Mini-Project-2 :

Deploying an Application on AWS Elastic Beanstalk with IAM Roles and Environment Swap

Statement :

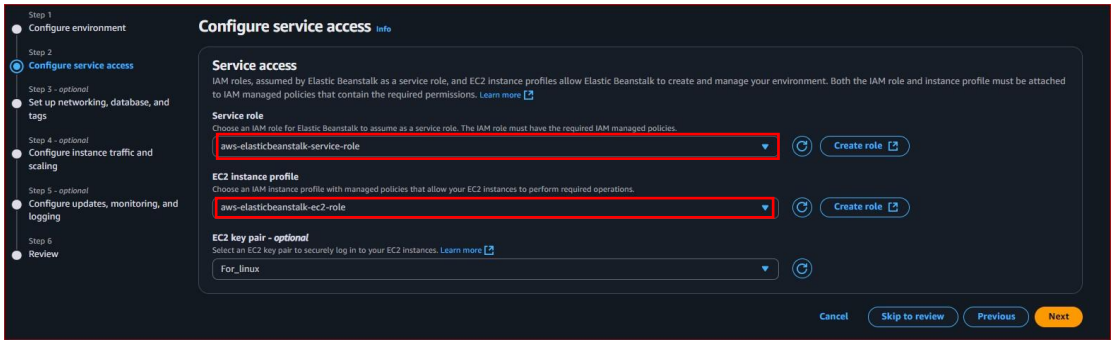
- Deploying applications on the cloud requires not only hosting infrastructure but also proper access management. The goal of this project was to deploy a sample application on **AWS Elastic Beanstalk** while ensuring secure and controlled permissions using **IAM roles**.

Elastic Beanstalk :

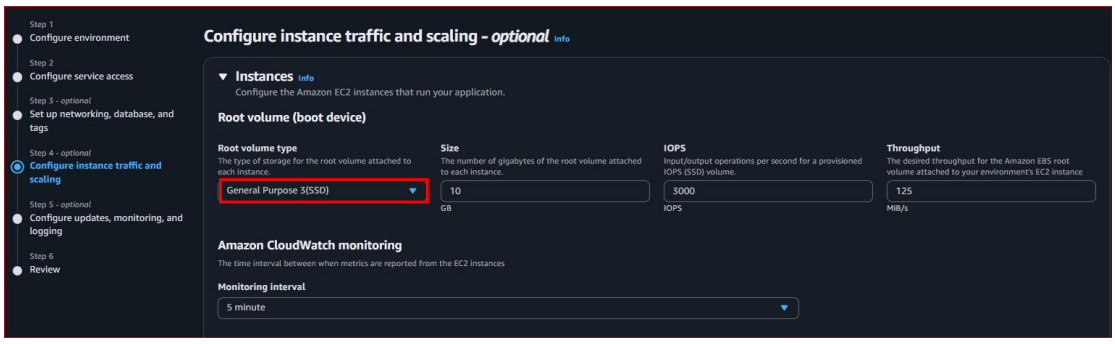
AWS Elastic Beanstalk is a Platform as a Service (PaaS) that helps developers quickly deploy and manage applications without handling the complexity of the underlying infrastructure.

Steps for creation of beanstalk :

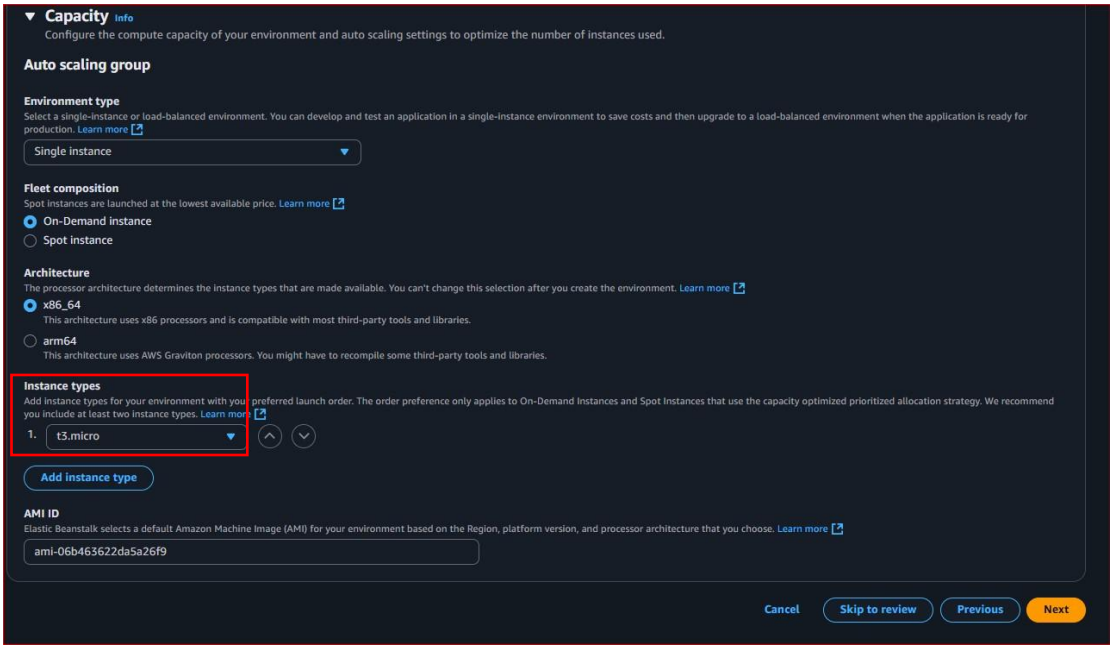
1. During the creation of Beanstalk selected roles which I has created related to requirement :



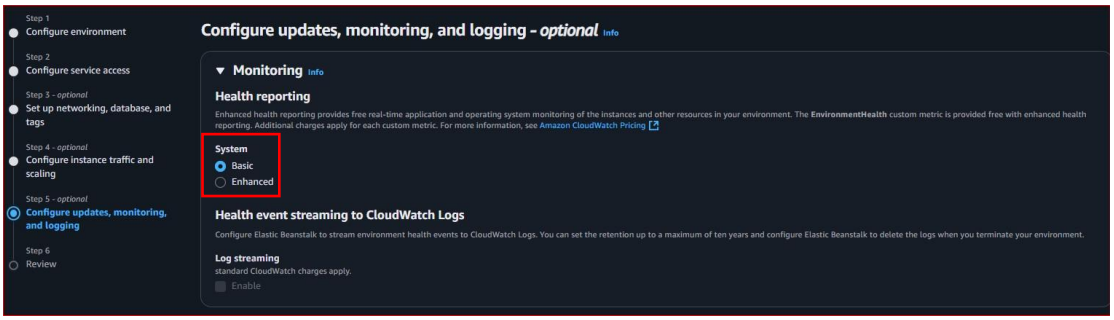
2. Skipped step 3 without modifying anything and in step 4 Added Root General Purpose3(SSD)



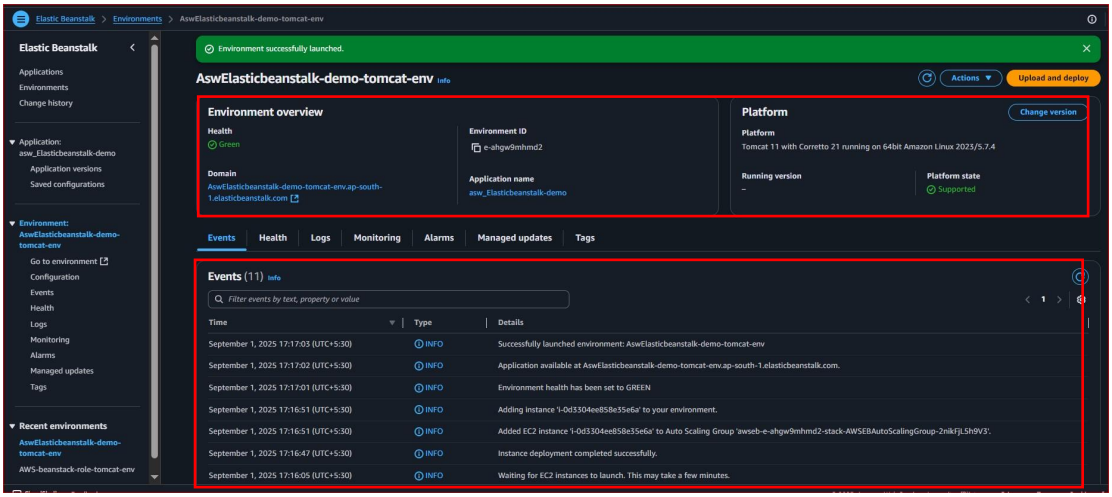
3. Selected Instance type :



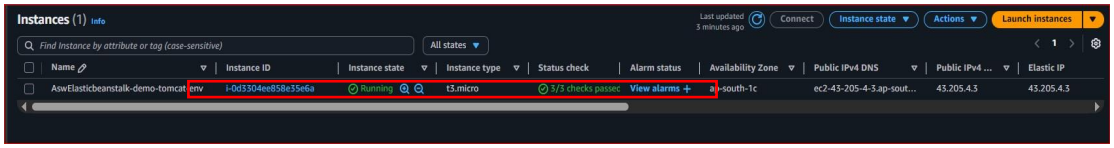
4. Select system in Health reporting :



5. Review and create



6. Successfully launched instance after creation of Beanstalk



When the environment was created in **AWS Elastic Beanstalk**, several automated steps took place in the background. The Events section shows the sequence of actions:

-**Environment Launch Initiated** – Beanstalk started creating the environment AwsElasticbeanstalk-demo-tomcat-env. (There was my typing mistake at giving name)

-**Instance Provisioning** – An **EC2 instance** was launched to host the application.

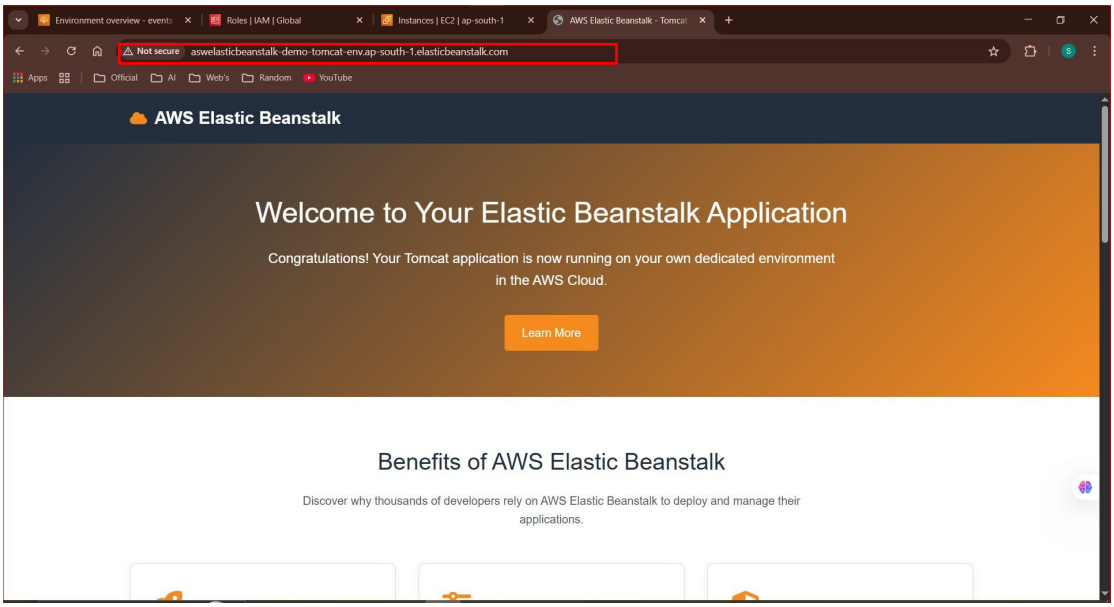
-**Auto Scaling Group Update** – The new EC2 instance was automatically added to the Auto Scaling Group, ensuring scalability and high availability.

-**Deployment Completed** – Once the EC2 instance was ready, Beanstalk deployed the application code onto it.

-**Application Accessible** – The application was made publicly available at the generated **Elastic Beanstalk domain URL**.

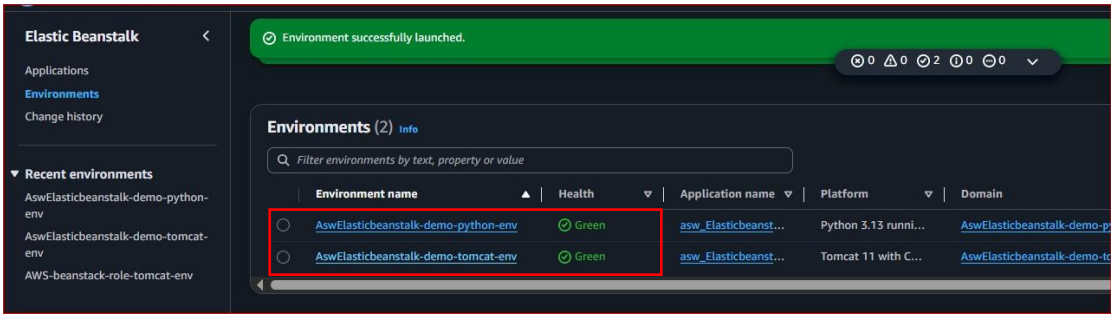
-**Health Check Performed** – Beanstalk automatically ran health checks, and the environment health was set to **GREEN**, meaning the application is running without issues.

6. Copy DNS name and Search on browser :



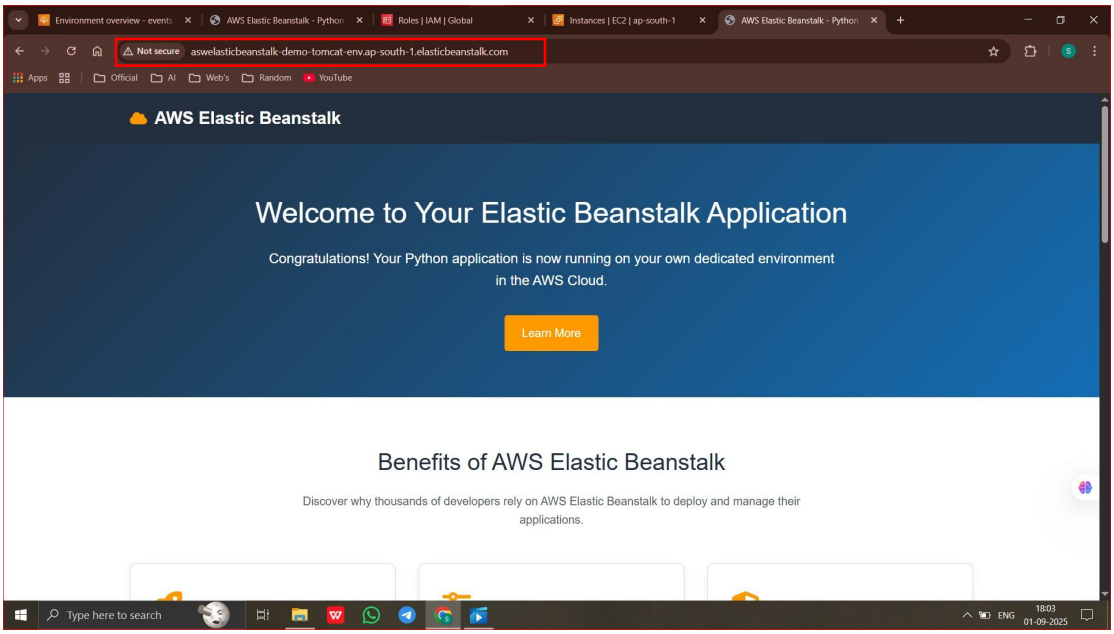
And finally our application is working on browser
Application was used tomcat, Now we swap environment of this application

7.Launched an environment for python application



8.Finally Swapping successfully :

Swap in Elastic Beanstalk changes the traffic from your **old environment (Blue)** to your **new environment (Green)** by Keeping their URLs same, giving you a smooth and risk-free way to update applications.



Stay Connected :

https://www.linkedin.com/in/siddheshwar-shinde-90b73a353

https://github.com/codexshwar

Thank you .