

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.

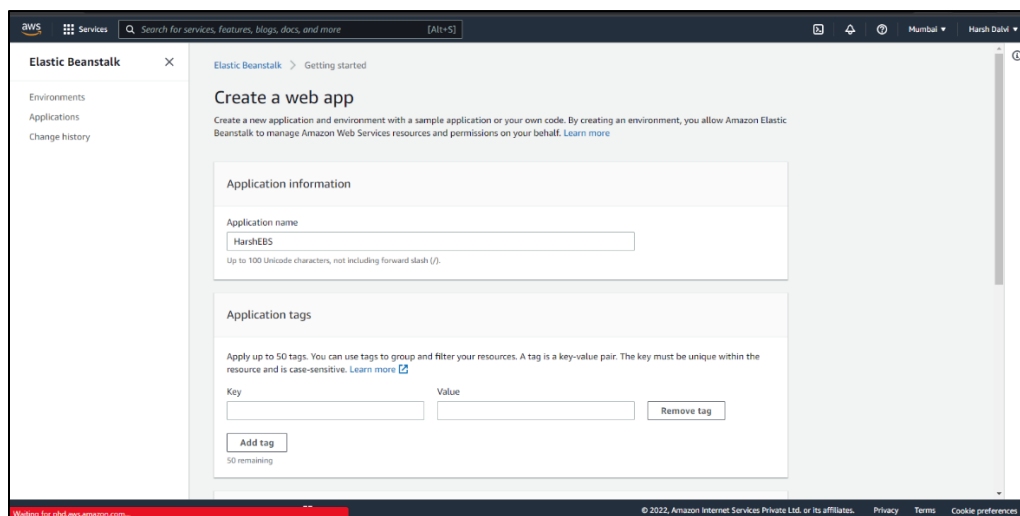
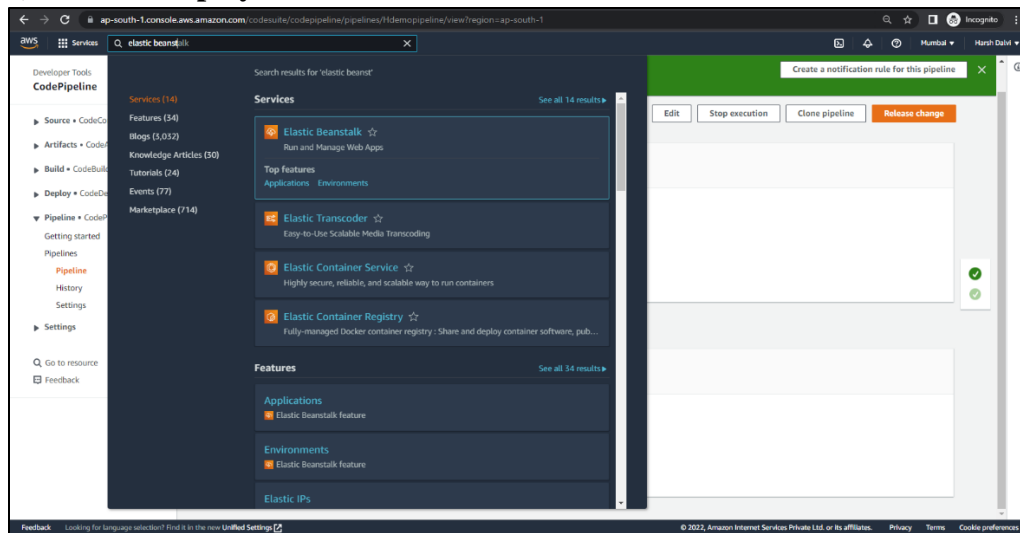
Name: Harsh Dalvi
Roll No: 13

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

- 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.

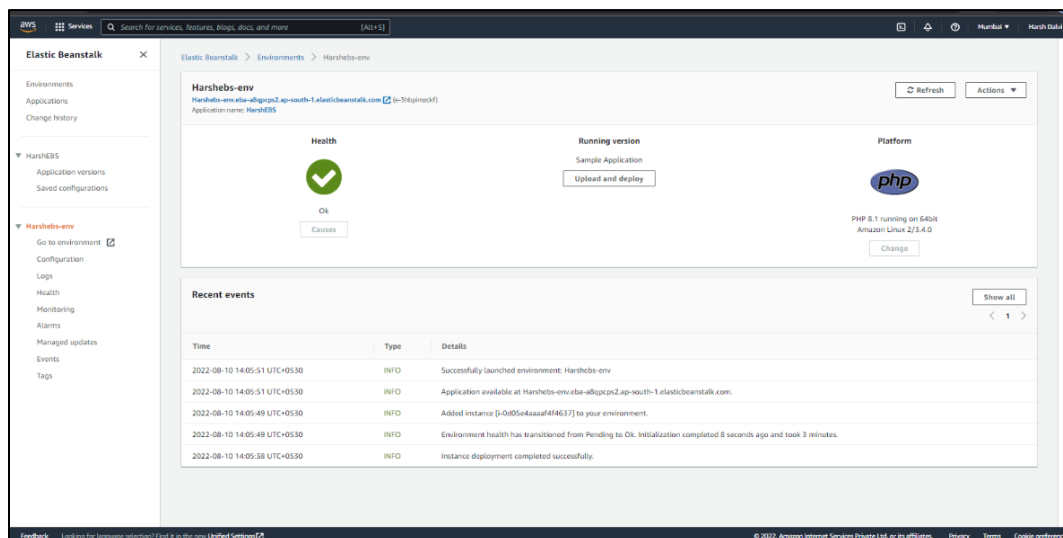
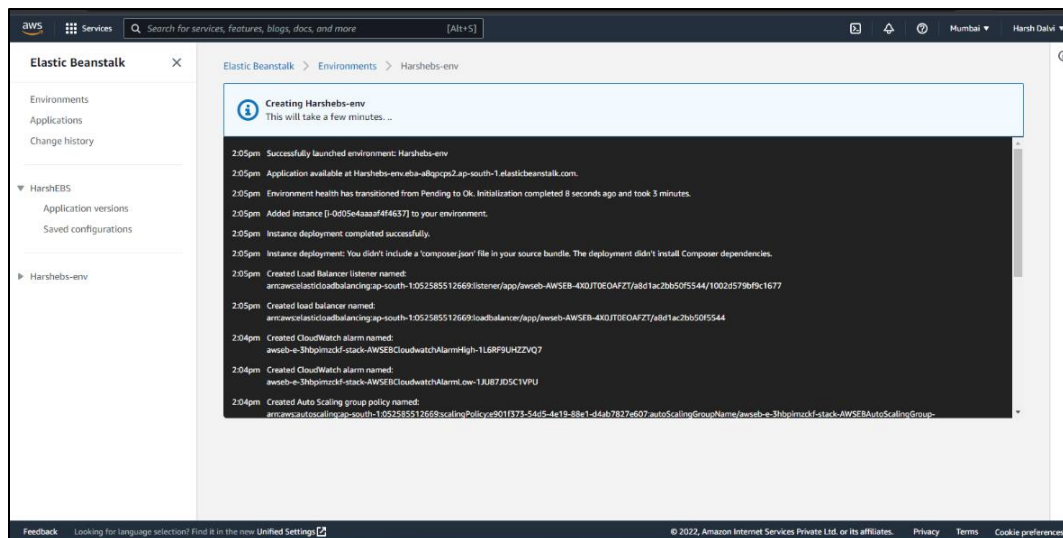
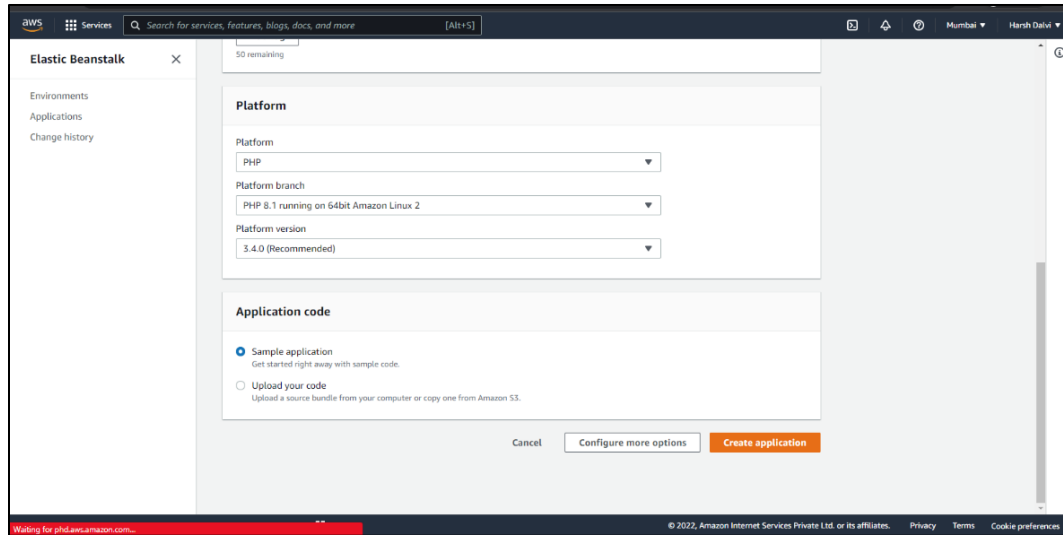
Using GitHub

1) Create a deployment environment.



Name: Harsh Dalvi
Roll No: 13

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

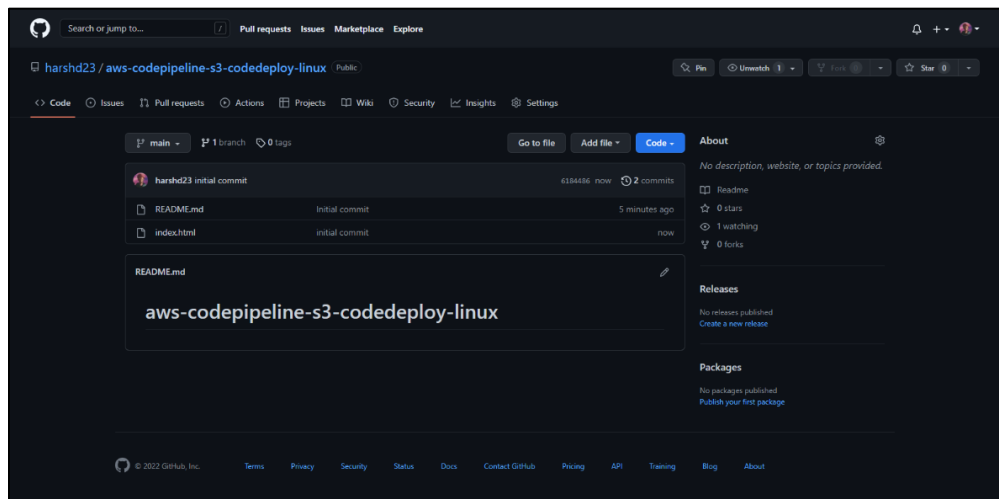
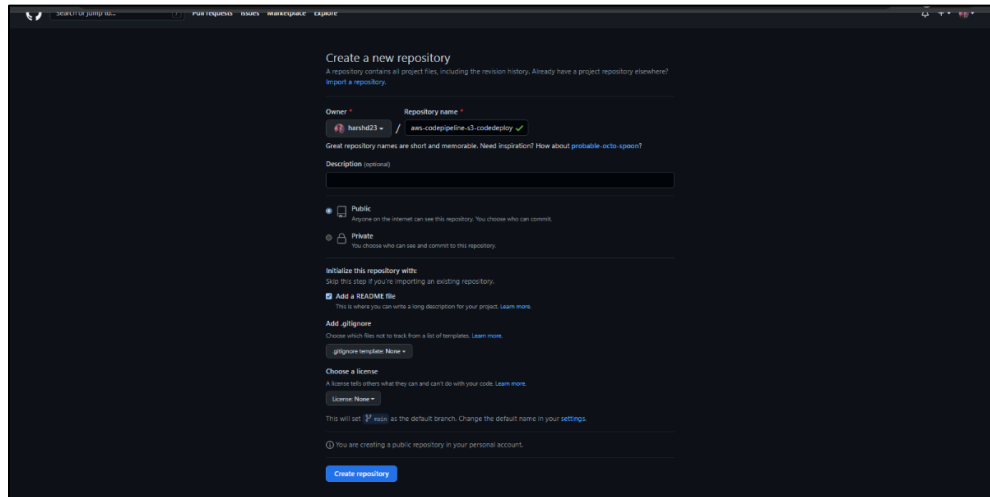


Name: Harsh Dalvi
Roll No: 13

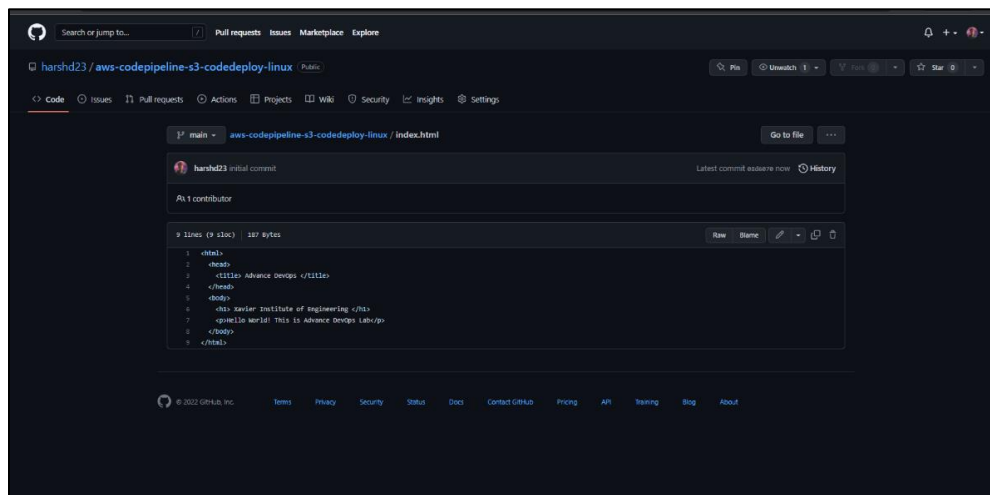
Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

2) Get a copy of a sample code.

a) Go to Github and create new repository.

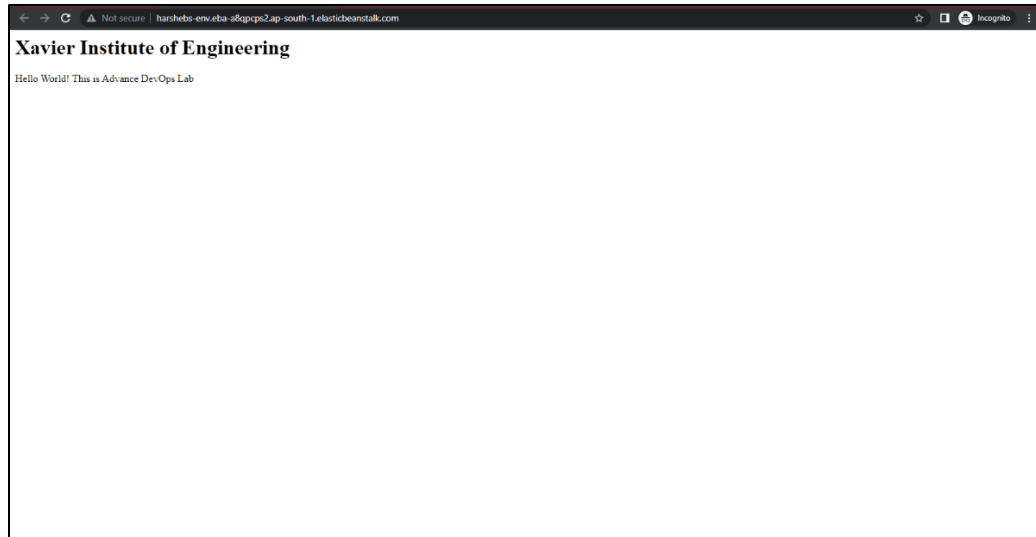


b) Create a new file with name index.html and write a sample html code. Code:

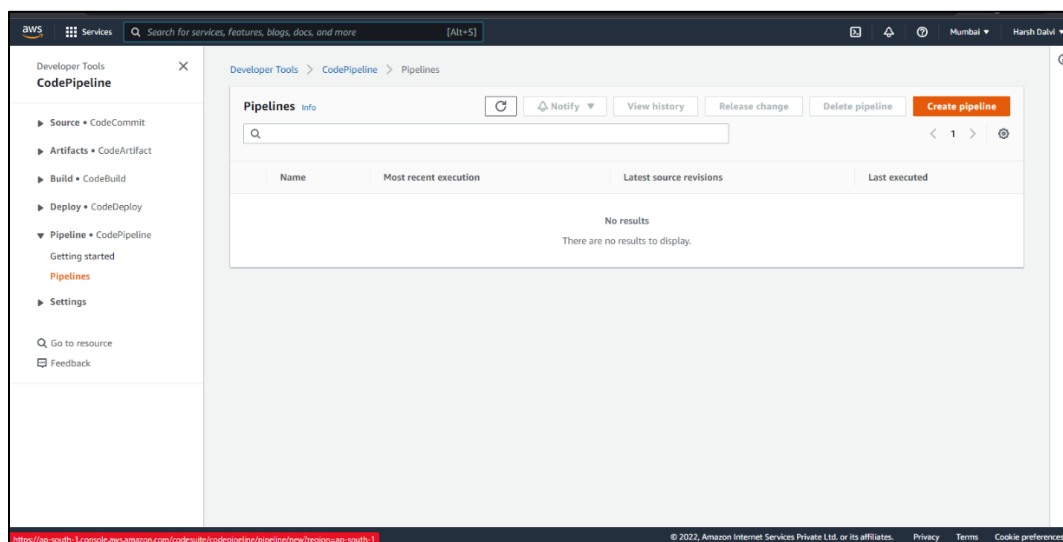
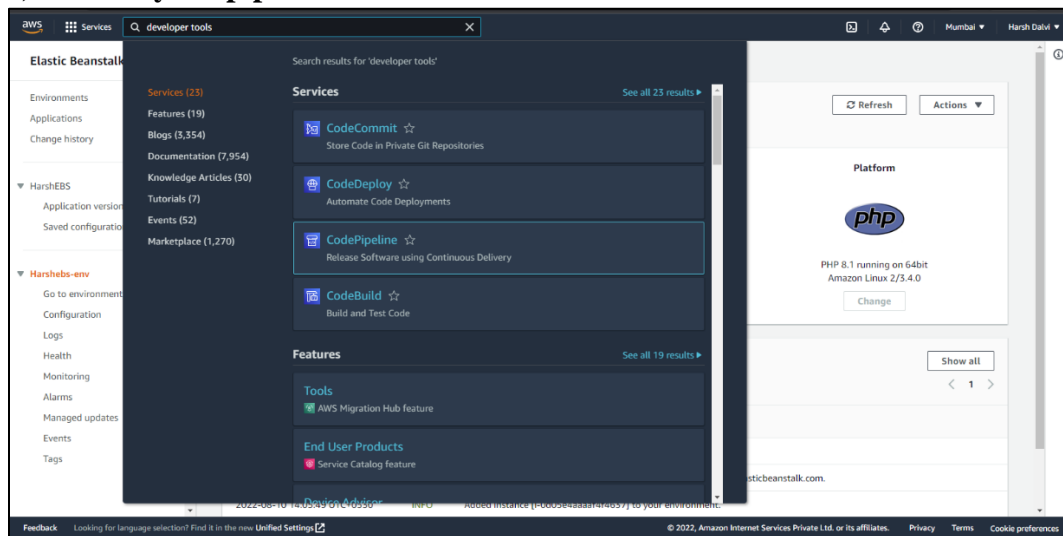


Name: Harsh Dalvi
Roll No: 13

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

The screenshot shows the AWS CodePipeline console interface. The breadcrumb navigation is 'Developer Tools > CodePipeline > Pipelines > Create new pipeline'. The left sidebar shows a five-step process: Step 1: Choose pipeline settings (active), Step 2: Add source stage, Step 3: Add build stage, Step 4: Add deploy stage, and Step 5: Review. The main content area is titled 'Choose pipeline settings' with an 'Info' link. It contains a 'Pipeline settings' section with the following fields: 'Pipeline name' (Hdemopipeline), 'Service role' (New service role selected), and 'Role name' (AWSCodePipelineServiceRole-ap-south-1-Hdemopipeline). A checkbox 'Allow AWS CodePipeline to create a service role so it can be used with this new pipeline' is checked. An 'Advanced settings' section is collapsed. At the bottom are 'Cancel' and 'Next' buttons.

The screenshot shows the 'Create a connection' screen in the AWS CodePipeline console. The breadcrumb navigation is 'Developer Tools > ... > Create connection'. The main content area is titled 'Create a connection' with an 'Info' link. Below it is a section 'Create GitHub App connection' with an 'Info' link. The 'Connection name' field contains 'harshd23'. There is a 'Tags - optional' section which is currently empty. A 'Connect to GitHub' button is located at the bottom right. The footer includes 'Feedback', 'Change language', 'Privacy', 'Terms', 'Cookie preferences', and a copyright notice for 2022.

The screenshot shows a GitHub interface with a dark theme. At the top, there is a search bar and navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The main content area displays the AWS logo and the title 'Install AWS Connector for GitHub'. Below this, it says 'Install on your personal account Harsh Dalvi'. There are two radio button options: 'All repositories' (selected) and 'Only select repositories'. Under 'Only select repositories', a 'Select repositories' button is shown, and one repository is listed: 'harshd23/aws-codepipeline-s3-codedeploy-linux'. Below the repository list, it says 'with these permissions:' and lists two permissions: 'Read access to issues and metadata' and 'Read and write access to administration, code, and pull requests'. At the bottom are 'Install' and 'Cancel' buttons. A small note at the very bottom says 'Next you'll be directed to the GitHub App's site to complete setup.'

Name: Harsh Dalvi
Roll No: 13

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

The screenshot shows the AWS CodePipeline console with the 'Connect to GitHub' dialog open. The dialog has a 'Connection name' field with the value 'harshd23'. Below it, there is a section for 'GitHub Apps' with a search bar containing '28124956' and a button 'Install a new app'. At the bottom right of the dialog is a 'Connect' button. The background shows the 'Developer Tools > Connections > Create connection' breadcrumb and the AWS Services search bar.

The screenshot shows the AWS CodePipeline console with the 'Add source stage' dialog open. The dialog has a 'Source' section with a dropdown for 'GitHub (Version 2)'. Below this is a 'Connection' section with a search bar and a 'Connect to GitHub' button. There is also a 'Repository name' field with the value 'harshd23-dev-codpipeline-ci-coddeploy-branch' and a 'Branch name' field with the value 'main'. At the bottom right of the dialog are 'Cancel', 'Previous', and 'Next' buttons. The background shows the 'Developer Tools > CodePipeline > Add source stage' breadcrumb and the AWS Services search bar.

4) Deploy stage

The screenshot shows the AWS CodePipeline console with the 'Add deploy stage' dialog open. The dialog has a 'Deploy' section with a 'Deploy provider' dropdown set to 'AWS Elastic Beanstalk'. Below this are 'Region' (Asia Pacific (Mumbai)) and 'Application name' (HarshEBS) fields. At the bottom right of the dialog are 'Cancel', 'Previous', and 'Next' buttons. The background shows the 'Developer Tools > CodePipeline > Add deploy stage' breadcrumb and the AWS Services search bar.

Name: Harsh Dalvi
Roll No: 13

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

The screenshot shows the AWS CodePipeline console interface. On the left, the 'Developer Tools' sidebar is open, showing the 'CodePipeline' section. The main content area displays the 'Review' step of the 'Create new pipeline' wizard. The 'Step 1: Choose pipeline settings' section is active, showing the following details:

- Pipeline settings**
 - Pipeline name: Hdemopipeline
 - Artifact location: A new Amazon S3 bucket will be created as the default artifact store for your pipeline
 - Service role name: AWSCodePipelineServiceRole-ap-south-1-Hdemopipeline

Below this, the 'Step 2: Add source stage' section is visible, showing the 'Source action provider' section with the following details:

- Source action provider: GitHub (Version 2)
- OutputArtifactFormat: CODE_ZIP
- ConnectionArn: arn:aws:codepipeline:ap-south-1:052585512669:connection/eb350e99-37d7-4c19-b3b2-02d2b7822672
- FullRepositoryId: harshd23/aws-codepipeline-s3-codedeploy-linux
- BranchName: main

The screenshot shows the AWS CodePipeline console interface. On the left, the 'Developer Tools' sidebar is open, showing the 'CodePipeline' section. The main content area displays the 'Review' step of the 'Create new pipeline' wizard. The 'Step 2: Add source stage' section is active, showing the following details:

- Source action provider**
 - Source action provider: GitHub (Version 2)
 - OutputArtifactFormat: CODE_ZIP
 - ConnectionArn: arn:aws:codepipeline:ap-south-1:052585512669:connection/eb350e99-37d7-4c19-b3b2-02d2b7822672
 - FullRepositoryId: harshd23/aws-codepipeline-s3-codedeploy-linux
 - BranchName: main

Below this, the 'Step 3: Add build stage' section is visible, showing the 'Build action provider' section with the following details:

- Build stage: No build

The screenshot shows the AWS CodePipeline console interface. On the left, the 'Developer Tools' sidebar is open, showing the 'CodePipeline' section. The main content area displays the 'Review' step of the 'Create new pipeline' wizard. The 'Step 3: Add build stage' section is active, showing the following details:

- Build action provider**
 - Build stage: No build

Below this, the 'Step 4: Add deploy stage' section is visible, showing the 'Deploy action provider' section with the following details:

- Deploy action provider: AWS Elastic Beanstalk
- ApplicationName: HarshEBS
- EnvironmentName: HarshEBS-env

At the bottom of the console, there are three buttons: 'Cancel', 'Previous', and 'Create pipeline'.

Name: Harsh Dalvi
Roll No: 13

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

The screenshot shows the AWS CodePipeline console. A green banner at the top states: "Success Congratulations! The pipeline Hdemopipeline has been created." Below this, the pipeline stages are listed:

- Source** (Succeeded): Pipeline execution ID: 638419e7-6b9d-437d-98a2-0a0f868429cd. Action: GitHub (Version 2). Status: Succeeded - 1 minute ago. 48252e42. Source: delete file.
- Deploy** (Succeeded): Pipeline execution ID: 638419e7-6b9d-437d-98a2-0a0f868429cd. Action: AWS Elastic Beanstalk. Status: Succeeded - Just now. 48252e42. Source: delete file.

A "Disable transition" button is visible between the Source and Deploy stages. The left sidebar shows the "CodePipeline" menu with options like Source, Artifacts, Build, Deploy, Pipeline, and Settings.

The screenshot shows the AWS CodePipeline console with search results for "elastic beanstalk". The search results are displayed in a list:

- Elastic Beanstalk** (Run and Manage Web Apps)
- Elastic Transcoder** (Easy-to-Use Scalable Media Transcoding)
- Elastic Container Service** (Highly secure, reliable, and scalable way to run containers)
- Elastic Container Registry** (Fully-managed Docker container registry: Store and deploy container software, pub...

Below the services, there are sections for "Features" and "Elastic IPs". The right sidebar shows the "CodePipeline" menu with options like Source, Artifacts, Build, Deploy, Pipeline, and Settings.

The screenshot shows the AWS CodePipeline console with the "Hdemopipeline" listed in the "Pipelines" section. The table below shows the pipeline details:

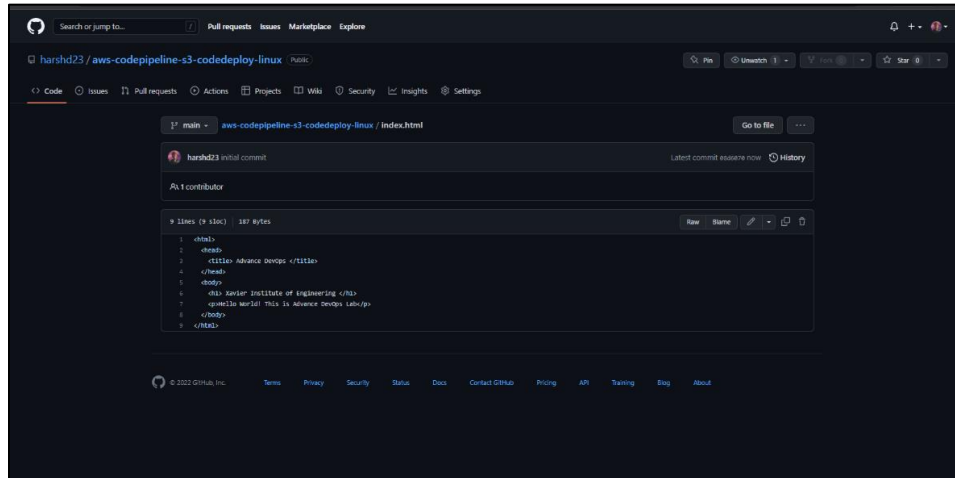
Name	Most recent execution	Latest source revisions	Last executed
Hdemopipeline	Succeeded	Source - 8d9f1a49 updated commit	26 minutes ago

The table has a "Create pipeline" button in the top right corner. The left sidebar shows the "CodePipeline" menu with options like Source, Artifacts, Build, Deploy, Pipeline, and Settings.

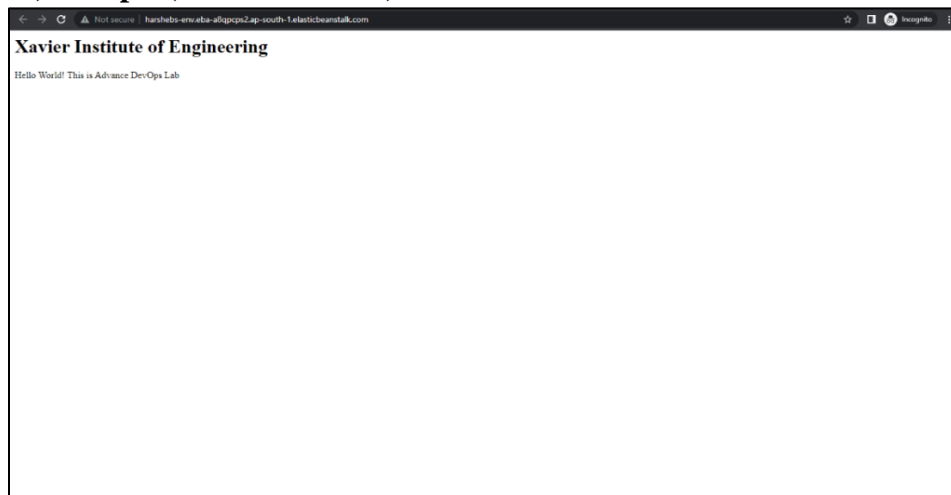
Name: Harsh Dalvi
Roll No: 13

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

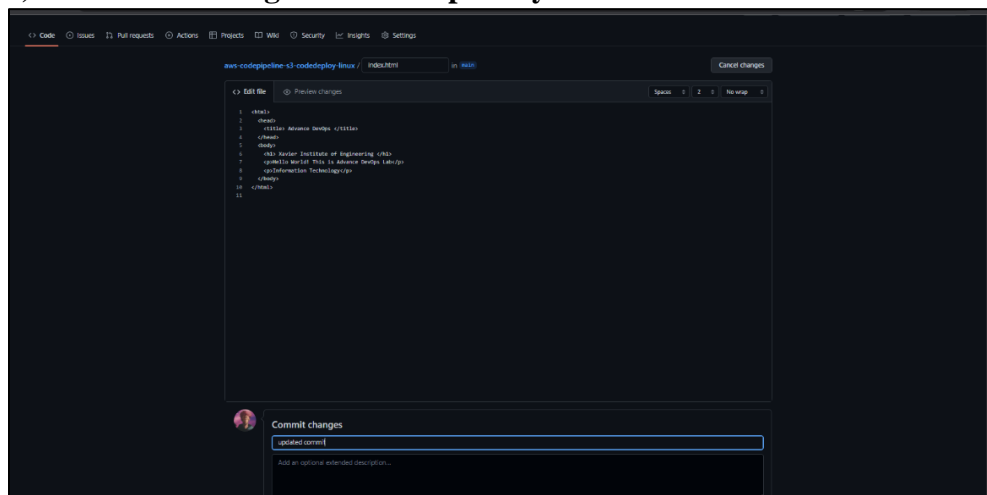
a) Click on the URL to get Output:
Code (present in GitHub Repository):



b) Output(on AWS screen):

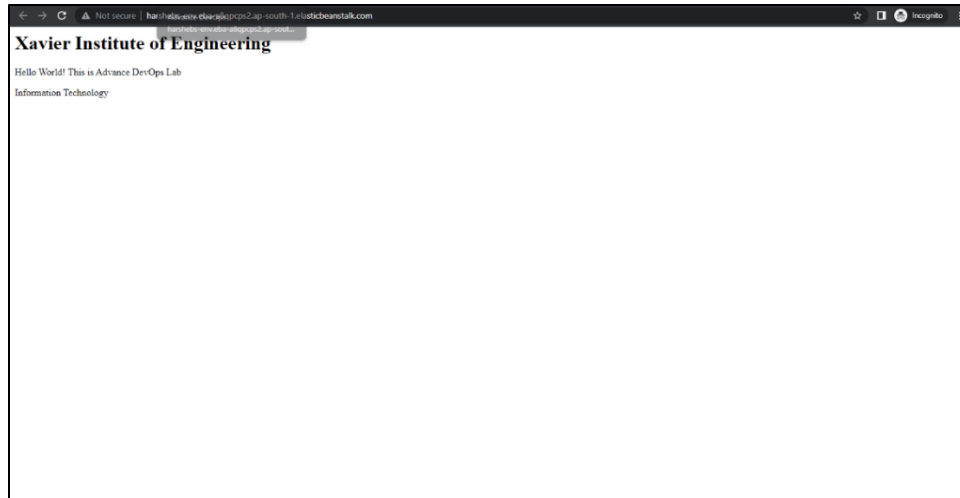


5) Commit a change and turn update your API.



Name: Harsh Dalvi
Roll No: 13

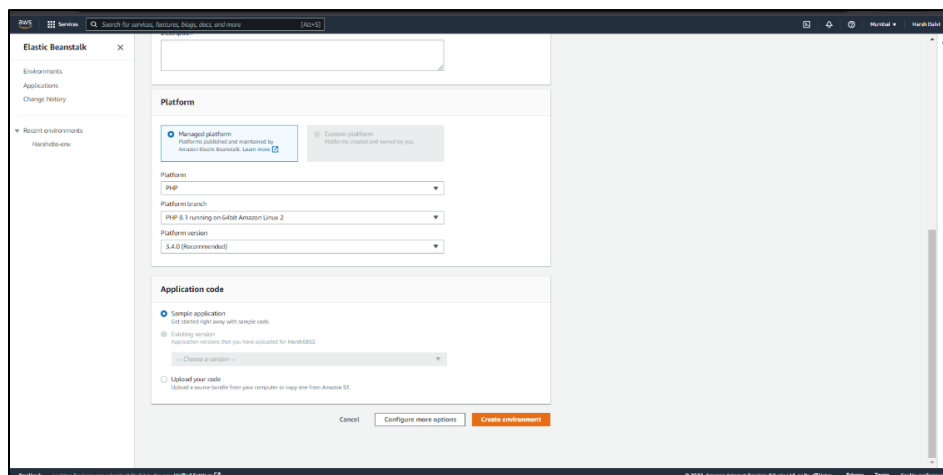
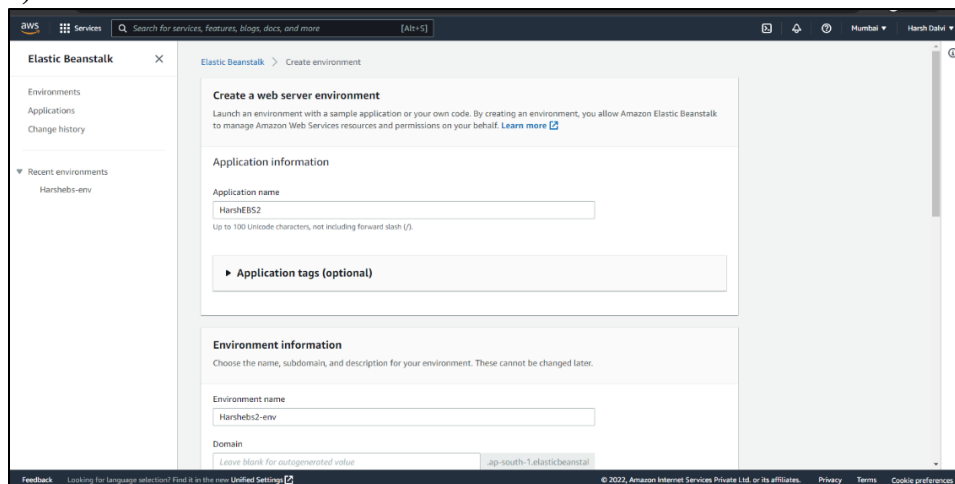
Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B



(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

The screenshot shows the AWS Elastic Beanstalk console. The left sidebar has a search bar and a list of services. The main content area displays the details for the 'HarshEBS2-env' environment. It includes a 'Health' section with a green checkmark and 'OK' status, a 'Running version' section with a 'Sample Application' and an 'Upload and deploy' button, and a 'Platform' section showing 'PHP 8.1 running on 64bit Amazon Linux 2/3.4.0'. Below these sections is a 'Recent events' table with columns for Time, Type, and Details.

Time	Type	Details
2022-08-10 14:49:41 UTC+0530	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 38 seconds ago and took 2 minutes.
2022-08-10 14:49:41 UTC+0530	INFO	Added instance [i-01b1296770164a785] to your environment.
2022-08-10 14:48:52 UTC+0530	INFO	Successfully launched environment: HarshEBS2-env
2022-08-10 14:48:51 UTC+0530	INFO	Application available at HarshEBS2-env.eba-vmgrpmmue.ap-south-1.elasticbeanstalk.com.
2022-08-10 14:48:38 UTC+0530	INFO	Instance deployment completed successfully.

The screenshot shows the AWS Elastic Beanstalk console with the 'All environments' view. It features a search bar, a table of environments, and a 'Create a new environment' button. The table has columns for Environment name, Health, Application name, Date created, Last modified, URL, and Running versions.

Environment name	Health	Application name	Date created	Last modified	URL	Running versions
HarshEBS-env	Ok	HarshEBS	2022-08-10 14:03:00 UTC+0530	2022-08-10 14:38:54 UTC+0530	HarshEBS-env.eba-a8gpcps2.ap-south-1.elasticbeanstalk.com	code-pipeline-1660122519024-8d8f1a49a60f3e6dd9aeeb8ae0b02c01cfa9f6b6
HarshEBS2-env	Info	HarshEBS2	2022-08-10 14:46:21 UTC+0530	2022-08-10 15:23:27 UTC+0530	HarshEBS2-env.eba-vmgrpmmue.ap-south-1.elasticbeanstalk.com	Sample Application

2) Create Buckets

The screenshot shows the AWS console with a search bar containing 'S3'. The search results are displayed in a list of services and features. The 'Services' section includes 'S3', 'S3 Glacier', 'Athena', and 'AWS Snow Family'. The 'Features' section includes 'Amazon S3 File Gateway' and 'Batch Operations'.

Services (7)

- S3** ★ Scalable Storage in the Cloud
Top features: Buckets, Access points, Batch Operations
- S3 Glacier** ★ Archive Storage in the Cloud
- Athena** ★ Query Data in S3 using SQL
- AWS Snow Family** ★ Large Scale Data Transport

Features See all 11 results ▶

- Amazon S3 File Gateway** Storage Gateway feature
- Batch Operations**

Name: Harsh Dalvi
Roll No: 13

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

The screenshot shows the 'Create bucket' page in the AWS Management Console. The breadcrumb navigation is 'Amazon S3 > Buckets > Create bucket'. The page title is 'Create bucket' with an 'Info' link. A sub-header states 'Buckets are containers for data stored in S3. Learn more'. The 'General configuration' section contains a 'Bucket name' text box with the value 'awscodepipeline-demobucket-harsh-variables', an 'AWS Region' dropdown menu set to 'Asia Pacific (Mumbai) ap-south-1', and a 'Choose bucket' button. Below this, the 'Object Ownership' section has two radio buttons: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'. The footer includes a 'Feedback' link, a language selection prompt, and copyright information for Amazon Internet Services Private Ltd.

3) Edit Bucket

The screenshot shows the 'Edit Bucket' page for 'awscodepipeline-demobucket-harsh-variables'. The breadcrumb navigation is 'Amazon S3 > Buckets > awscodepipeline-demobucket-harsh-variables'. The page title is 'awscodepipeline-demobucket-harsh-variables' with an 'Info' link. Below the title are tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab is active, showing 'Objects (0)'. A message states 'Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more'. Below this is a toolbar with icons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. A search bar 'Find objects by prefix' is present. Below the search bar is a table with columns: 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. The table is empty with the message 'No objects. You don't have any objects in this bucket.' and an 'Upload' button. The footer includes a 'Feedback' link, a language selection prompt, and copyright information for Amazon Internet Services Private Ltd.

The screenshot shows the 'Buckets' page in the AWS Management Console. A green banner at the top states 'Successfully created bucket "awscodepipeline-demobucket-harsh-variables"'. Below the banner, the breadcrumb navigation is 'Amazon S3 > Buckets'. The page title is 'Buckets (3)' with an 'Info' link. A sub-header states 'Buckets are containers for data stored in S3. Learn more'. Below this is a toolbar with icons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. A search bar 'Find buckets by name' is present. Below the search bar is a table with columns: 'Name', 'AWS Region', 'Access', and 'Creation date'. The table lists three buckets:

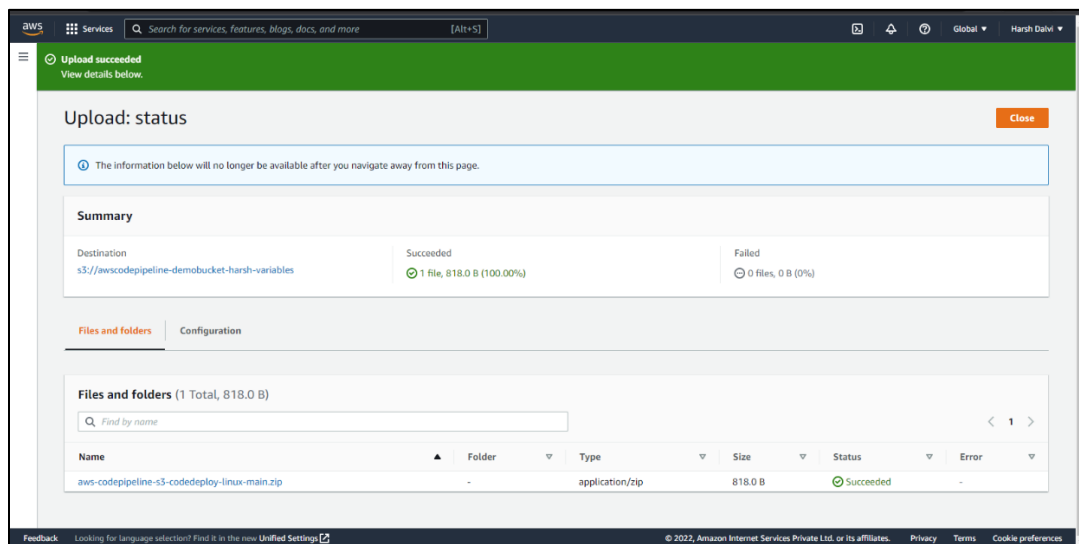
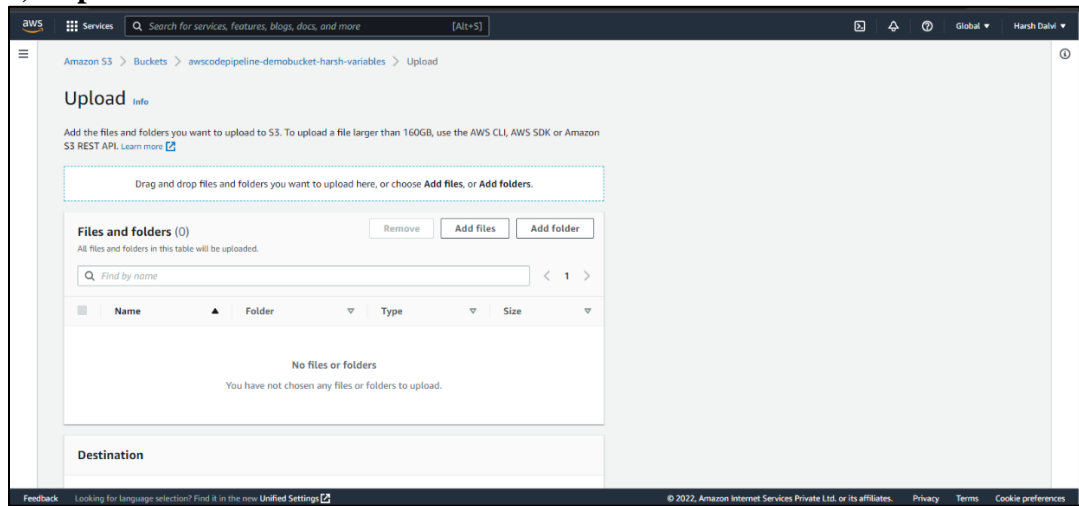
Name	AWS Region	Access	Creation date
awscodepipeline-demobucket-harsh-variables	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	August 10, 2022, 14:59:54 (UTC+05:30)
codepipeline-ap-south-1-932506972081	Asia Pacific (Mumbai) ap-south-1	Objects can be public	August 10, 2022, 14:29:40 (UTC+05:30)
elasticbeanstalk-ap-south-1-052585512669	Asia Pacific (Mumbai) ap-south-1	Objects can be public	August 10, 2022, 14:02:49 (UTC+05:30)

The footer includes a 'Feedback' link, a language selection prompt, and copyright information for Amazon Internet Services Private Ltd.

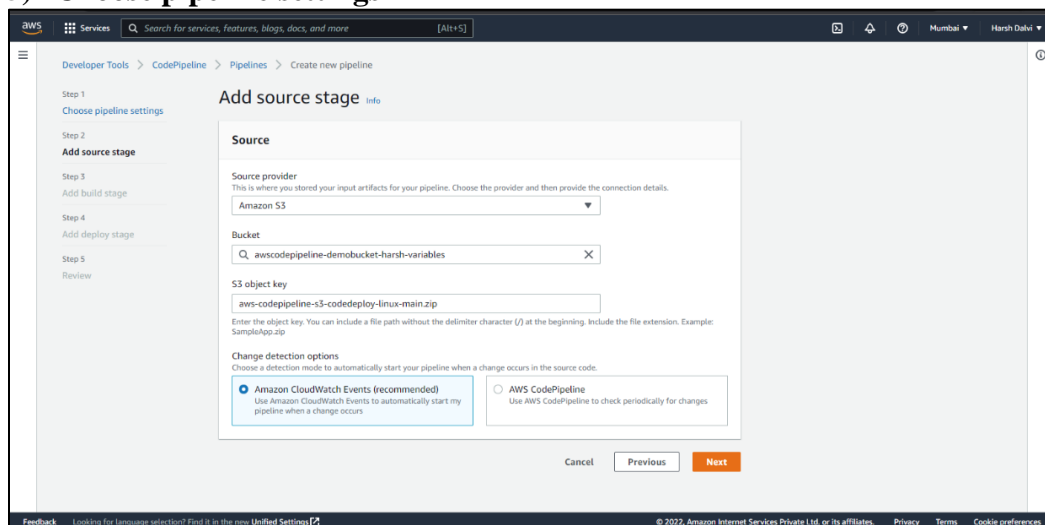
Name: Harsh Dalvi
Roll No: 13

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

4) Upload the file/folder

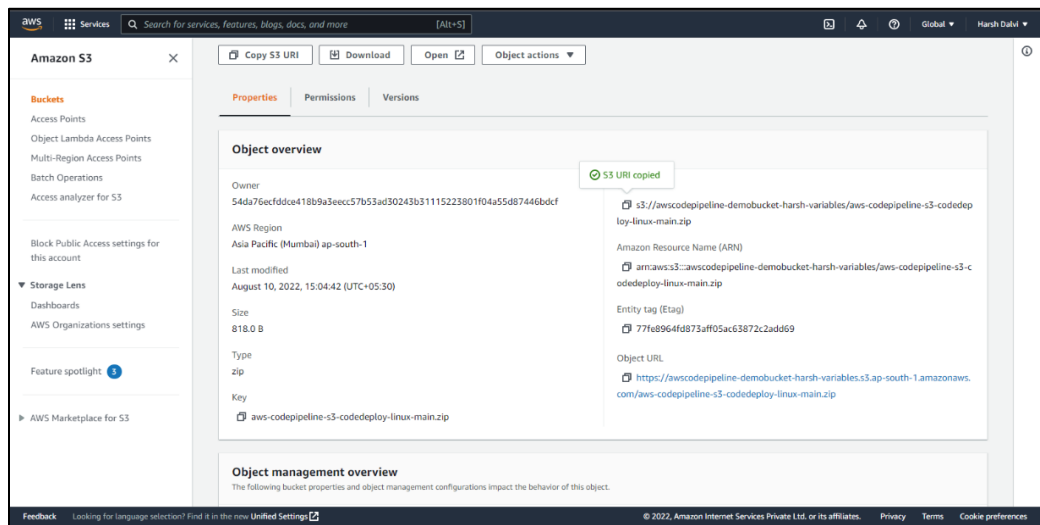


5) Choose pipeline settings

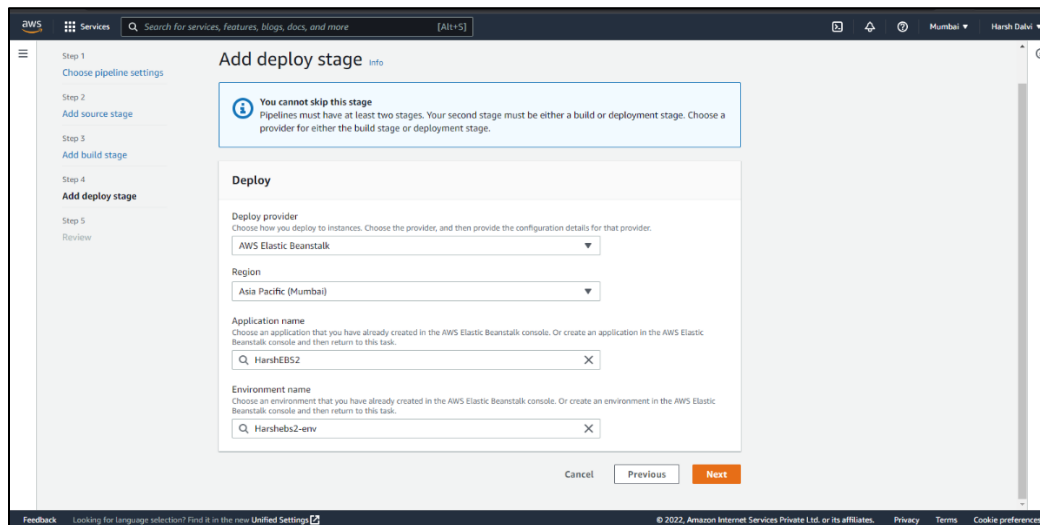
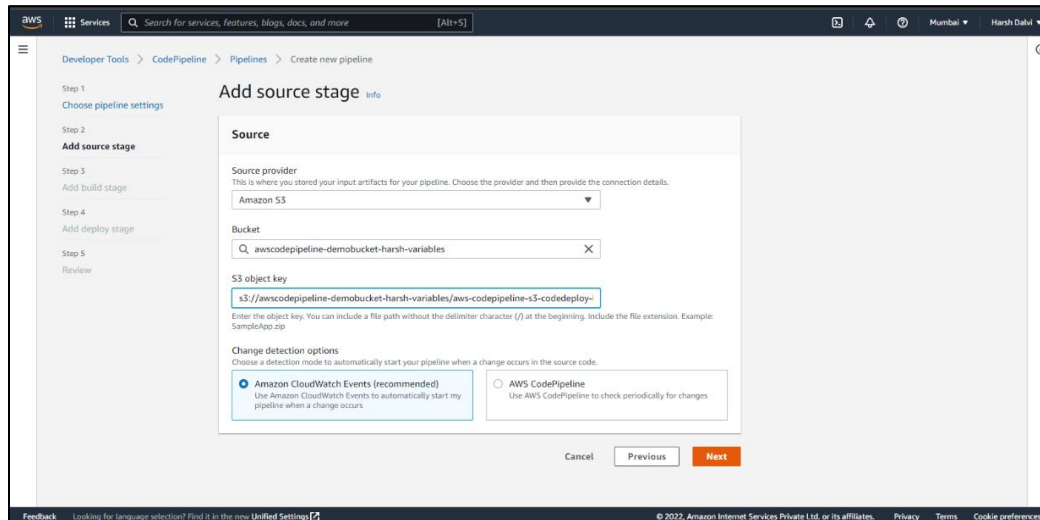


Name: Harsh Dalvi
Roll No: 13

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

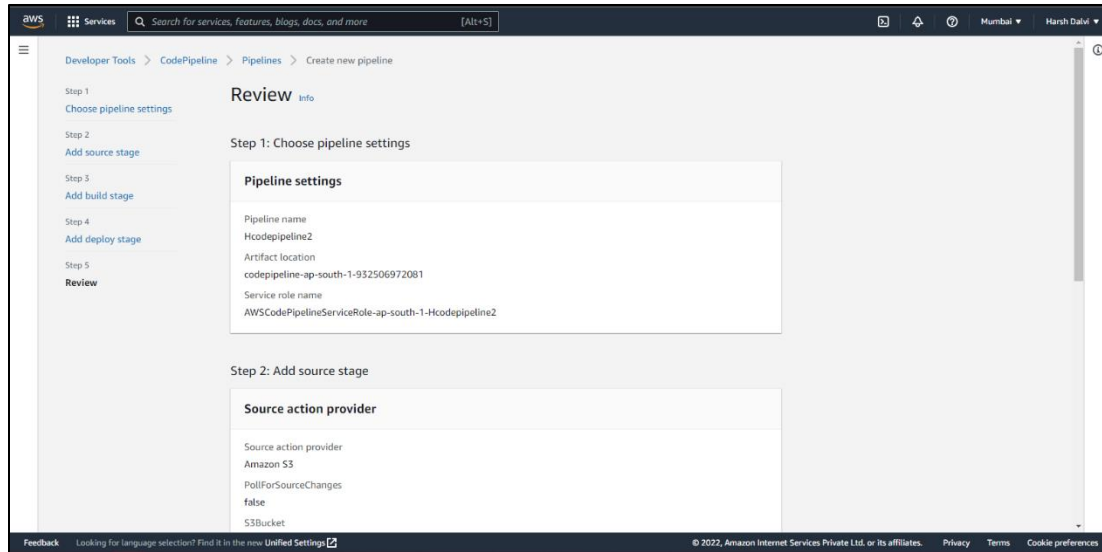


a) Key is the zip_file_name_zip



Name: Harsh Dalvi
Roll No: 13

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



This screenshot shows the 'Review' step of the AWS CodePipeline console. The left sidebar lists the steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main content area is titled 'Review' and shows the configuration for Step 1.

Step 1: Choose pipeline settings

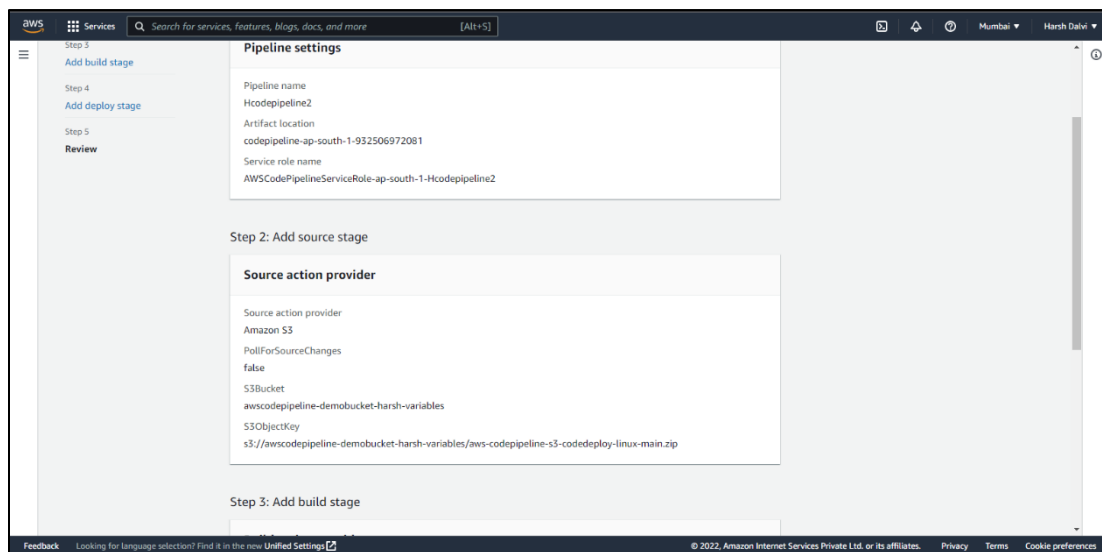
Pipeline settings

- Pipeline name: Hcodepipeline2
- Artifact location: codepipeline-ap-south-1-932506972081
- Service role name: AWSCodePipelineServiceRole-ap-south-1-Hcodepipeline2

Step 2: Add source stage

Source action provider

- Source action provider: Amazon S3
- PollForSourceChanges: false
- S3Bucket: S3Bucket



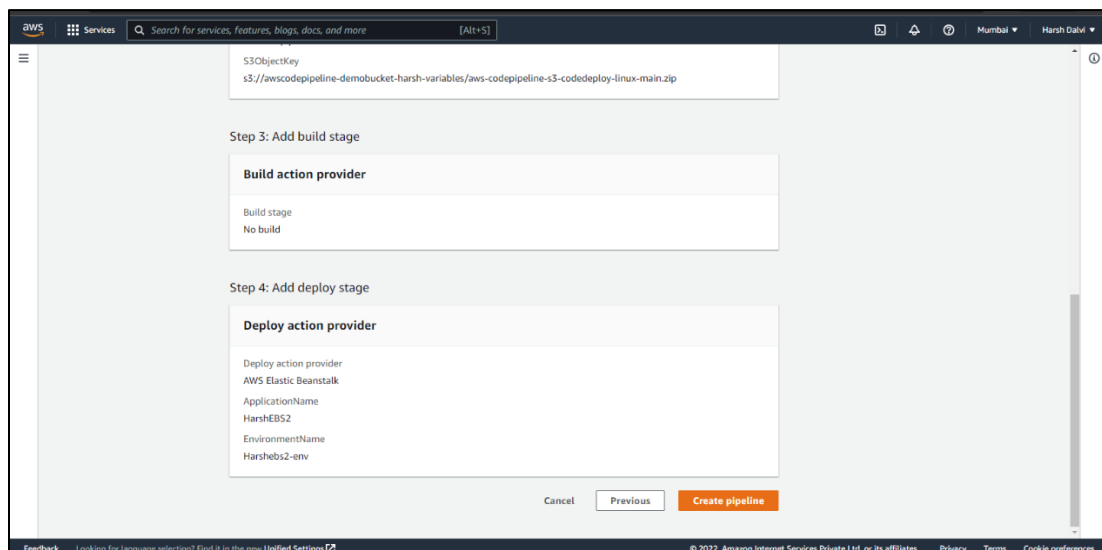
This screenshot shows the 'Add source stage' step of the AWS CodePipeline console. The left sidebar lists the steps: Step 3 (Add build stage), Step 4 (Add deploy stage), and Step 5 (Review). The main content area is titled 'Add source stage' and shows the configuration for Step 2.

Step 2: Add source stage

Source action provider

- Source action provider: Amazon S3
- PollForSourceChanges: false
- S3Bucket: awscodepipeline-demobucket-harsh-variables
- S3ObjectKey: s3://awscodepipeline-demobucket-harsh-variables/aws-codepipeline-s3-codedeploy-linux-main.zip

Step 3: Add build stage



This screenshot shows the 'Add build stage' and 'Add deploy stage' steps of the AWS CodePipeline console. The left sidebar lists the steps: Step 3 (Add build stage) and Step 4 (Add deploy stage). The main content area is titled 'Add build stage' and 'Add deploy stage'.

Step 3: Add build stage

Build action provider

- Build stage: No build

Step 4: Add deploy stage

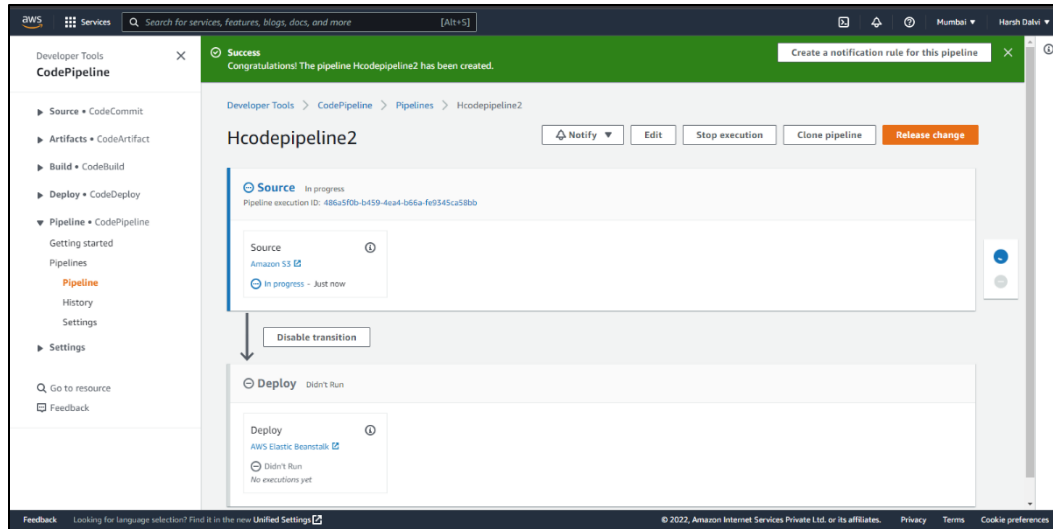
Deploy action provider

- Deploy action provider: AWS Elastic Beanstalk
- ApplicationName: HarshEBS2
- EnvironmentName: HarshEBS2-env

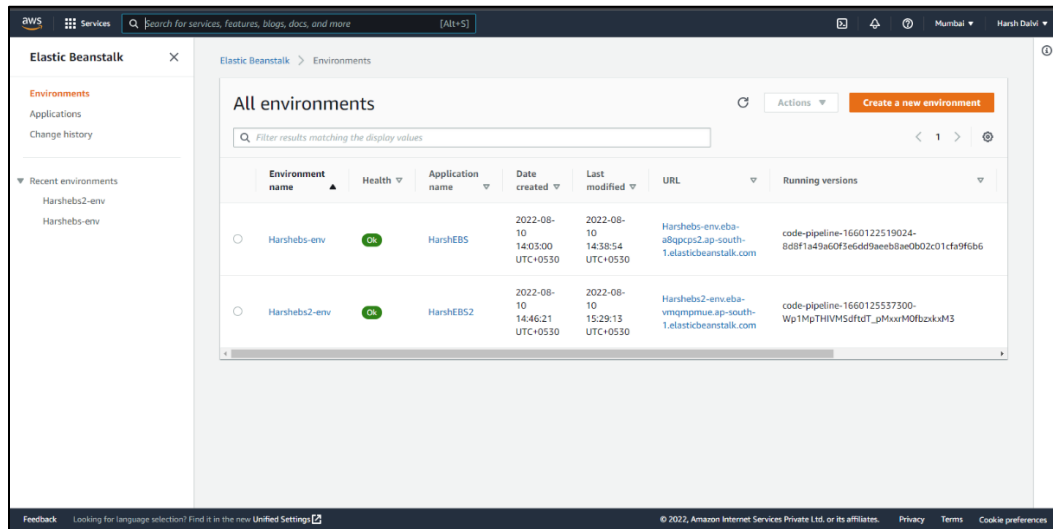
At the bottom, there are buttons for 'Cancel', 'Previous', and 'Create pipeline'.

Name: Harsh Dalvi
Roll No: 13

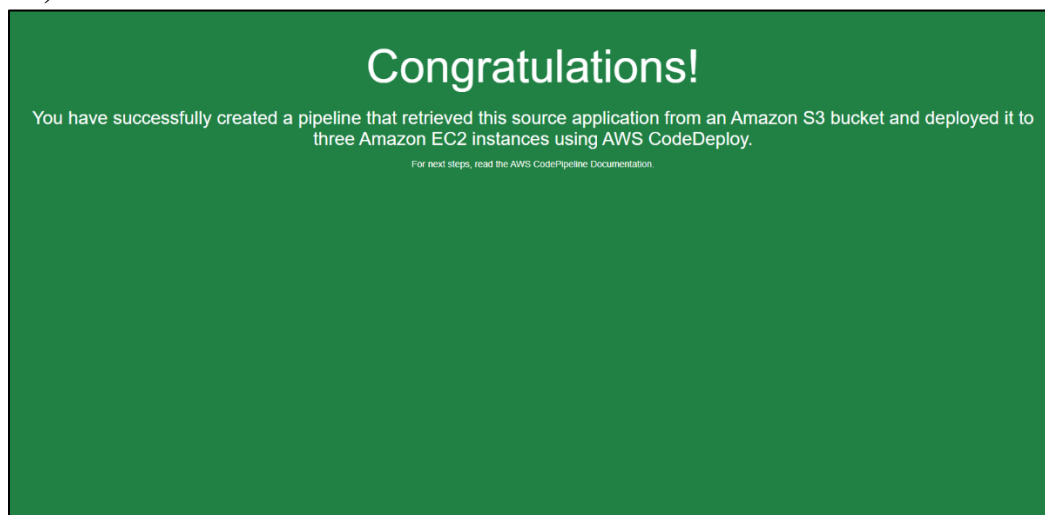
Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B



b) Go to elastic bean stalk



c) Click on environment URL

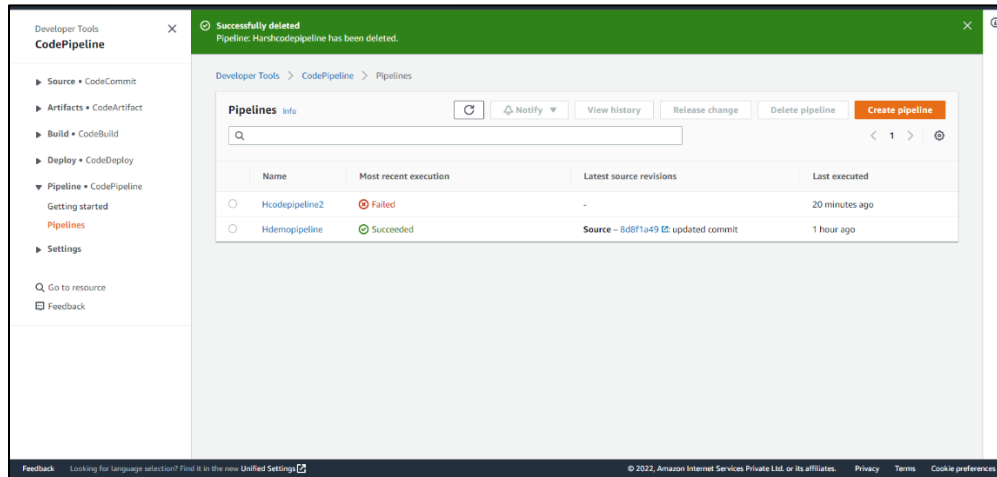


Name: Harsh Dalvi
Roll No: 13

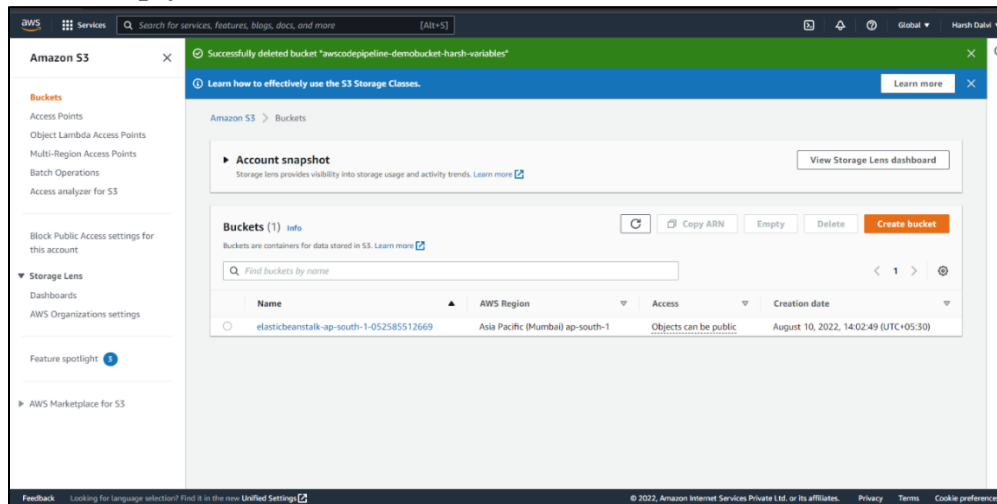
Subject: Advance DevOps , Sem: SEM V
Class / Batch: TE-IT / Batch B

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).