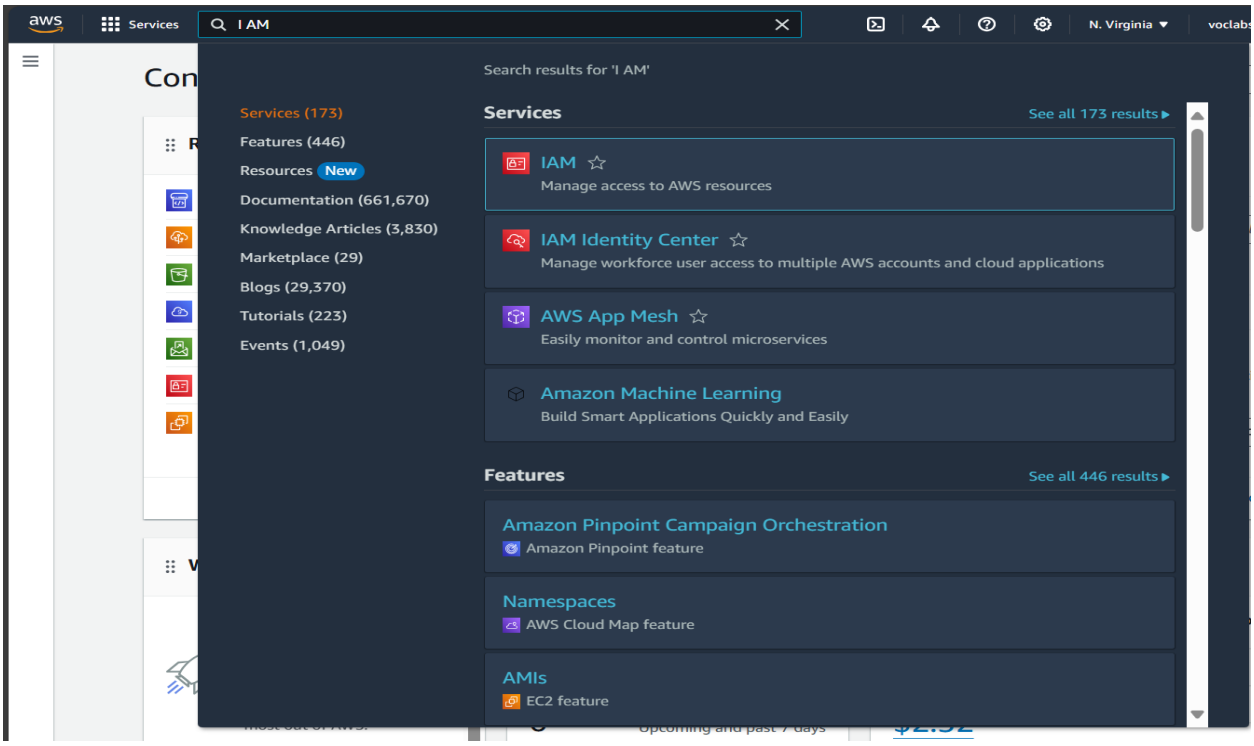
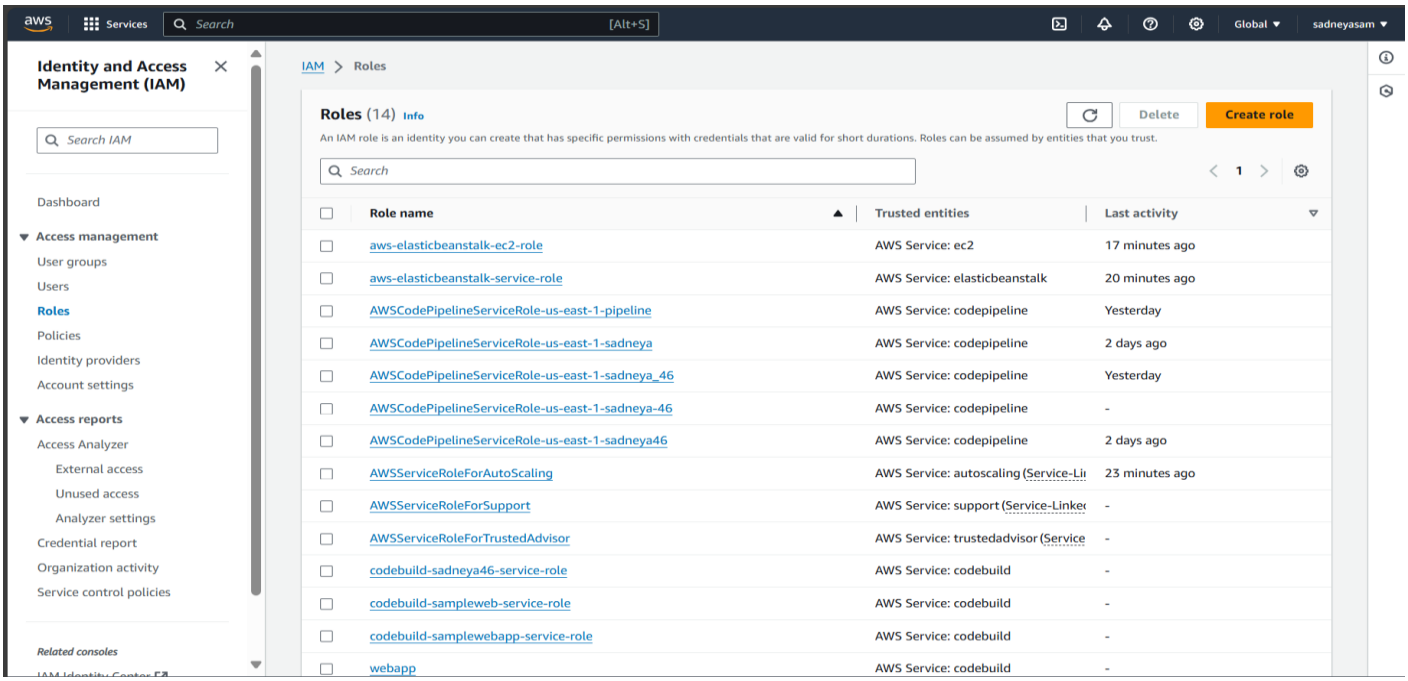
Experiment No: 2

Step1:- Creation of role:-

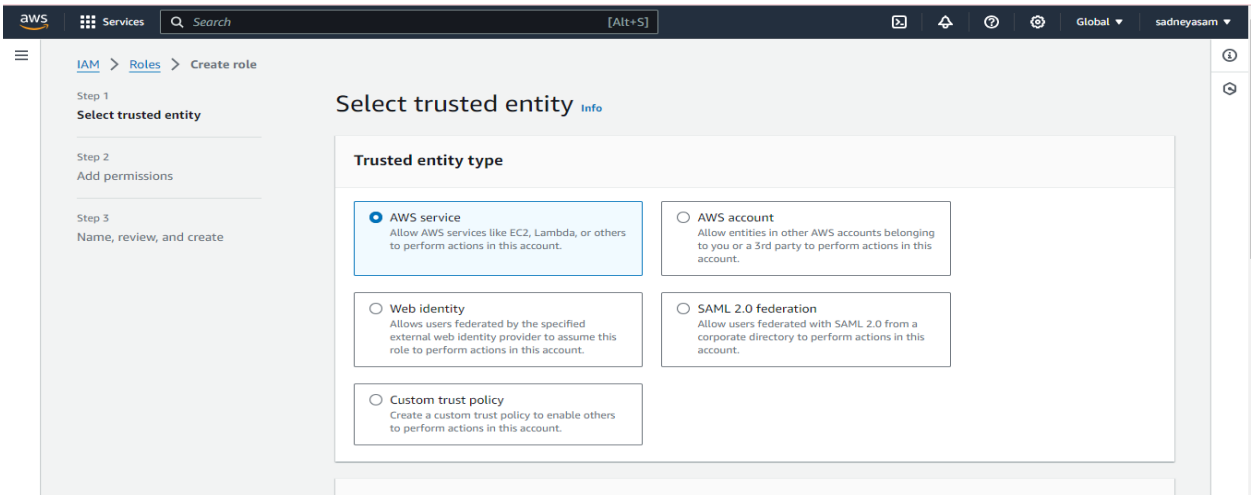
1. Login to your AWS account and search for IAM



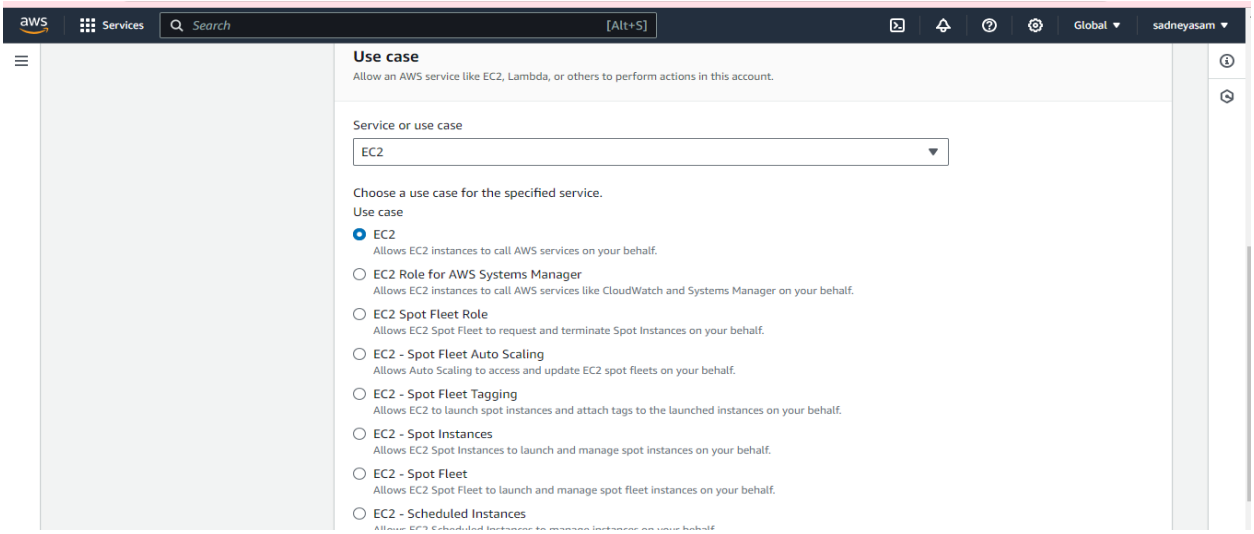
2. Then go into the role section and click on create role.



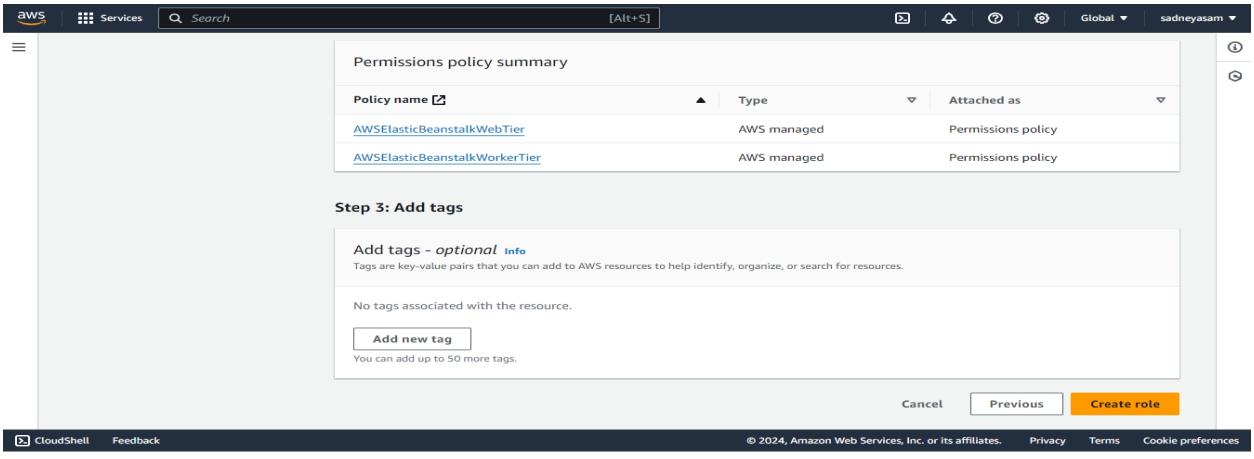
3. Then select a trusted entity as AWS service.



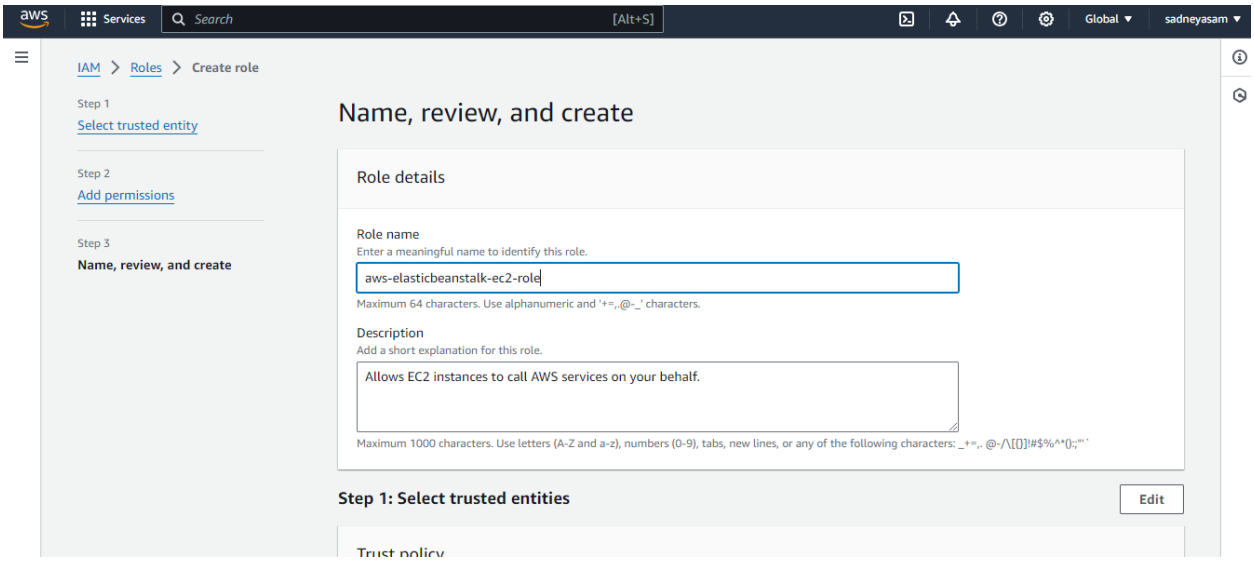
4. Select use case as EC2.



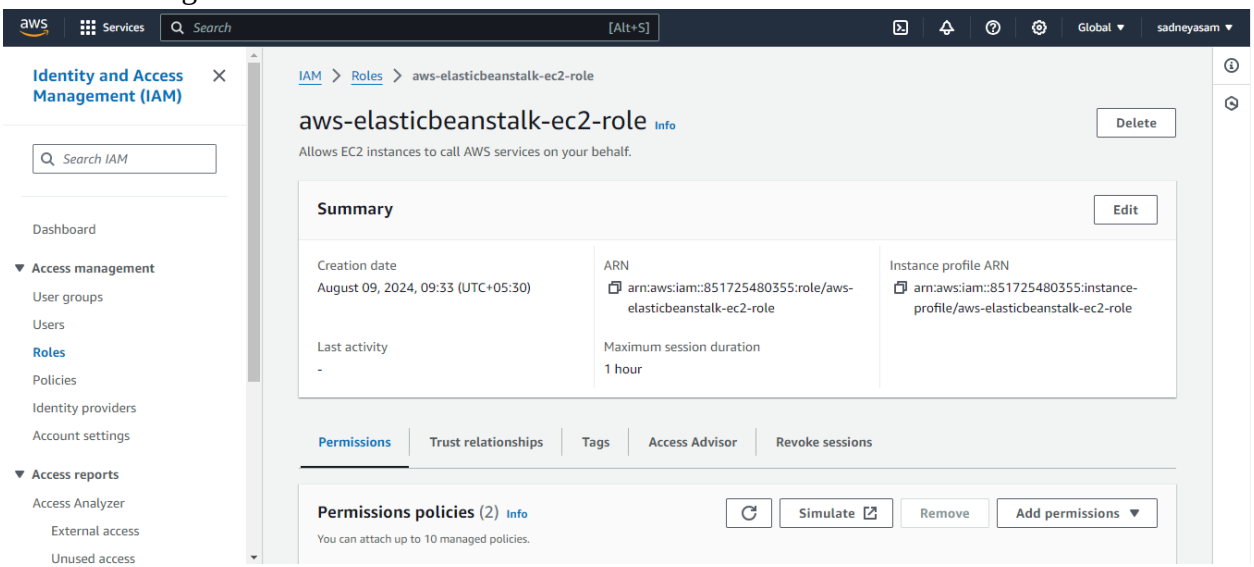
5. Select permissions as AWS Elastic Benstalk Web Tier.and AWS elastic Benstalk worker tier.



6. Give a name to Role. Here I have given my role name as aws -elasticbenstalk -ec2 -role.

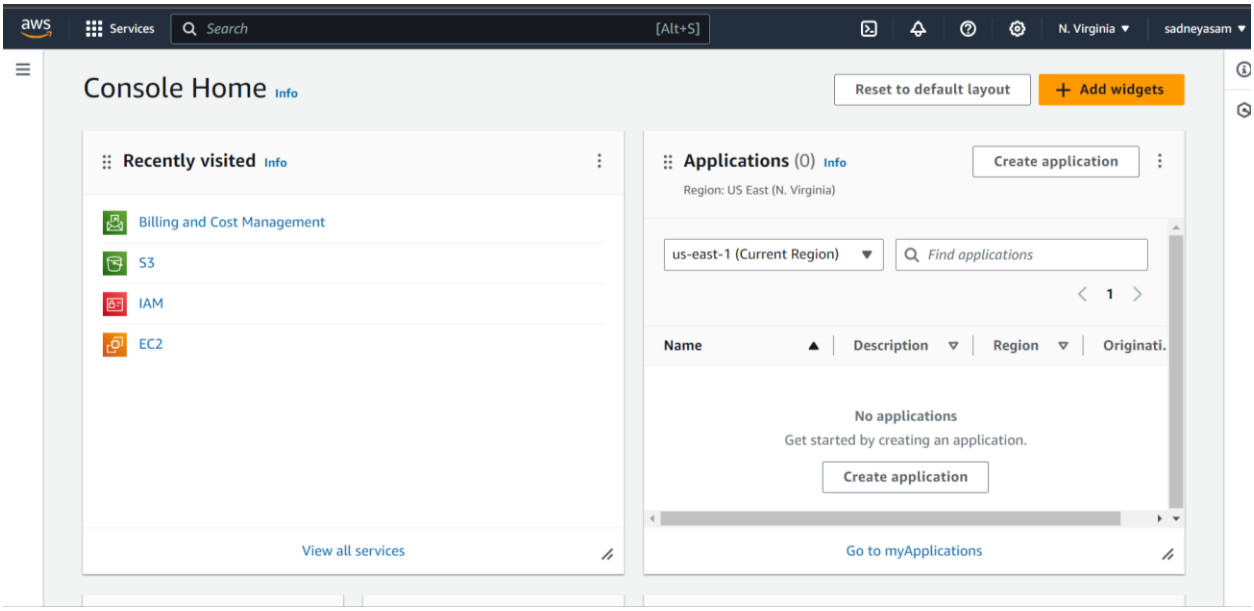


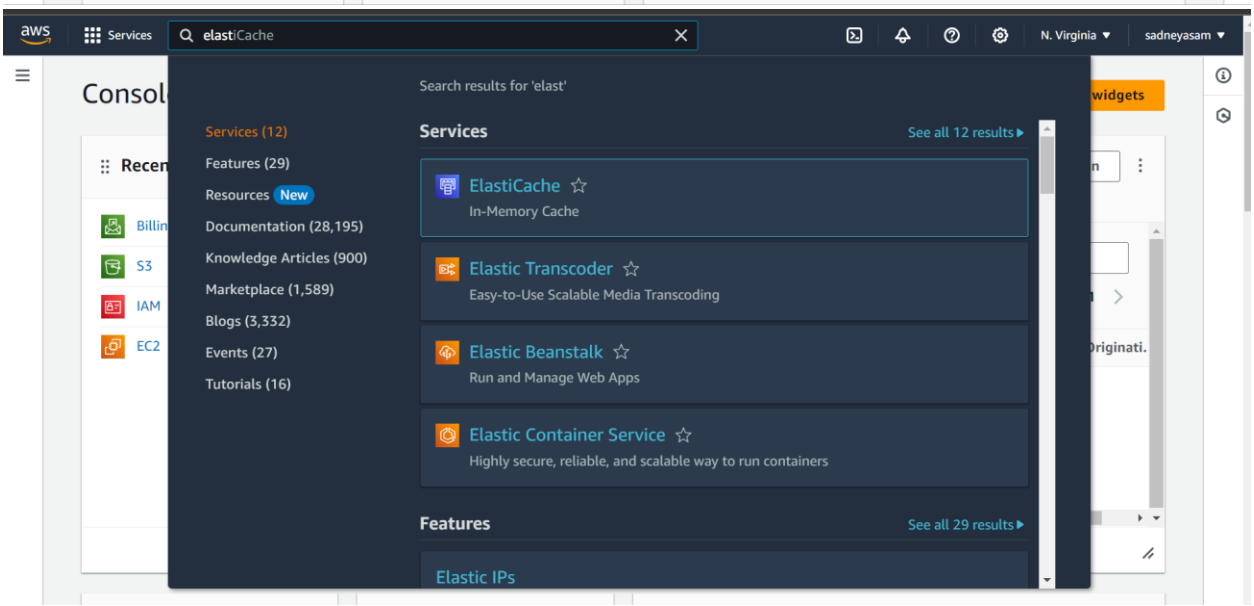
7. Then the role gets created.



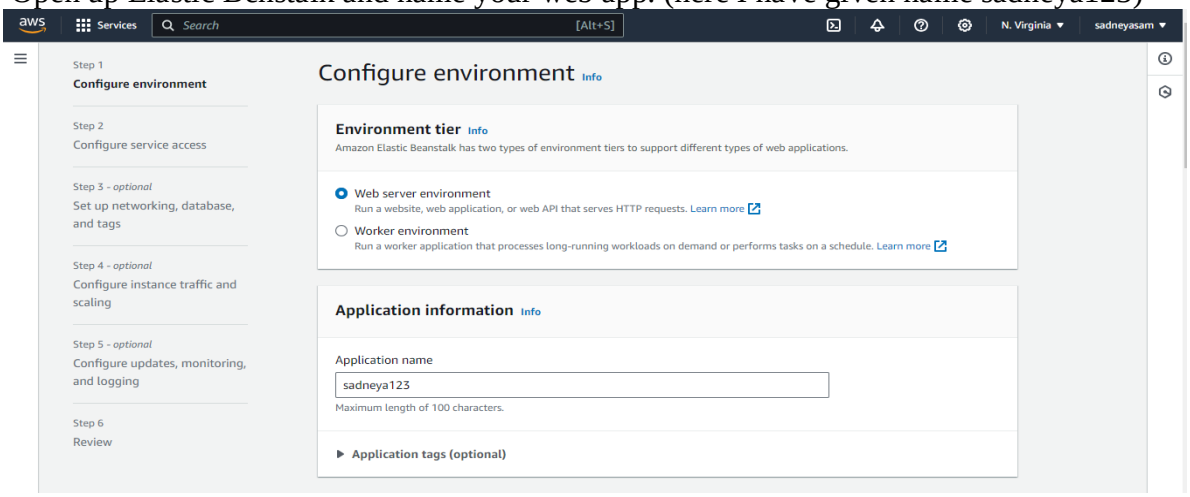
Step2:- Creation Elastic Benstalk Environment

1. search for Elastic Beanstalk in the search box.

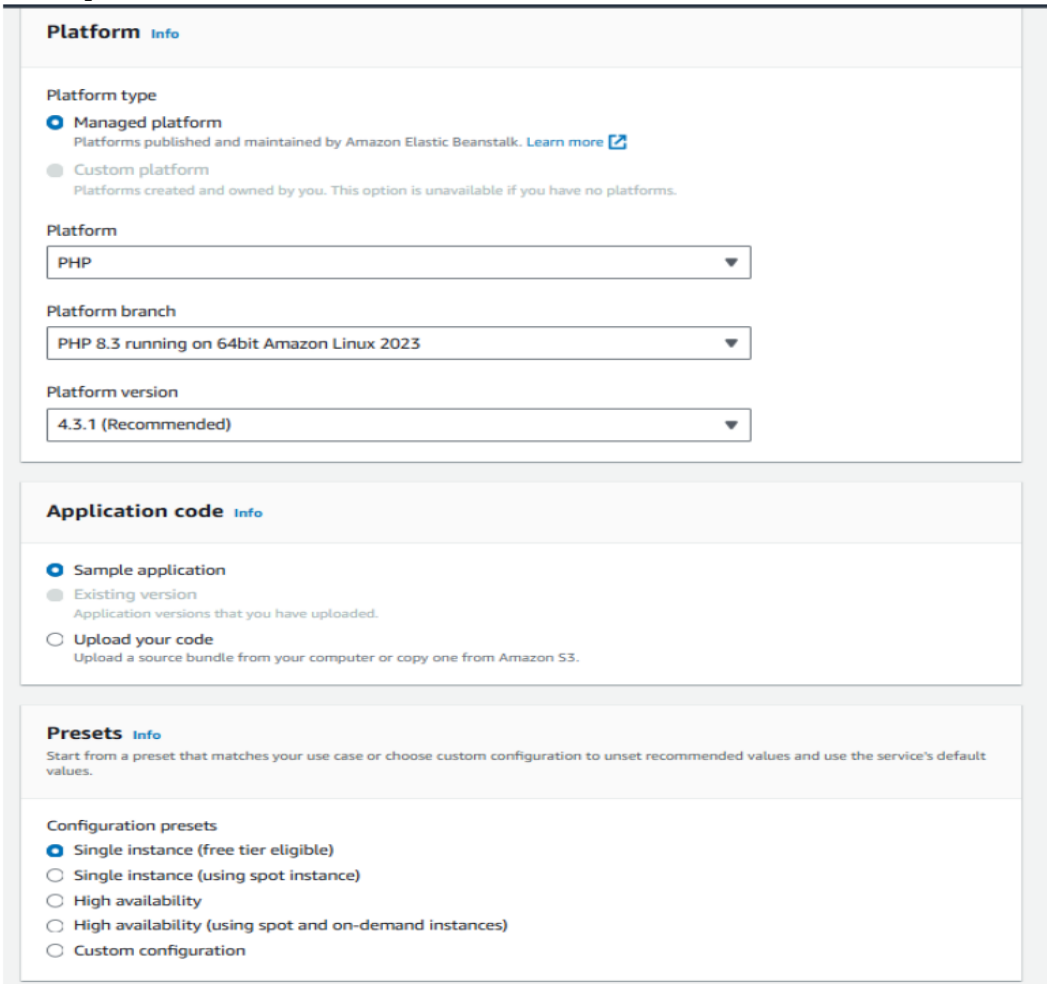




2. Open up Elastic Benstalk and name your web app. (here I have given name sadneya123)

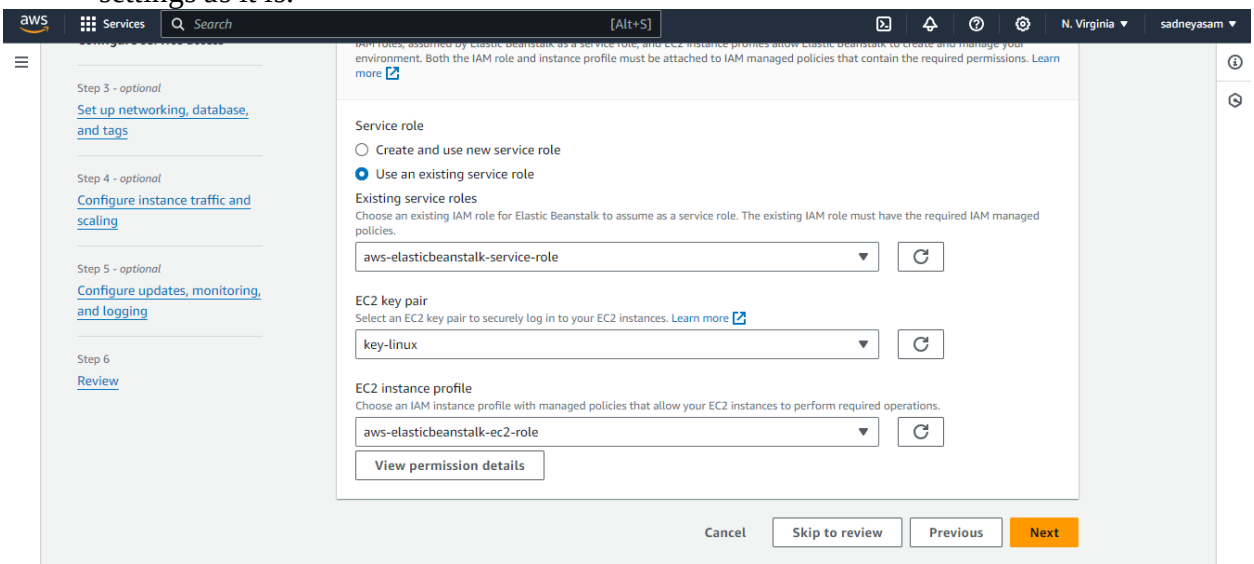


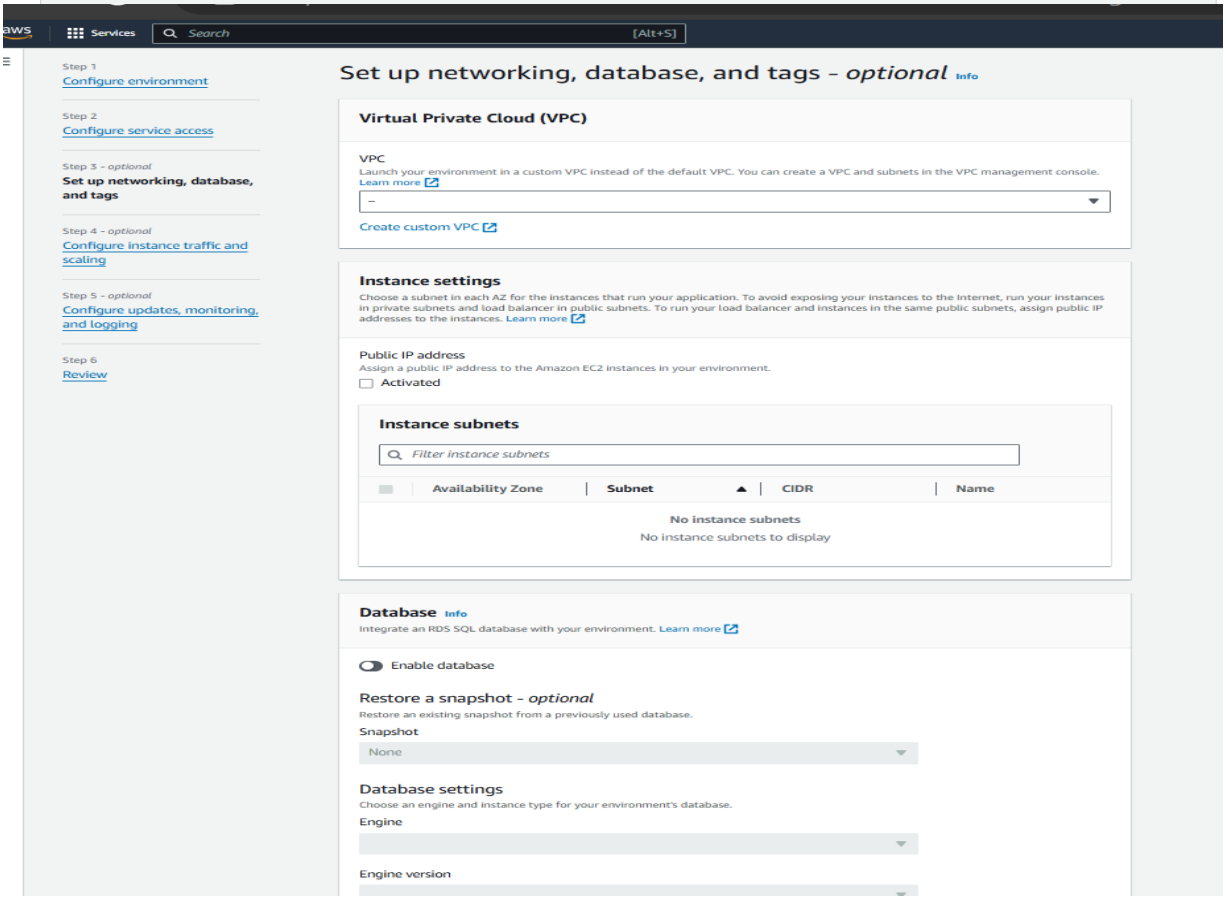
3. Select platform as PHP.

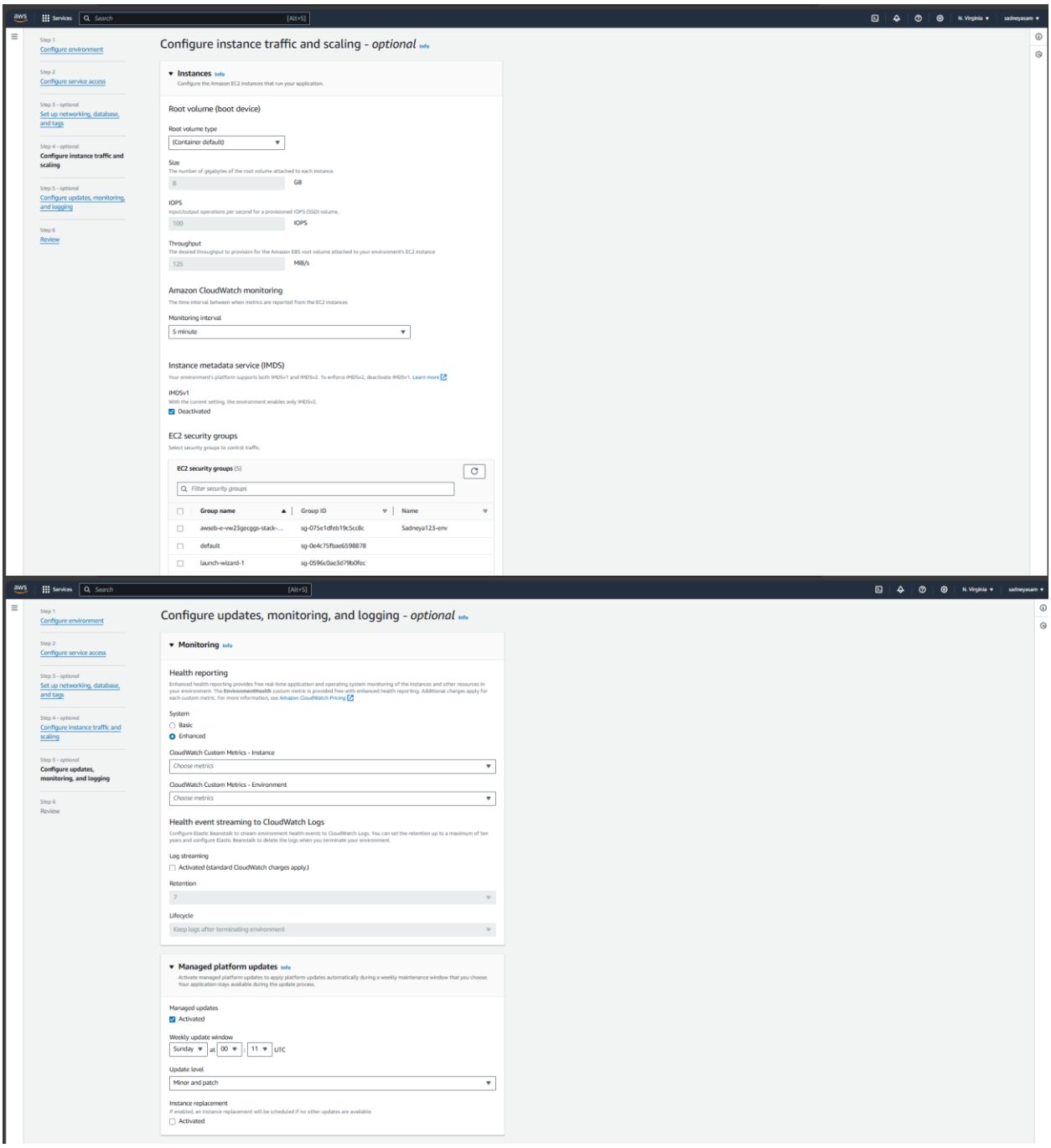


4. After clicking on next u need to select the use existing role. Then you will see the existing role select it like here it is aws-elasticbeanstalk-service-role. Which we created

in 1st part.Select role,then select key you have created then profile will be automatically selected according to role.then click on create application by keeping all the remaining settings as it is.

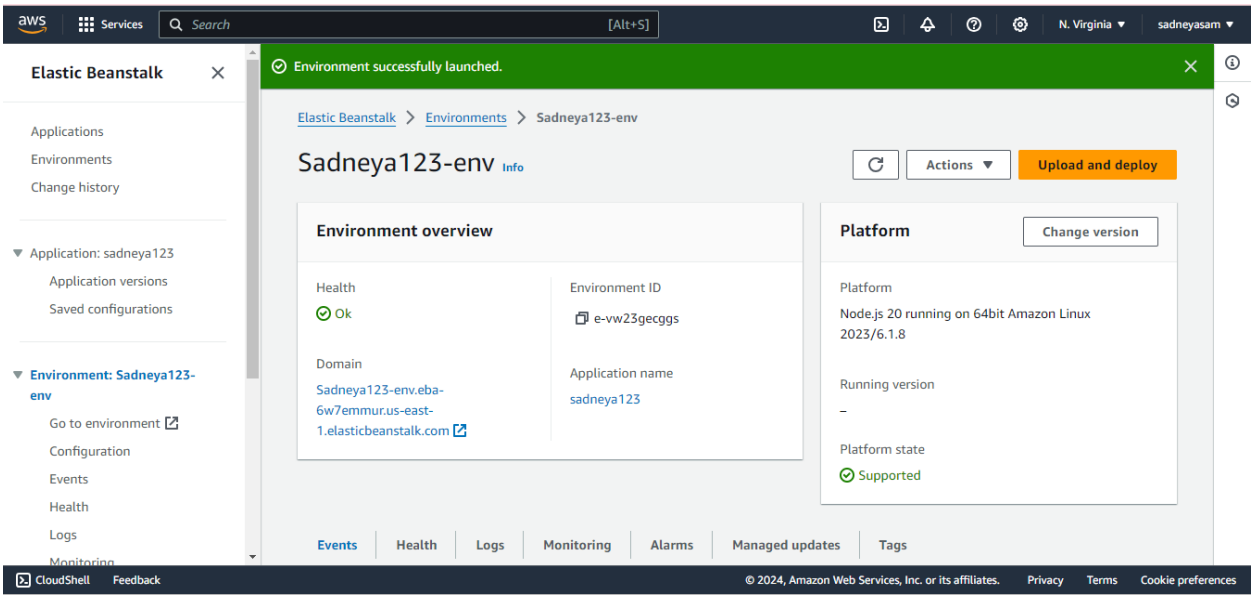




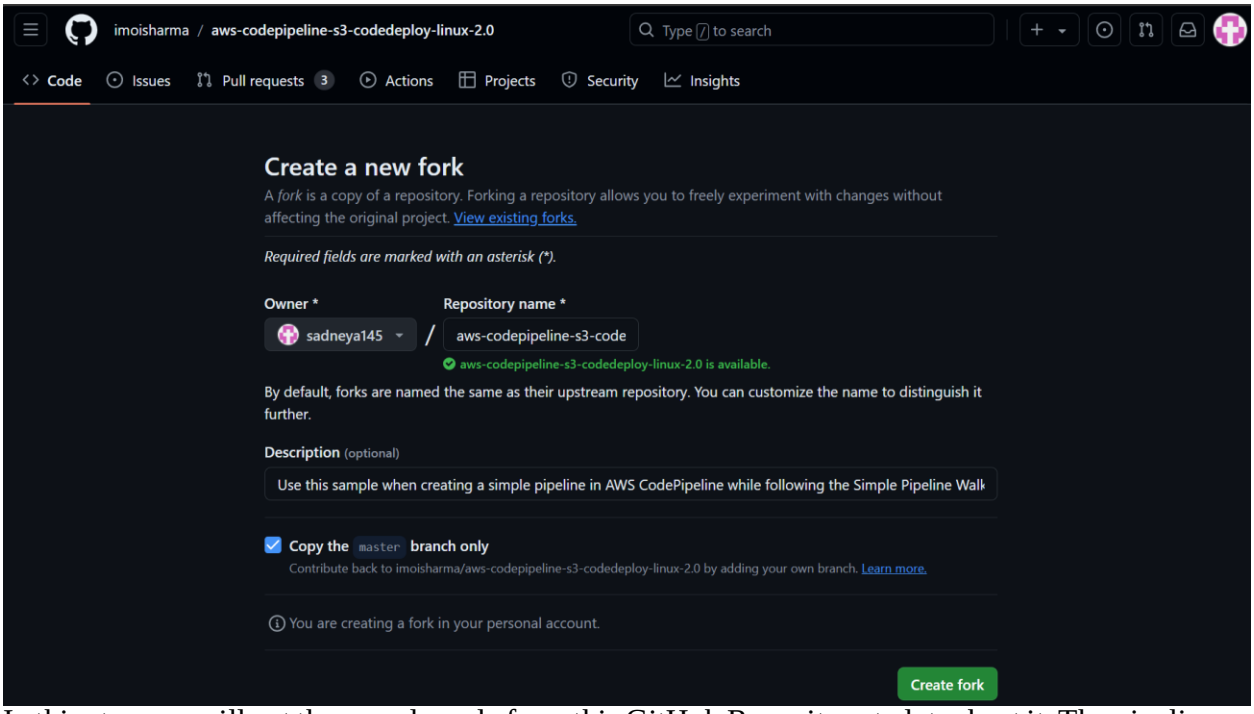


Keep Set up networking, database, and tags ,Configure instance traffic and scaling,Configure updates, monitoring, and logging all these default.

5. Beanstalk creates a sample environment for you to deploy your application. By default, it creates an EC2 instance, a security group, an Auto Scaling group, an Amazon S3 Bucket, Amazon CloudWatch alarms and a domain name for your Application.



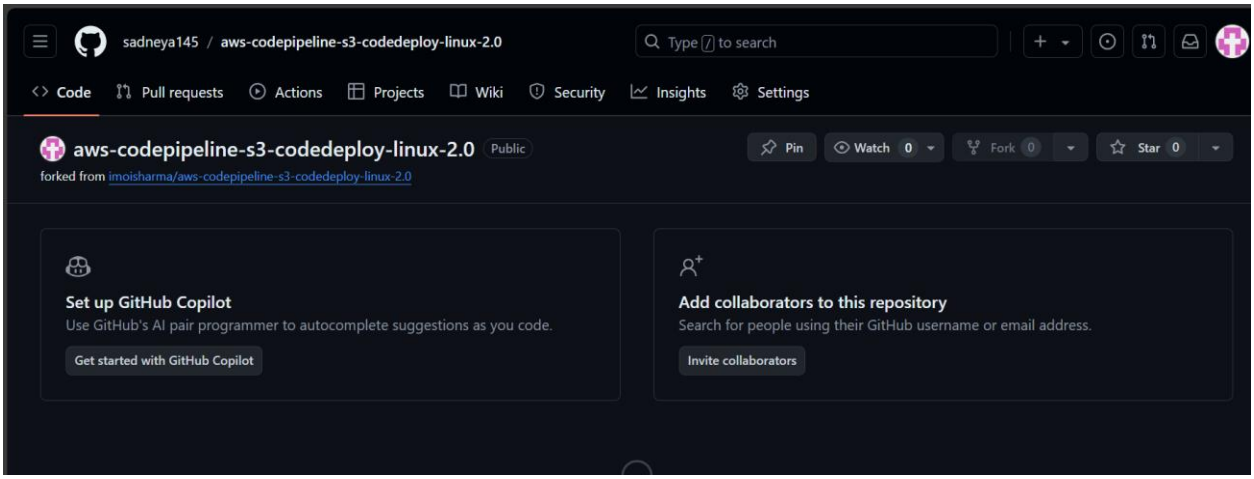
Step 3: Get a copy of your sample code



In this step, we will get the sample code from this GitHub Repository to later host it. The pipeline takes code from the source and then performs actions on it.

For this experiment, as a source, we will use this forked GitHub repository. We can alternatively also use Amazon S3 and AWS CodeCommit.

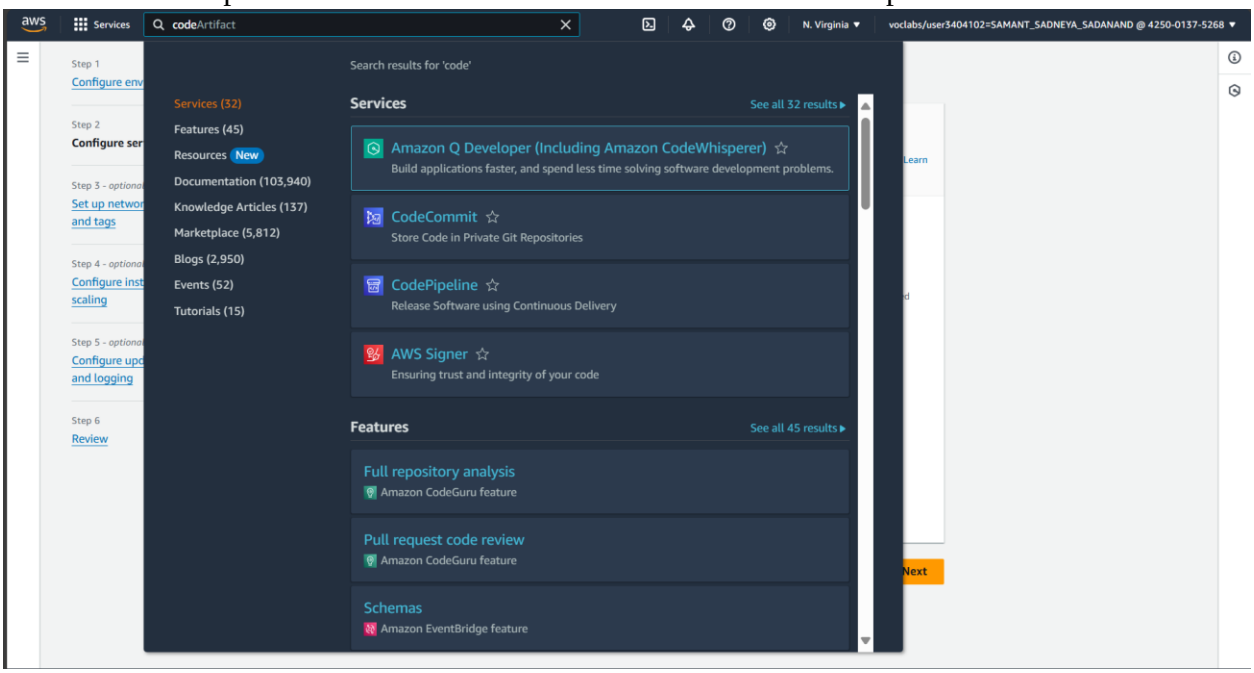
Go to the repository shared above and simply fork it.

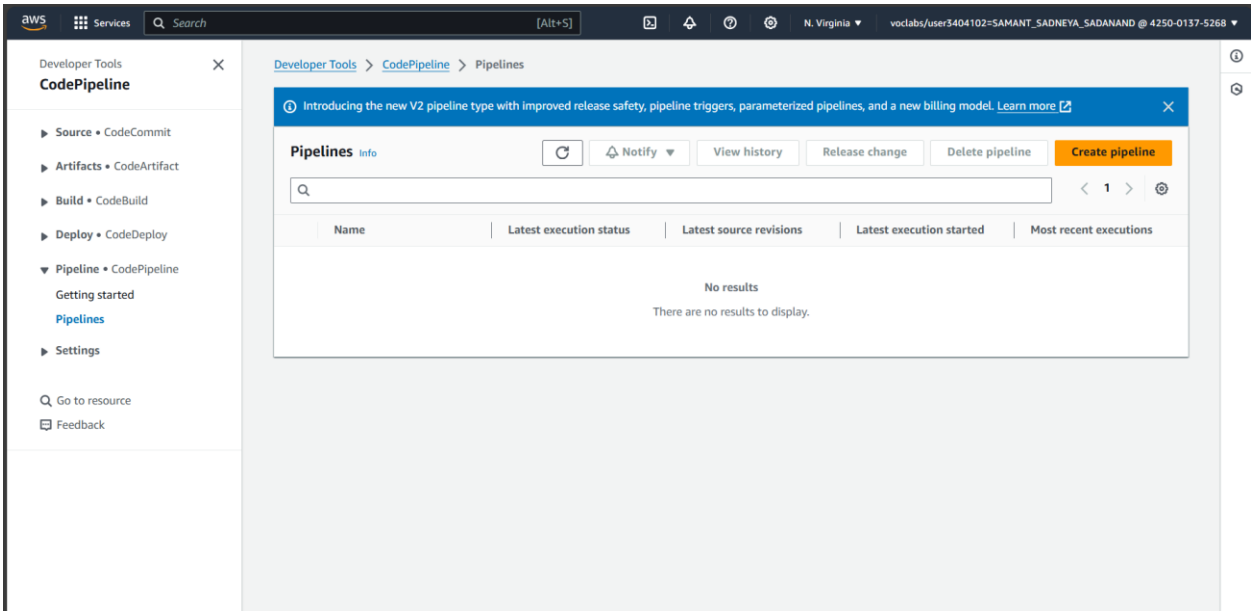


Step 4: Creating a CodePipeline

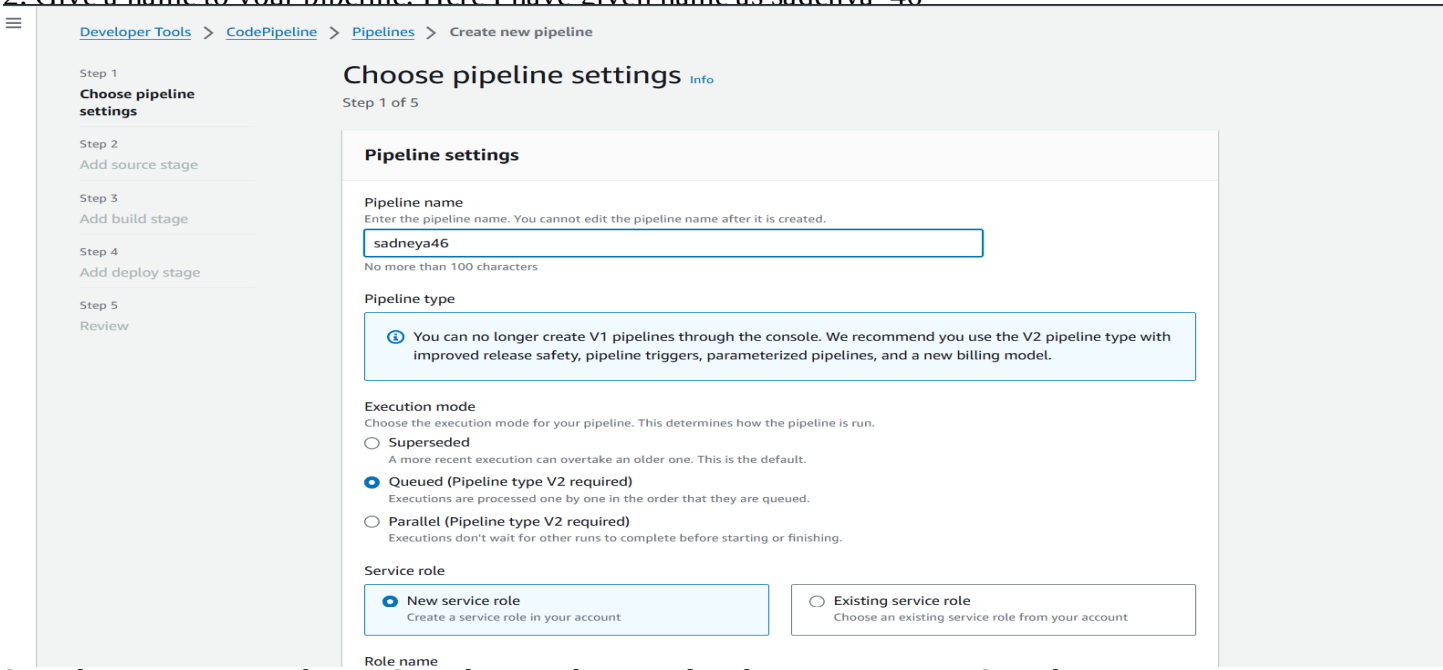
In this step, we'll create a simple pipeline that has its source and deployment information. In this case, however, we will skip the build stage where you get to plug in our preferred build provider.

1. Search CodePipeline in the search bar and click on create a new Pipeline.

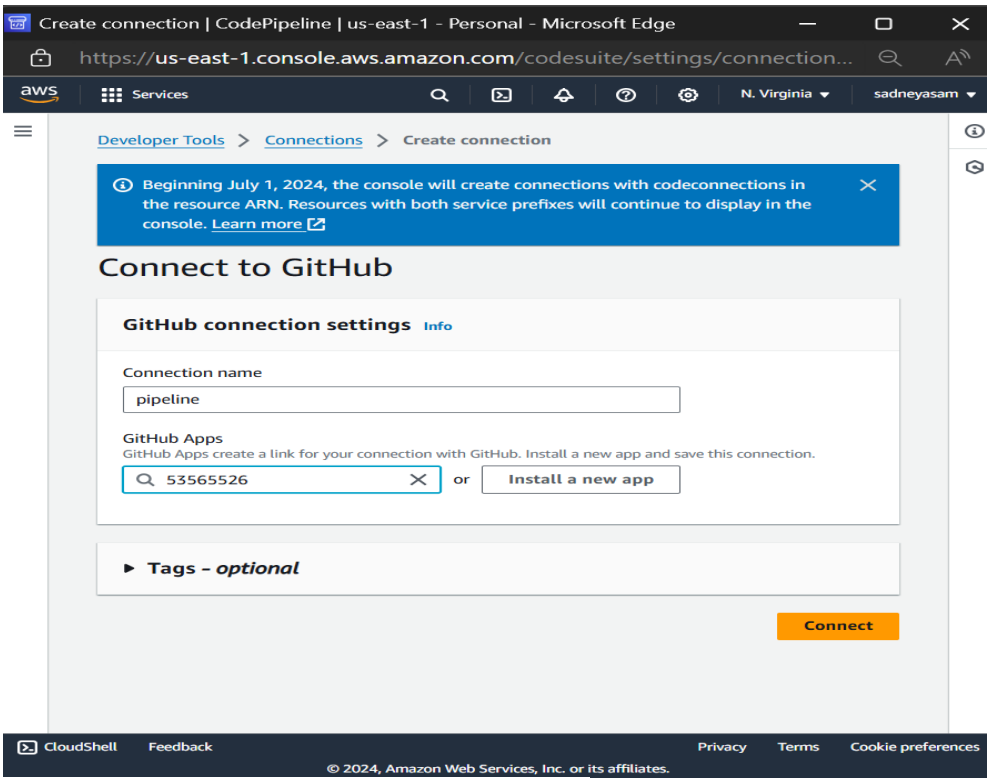


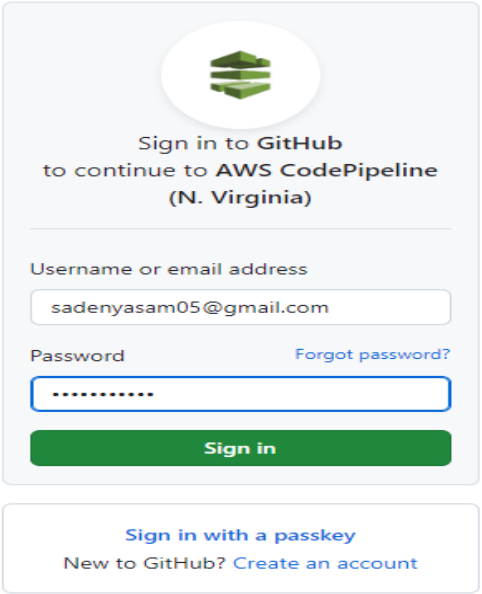


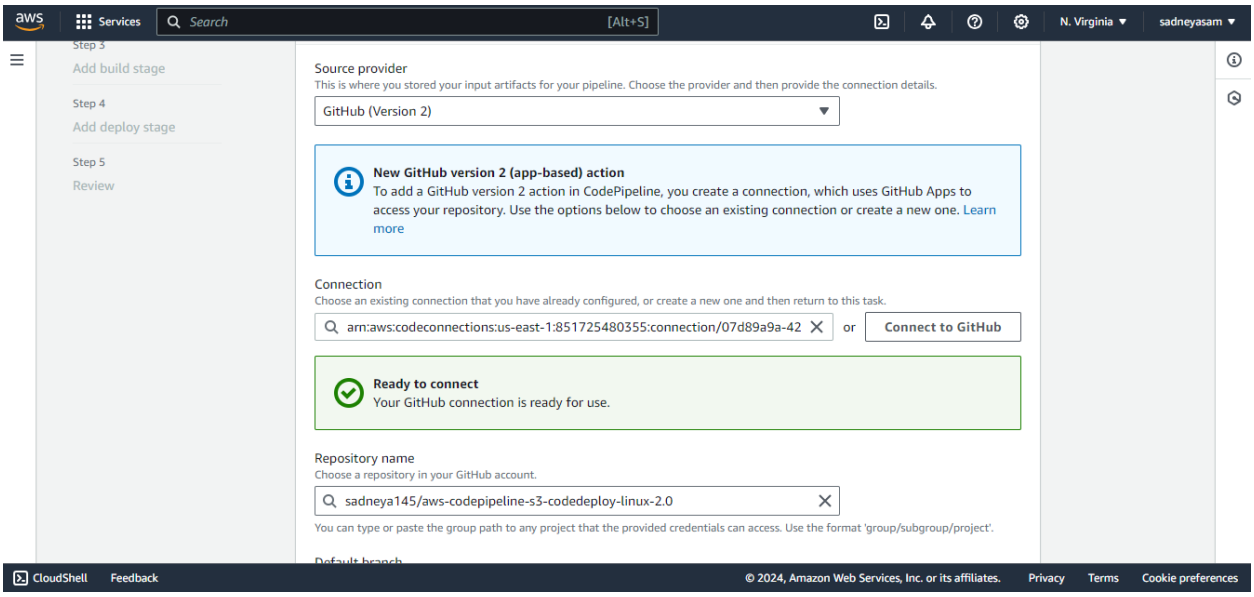
2. Give a name to your pipeline. Here I have given name as sadenya_46



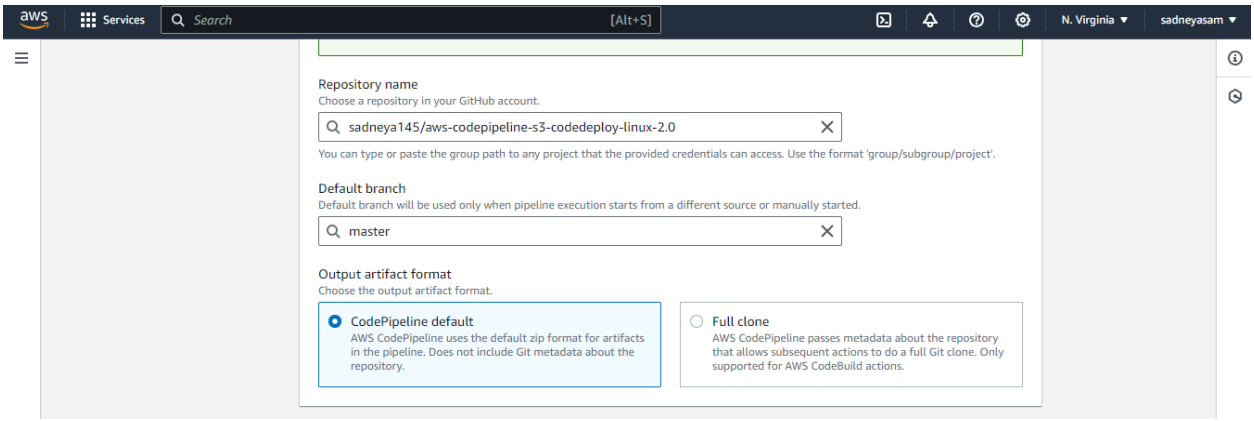
3. In the source stage, choose GitHub v2 as the provider, then connect your GitHub account to AWS by creating a connection. You'd need your GitHub credentials and then you'd need to authorize and install AWS on the forked GitHub Repository.


Create connection | CodePipeline | us-east-1 - Personal - Microsoft Edge
https://us-east-1.console.aws.amazon.com/codesuite/settings/connection...
aws
Services
N. Virginia
sadneyasam
Developer Tools > Connections > Create connection
Beginning July 1, 2024, the console will create connections with codeconnections in the resource ARN. Resources with both service prefixes will continue to display in the console. Learn more
Connect to GitHub
GitHub connection settings Info
Connection name
pipeline
GitHub Apps
GitHub Apps create a link for your connection with GitHub. Install a new app and save this connection.
53565526
or
Install a new app
Tags - optional
Connect
CloudShell
Feedback
Privacy
Terms
Cookie preferences
© 2024, Amazon Web Services, Inc. or its affiliates.

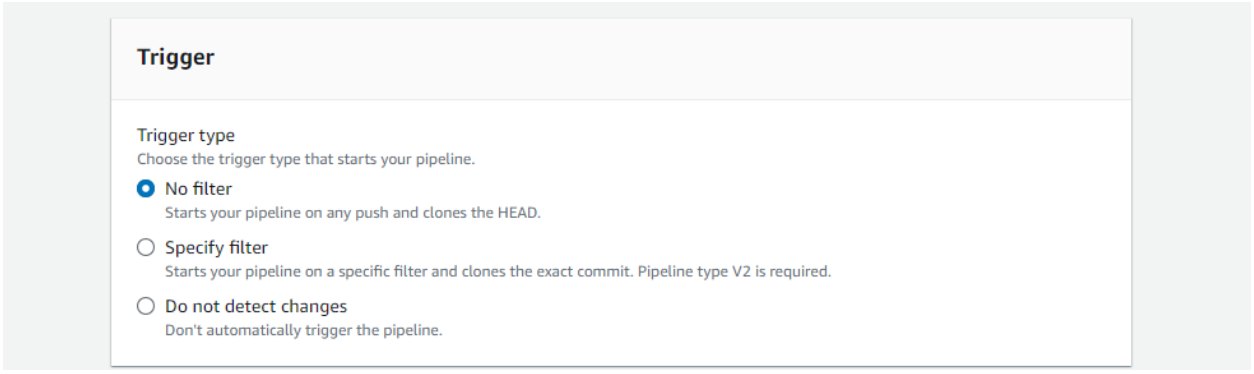

Sign in to GitHub
to continue to AWS CodePipeline
(N. Virginia)
Username or email address
sadenyasam05@gmail.com
Password
Forgot password?
Sign in
Sign in with a passkey
New to GitHub? Create an account



4. select the forked repository then select the master branch.



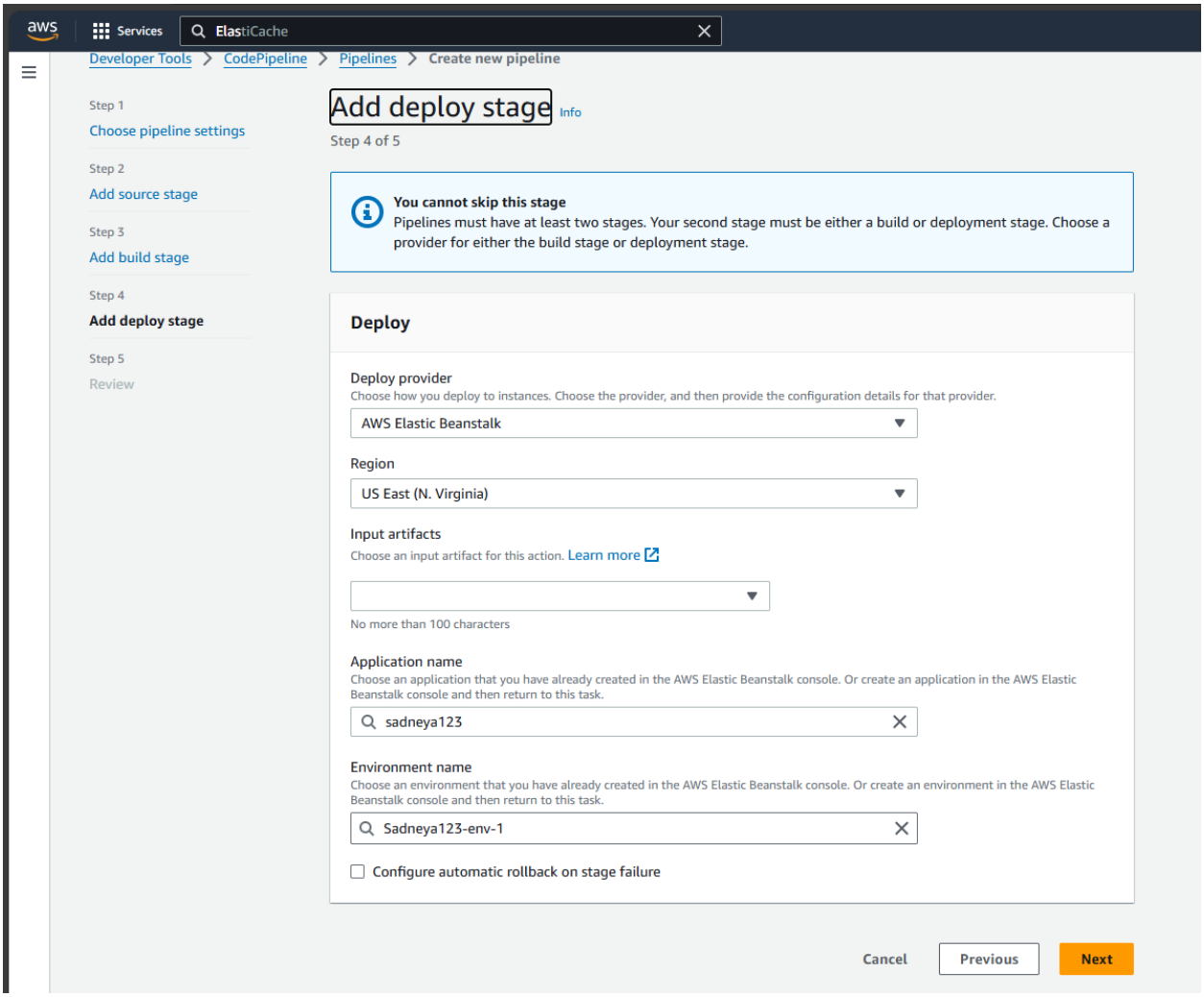
5. Then select trigger type none.



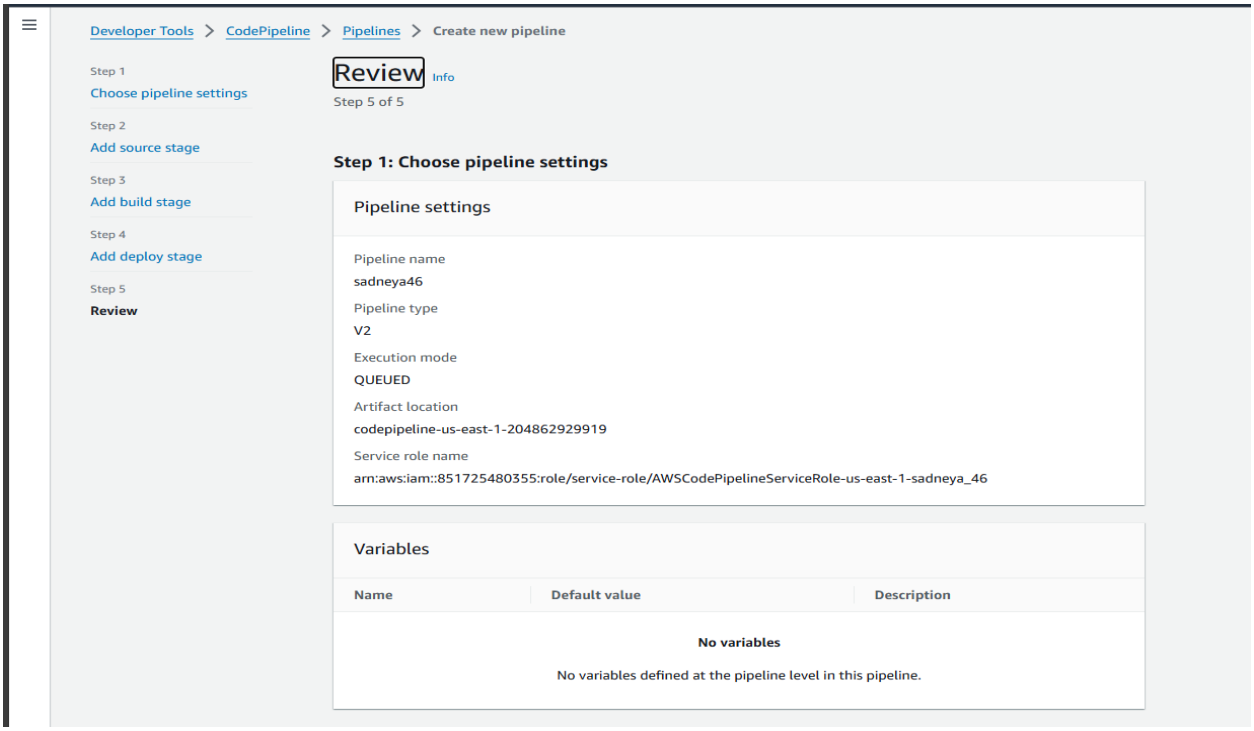
After that, click Continue and skip the build stage. Proceed to the Deployment stage.

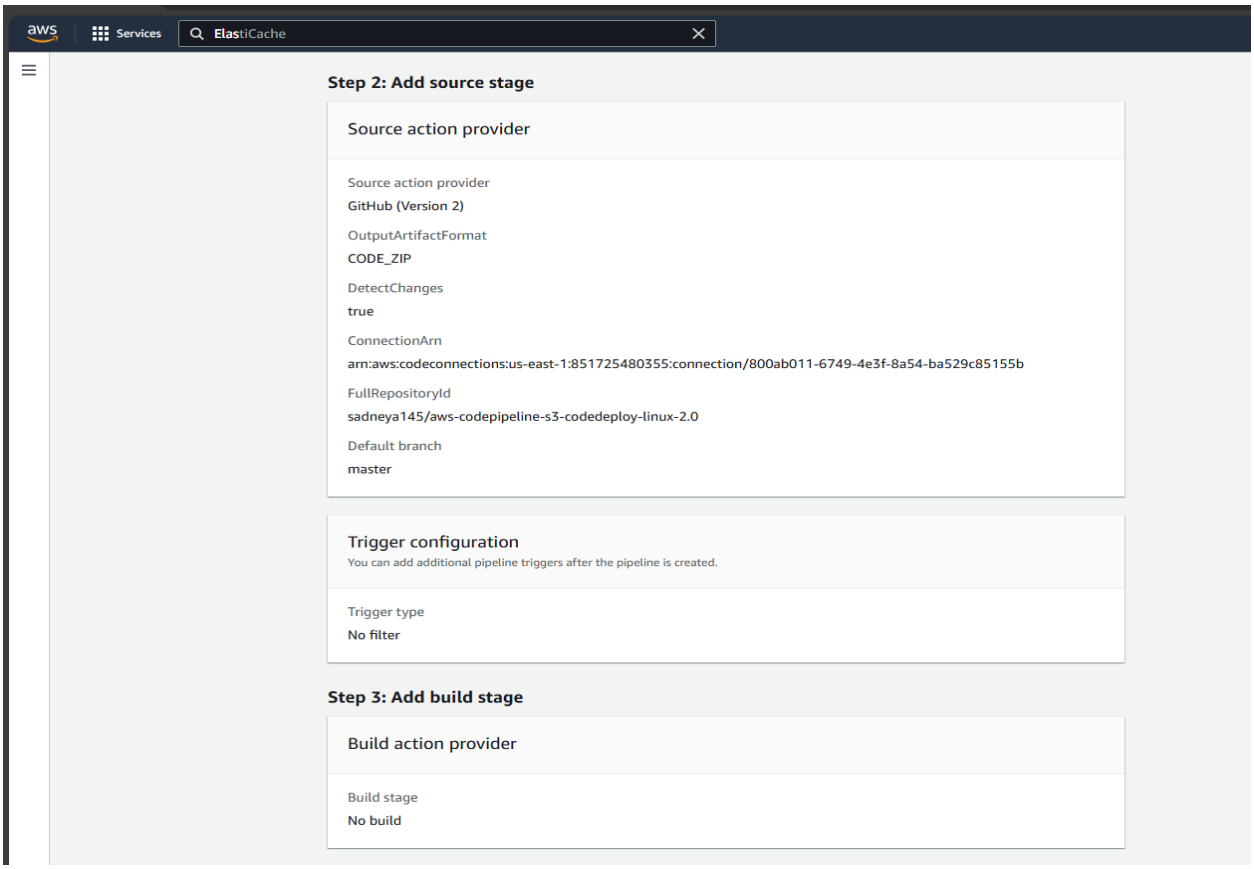
Step 5: Deployment

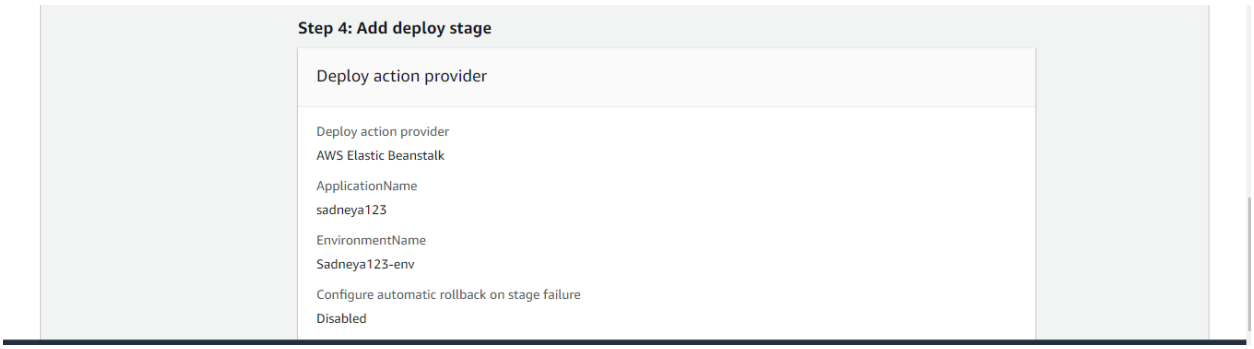
1. Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name.



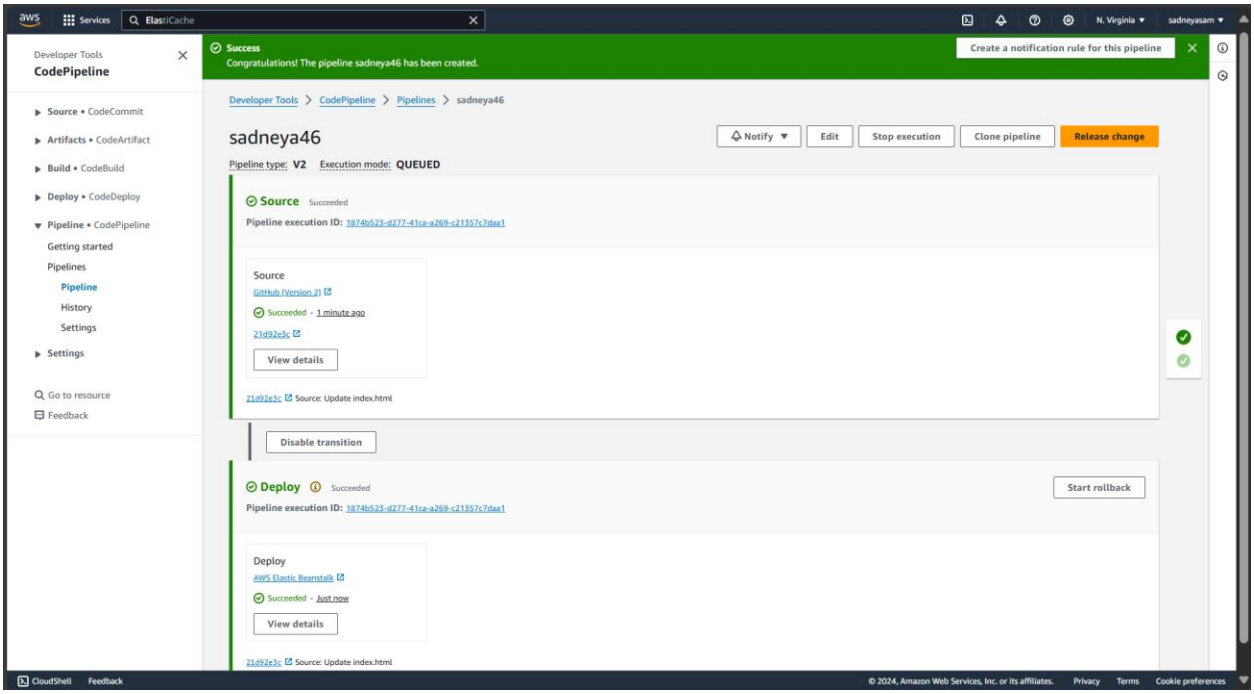
2.Click Next, Review and create the pipeline.



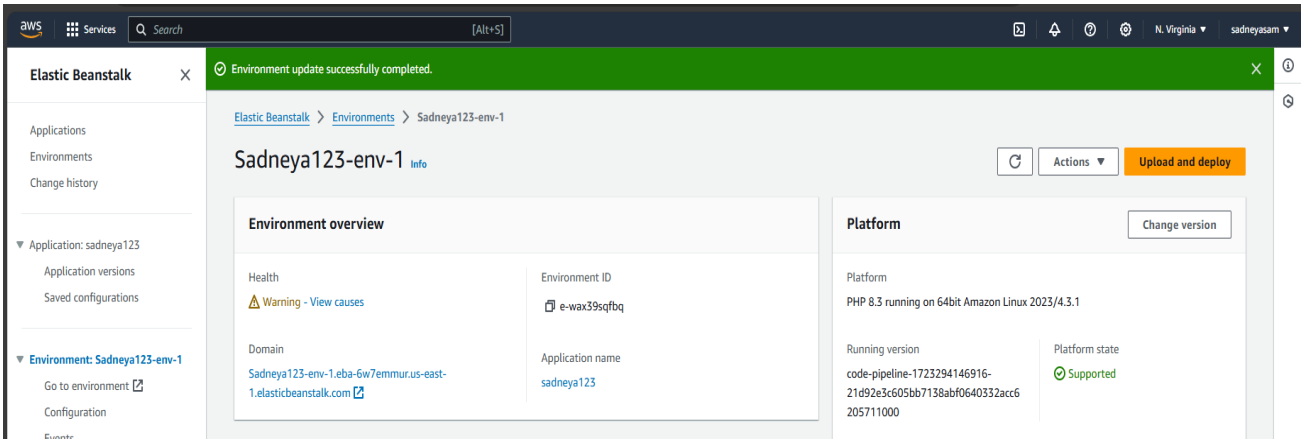


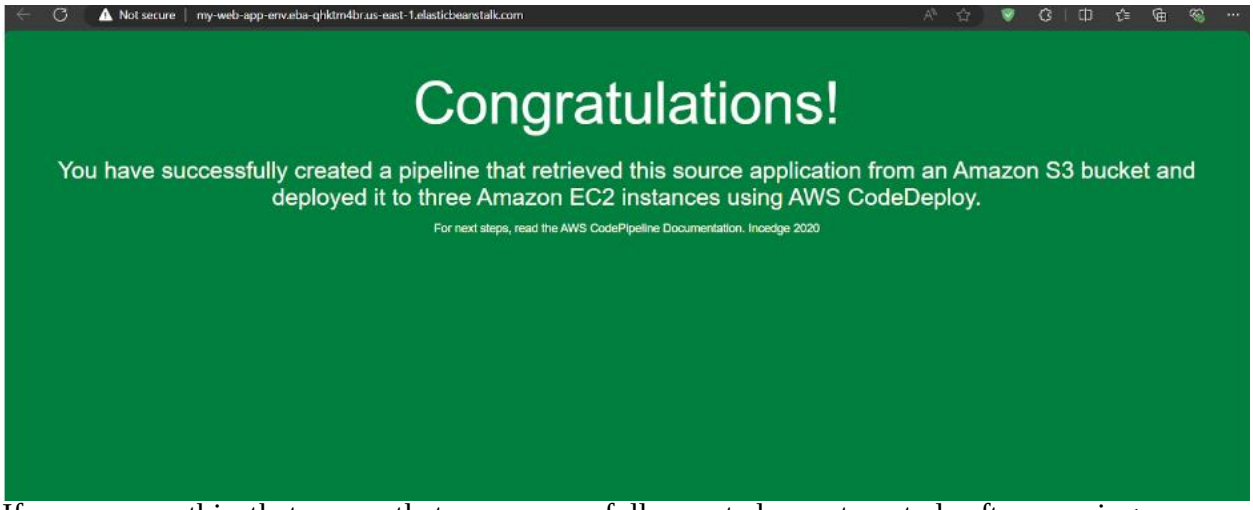


3. Then it will give you this result on screen. i.e. deployed successfully.



4. In a few minutes the website will get hosted successfully. Then click on the url present over the environment created on Elastic Benstalk.

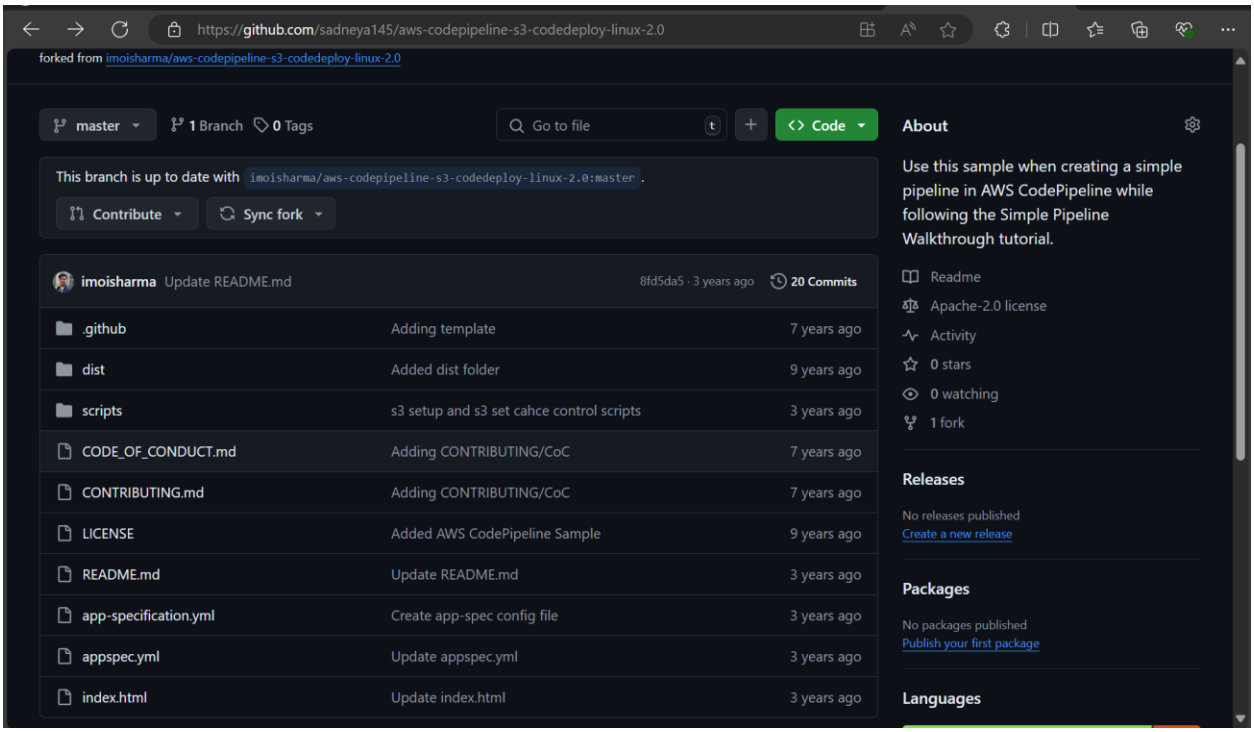


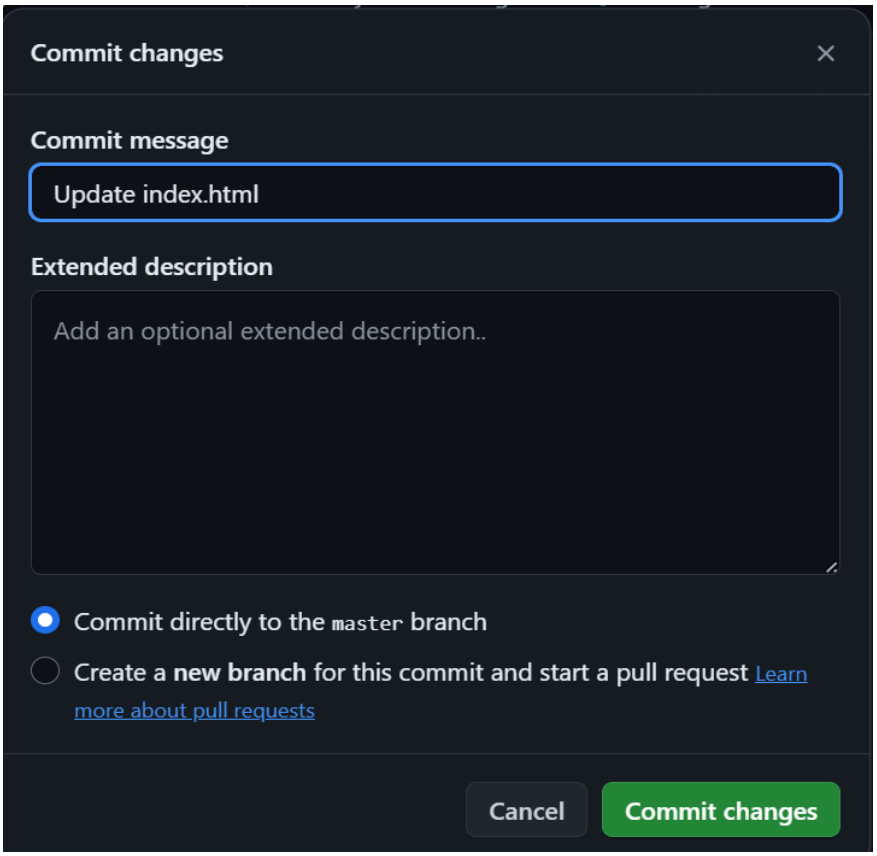


If you can see this, that means that you successfully created an automated software using CodePipeline.

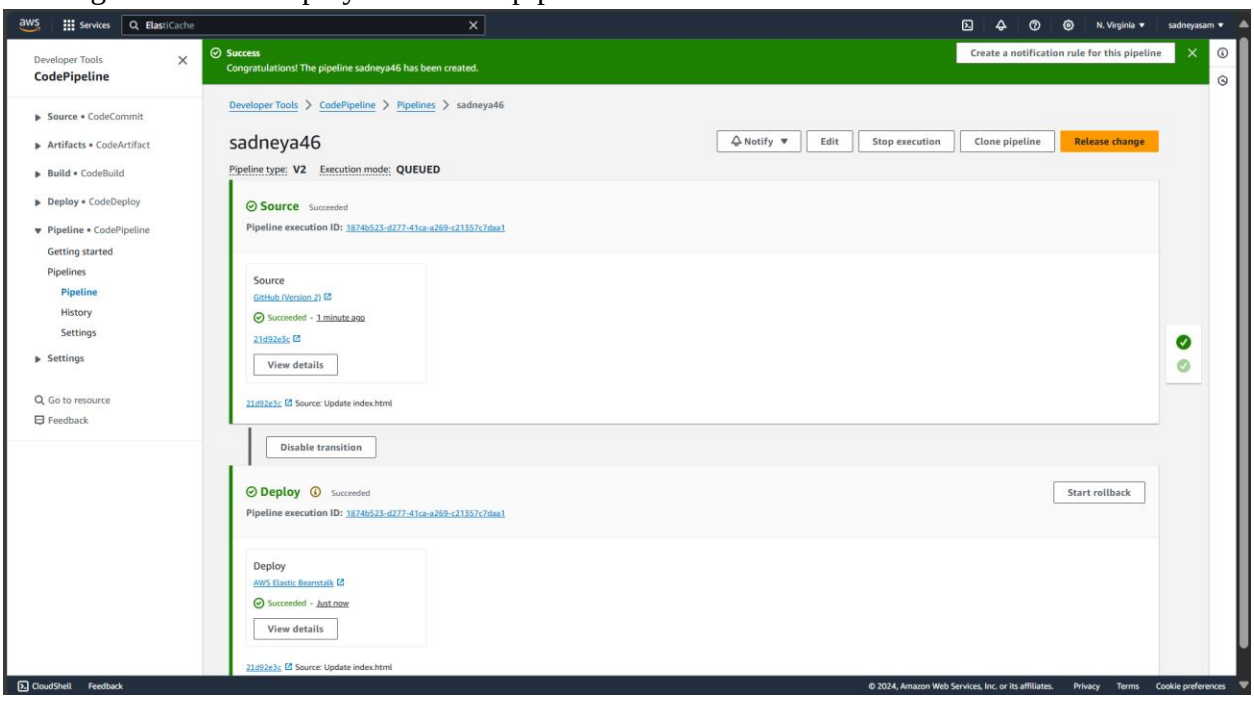
Step 6: Committing changes to update app

1. In this we make some changes in the file.Open github.com then open the forked repository. Then update the changes in the index.html file and finally commit those changes.





2. Then again start the deployment of the pipeline.



3. Check the changes in the website , here I have added a message in h3 tag.


Not secure | sadneya123-env-1.eba-6w7emmur.us-east-1.elasticbeanstalk.com
Congratulations!
You have successfully created a pipeline that retrieved this source application from an Amazon S3 bucket and deployed it to three Amazon EC2 instances using AWS CodeDeploy.
Hello It's me D15C-46
For next steps, read the AWS CodePipeline Documentation. Incedge 2020