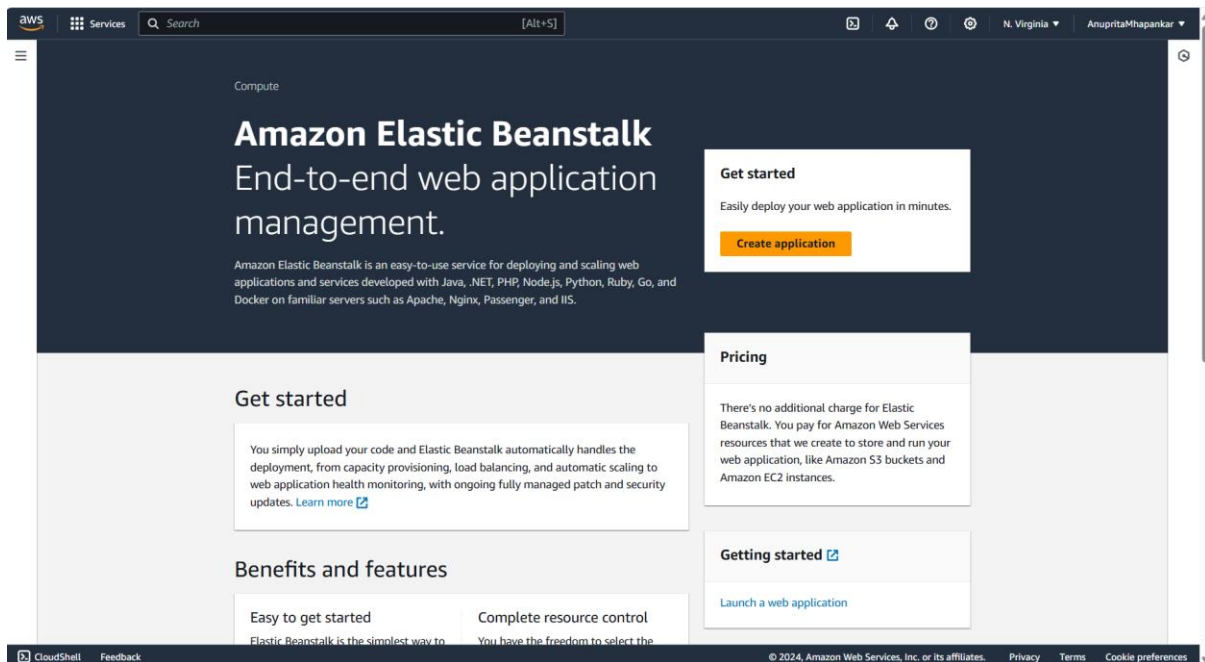


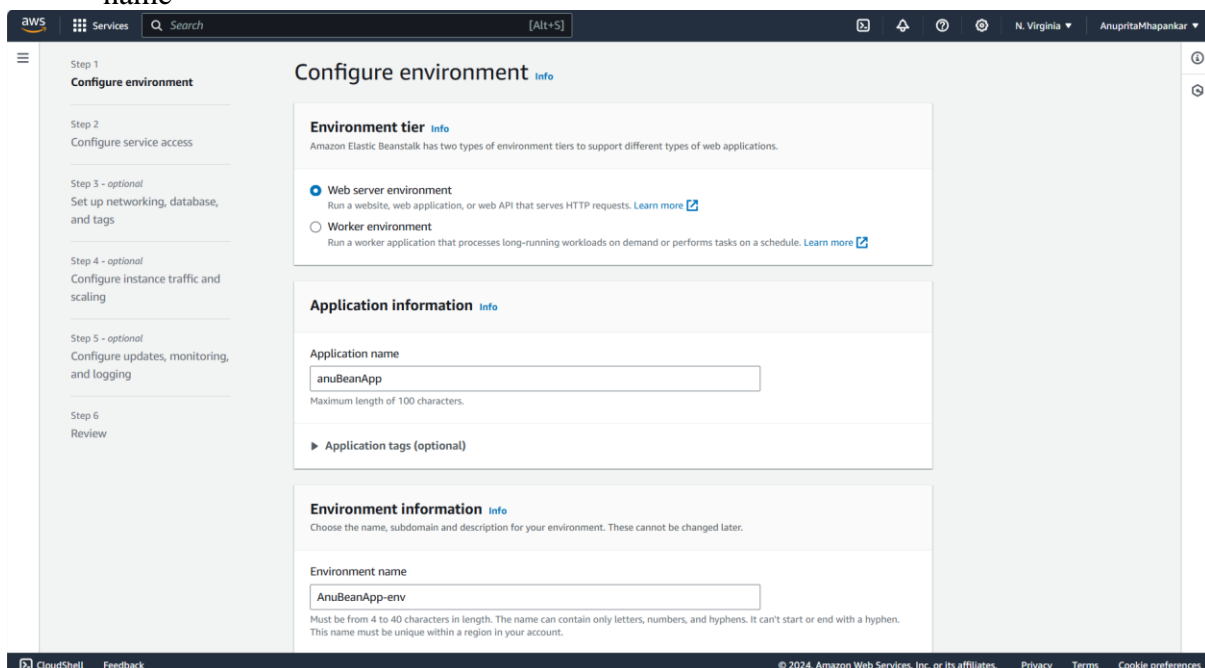
Experiment 2

Using Beanstalk

1. Open the aws console and then search Elastic Beanstalk (Opens a dashboard as seen below)



2. Click on create application and configure the environment by adding your application name



3. Choose PHP from the drop-down menu and click next

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

4.3.2 (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

- Now comeback to Elastic Beanstalk page and from the drop down menu select the newly created key pair and instance profile

Configure service access Info

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role

- ☒ **Create and use new service role**
- ☐ **Use an existing service role**

Service role name

Enter the name for an IAM role that Elastic Beanstalk will create to assume as a service role. Beanstalk will attach the required managed policies to it.

aws-elasticbeanstalk-service-role

[View permission details](#)

EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

myKey

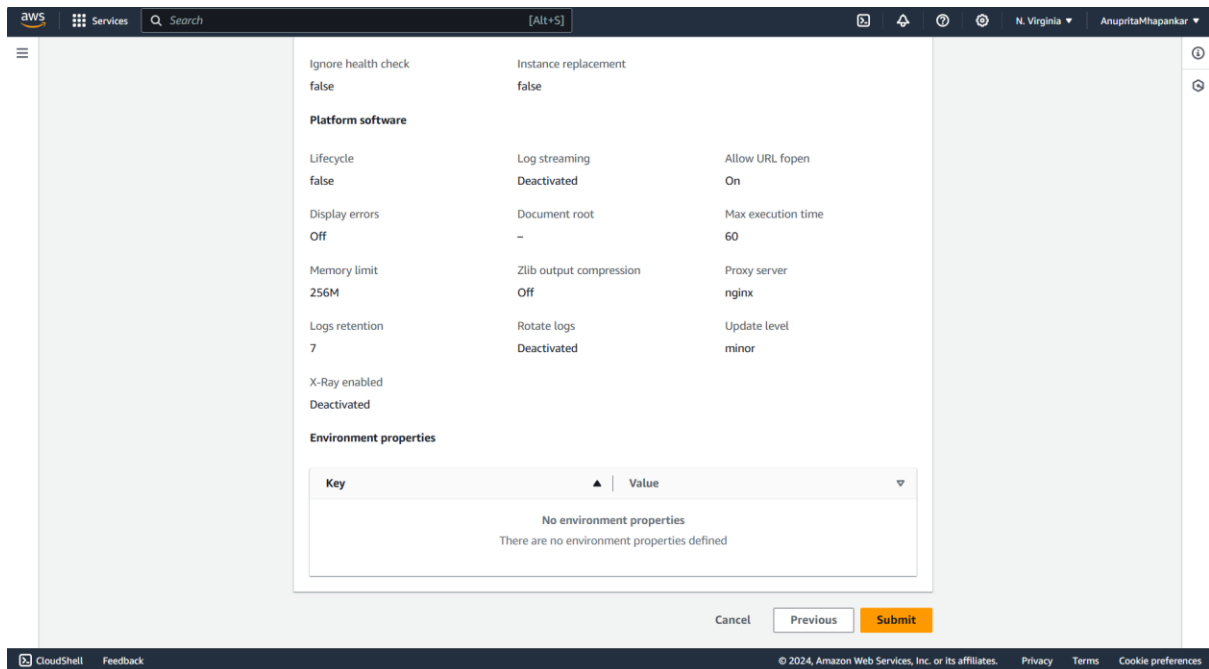
EC2 instance profile

Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

[View permission details](#)

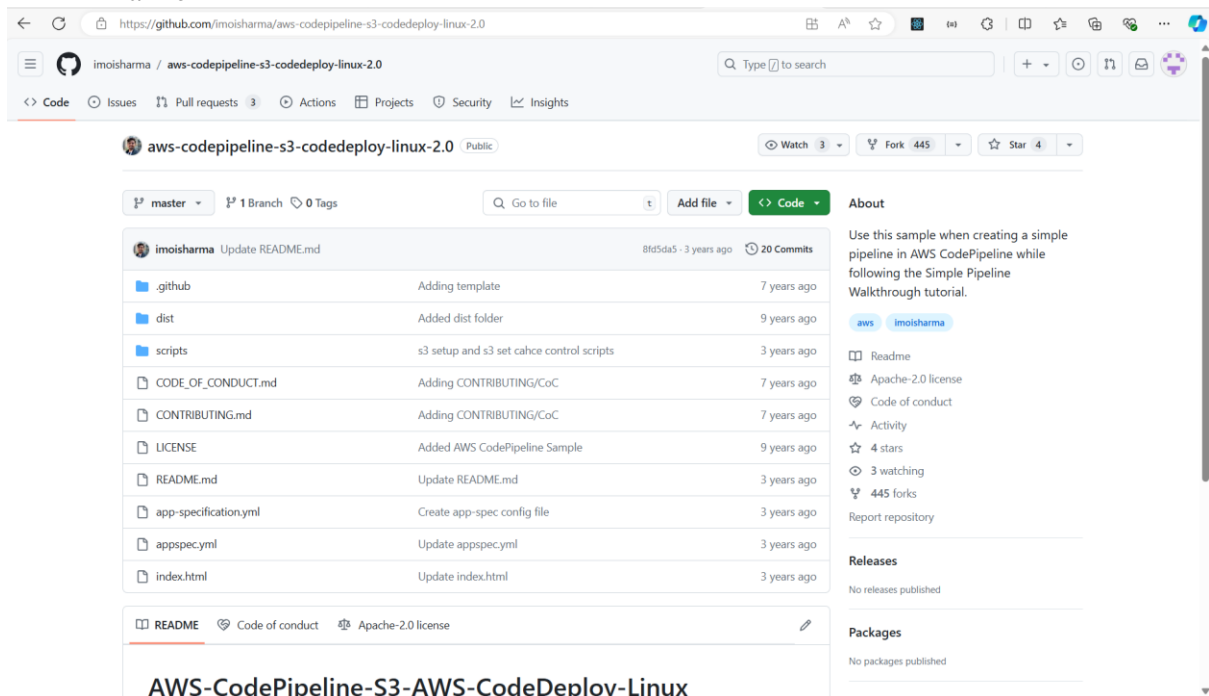
Cancel Skip to review Previous **Next**

- Now review the changes made and click on create application

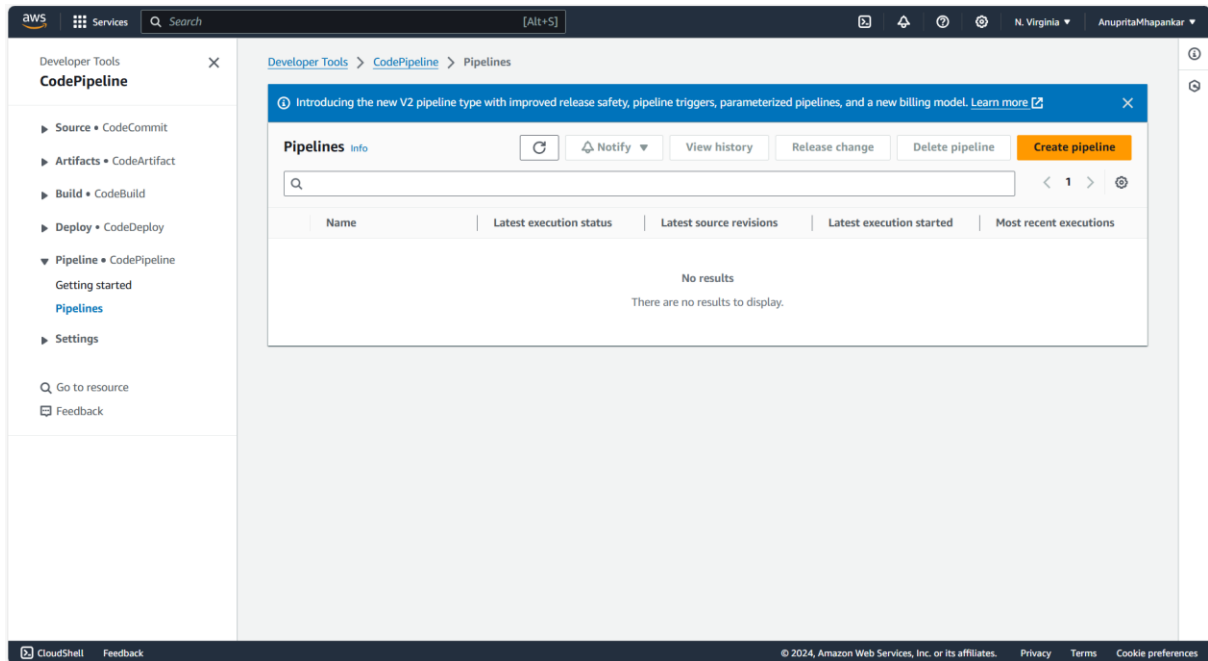


Pipeline Creation :

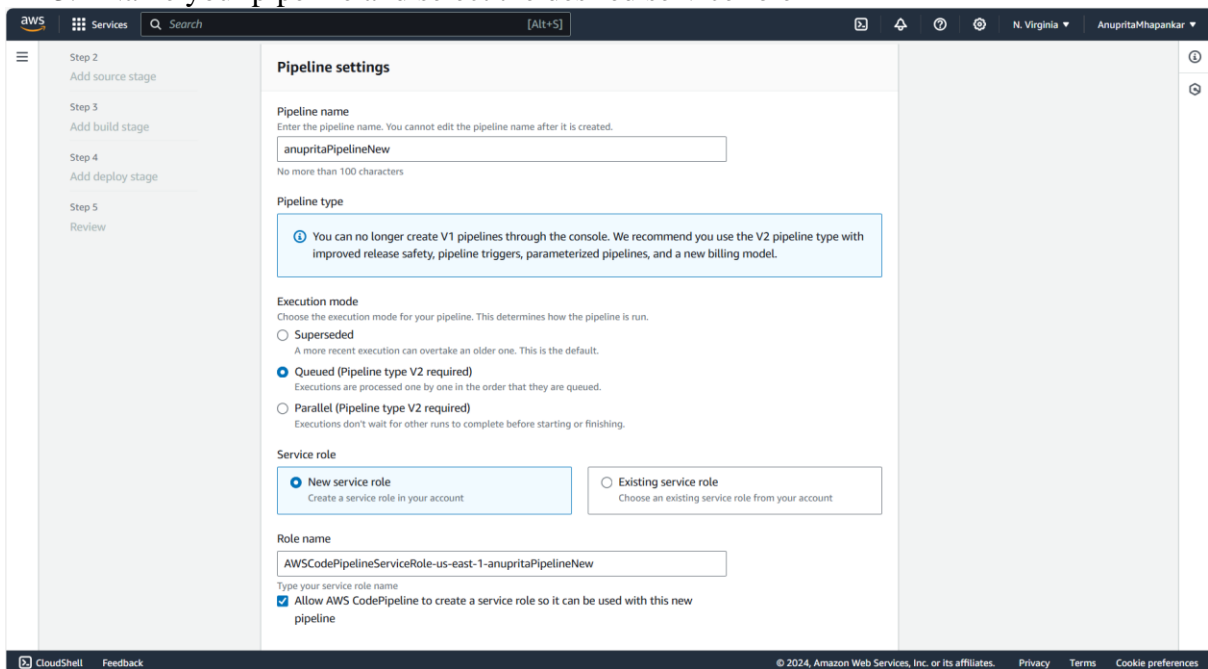
1. Fork a github repo for aws codepipeline available as The pipeline takes code from the source and then performs actions on it.We don't need to code from scratch in this manner



2. Go to developer tools and select CodePipeline and create a new pipeline



3. Name your pipeline and select the desired service role



Service role

☒ New service role
Create a service role in your account

☐ Existing service role
Choose an existing service role from your account

Role name

AWSCodePipelineServiceRole-us-east-1-anupritaPipeline

Type your service role name

☒ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Variables

You can add variables at the pipeline level. You can choose to assign the value when you start the pipeline. Choosing this option requires pipeline type V2. [Learn more](#)

No variables defined at the pipeline level in this pipeline.

[Add variable](#)

You can add up to 50 variables.

[The first pipeline execution will fail if variables have no default values.](#)

[Advanced settings](#)

Cancel [Next](#)

4. In the source stage select Github v2 as the provider and then connect your github connect so that the pipeline can access the forked source code. For this purpose create aws github connection and with your credentials install the AWS under the forked repository



Create a connection [Info](#)

Create GitHub App connection [Info](#)


Connection name


► **Tags - optional**



[Connect to GitHub](#)



AWS Connector for GitHub by **Amazon Web Services** would like permission to:

 Verify your GitHub identity (Anuprita579)

 Know which resources you can access


 Act on your behalf
 [Learn more](#)

[Learn more about AWS Connector for GitHub](#)

Cancel


Authorize AWS Connector for
GitHub

Authorizing will redirect to
<https://redirect.codestar.aws>


 Not owned or operated by GitHub

 Created 4 years ago

https://github.com/apps/aws-connector-for-github/installations/new/per... A



Install AWS Connector for GitHub

Install on your personal account Anuprita Mhapankar 

for these repositories:

☒ **All repositories**
This applies to all current *and* future repositories owned by the resource owner.
Also includes public repositories (read-only).

☐ **Only select repositories**
Select at least one repository.
Also includes public repositories (read-only).

with these permissions:

✓ **Read** access to issues and metadata

✓ **Read and write** access to administration, code, commit statuses, pull

5. Once the connection is established from the drop down menu select the repository and the branch

New GitHub version 2 (app-based) action
To add a GitHub version 2 action in CodePipeline, you create a connection, which uses GitHub Apps to access your repository. Use the options below to choose an existing connection or create a new one. [Learn more](#)

Connection
Choose an existing connection that you have already configured, or create a new one and then return to this task.

Q `arn:aws:codeconnections:us-east-1:557690619479:connection/fbe678f5-a05` X or [Connect to GitHub](#)

Ready to connect
Your GitHub connection is ready for use.

Repository name
Choose a repository in your GitHub account.

Q `Anuprita579/aws-codepipeline-s3-codedeploy-linux-2.0` X

You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

Default branch
Default branch will be used only when pipeline execution starts from a different source or manually started.

Q `master` X

Output artifact format
Choose the output artifact format.

☒ **CodePipeline default**
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

☐ **Full clone**
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

- Skip the build stage part as we are not plugging in any build provider and in choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name.

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
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.

Q `myBeanApp` X

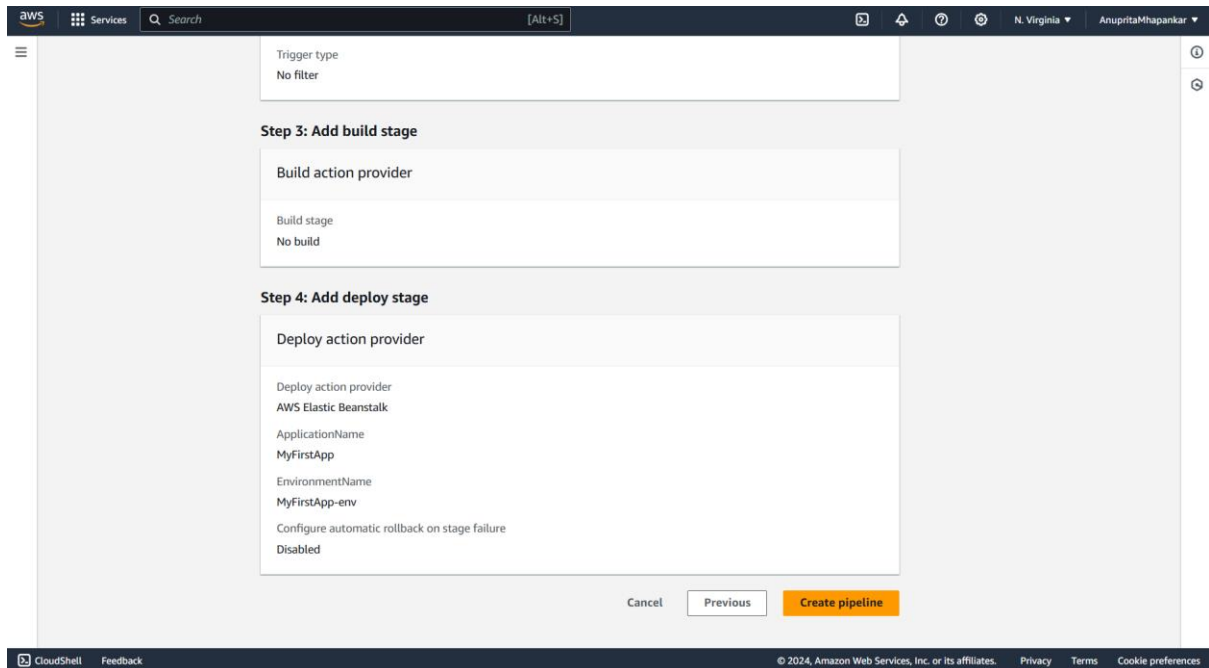
Environment name
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.

Q `MyBeanApp-env` X

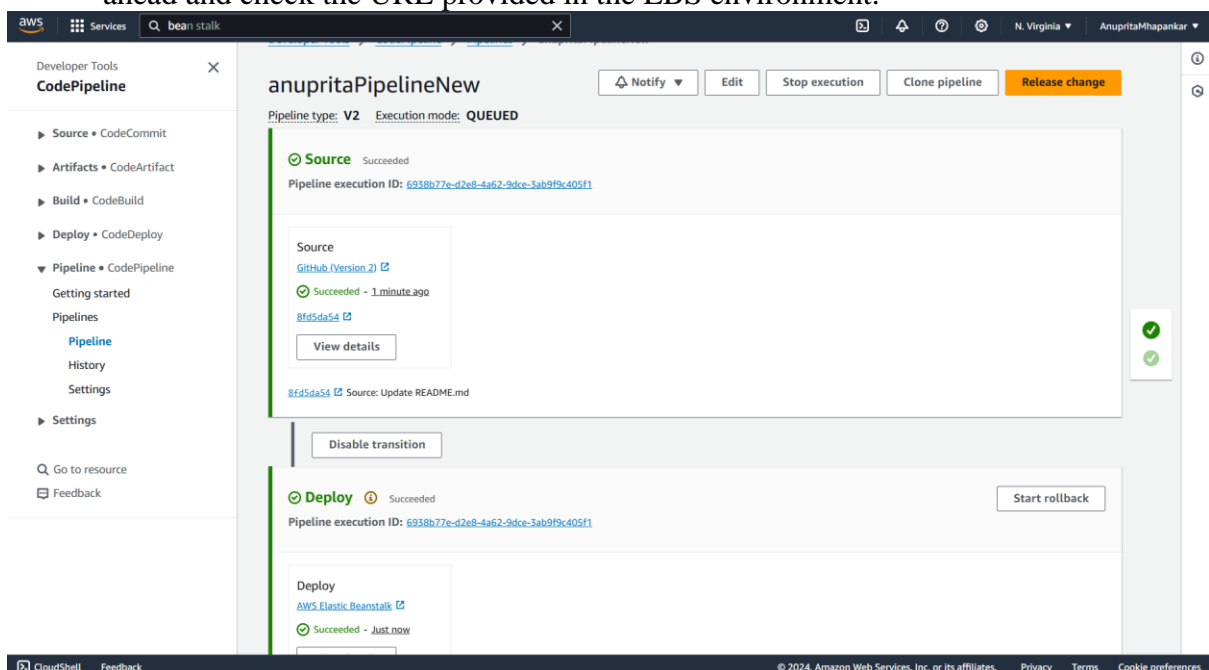
☐ **Configure automatic rollback on stage failure**

[Cancel](#) [Previous](#) [Next](#)

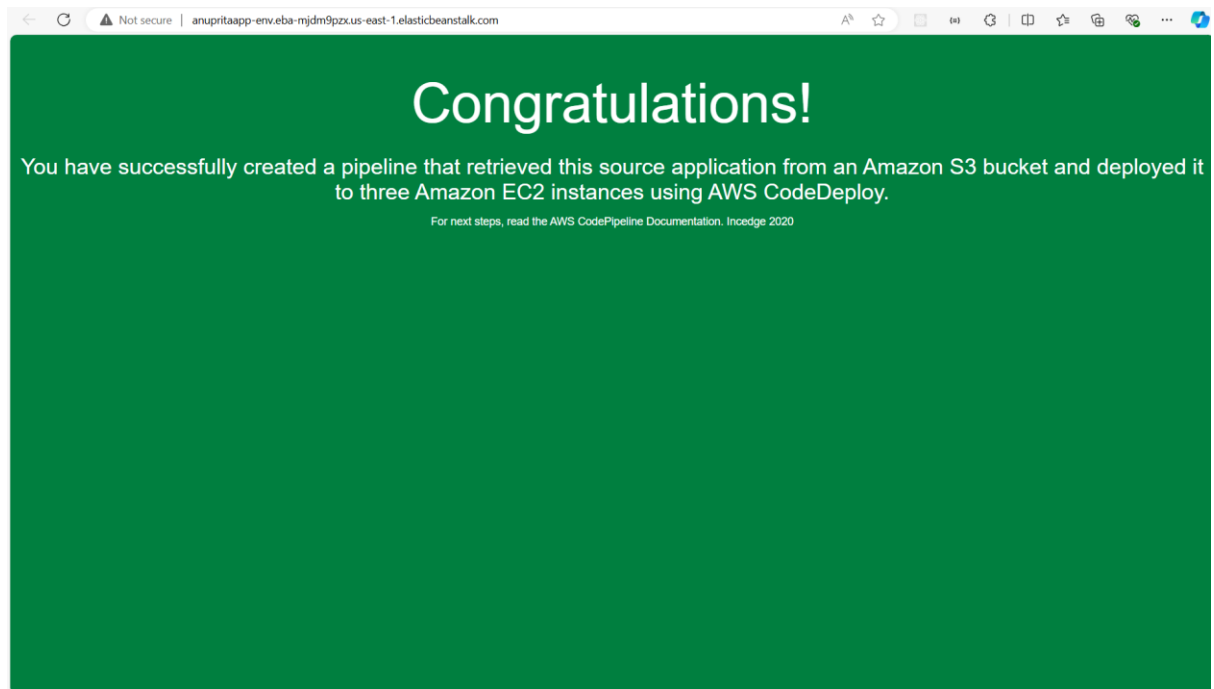
- Review the settings and click on create pipeline



8. Once the Successfully created message appears, your pipeline is created. Then go ahead and check the URL provided in the EBS environment.

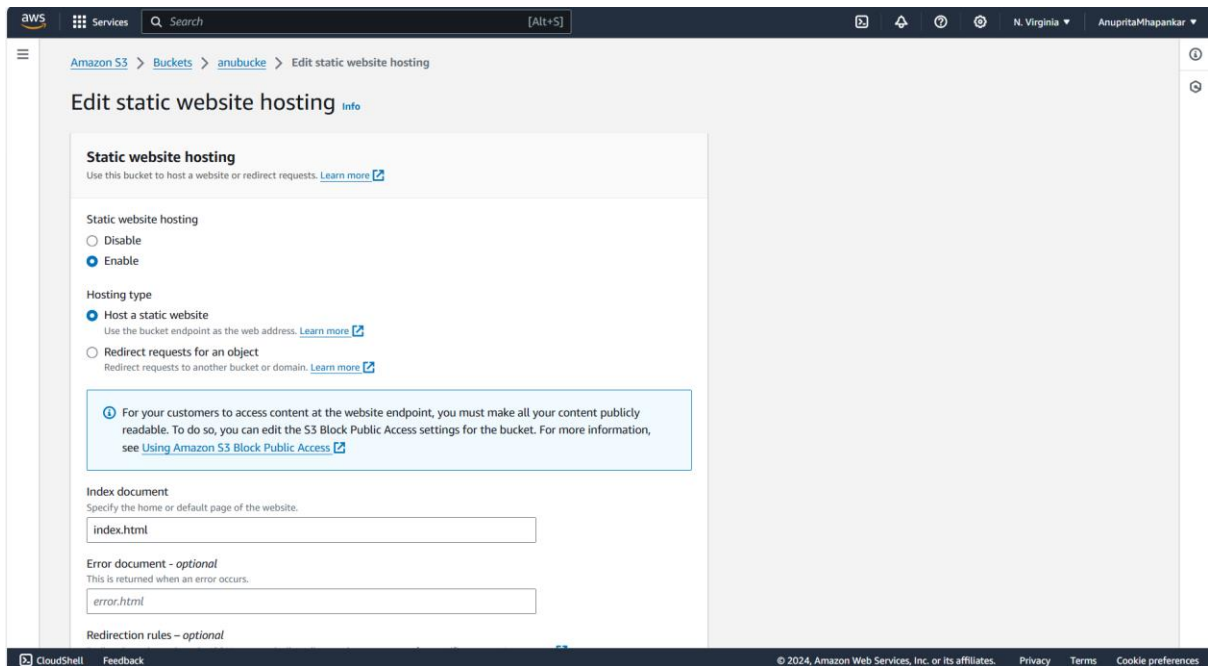


9. This is the website hosted from that forked repo in our beanstalk environment

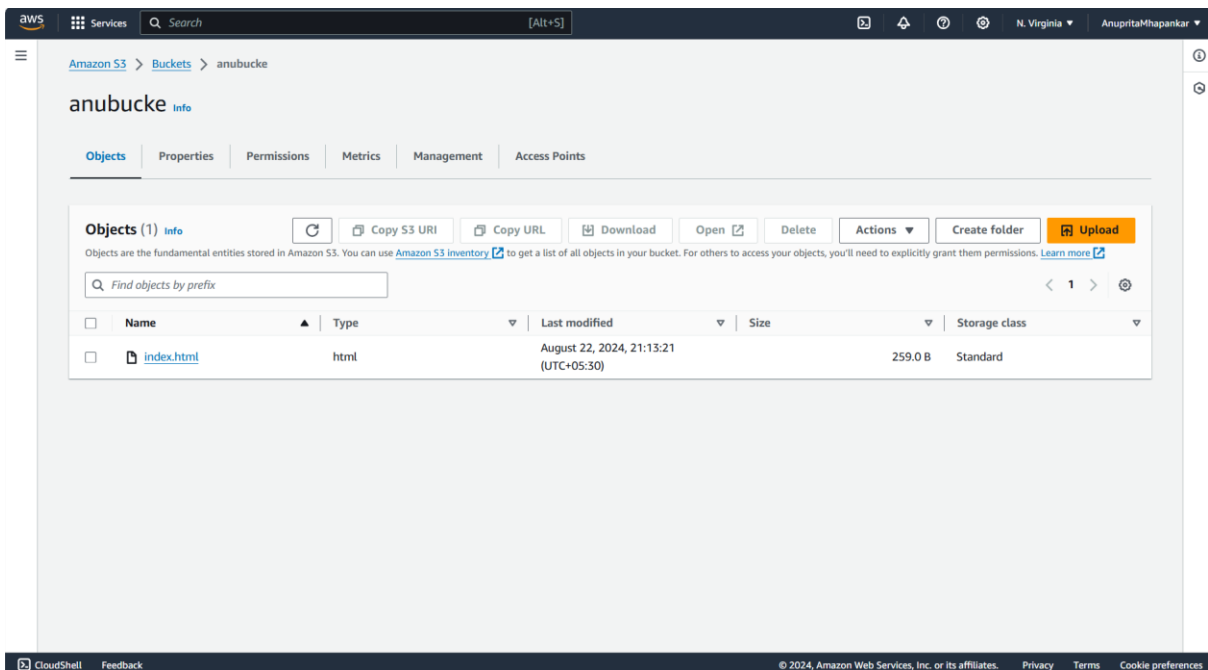


Using S3

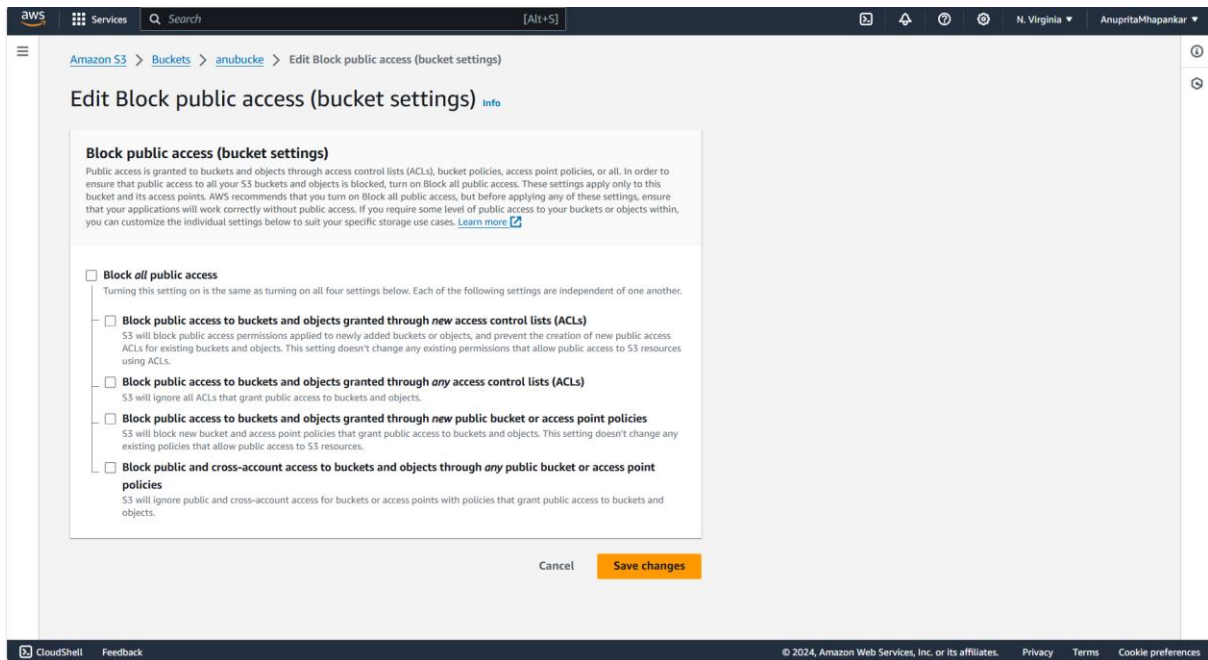
1. Visit S3 under the developer tools and create a Bucket. Click on the Edit Static Website Hosting under the properties tab



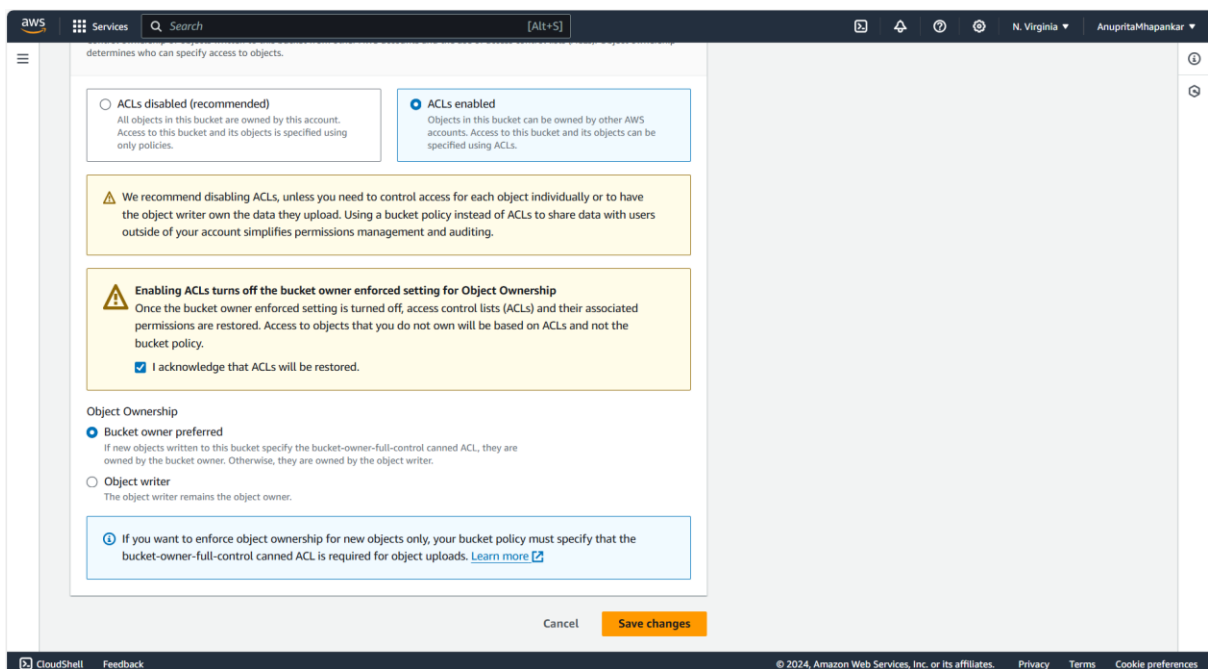
2. Upload a file



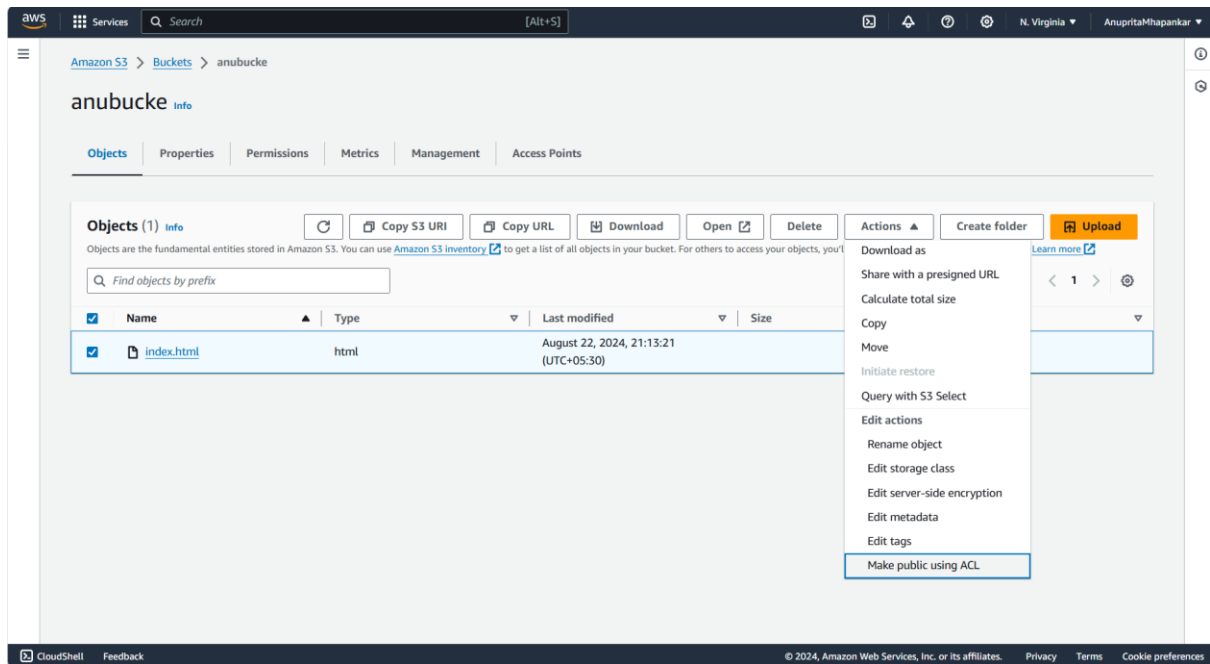
3. Click on the Edit block public access under the Permissions tab



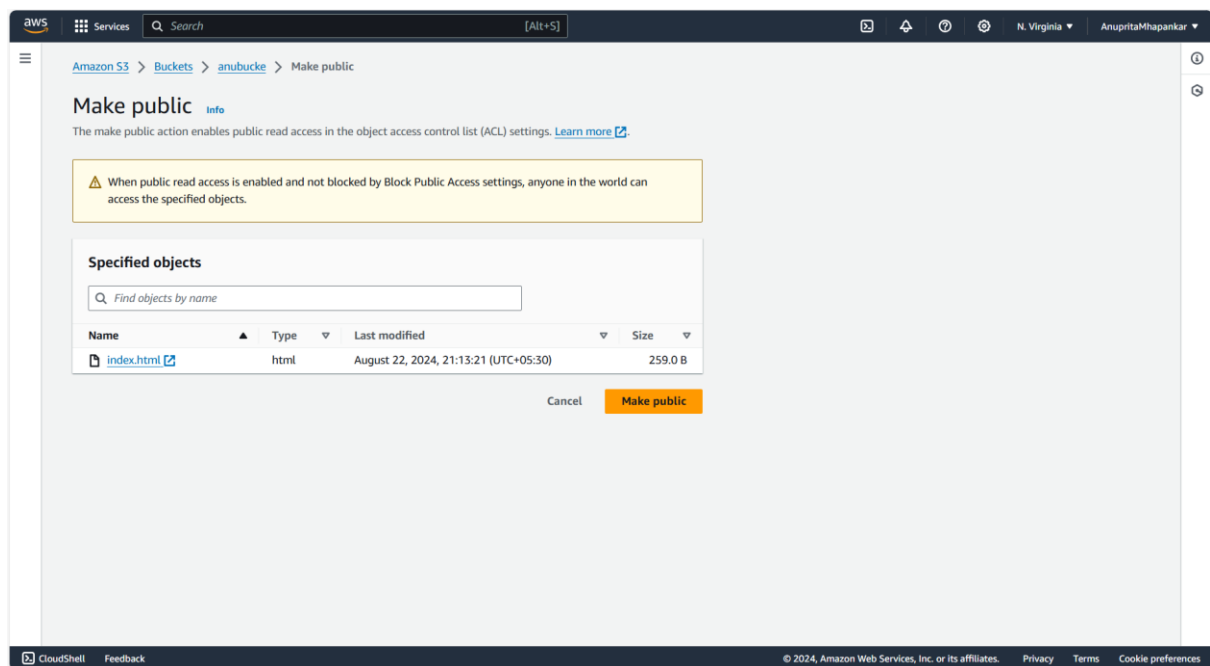
4. Click on Object Ownership under Permission Tab



5. Select the file and click on Actions and select the option Make Public using ACL from the dropdown



6. Select on Make Public



7. Visit the domain and the website hosted.

