Terna Engineering College

**Computer Engineering Department**

Program: Sem VI

**Faculty: PREETI PATIL**

* 1. **Aim:**

**Course: Cloud Computing Lab (CSL603) Experiment No.4**

**To study and Implement Platform as a Service using AWS Elastic Beanstalk/ Microsoft Azure App Service.**

* 1. **Prerequisite:**

Understanding of Virtualization,Basics of Networking ,Basics of security and privacy.

* 1. **Objective**:

**To demonstrate the steps to deploy Web applications or Web services written in different languages on AWS Elastic Beanstalk/ Microsoft Azure App Service.**

* 1. **Outcome: (LO2)**

After successful completion of this experiment students will be able to deploy the web application using AWS Elastic Beanstalk.

* 1. **Theory:**

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with popular programming languages such as Java,

.NET, PHP, Node.js, Python and Ruby. You simply upload your application and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling and application health monitoring. At the same time, with Elastic Beanstalk, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

Most existing application containers or platform-as-a-service solutions, while reducing the amount of programming required, significantly diminish developers' flexibility and control. Developers are forced to live with all the decisions pre-determined by the vendor

- with little to no opportunity to take back control over various parts of their application's infrastructure. However, with Elastic Beanstalk, you retain full control over the AWS resources powering your application. If you decide you want to take over some (or all) of

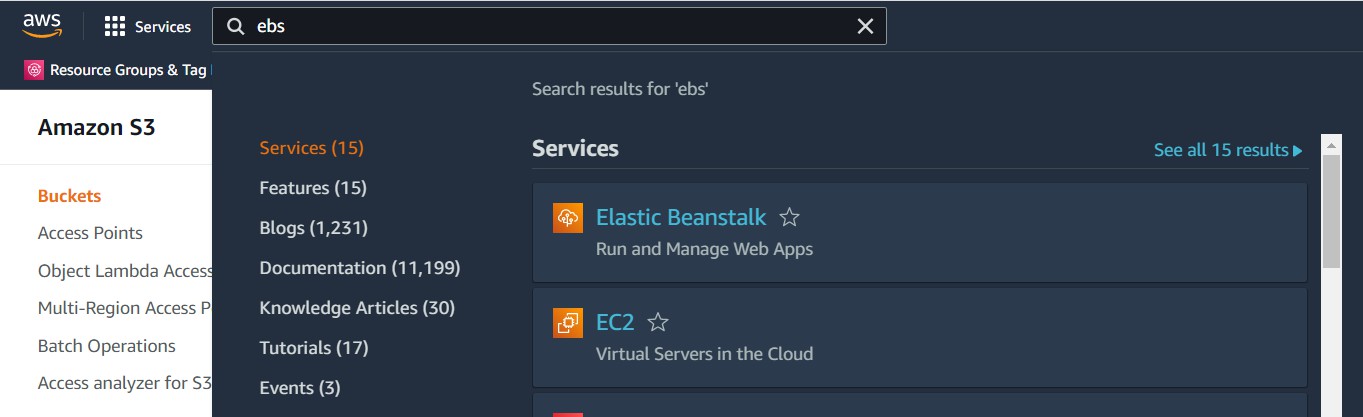
the elements of their infrastructure, you can do so seamlessly by using Elastic Beanstalk's management capabilities.

To ensure easy portability of your application, Elastic Beanstalk is built using familiar application/web servers such as Apache HTTP Server, Apache Tomcat, Nginx, Passenger and IIS 7.5/8.

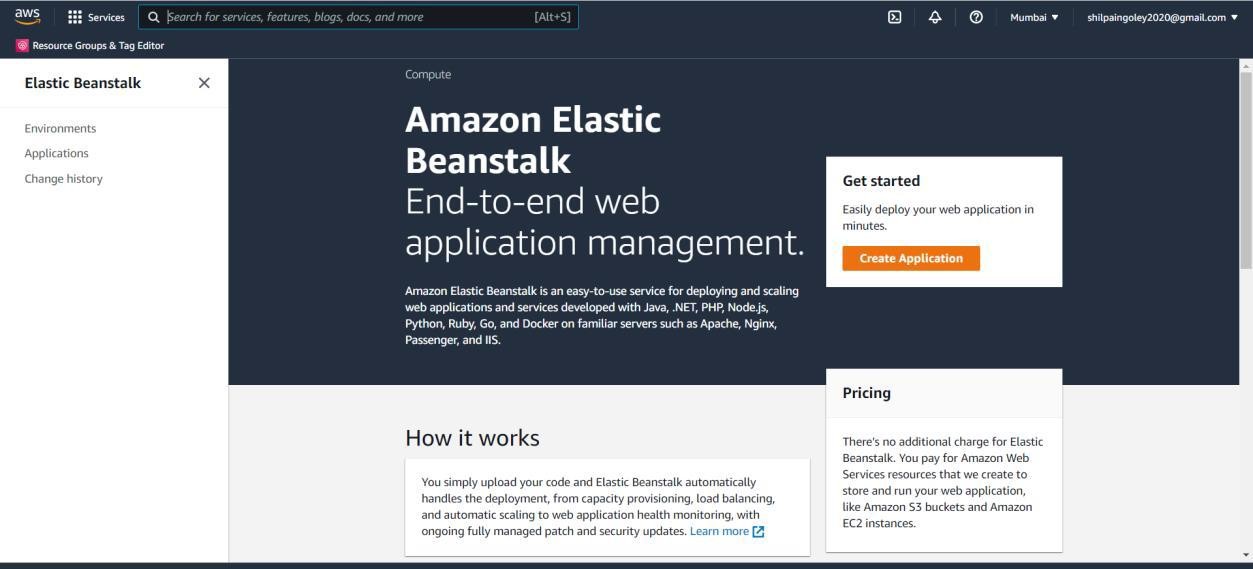
Following are steps to host a website using Elastic Beanstalk:

**(Creating Sample Application )**

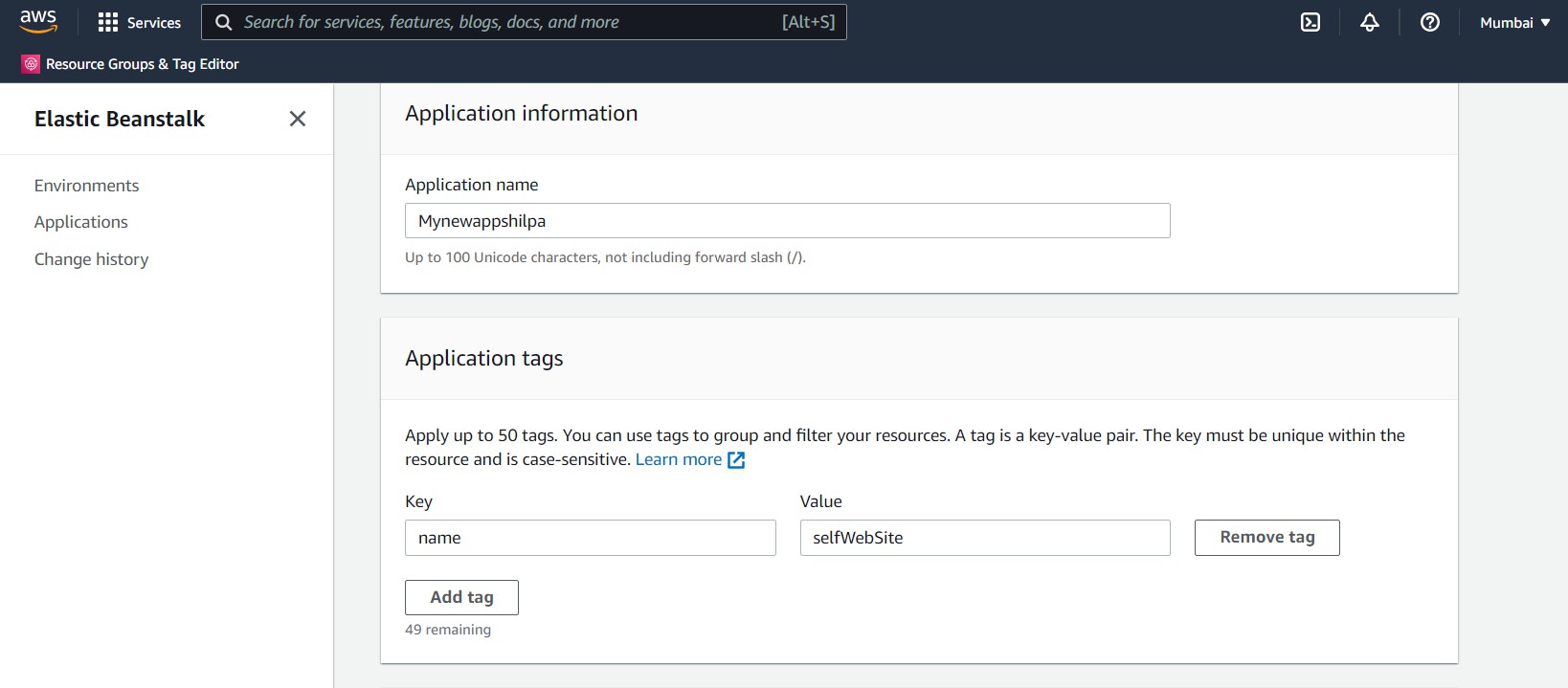
Step1 : Login to AWS console and go to Elastic Beanstalk

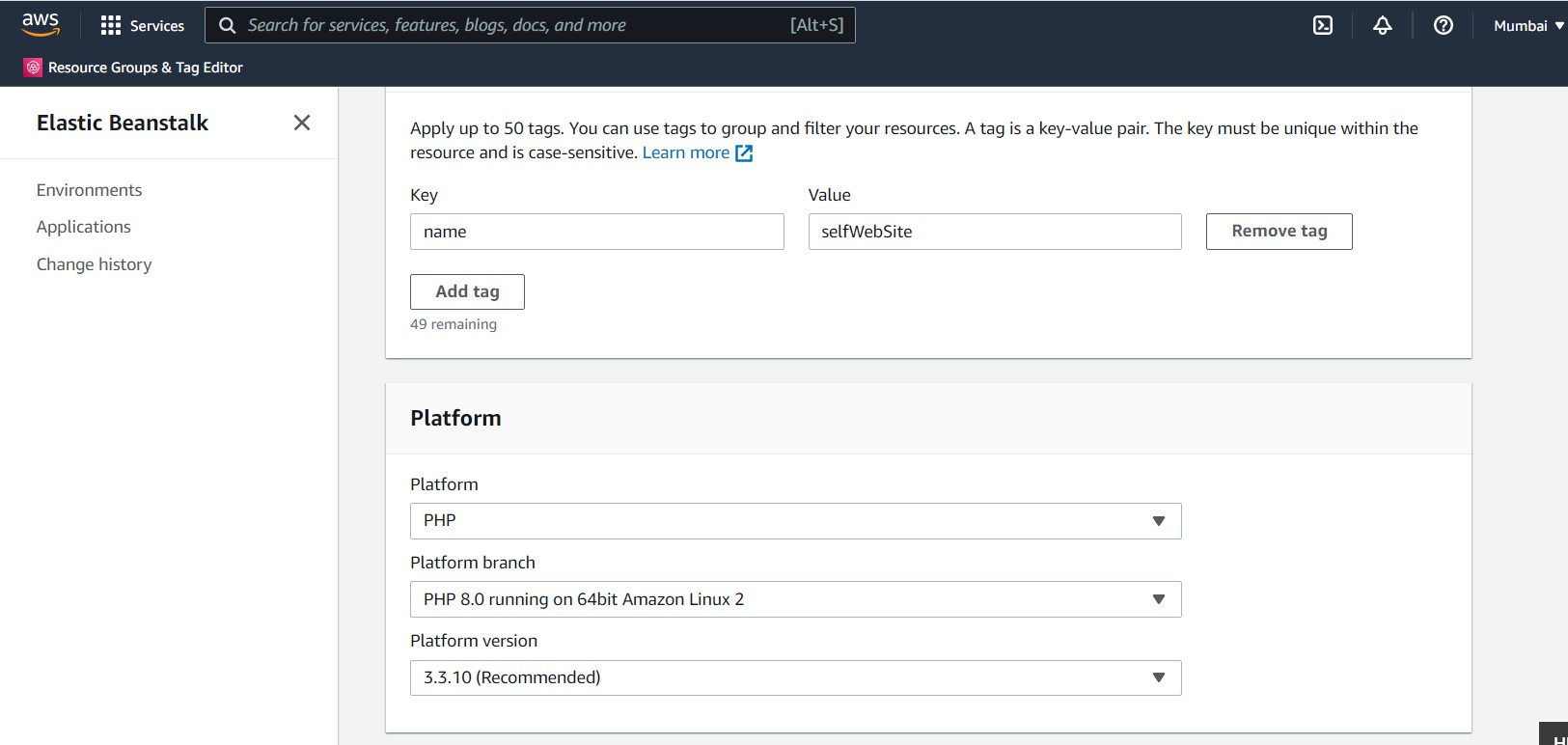


Step 2: Click on Create Application



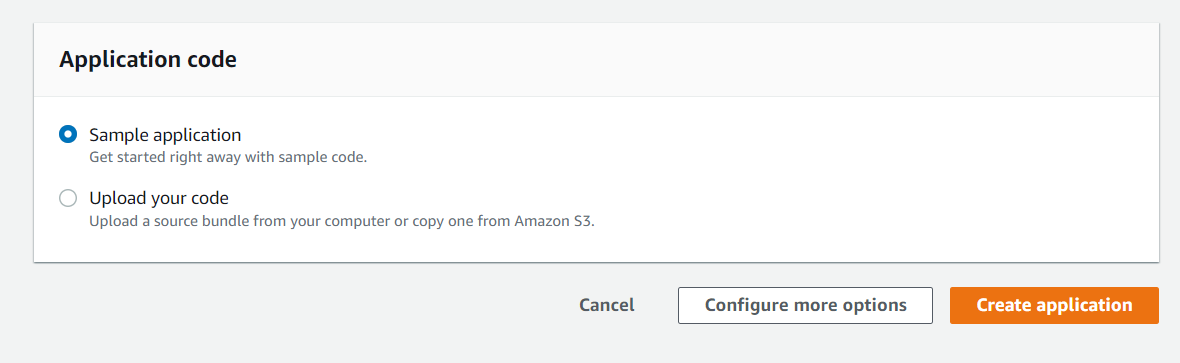
Step 3: Write Application information: Name, Tag,Platform etc.

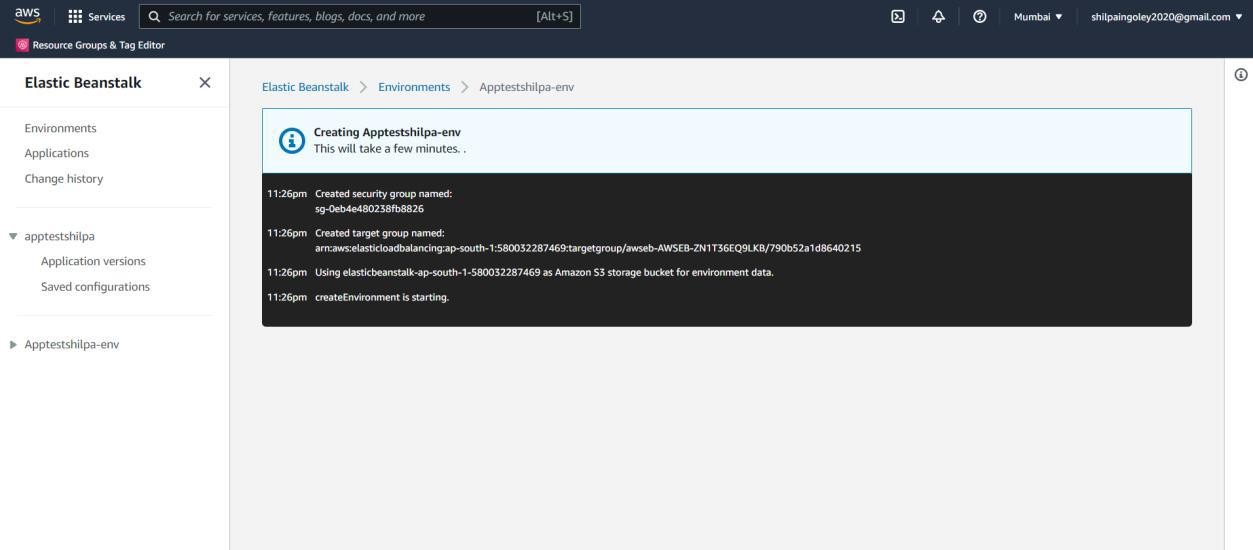


Step 4: In Application Code: select sample application and then Click on button Create Application

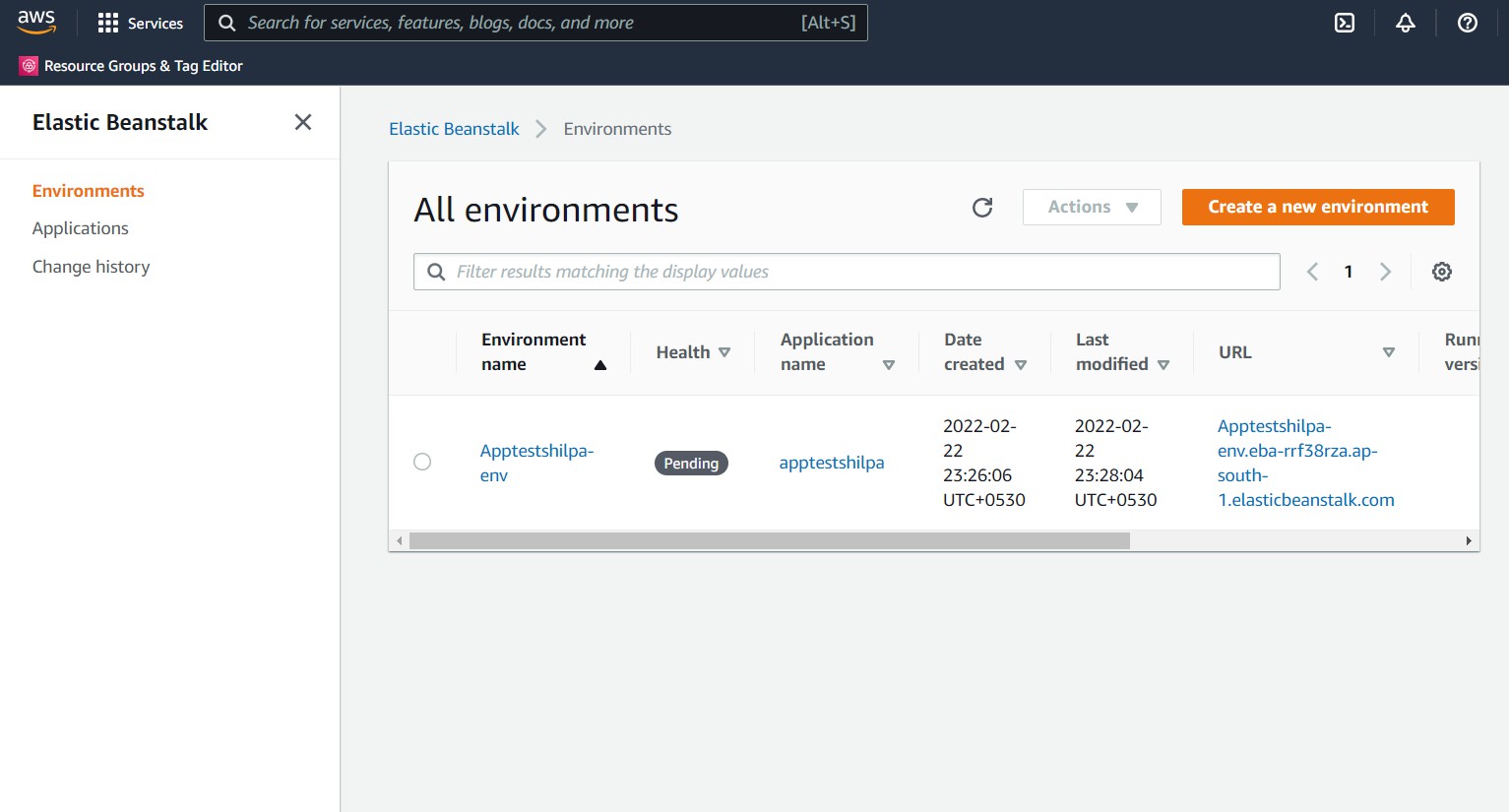
This will take a few minutes.

Step 5: Click on Environments -> Check the health of Environment wait till it becomes ‘OK’

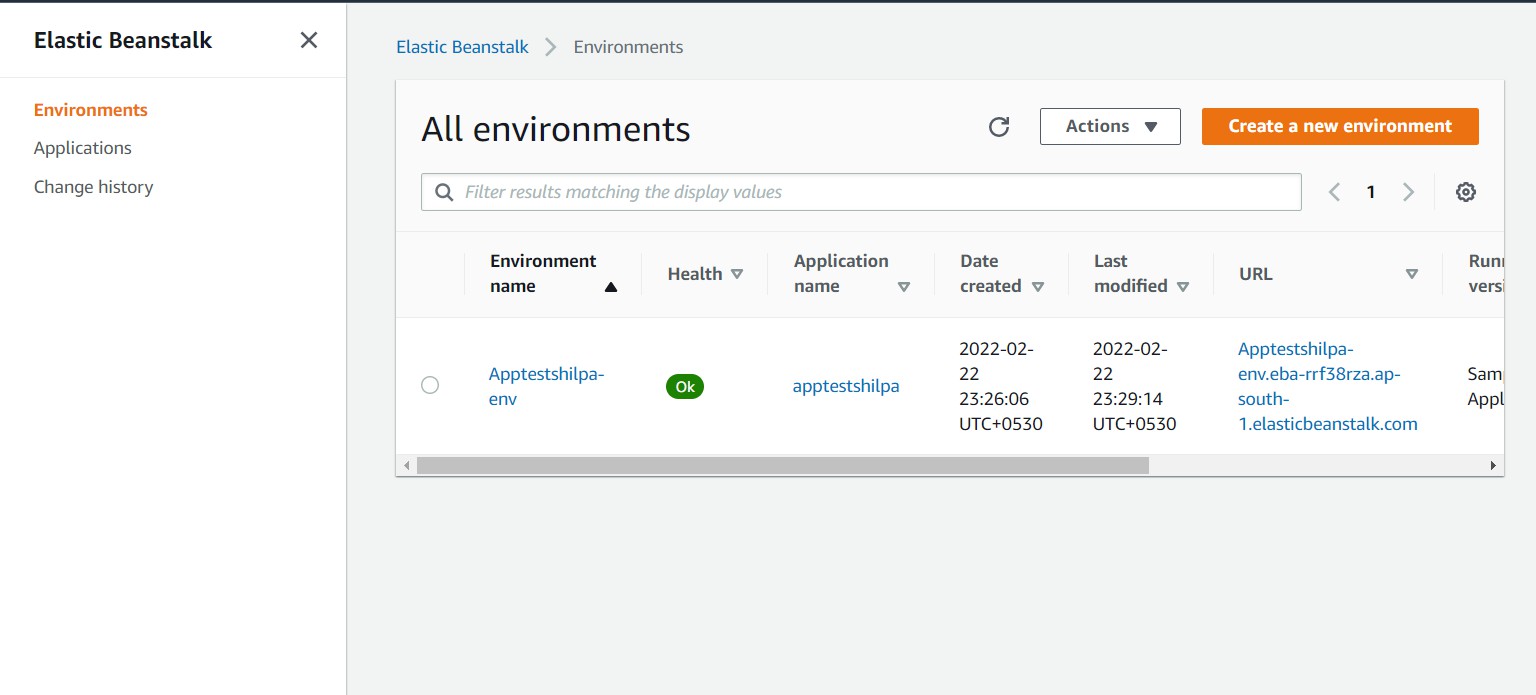




Step 6: Click the URL



To Delete the application and Environment (Select it and in **Action** -Delete/Terminate : give conformation)





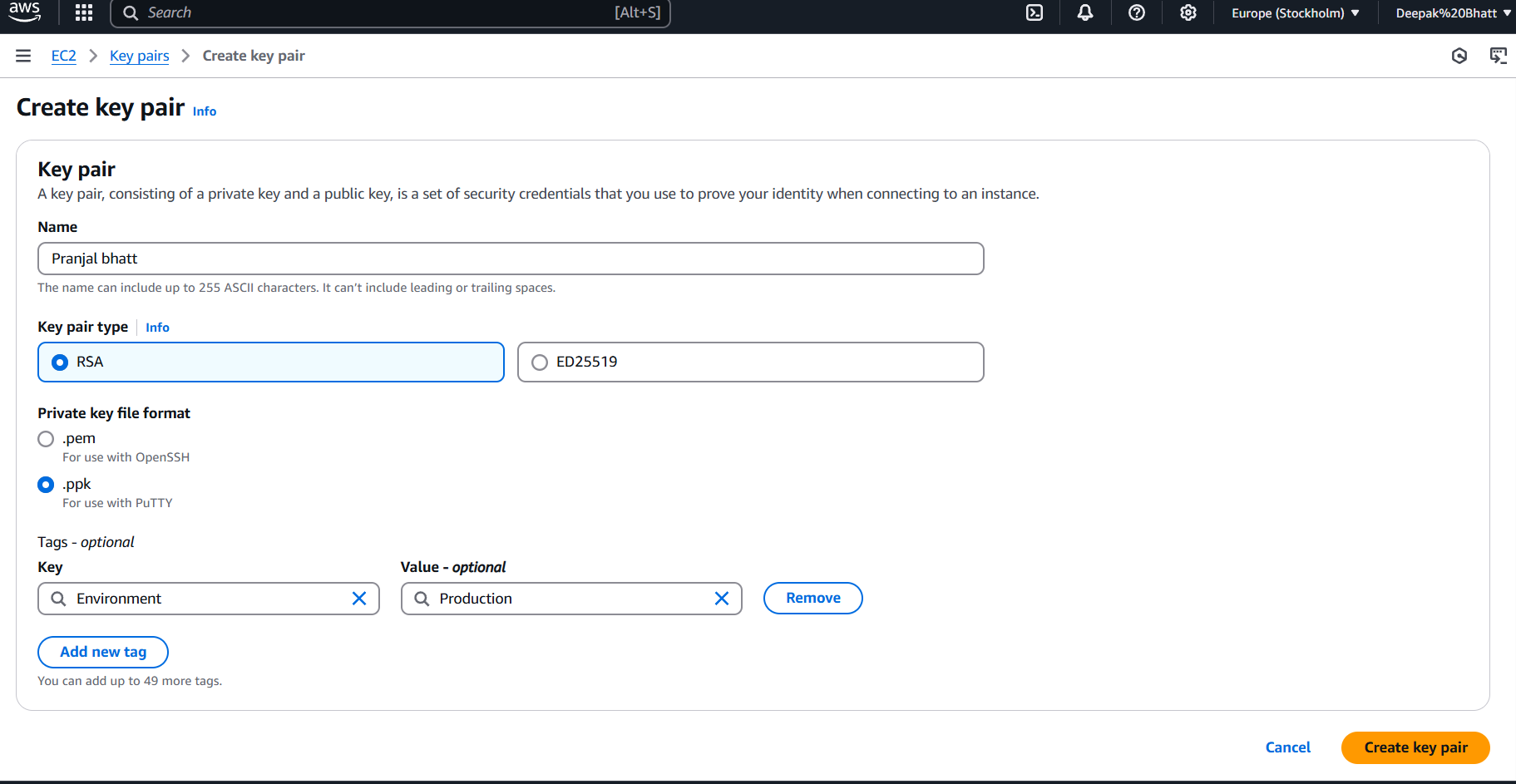
**PART B**

**(PART B: TO BE COMPLETED BY STUDENTS)**

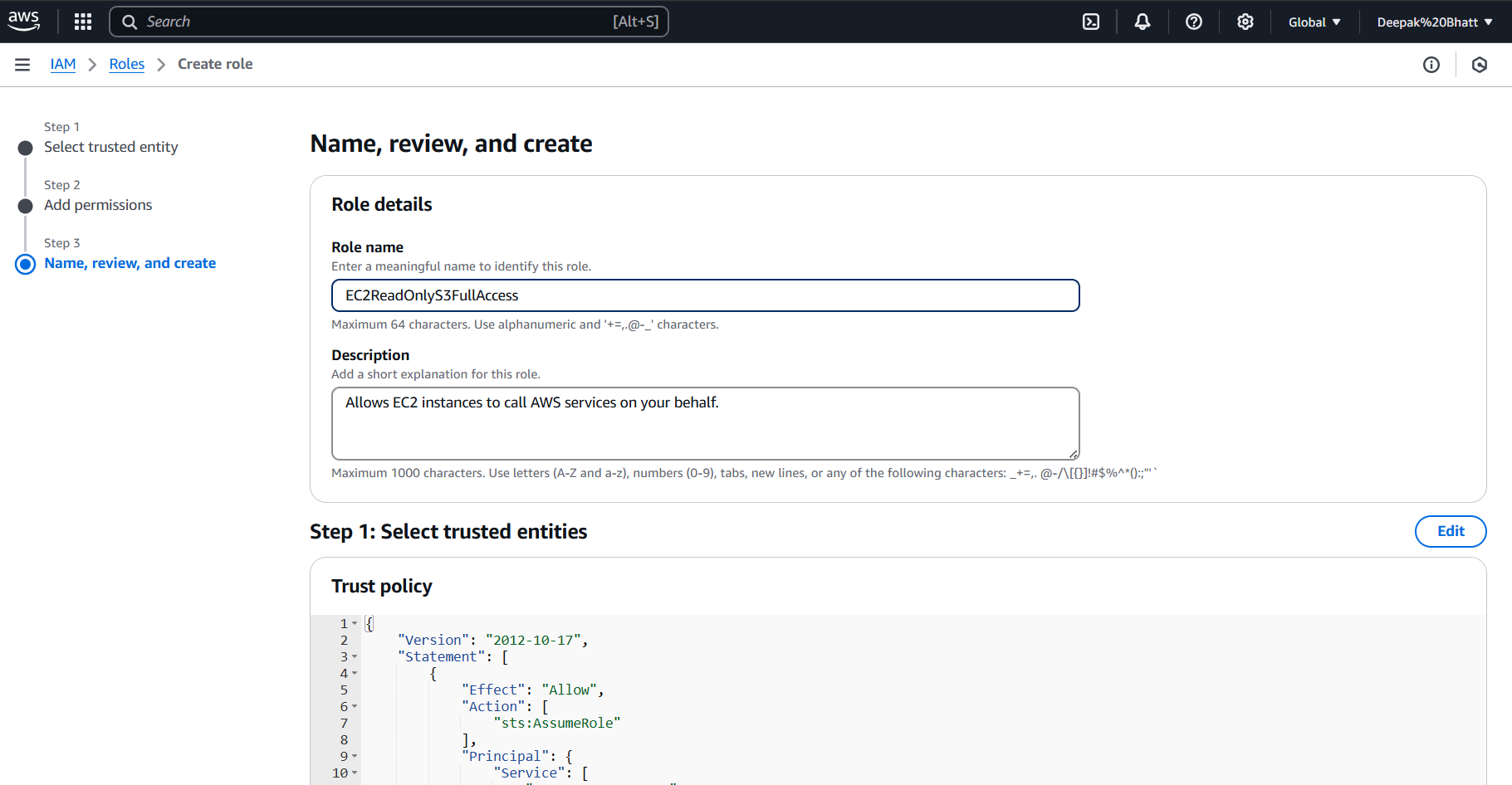
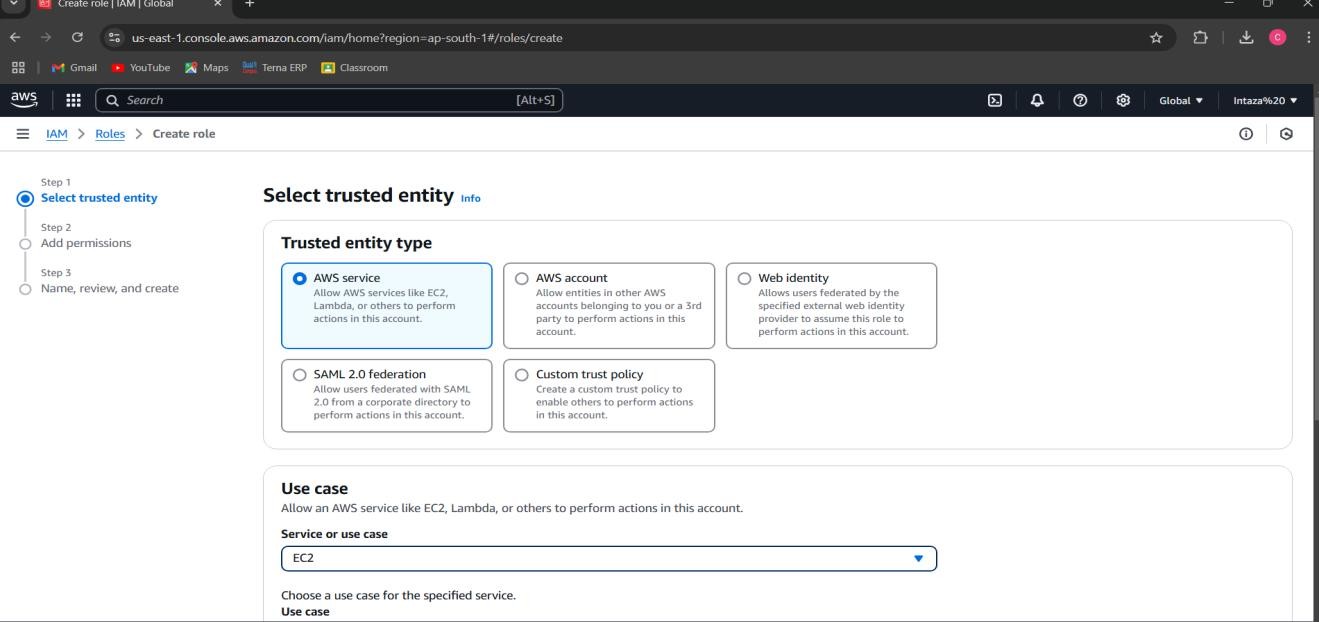
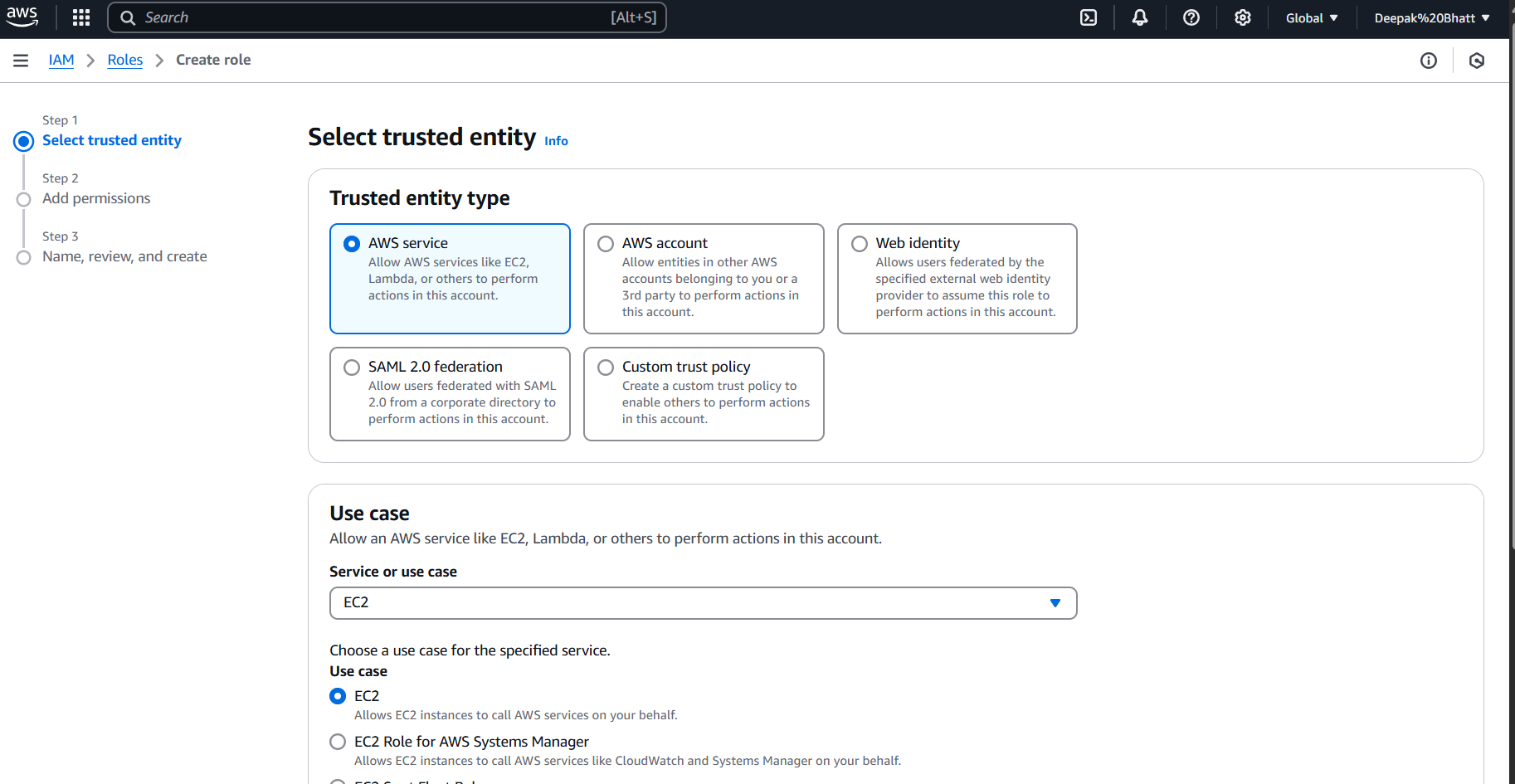
# (Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the ERP or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no ERP access available)

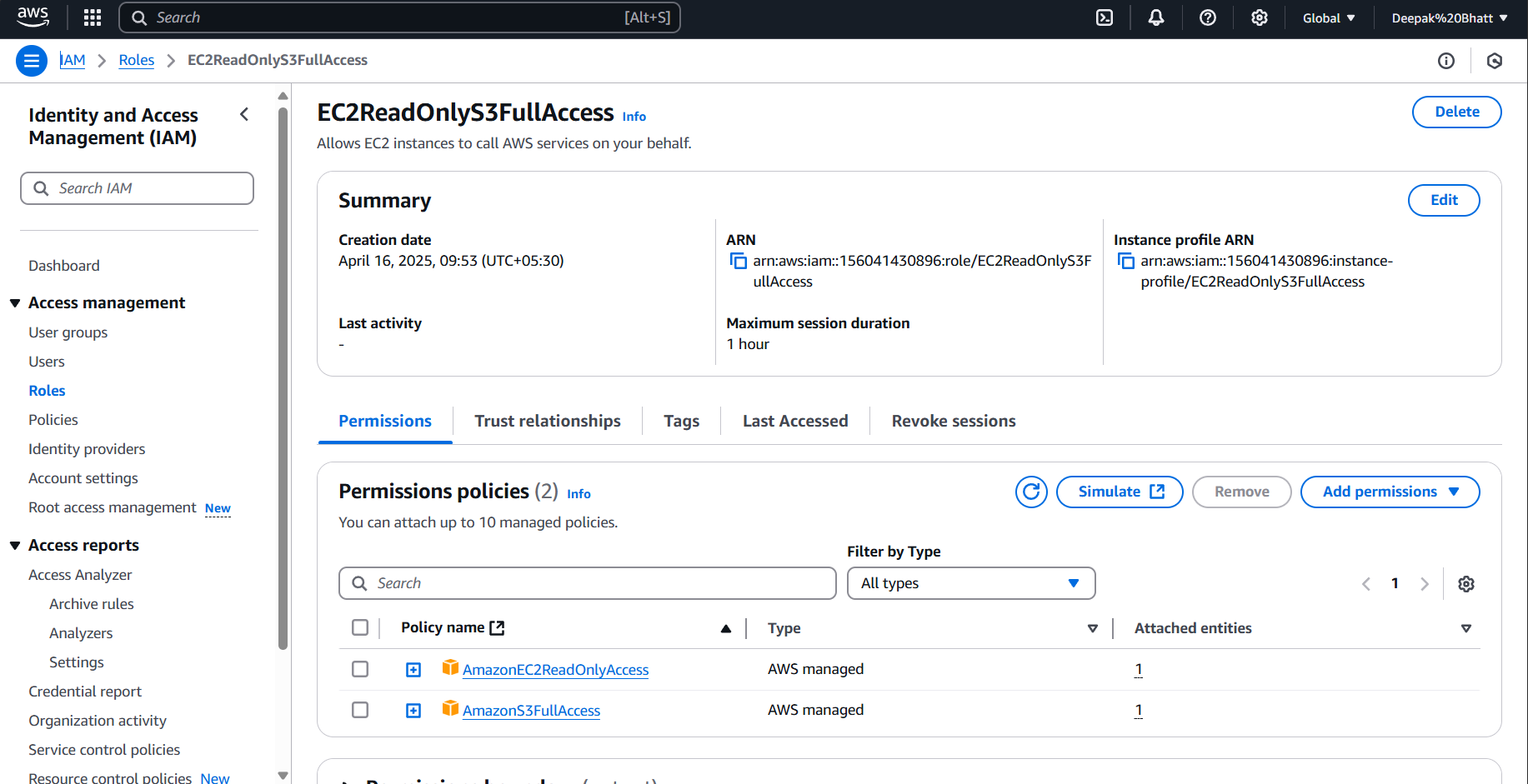
|  |  |
| --- | --- |
| Roll No. B30 | Name: Pranjal Bhatt |
| Class :TE COMPS B | Batch :B2 |
| Date of Experiment: | Date of Submission: |
| Grade : |  |

Step 1:Login to AWS Console and Create EC2 key pair.

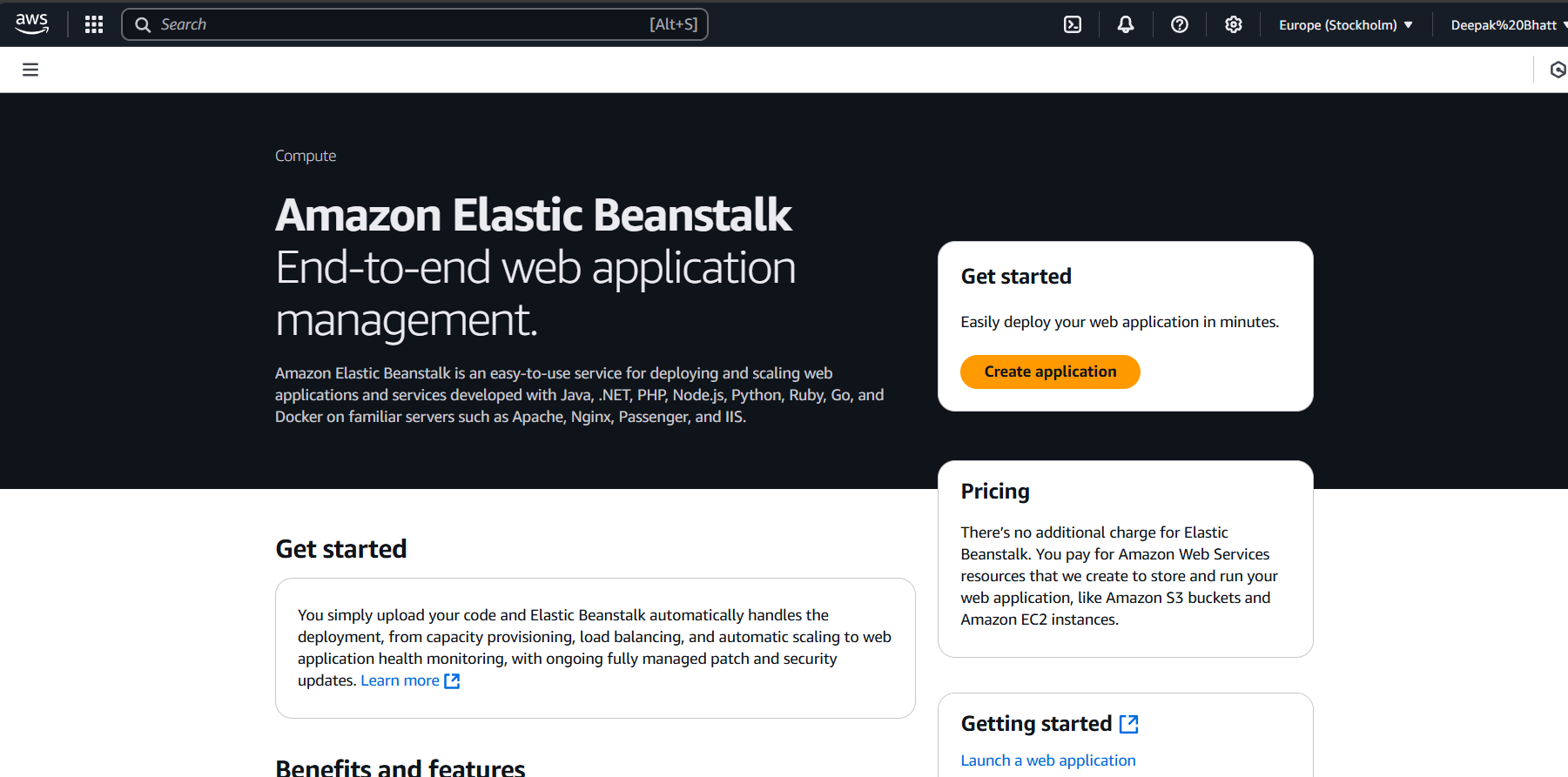


Step 2: Create Roles For EC2:

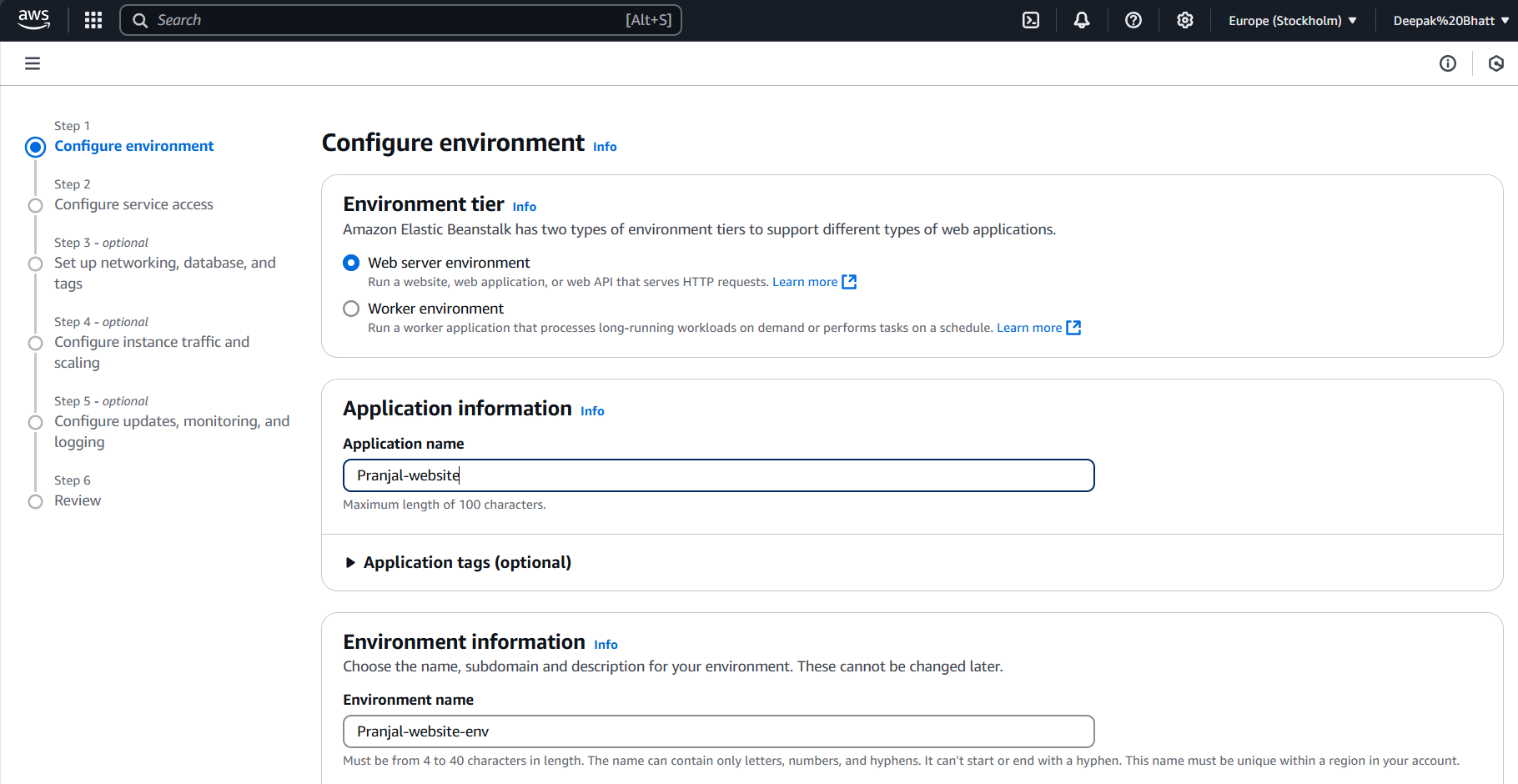


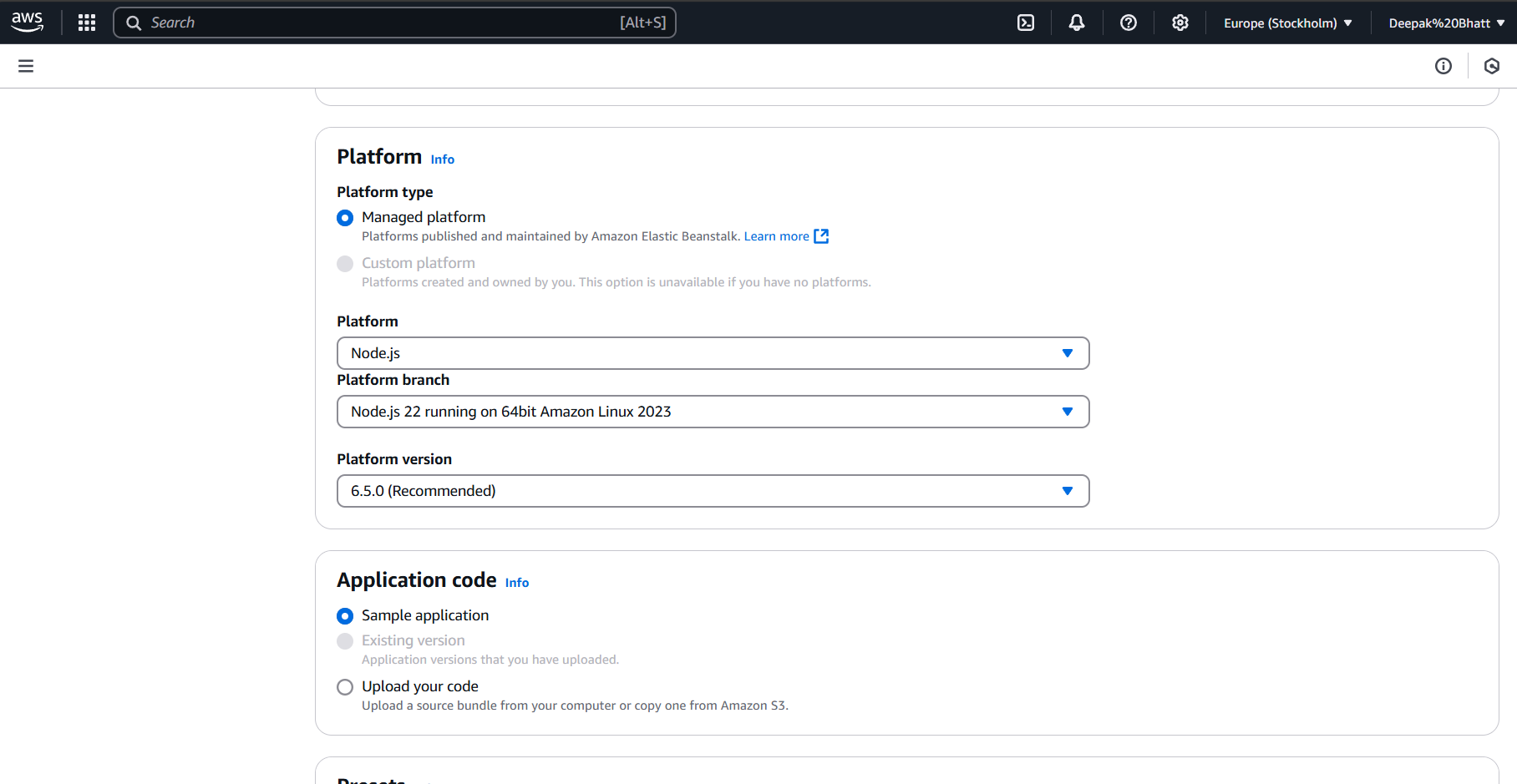


Step 3: Navigate to Elastic Beanstalk & Click on "Create Application"

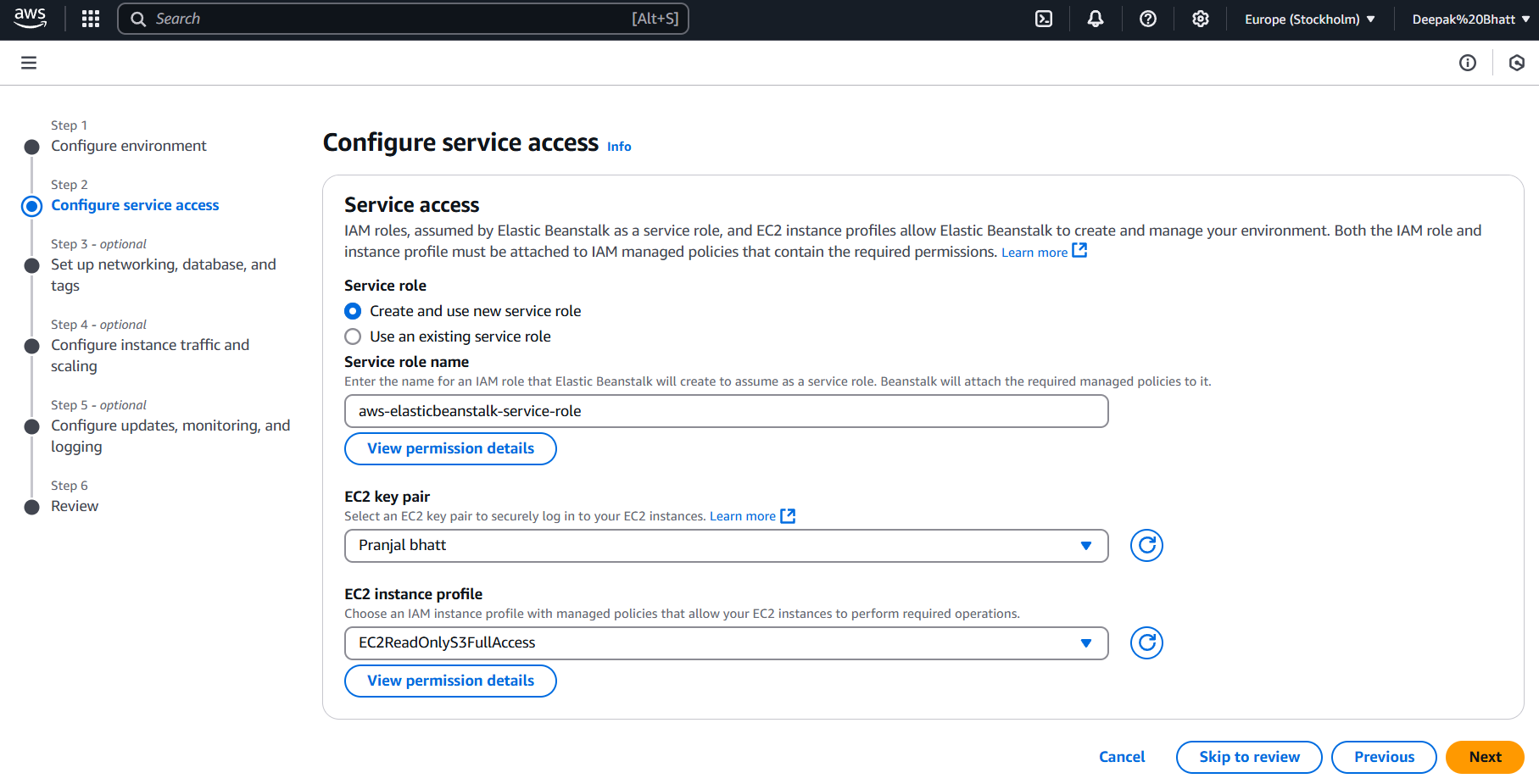


Step 4: Provide Application Information(Choose Environment tier,Application Name,Tags,Platform)

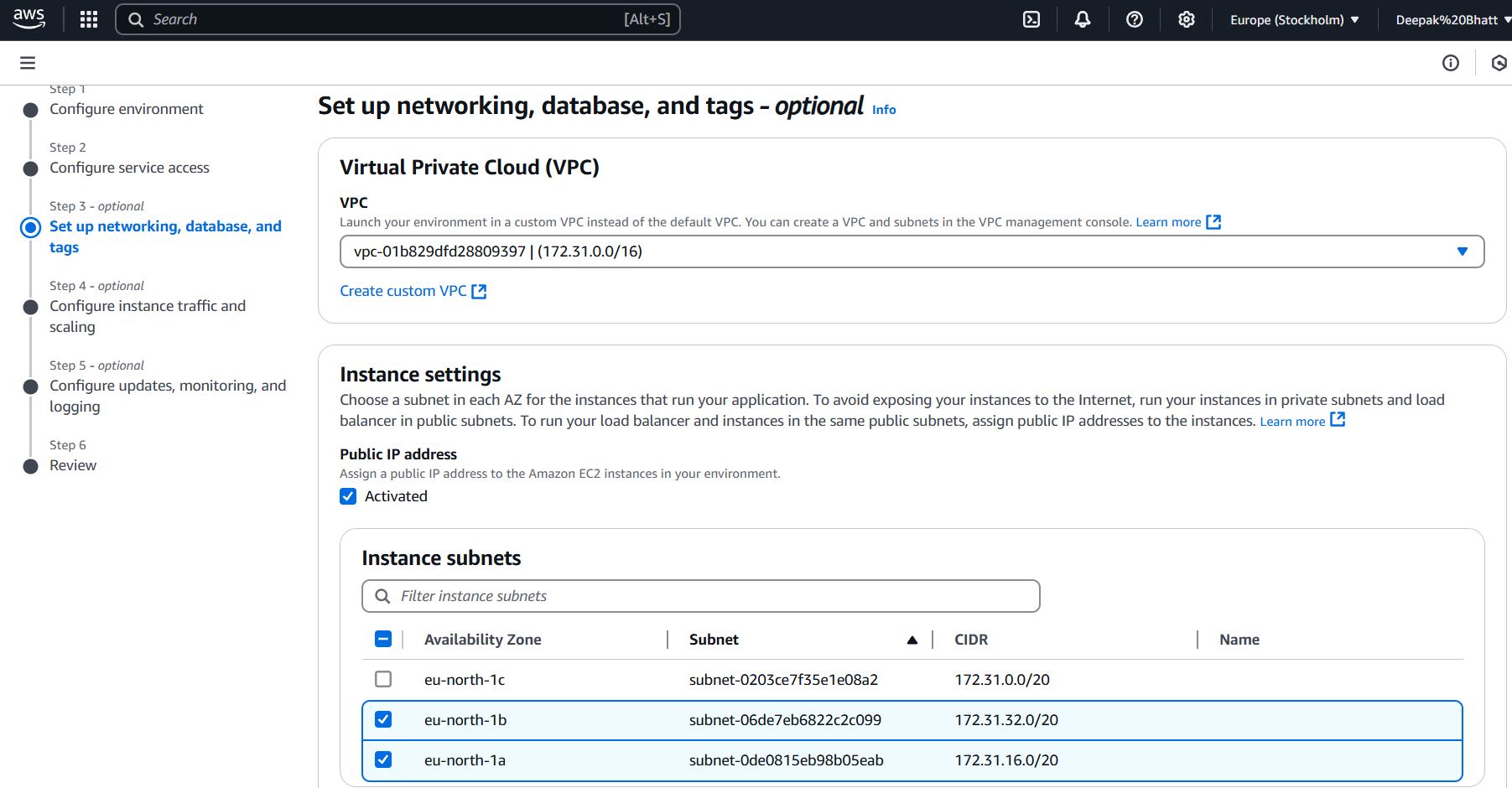


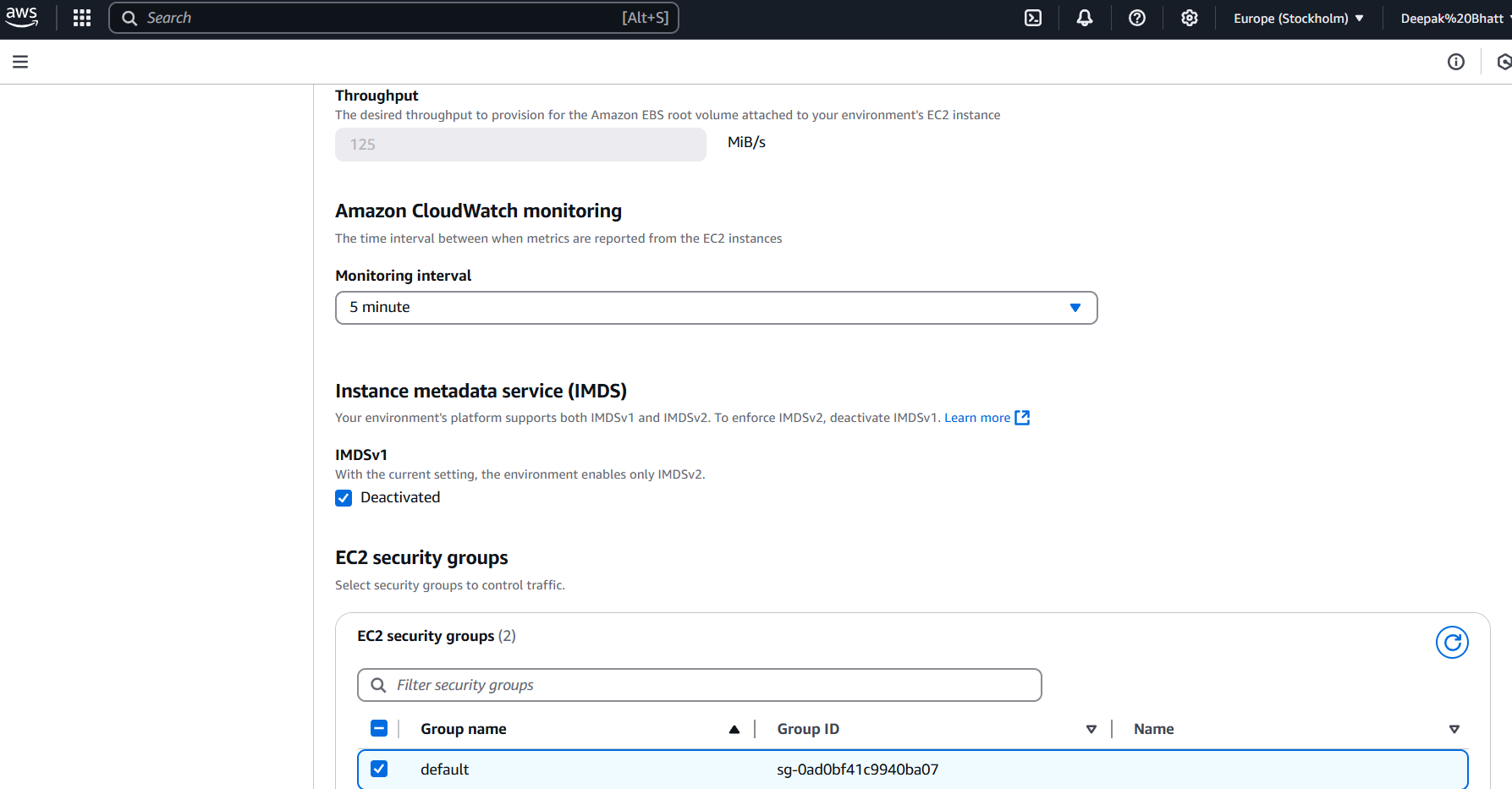


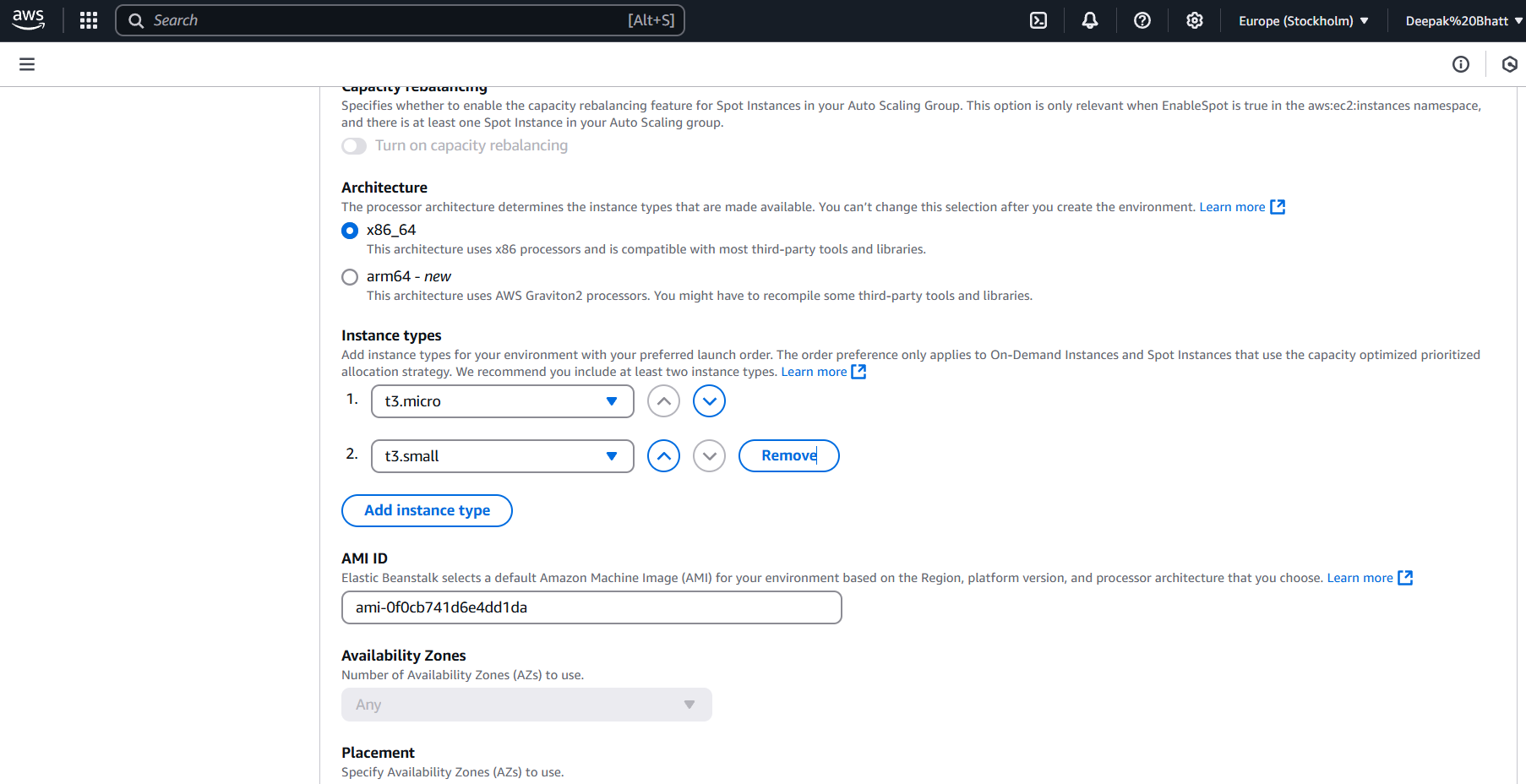
Step 5 : Configure service access



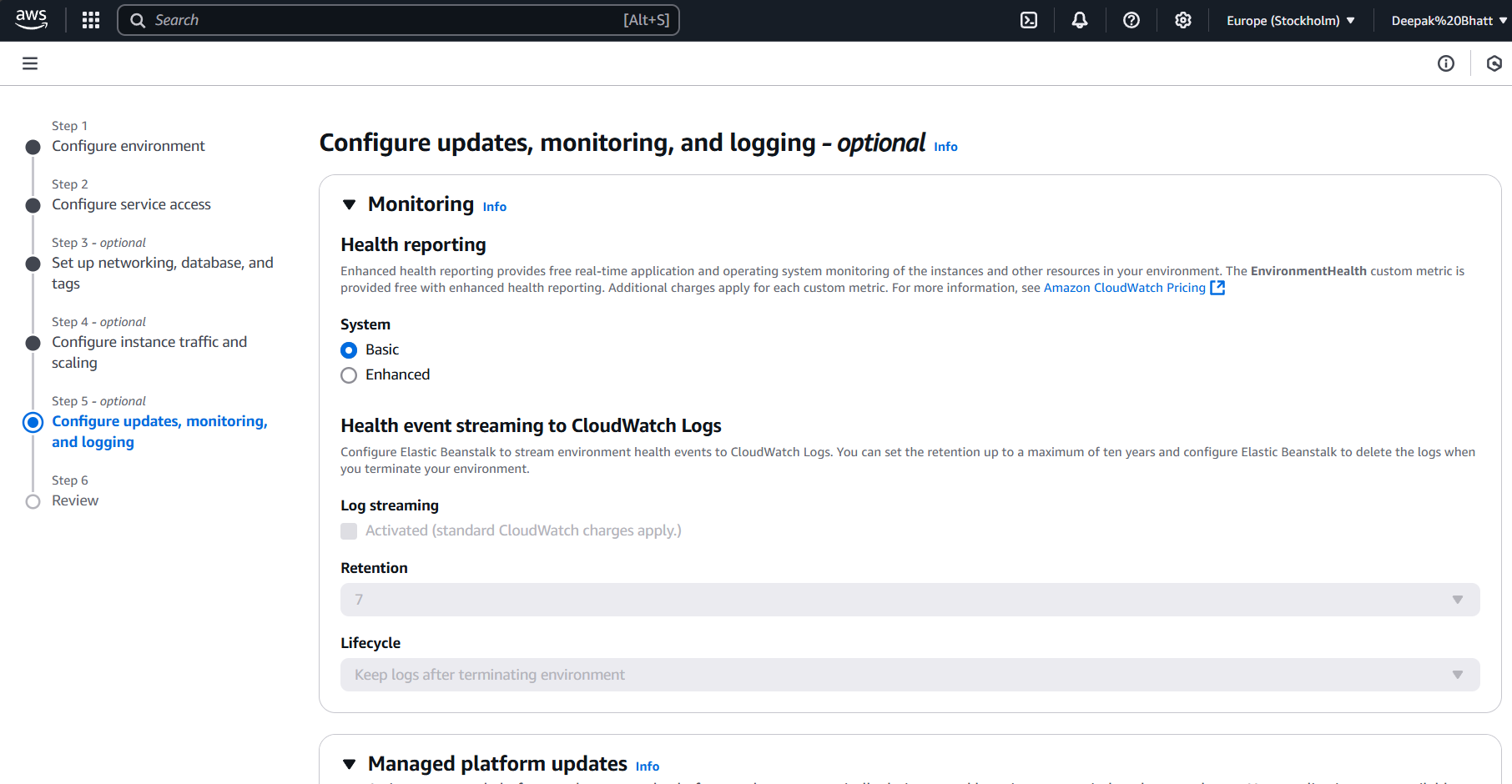
Step 6: Set up networking, database, and tags



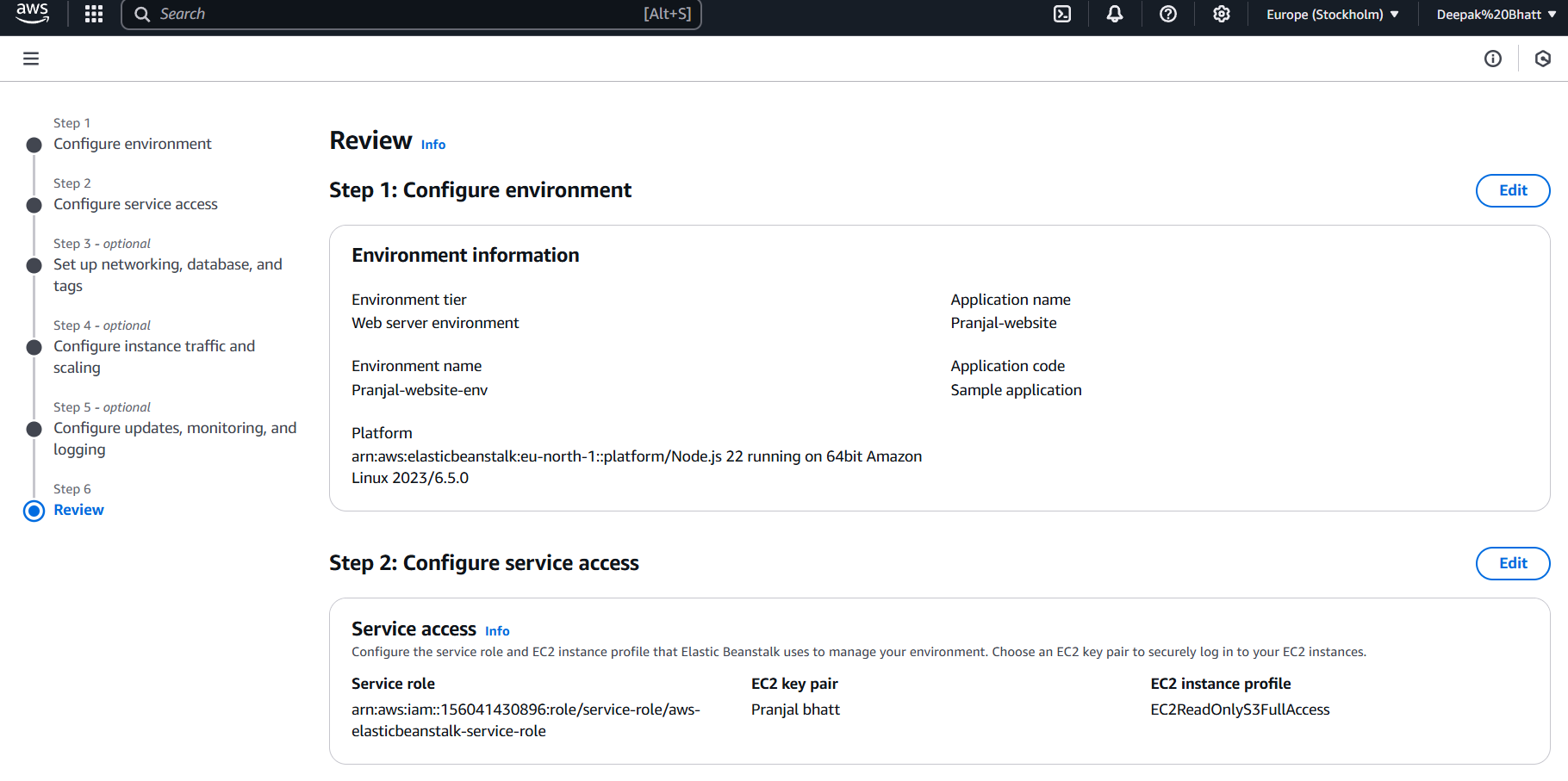
Step 7*-* Configure instance traffic and scaling

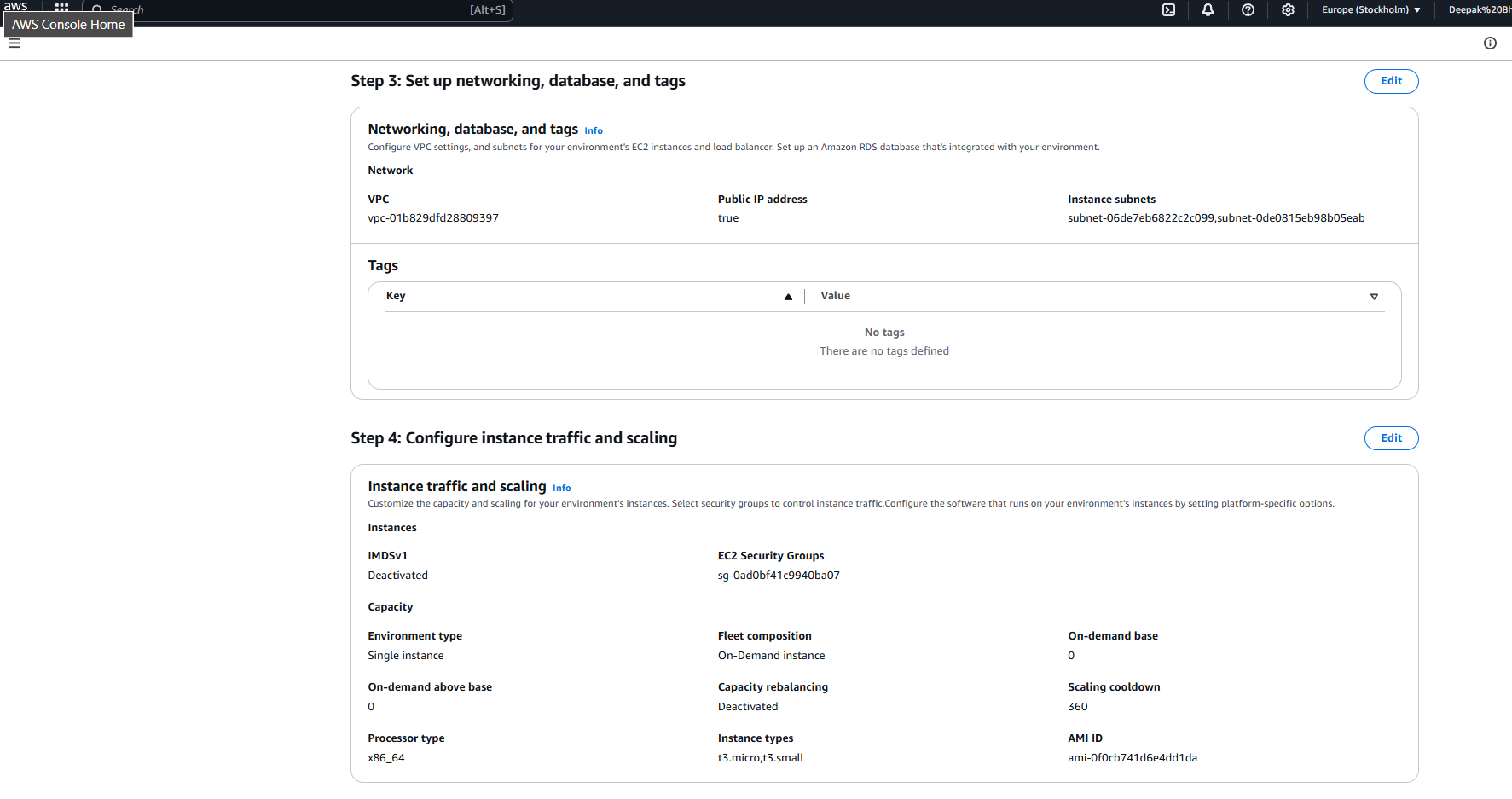


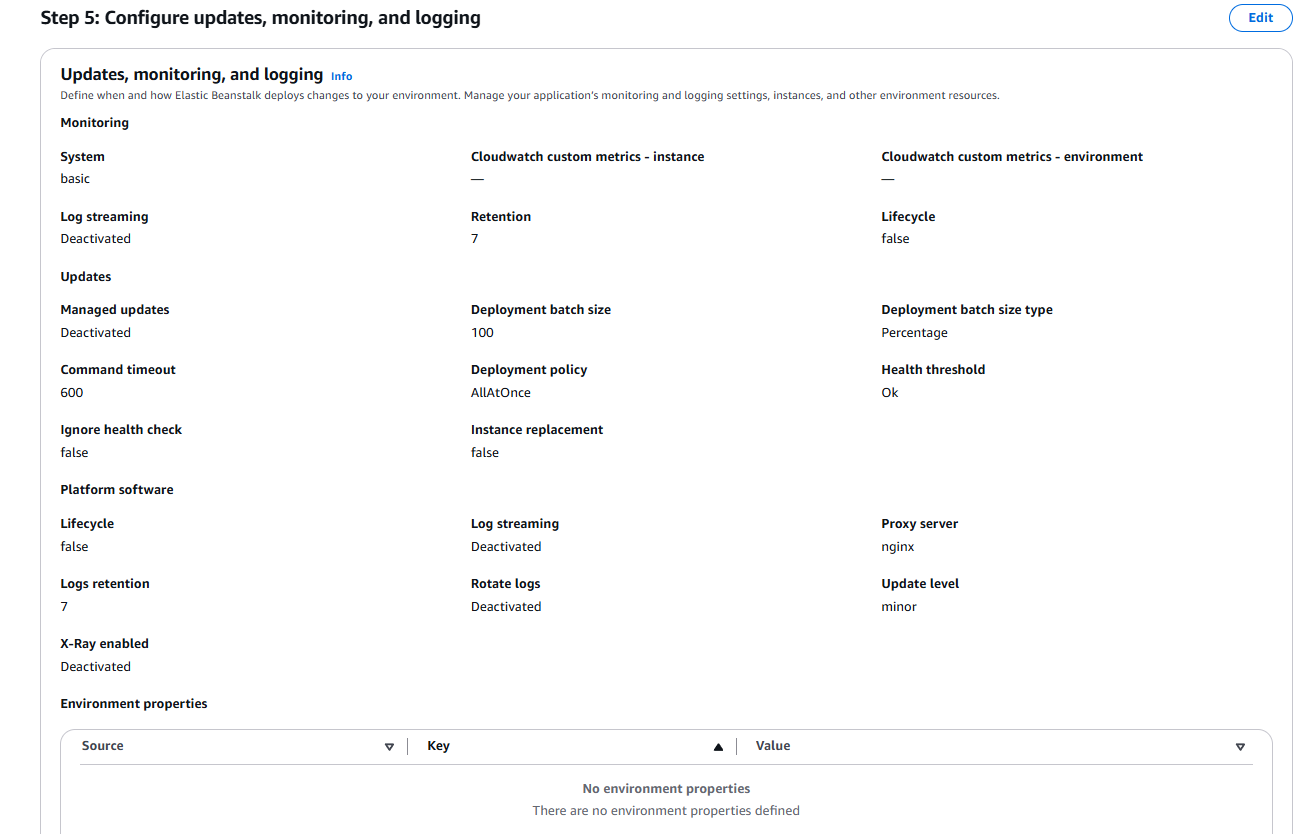
Step 8 *-* Configure updates, monitoring, and logging



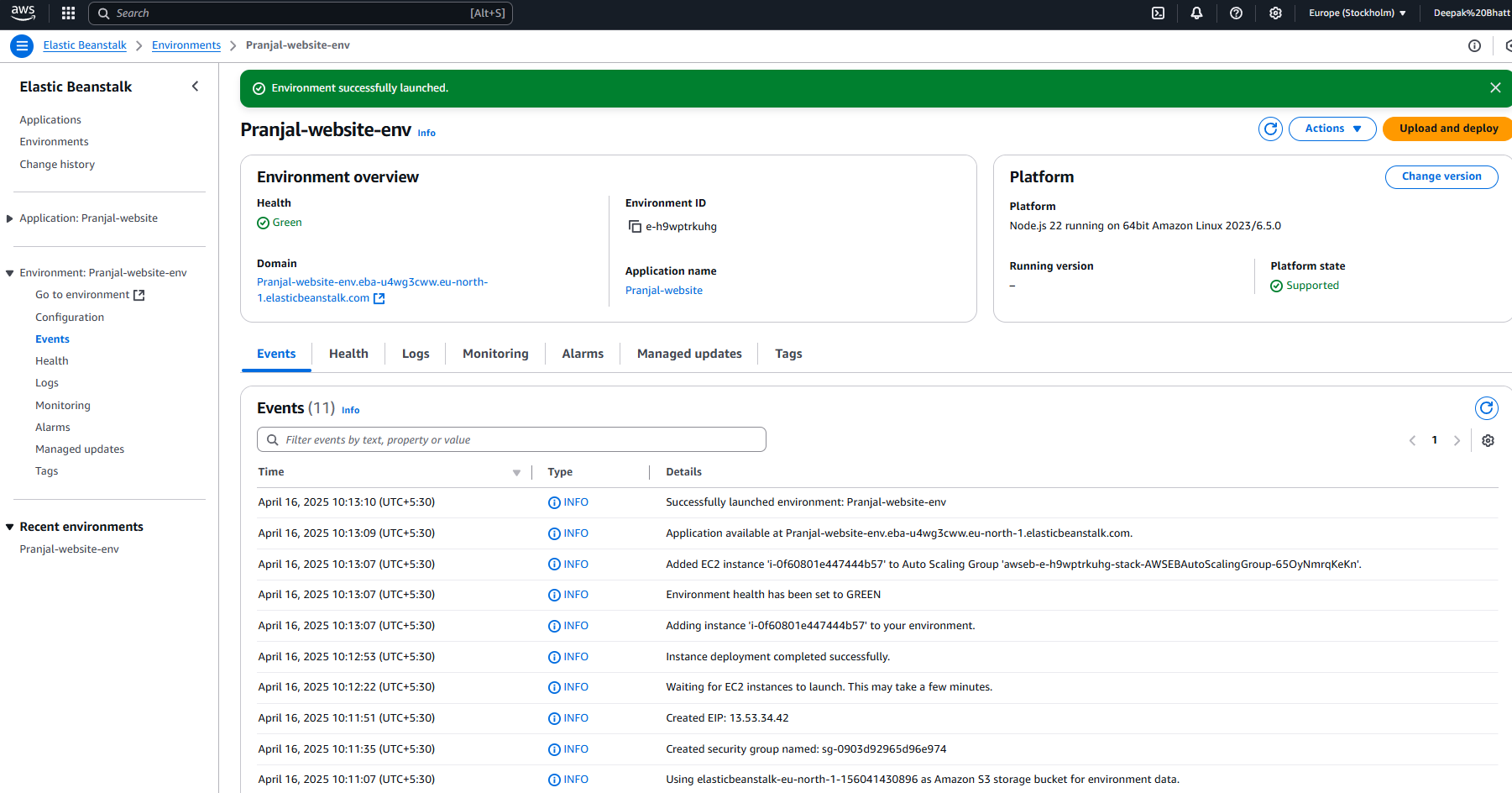
Step 9: Review

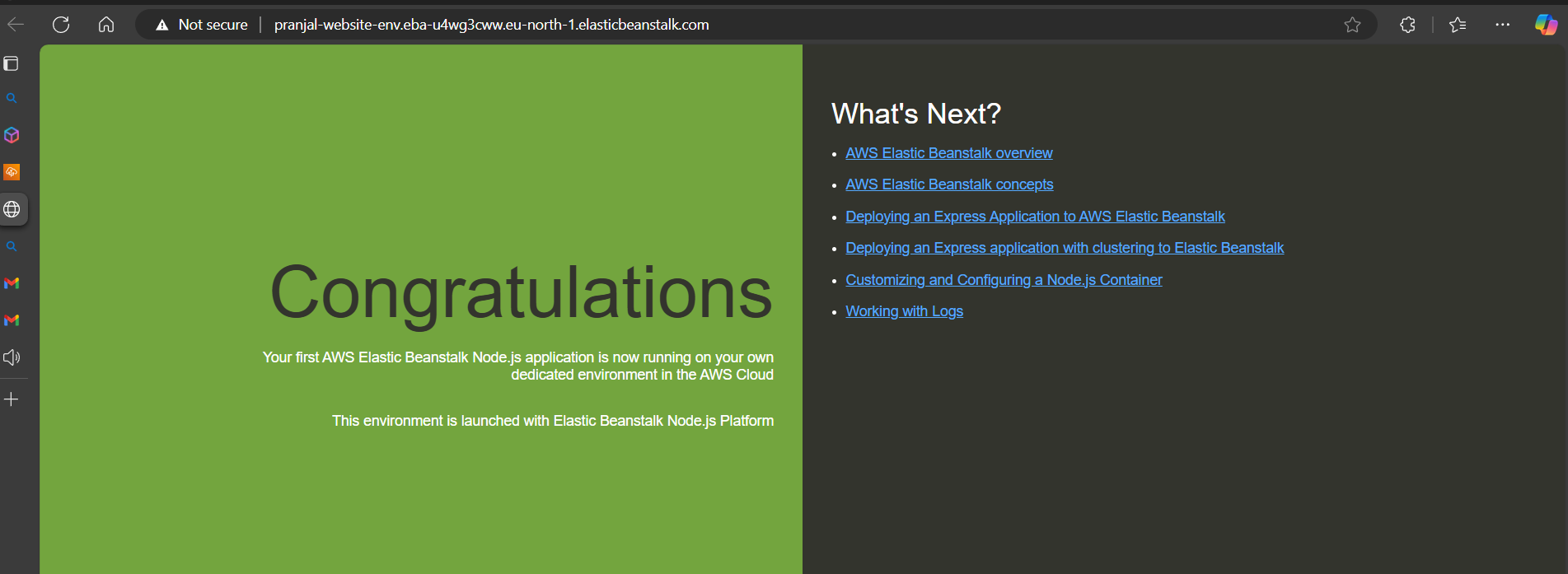


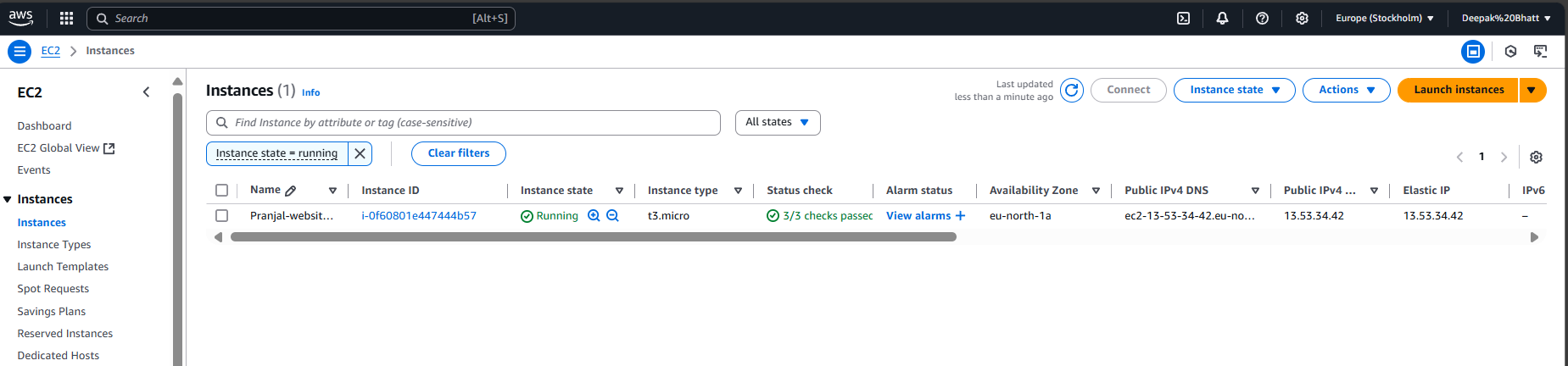
0

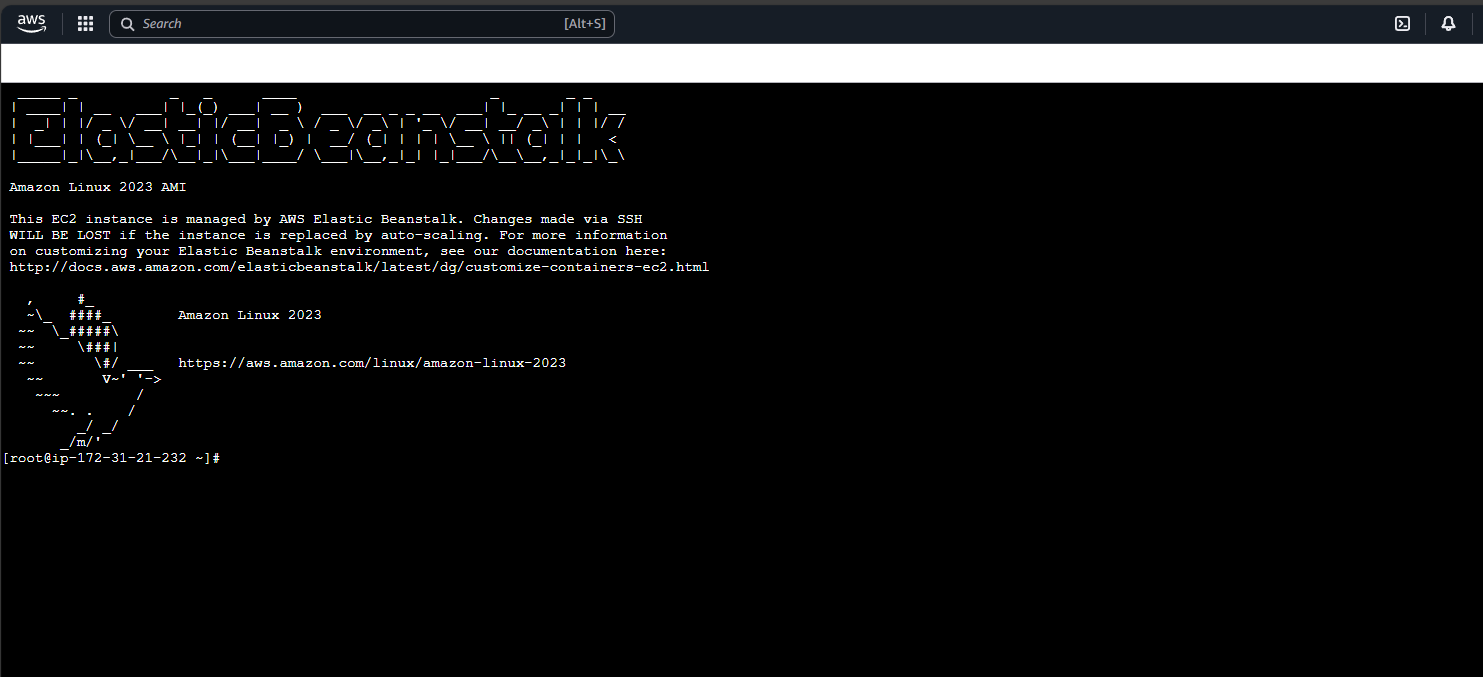
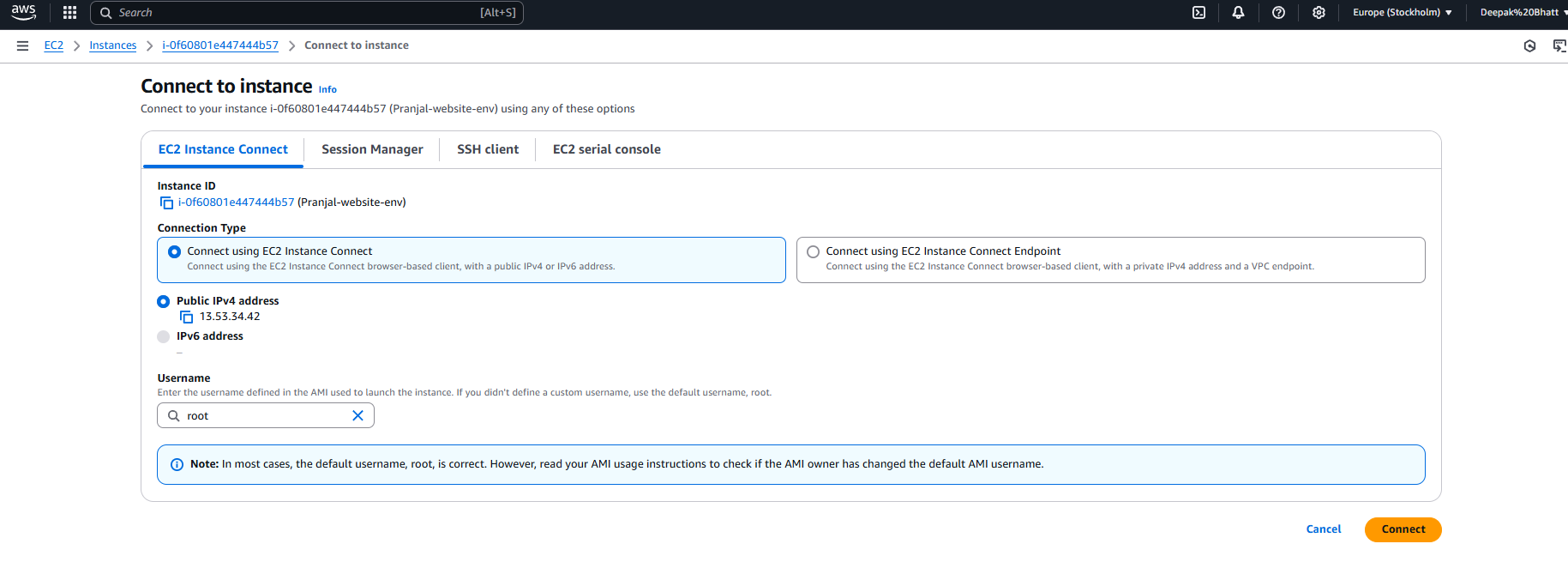


Step 10: Click the URL

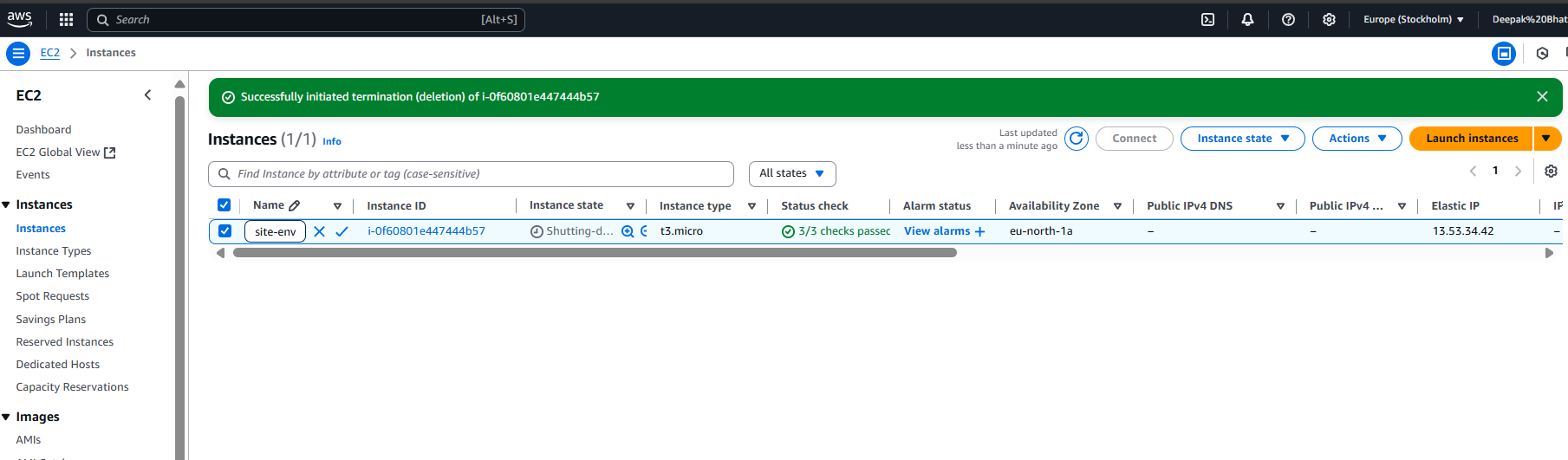


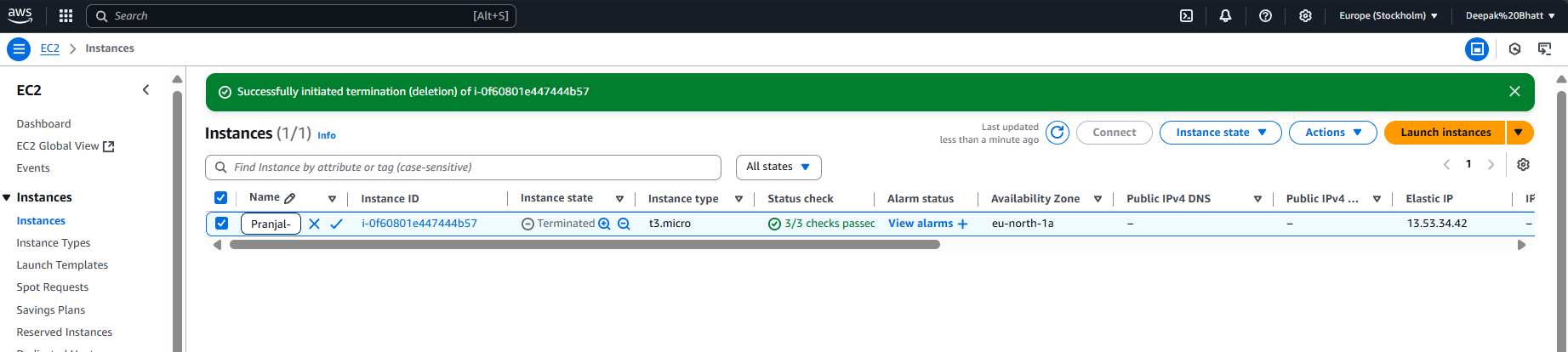






Step 11:Terminate the Environment





**B.1Question of Curiosity:**

# Q1: What are the benefits of Paas in cloud computing?

**Platform as a Service (PaaS)** offers several benefits in cloud computing:

1. **Faster Development & Deployment** – Developers can quickly build, test, and deploy applications without managing infrastructure.
2. **Scalability** – Automatically scales resources based on demand.
3. **Cost-Efficient** – Reduces operational costs by eliminating hardware and infrastructure maintenance.
4. **Built-in Security** – Provides automatic updates, security patches, and compliance features.
5. **Multi-Platform Support** – Supports different programming languages, frameworks, and databases.
6. **Automatic Load Balancing** – Ensures high availability and performance.
7. **Easy Integration** – Seamlessly connects with databases, APIs, and third-party services.
8. **Collaboration & Accessibility** – Enables remote teams to collaborate and access cloud resources from anywhere.

# Q2:What is Paas?

Platform as a Service (PaaS) is a cloud computing model where a cloud provider offers a complete development and deployment environment without requiring users to manage the underlying infrastructure.

Key Features of PaaS:

* + Provides runtime environments, databases, and development tools.
  + Manages servers, networking, and storage for developers.
  + Examples: AWS Elastic Beanstalk, Google App Engine, Microsoft Azure App Service.

Use Case:

A developer can deploy a web application on AWS Elastic Beanstalk without worrying about setting up servers, load balancers, or networking.

# Q3) what is elastic bean stack ?

AWS Elastic Beanstalk is a Platform as a Service (PaaS) offered by AWS that automates the deployment, scaling, and management of applications.

Key Features:

* + Supports multiple languages: Node.js, Python, Java, .NET, PHP, Ruby, Go.
  + Automatically provisions EC2 instances, load balancers, databases, and other AWS services.
  + Provides a fully managed environment for deploying web applications and APIs.

Allows developers to focus on writing code while AWS handles infrastructure management.

Example Use Case:

A company can deploy a Node.js application on Elastic Beanstalk, and AWS will handle scaling, monitoring, and load balancing automatically.

**B.2 Conclusion:**

Through this hands-on implementation of Platform as a Service (PaaS) using AWS Elastic Beanstalk, I have gained practical knowledge about deploying, managing, and scaling web applications in a cloud environment. This experiment helped me understand how PaaS simplifies application deployment by handling infrastructure, scaling, and resource management.

I successfully deployed a sample web application, monitored its health, and observed how AWS Elastic Beanstalk automatically manages load balancing, scaling, and provisioning of instances. Additionally, I explored how Elastic Beanstalk integrates with other AWS services like EC2, S3, RDS, and CloudWatch, ensuring seamless cloud operations. The experiment also highlighted how PaaS improves cost efficiency by eliminating the need for manual server management, making application development more efficient and scalable. Overall, this learning experience reinforced my understanding of cloud computing models and how PaaS accelerates deployment while reducing infrastructure complexities.