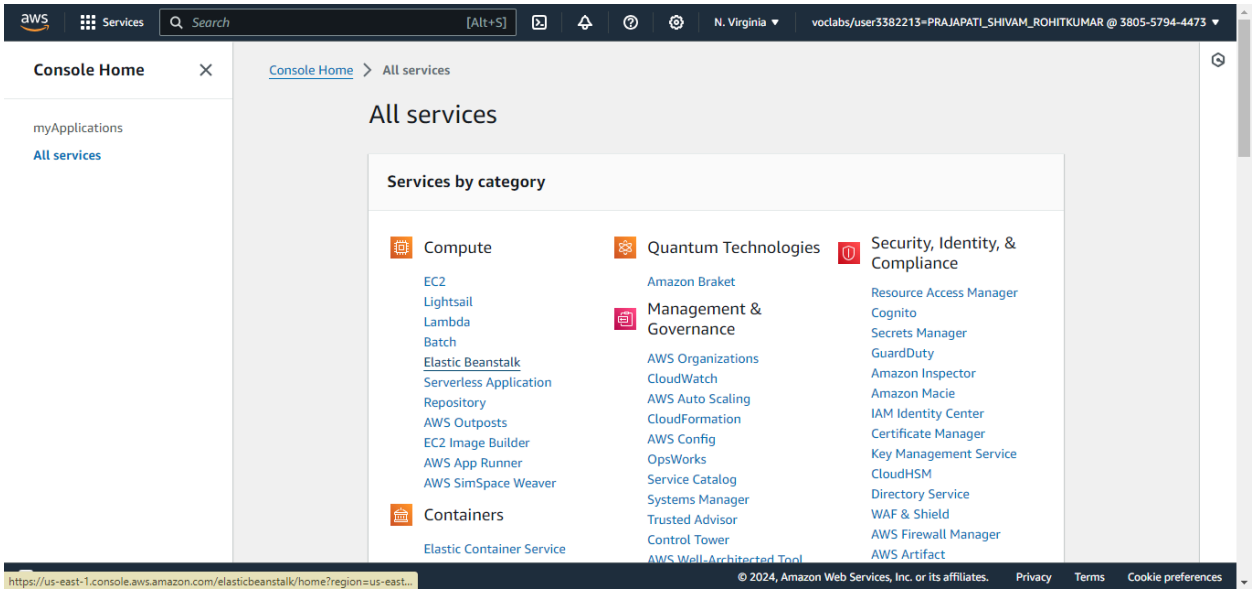
Name:Shivam.R.Prajapati **Div:D15C** **Roll No:41** **Academic Year: 2024-25**

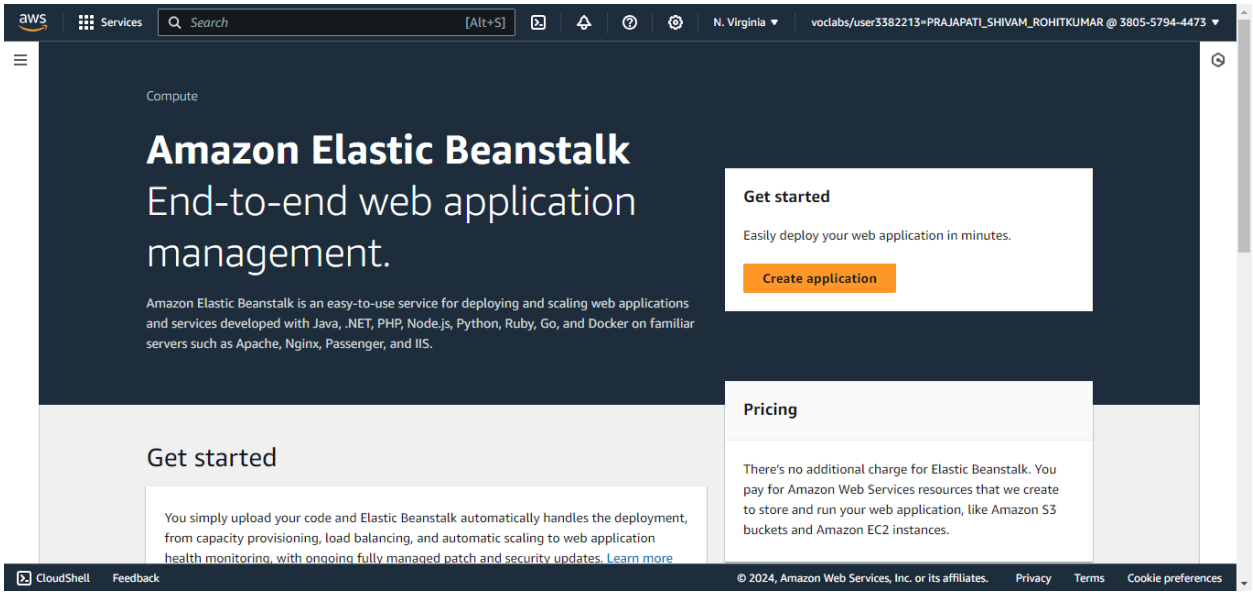
Experiment No :2

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.

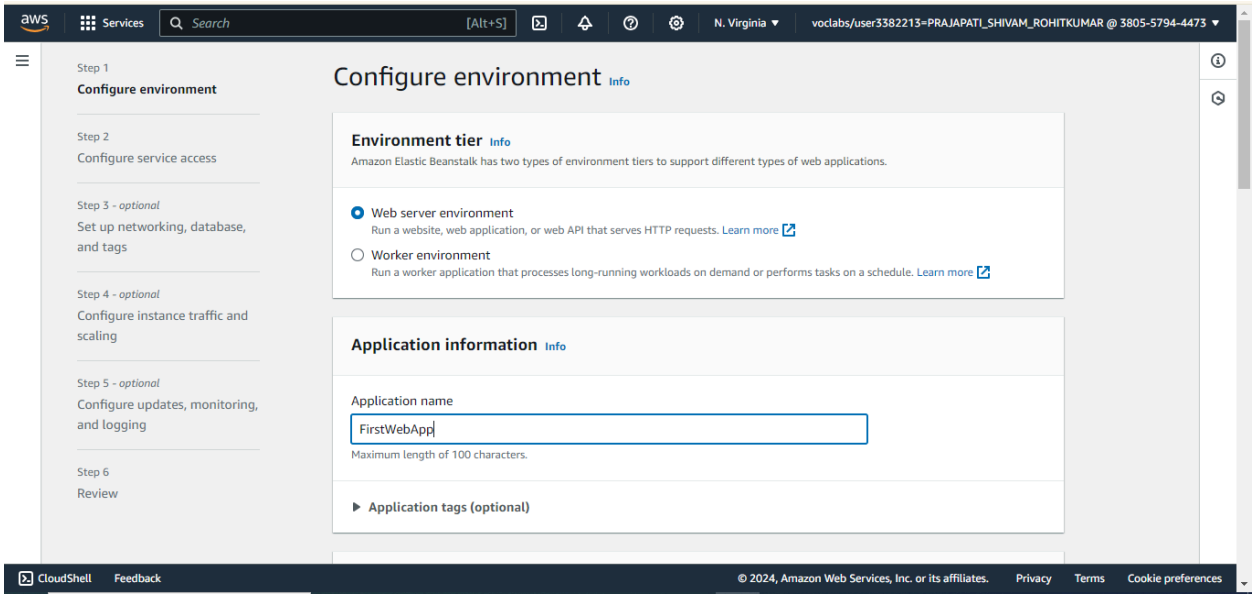
Step 1: Login to your AWS console. Search for Elastic Beanstalk in the searchbar near services.

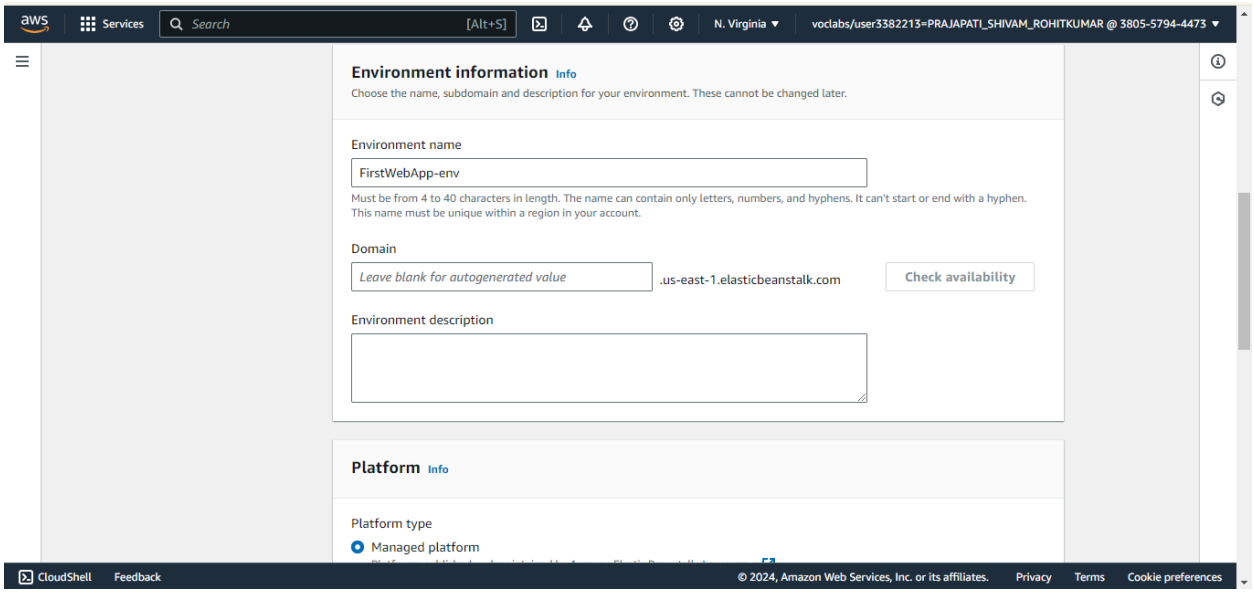


Step 2: Go to Elastic Beanstalk and click on Create Application

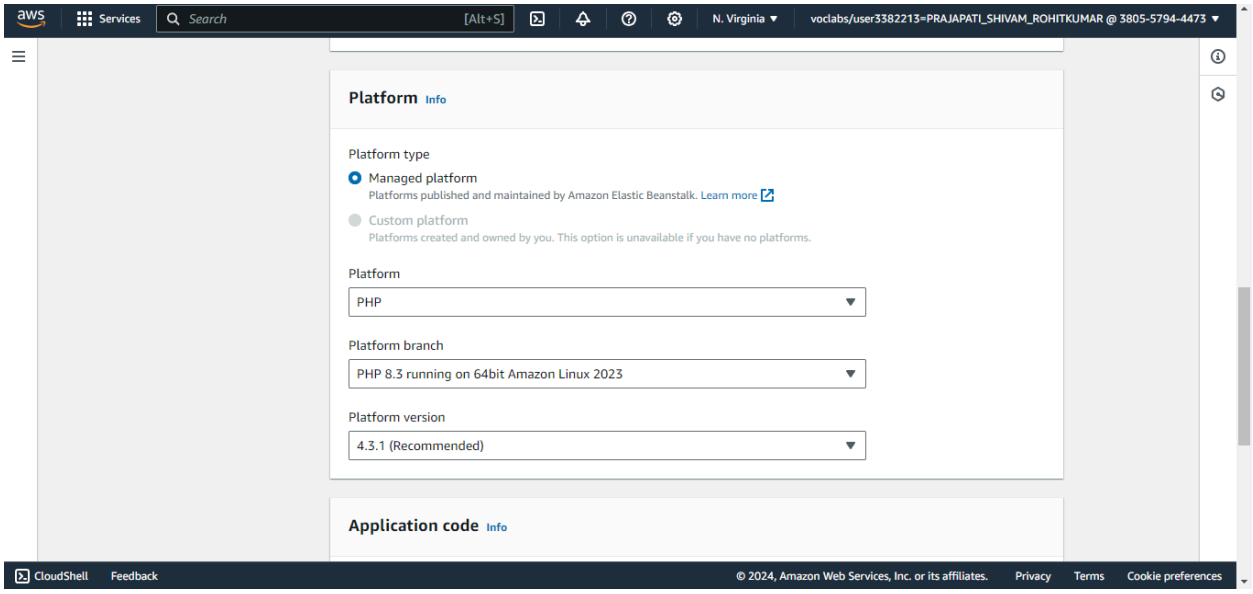


Step 3: Enter the name of your application. Scroll down and in the platform, select platform as PHP. Keep the application code as Sample Application. Set the instance to single instance. Click on NEXT.

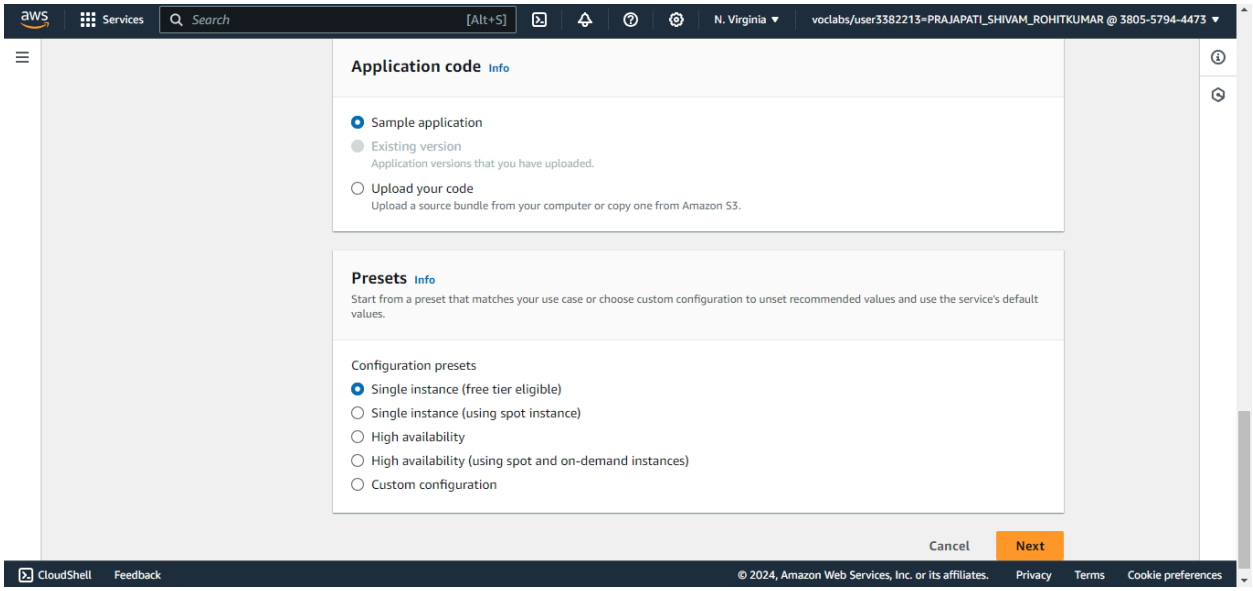




Environment name will be automatically displayed based on the Application name provided by the user

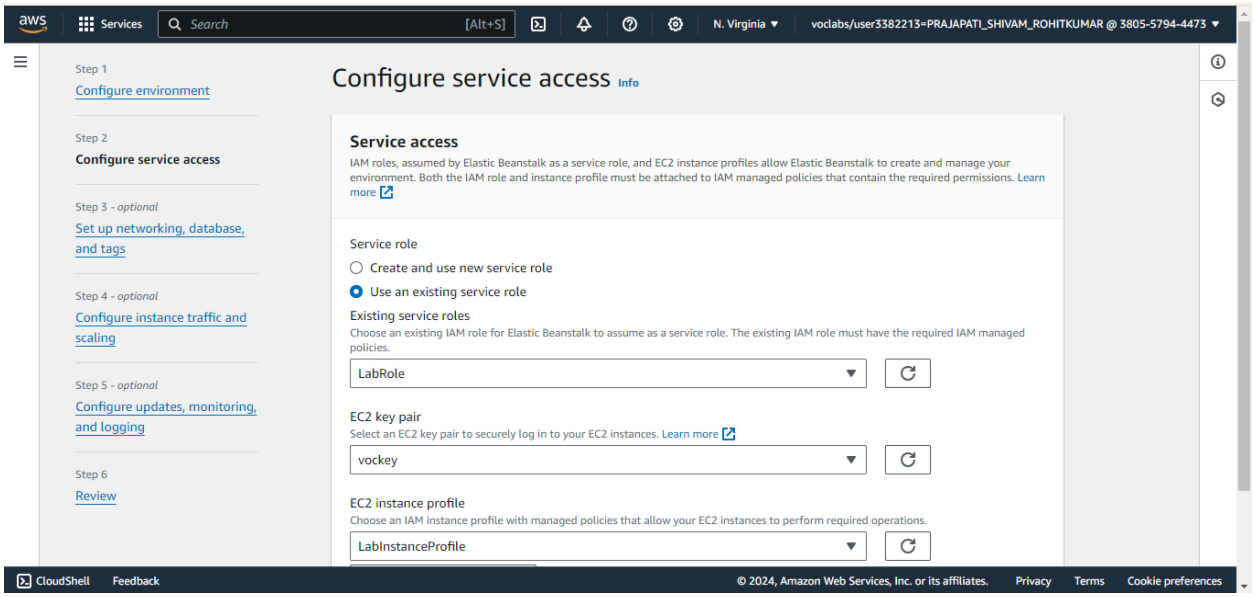


On Selecting Platform type as PHP then Platform branch and Platform version will be displayed Automatically

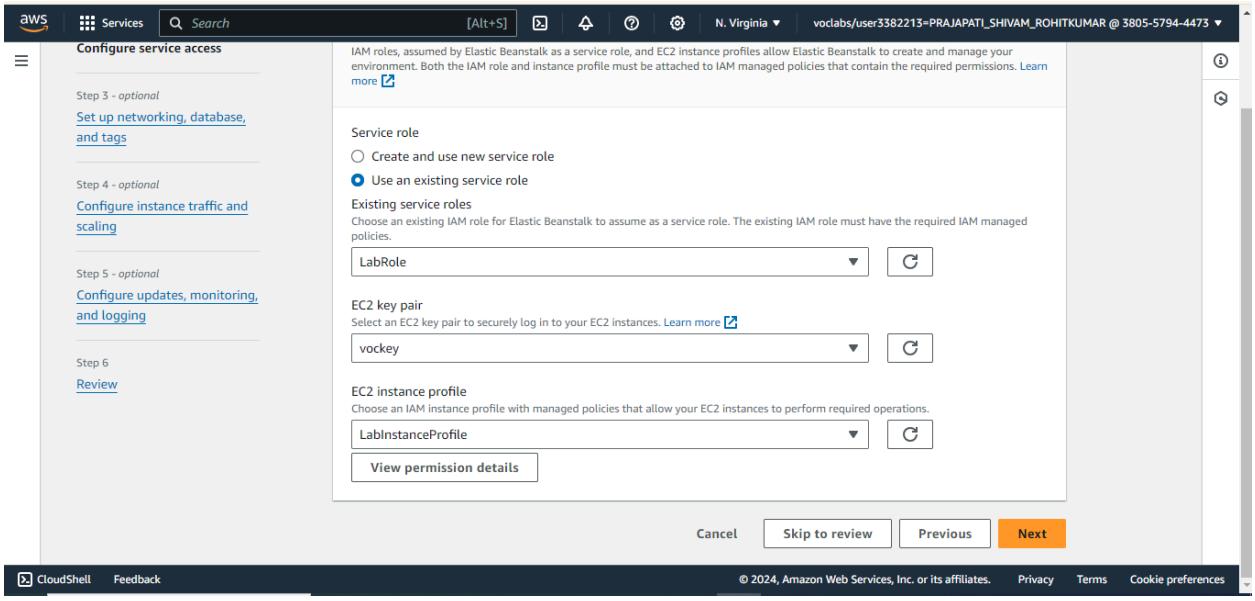


After doing the necessary steps Click on Next button

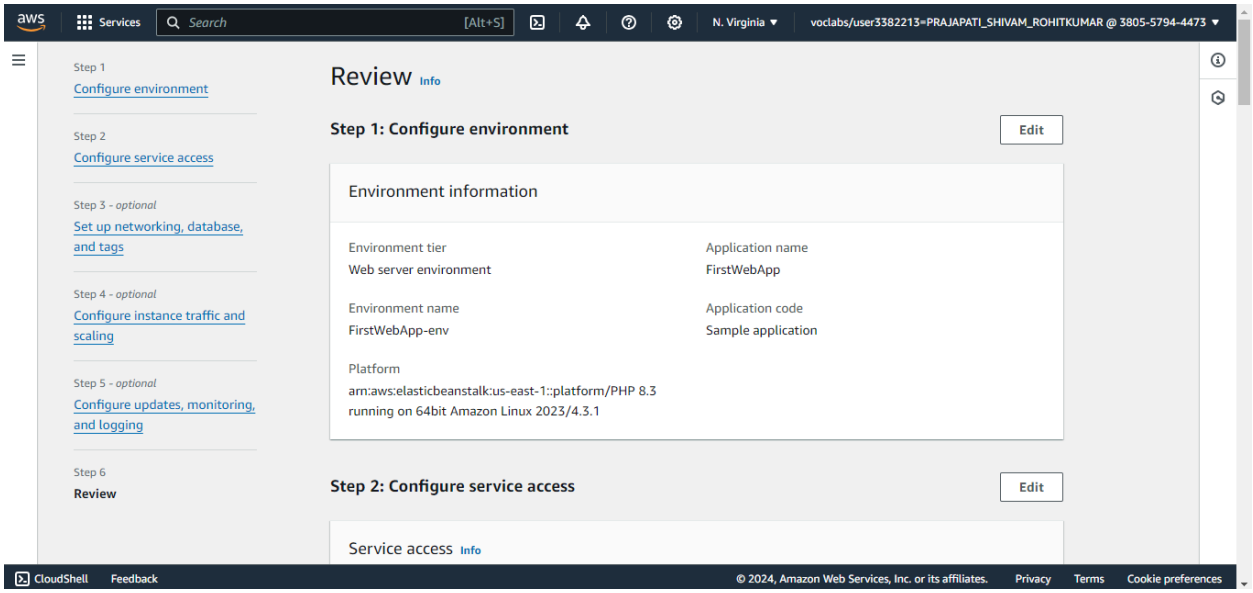
Step 4: Use an existing service role and choose whatever service role is present on your account Because existing service role is different for different user and it is automatically generated ,select the key-pair which is default given as vockey which is going to be sued to securely log in to your EC2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

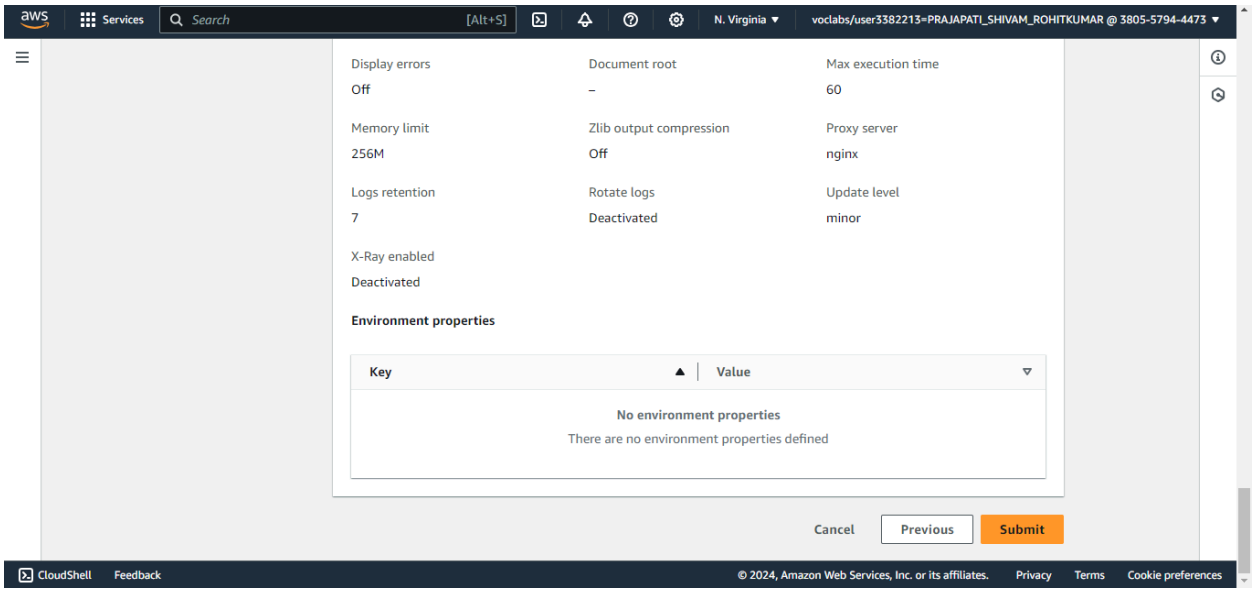


Step 5: Click on Skip to Review Button



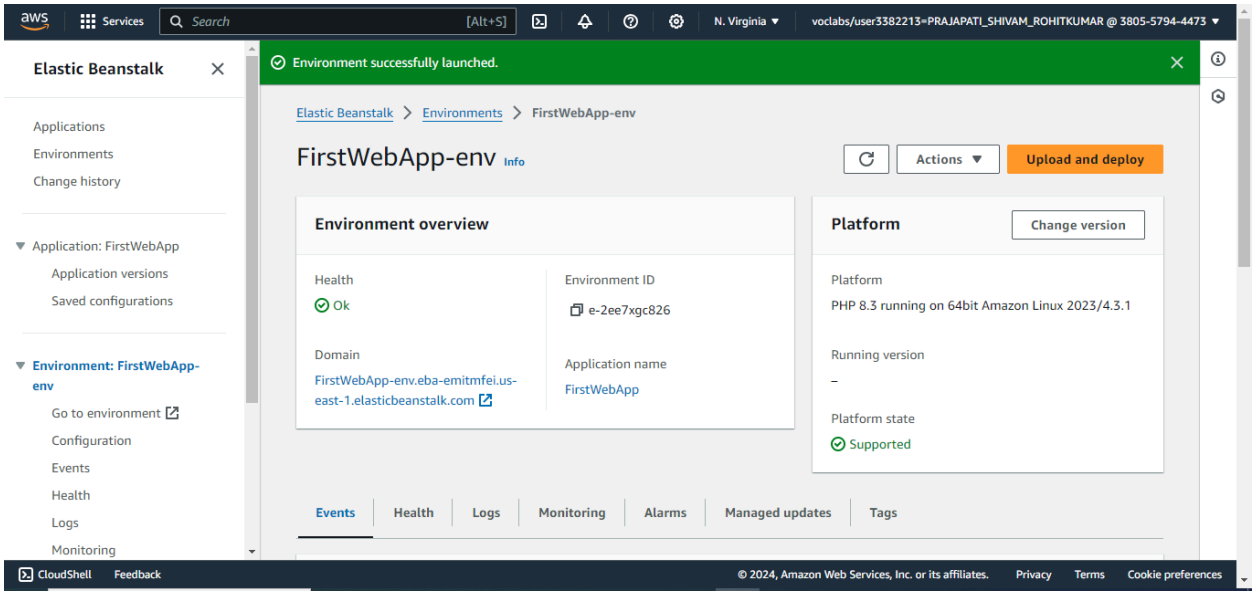
Step 6: Review the settings that you have set up for your application and submit your application thereafter by clicking on submit button



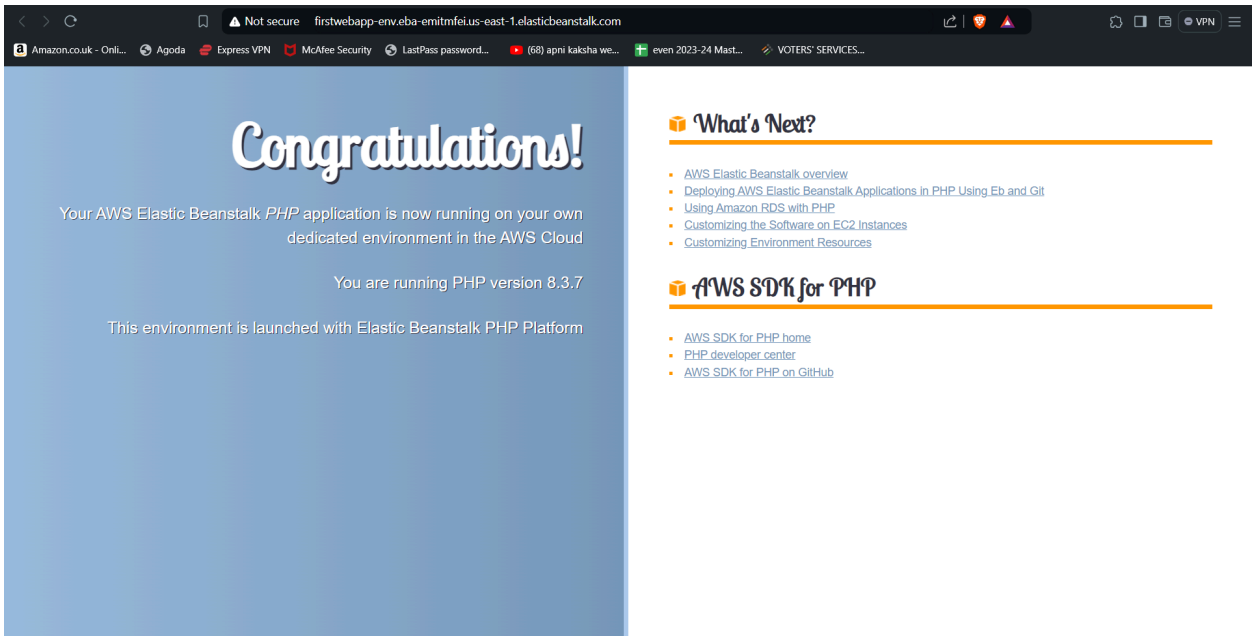


Click on the Submit

Step 7: We have Successfully launched our environment

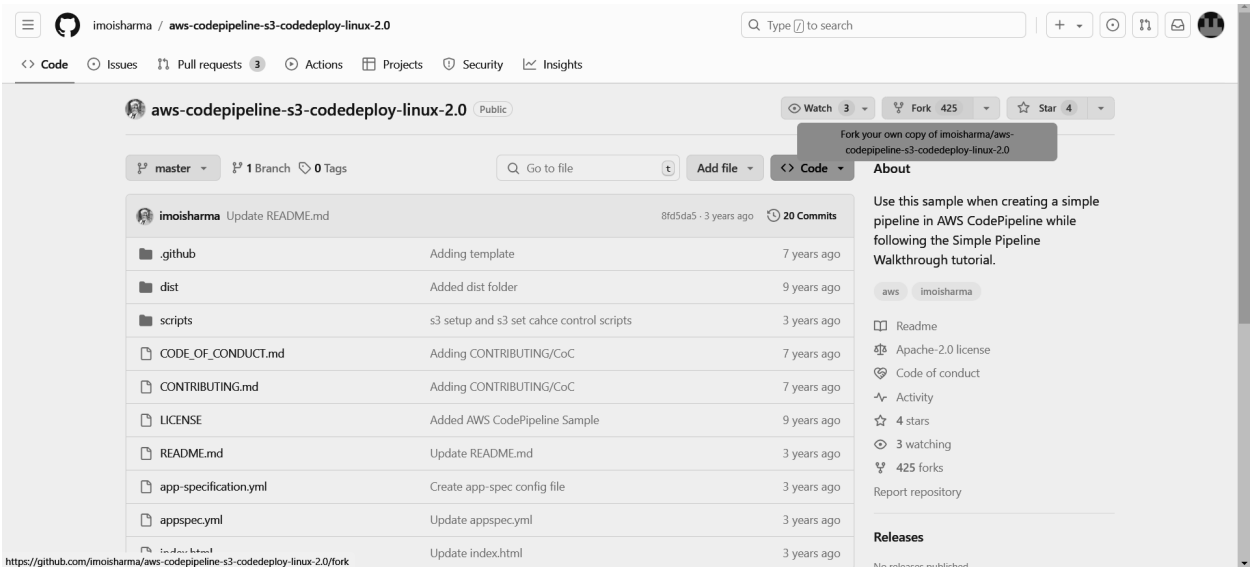


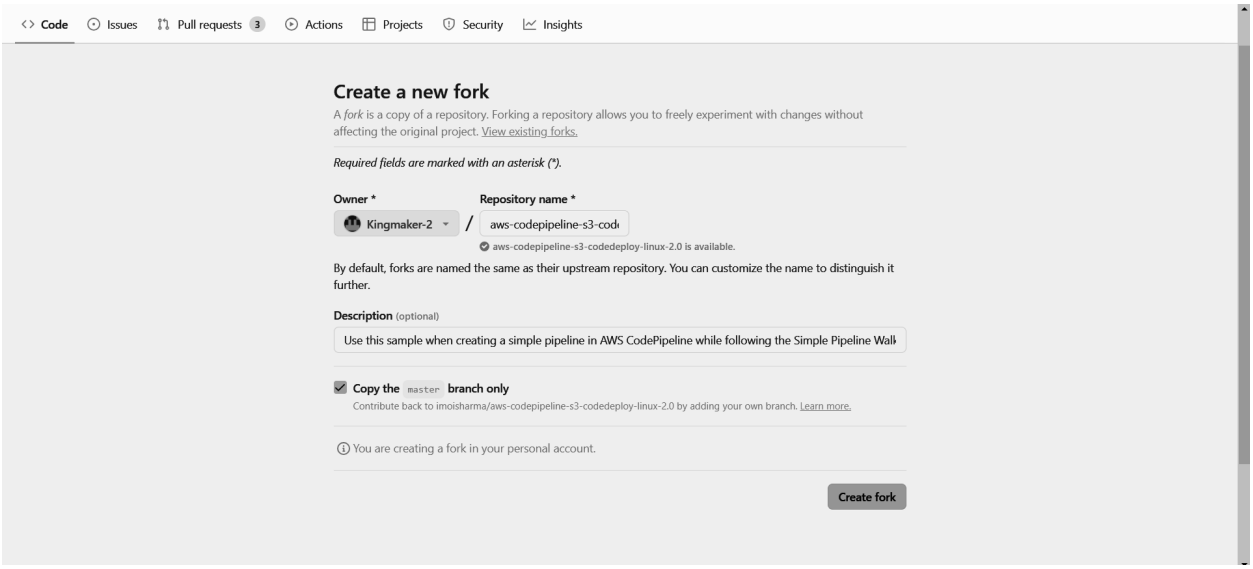
Step 8: Click on the link under domain section and it will redirect you to a new page .



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.

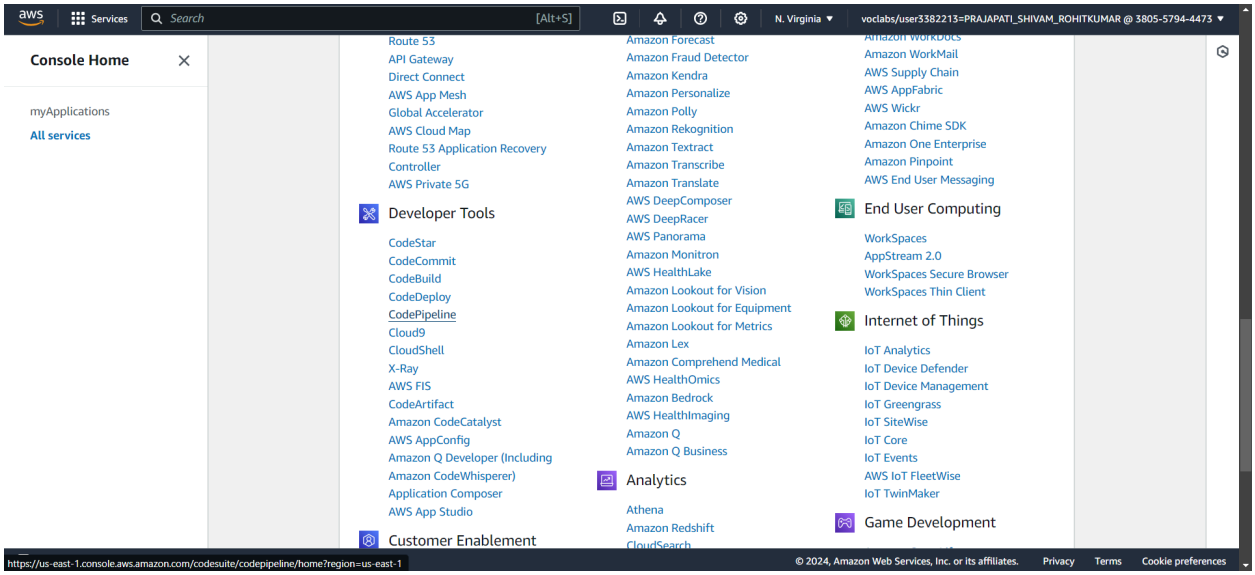
https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux



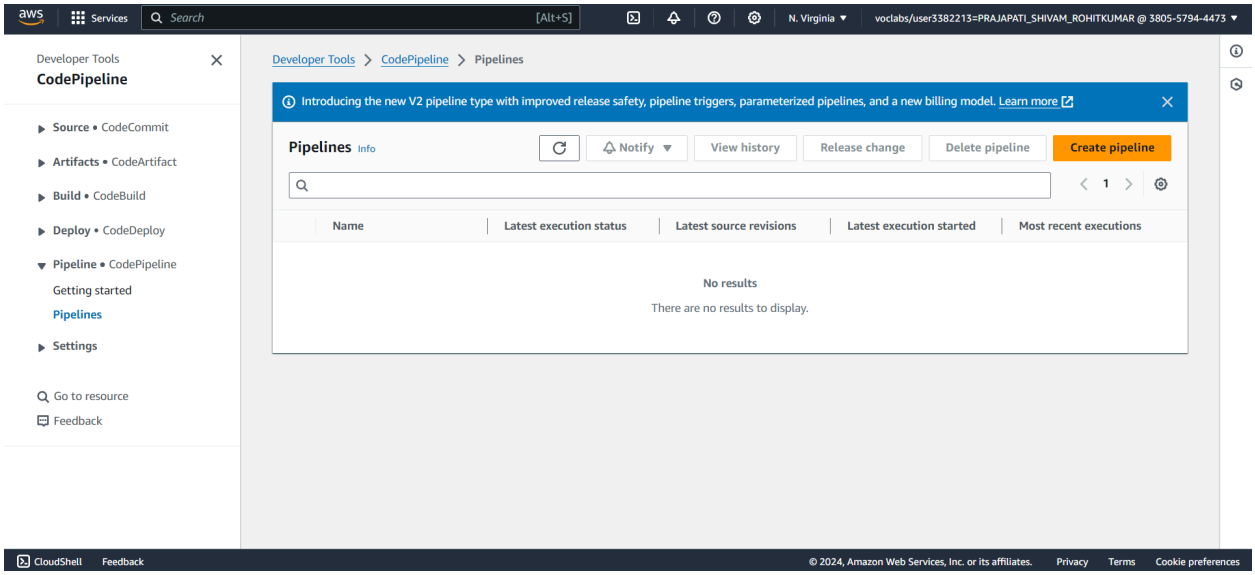


Click on Create Fork this will Duplicate the entire repository to your Github Account

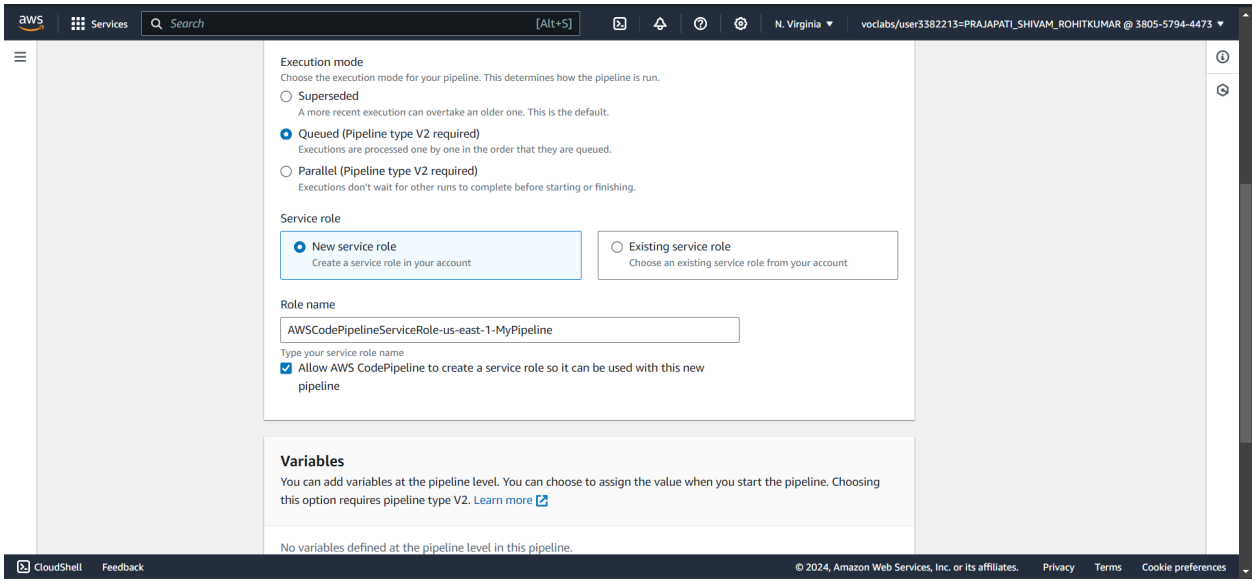
Step 10: Search CodePipeline in the services tab and click on it.

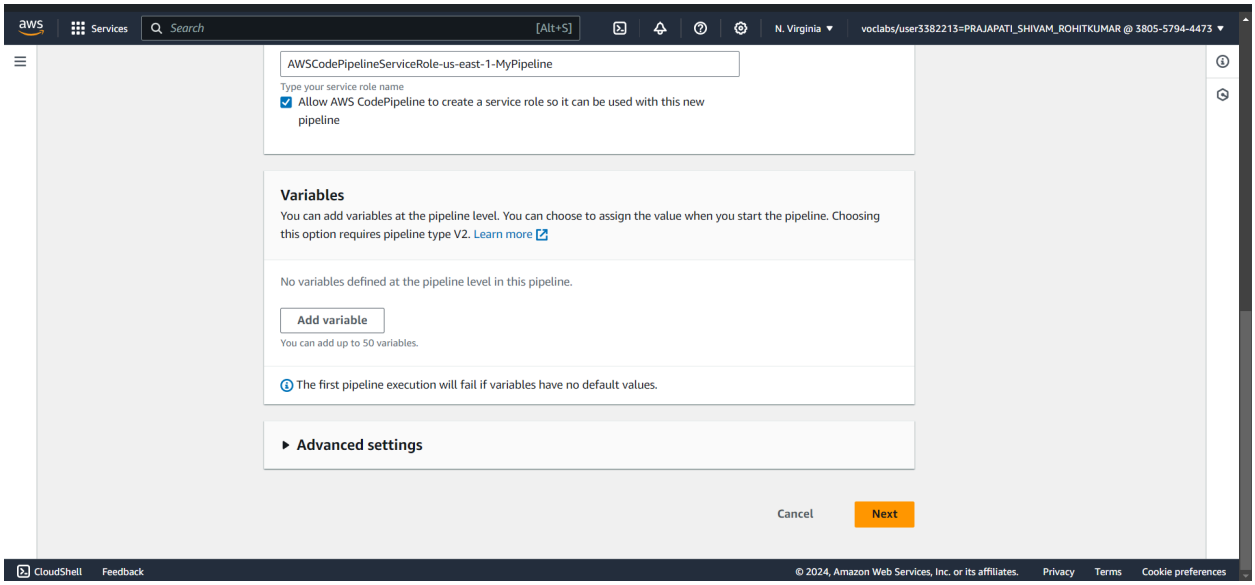


Step 11: Click on Create Pipeline.



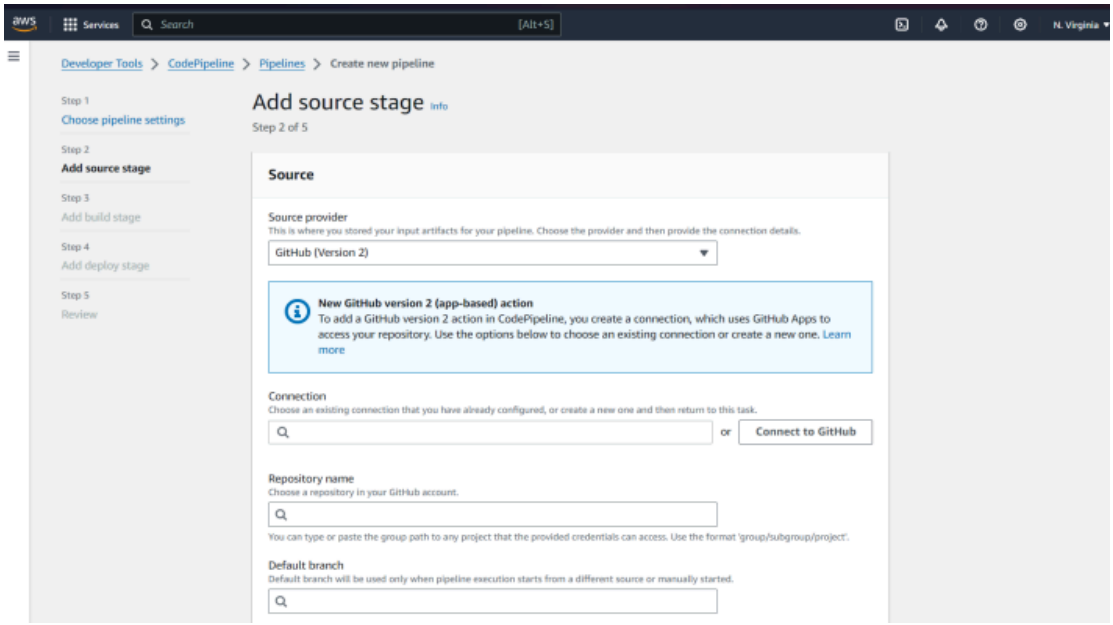
Step 12: Give a name to your Pipeline. A new service role would be created with the name of the pipeline. Make Sure the service role should be New Service Role



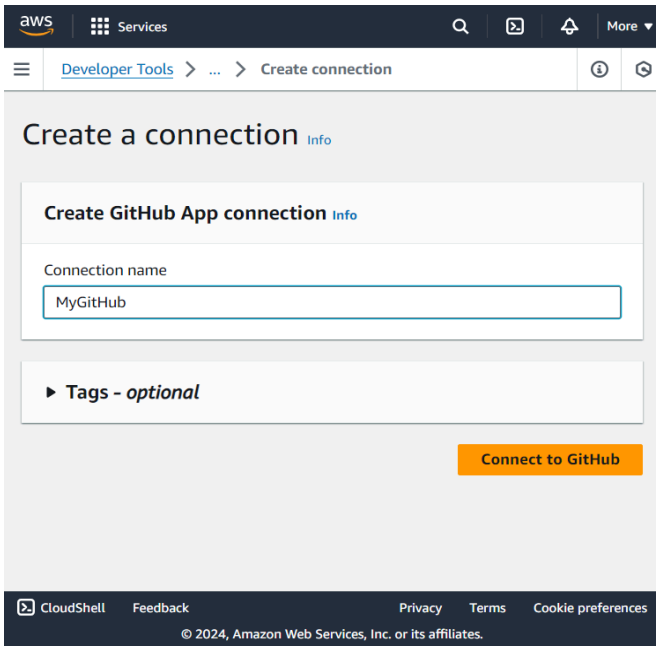


Under variables section it is not necessary to add variables
Click on Next Button

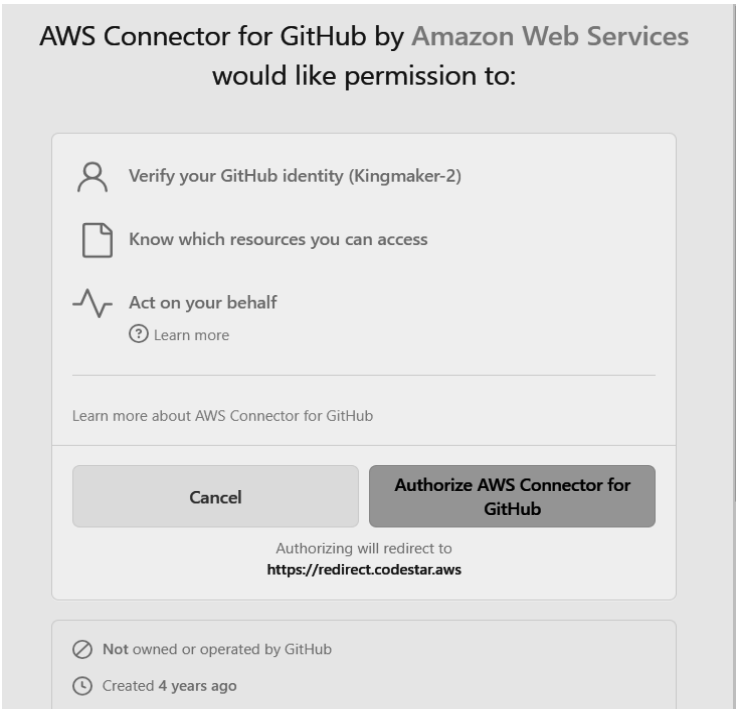
Step 13: Select a source provider (as GitHub Version (2)). Click on connect to Github (*This part have to be done in the personal account of aws as in academy account it won't allow you to create pipeline with github version 1 or 2*)



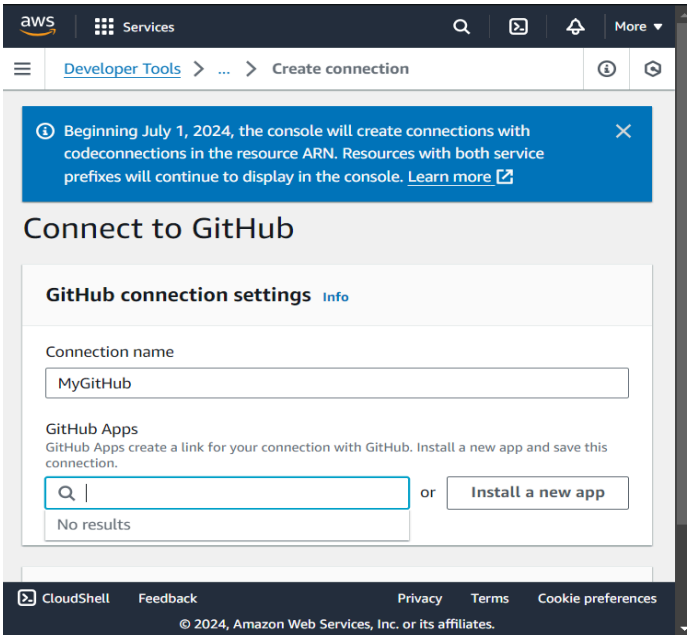
Step 14: Give a name to your GitHub app Connection and click on Connect. This will give you a prompt to either to select a GitHub app or to install a new app. If it is your first time, click on Install a new app.



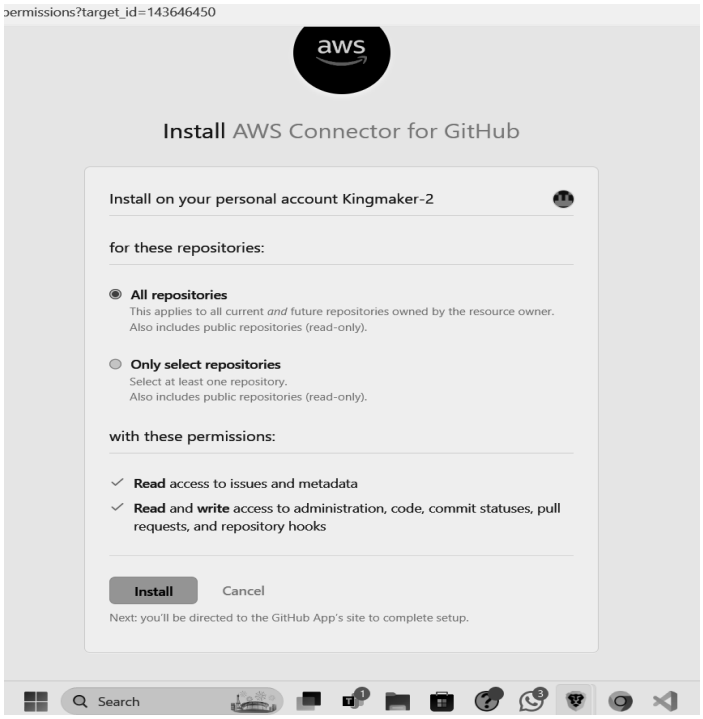
Step 15: After giving the connection name then AWS will Authorize the user



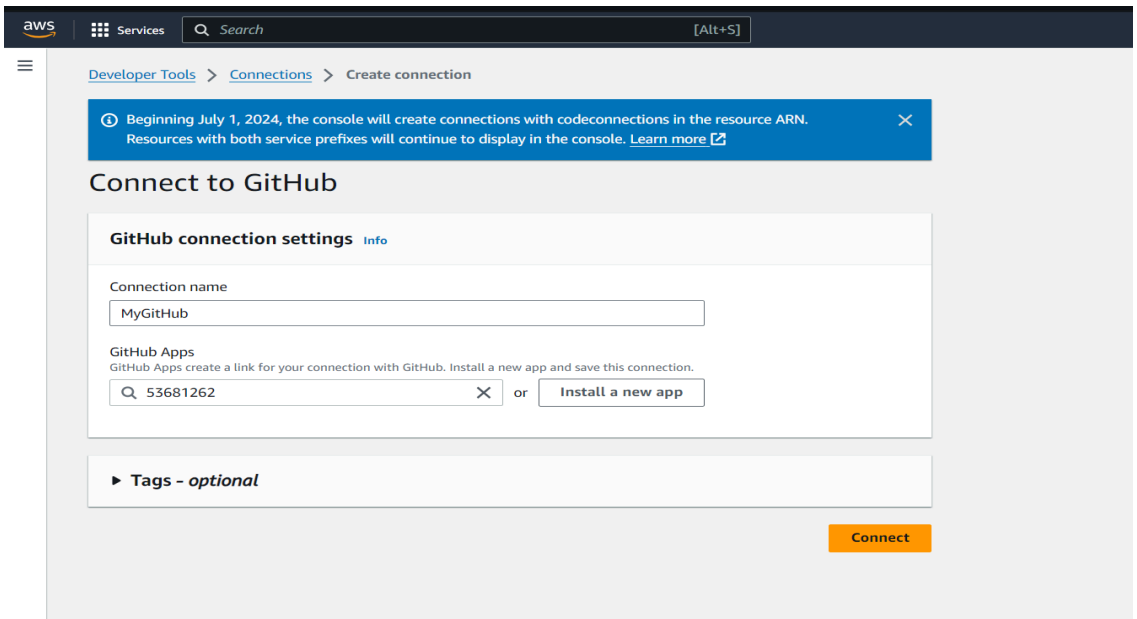
Step 16: Click on New App as we are doing for the first time which was already mentioned in step 14

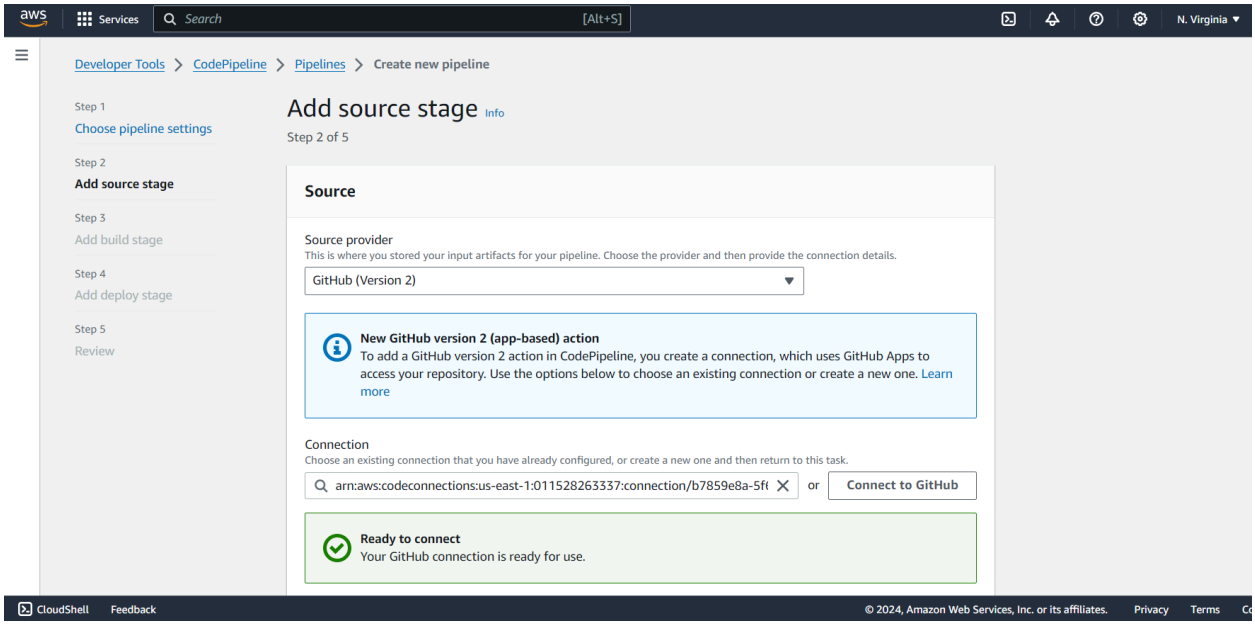


Step 17: This will direct you to install AWS connector on your GitHUb .Install it to your account and give it its permissions

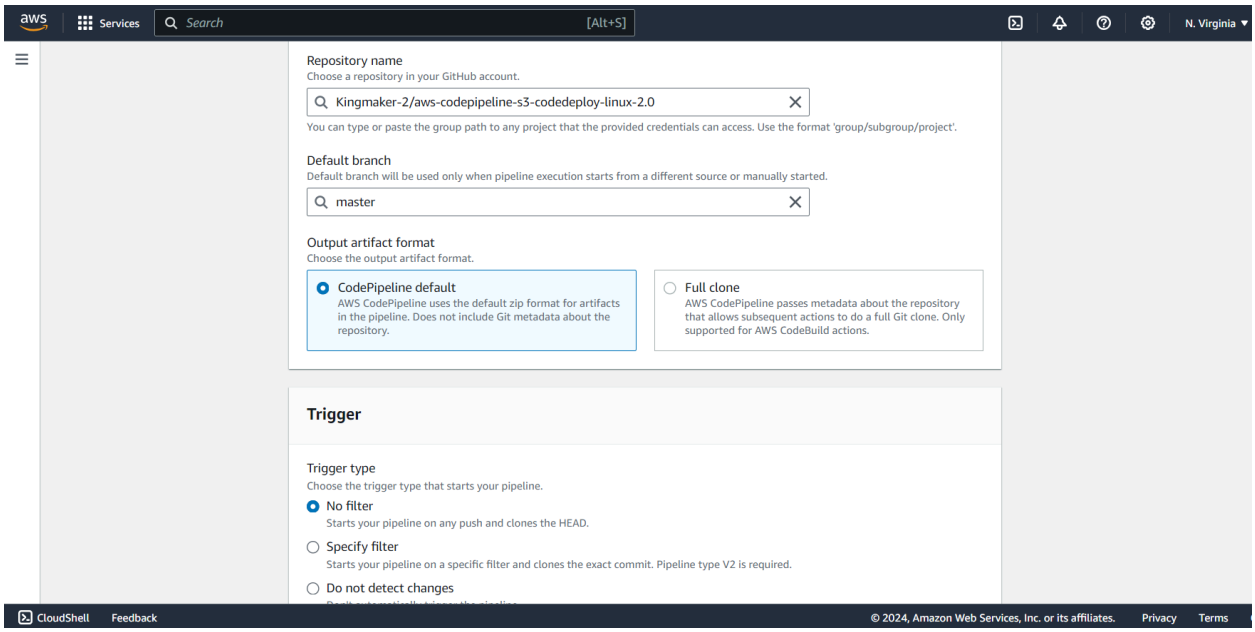


Step 18: After the app is set up, it gives the number in the text field. Click on Connect. After clicking on connect, the link is shown in the connection field and AWS shows that GitHub connection is ready to use.

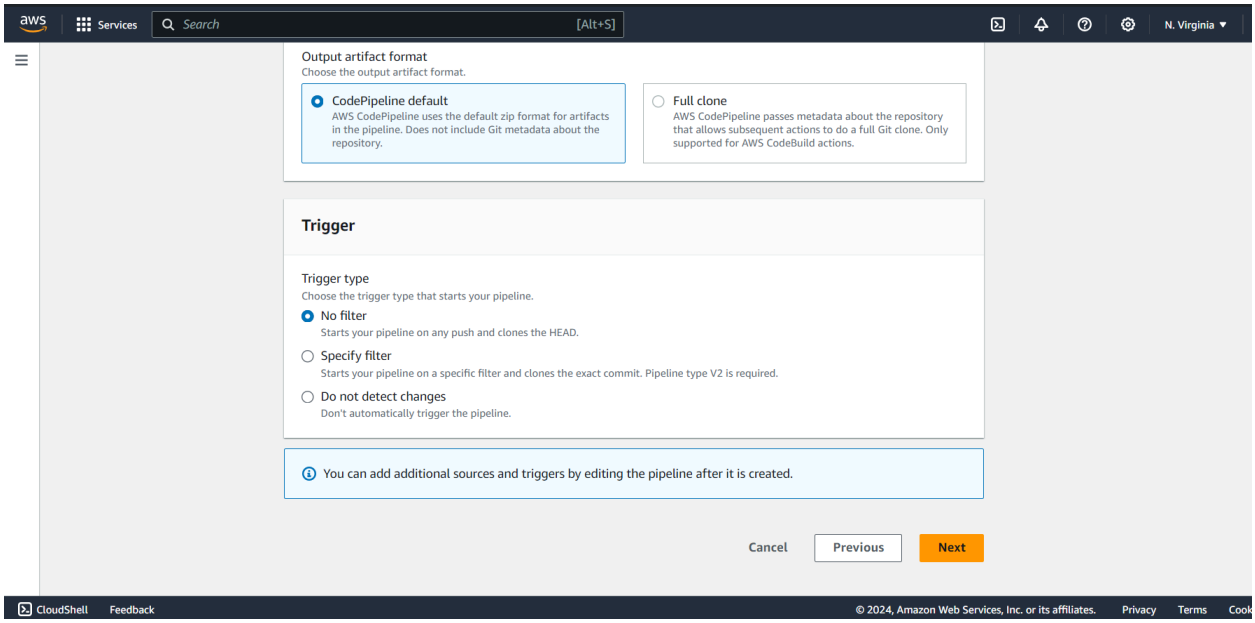




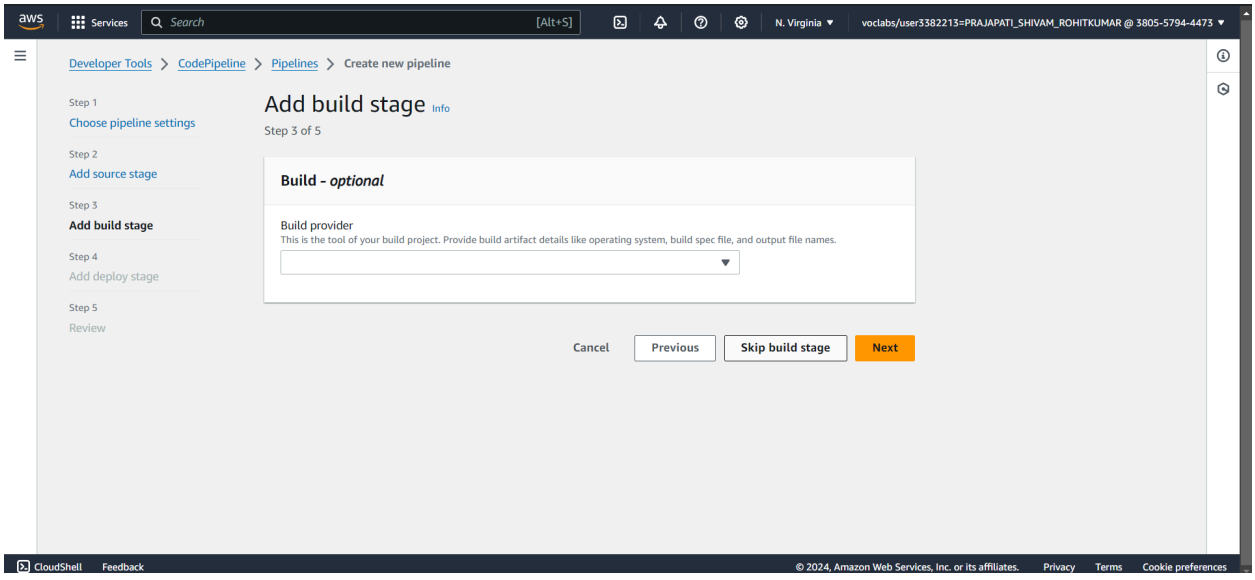
Step 19: Select the repository that you had forked to your GitHub. After that select the branch on which the files are present (default is Master).

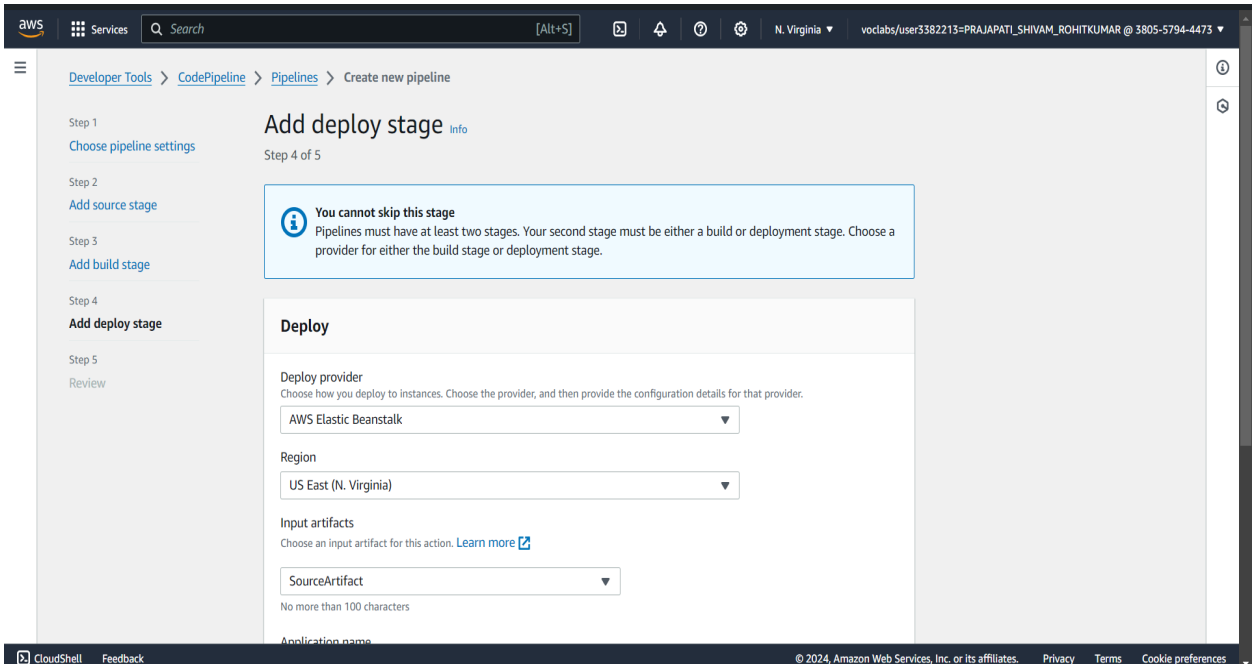


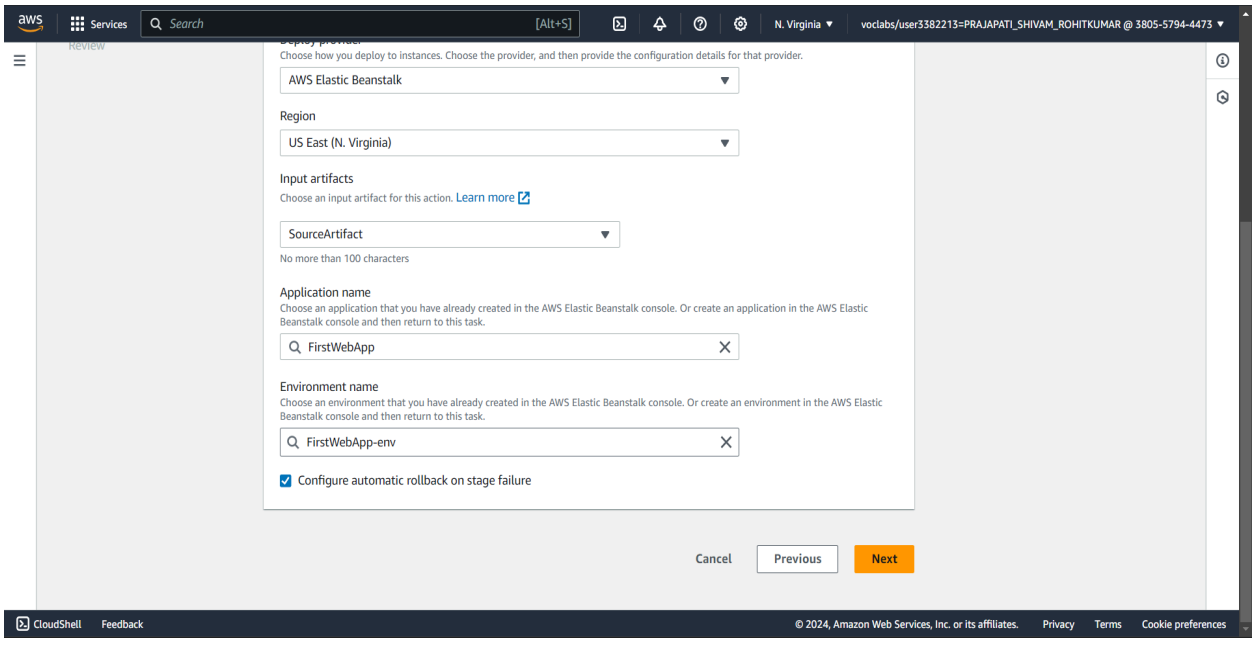
Step 20: Set the Trigger type as no filter. This would allow it to the website to update as soon as some change is made in the github. Click on Next



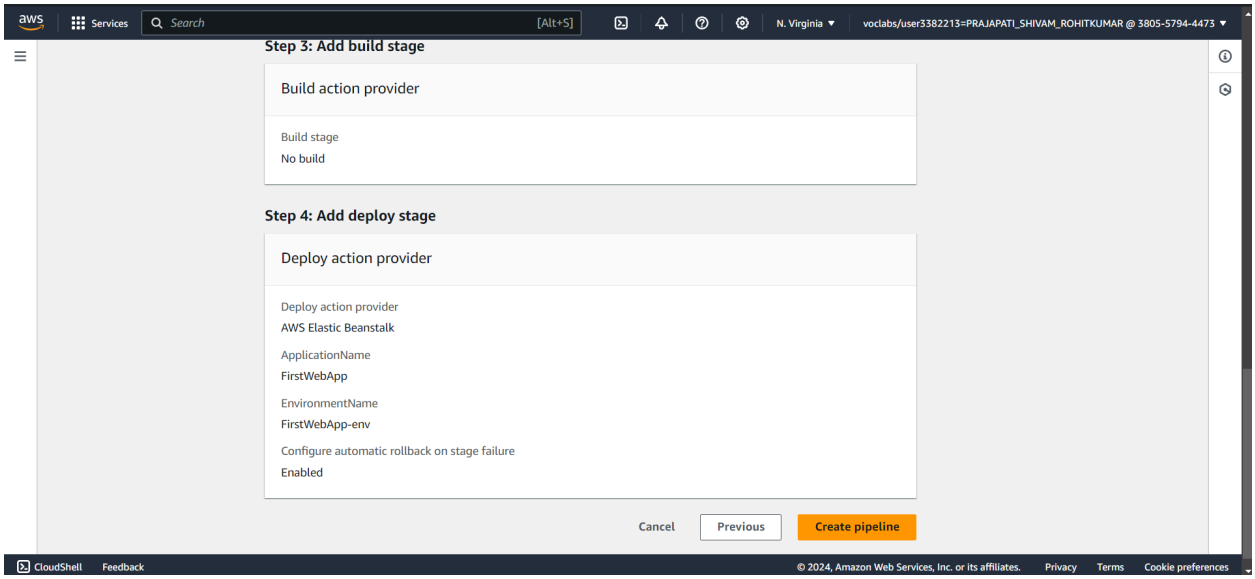
Step 21: Skip the build stage and go to Deploy. Select the deploy provider as AWS Elastic Beanstalk and Input Artifact as SourceArtifact. The application name would be the name of your Elastic Beanstalk. Then click on next.



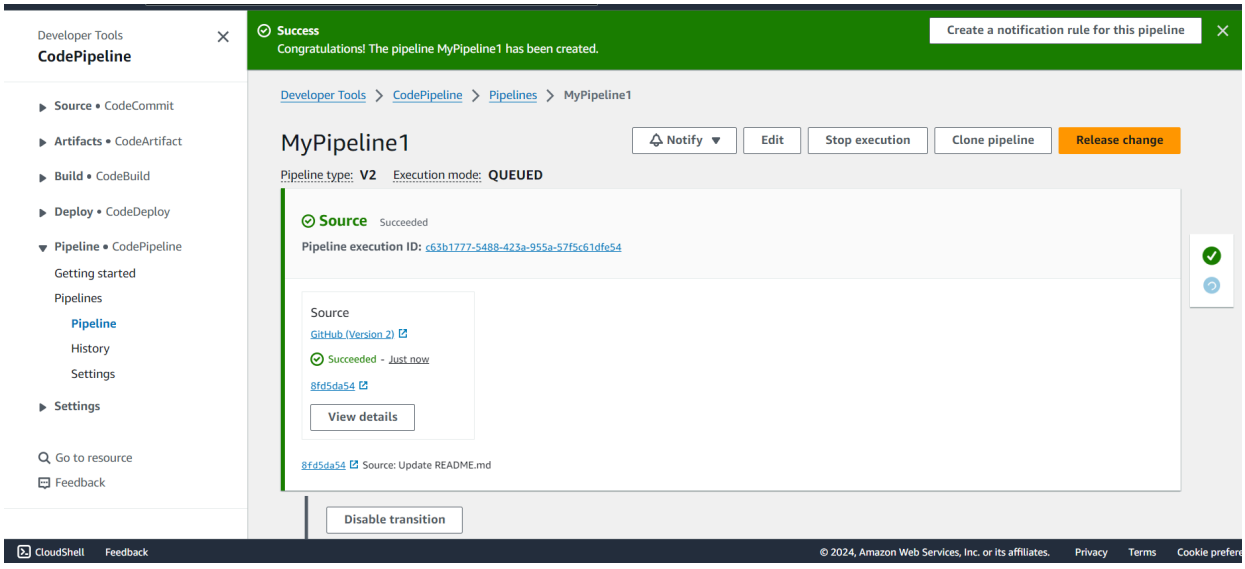

Services
Search
[Alt+S]
N. Virginia
voclabs/user3382213=PRAJAPATI_SHIVAM_ROHITKUMAR @ 3805-5794-4473
Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Add deploy stage Info
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
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

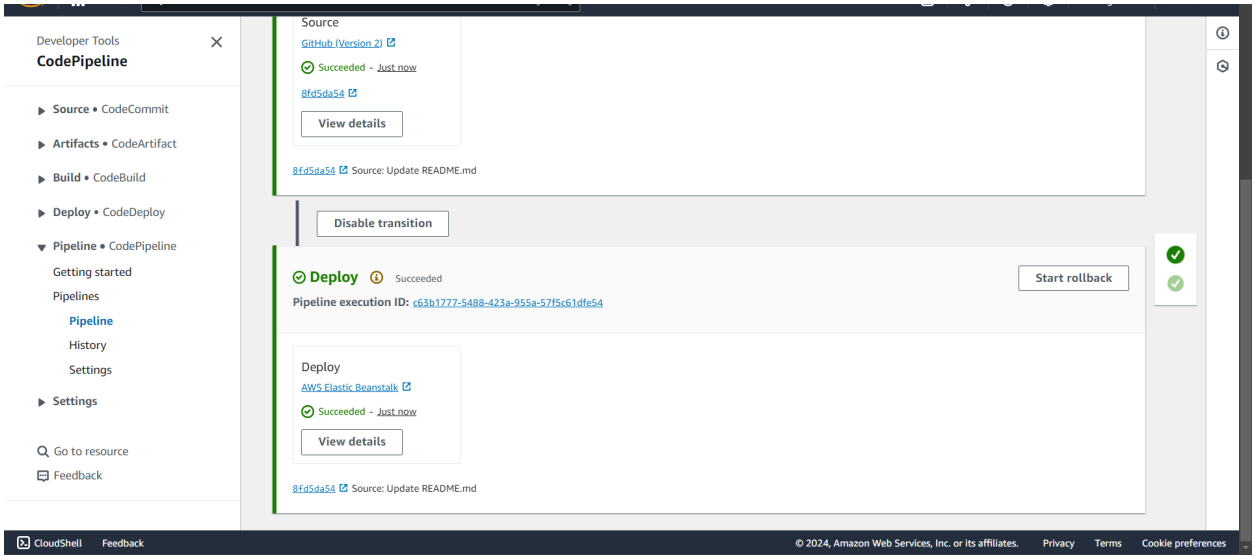

Services
Search
[Alt+S]
N. Virginia
voclabs/user3382213=PRAJAPATI_SHIVAM_ROHITKUMAR @ 3805-5794-4473
Review
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS Elastic Beanstalk
Region
US East (N. Virginia)
Input artifacts
Choose 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



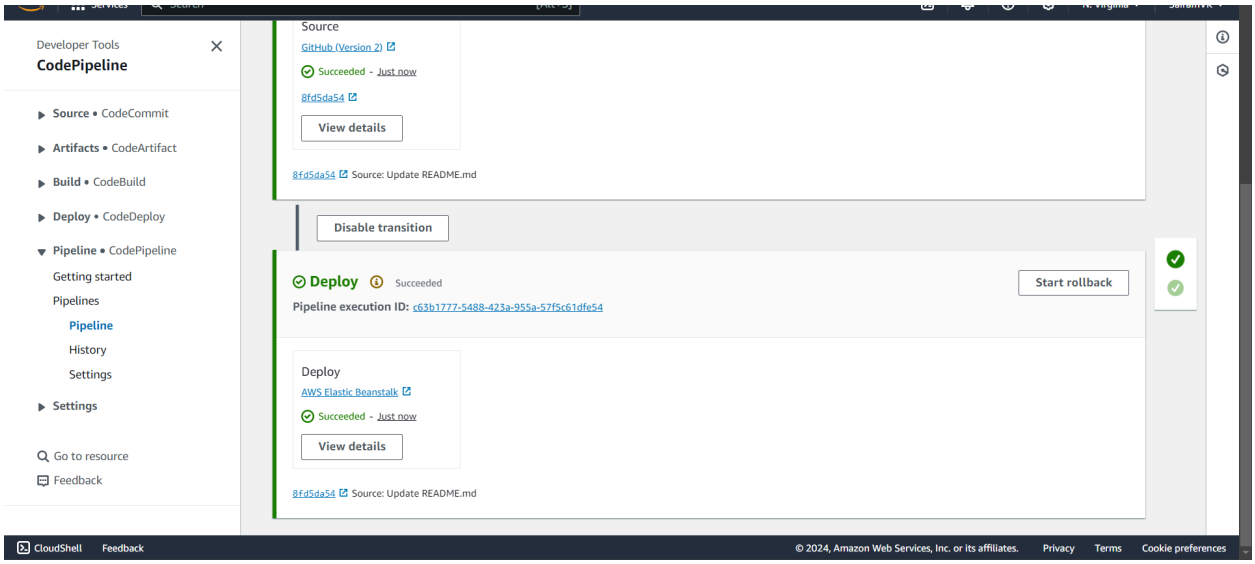
Step 23: If the pipeline is successfully deployed, this screen comes up where the source is set up and then it is transitioned to deploy



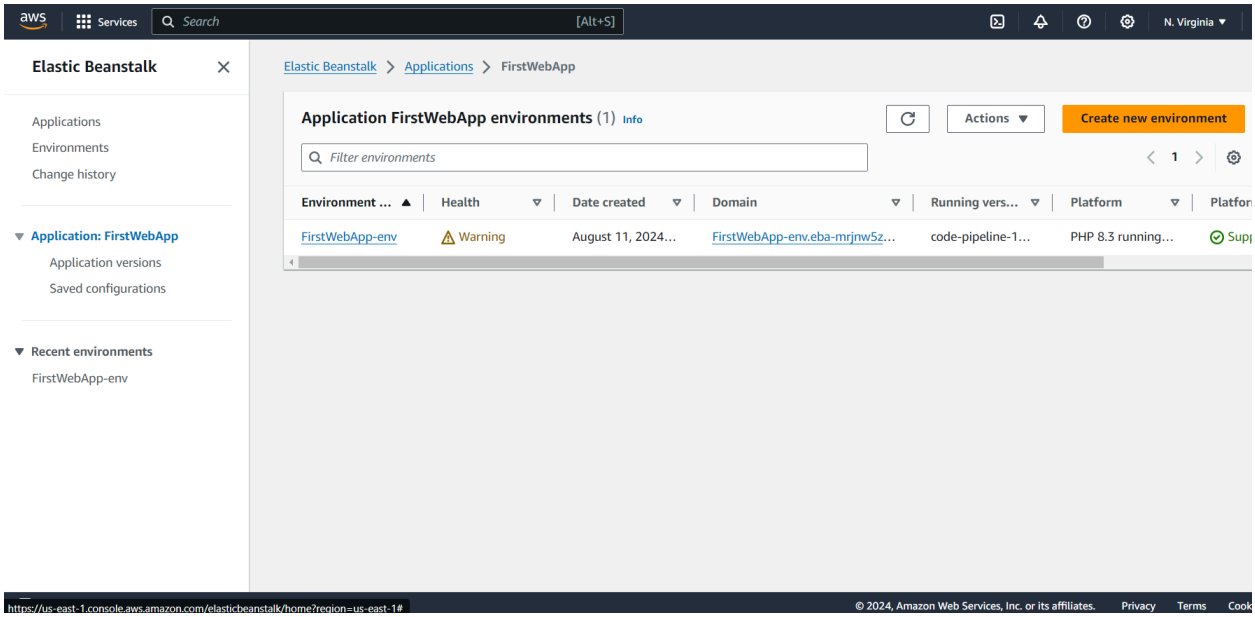


Deploy is done successfully as we can blue tick mark at the right corner

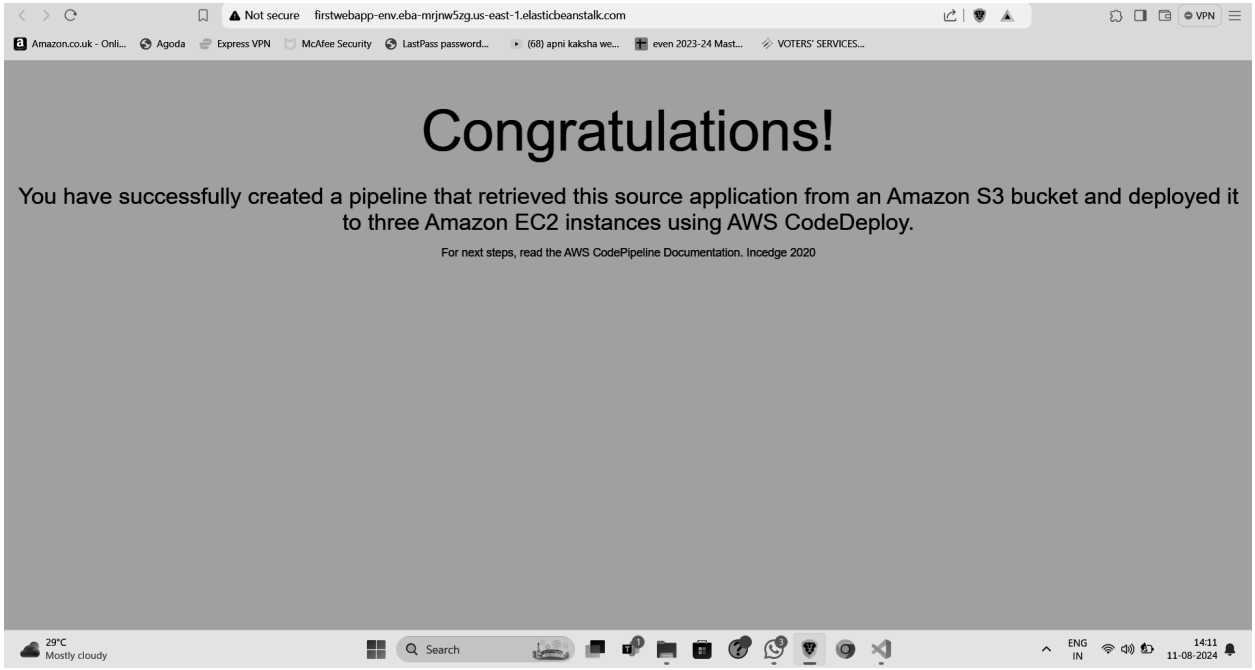
Step 24: Once the deployment is complete, click on the AWS Elastic Beanstalk under Deploy.



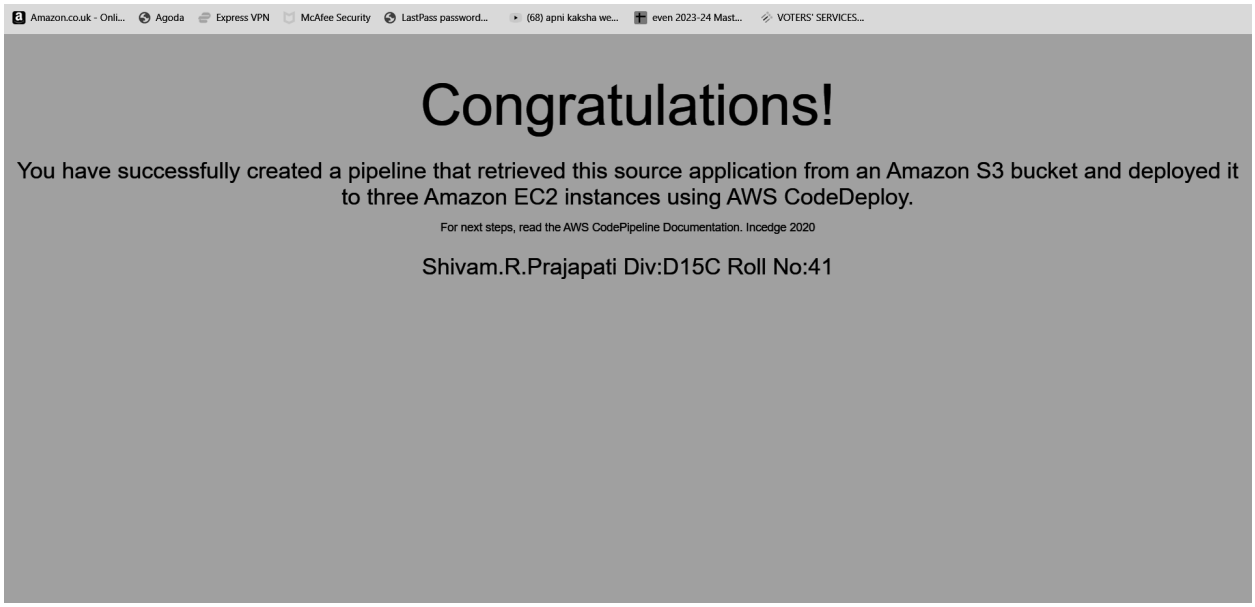
Step 25: This will redirect you to the application screen of Elastic Beanstalk. Click on the link shown under Domain



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.



****