Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

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.

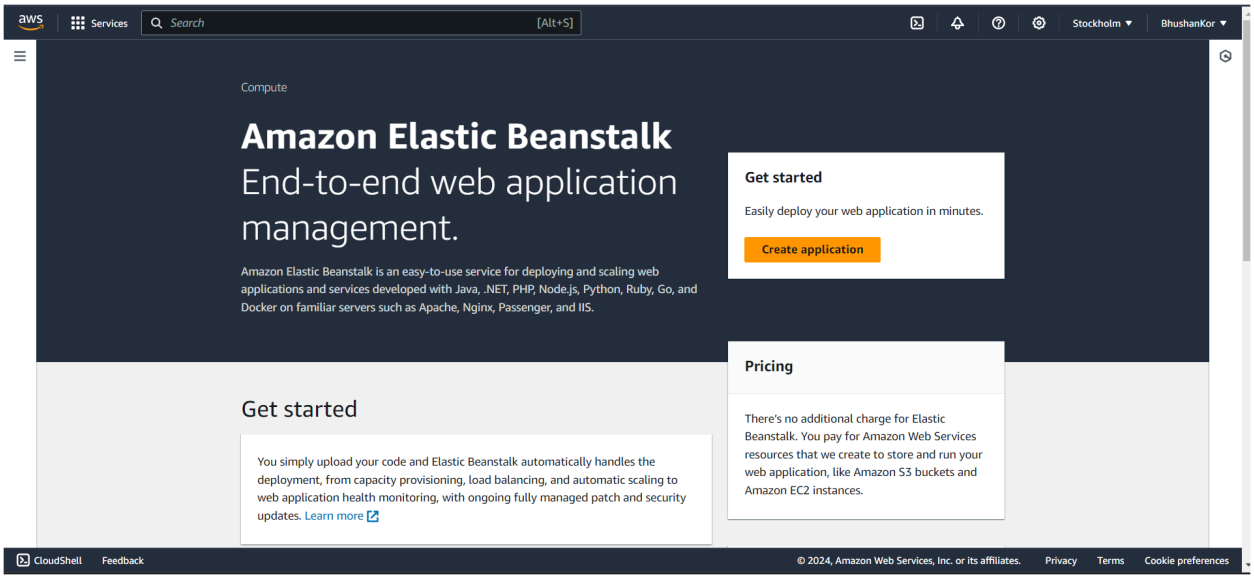
Elastic Beanstalk:
Amazon Elastic Beanstalk is a fully managed AWS service that streamlines the deployment and scaling of web applications. Developers simply upload their code, and Elastic Beanstalk automatically handles tasks like resource provisioning, load balancing, auto-scaling, and application health monitoring. It supports multiple programming languages and frameworks, offering flexibility and control over the underlying AWS resources if needed. This enables rapid application deployment without the complexities of managing infrastructure.

CodePipeline:
A CodePipeline is an AWS service that automates the build, test, and deploy phases of your release process. It enables continuous integration and continuous delivery (CI/CD) by defining the workflow of code changes from source to production. CodePipeline integrates with various AWS services and third-party tools to streamline and accelerate the release of new features.

Prerequisites: Before you start go to the end of this document and do the steps of New learning because it is necessary for step 5 and codePipeline.

A)Elastic Beanstalk

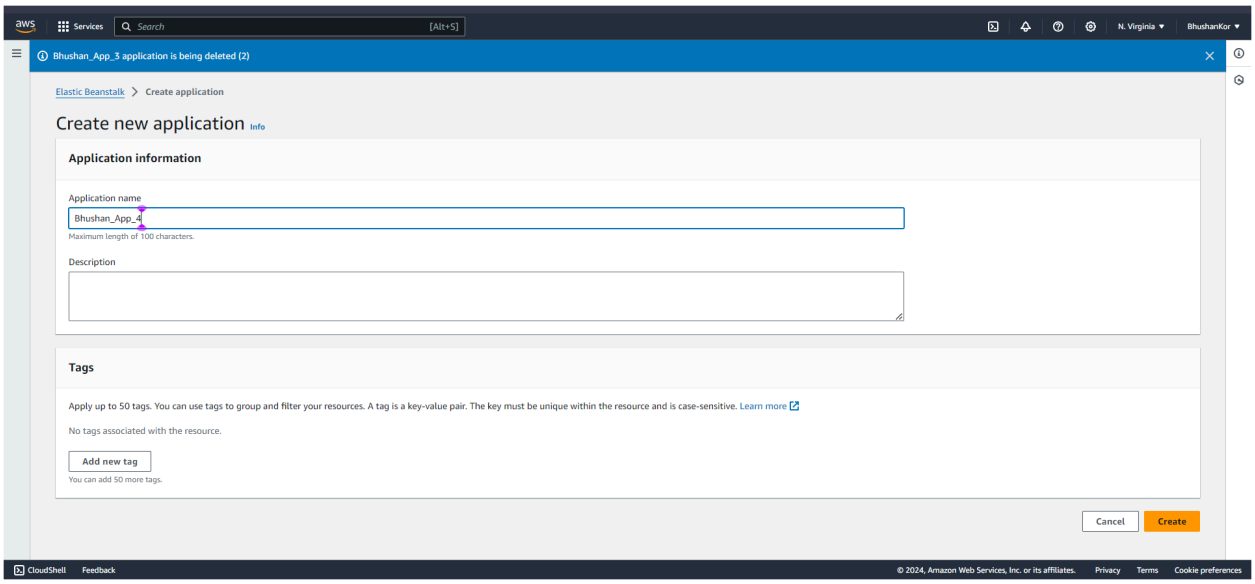
Step 1: Open Your **personal AWS** Account Because Elastic Beanstalk can be created in AWS Academy but connecting GitHub with the code pipeline requires an IAM policy that is not present in the AWS Academy account.
In services search for Elastic Beanstalk and click on it.



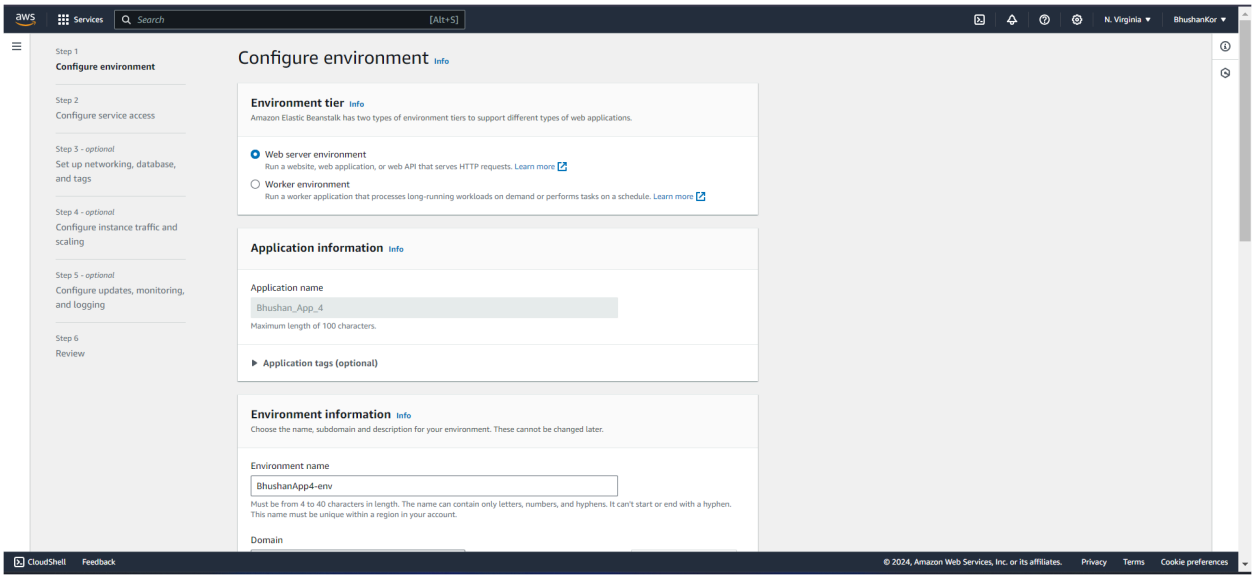
Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

Step 2: Click on Create a new application and give a name to your application.



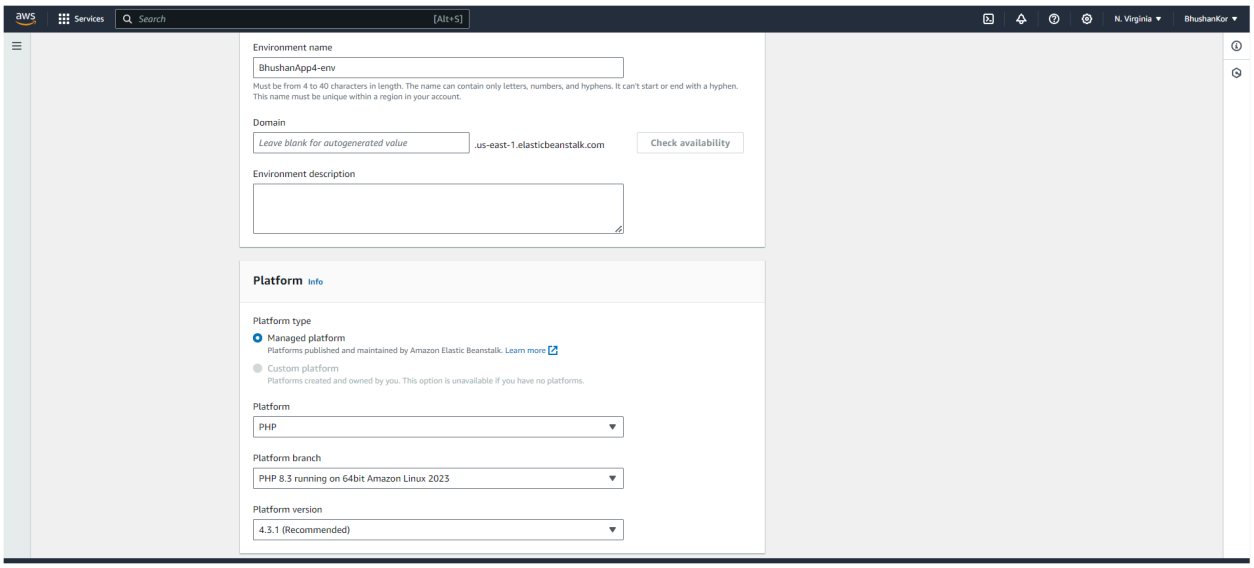
Step 3: After creation of the application click on Create Environment and select Web Server Environment.

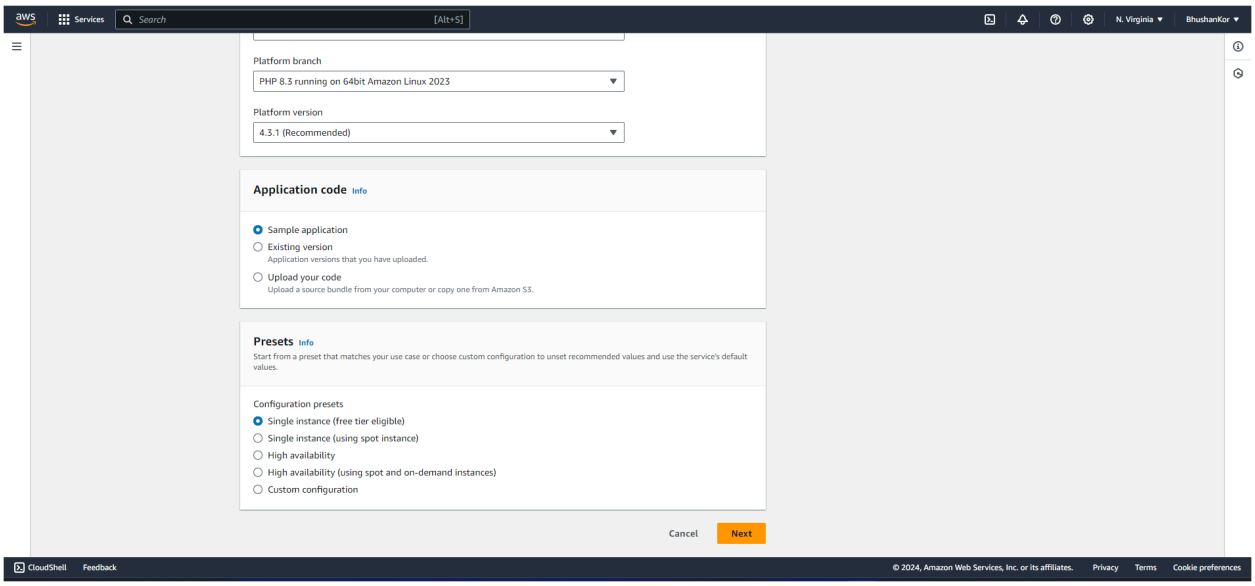


Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

Step 4: Select the platform as PHP and application code to sample or if you have you can upload and keep others to default.



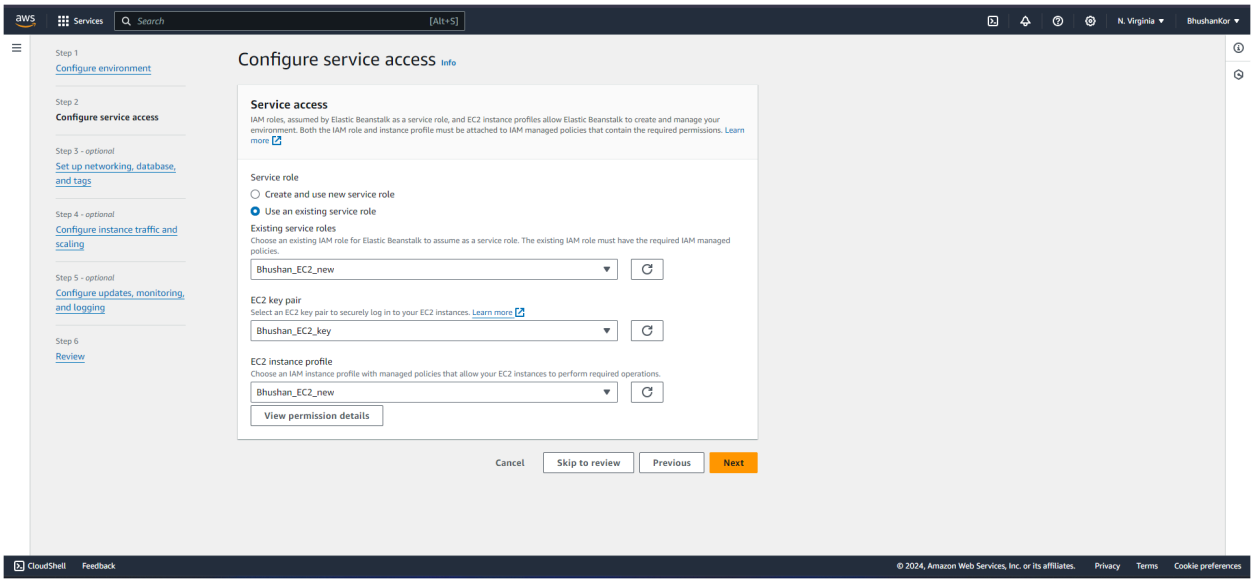


Name:Bhushan Mukund Kor **Academic Year:2024-2025**

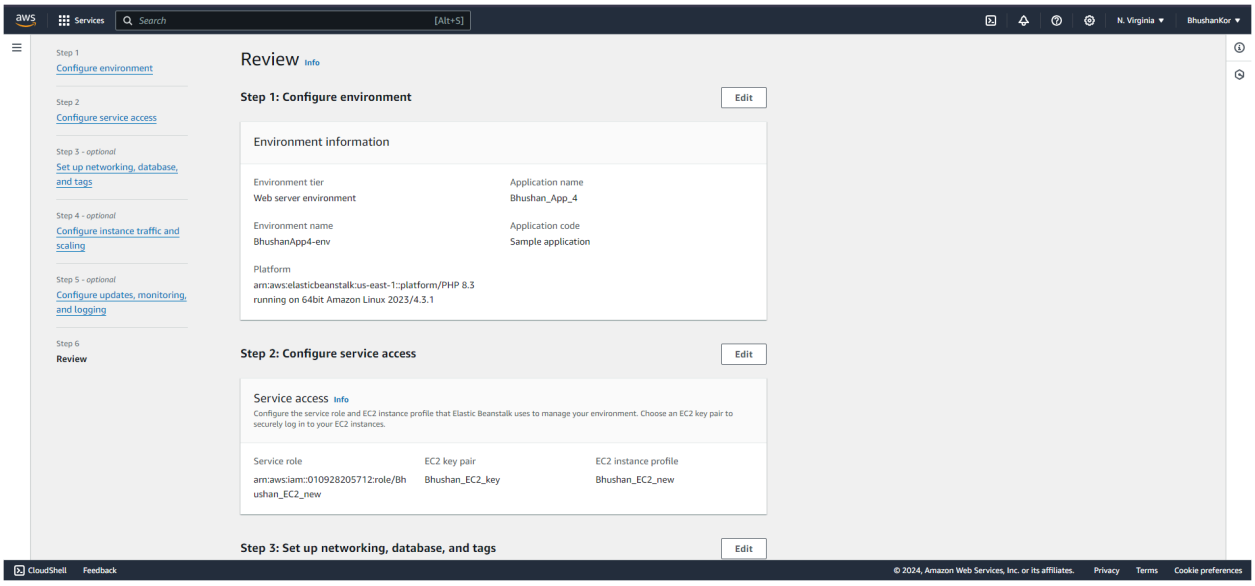
Division: D15C **Roll No: 28**

Step 5: Select the use an existing service role option because create and use new service role might not work according to the new policy of AWS (Refer to the New learnings of this document).

Select the existing roles and instances and keys.

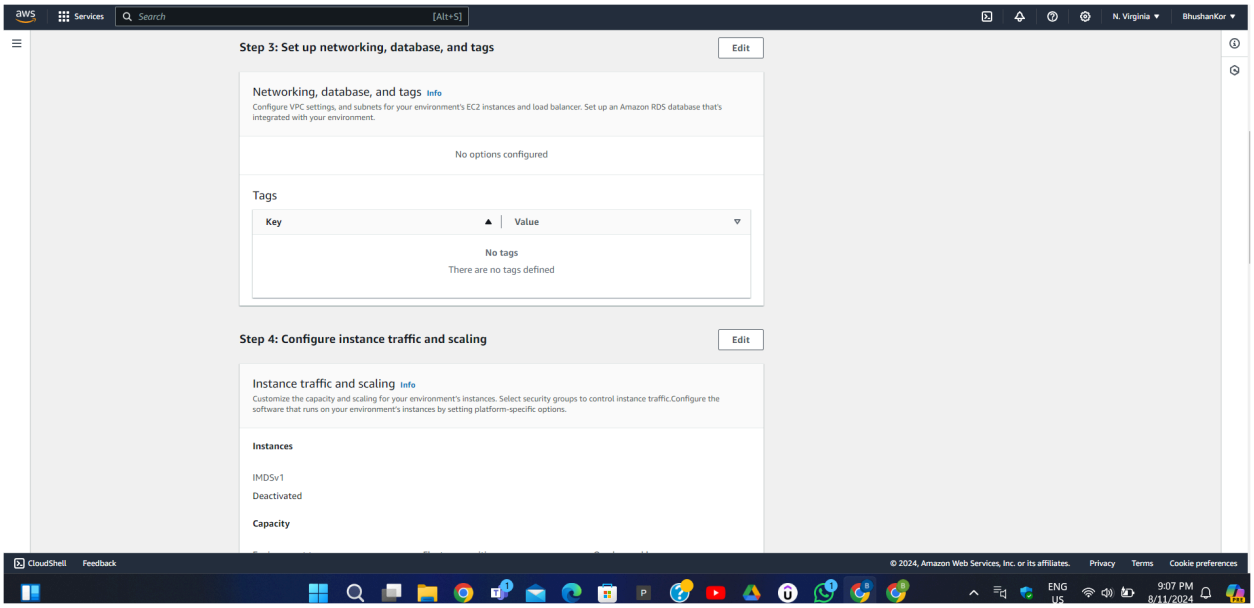


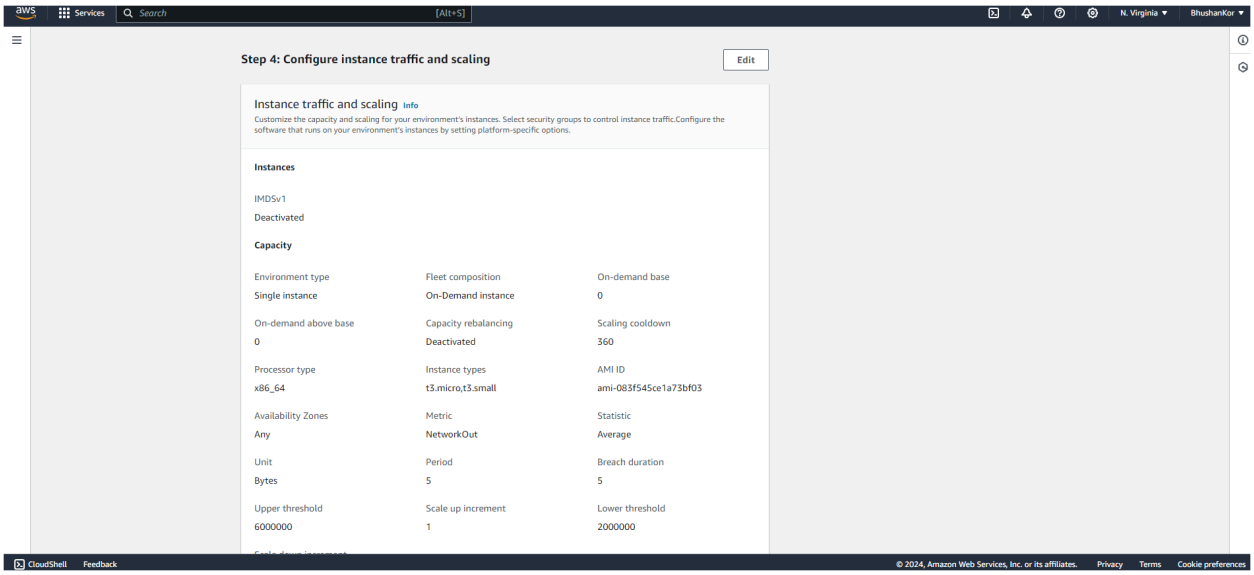
Step 6: Review and click on submit.

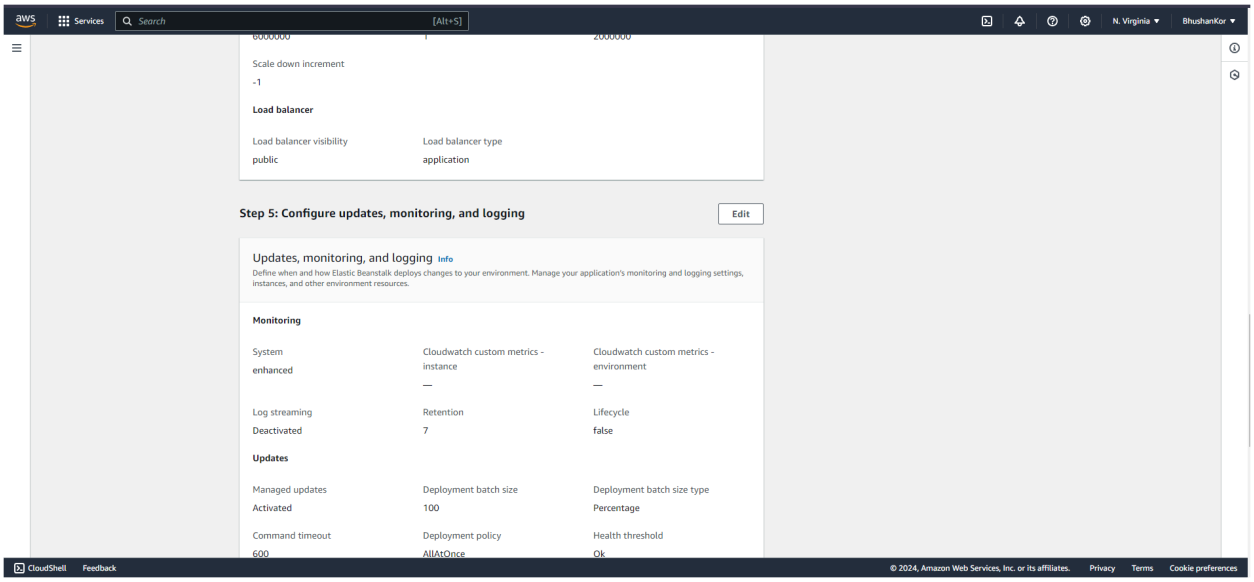


Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

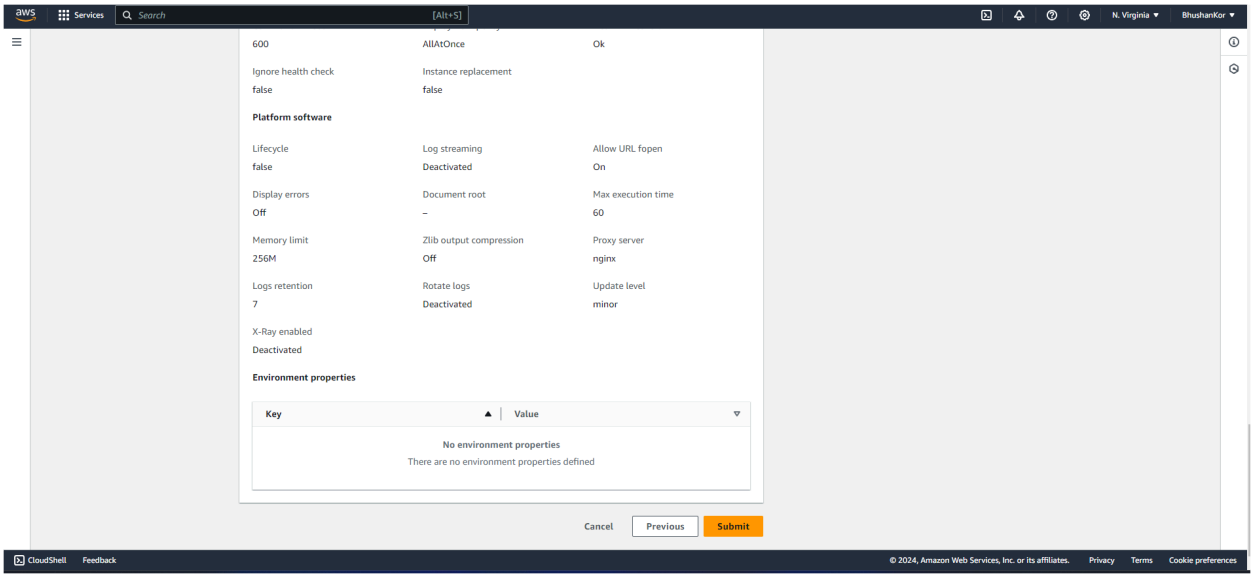




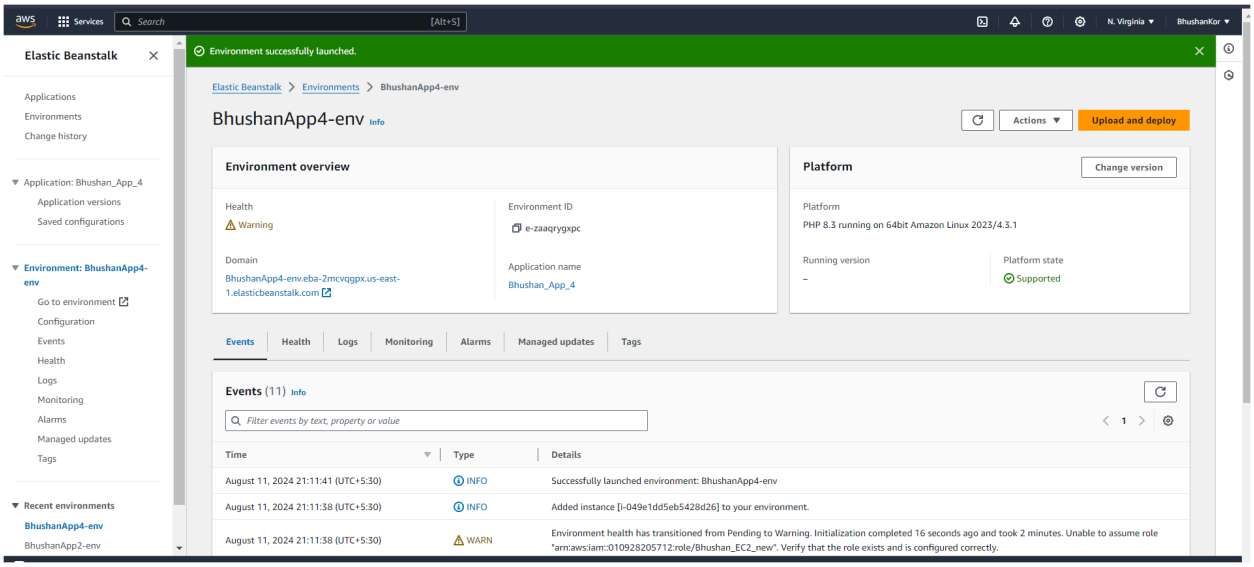


Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

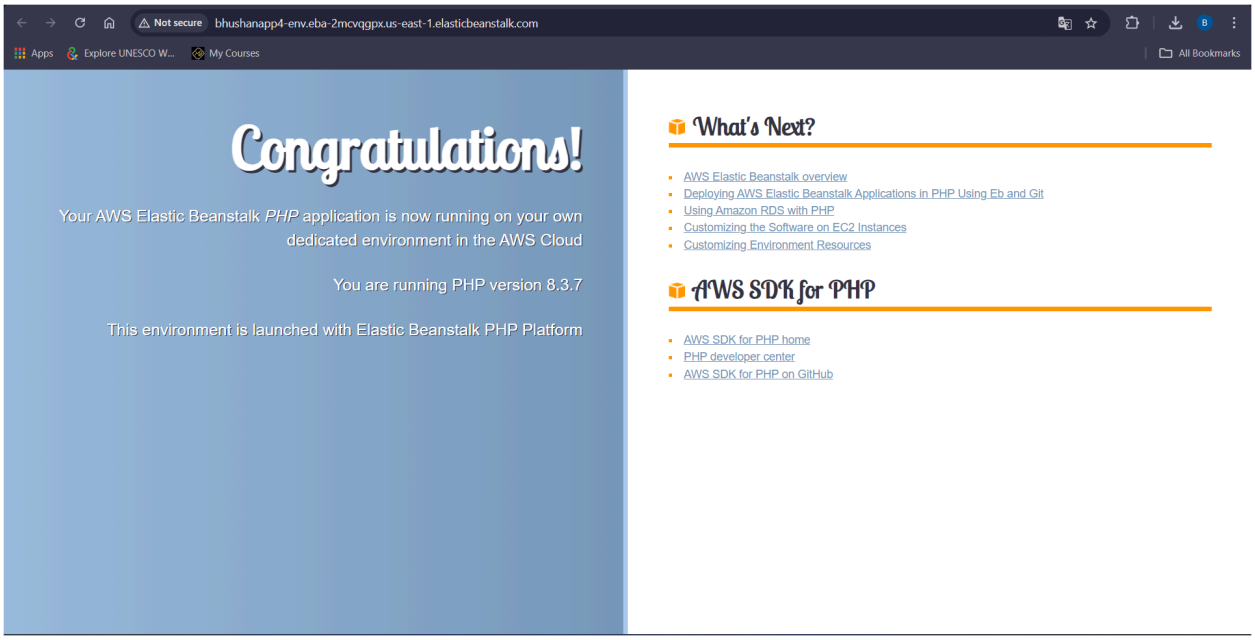


Step 7: Done Elastic BeanStalk environment is created successfully, click on the link to view the preview of your sample code or uploaded code.



Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

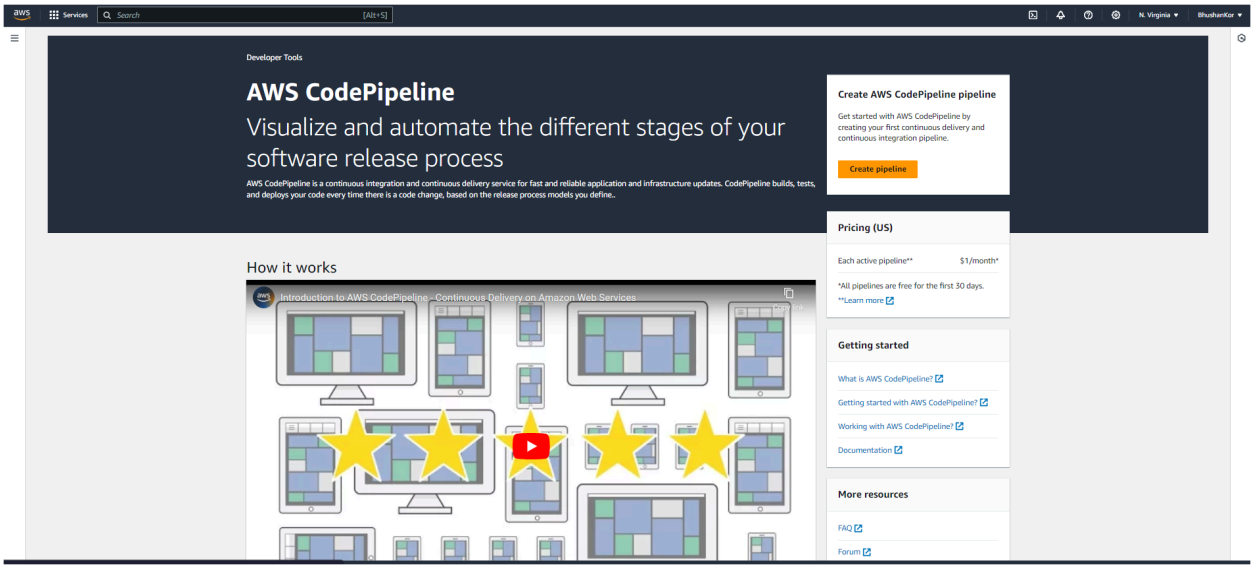


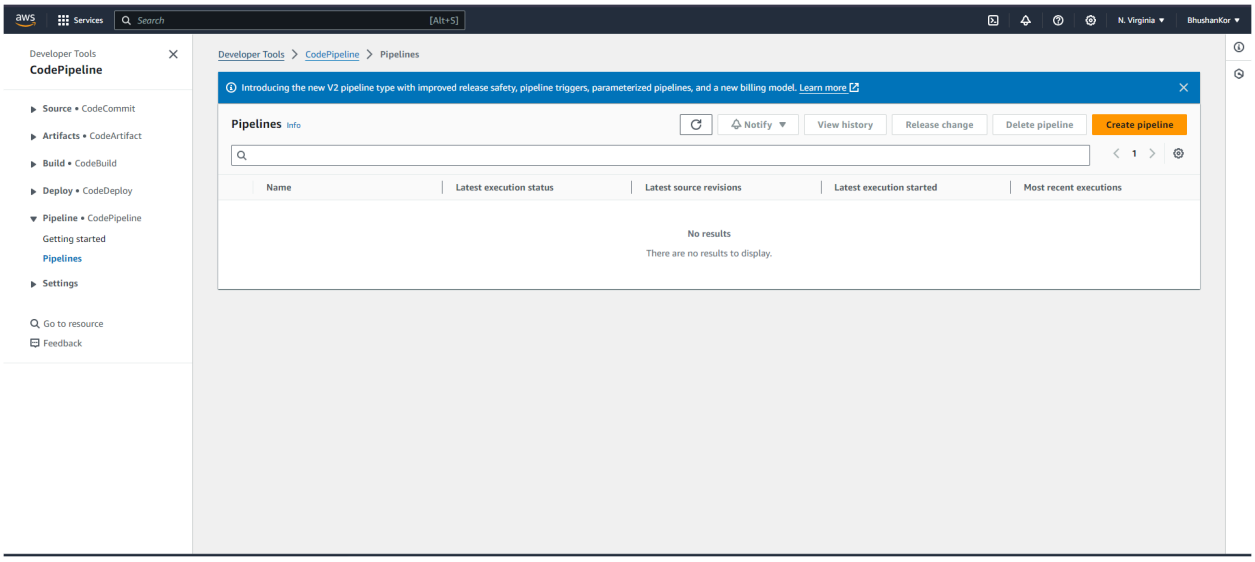
Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

B)CodePipline

Step 1: Search for codePipeline in services and click on create pipeline.

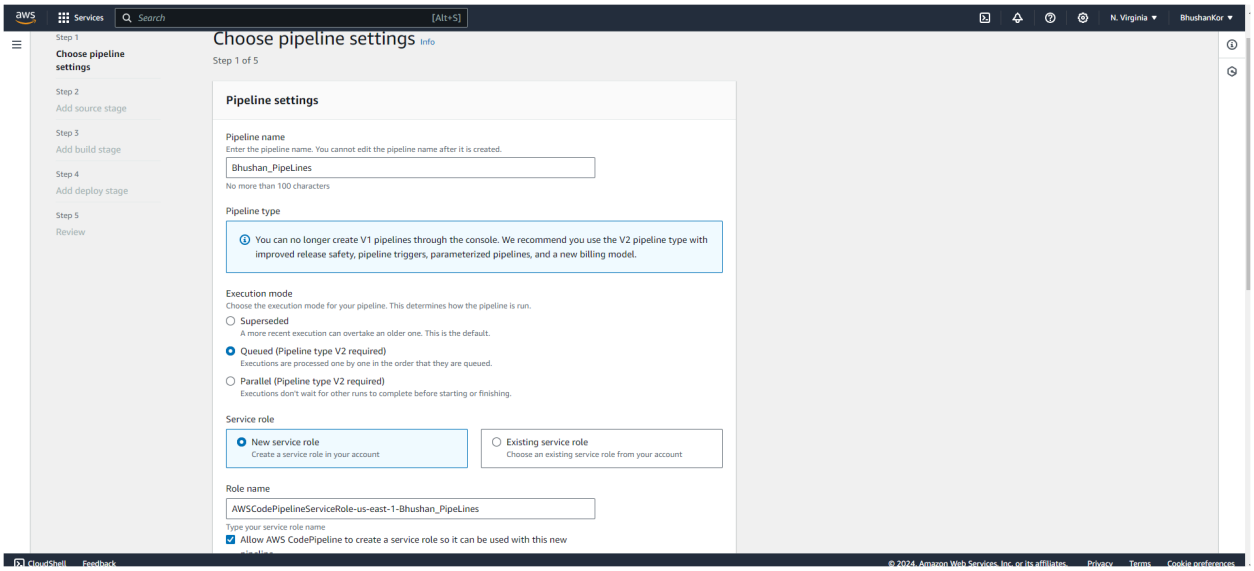


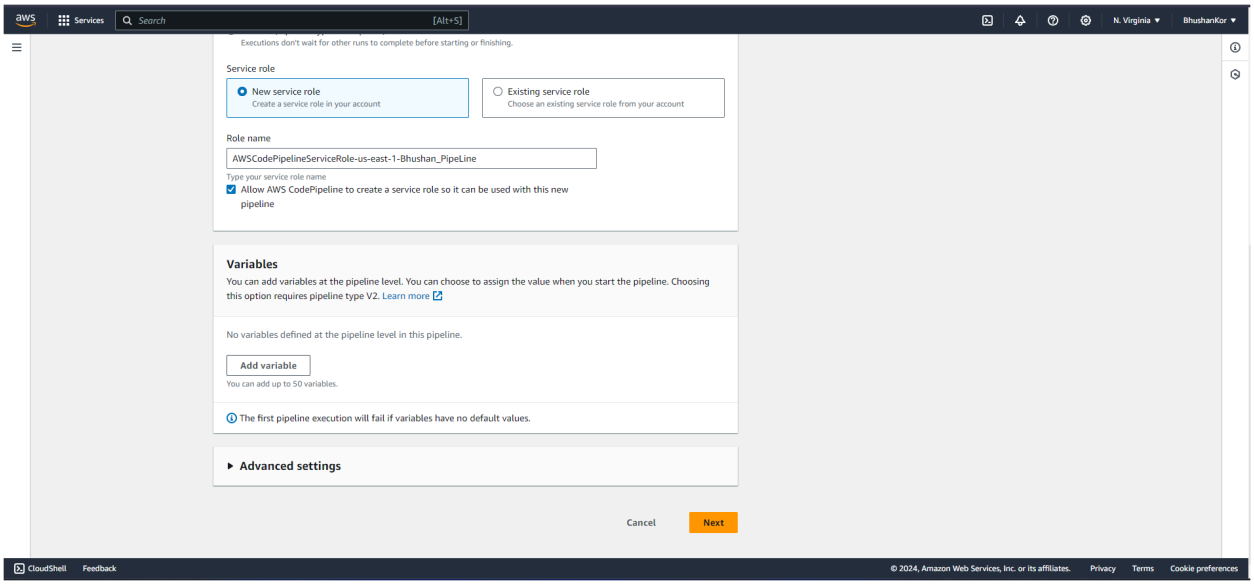


Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

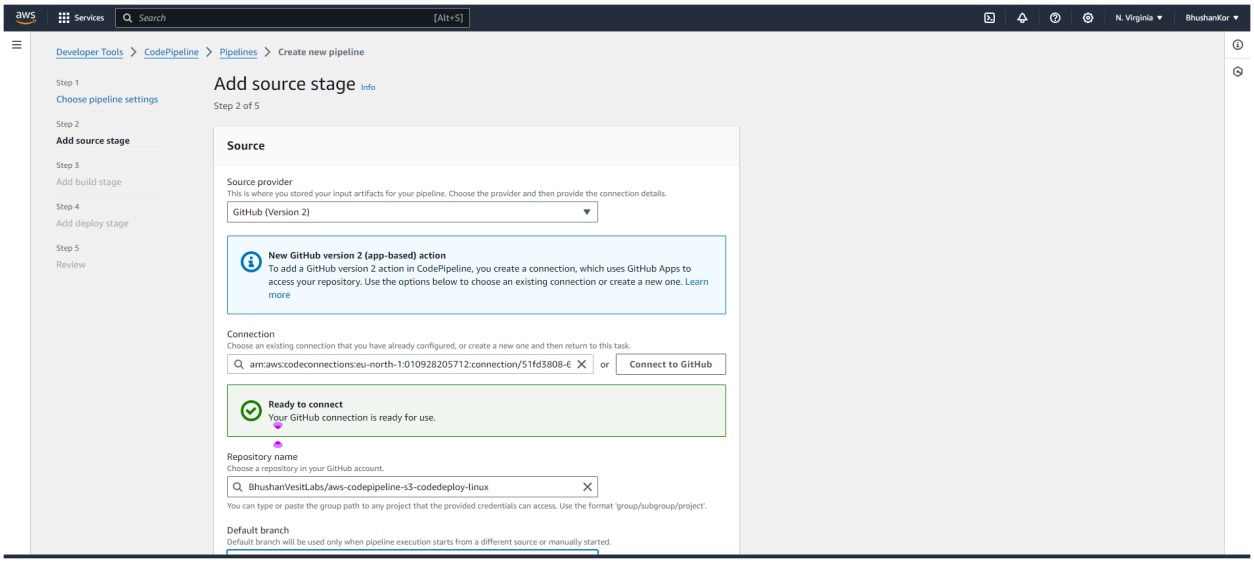
Step 2: Give the name to the pipeline, select a new service role, and keep the rest all to default.



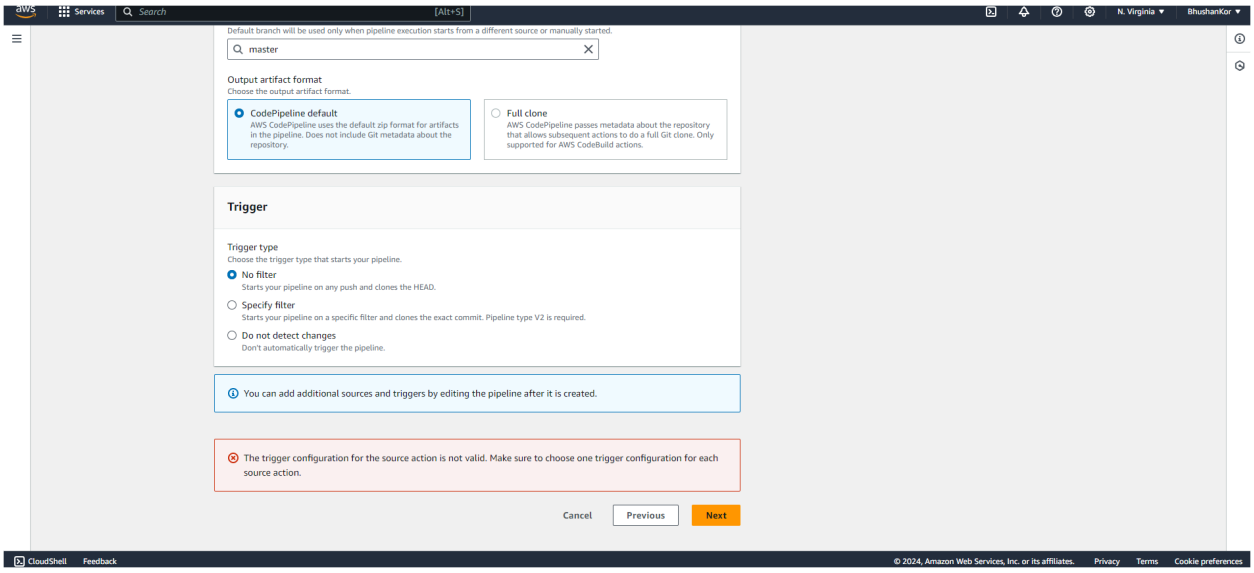


Before Step 3 Fork or clone the below repository to your GitHub account.
https://github.com/aws-samples/aws-codepipeline-s3-codedeploy-linux.git

Step 3: Now in the Add source stage Select GitHub (Version 2) [This will work only in personal AWS academy], then connect your GitHub account and select repository name, branch to main/master.



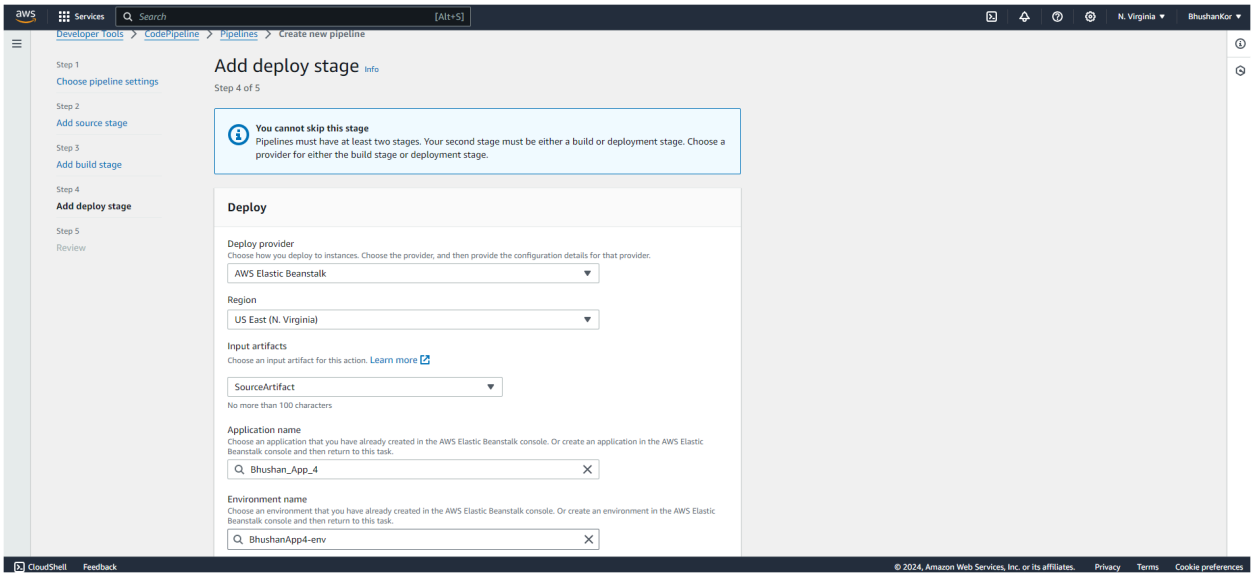
Step 4: Select Trigger to No filter otherwise it will give an error.

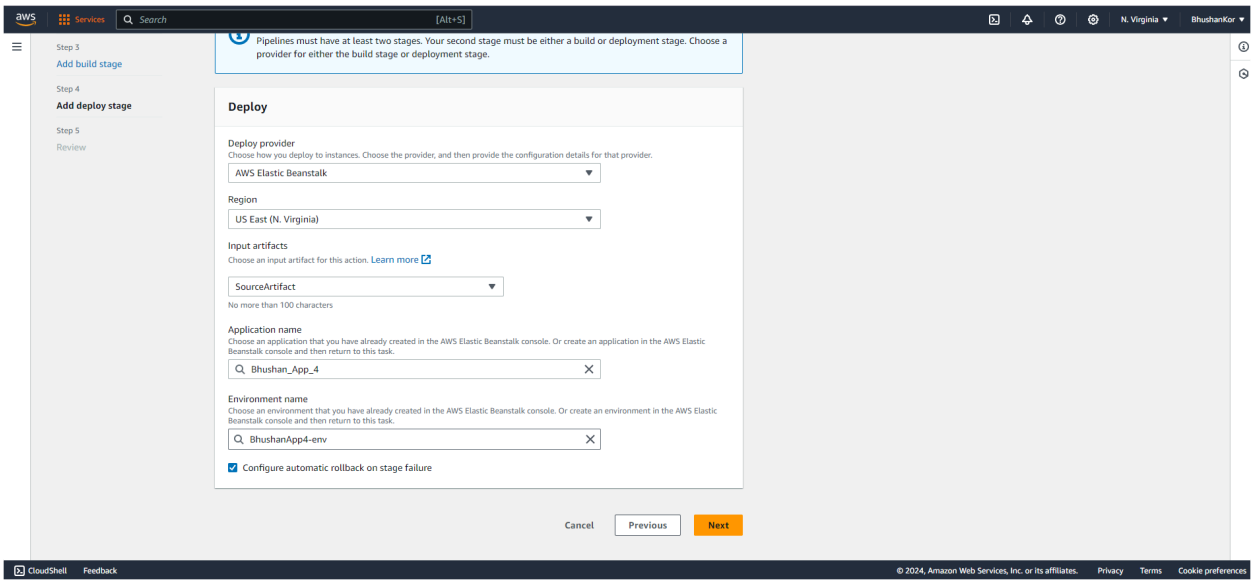


Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

Step 5: In the Depoly stage select AWS ElasticBeanstalk as the deploy provider, Select a region to US East (N.Virginia) or any with N.Virginia. Input artifacts to default and the Application name and environment name of your Elastic Beanstalk.

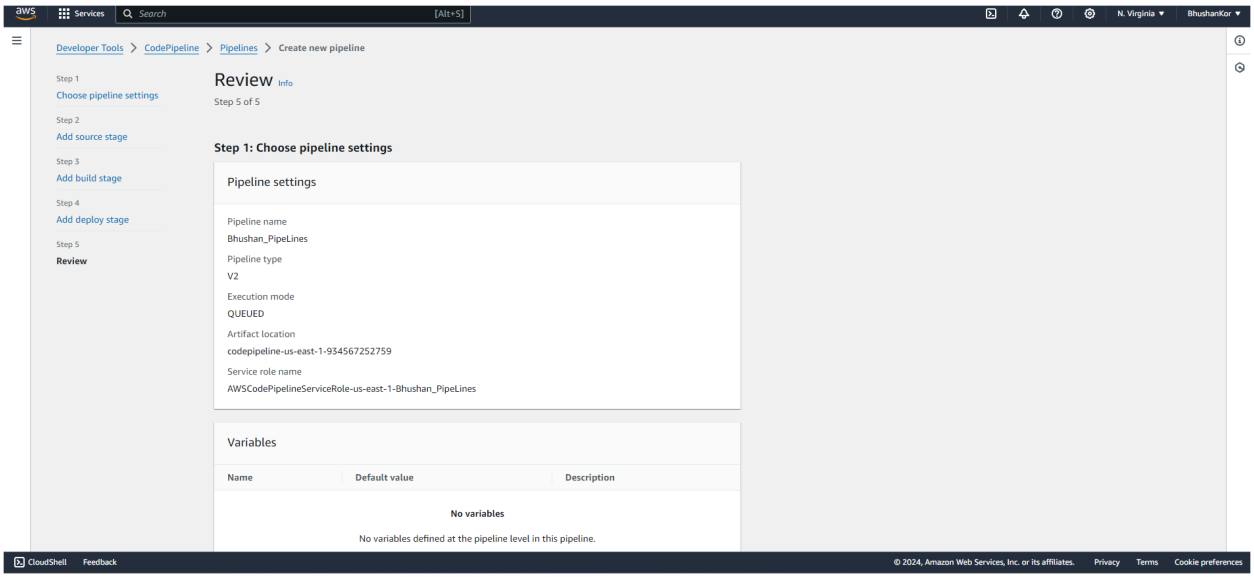


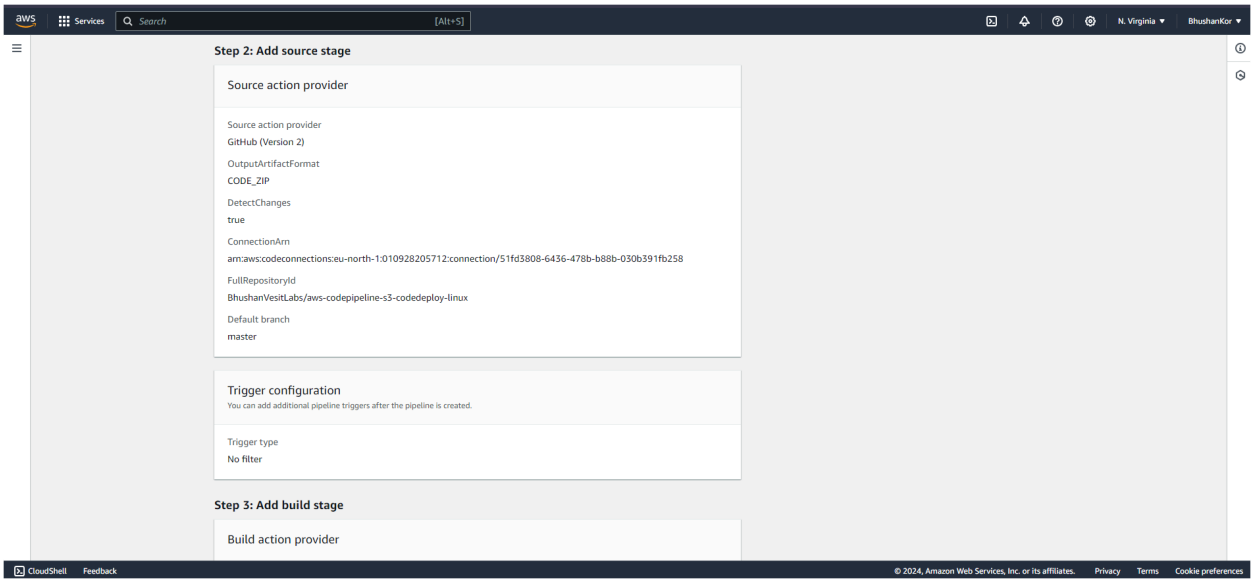


Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

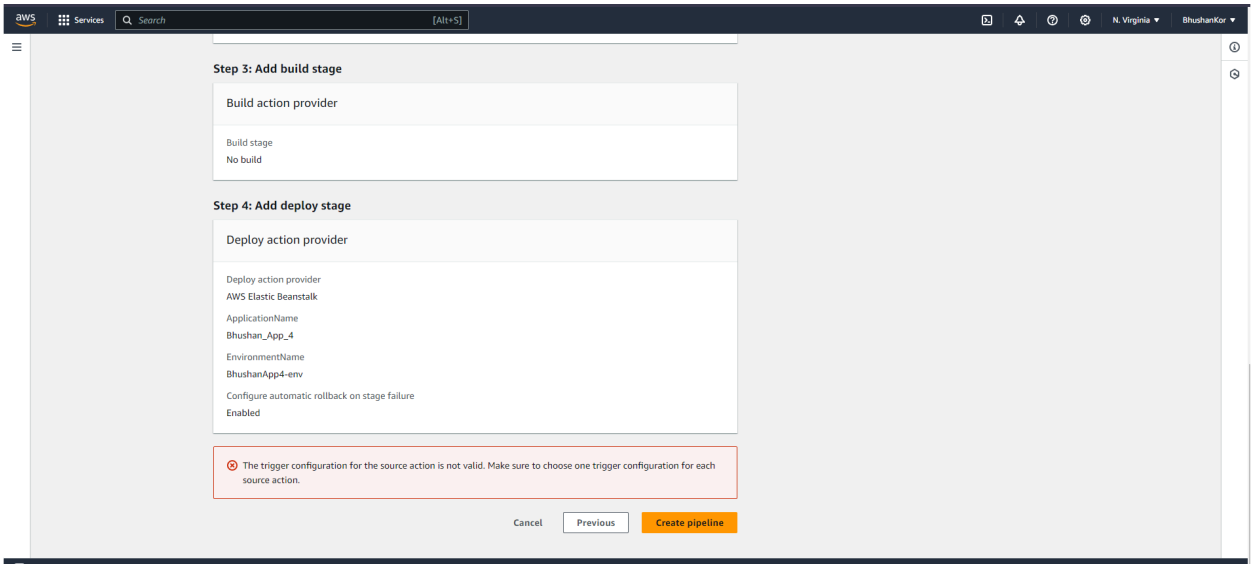
Step 6: Review and Click on Create Pipeline.



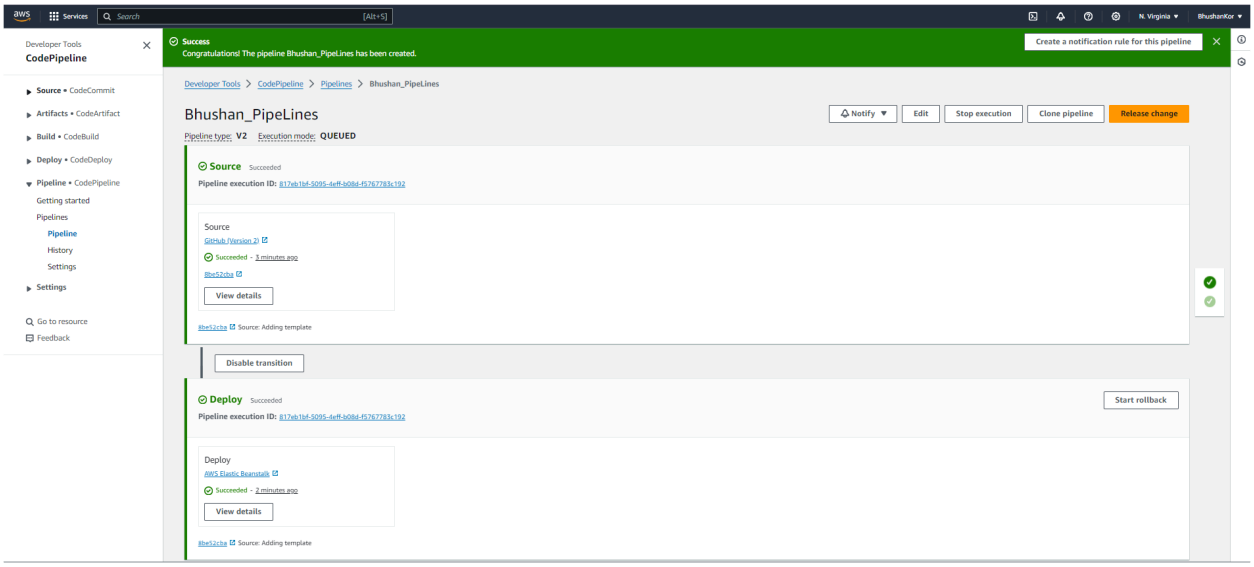


Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**



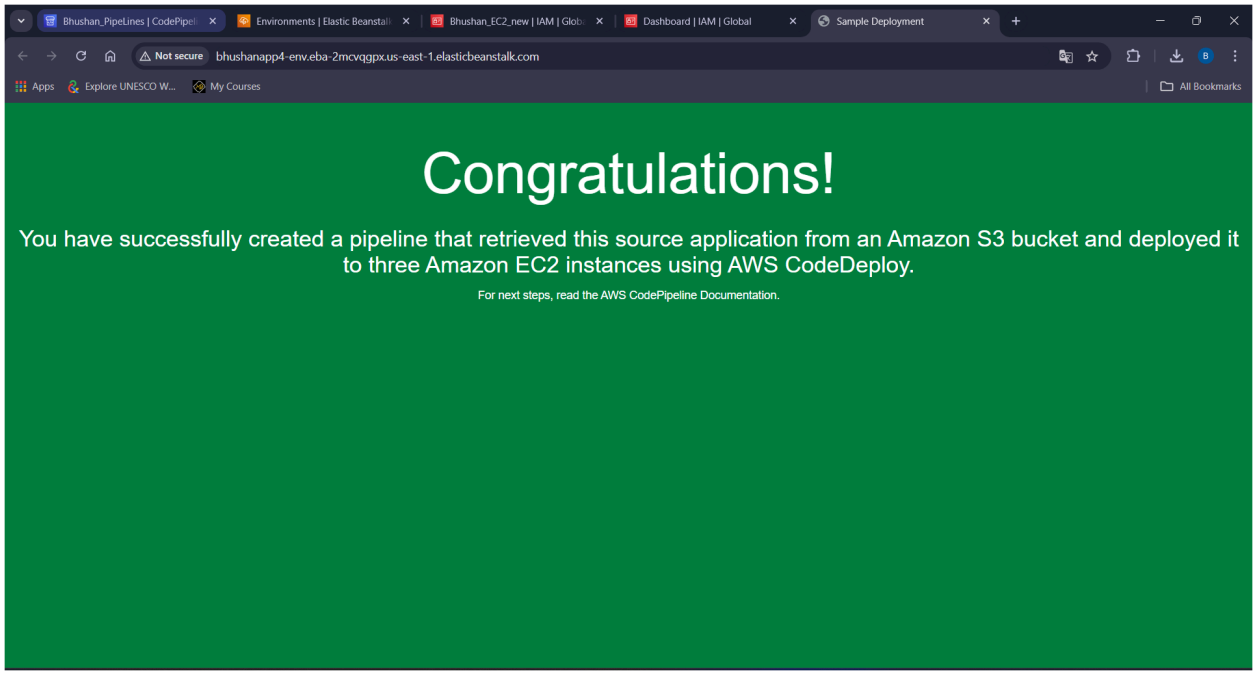
Step 7: Your Pipeline is Created Successfully.Also it is successfully connected to the source and successfully Deployed.



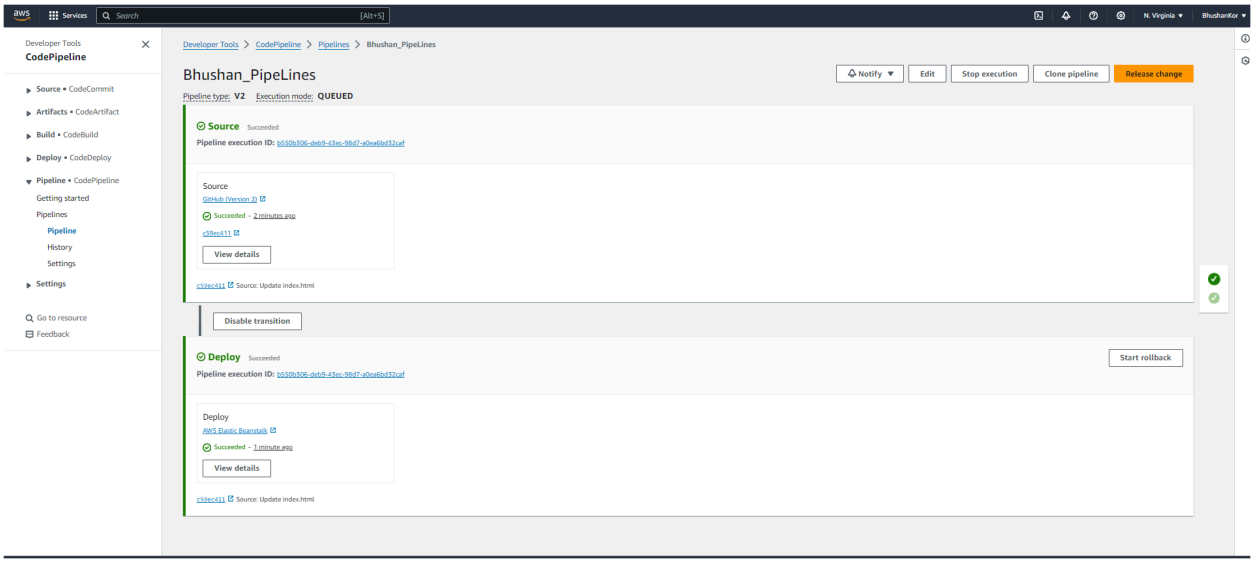
Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

Step 8: Click on the given link and you will see the Result.



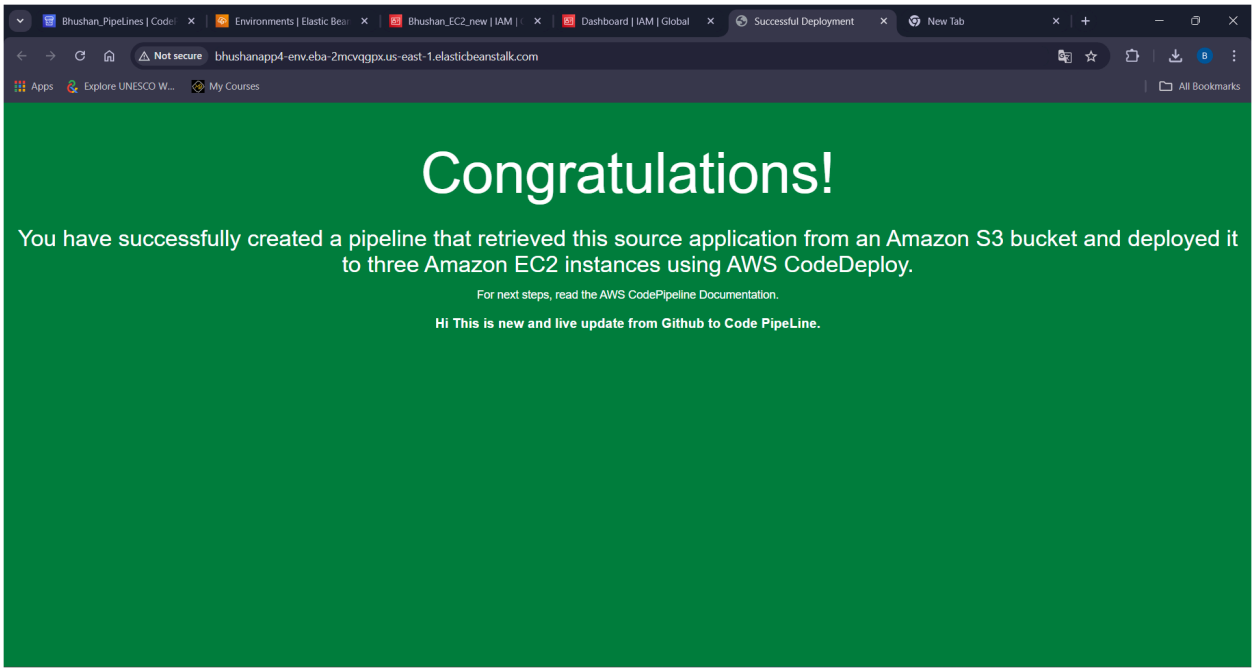
Step 9: Make Some changes in the Source code file on GitHub and commit changes and then deploy the changes in pipeline.



Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

Step 10: Name of website is changed also one we line is added at the end.



Name:Bhushan Mukund Kor **Academic Year:2024-2025**

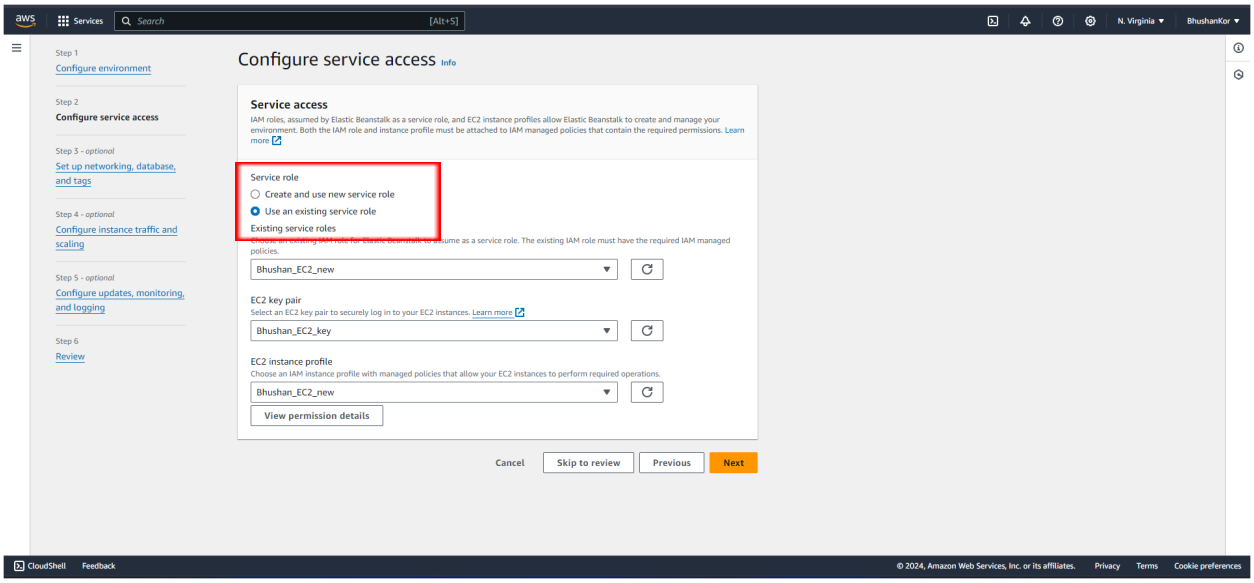
Division: D15C **Roll No: 28**

New Learnings in this experiment:

Note:

Previously Elastic Beanstalk created a default EC2 instance profile named `aws-elasticbeanstalk-ec2-role` the first time an AWS account created an environment. This instance profile included default managed policies. If your account already has this instance profile, it will remain available for you to assign to your environments.

However, recent AWS security guidelines don't allow an AWS service to automatically create roles with trust policies to other AWS services, EC2 in this case. Because of these security guidelines, Elastic Beanstalk no longer creates a default `aws-elasticbeanstalk-ec2-role` instance profile.



Because of the above problem, we have chosen the existing role and for that, we have to create a new role with certain policies.

Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

To create an instance profile

1. Open the Roles page in the IAM console.
2. Choose Create role.
3. Under the Trusted entity type, choose AWS service.
4. Under Use case, choose EC2.
5. Choose Next.
6. Attach the appropriate managed policies provided by Elastic Beanstalk and any additional policies that provide permissions that your application needs.
7. Choose Next.
8. Enter a name for the role.
9. (Optional) Add tags to the role.
10. Choose Create role.

For Point 6 Do the below Steps

To add managed policies to the role attached to the default instance profile

1. Type `AWSElasticBeanstalk` to filter the policies.
2. Select the following policies, and then choose Attach policy:
   - `AWSElasticBeanstalkWebTier`
   - `AWSElasticBeanstalkWorkerTier`
   - `AWSElasticBeanstalkMulticontainerDocker`
3. Also, add `AmazonS3FullAccess` or `AmazonDynamoDBFullAccess`.
4. Choose Attach policy.

Add the Trust relationship policy for EC2

To allow the EC2 instances in your environment to assume the required role, the instance profile must specify Amazon EC2 as a trusted entity in the trust relationship policy, as follows.

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
```

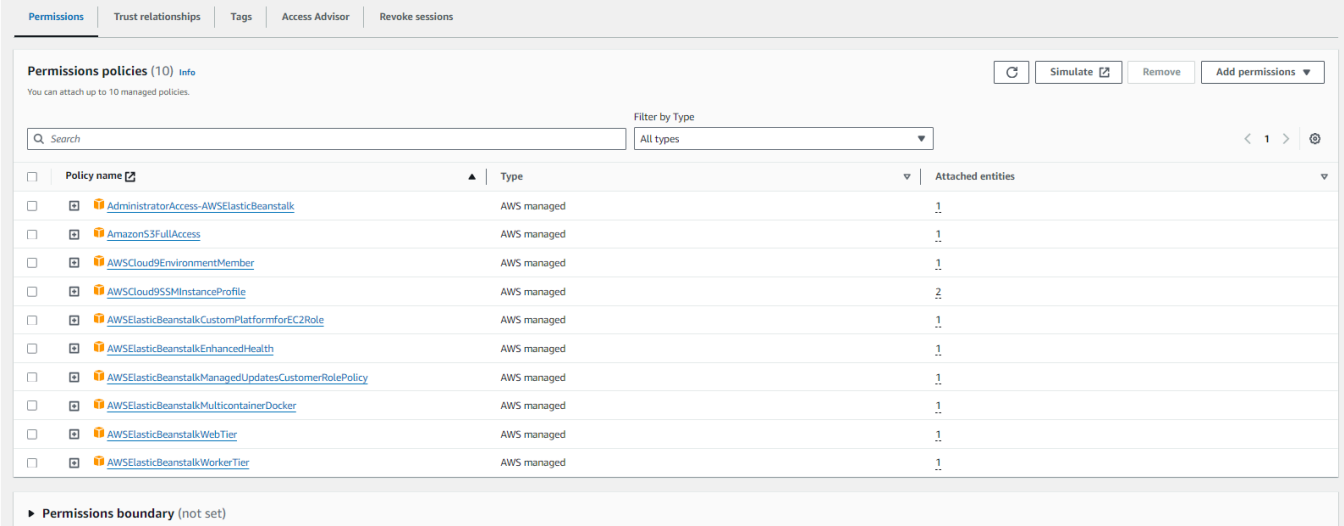
```
            "Service": "ec2.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
  ]

}
```

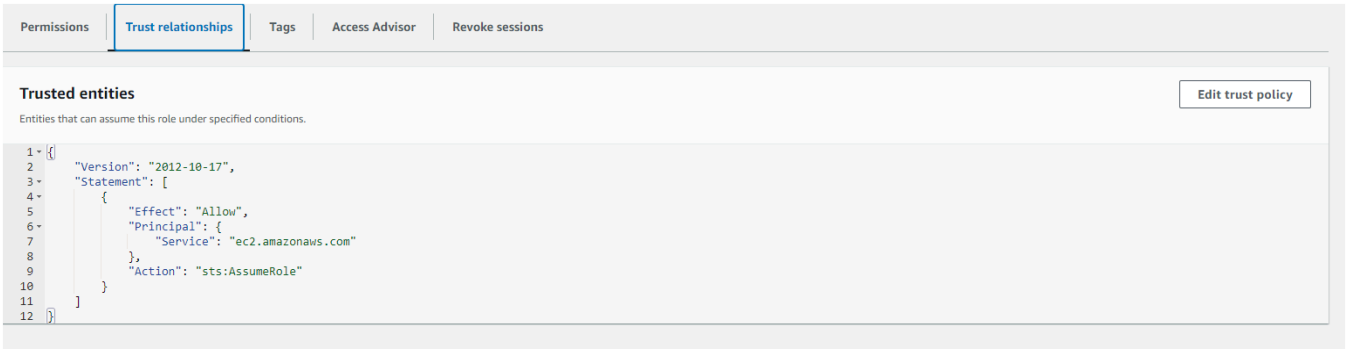
To customize permissions, you can add policies to the role attached to the default instance profile or create your own instance profile with a restricted set of permissions.

Screenshot for reference

Policies:



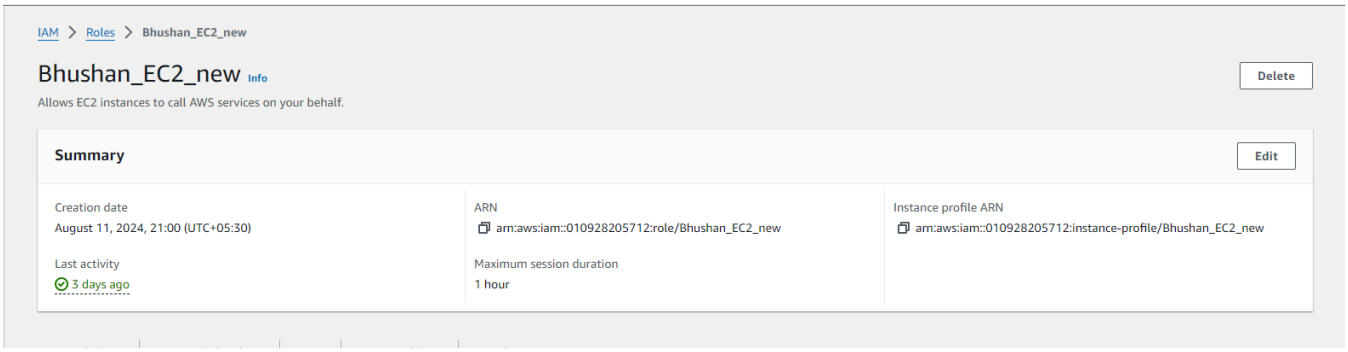
Trust Relationship:



Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

IAM Role:



EC2 Instance with attached IAM Role: