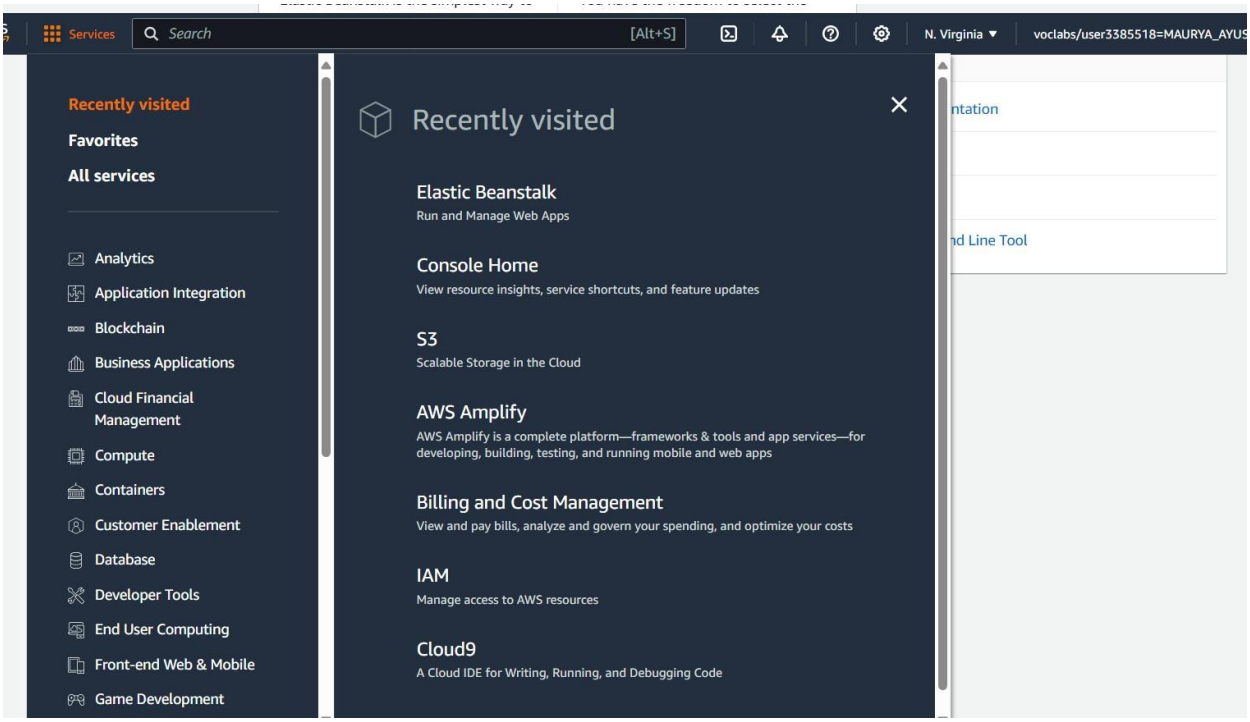
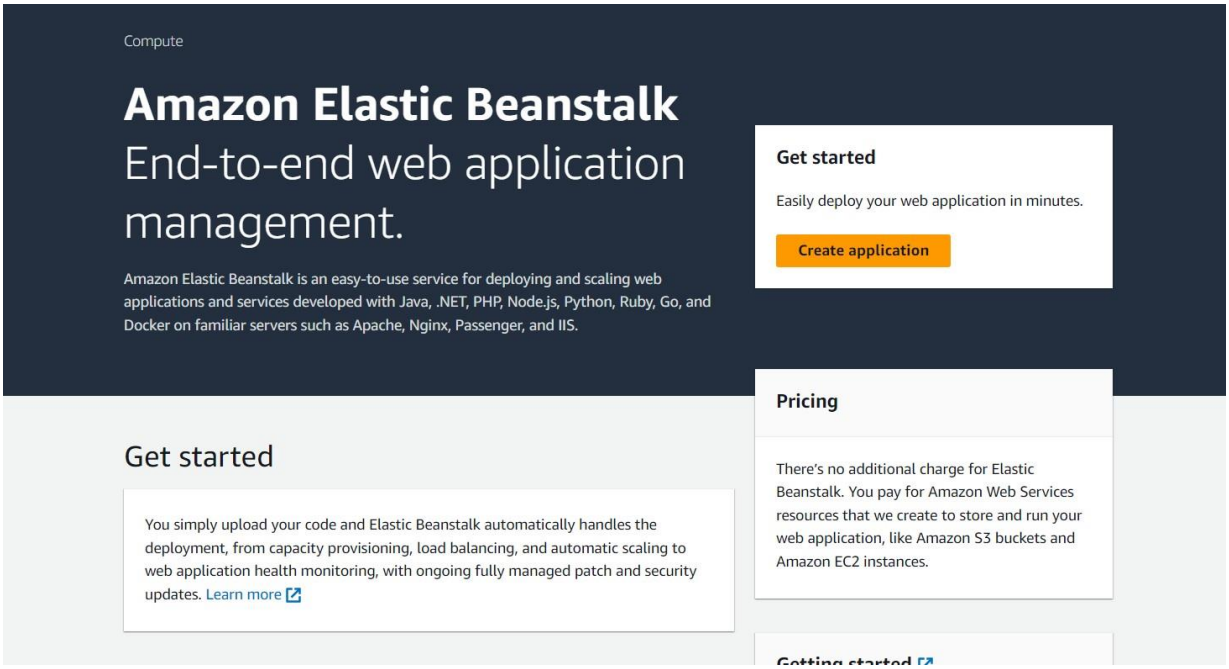
Name :Aditya kirtane ,Roll No.26 , Div:D15C

Experiment No.2

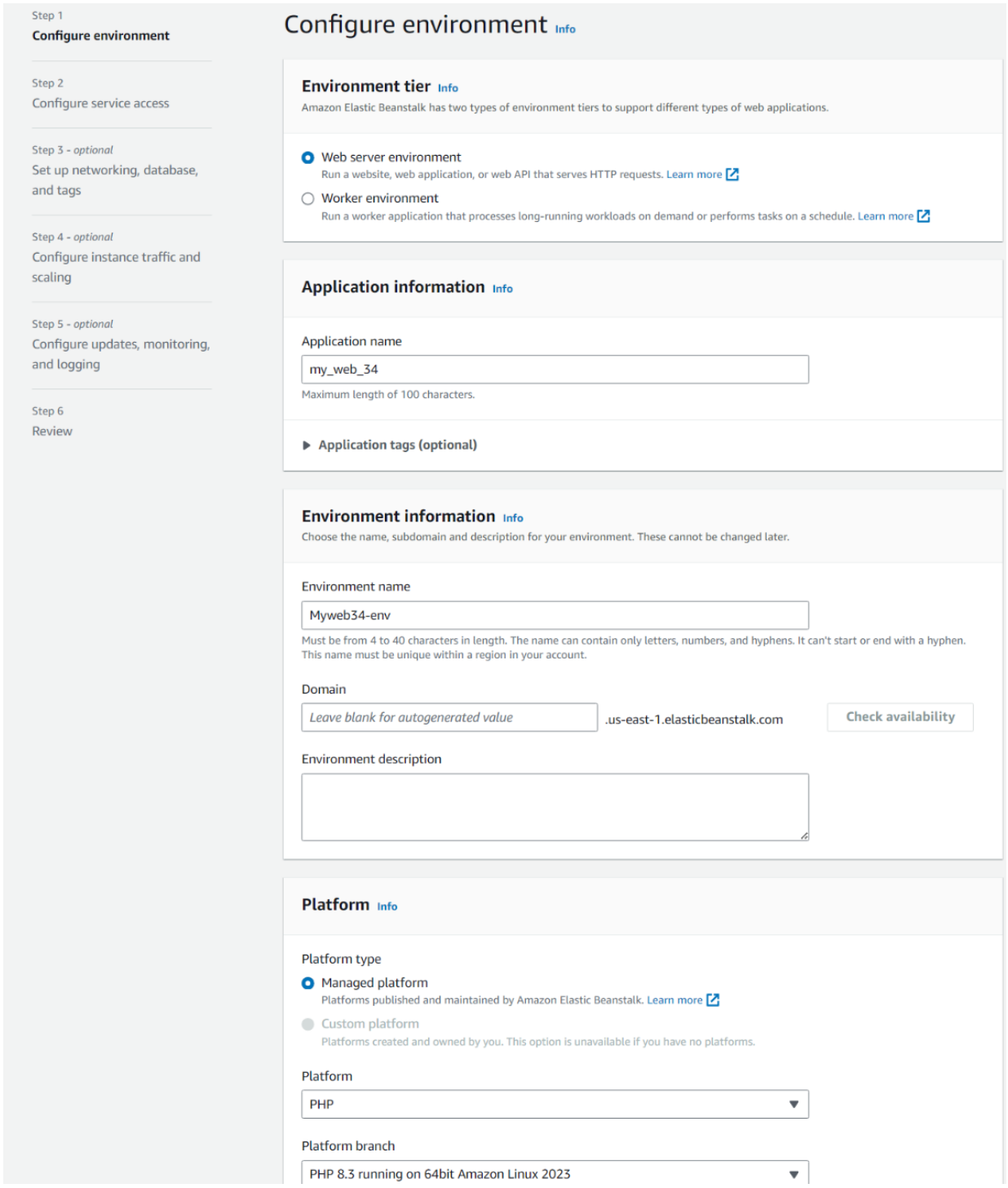
Step 1: Login to your AWS console. Search for Elastic Beanstalk in the searchbar near services.

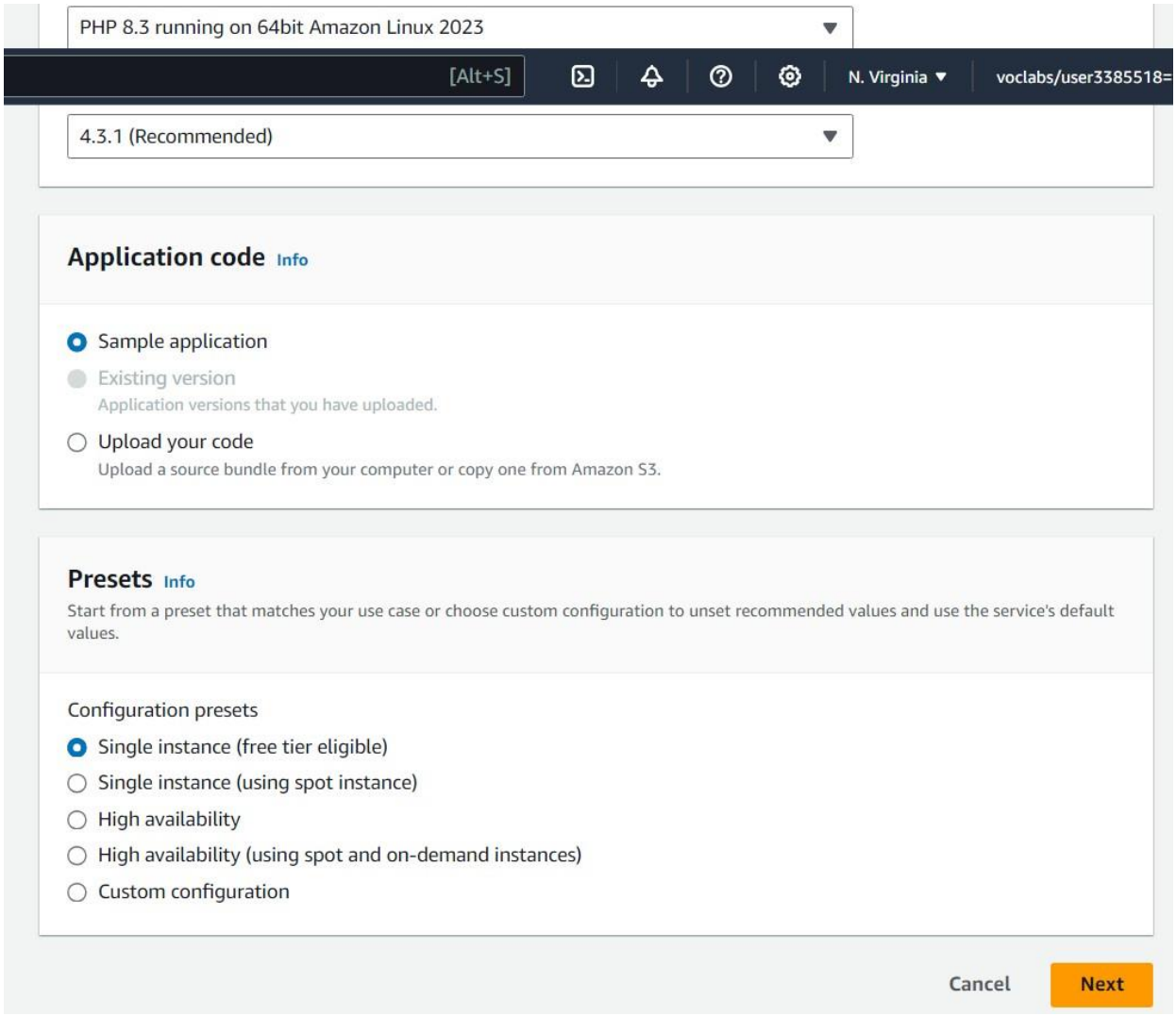


Step 2: Go to Elastic Beanstalk and click on Create Application

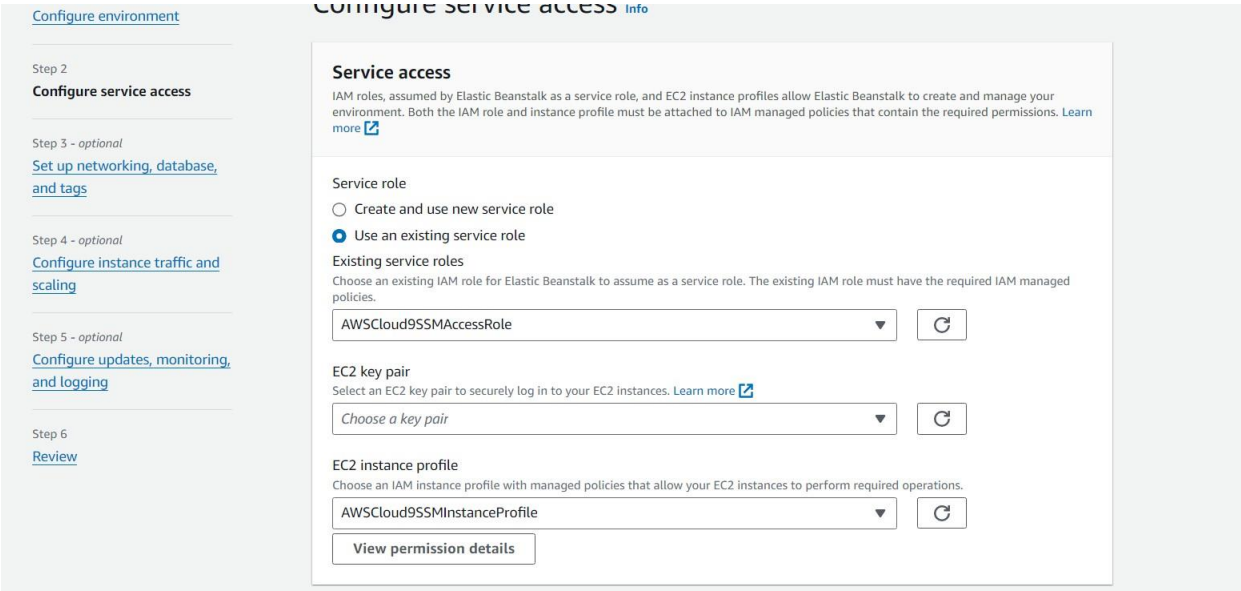


Step 3: Enter the name of your application. Scroll down and in the platform, select platform as PHP. Keep the application code as Sample Application. Set the instance to single instance. Click on NEXT

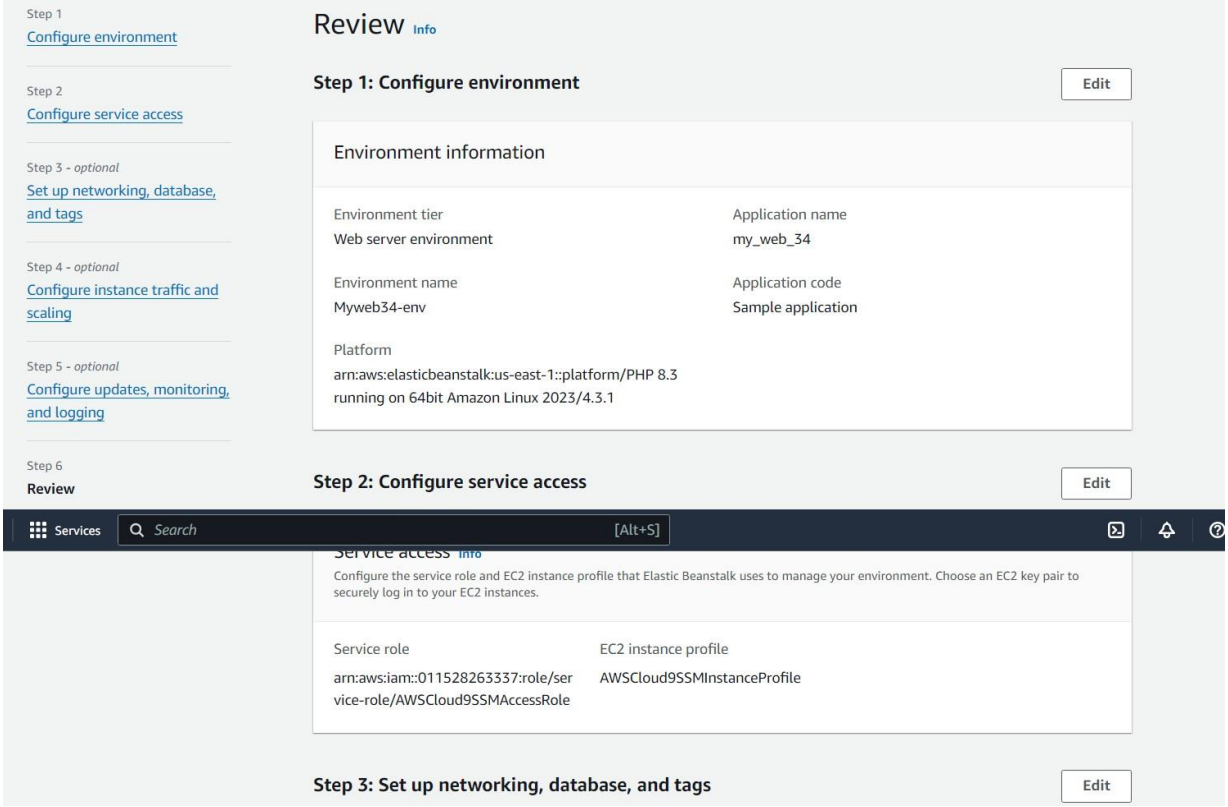



PHP 8.3 running on 64bit Amazon Linux 2023
[Alt+S]
N. Virginia
voclabs/user3385518=
4.3.1 (Recommended)
Application code Info
Sample application
Existing version
Application versions that you have uploaded.
Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.
Presets 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

Step 4 : Use an existing service role and choose whatever service role is available on your account.



Step 5 : Review the settings that you have set up for your application and submit your application.



Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

Monitoring

System enhanced	Cloudwatch custom metrics - instance —	Cloudwatch custom metrics - environment —
Log streaming Deactivated	Retention 7	Lifecycle false

Updates

Managed updates Activated	Deployment batch size 100	Deployment batch size type Percentage
Command timeout 600	Deployment policy AllAtOnce	Health threshold Ok
Ignore health check	Instance replacement	

[Alt+S]

Platform software

Lifecycle false	Log streaming Deactivated	Allow URL fopen On
Display errors Off	Document root –	Max execution time 60
Memory limit 256M	Zlib output compression Off	Proxy server nginx
Logs retention 7	Rotate logs Deactivated	Update level minor
X-Ray enabled Deactivated		

Environment properties

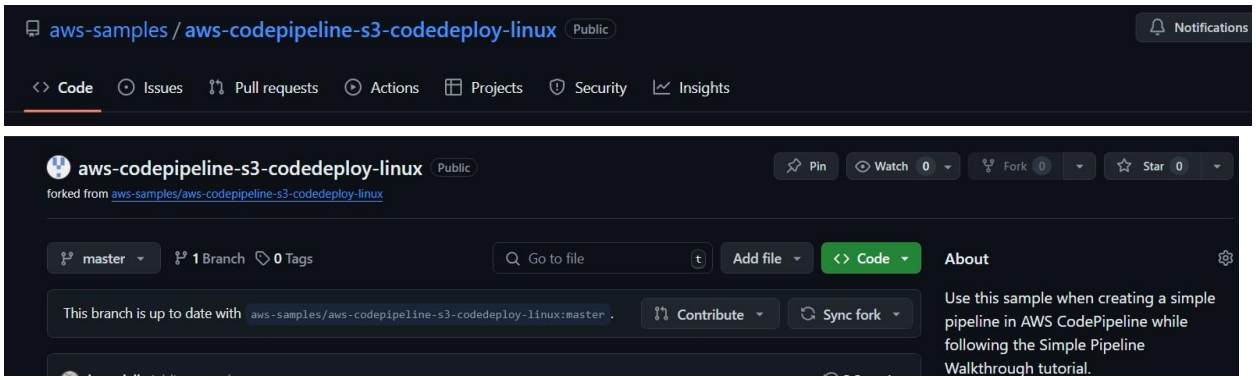
Key ▲	Value ▽

No environment properties

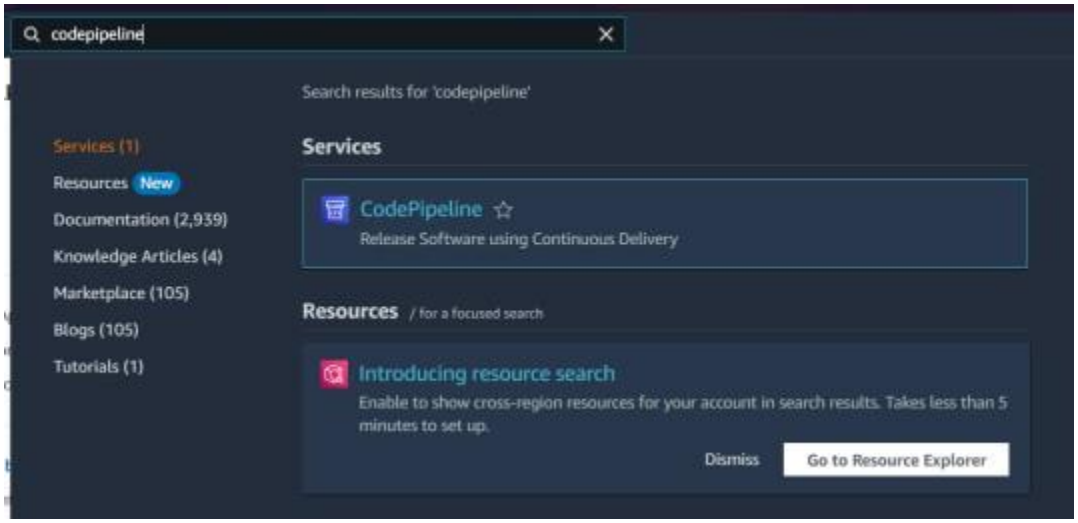
There are no environment properties defined

Cancel Previous Submit

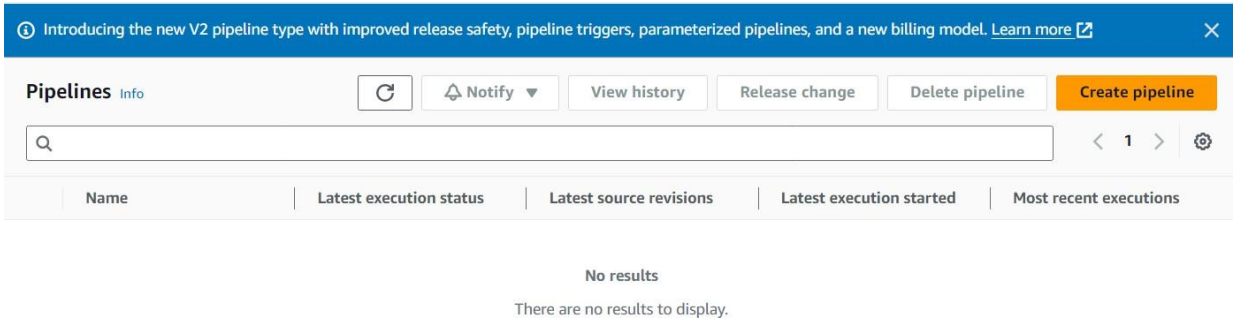
Step 6: Go to the github link below. This is a github with a sample code for deploying a file on AWS CodePipeline. Fork this repository into your personal github.
https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux



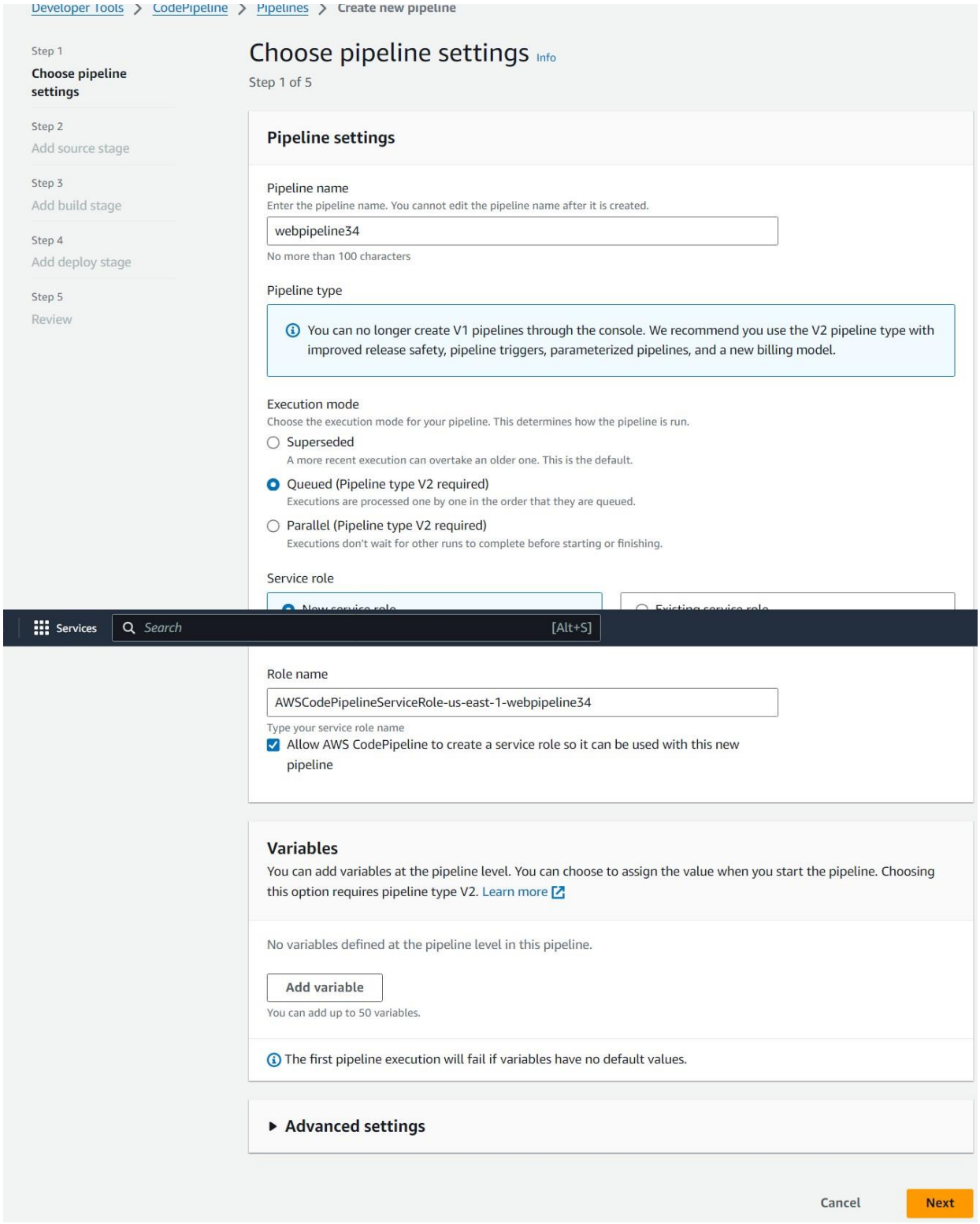
Step 7: Search CodePipeline in the services tab and click on it.



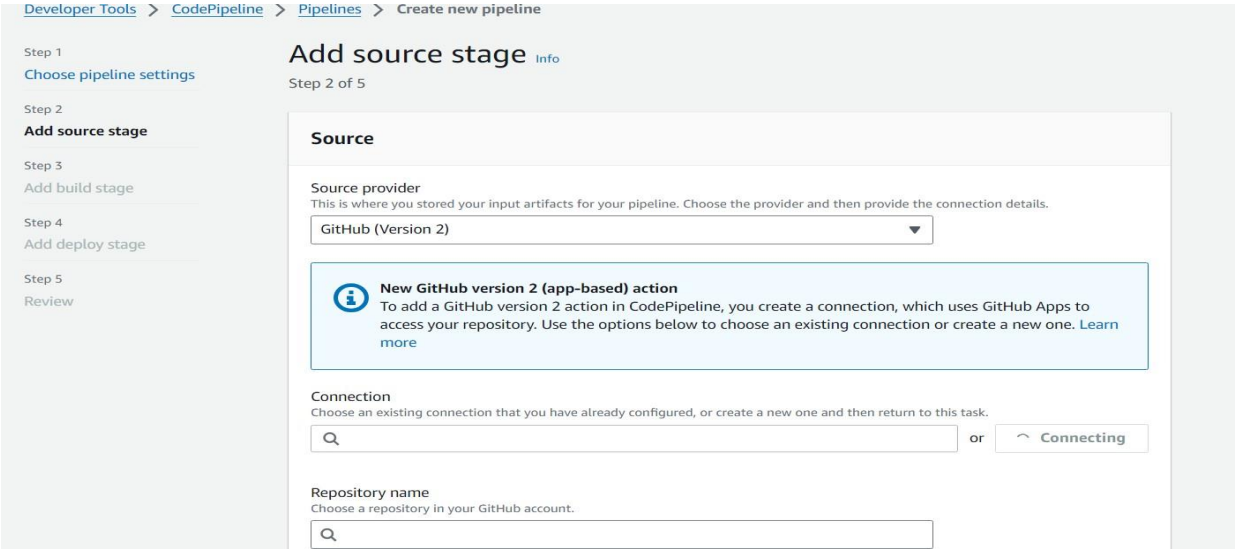
Step 8: Click on Create Pipeline.



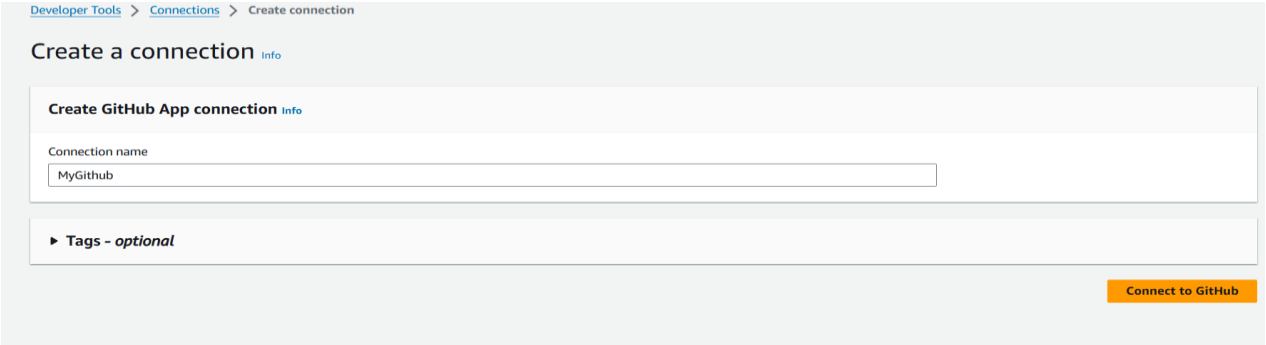
Step 9: Give a name to your Pipeline. A new service role would be created with the name of the pipeline

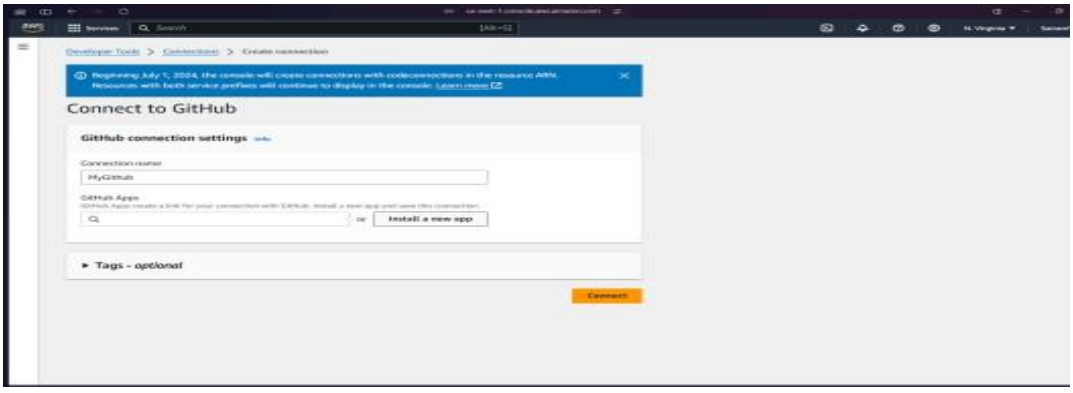


Step 10 : Select a source provider (as Github (Version 2)). Click on Connect to Github to connect your github.

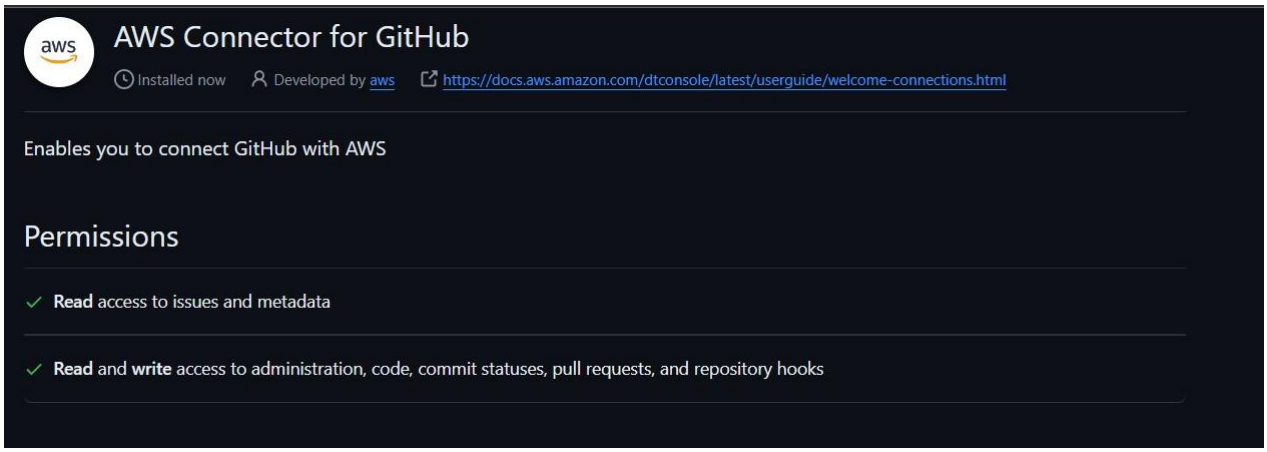


Step 11: Give a name to your GitHub app Connection and click on Connect. This will give you a prompt to either to select a GitHub app or to install a new app. If it is your first time, click on Install a new app

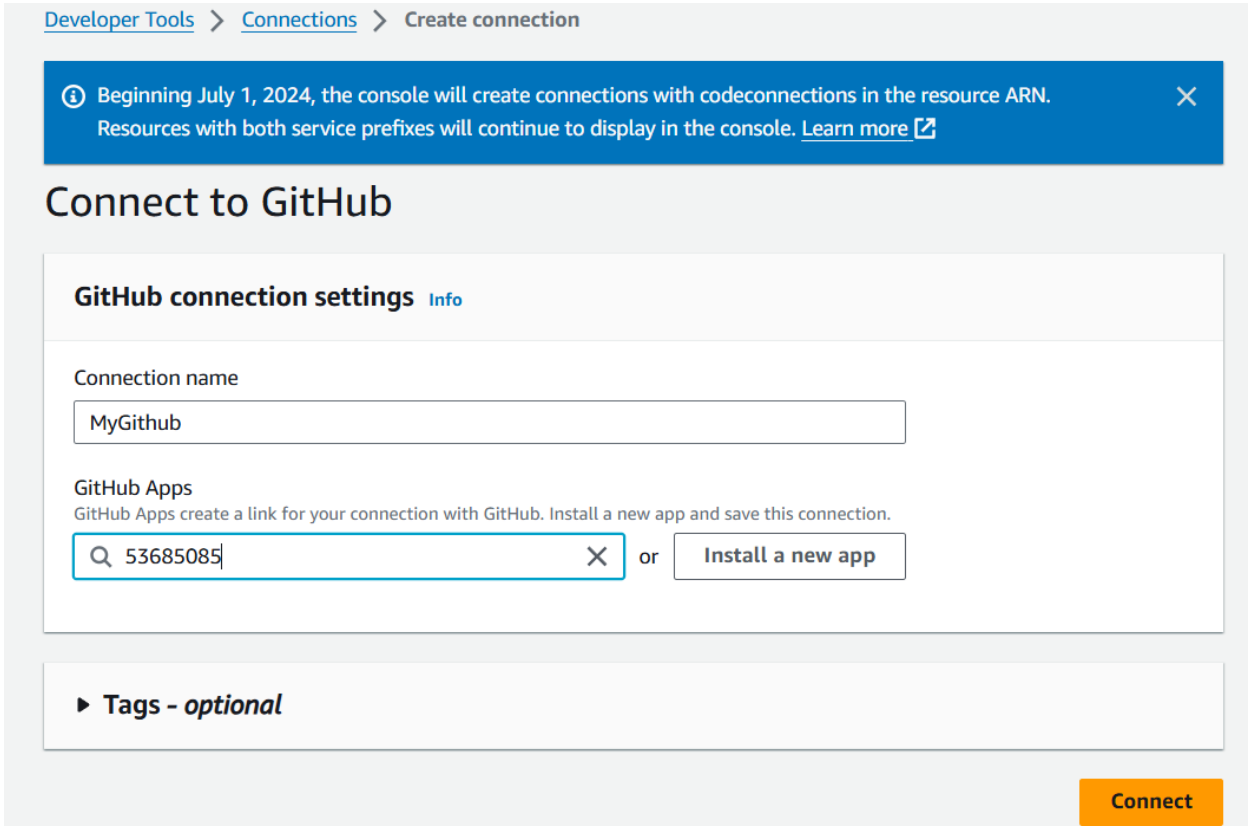


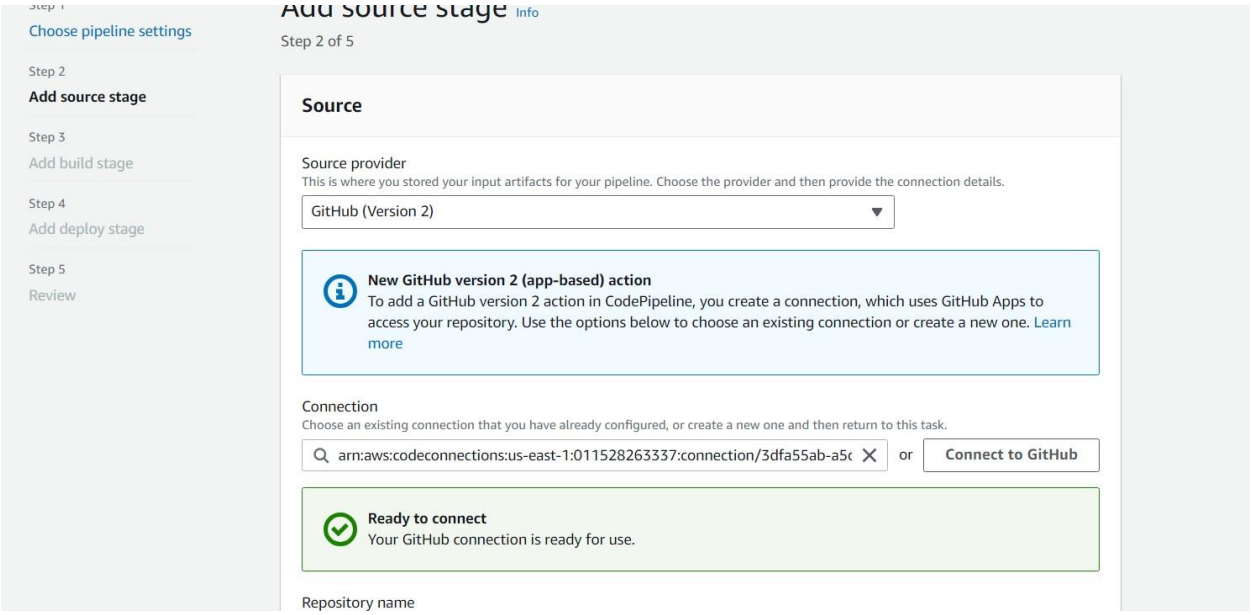


Step 12 : This will direct you to install AWS Connector On Your GitHub. Install it to your account and give it its permissions.

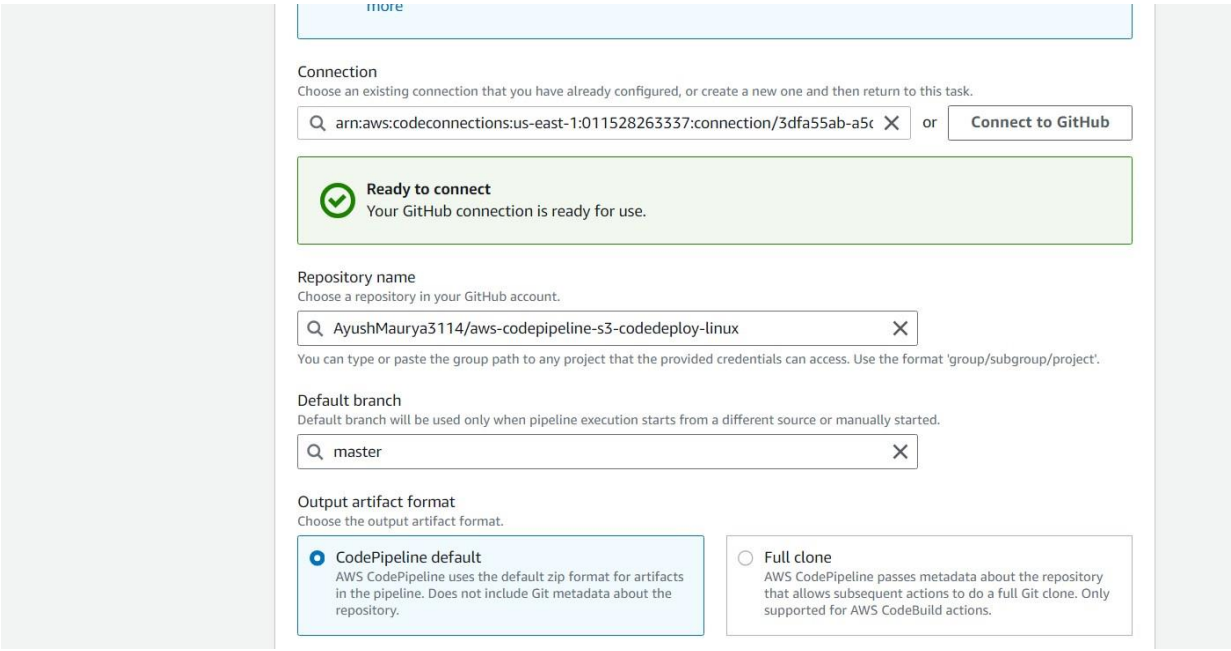


Step 13: After the app is set up, it gives the number in the text field. Click on Connect. After clicking on connect, the link is shown in the connection field and AWS shows that GitHub connection is ready to use

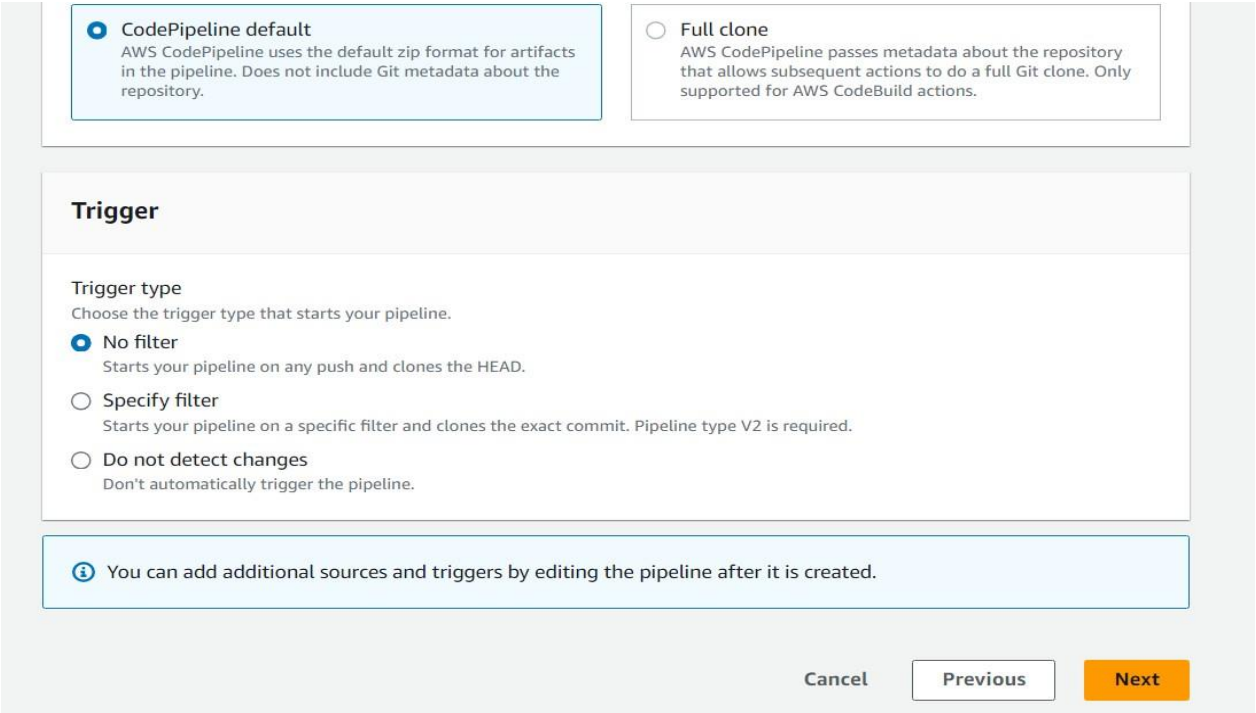




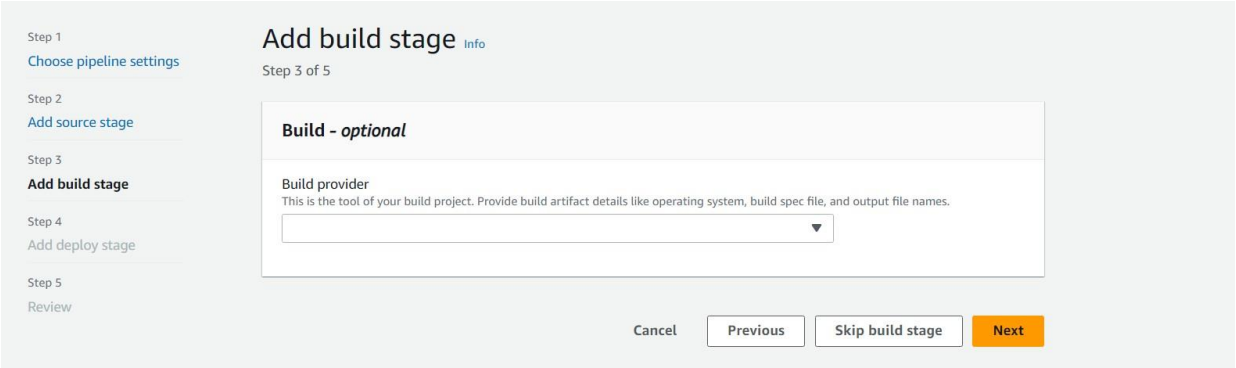
Step 14: Select the repository that you had forked to your GitHub. After that select the branch on which the files are present (default is Master).

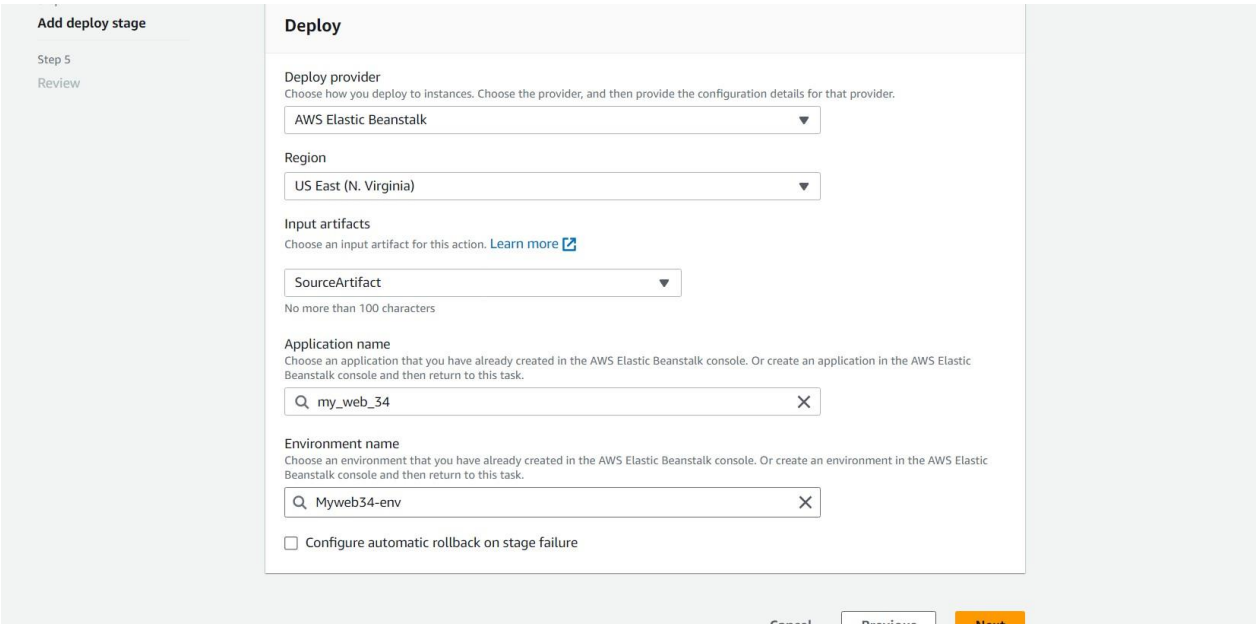


Step 15: Set the Trigger type as no filter. This would allow it to the website to update as soon as some change is made in the github.

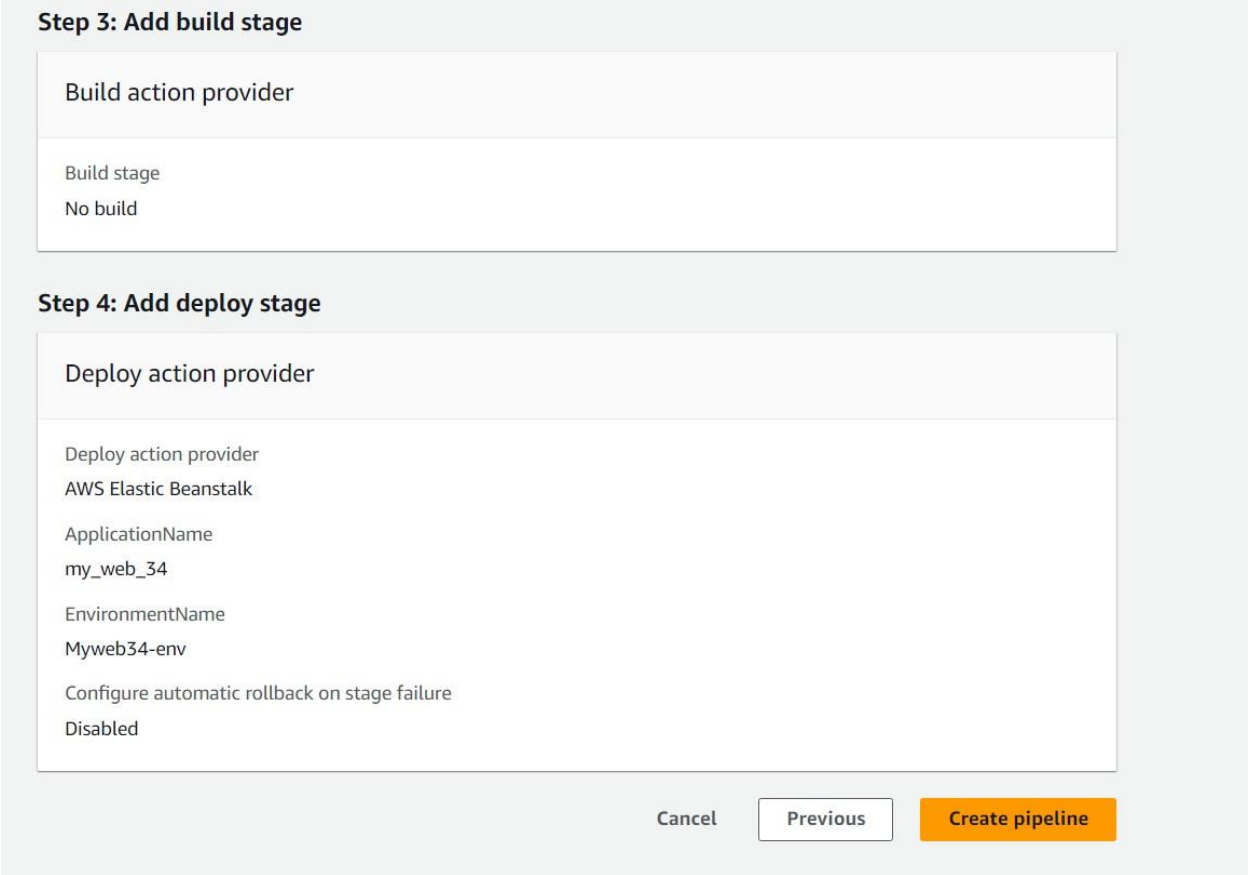


Step 16: Skip the build stage and go to Deploy. Select the deploy provider as AWS Elastic Beanstalk and Input Artifact as SourceArtifact. The application name would be the name of your Elastic Beanstalk. Then click on next.

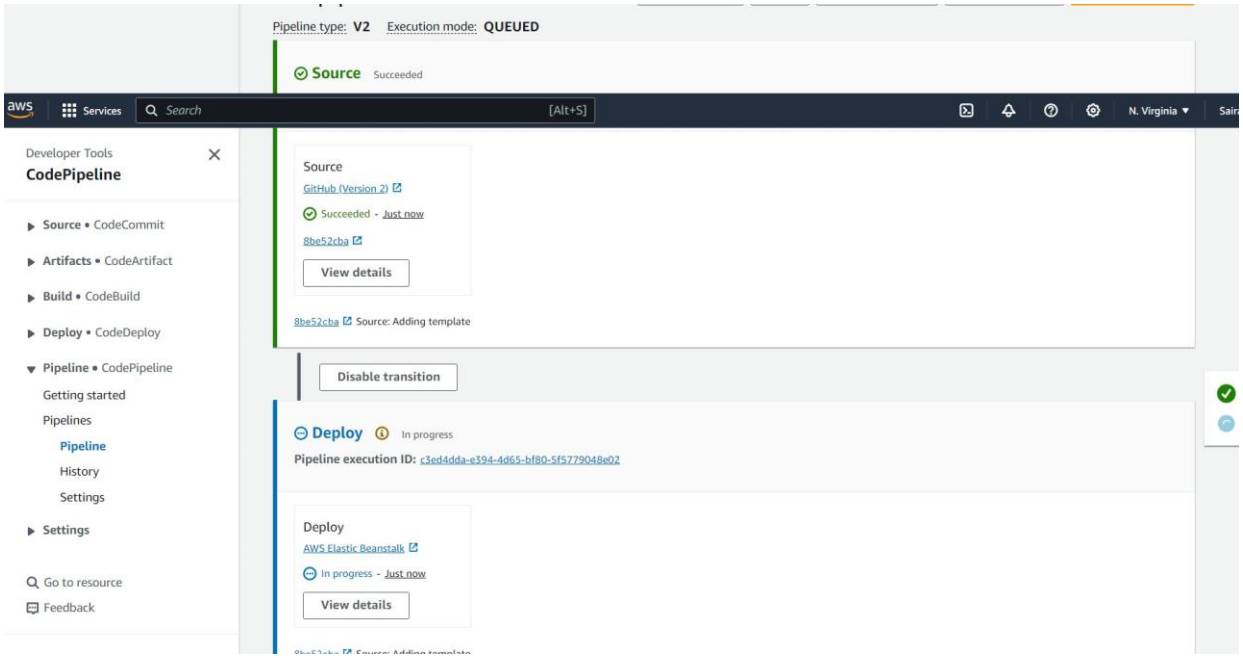




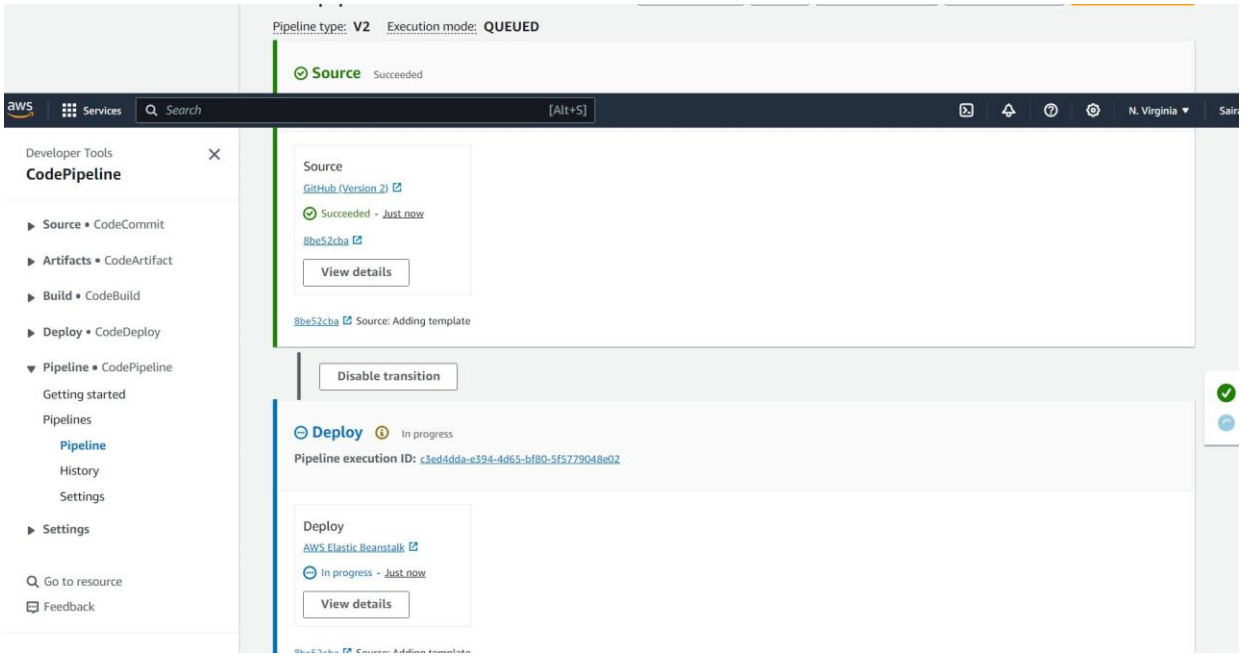
Step 17: Check all the information and click on create Pipeline.



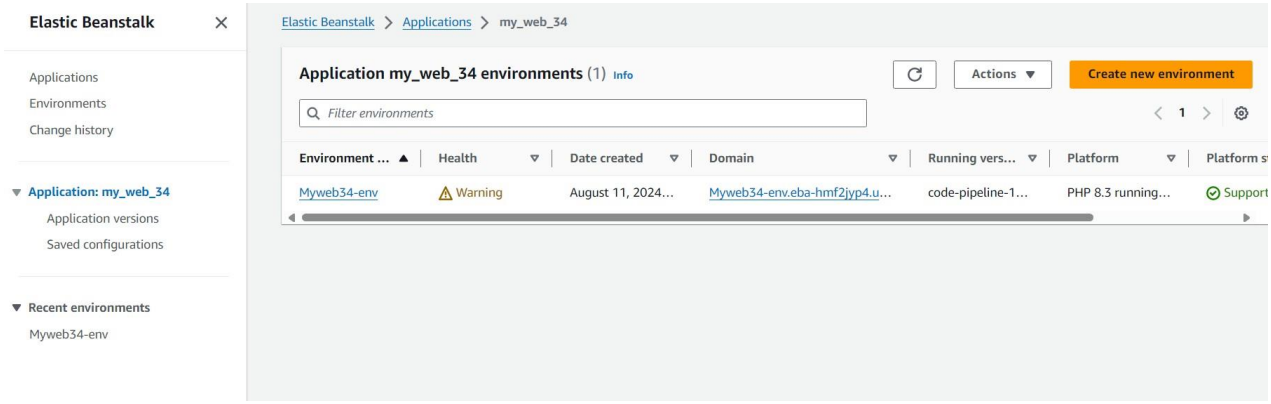
Step 18: If the pipeline is successfully deployed, this screen comes up where the source is set up and then it is transitioned to deploy.



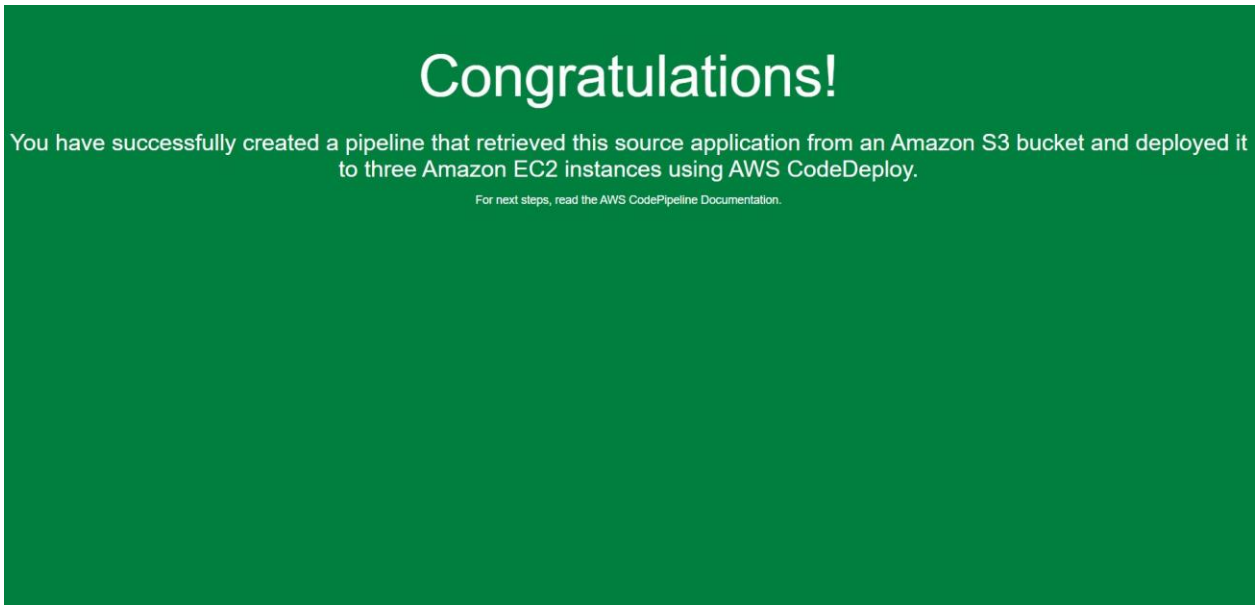
Step 19: Once the deployment is complete, click on the AWS Elastic Beanstalk under Deploy.



Step 20: This will redirect you to the application screen of Elastic Beanstalk. Click on the link shown under Domain.



Step 21: This will successfully show the sample website hosted.



Step 22: Now, we make some changes to the index.html file in the github.
For eg: If you make some changes to the <h2>tag.
Once the changes are committed, when the website is refreshed, the changes can be seen.