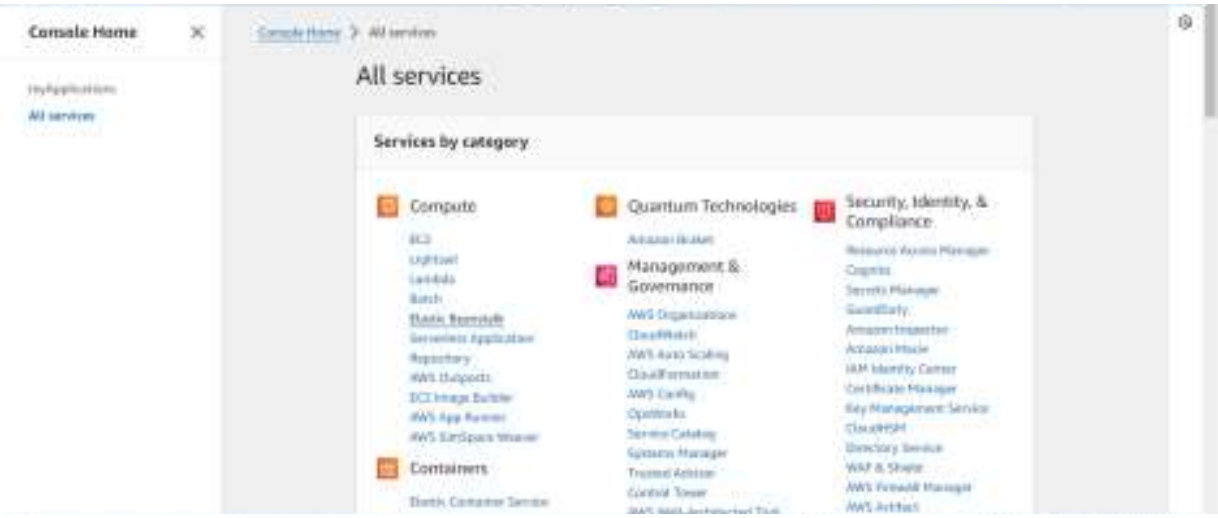
Name : Sandeshkumar.M.Yadav Div : D15C Roll : 61

Experiment No :2

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

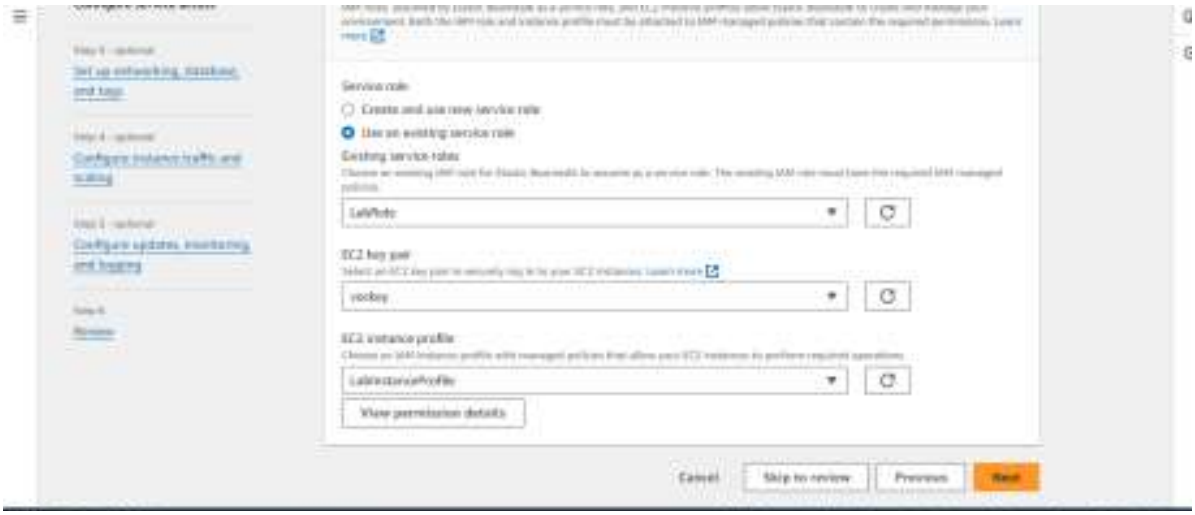


Step 2: Go to Elastic Beanstalk and click on Create Application

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.

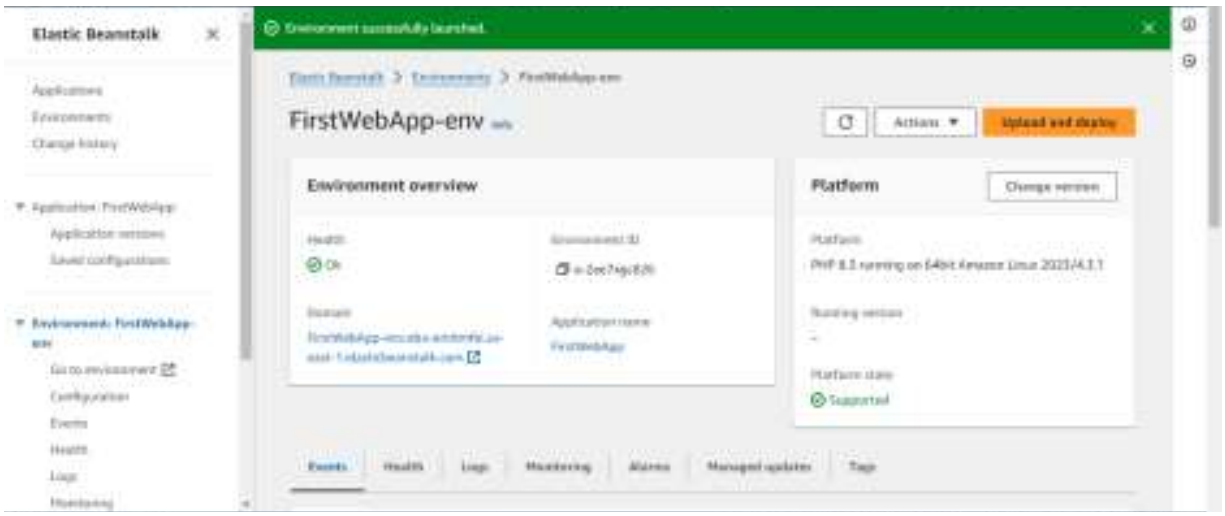
Use an existing service role and choose whatever service role is present on your account

Step 5: Click on Skip to Review



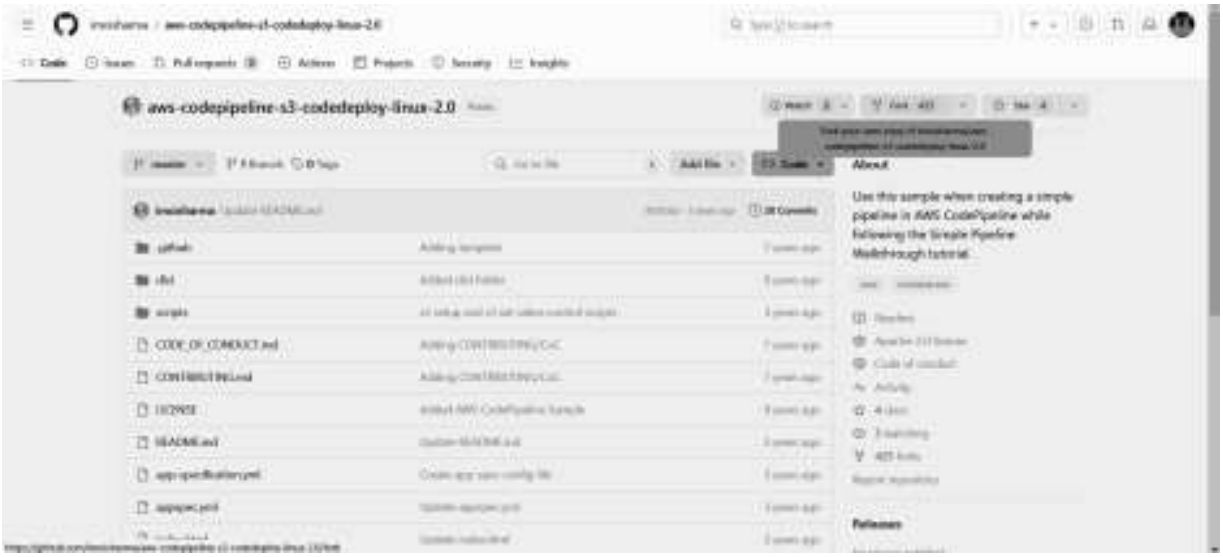
Name : Sandeshkumar.M.Yadav Div : D15C Roll : 61

Review the settings that you have set up for your application and submit your application

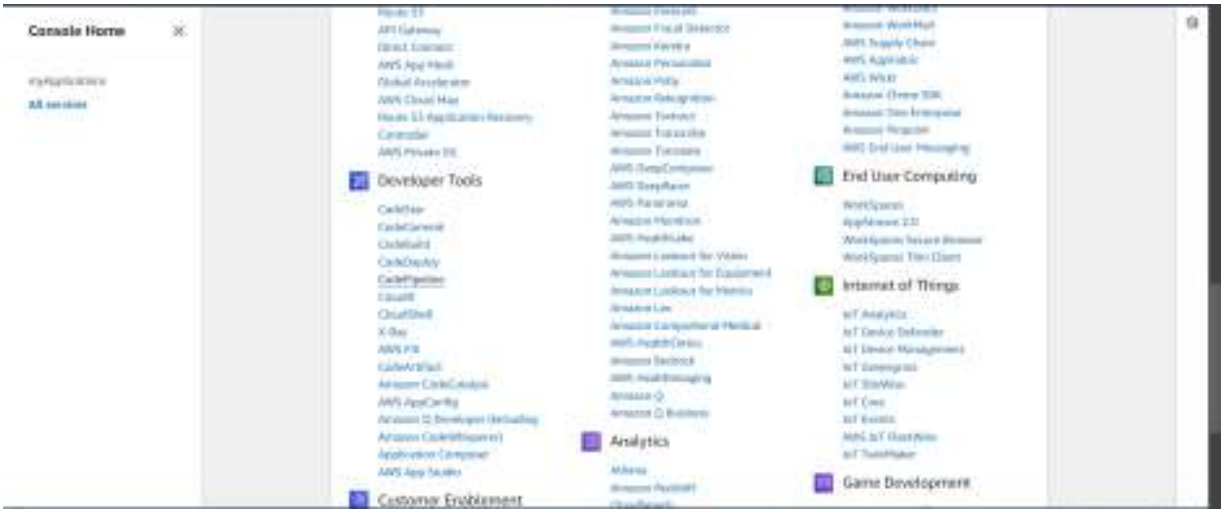


Step 7 : Go to the github link below. This is a github with a sample code for deploying a file on AWS CodePipeline. Fork this repository into your personal github.

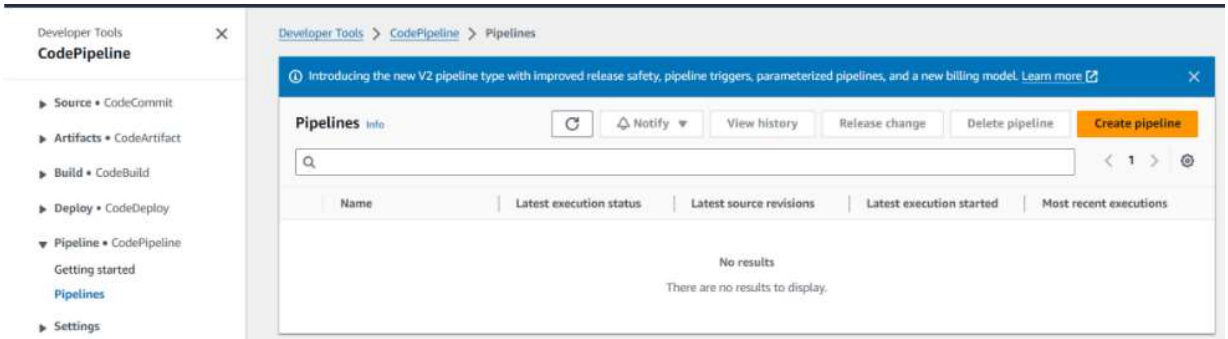
https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux



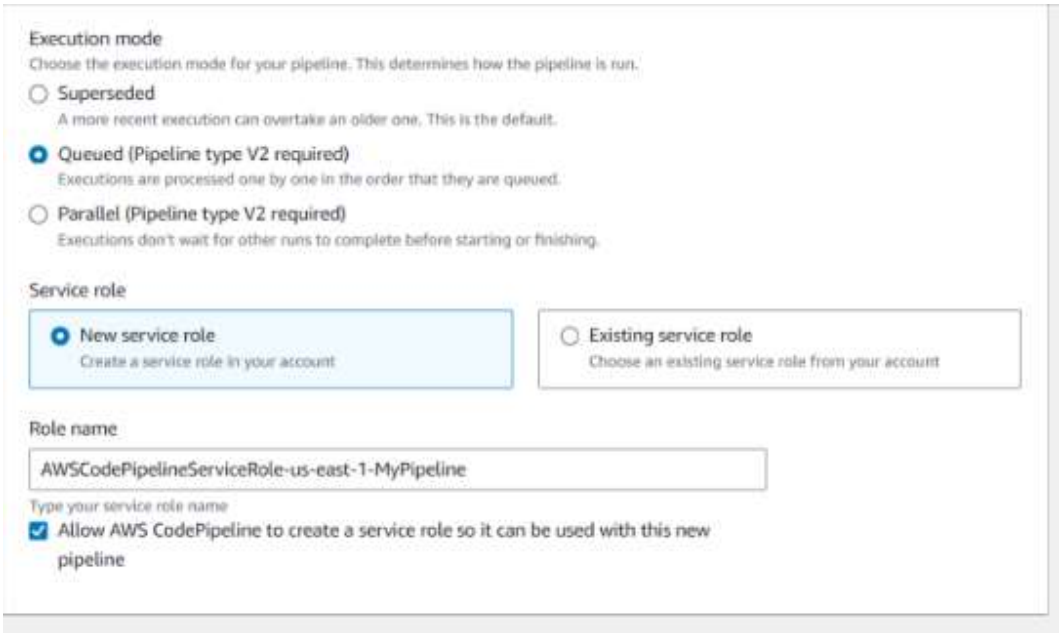
Step 8: Search CodePipeline in the services tab and click on it.



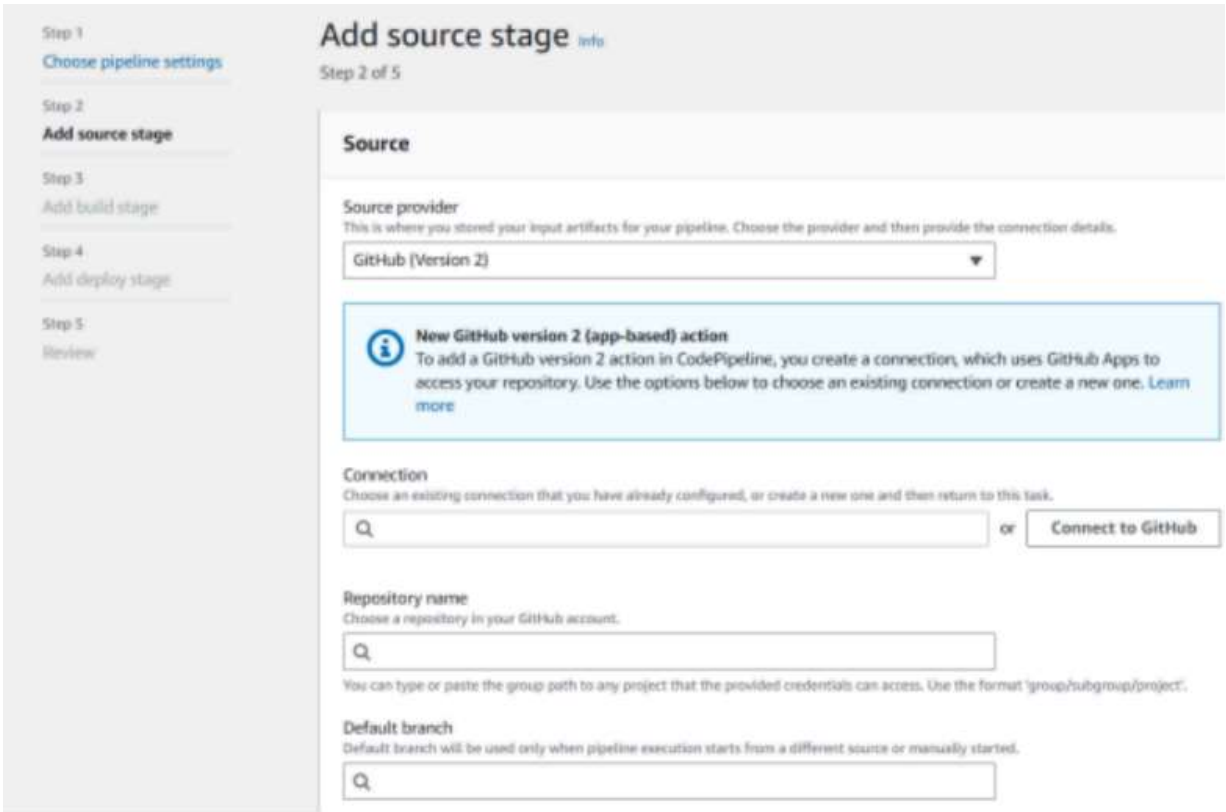
Step 9: Click on Create Pipeline.



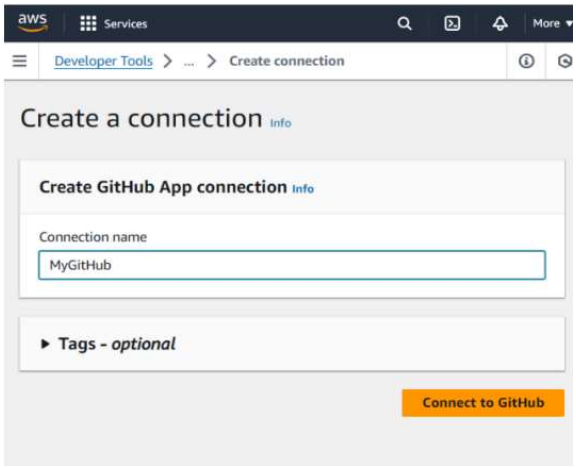
Step 10: Give a name to your Pipeline. A new service role would be created with the name of the pipeline.

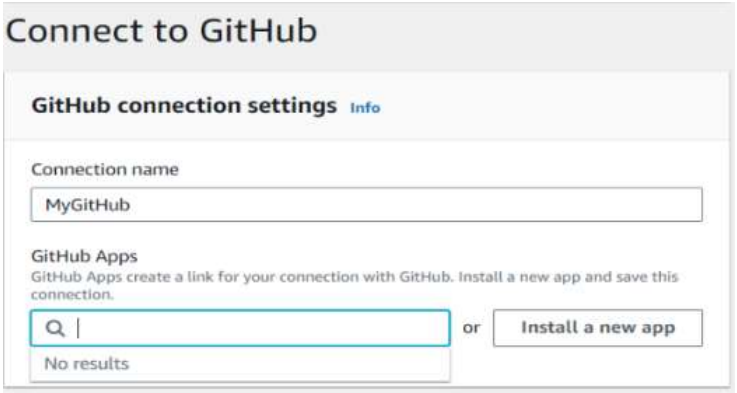


Step 11: Select a source provider (as GitHub Version (2)). Click on connect to Github

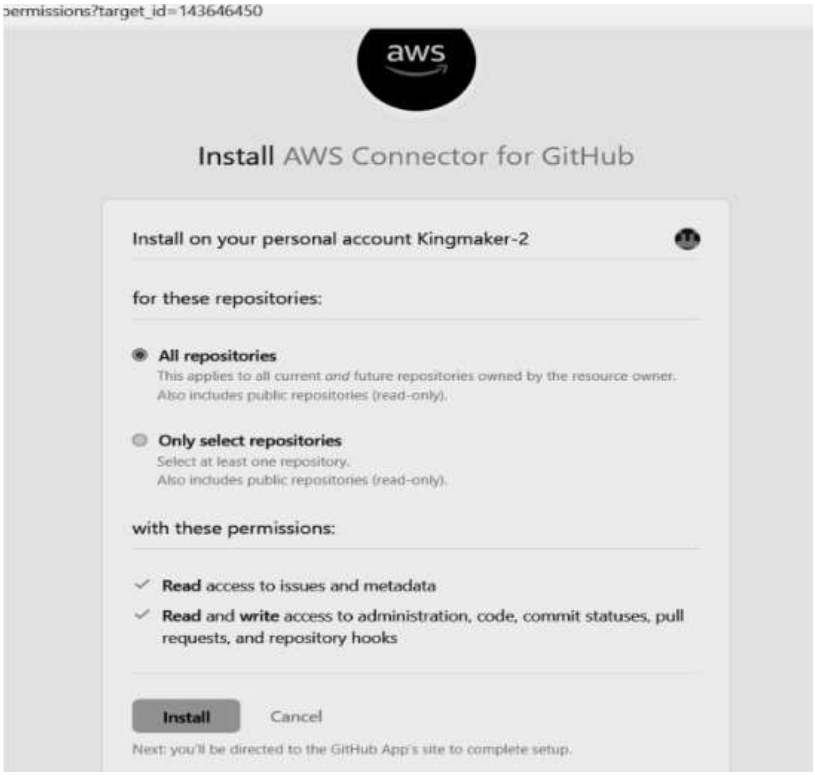


Step 12: Give a name to your GitHub app Connection and click on Connect. This will give you a prompt to either to select a GitHub app or to install a new app. If it is your first time, click on Install a new app.

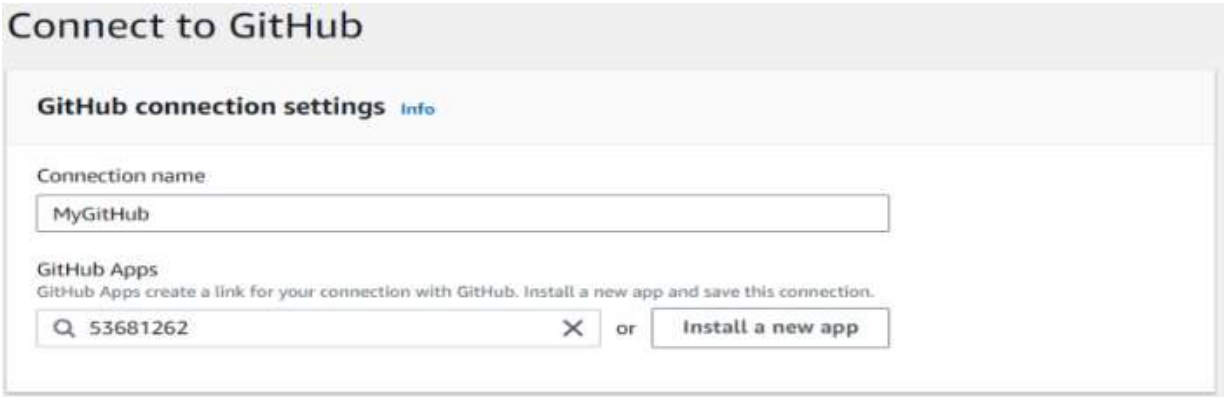


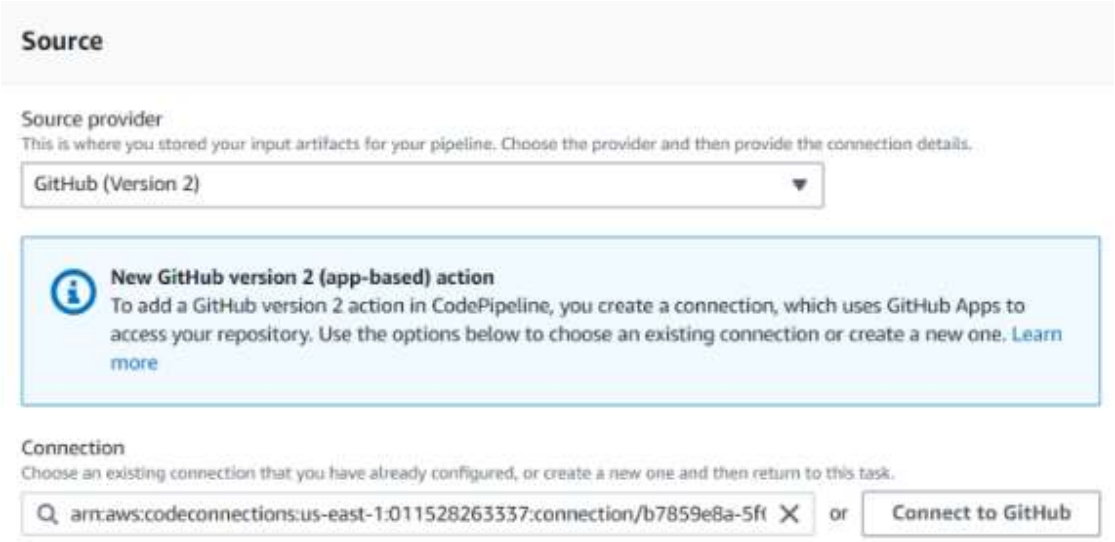


Step 13: This will direct you to install AWS connector on your GitHUb .Install it to your account and give it its permissions

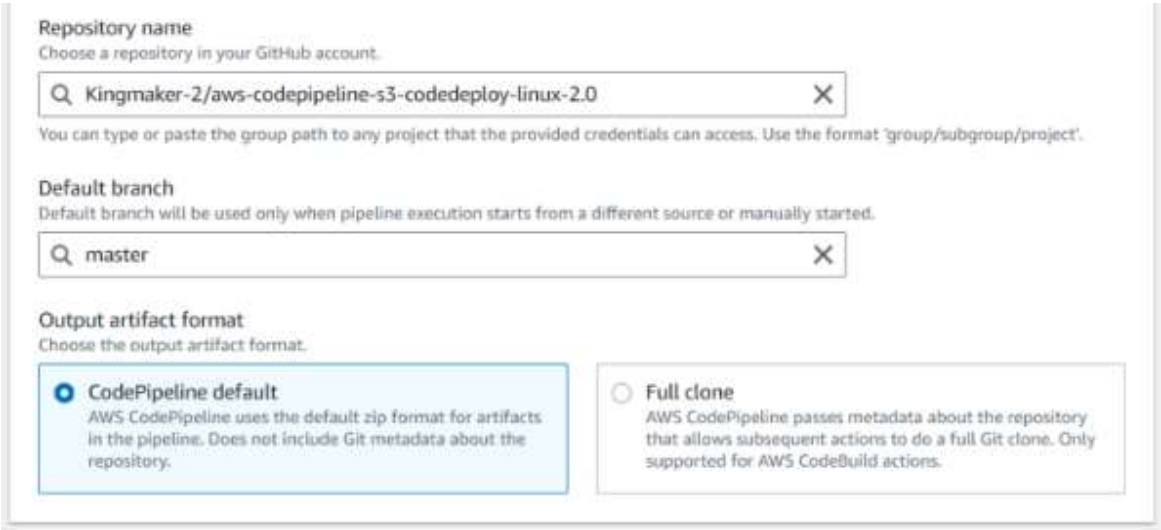


Step 14: After the app is set up, it gives the number in the text field. Click on Connect. After clicking on connect, the link is shown in the connection field and AWS shows that GitHub connection is ready to use.



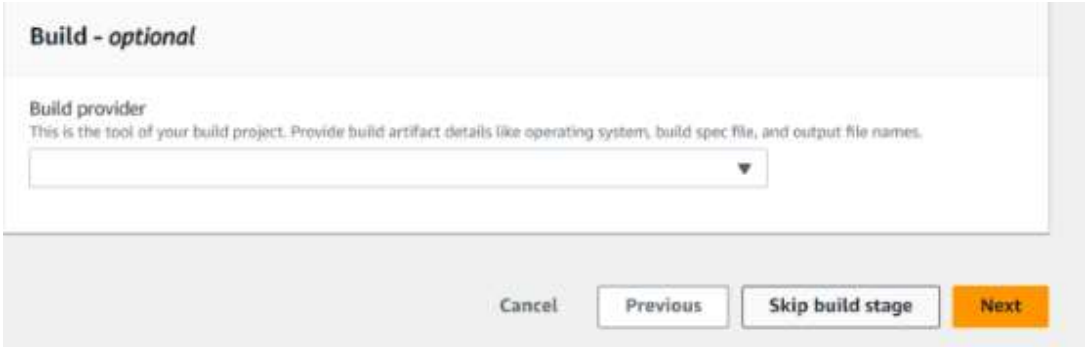


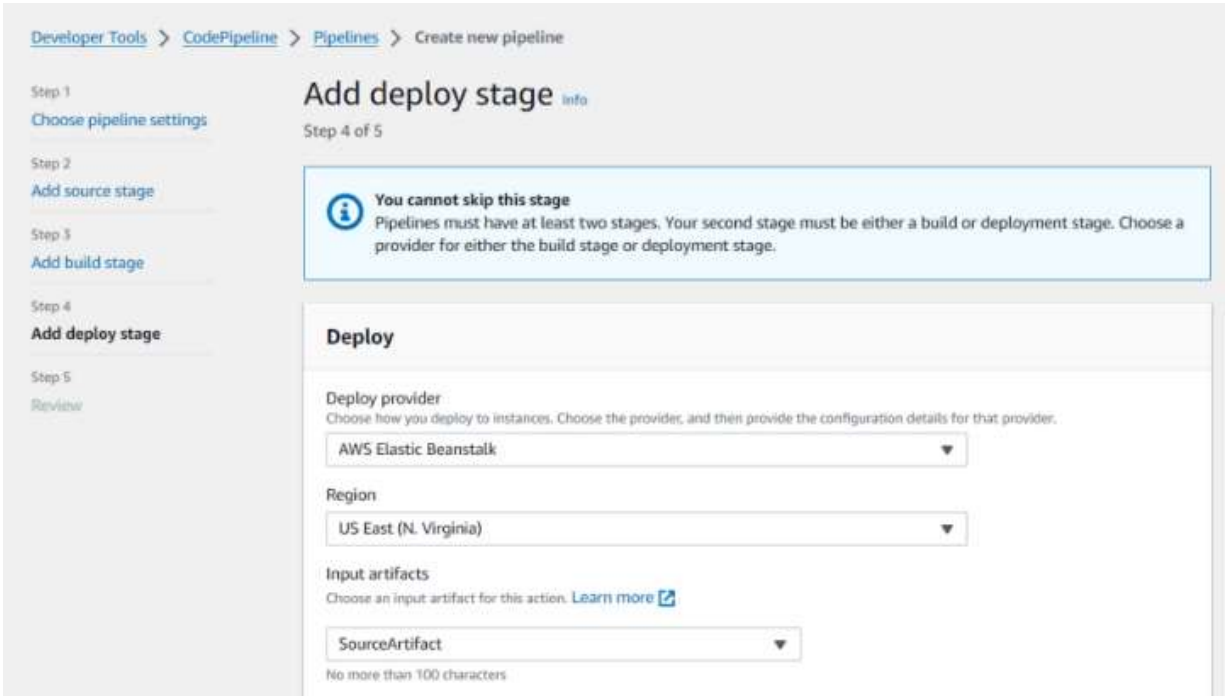
Step 15: Select the repository that you had forked to your GitHub. After that select the branch on which the files are present (default is Master).

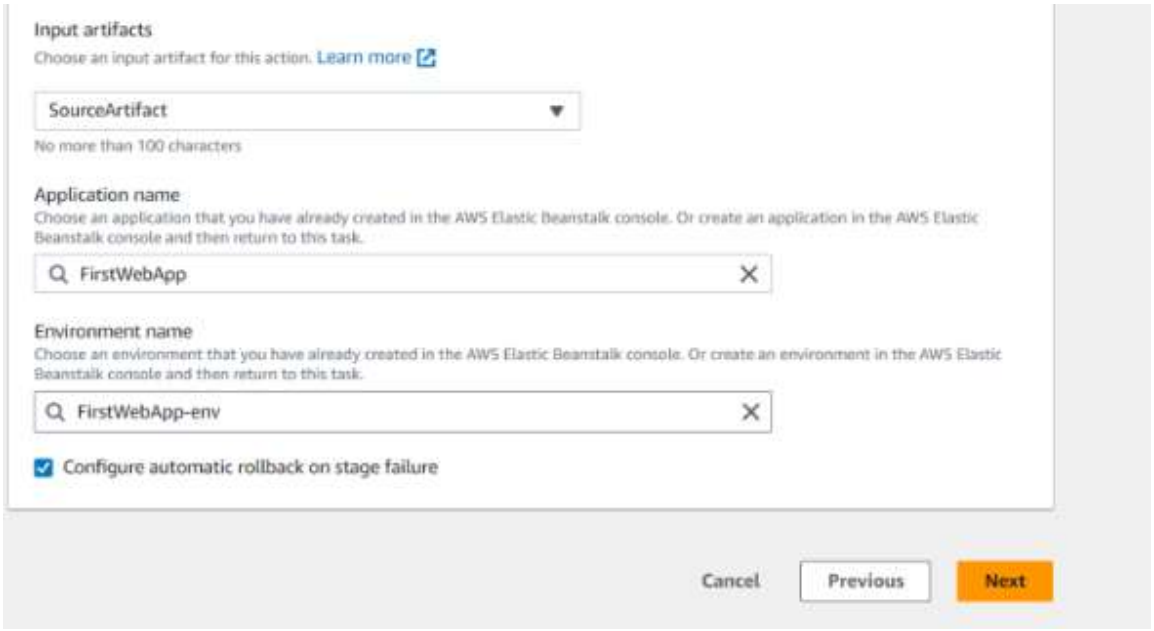


Step 16: Set the Trigger type as no filter. This would allow it to the website to update as soon as some change is made in the github.

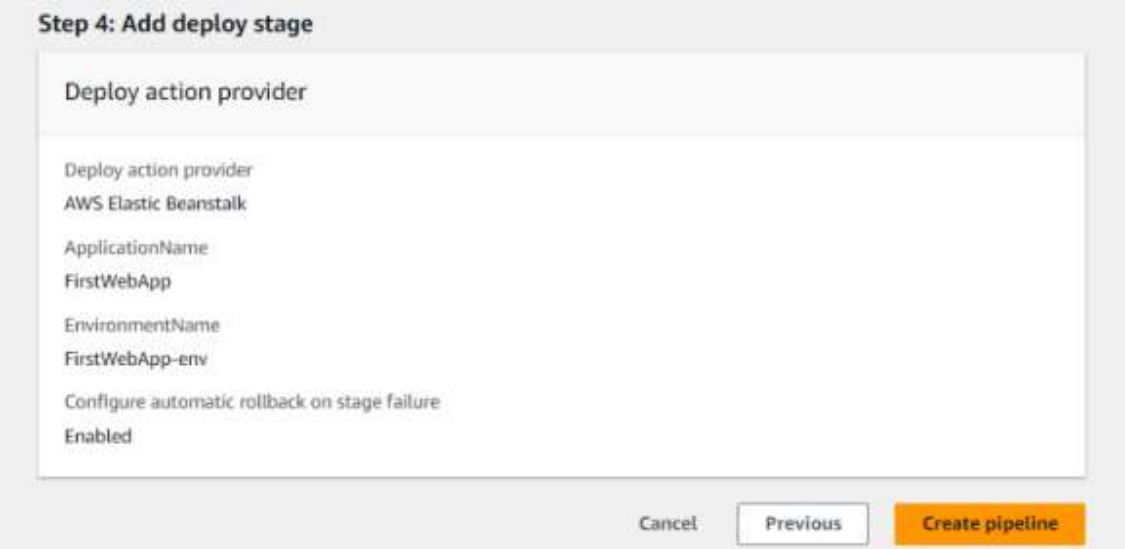
Step 17: Skip the build stage and go to Deploy. Select the deploy provider as AWS Elastic Beanstalk and Input Artifact as SourceArtifact. The application name would be the name of your Elastic Beanstalk. Then click on next.



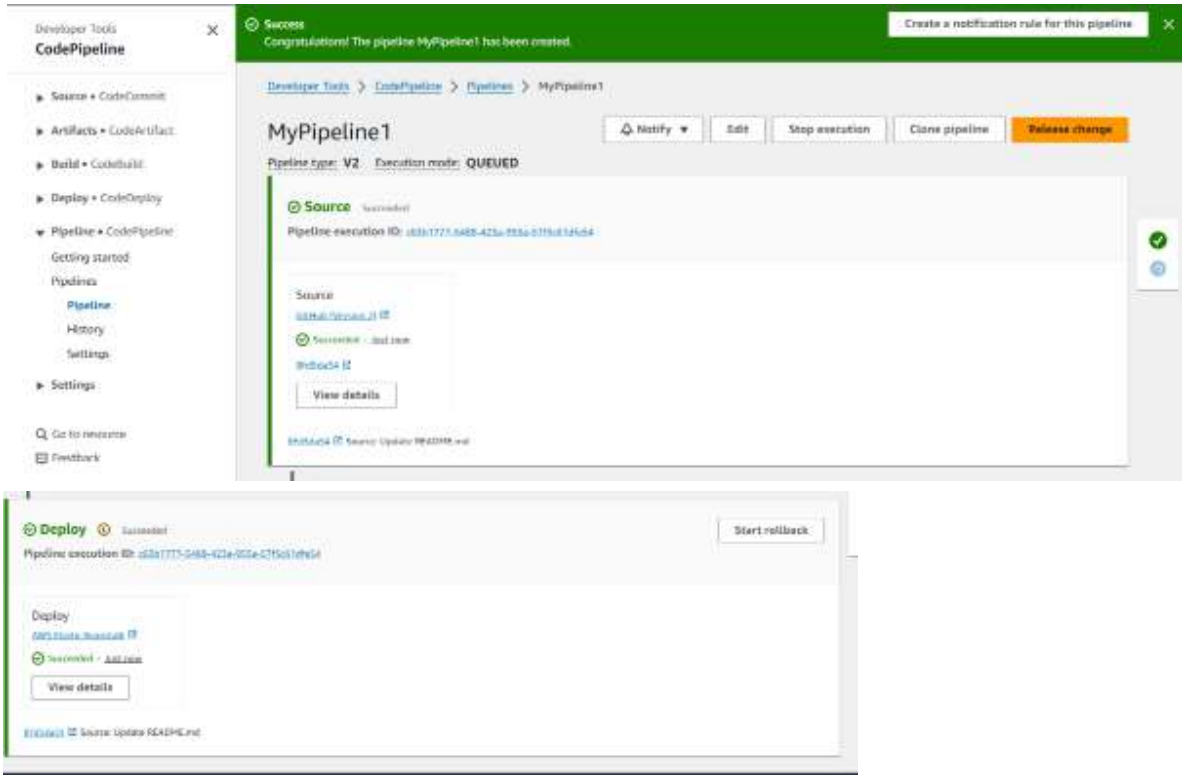

Developer Tools > CodePipeline > Pipelines > Create new pipeline
Step 1
Choose pipeline settings
Step 2
Add source stage
Step 3
Add build stage
Step 4
Add deploy stage
Step 5
Review
Add deploy stage info
Step 4 of 5
You cannot skip this stage
Pipelines must have at least two stages. Your second stage 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


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

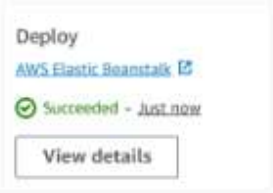
Step 18: Check all the information and click on create Pipeline



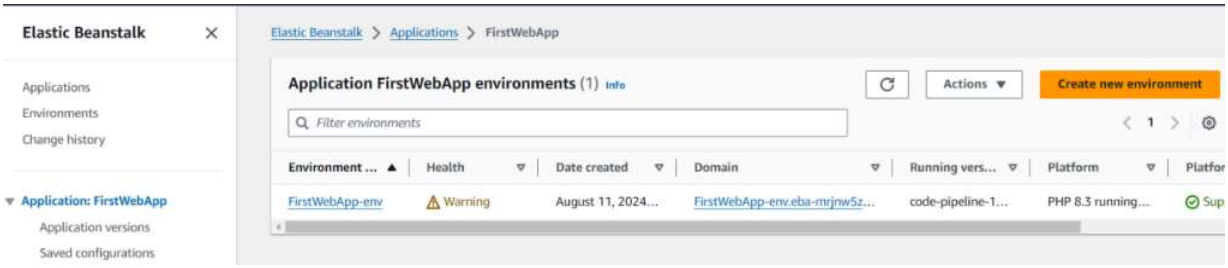
Step 19: If the pipeline is successfully deployed, this screen comes up where the source is set up and then it is transitioned to deploy



Step 20: Once the deployment is complete, click on the AWS Elastic Beanstalk under Deploy.



Step 21: This will redirect you to the application screen of Elastic Beanstalk. Click on the link shown under Domain



Step 22: This will successfully show the sample website hosted.