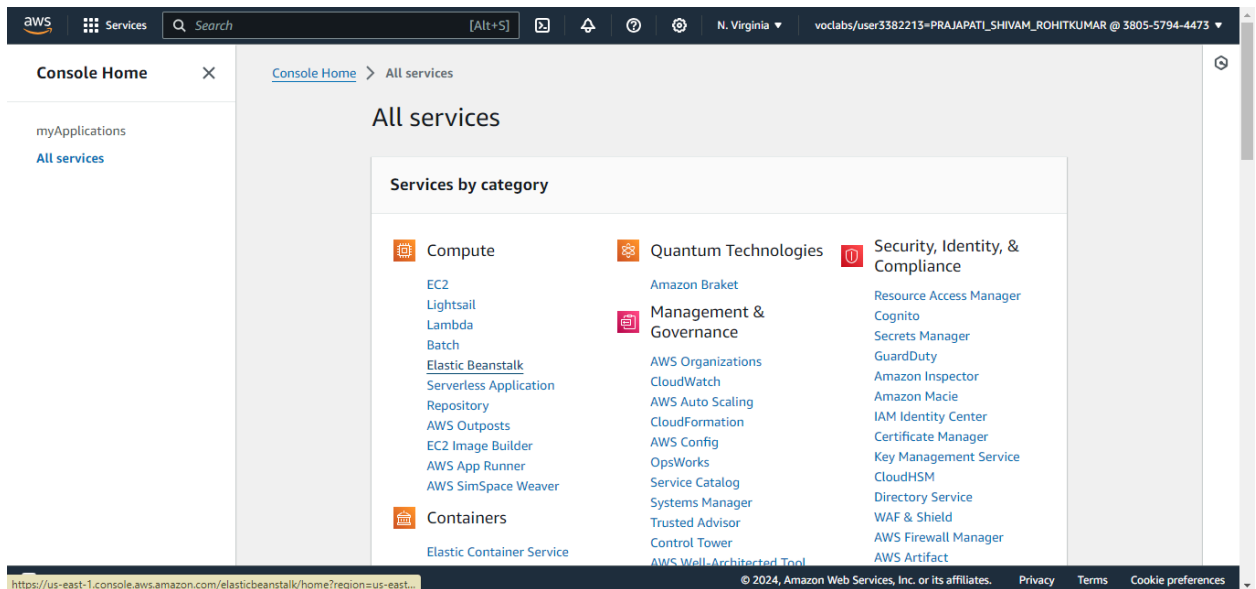


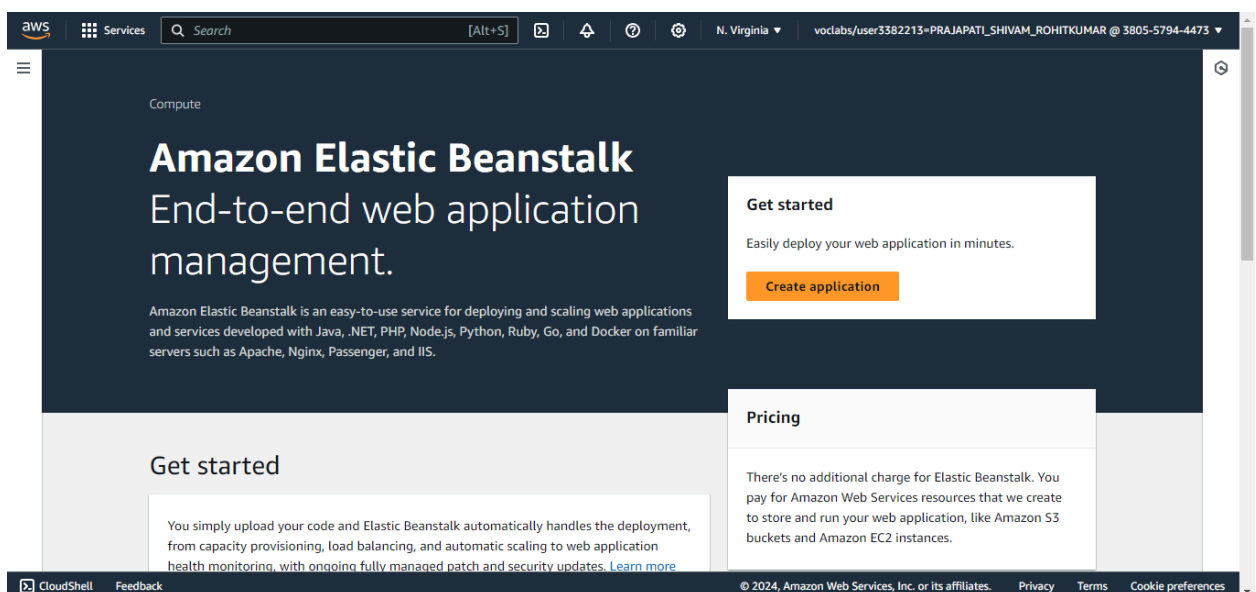
Experiment No :2

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.

Step 1: Login to your AWS console. Search for Elastic Beanstalk in the searchbar near services.



Step 2: Go to Elastic Beanstalk and click on Create Application



Step 3: Enter the name of your application. Scroll down and in the platform, select platform as PHP. Keep the application code as Sample Application. Set the instance to single instance. Click on NEXT.

Configure environment Info

Environment tier Info
Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

☒ **Web server environment**
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)

☐ **Worker environment**
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. [Learn more](#)

Application information Info

Application name

Maximum length of 100 characters.

► Application tags (optional)

Environment information Info
Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

Domain
 .us-east-1.elasticbeanstalk.com

Environment description

Platform Info

Platform type
☒ **Managed platform**

Environment name will be automatically displayed based on the Application name provided by the user

Platform [Info](#)

Platform type

- ☒ **Managed platform**
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)
- ☐ **Custom platform**
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

PHP

Platform branch

PHP 8.3 running on 64bit Amazon Linux 2023

Platform version

4.3.1 (Recommended)

Application code [Info](#)

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

On Selecting Platform type as PHP then Platform branch and Platform version will be displayed Automatically

Application code [Info](#)

- ☒ **Sample application**
Application versions that you have uploaded.
- ☐ **Existing version**
Application versions that you have uploaded.
- ☐ **Upload your code**
Upload a source bundle from your computer or copy one from Amazon S3.

Presets [Info](#)

Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

Configuration presets

- ☒ **Single instance (free tier eligible)**
- ☐ Single instance (using spot instance)
- ☐ High availability
- ☐ High availability (using spot and on-demand instances)
- ☐ Custom configuration

Cancel **Next**

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

After doing the necessary steps Click on Next button

Step 4: Use an existing service role and choose whatever service role is present on your account
Because existing service role is different for different user and it is automatically generated ,select the key-pair which is default given as vockey which is going to be sued to securely log in to your EC2 instance

The screenshot shows the AWS Management Console interface for the 'Configure service access' step. The left sidebar contains a navigation menu with steps 1 through 6. Step 2, 'Configure service access', is the active step. The main content area is titled 'Configure service access' and includes a 'Service access' section with a description of IAM roles and instance profiles. Below this, there are three sections: 'Service role' with radio buttons for 'Create and use new service role' and 'Use an existing service role' (selected); 'Existing service roles' with a dropdown menu showing 'LabRole'; 'EC2 key pair' with a dropdown menu showing 'vockey'; and 'EC2 instance profile' with a dropdown menu showing 'LabInstanceProfile'. At the bottom of the console, there is a footer with 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates.

Step 5: Click on Skip to Review Button

This screenshot shows the same 'Configure service access' step as the previous one, but with the 'Skip to review' button highlighted in orange at the bottom right of the main content area. The 'Service role' section remains selected. The 'Existing service roles' dropdown shows 'LabRole', the 'EC2 key pair' dropdown shows 'vockey', and the 'EC2 instance profile' dropdown shows 'LabInstanceProfile'. A 'View permission details' button is now visible below the instance profile dropdown. The footer at the bottom of the console remains the same, showing 'CloudShell', 'Feedback', and copyright information.

Step 6: Review the settings that you have set up for your application and submit your application thereafter by clicking on submit button

Review [Info](#)

Step 1: Configure environment [Edit](#)

Environment information

Environment tier	Application name
Web server environment	FirstWebApp
Environment name	Application code
FirstWebApp-env	Sample application
Platform	
arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1	

Step 2: Configure service access [Edit](#)

Service access [Info](#)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Display errors

Off

Memory limit

256M

Logs retention

7

X-Ray enabled

Deactivated

Document root

-

Zlib output compression

Off

Rotate logs

Deactivated

Max execution time

60

Proxy server

nginx

Update level

minor

Environment properties

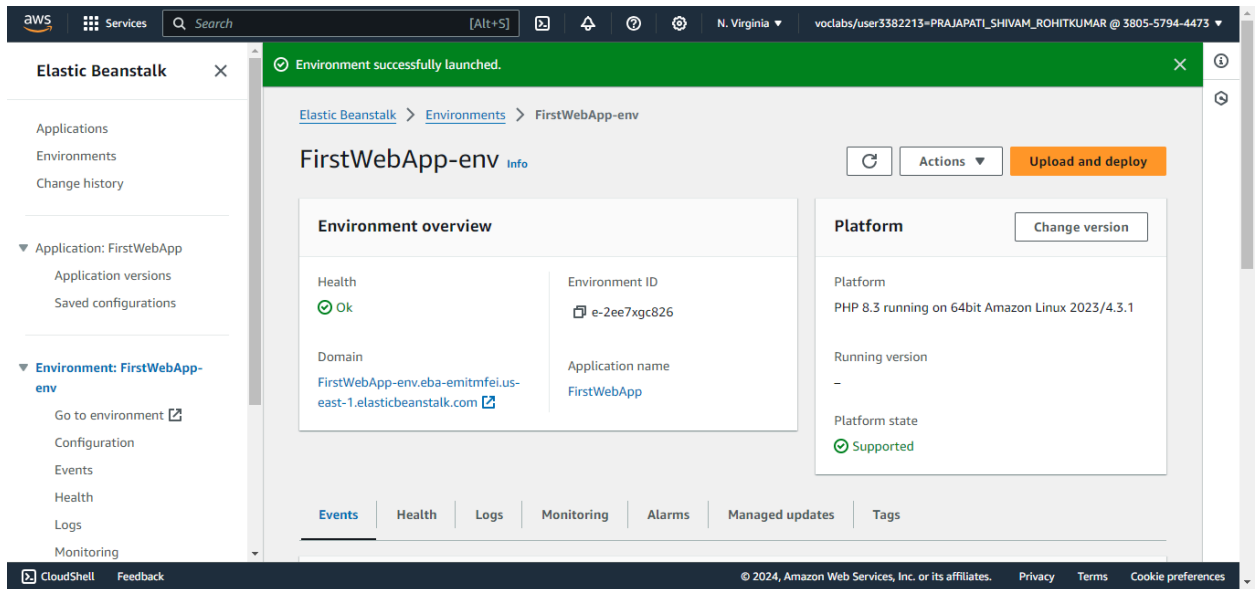
Key	Value
No environment properties	
There are no environment properties defined	

Cancel Previous **Submit**

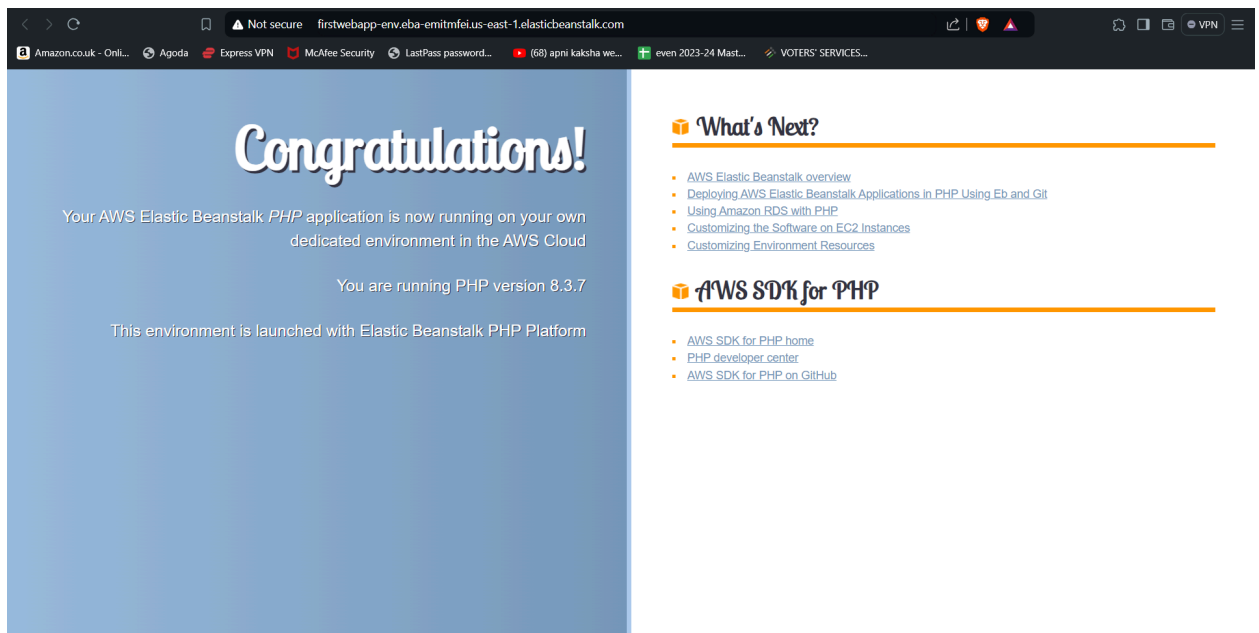
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on the Submit

Step 7: We have Successfully launched our environment

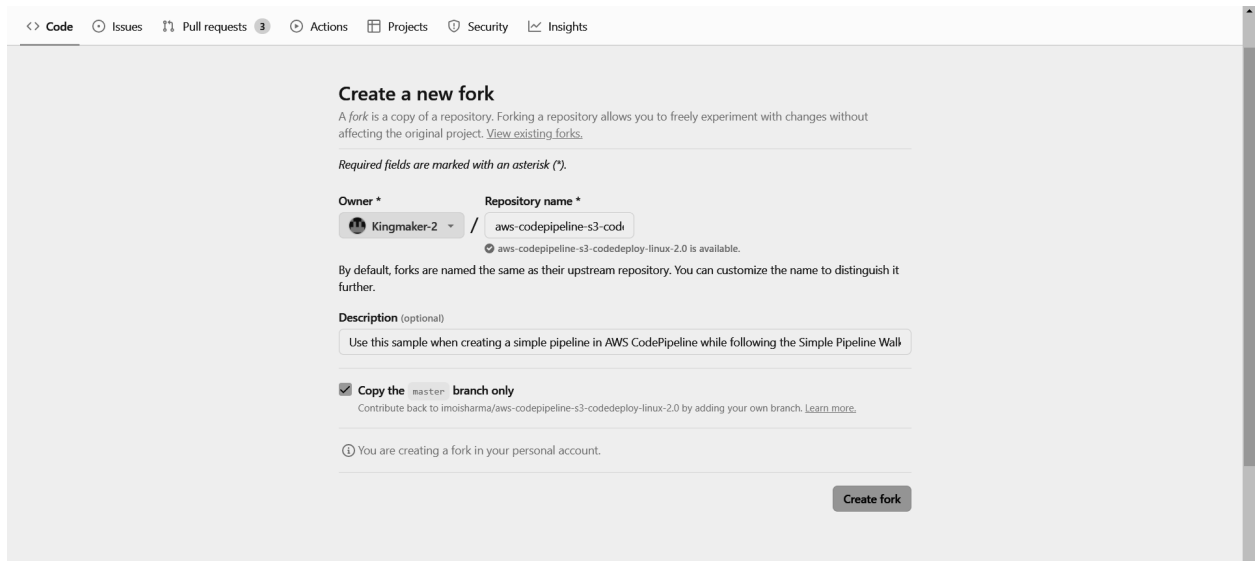
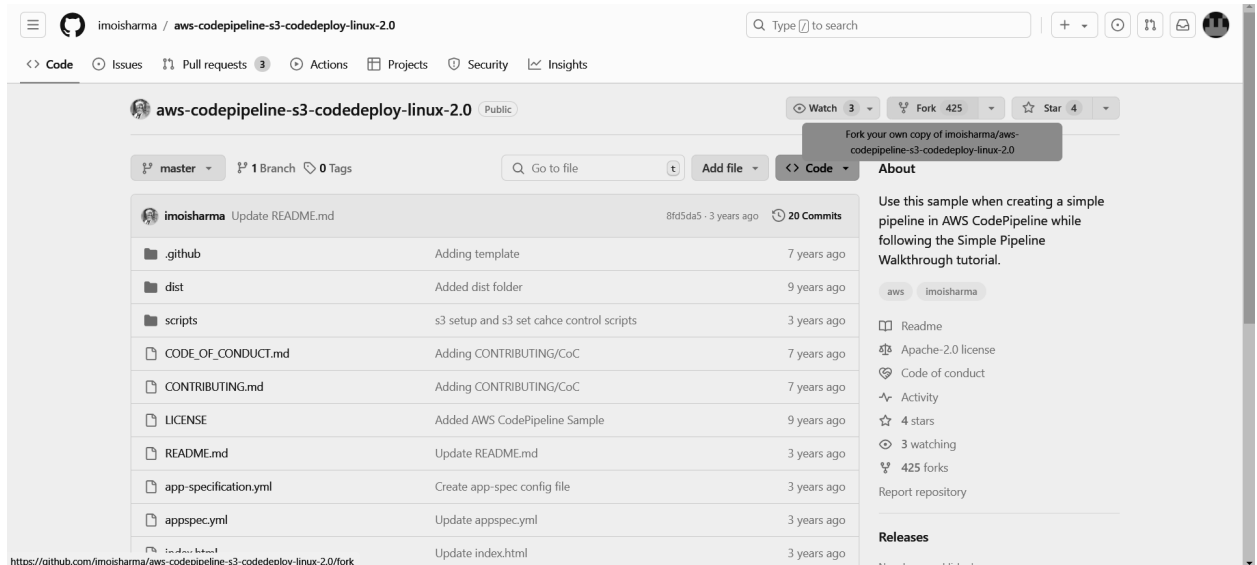


Step 8: Click on the link under domain section and it will redirect you to a new page .



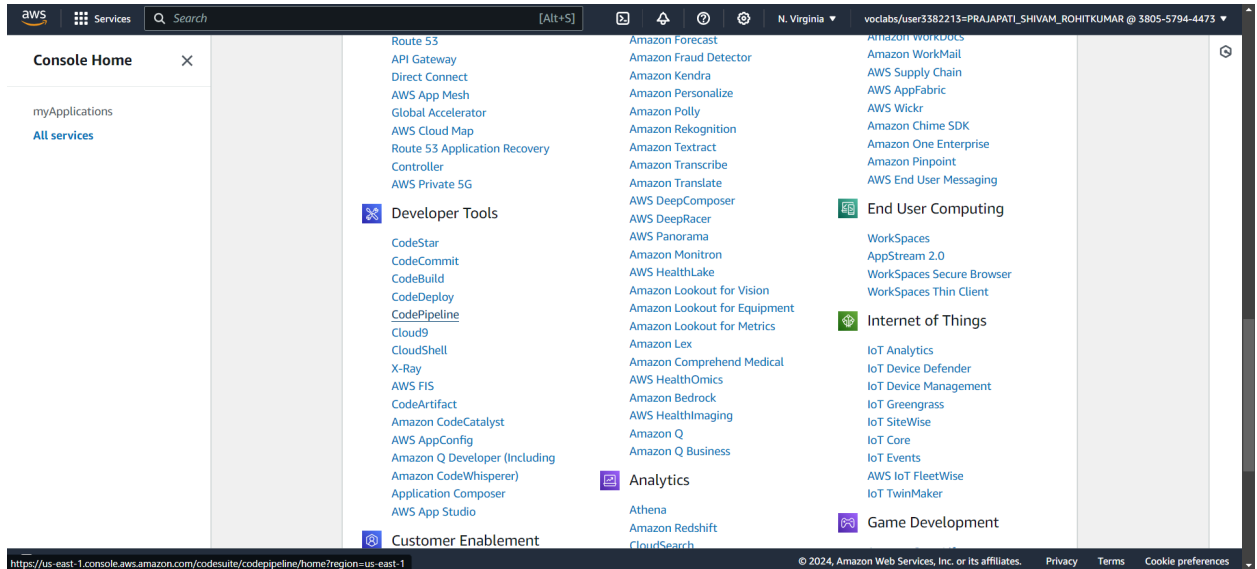
Step 9 : Go to the github link below. This is a github with a sample code for deploying a file on AWS CodePipeline. Fork this repository into your personal github.

<https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux>

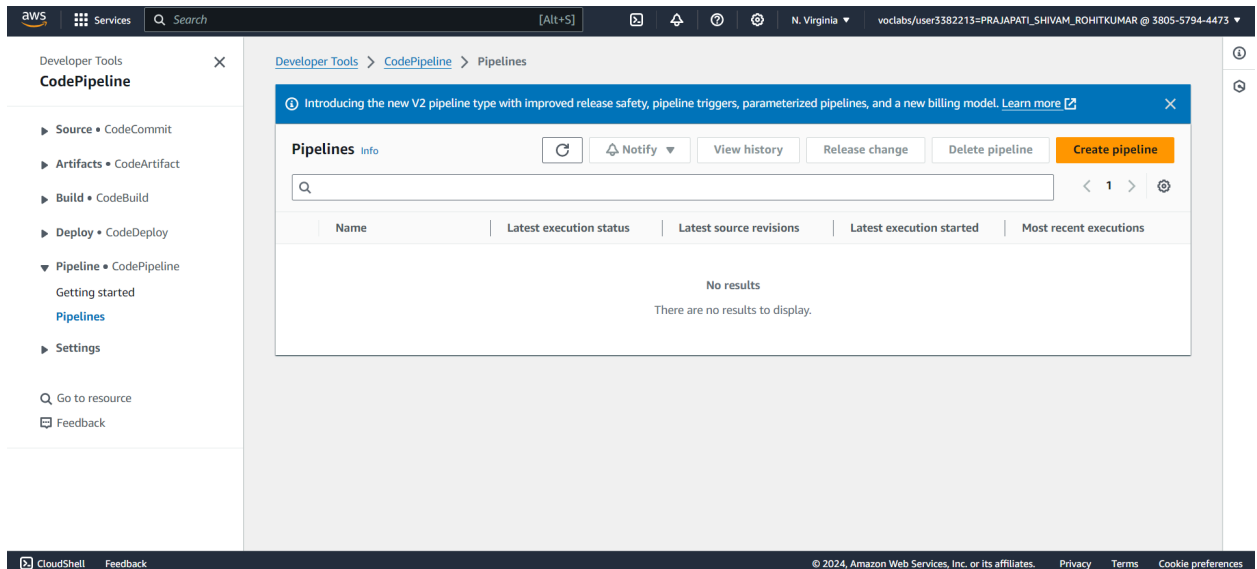


Click on Create Fork this will Duplicate the entire repository to your Github Account

Step 10: Search CodePipeline in the services tab and click on it.



Step 11: Click on Create Pipeline.



Step 12: Give a name to your Pipeline. A new service role would be created with the name of the pipeline. Make Sure the service role should be New Service Role

The screenshot shows the AWS CodePipeline console interface. The 'Execution mode' section has three radio buttons: 'Superseded', 'Queued (Pipeline type V2 required)' (which is selected), and 'Parallel (Pipeline type V2 required)'. The 'Service role' section has two radio buttons: 'New service role' (selected) and 'Existing service role'. Below these, the 'Role name' field is populated with 'AWSCodePipelineServiceRole-us-east-1-MyPipeline'. There is a checkbox 'Allow AWS CodePipeline to create a service role so it can be used with this new pipeline' which is checked. The 'Variables' section is empty, showing 'No variables defined at the pipeline level in this pipeline.'

The screenshot shows the AWS CodePipeline console interface, continuing from the previous step. The 'Role name' field is still 'AWSCodePipelineServiceRole-us-east-1-MyPipeline'. The 'Variables' section is empty, showing 'No variables defined at the pipeline level in this pipeline.' Below this is an 'Add variable' button. At the bottom, there is a section for 'Advanced settings' and two buttons: 'Cancel' and 'Next'.

*Under variables section it is not necessary to add variables
Click on Next Button*

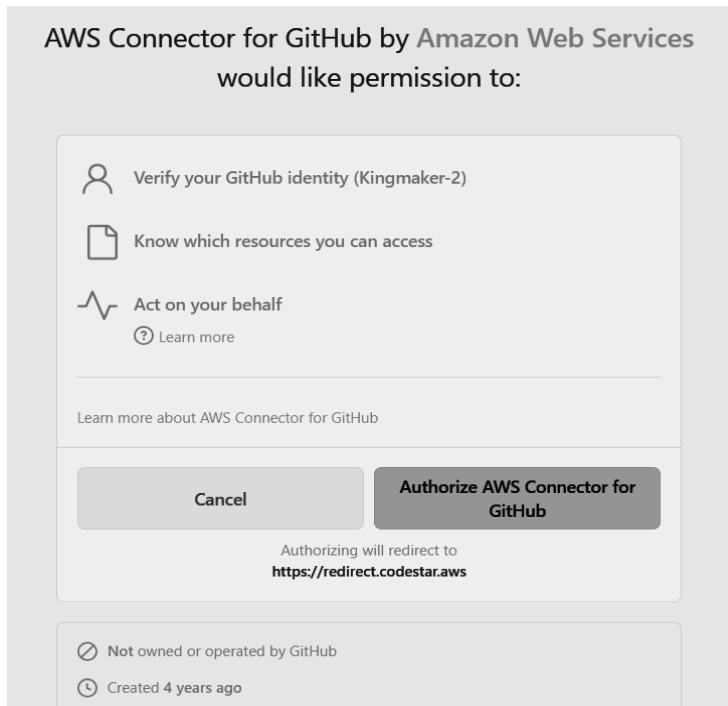
Step 13: Select a source provider (as GitHub Version (2)). Click on connect to Github (*This part have to be done in the personal account of aws as in academy account it won't allow you to create pipeline with github version 1 or 2*)

The screenshot shows the AWS CodePipeline console interface. The breadcrumb navigation is 'Developer Tools > CodePipeline > Pipelines > Create new pipeline'. The left sidebar shows a sequence of steps: Step 1: Choose pipeline settings, Step 2: Add source stage (highlighted), Step 3: Add build stage, Step 4: Add deploy stage, and Step 5: Review. The main content area is titled 'Add source stage' and 'Step 2 of 5'. It contains a 'Source' section with a 'Source provider' dropdown set to 'GitHub (Version 2)'. Below this is a blue information box titled 'New GitHub version 2 (app-based) action' with text explaining that it uses GitHub Apps to access repositories. Underneath is a 'Connection' section with a search input and a 'Connect to GitHub' button. The 'Repository name' section has a search input and a note about the format 'group/subgroup/project'. The 'Default branch' section also has a search input.

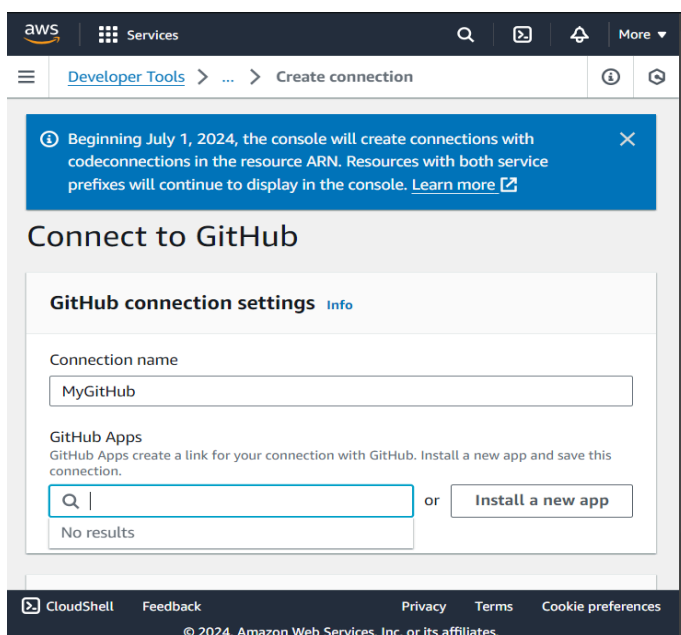
Step 14: Give a name to your GitHub app Connection and click on Connect. This will give you a prompt to either to select a GitHub app or to install a new app. If it is your first time, click on Install a new app.

The screenshot shows the 'Create a connection' page in the AWS CodePipeline console. The breadcrumb navigation is 'Developer Tools > ... > Create connection'. The main heading is 'Create a connection'. Below it is a section titled 'Create GitHub App connection'. There is a 'Connection name' input field with the text 'MyGitHub' entered. Below this is a section titled 'Tags - optional' with a right-pointing triangle icon. At the bottom right of the main content area is an orange 'Connect to GitHub' button. The footer of the console shows links for 'CloudShell', 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences', along with the copyright notice '© 2024, Amazon Web Services, Inc. or its affiliates.'

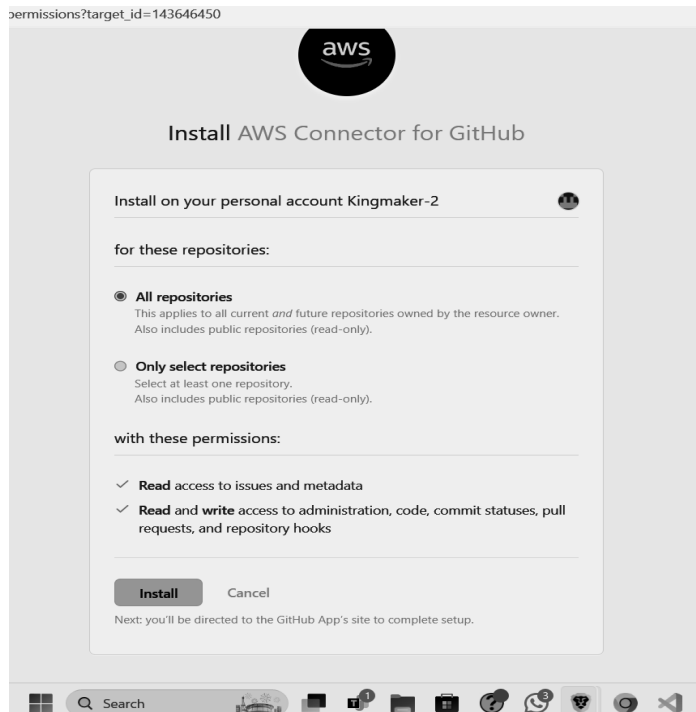
Step 15: After giving the connection name then AWS will Authorize the user



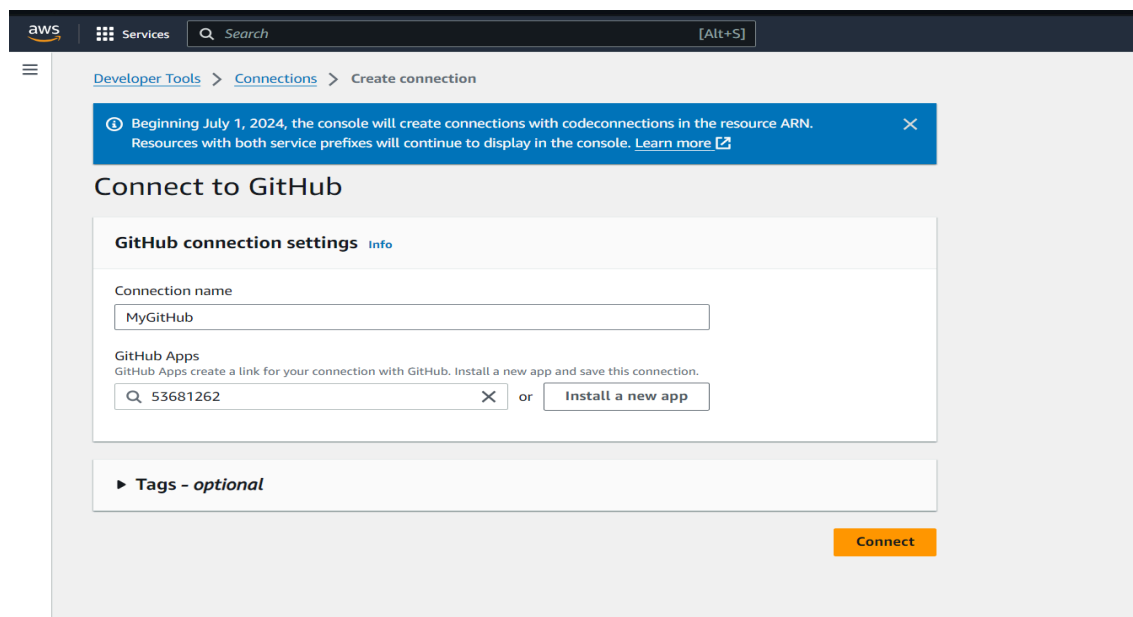
Step 16: Click on New App as we are doing for the first time which was already mentioned in step 14

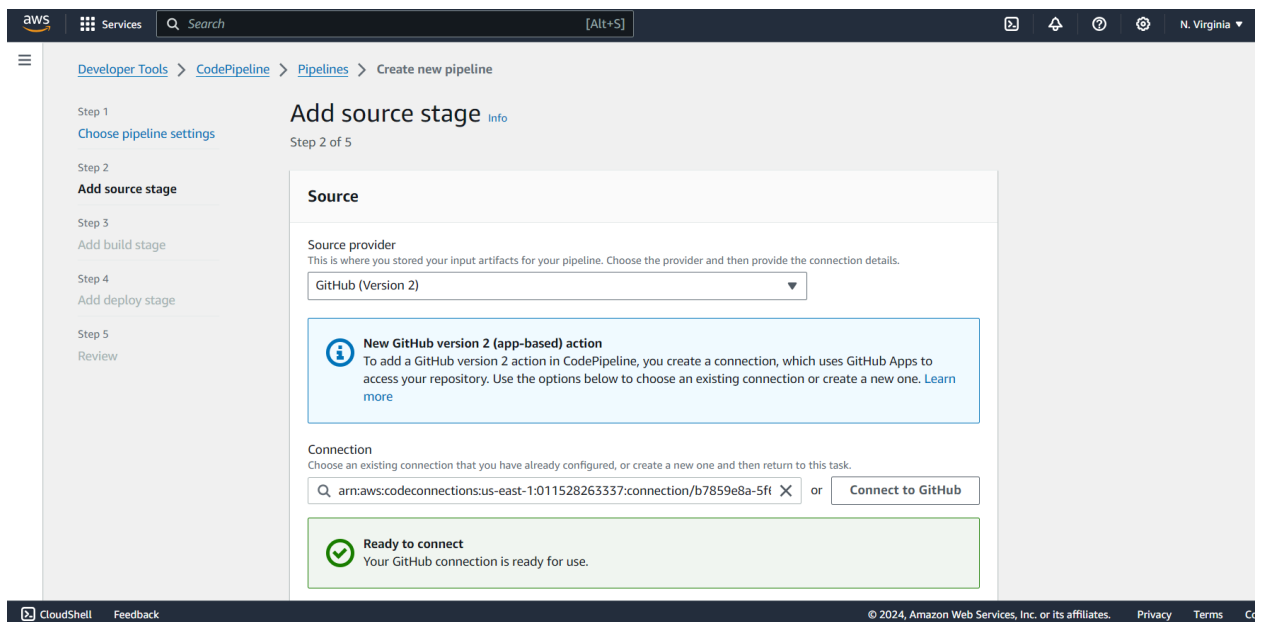


Step 17: This will direct you to install AWS connector on your GitHub .Install it to your account and give it its permissions

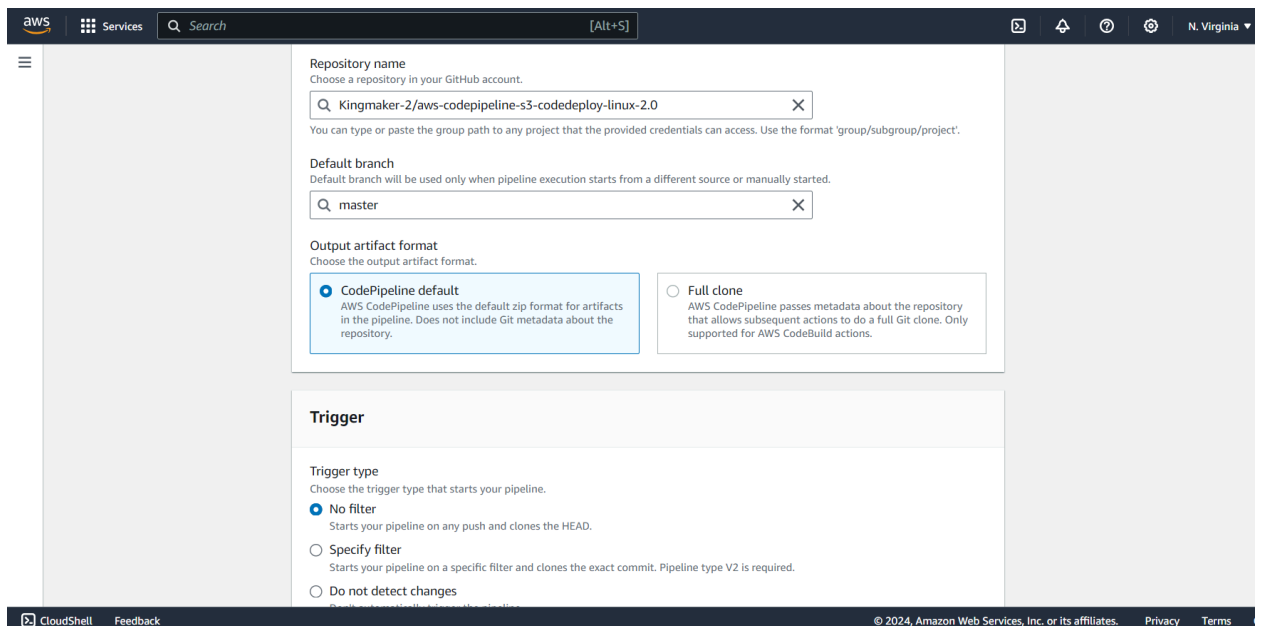


Step 18: After the app is set up, it gives the number in the text field. Click on Connect. After clicking on connect, the link is shown in the connection field and AWS shows that GitHub connection is ready to use.





Step 19: Select the repository that you had forked to your GitHub. After that select the branch on which the files are present (default is Master).



Step 20: Set the Trigger type as no filter. This would allow it to the website to update as soon as some change is made in the github. Click on Next

The screenshot shows the AWS CodePipeline console interface. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and a user profile dropdown for 'N. Virginia'. Below this, the main content area is titled 'Output artifact format' and 'Trigger'. Under 'Output artifact format', there are two radio button options: 'CodePipeline default' (selected) and 'Full clone'. The 'Trigger' section has a 'Trigger type' dropdown set to 'No filter'. Below this, there are three radio button options: 'No filter' (selected), 'Specify filter', and 'Do not detect changes'. A blue information box at the bottom states: 'You can add additional sources and triggers by editing the pipeline after it is created.' At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'.

Step 21: Skip the build stage and go to Deploy. Select the deploy provider as AWS Elastic Beanstalk and Input Artifact as SourceArtifact. The application name would be the name of your Elastic Beanstalk. Then click on next.

The screenshot shows the AWS CodePipeline console interface. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and a user profile dropdown for 'N. Virginia'. Below this, the main content area is titled 'Add build stage'. On the left, there's a sidebar with a list of steps: 'Step 1: Choose pipeline settings', 'Step 2: Add source stage', 'Step 3: Add build stage' (selected), 'Step 4: Add deploy stage', and 'Step 5: Review'. The main content area has a title 'Add build stage' and a subtitle 'Step 3 of 5'. Below this, there's a section titled 'Build - optional'. Under this, there's a 'Build provider' dropdown menu. At the bottom right, there are four buttons: 'Cancel', 'Previous', 'Skip build stage', and 'Next'.

The screenshot shows the AWS CodePipeline console interface. The top navigation bar includes the AWS logo, a search bar, and the user's account information. The left sidebar shows the navigation menu with options like 'Developer Tools', 'CodePipeline', and 'Pipelines'. The main content area is titled 'Add deploy stage' and is part of a 5-step process. A warning message states: 'You cannot skip this stage. Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.' The 'Deploy' section includes a 'Deploy provider' dropdown set to 'AWS Elastic Beanstalk', a 'Region' dropdown set to 'US East (N. Virginia)', and an 'Input artifacts' dropdown set to 'SourceArtifact'. The 'Application name' field is partially visible at the bottom.

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Add deploy stage [info](#)

Step 4 of 5

You cannot skip this stage
Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.

Deploy

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS Elastic Beanstalk

Region
US East (N. Virginia)

Input artifacts
Choose an input artifact for this action. [Learn more](#)

SourceArtifact

No more than 100 characters

Application name

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the AWS CodePipeline console interface, specifically the 'Review' step of the 'Add deploy stage' process. The 'Deploy provider' is set to 'AWS Elastic Beanstalk', the 'Region' is 'US East (N. Virginia)', and the 'Input artifacts' are 'SourceArtifact'. The 'Application name' field is set to 'FirstWebApp'. The 'Environment name' field is set to 'FirstWebApp-env'. The 'Configure automatic rollback on stage failure' checkbox is checked. The bottom navigation bar includes 'Cancel', 'Previous', and 'Next' buttons.

REVIEW

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

AWS Elastic Beanstalk

Region
US East (N. Virginia)

Input artifacts
Choose an input artifact for this action. [Learn more](#)

SourceArtifact

No more than 100 characters

Application name
Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.

FirstWebApp

Environment name
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.

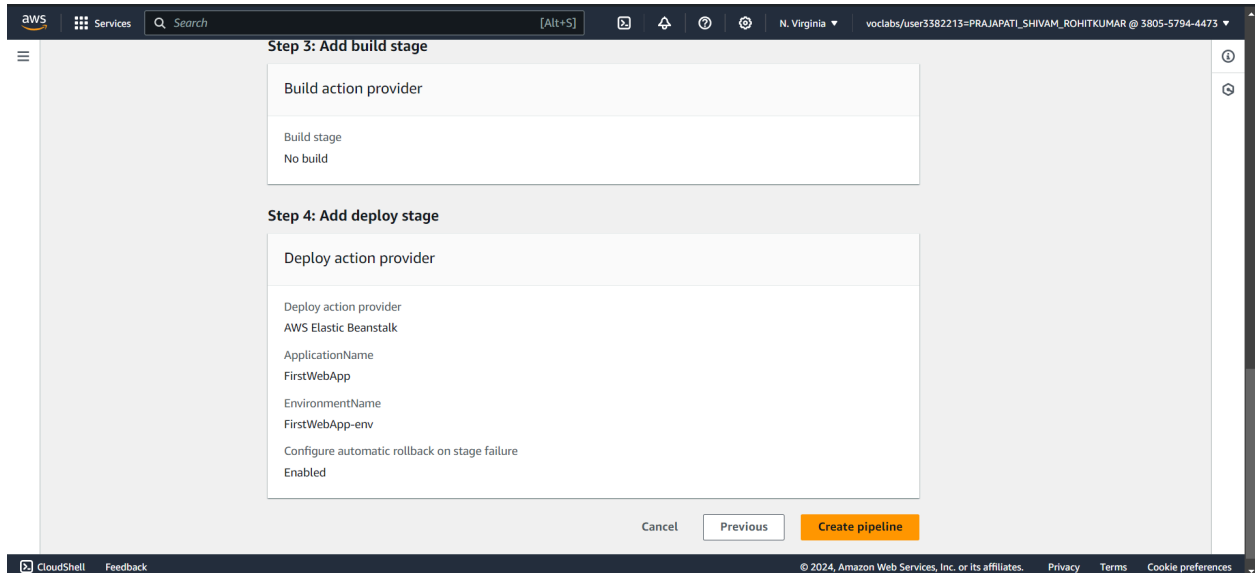
FirstWebApp-env

☒ Configure automatic rollback on stage failure

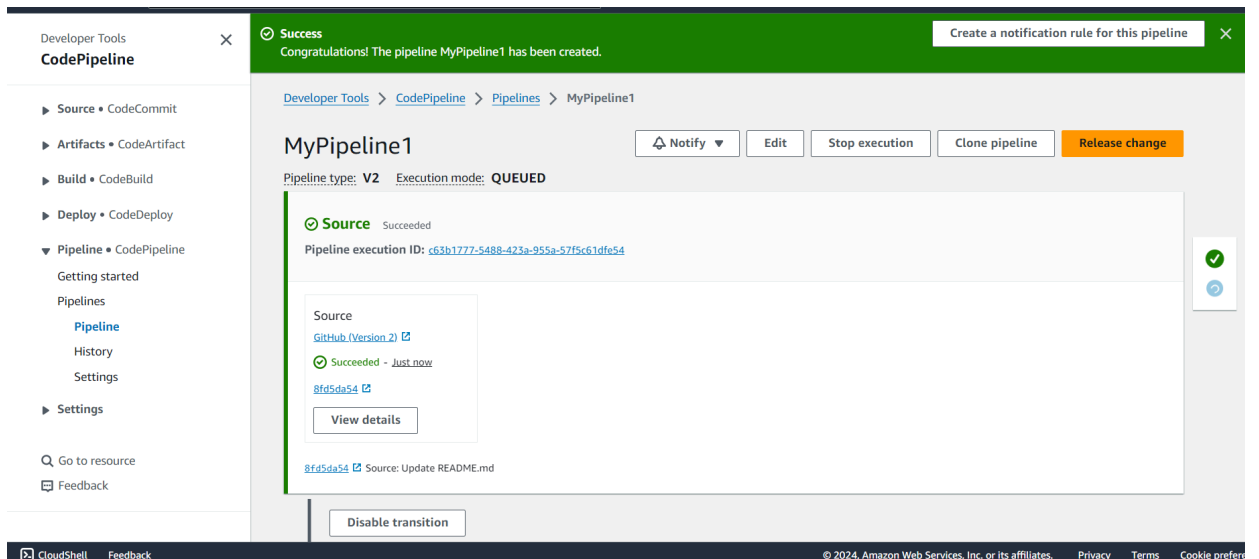
Cancel Previous Next

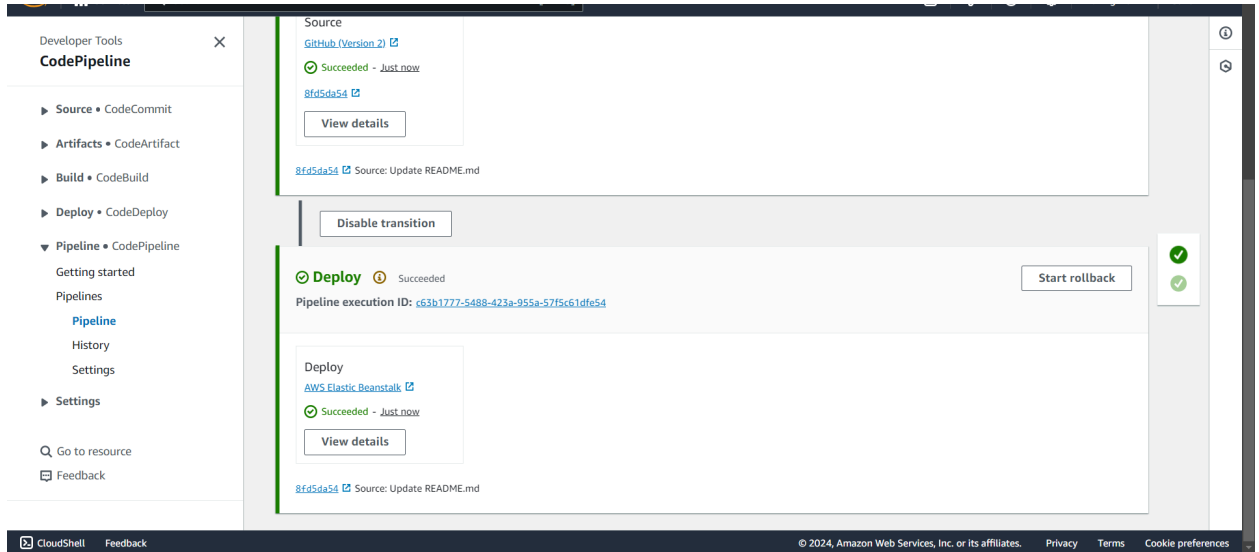
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Step 22: Check all the information and click on create Pipeline



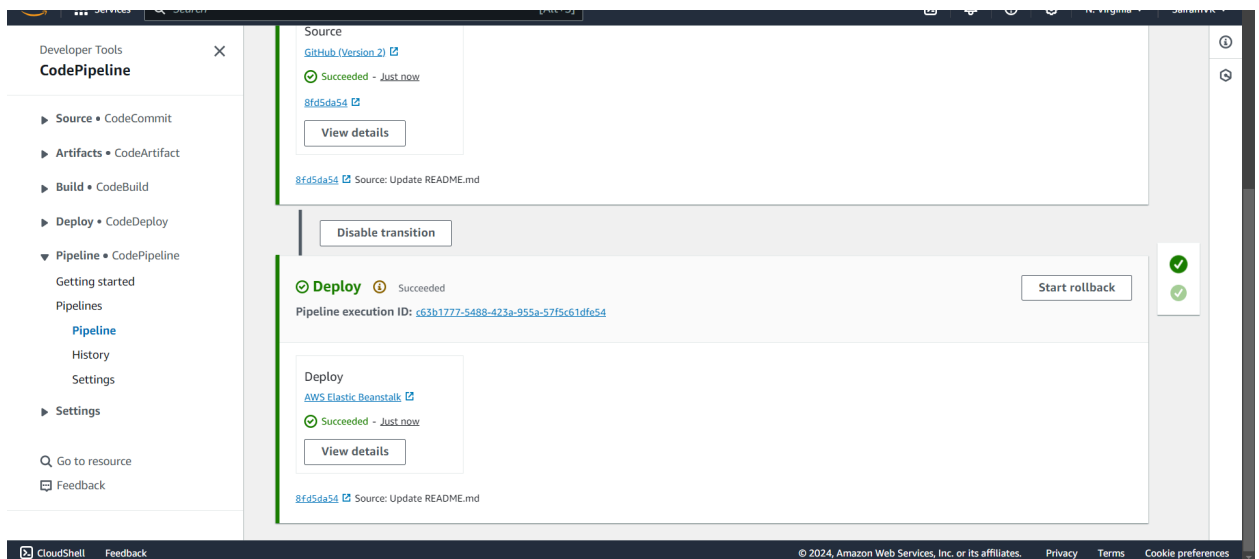
Step 23: If the pipeline is successfully deployed, this screen comes up where the source is set up and then it is transitioned to deploy



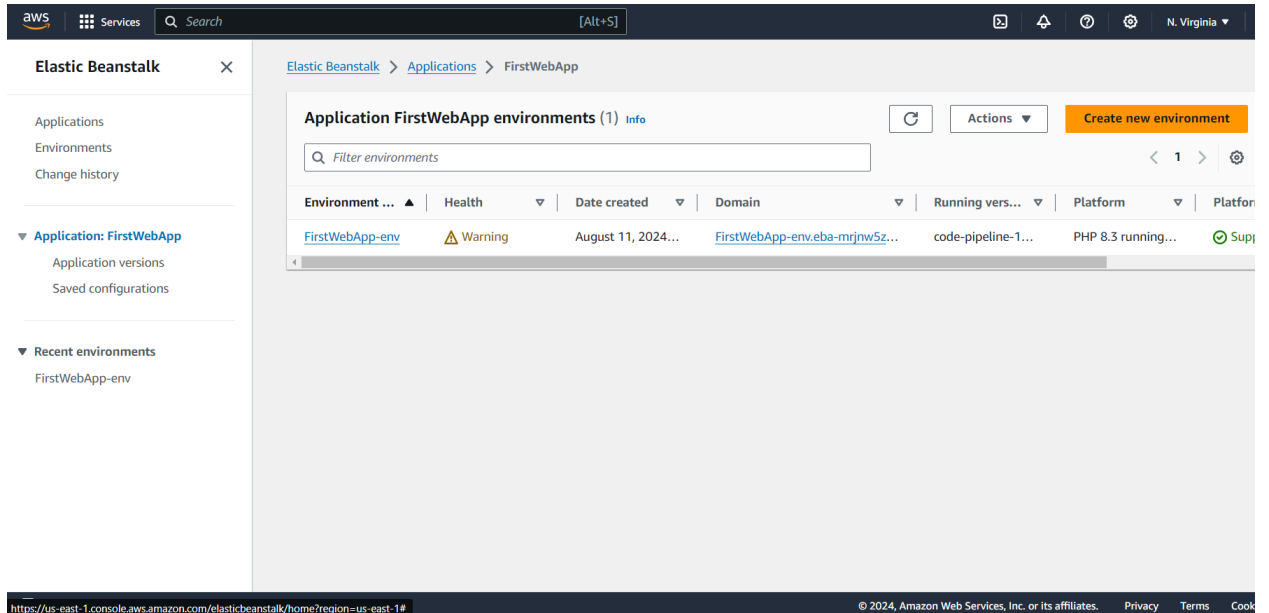


Deploy is done successfully as we can blue tick mark at the right corner

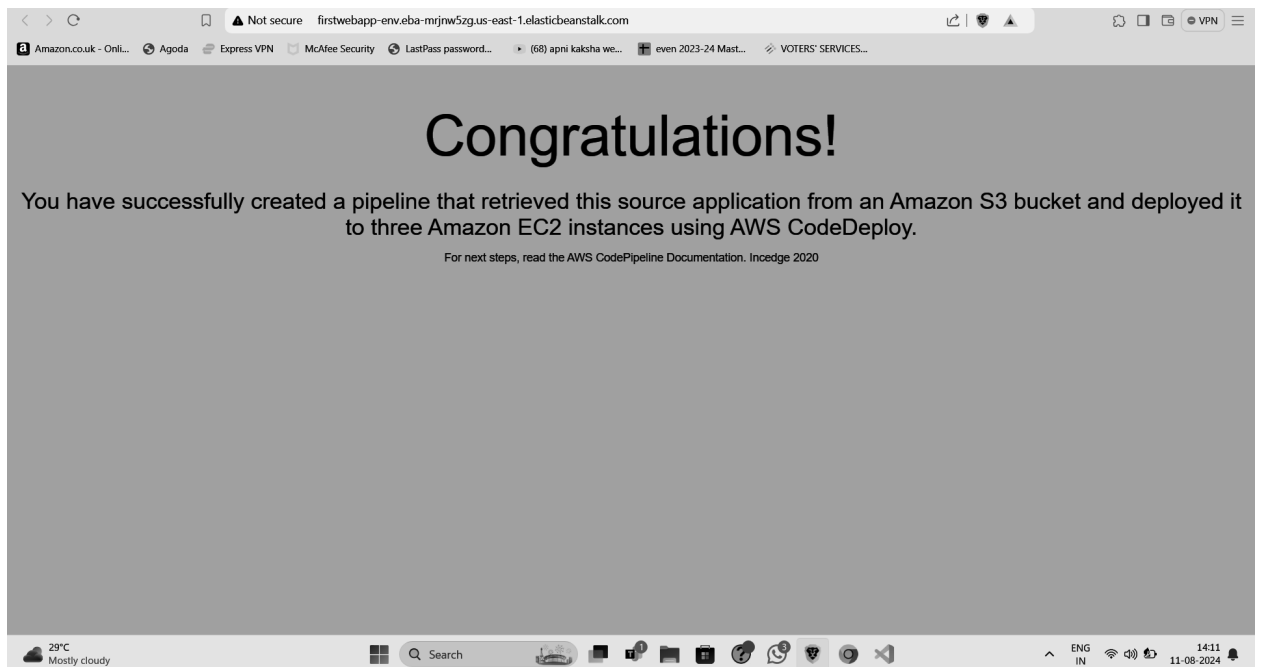
Step 24: Once the deployment is complete, click on the AWS Elastic Beanstalk under Deploy.



Step 25: This will redirect you to the application screen of Elastic Beanstalk. Click on the link shown under Domain



Step 26: This will successfully show the sample website hosted.



Step 27: Now, we make some changes to the index.html file in the github. For eg: If you make some changes to the tag .Once the changes are committed ,when the website is refreshed ,the changes will be seen.

