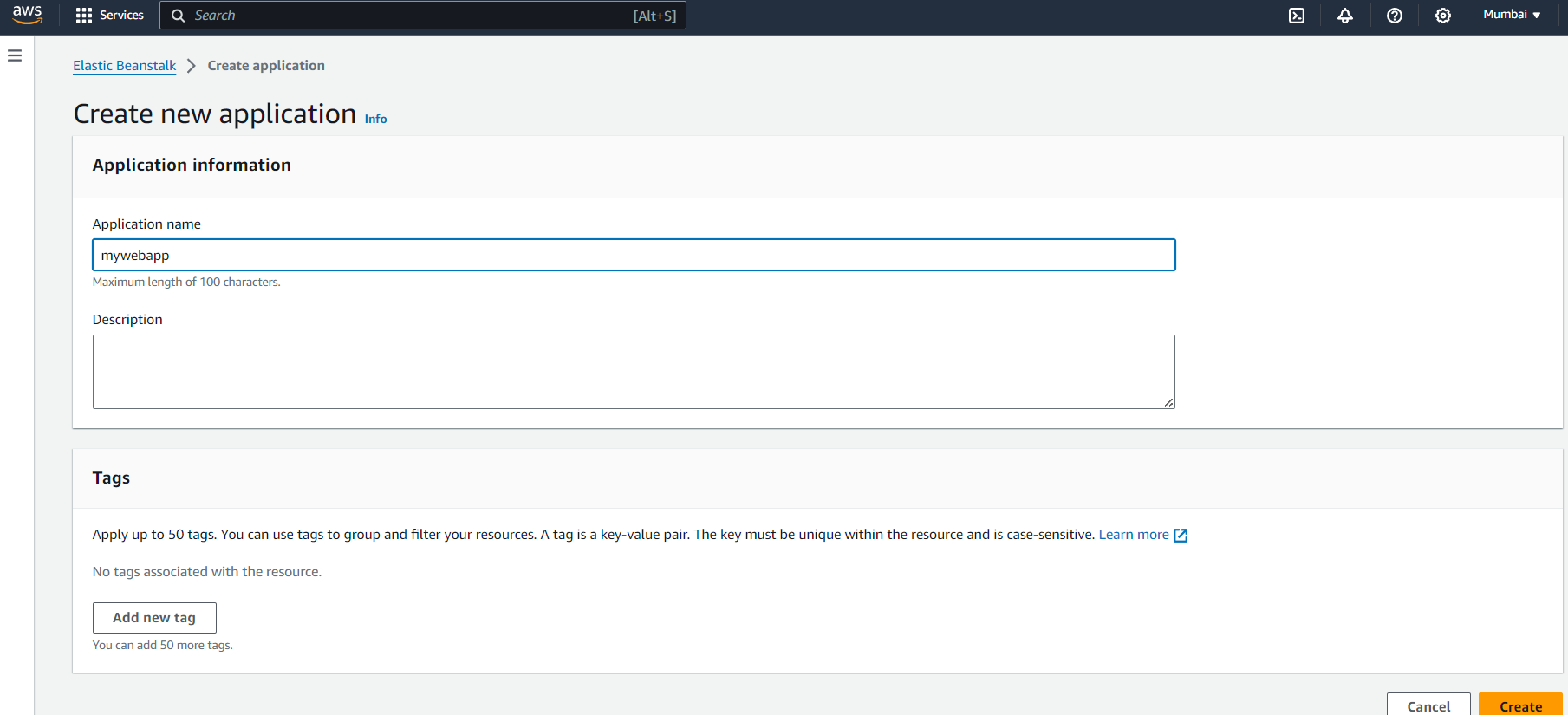
**Experiment No: 2**

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.

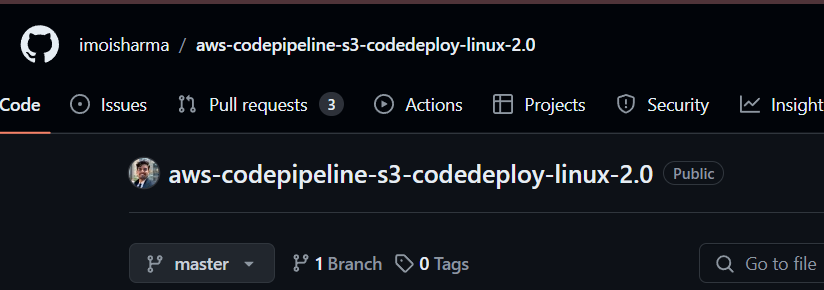
Your continuous deployment pipeline will need a target environment containing virtual servers, or Amazon EC2 instances, where it will deploy sample code. You will prepare this environment before creating the pipeline.

1. Open up Elastic Beanstalk and name your web app.



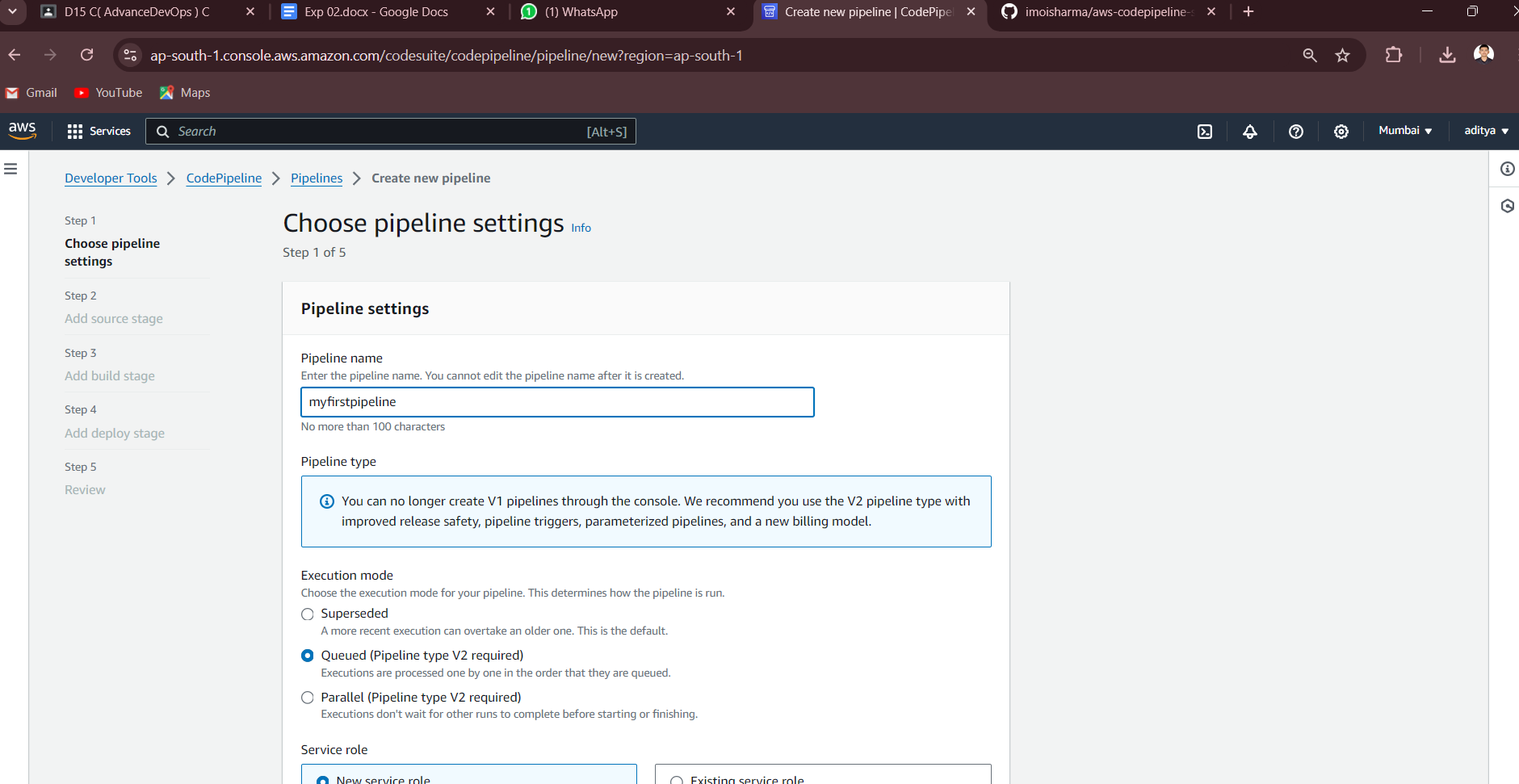
4. Beanstalk creates a sample environment for you to deploy your application. By default, it creates an EC2 instance, a security group, an Auto Scaling group, an Amazon S3 Bucket, Amazon CloudWatch alarms and a domain name for your Application.

**Step 2: Get a copy of your sample code**

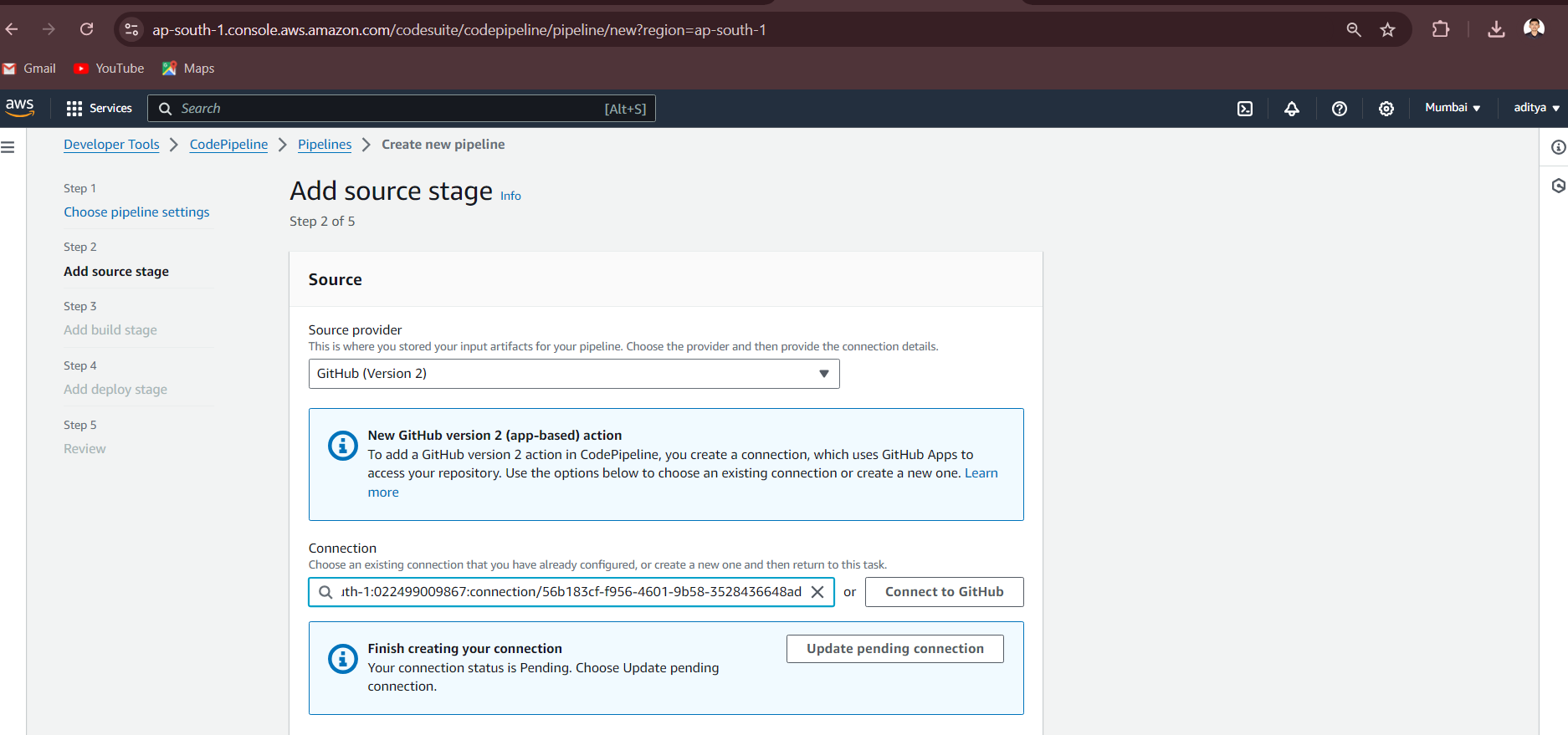
****In this step, we will get the sample code from this GitHub Repository to later host it. The pipeline takes code from the source and then performs actions on it.

**Step 3: Creating a CodePipeline**

1. Go to AWS Developer Tools -> CodePipeline and create a new Pipeline. Fill in the initial settings first.

2. In the source stage, choose GitHub v2 as the provider, then connect

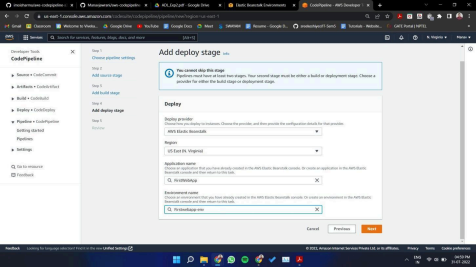
your GitHub account to AWS by creating a connection. You’d need your GitHub credentials and then you’d need to authorize and install AWS on the forked GitHub Repository.



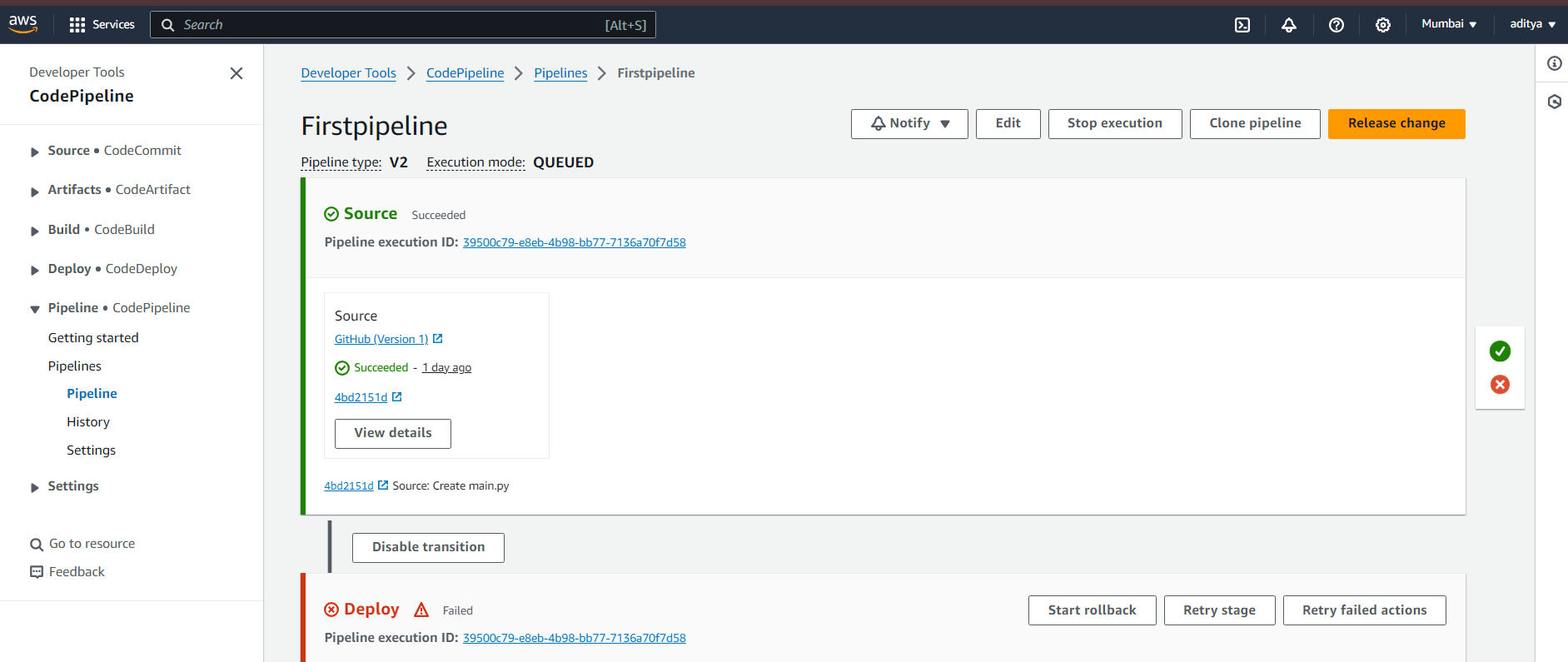
3. Then, simply choose this forked repository and the branch which you will be able to find in the search box. After that, click Continue and skip the build stage. Proceed to the Deployment stage.

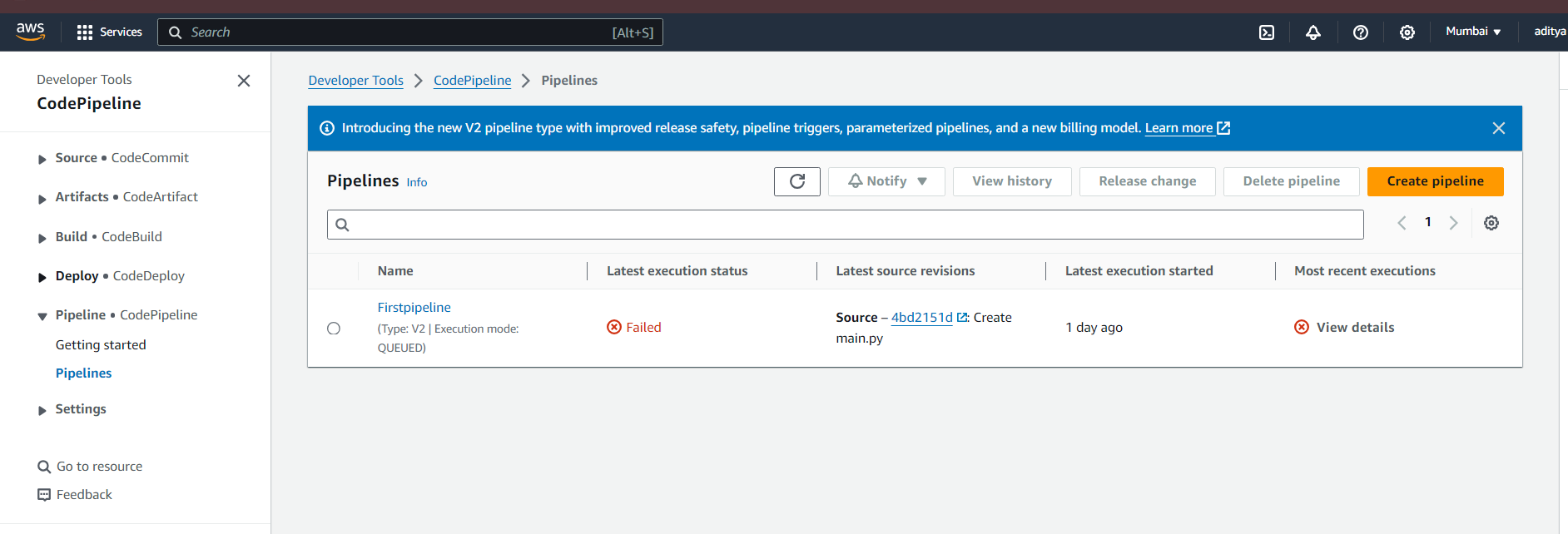
**Step 4: Deployment**

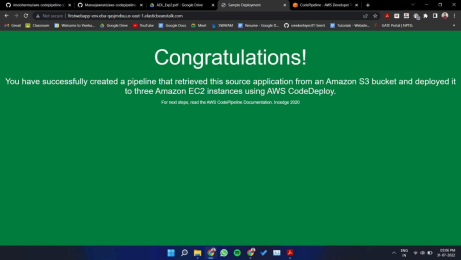
1. Choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name. Click Next, Review and create the pipeline.



2. Review all the settings and click on create pipeline



This is the sample website we just created.

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

**Step 5: Committing changes to update app**

1. In this step, we will update the code which we had and make a few changes to the HTML file (keep in mind, this is in our version of the forked repository).

2. In GitHub, open index.html. Then, make changes to either the heading tag or the paragraph tag. Commit these changes on the fly on GitHub. 3. When you commit these changes to your forked version, you’ll notice the changes being made in real-time on the Source panel.

4. You can view the changes on the website using the same URL, once the deployment section shows success.

