
Code:

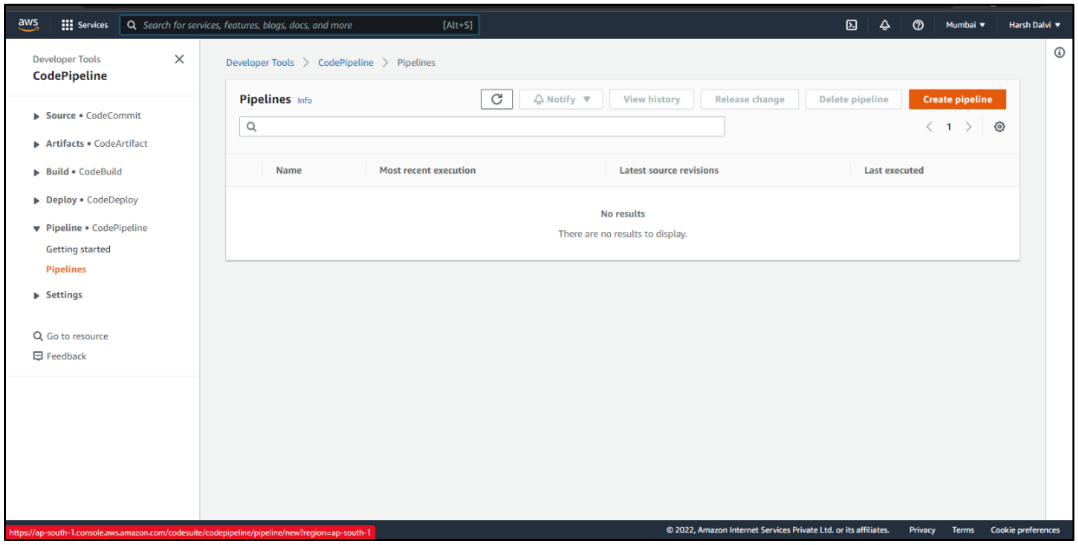


Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

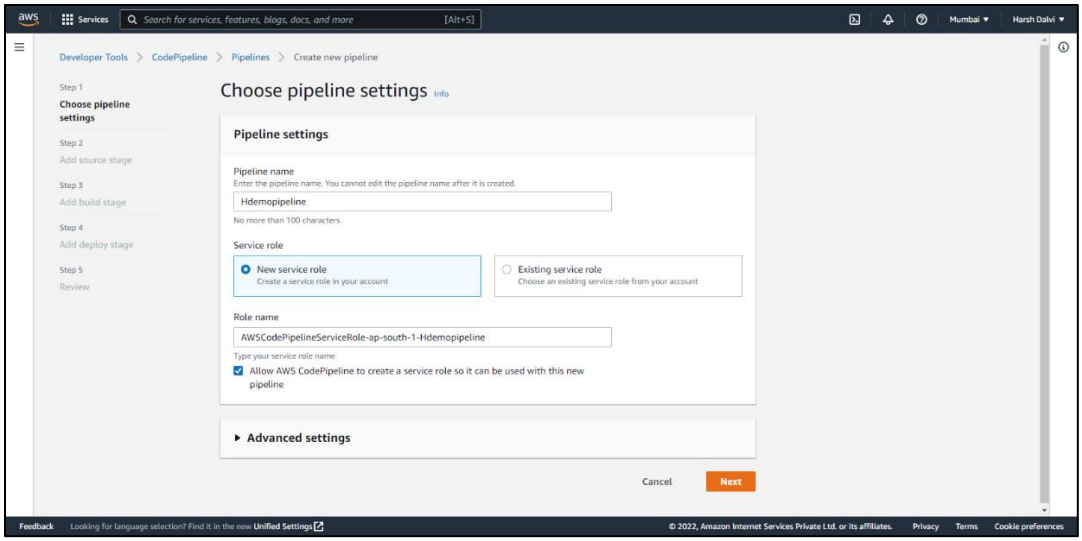
3) Create your pipeline.

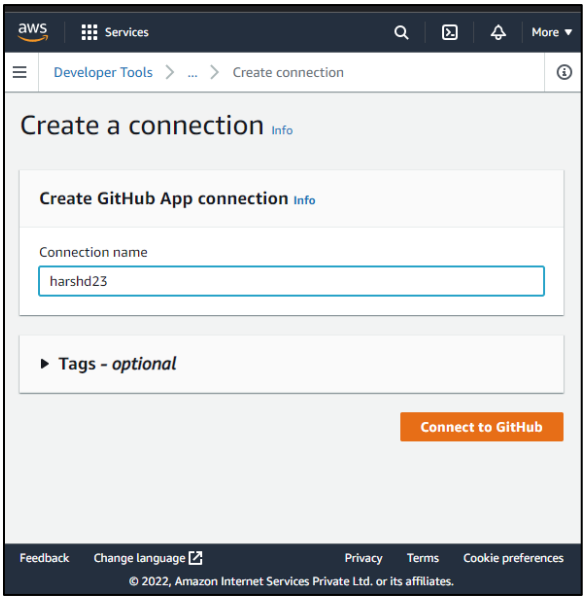


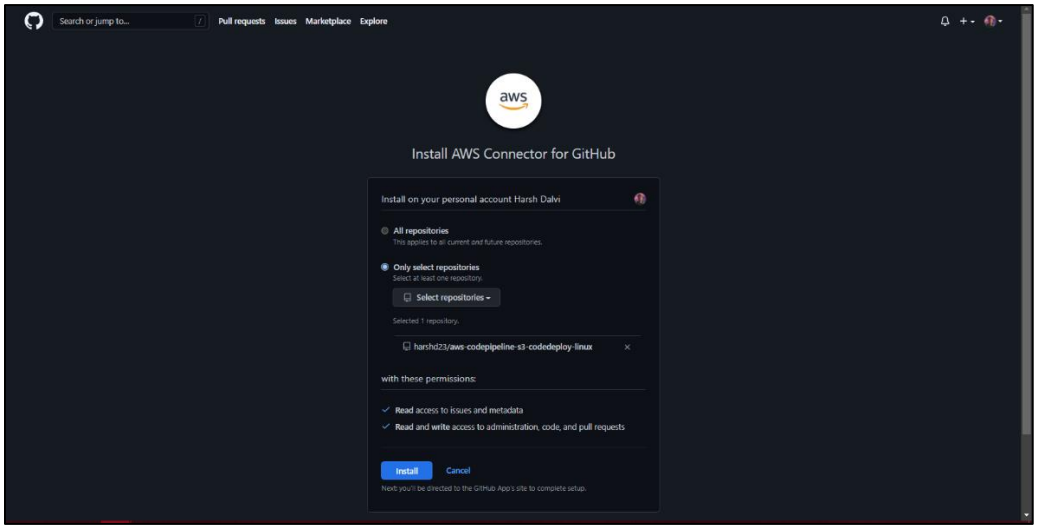


Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

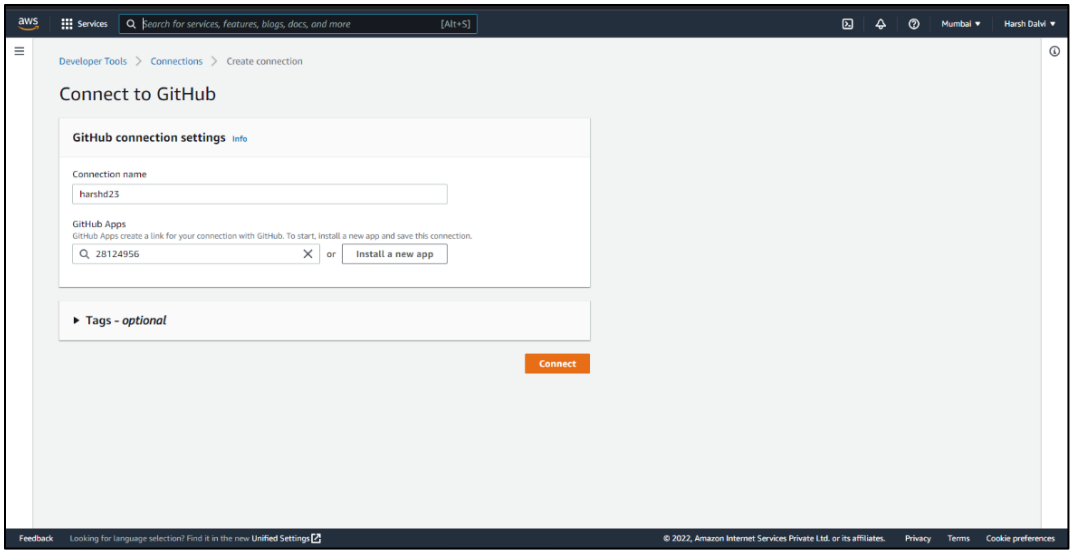


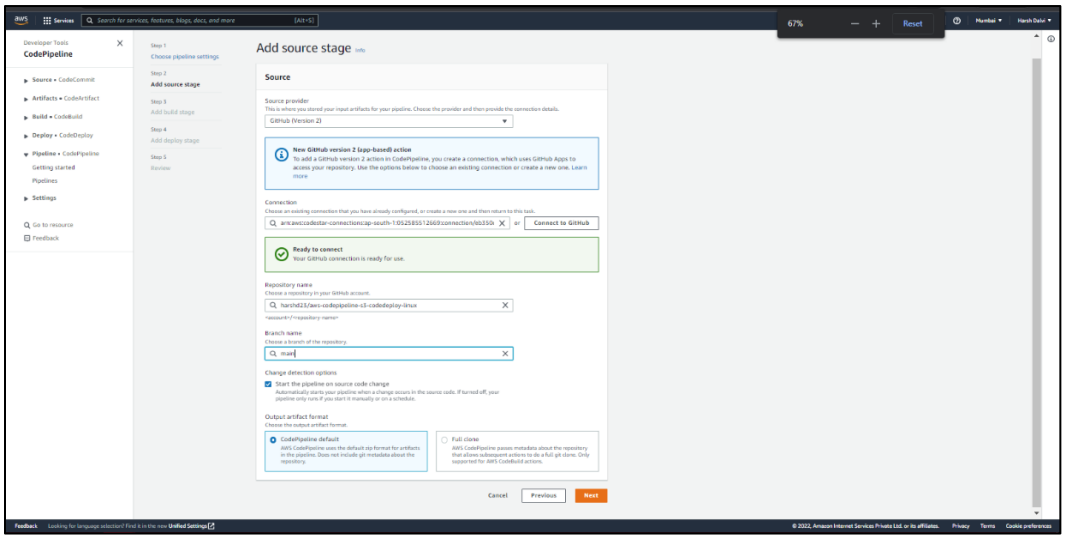




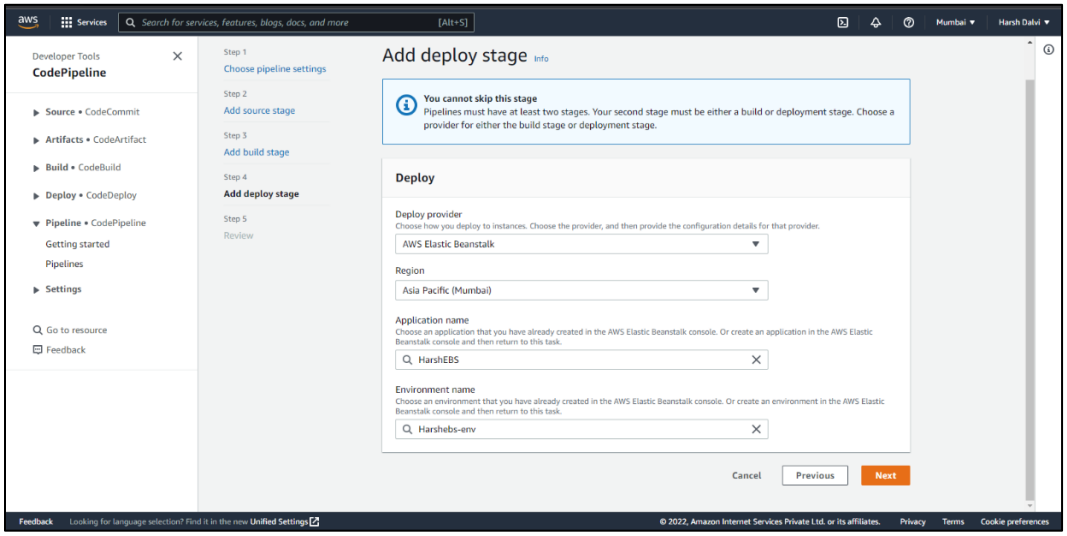
Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B



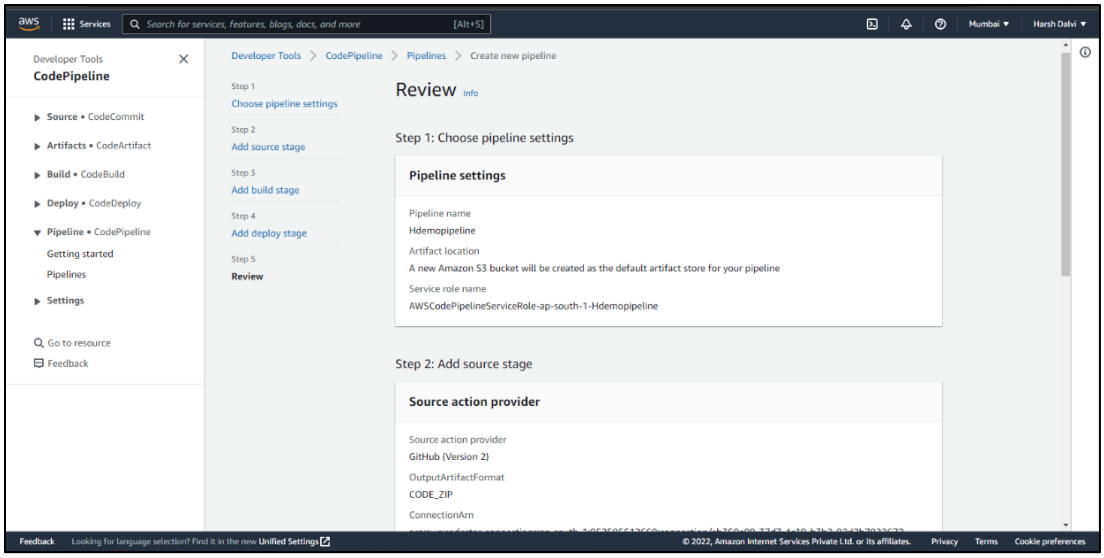


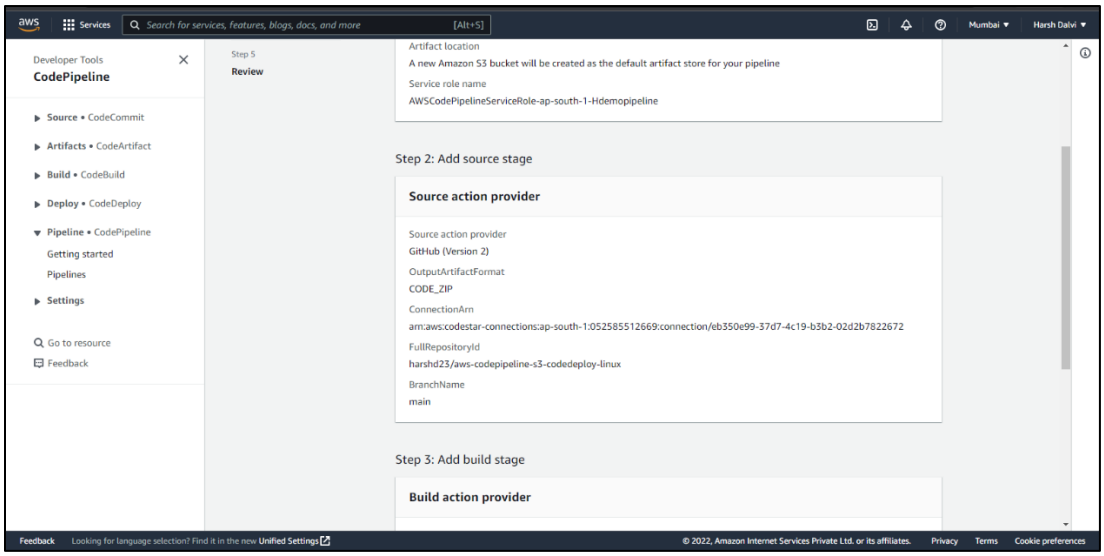
4) Deploy stage

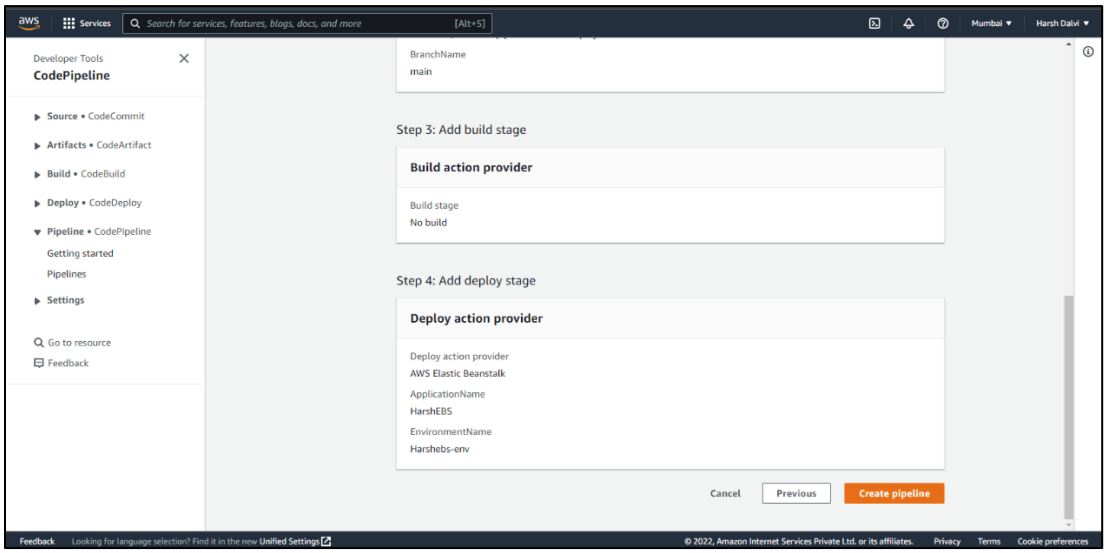


Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

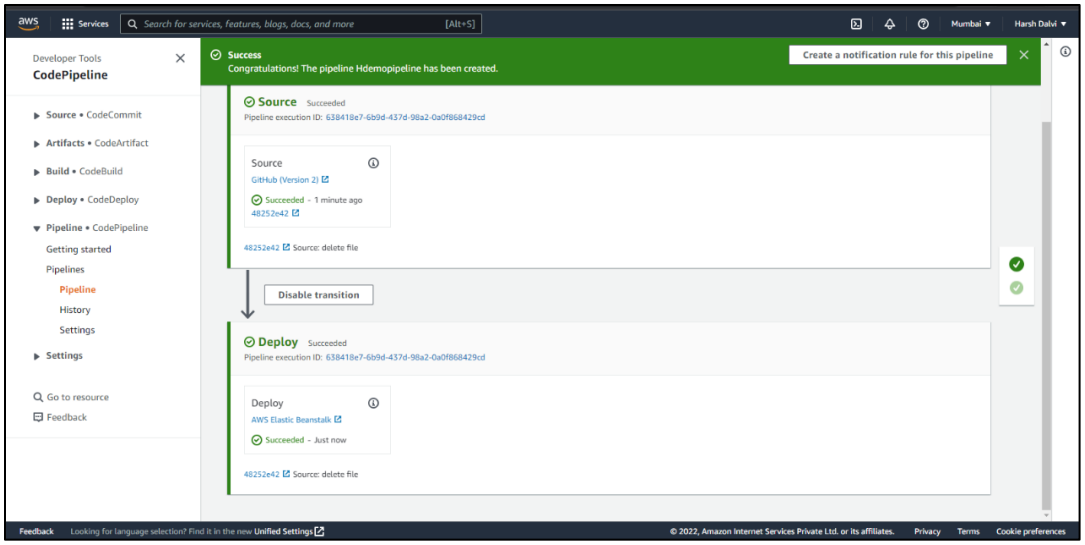


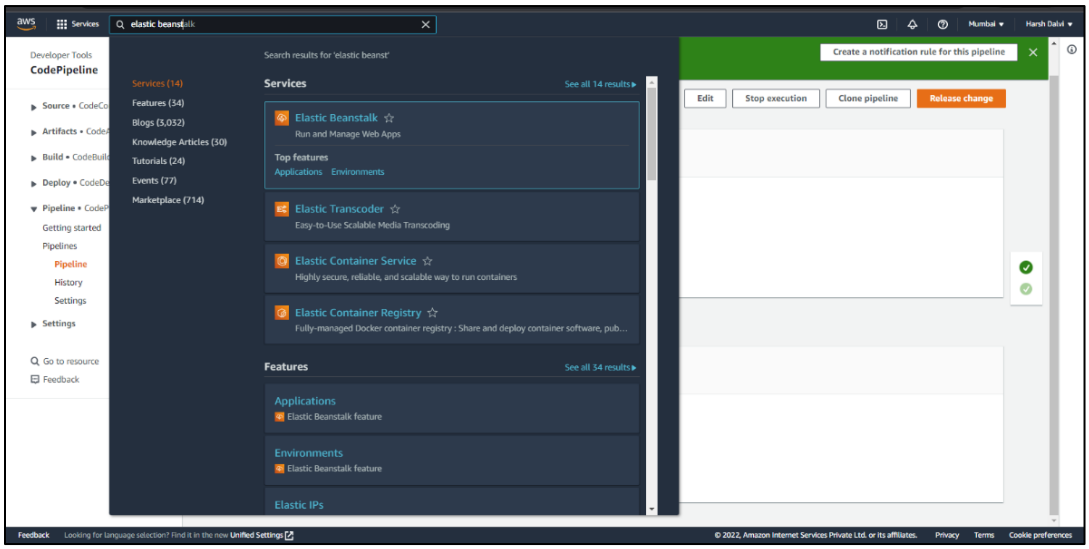


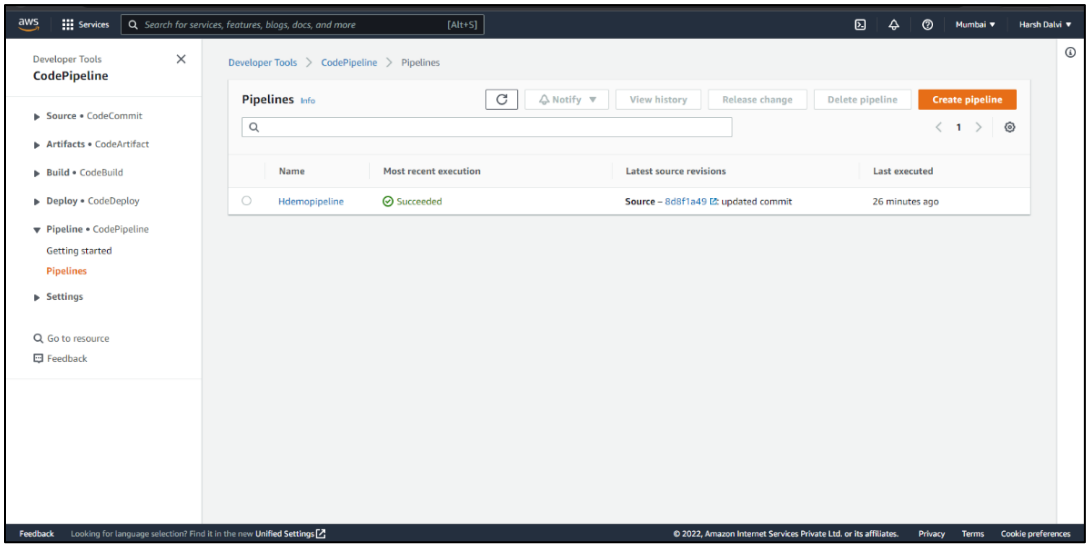


Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B





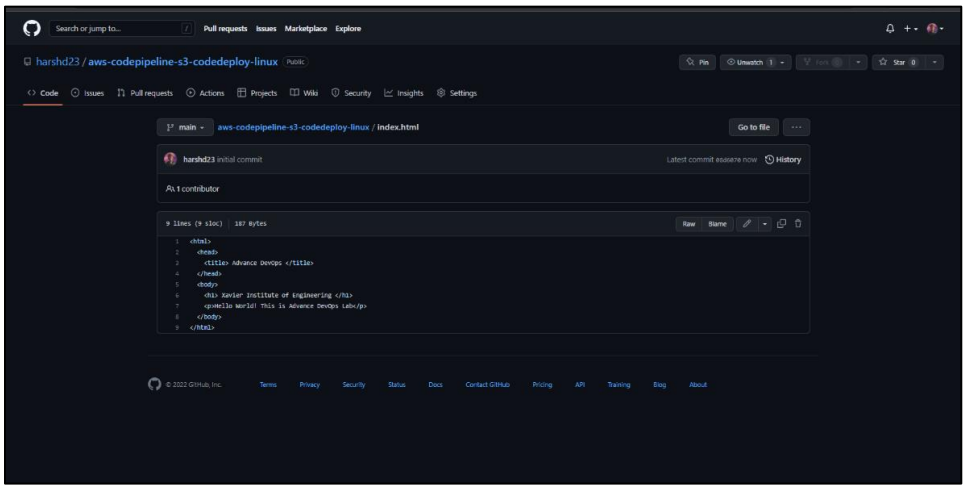


Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

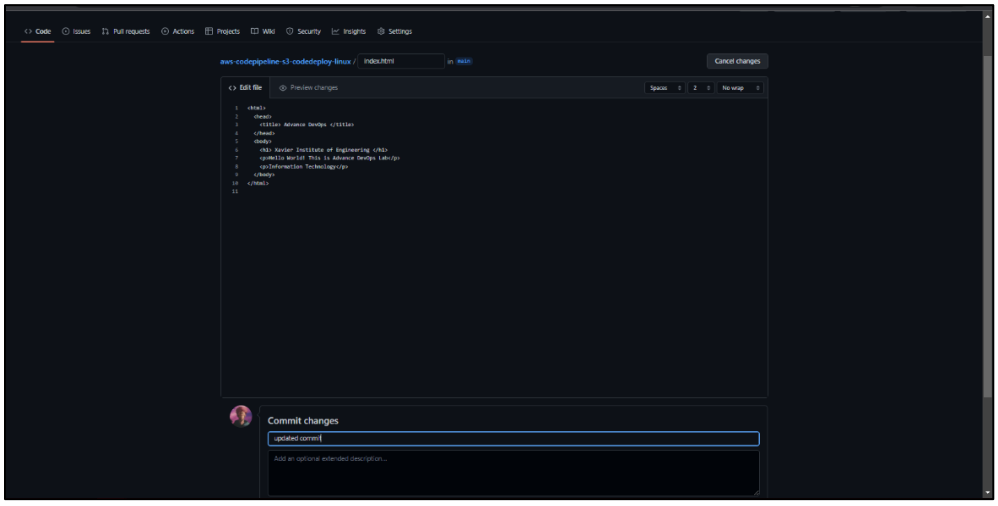
a) Click on the URL to get Output:

Code (present in GitHub Repository):



b) Output(on AWS screen):

5) Commit a change and turn update your API.



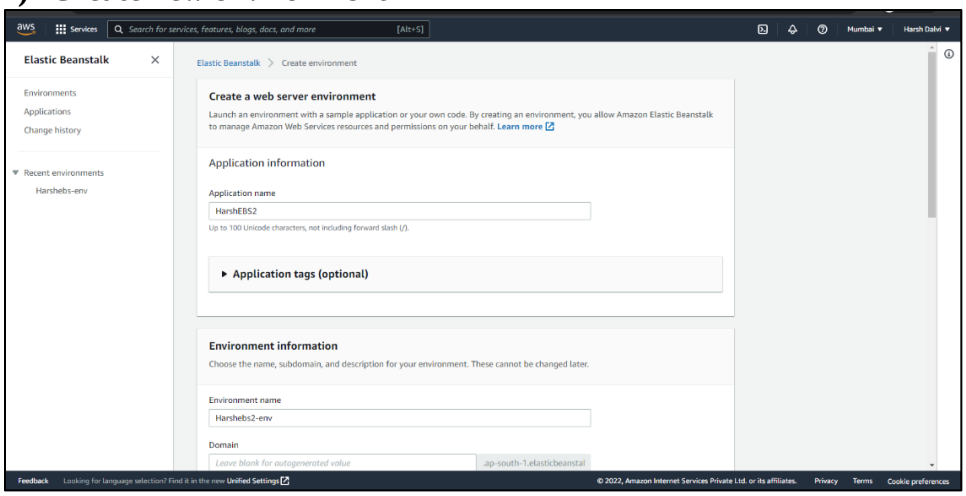
Name: Harsh Dalvi
Roll No: 13

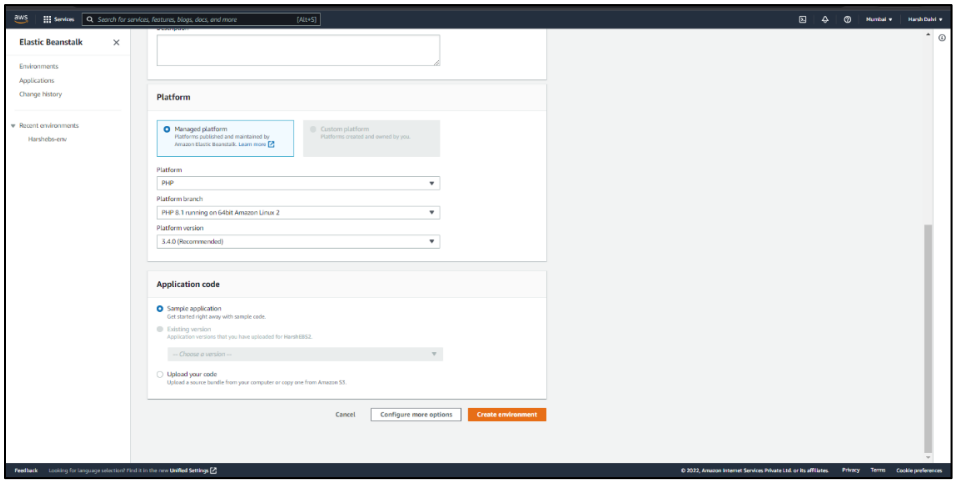
Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

(In the above steps we have created an application using Github. We can create the same application using AWS S3 service.)

Using S3 service

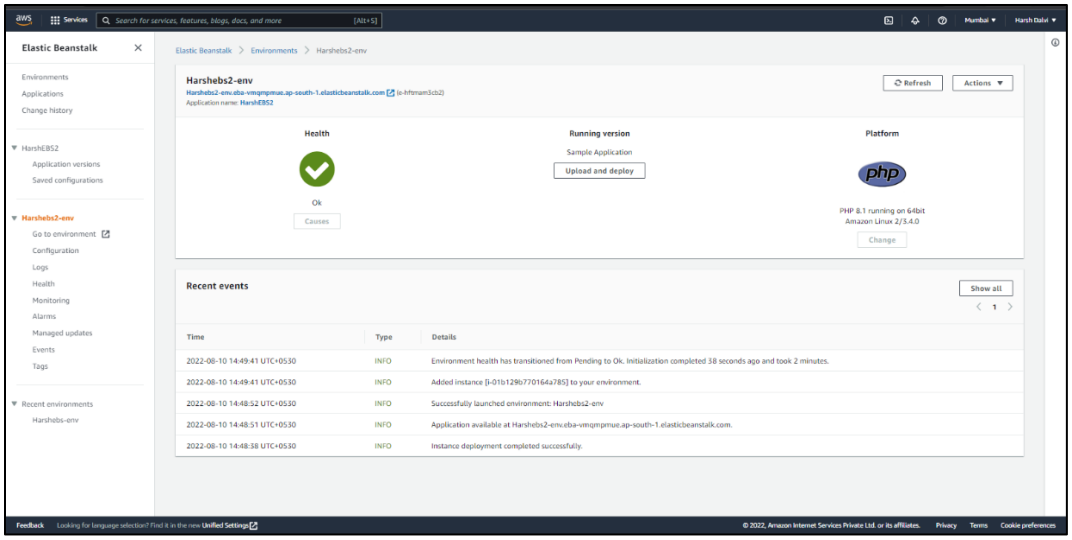
1) Create new environment

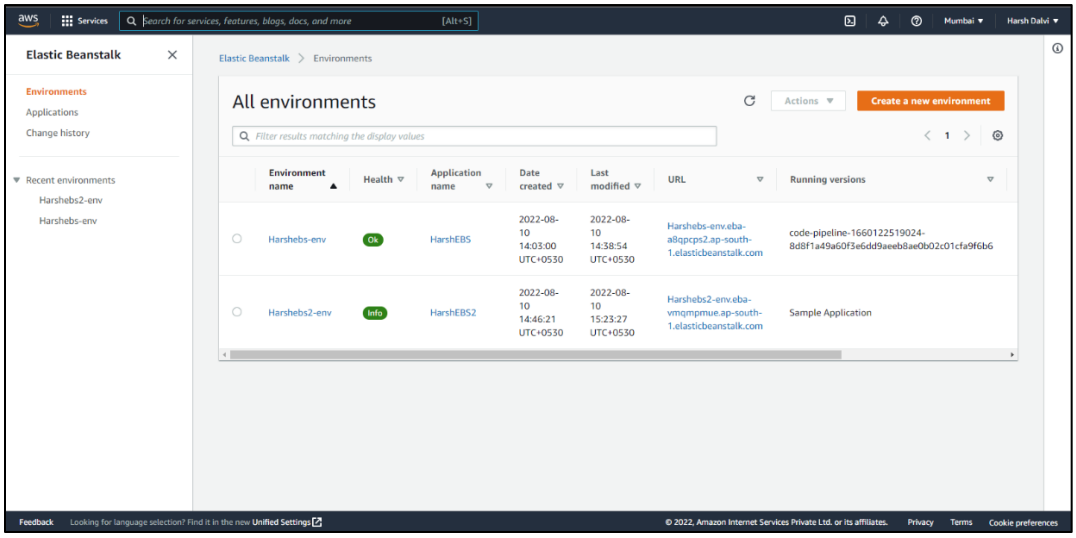




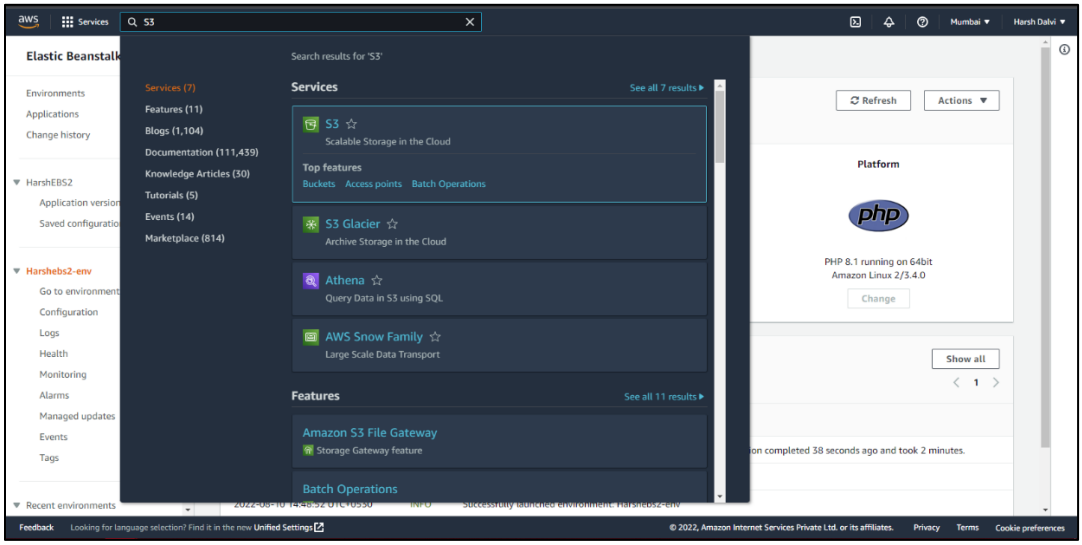
Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B



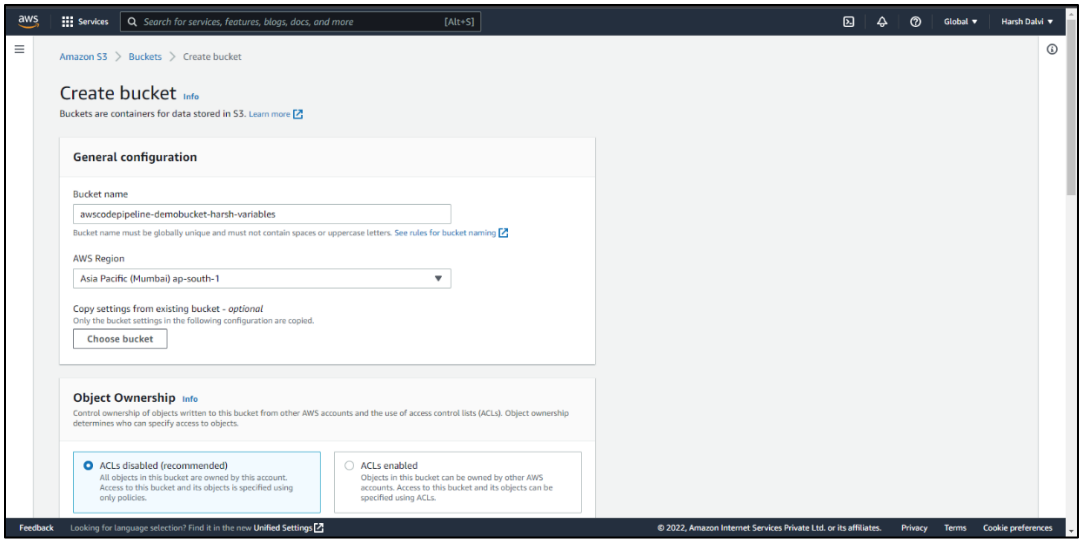


2) Create Buckets

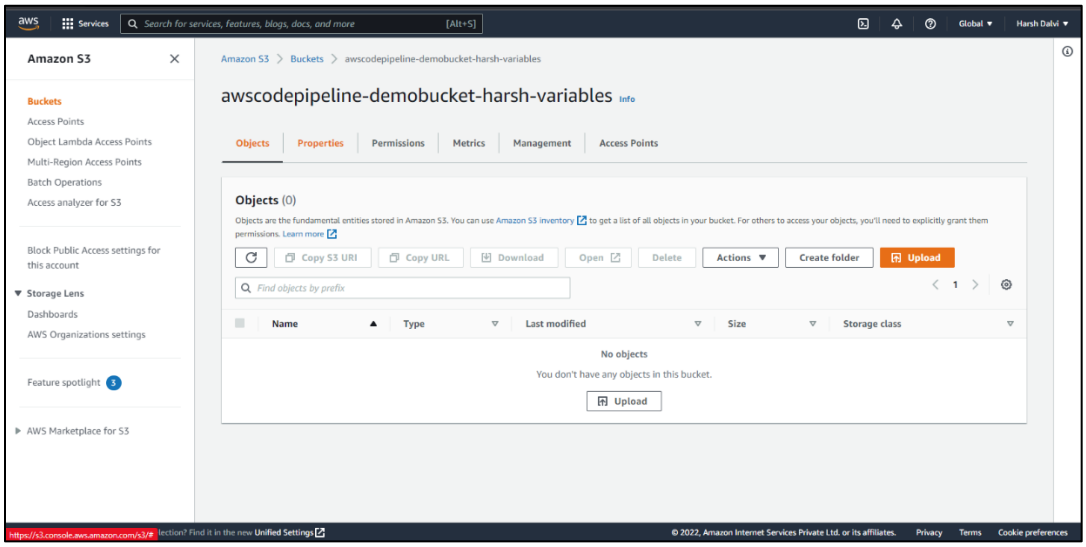


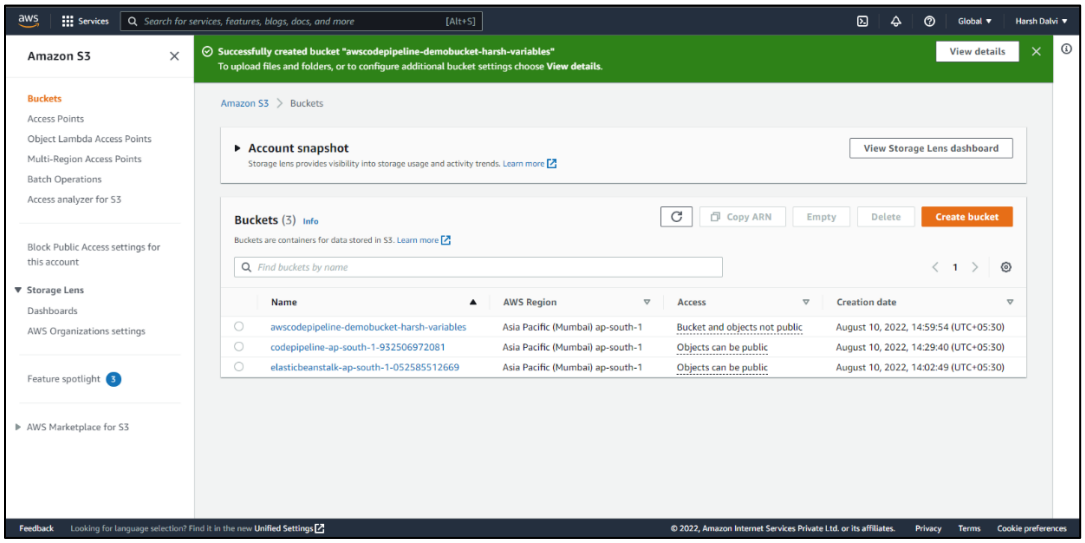
Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B



3) Edit Bucket

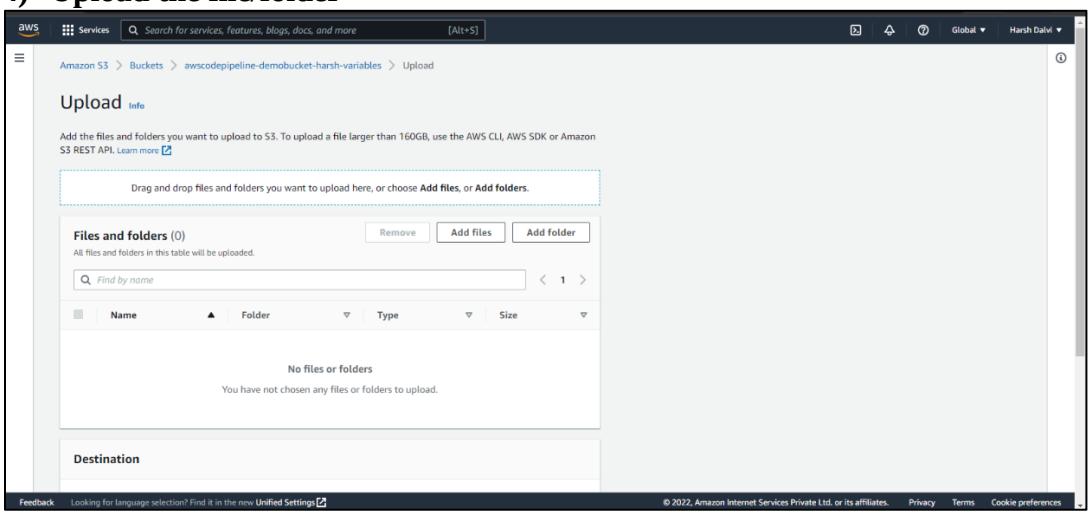


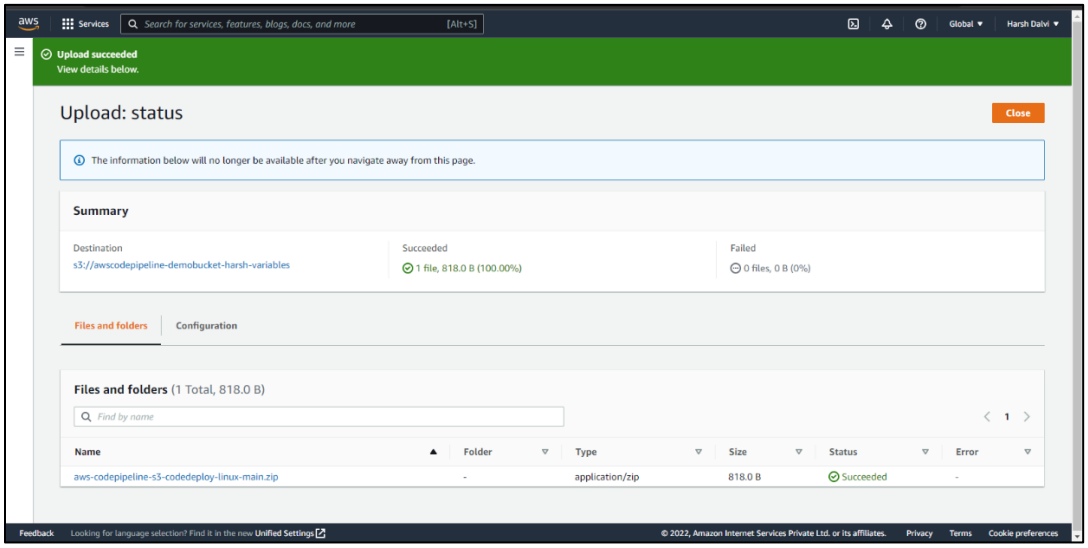


Name: Harsh Dalvi
Roll No: 13

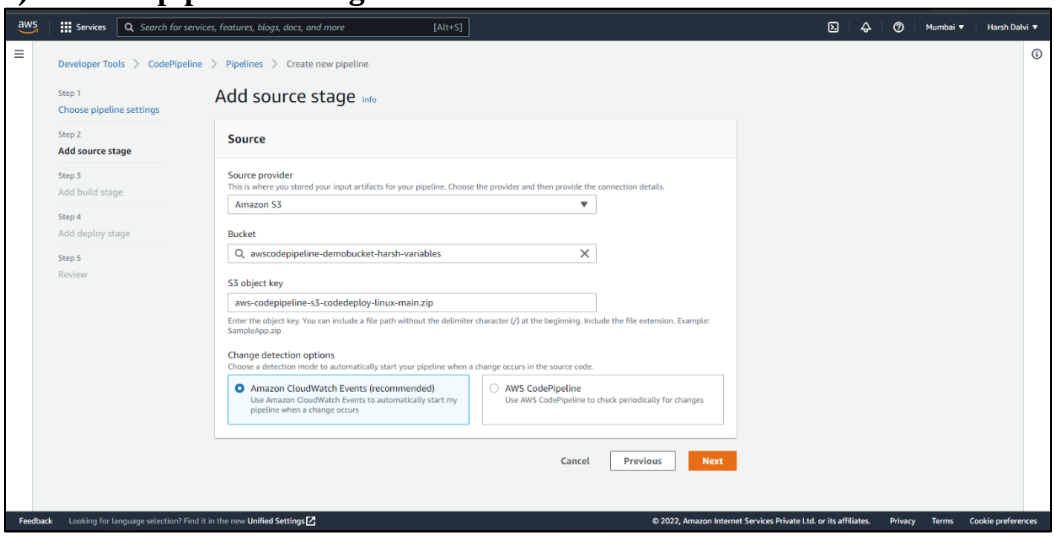
Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

4) Upload the file/folder



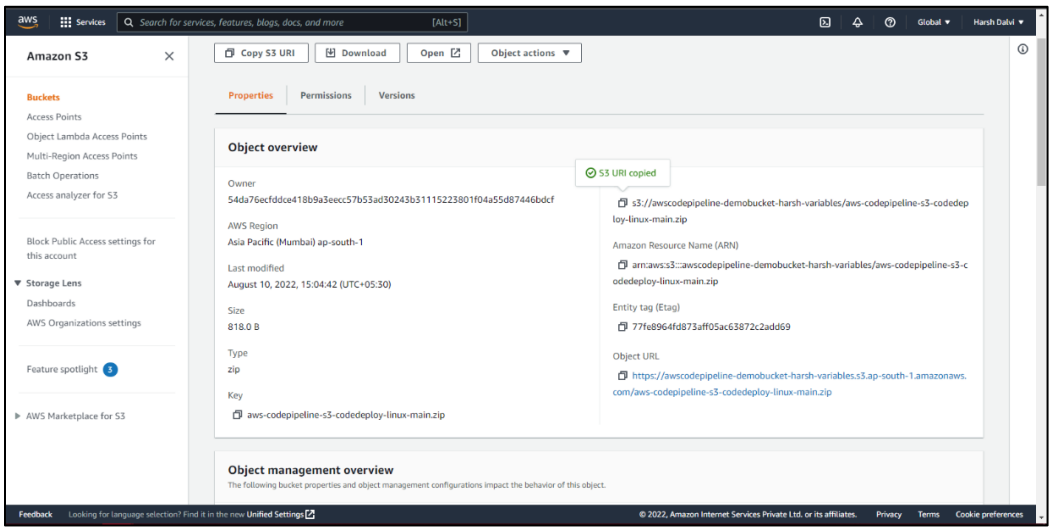


5) Choose pipeline settings

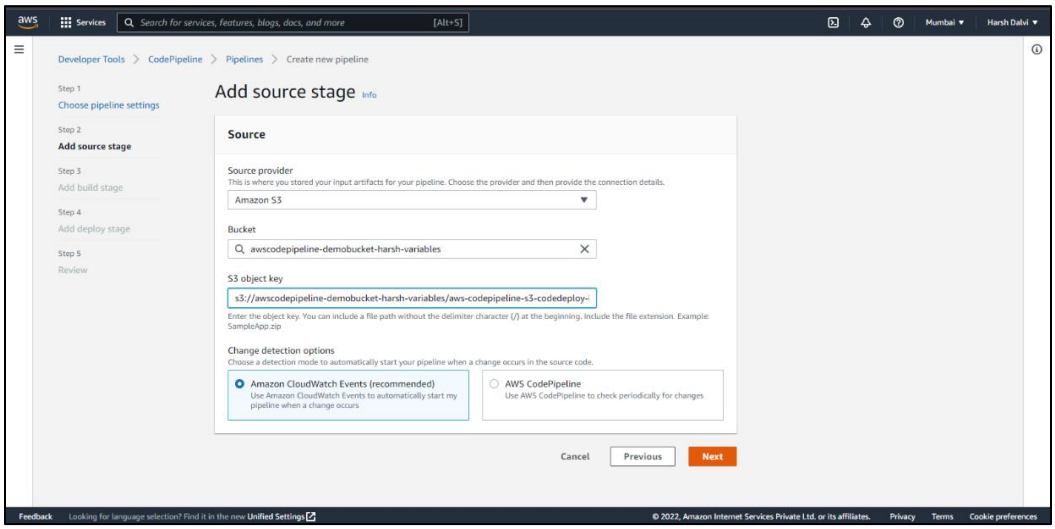


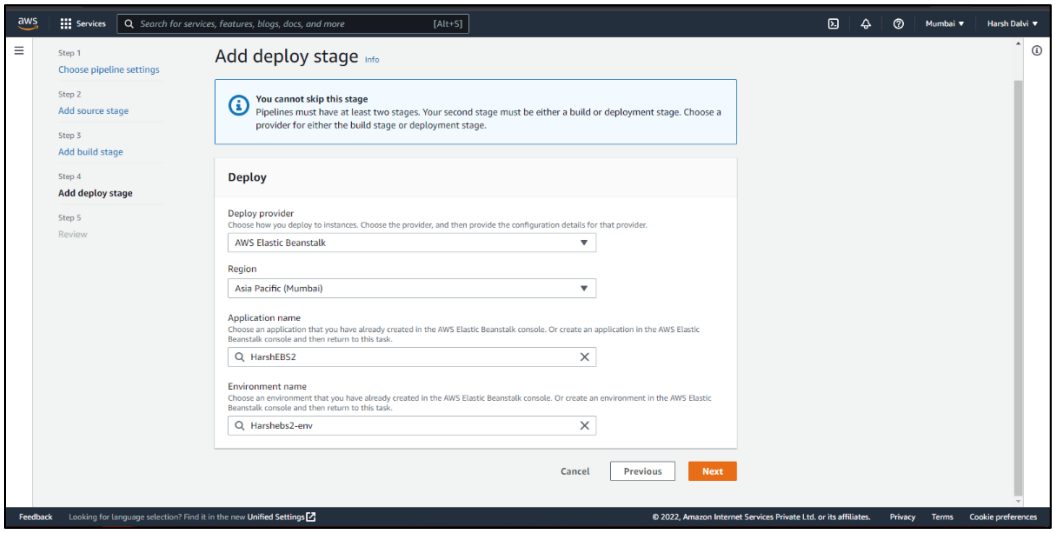
Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B



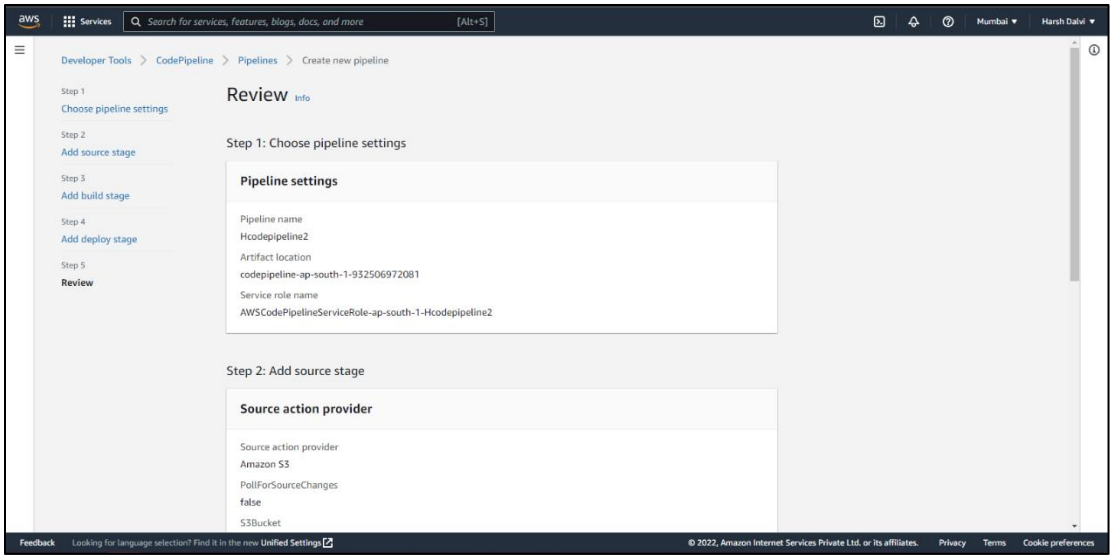
a) Key is the zip_file_name_zip

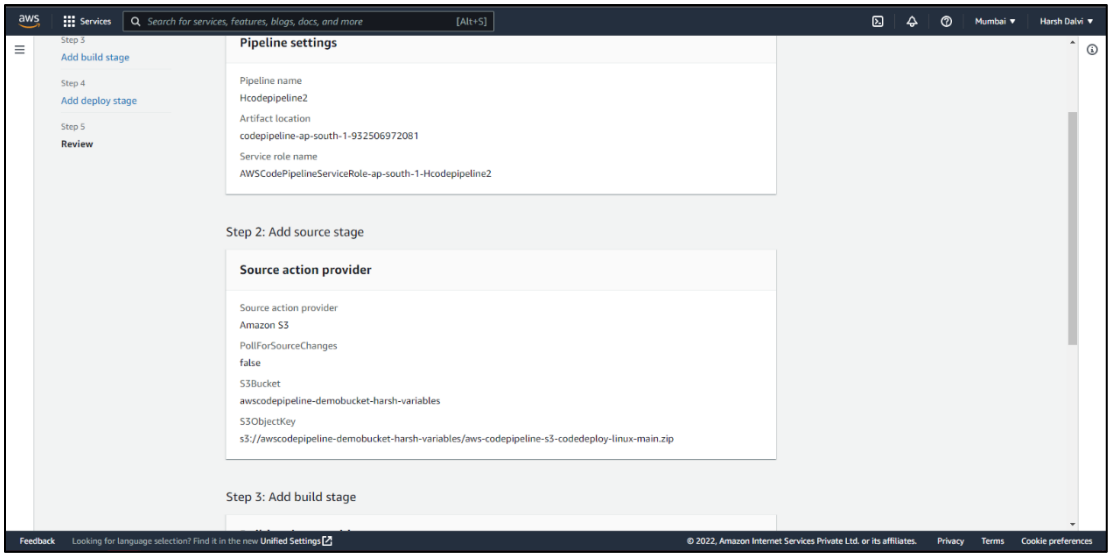


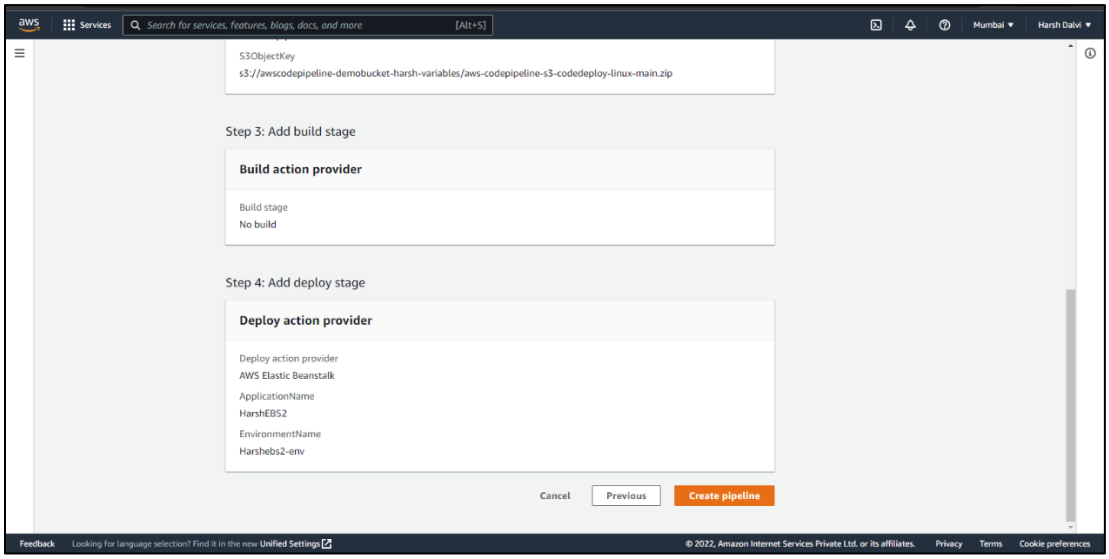


Name: Harsh Dalvi
Roll No: 13

Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

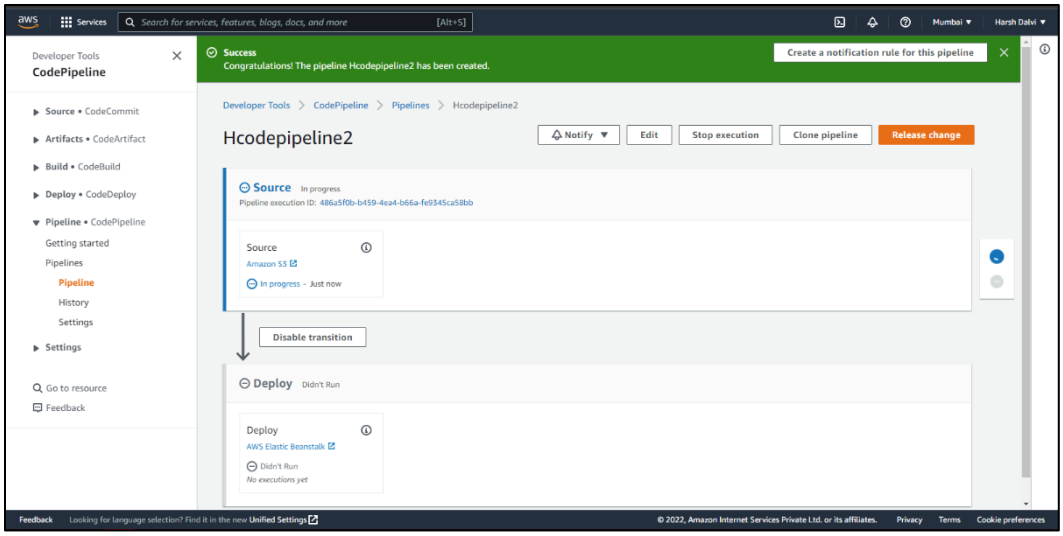




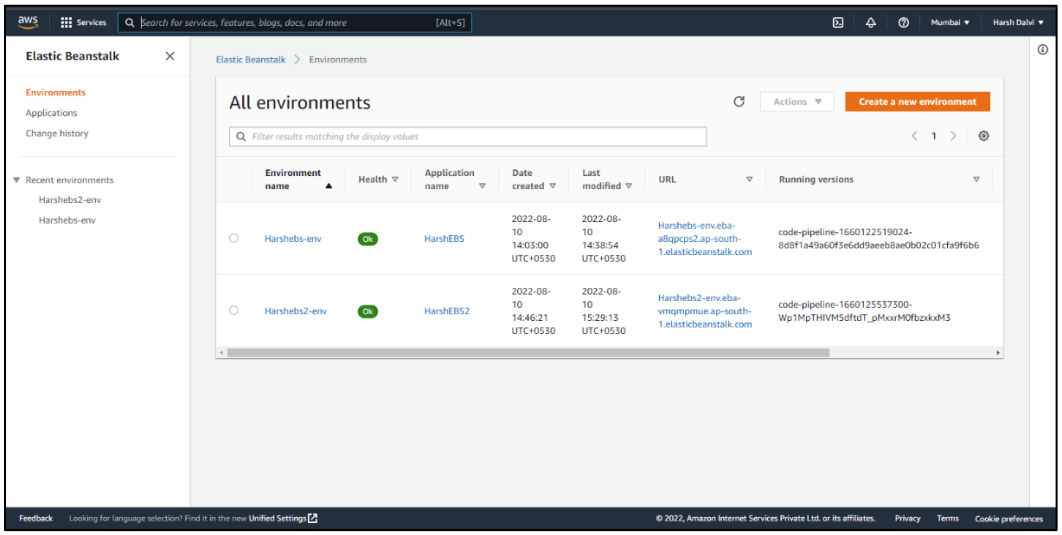


Name: Harsh Dalvi
Roll No: 13

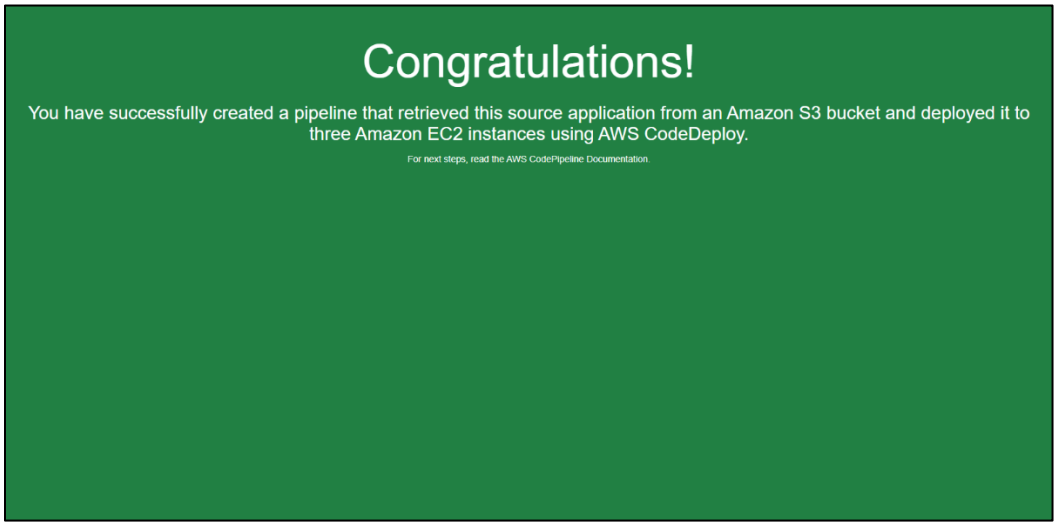
Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B



b) Go to elastic bean stalk



c) Click on environment URL

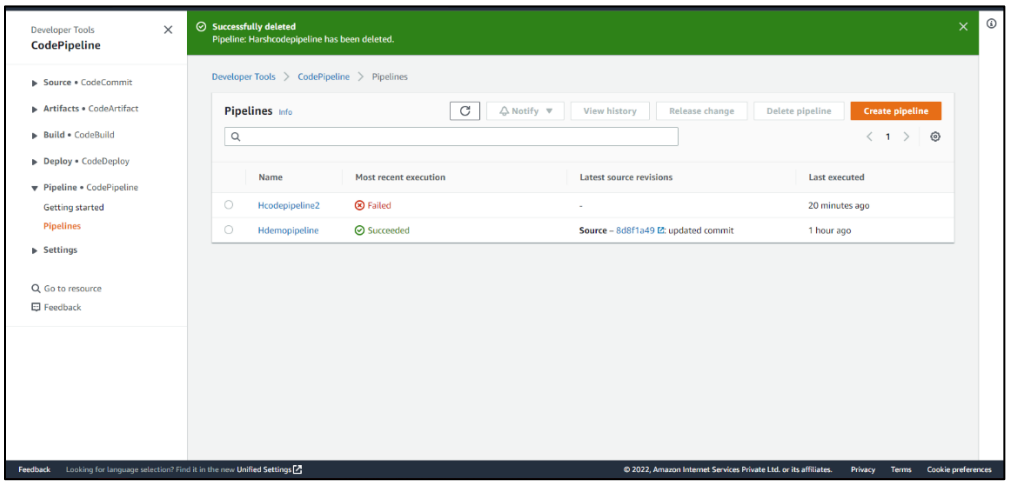


Name: Harsh Dalvi
Roll No: 13

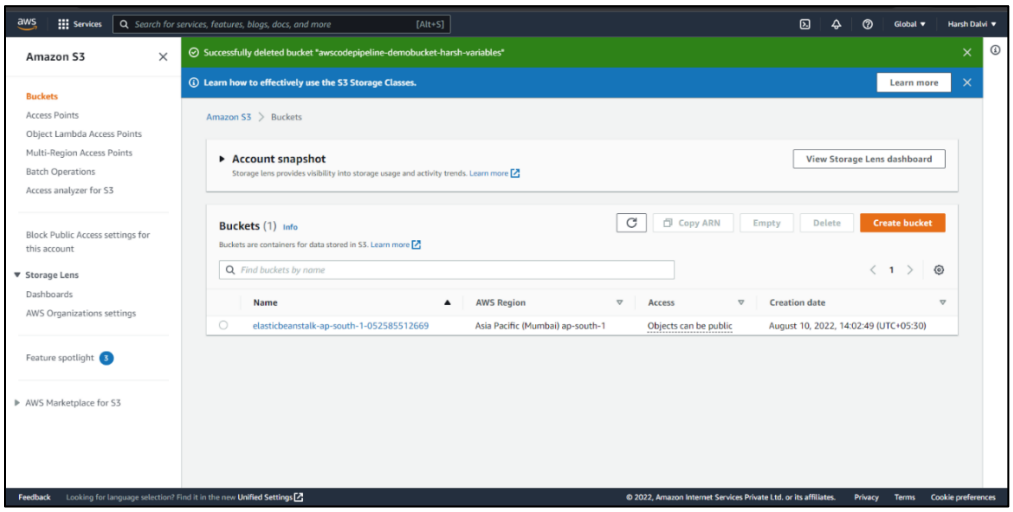
Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

6) Clean up your resources.

a) Delete Pipelines



b) Empty and Delete Buckets



Conclusion: From the above experiment, it is concluded that we build an application using AWS CodeBuild and Deploy on S3 / SEBS using AWS CodePipeline. Also, We have deployed sample Application on EC2 instance using AWS CodeDeploy. Hence, we have successfully achieved Lab Outcome 1 (LO1).