



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Semester: V

Academic Year: 2021-22

Class / Branch: TE IT

Subject: Advanced Devops Lab (ADL)

Name of Instructor: Prof. Manjusha K.

Name of Student: Chirag Jayesh Malde

Student ID: 22104186

EXPERIMENT NO. 02

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.

Step1: Create two S3 Buckets one for Storage and one for Deployment.

a. If you plan to use Amazon S3 as your source, you will retrieve the sample code from the AWS GitHub repository, save it to your computer, and upload it to an Amazon S3 bucket.

- Visit our GitHub repository containing the sample code at

<https://github.com/imoisharma/aws-codepipeline-s3-codedeploy-linux-2.0>

- Click the dist folder.

b. Save the source files to your computer:

- Click the file named aws-codepipeline-s3-aws-codedeploy_linux.zip
- Click View Raw.
- Save the sample file to your local computer.

c. open the Amazon S3 console and create your Amazon S3 bucket:

- Click Create Bucket

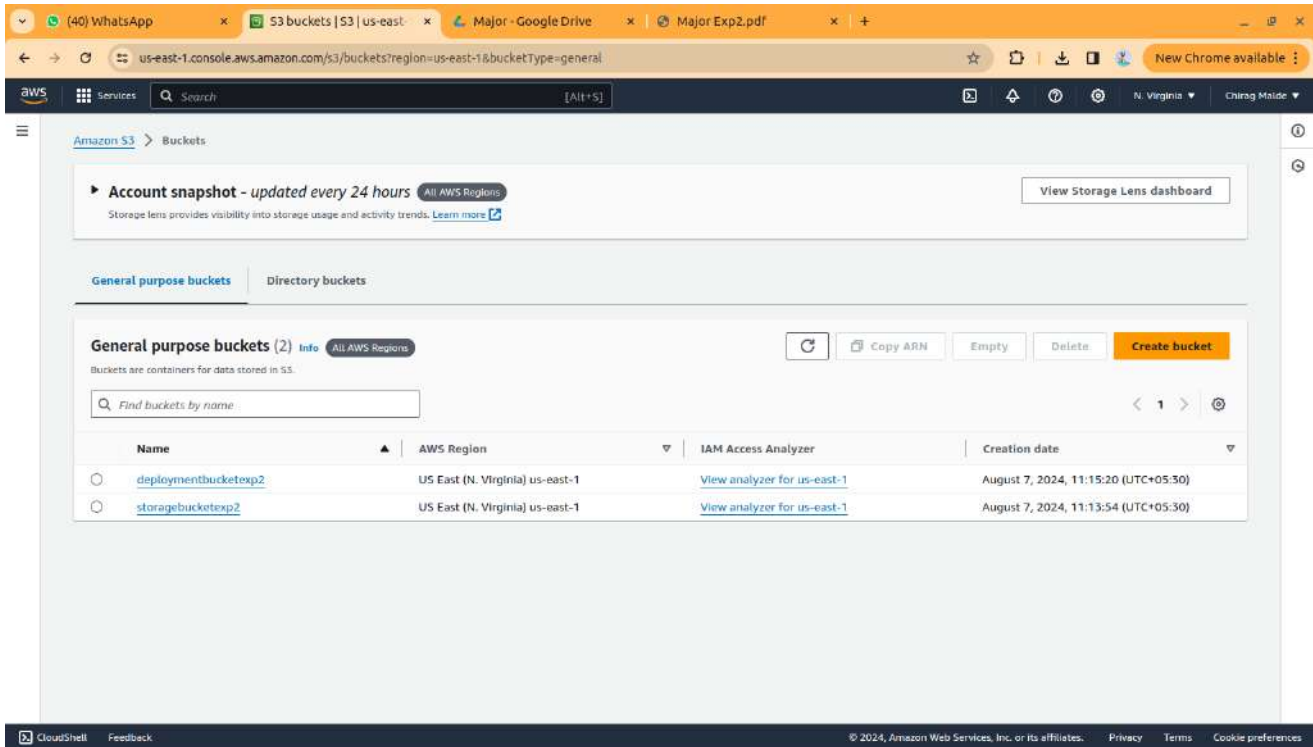
- Bucket Name: type a unique name for your bucket, such as awscodepipeline-demobucket-variables. All bucket names in Amazon S3 must be unique, so use one of your own, not one with the name shown in the example.
- Region: In the drop-down, select the region where you will create your pipeline, such as ap-South-1
- Click Create.

d. The console displays the newly created bucket, which is empty.

- Click Properties.
- Expand Versioning and select Enable Versioning. When versioning is enabled, Amazon S3 saves every version of every object in the bucket.

e. You will now upload the sample code to the Amazon S3 bucket:

- Click Upload.
- Follow the on-screen directions to upload the .zip file containing the sample code you downloaded from GitHub.

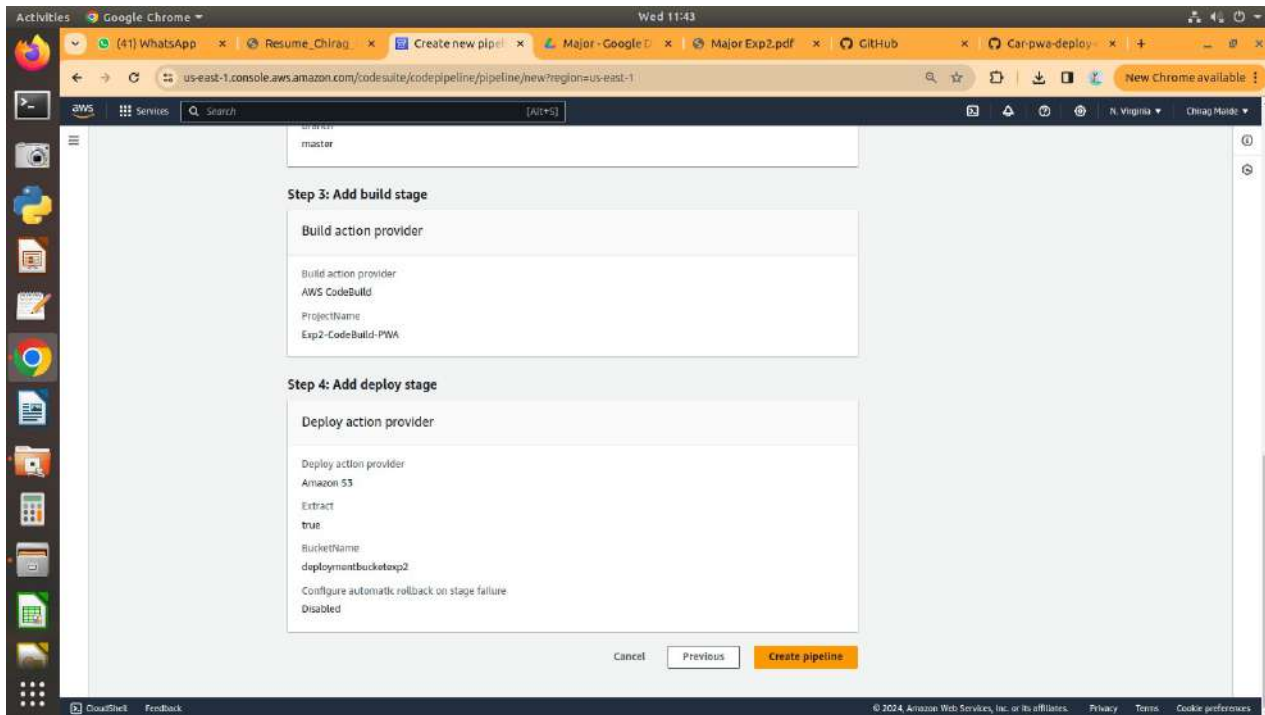


Step2: Create a Pipeline

In this step, you will retrieve a copy of the sample app's code and choose a source to host the code.

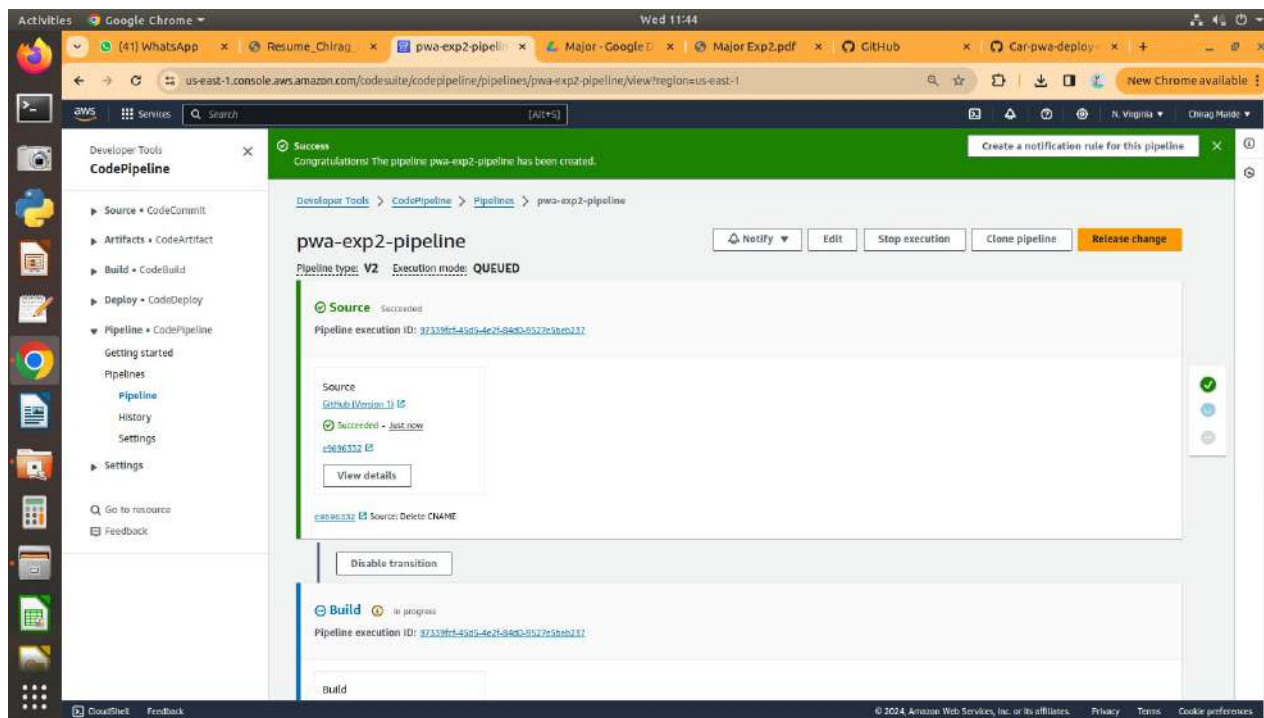
The pipeline takes code from the source and then performs actions on it.

You can use one of three options as your source: a GitHub repository, an Amazon S3 bucket, or an AWS CodeCommit repository. Select your preference and follow the steps below:



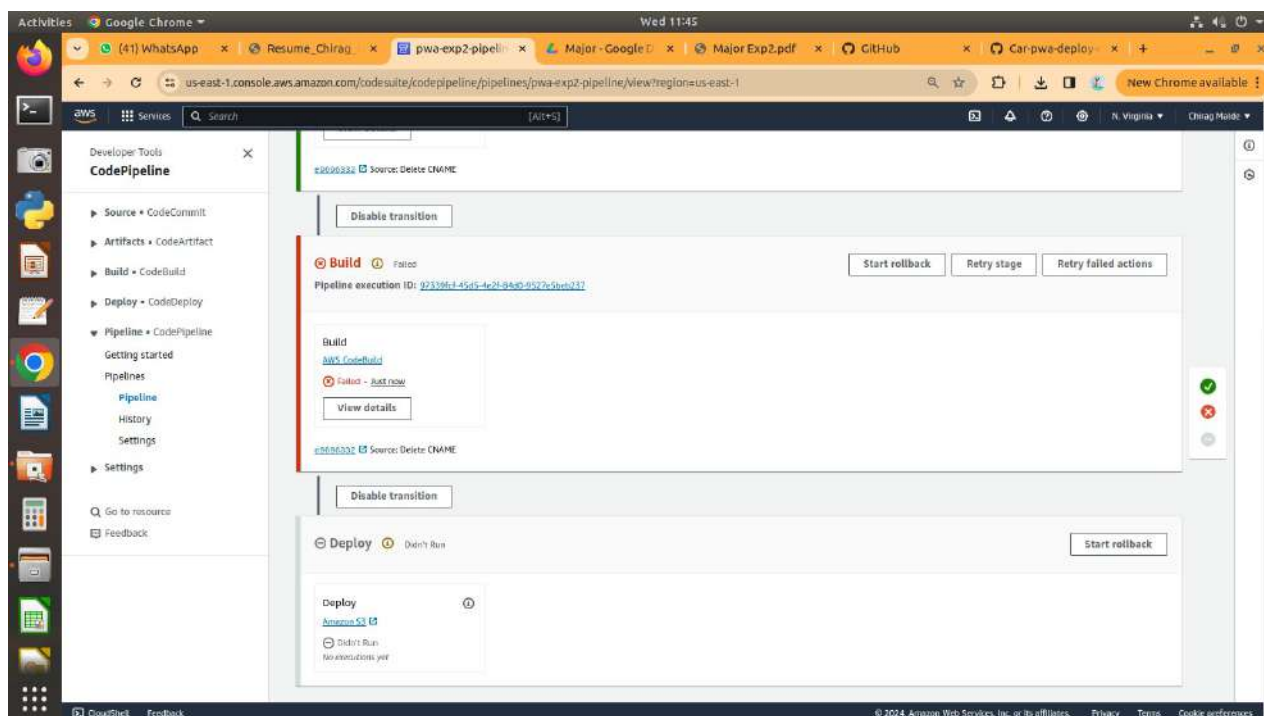
After your pipeline is created, the pipeline status page appears and the pipeline automatically starts to run. You can view progress as well as success and failure messages as the pipeline perform each action.

To verify your pipeline ran successfully, monitor the progress of the pipeline as it moves through each stage. The status of each stage will change from No executions yet to In Progress, and then to either Succeeded or Failed. The pipeline should complete the first run within a few minutes.

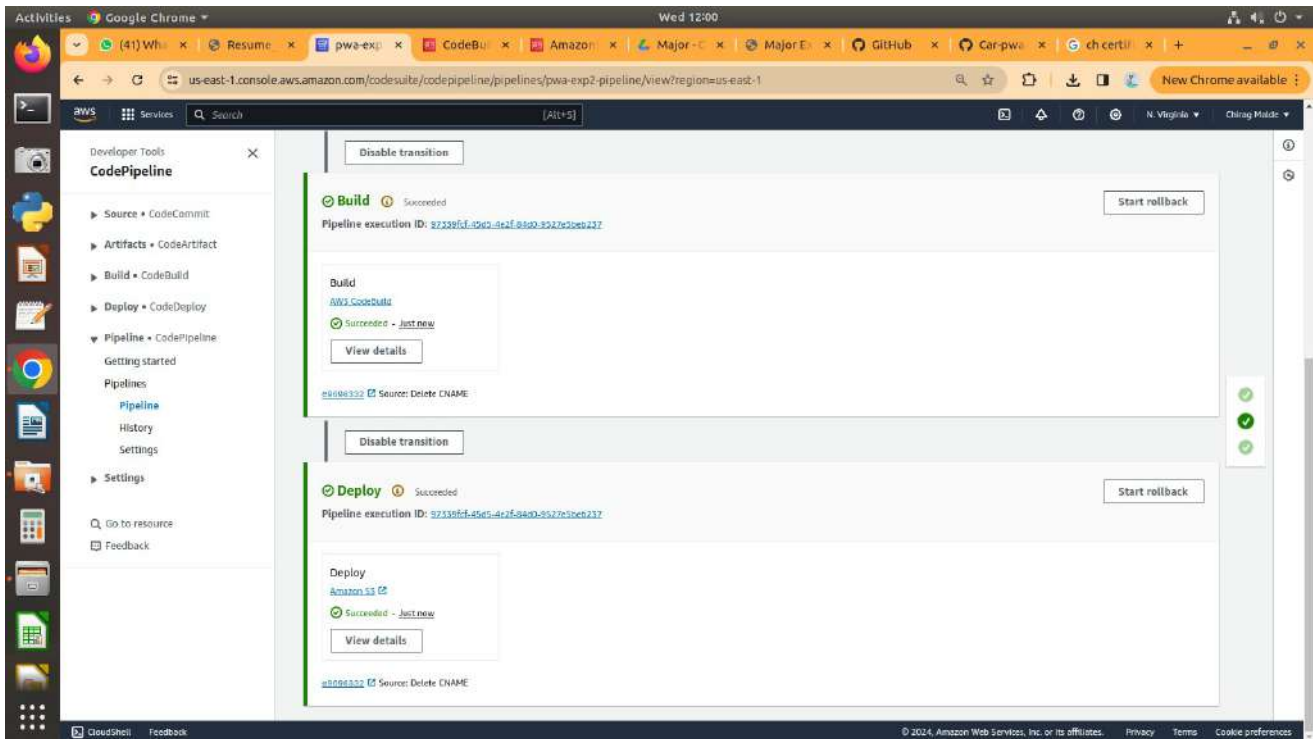


You have successfully created an automated software release pipeline using AWS CodePipeline!

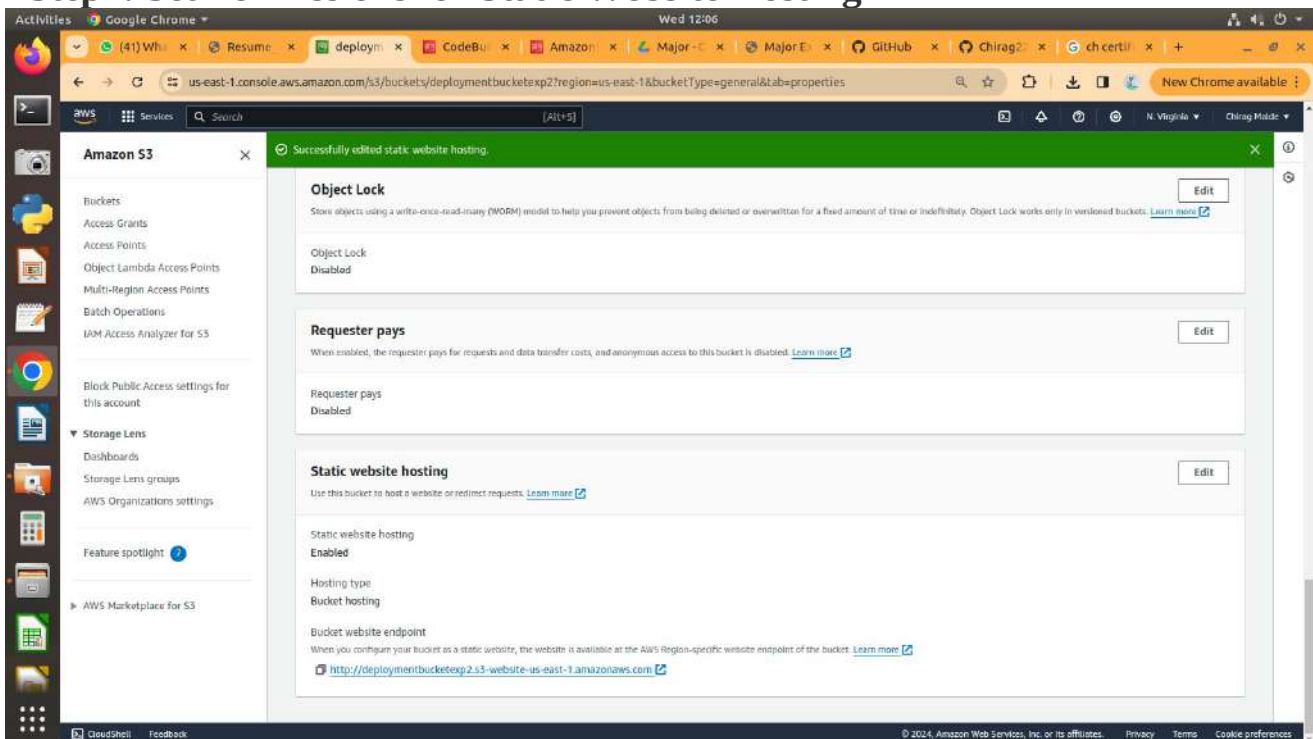
Using CodePipeline, you created a pipeline that uses GitHub, Amazon S3, or AWS CodeCommit as the source location for application code and then deploys the code to an Amazon EC2 instance

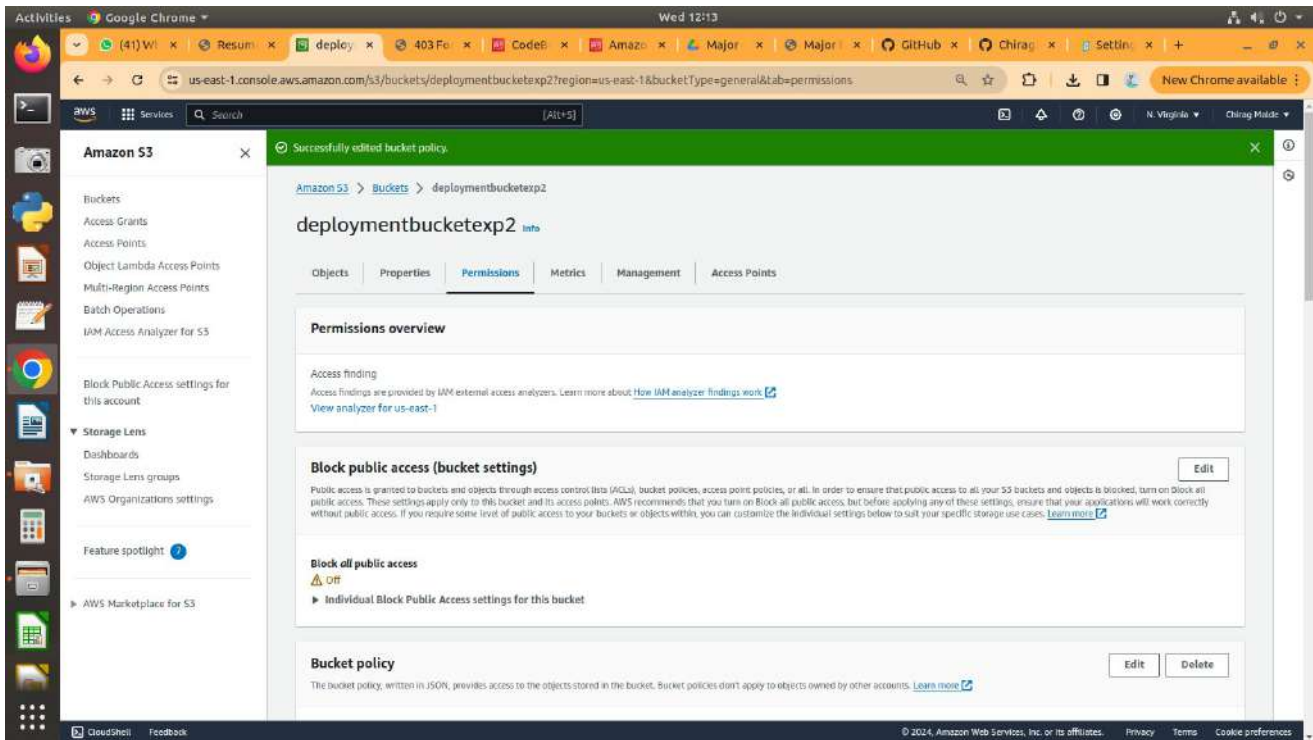


Step3: Change the properties of Build

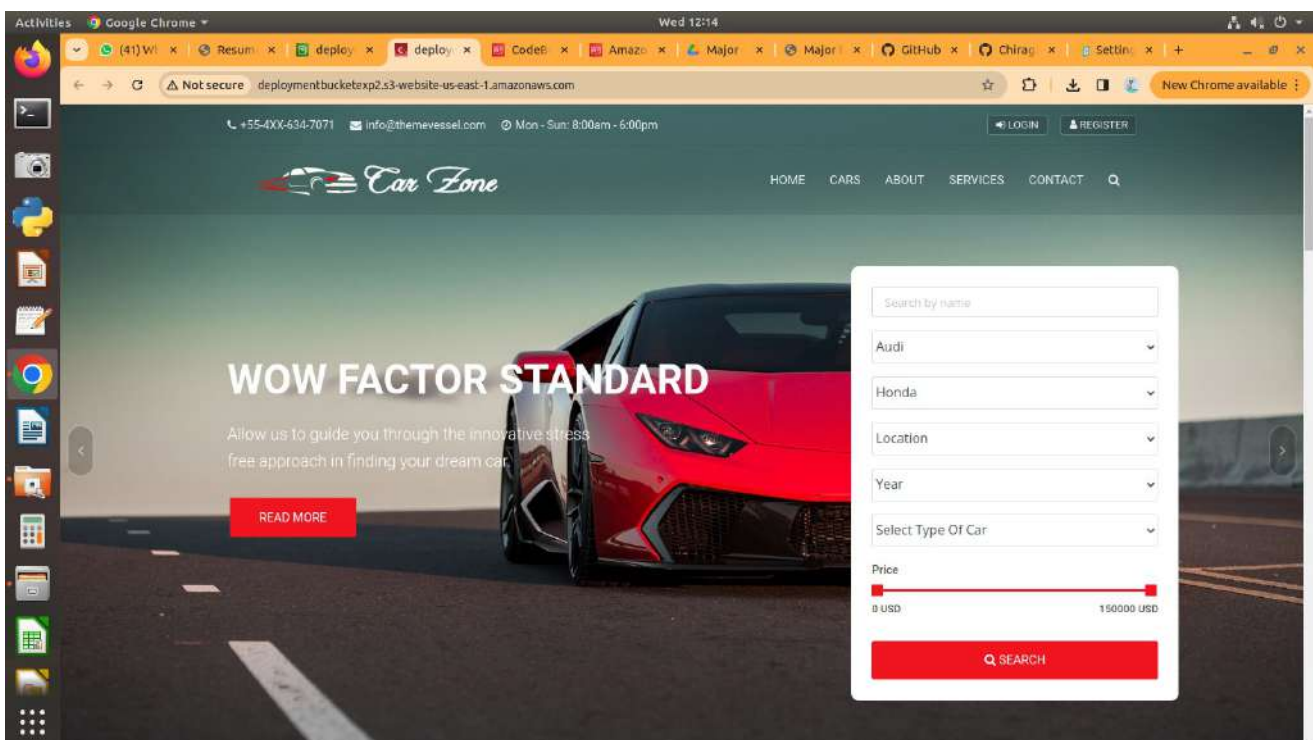


Step4: Set Permissions for Static Website Hosting





Step5: Check your Deployment



Now go to your EBS environment and click on the URL to view the sample website you deployed.

Step 6: Clean up your resources.

To avoid future charges, you will delete all the resources you launched throughout this tutorial, which includes the pipeline and the source you set up to host the code. You will delete your pipeline.

Conclusion: We successfully created two buckets one for storage and one for deployment using S3 buckets then we created a pipeline using CodeBuild then we connected all these and deployed our website by changing some settings.