SVKM'S NMIM'S Nilkamal School of Mathematics, Applied Statistics & Analytics
Master of Science (Data Science)

Practical-2 Platform as a service using AWS.

Date:-23/01/2024 **Submission Date:- 30/01/2024**

Writeup:-

- **Platform as a service**
  AWS Elastic Beanstalk is the PaaS offering from AWS that makes it easy to deploy and manage applications in the cloud without worrying about the infrastructure. Developers can simply upload their code and Elastic Beanstalk automatically handles provisioning, load balancing, auto-scaling and monitoring.

- **Elastic Beanstalk**
  Elastic Beanstalk is a PaaS service offered by AWS to deploy and scale web applications quickly without worrying about the infrastructure. It automatically handles capacity provisioning, load balancing, scaling and application health monitoring. Developers just have to upload their code and Elastic Beanstalk will deploy it on AWS infrastructure like EC2, auto scale it and monitor it.

  Why Elastic BeanStalk
  i. It supports multiple languages like Java, Python, Go etc. and platforms like Docker.
  ii. Beanstalk integrates well with other AWS services like EC2, S3, RDS etc.
  iii. The main benefits are fast and automated application deployment and management, multiple environments, auto scaling, and cost efficiency.

- **Components of beanstalk**

  i. Application: This is the actual web application code packaged into a zip and uploaded to Elastic Beanstalk.
  ii. Application Version: Each deployment of code is an application version. Rollbacks can be done to previous versions.
  iii. Environment: This is a version of the application running on AWS resources. We can create multiple environments like dev, test, prod etc from the same application.

iv. Configuration Templates: These allow customization of the AWS resources powering an environment like EC2 instance type, autoscaling settings etc.

v. Events: Important lifecycle events like deployments, scaling etc are logged for debugging.

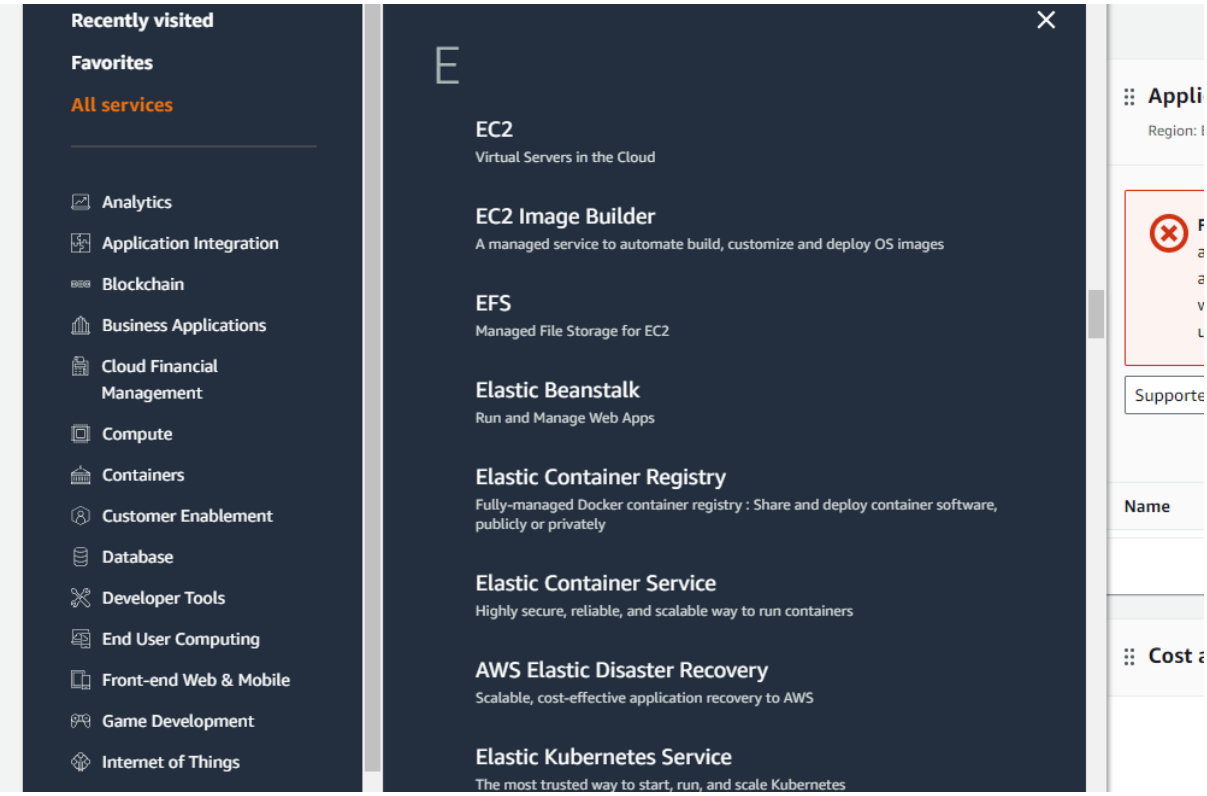
- **IAM**

i. IAM allows managing users, roles and permissions to access AWS services and resources.
ii. Users can be created and assigned granular permissions policies.
iii. Roles can be created with permissions and then assigned to AWS resources like EC2 instances.
iv. Policies define the permissions like which AWS actions can be performed on which resources.
v. IAM is important for security, access control and compliance in AWS.

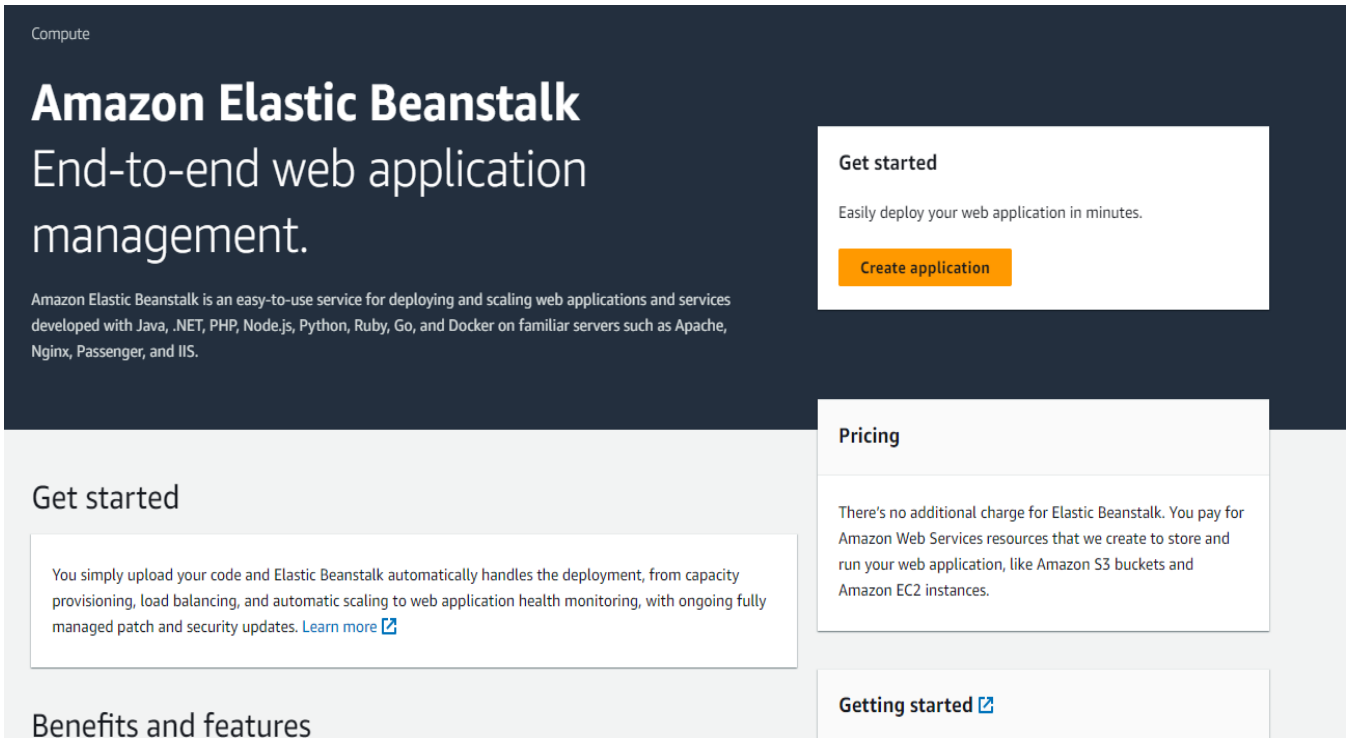
- Implement PAAS using elastic beanstalk for the following.
  1. Server
  2. Java
  3. Python
  4. Node.js

For Python

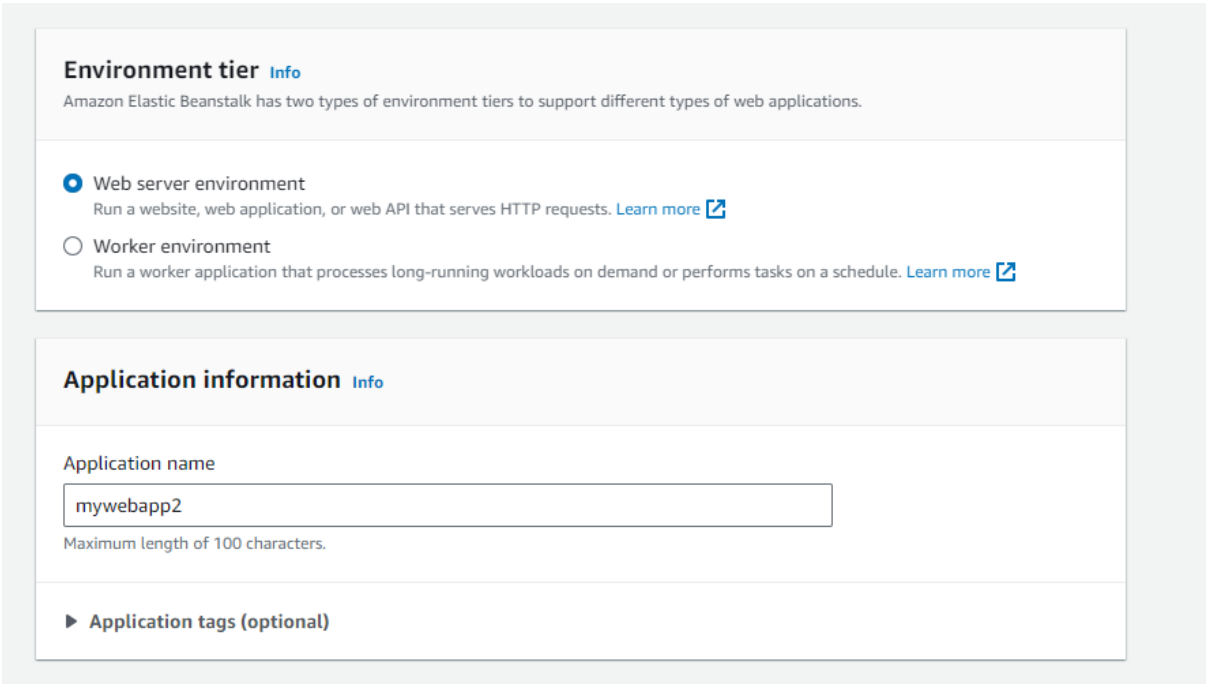
Step 1:- Select Elastic BeanStalk



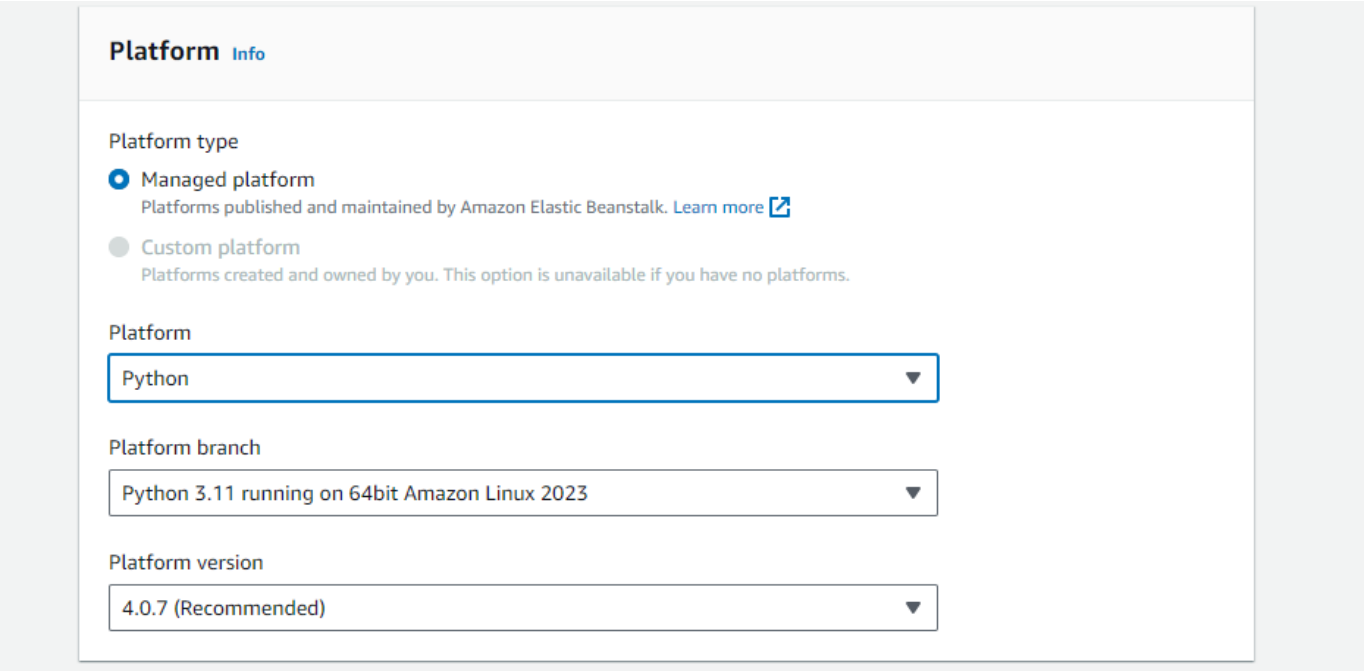
Step 2- Create a new application



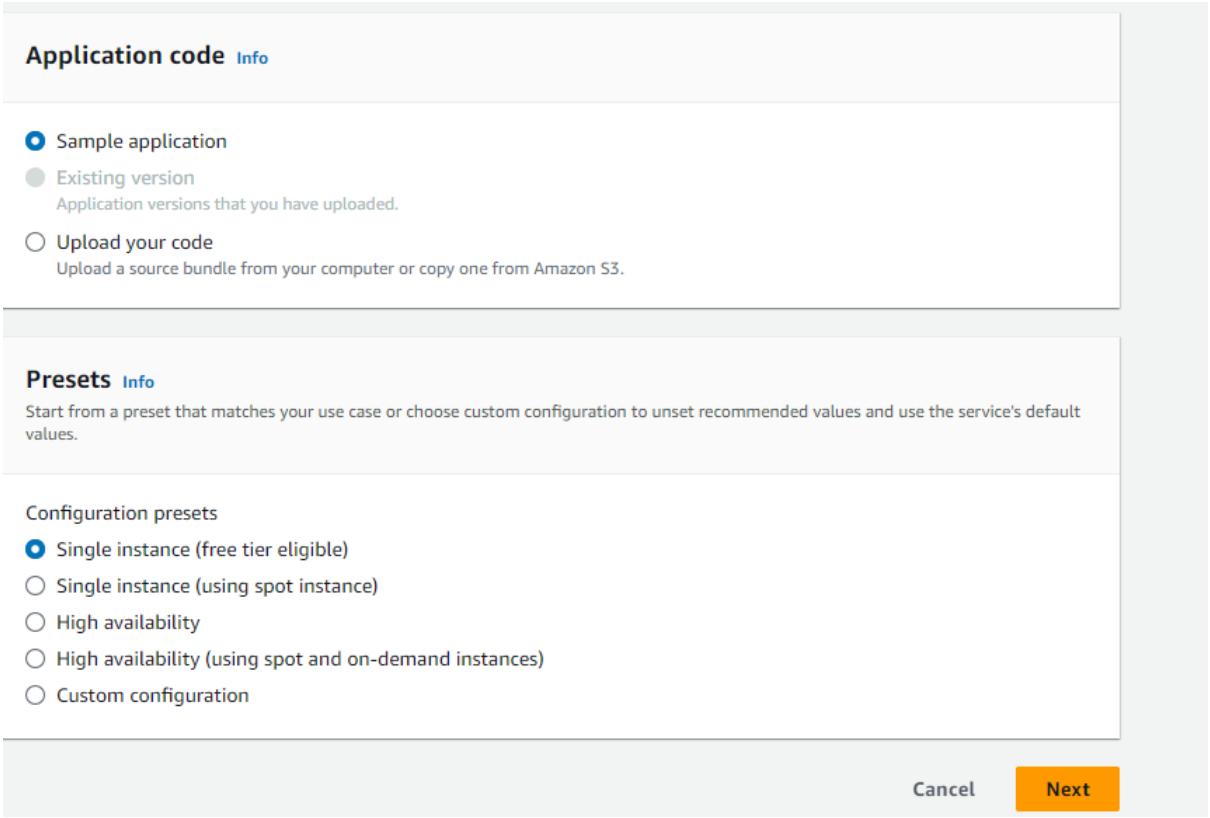
Step 3- Provide a new Name for the Application



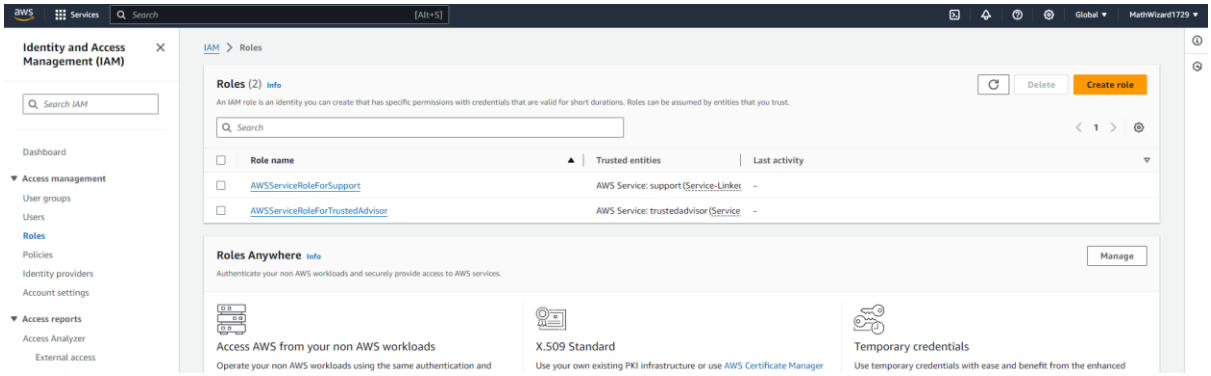
Step 4- Choose the Platform as Python



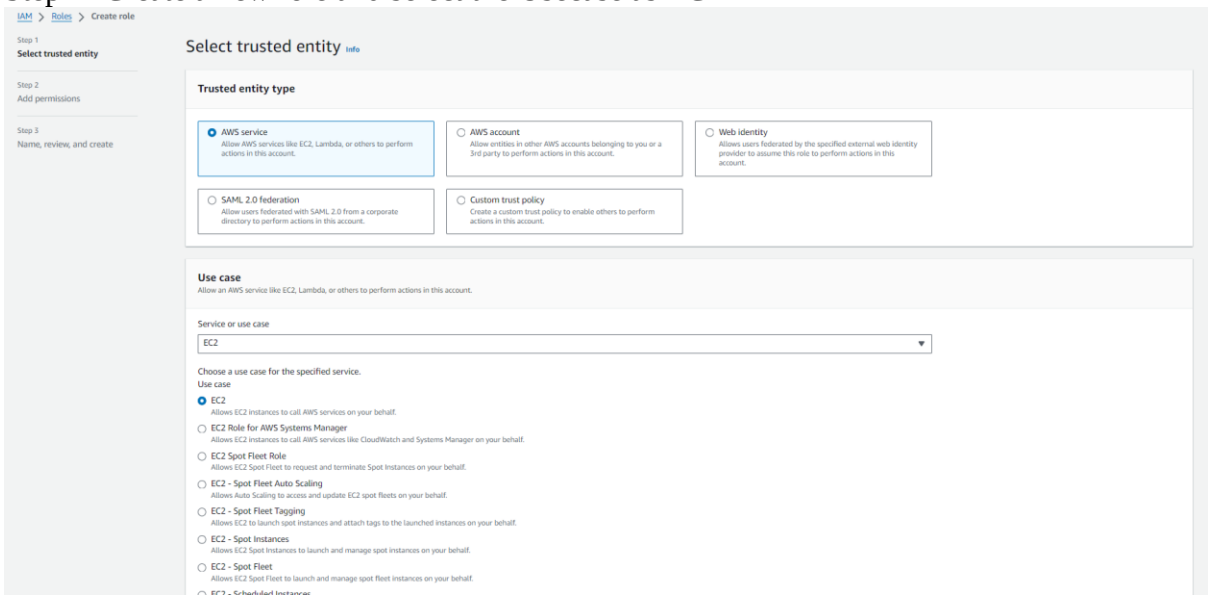
Step 5- Keep it as Single Instance



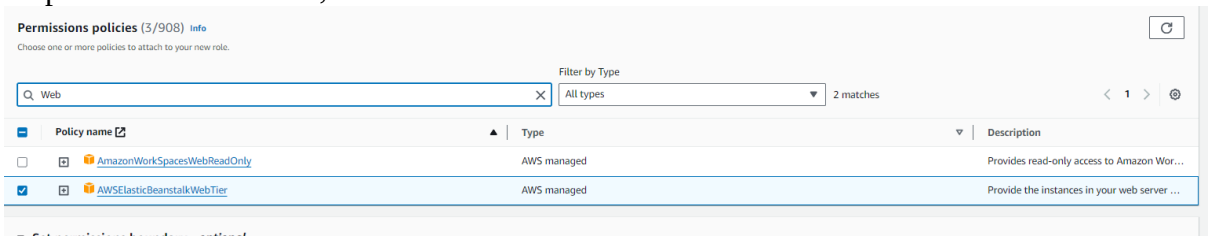
Step 6- Create a new Role under IAM(Identity Access Management)

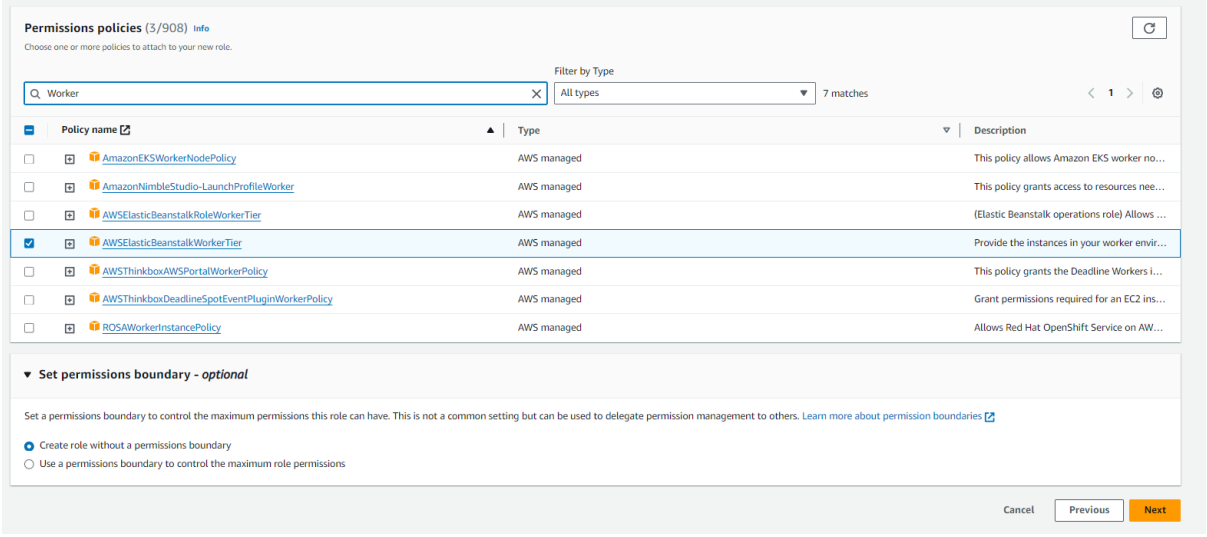


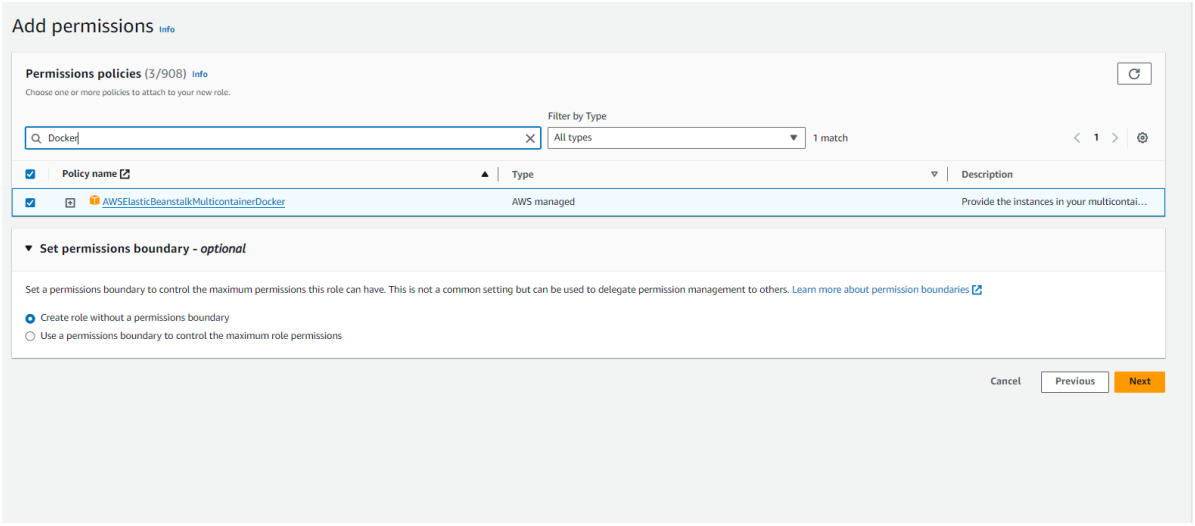
Step 7- Create a new role and select the Usecase as EC2



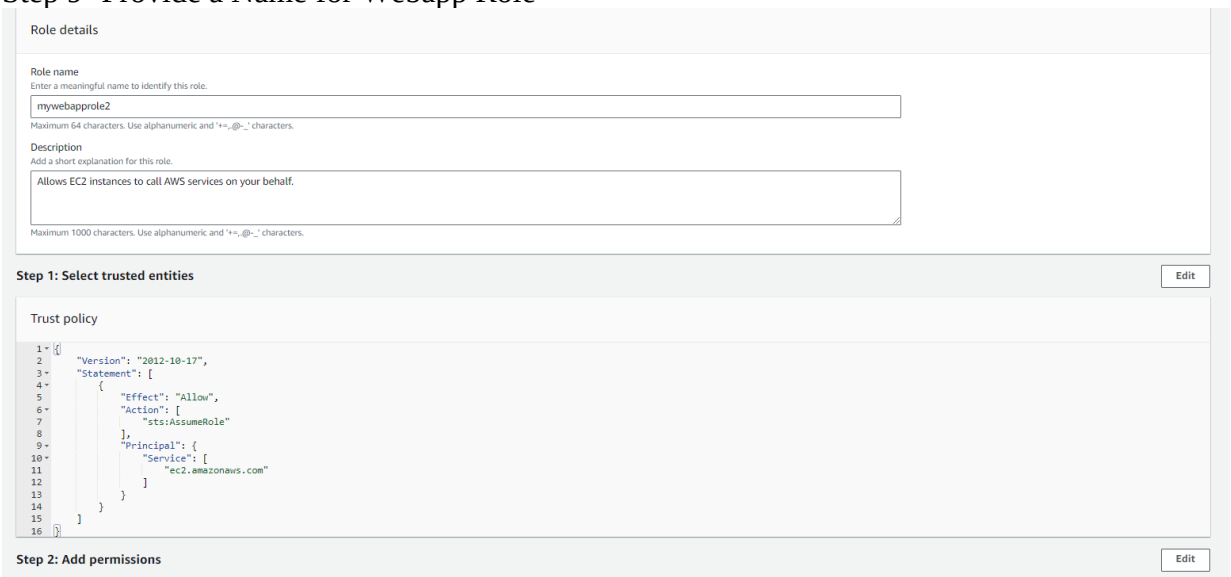
Step 8- Select Web Tier, Worker Tier and MultiContainer docker tier



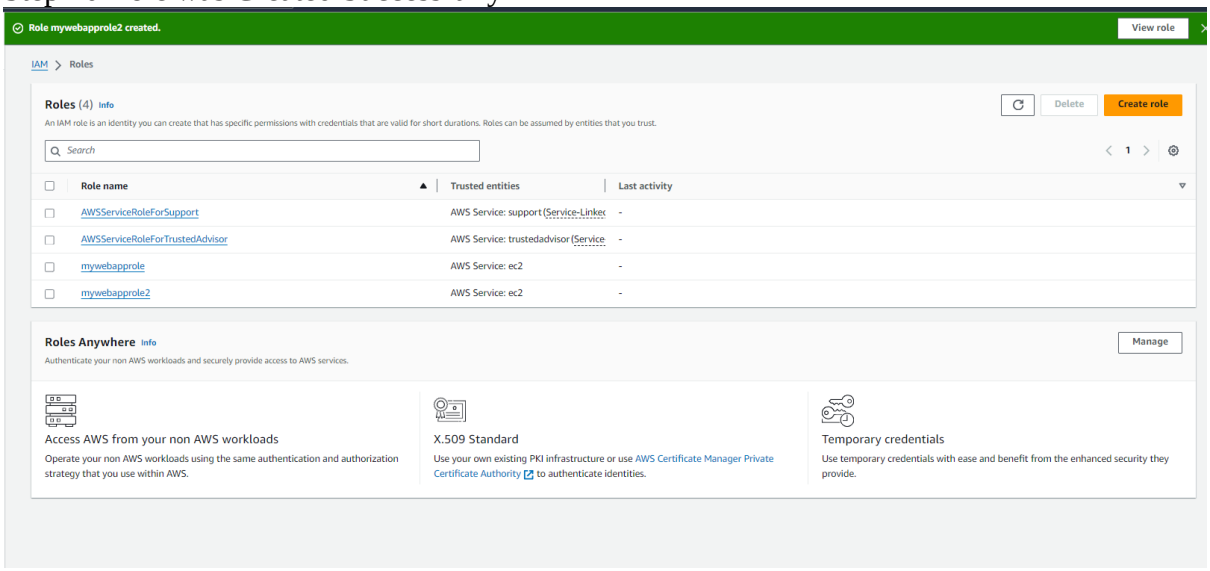




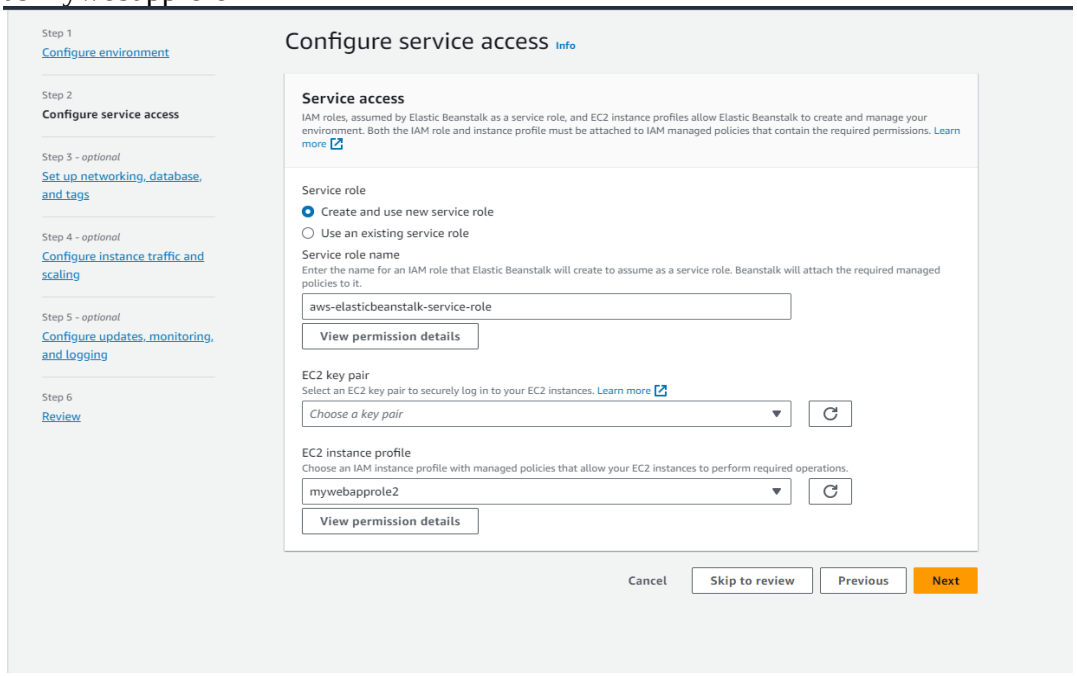
Step 9- Provide a Name for Webapp Role



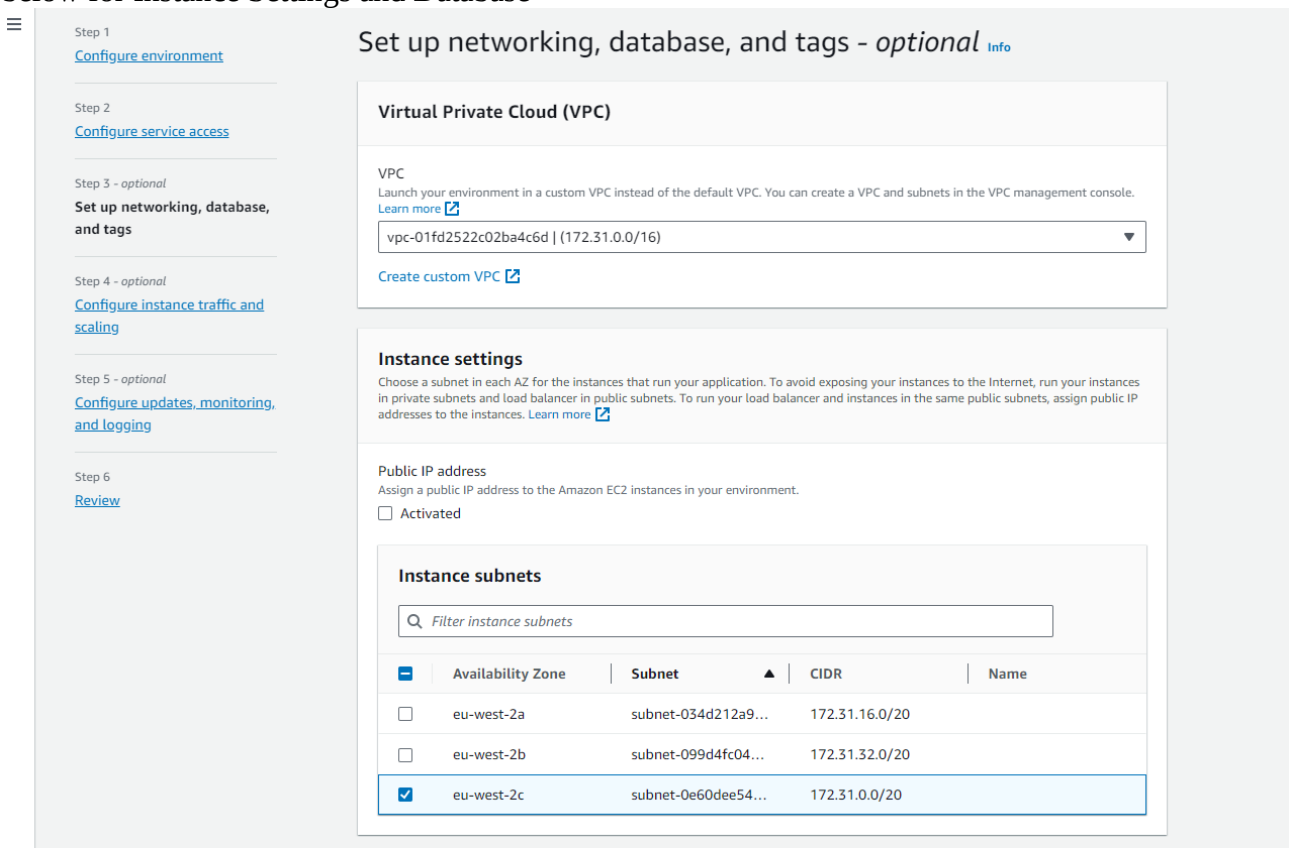
Step 10- role was Created Successfully



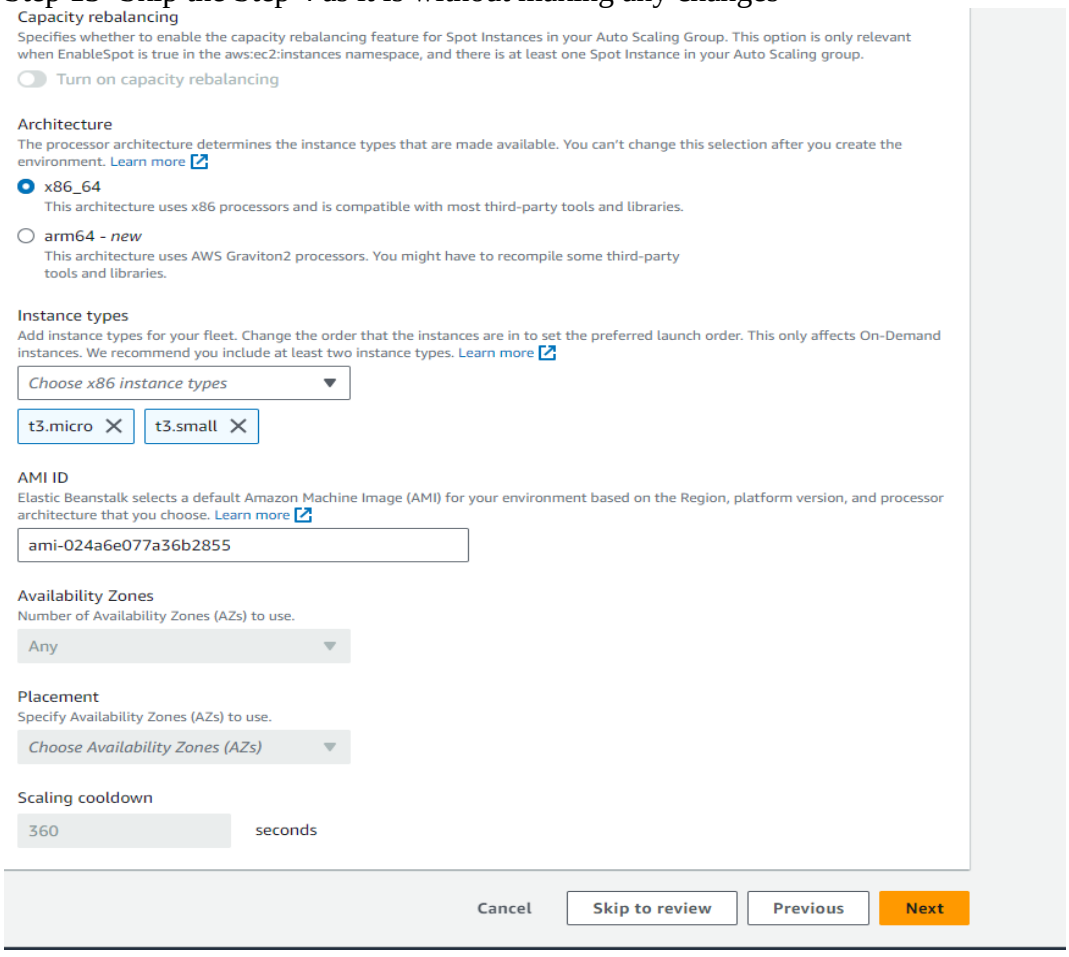
Step 11- For this Access Create a new use case and service role and for Instance profile select as mywebapprole2



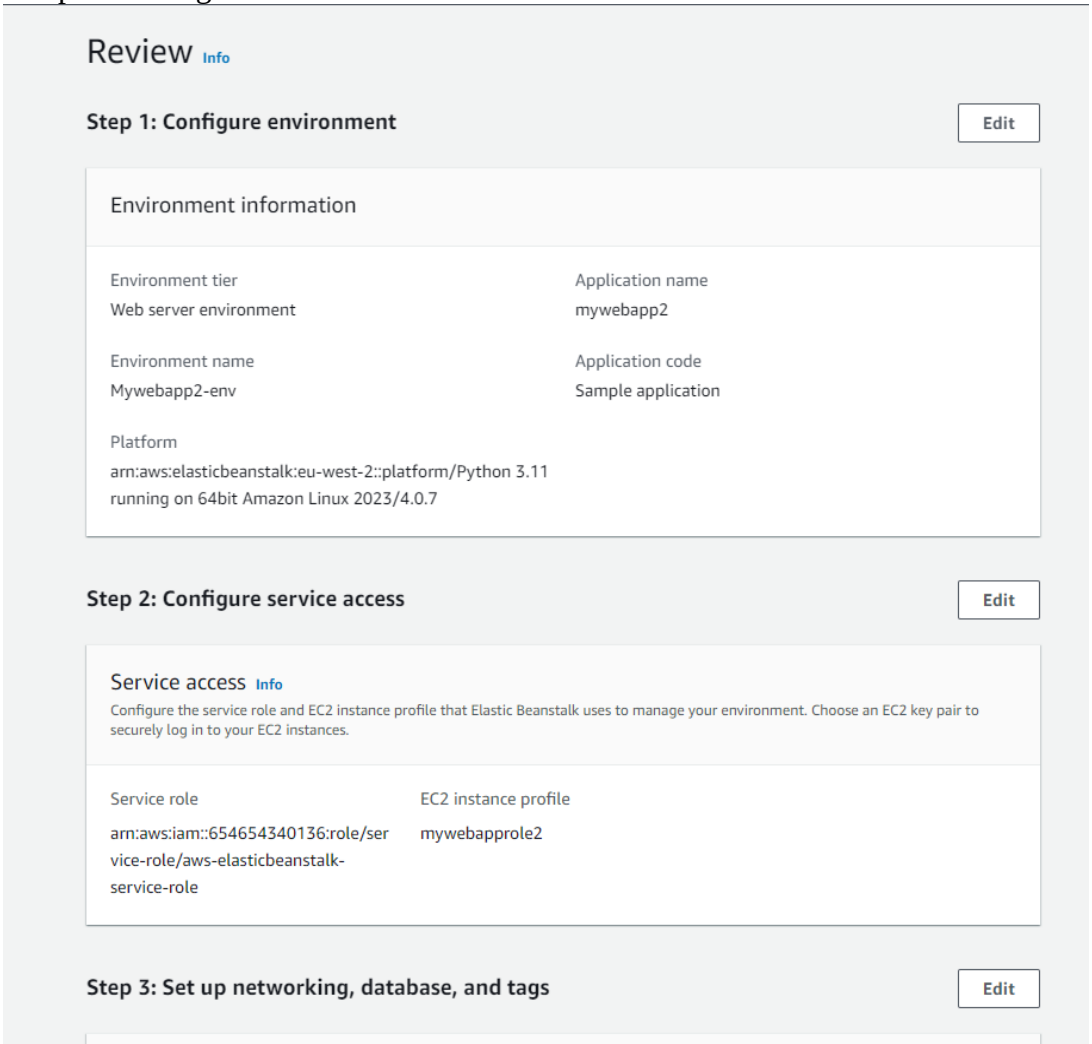
Step 12- Select the VP name as given and from the IP address provided select the same down below for Instance Settings and Database



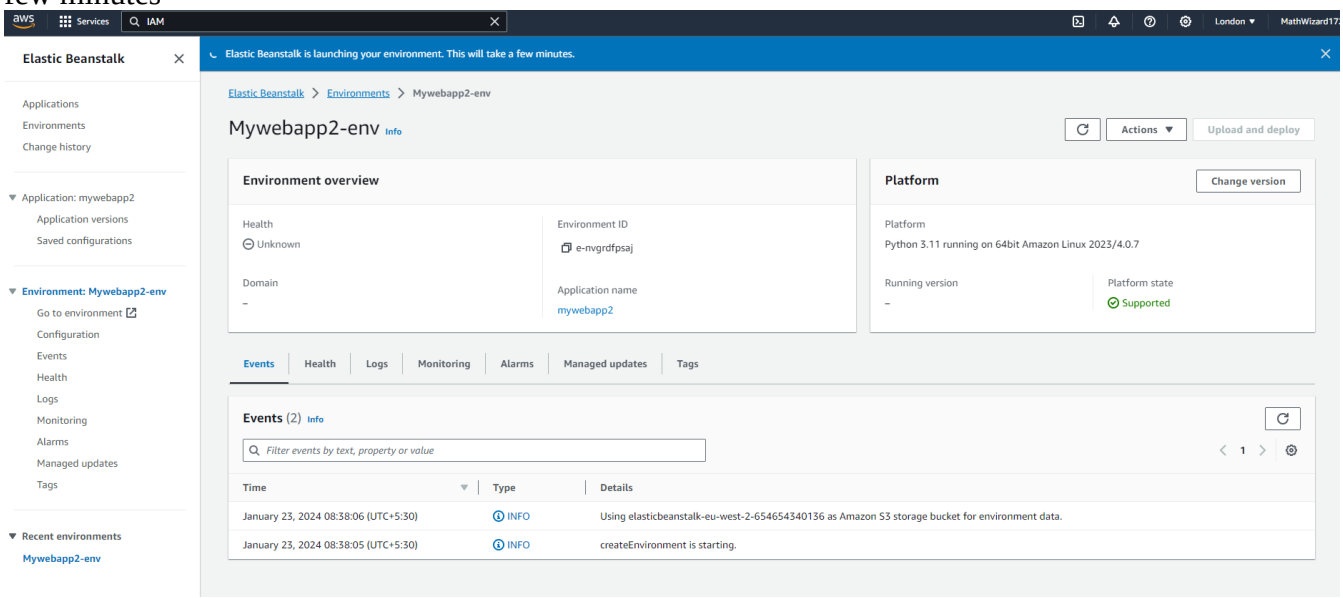
Step 13- Skip the Step 4 as it is without making any changes



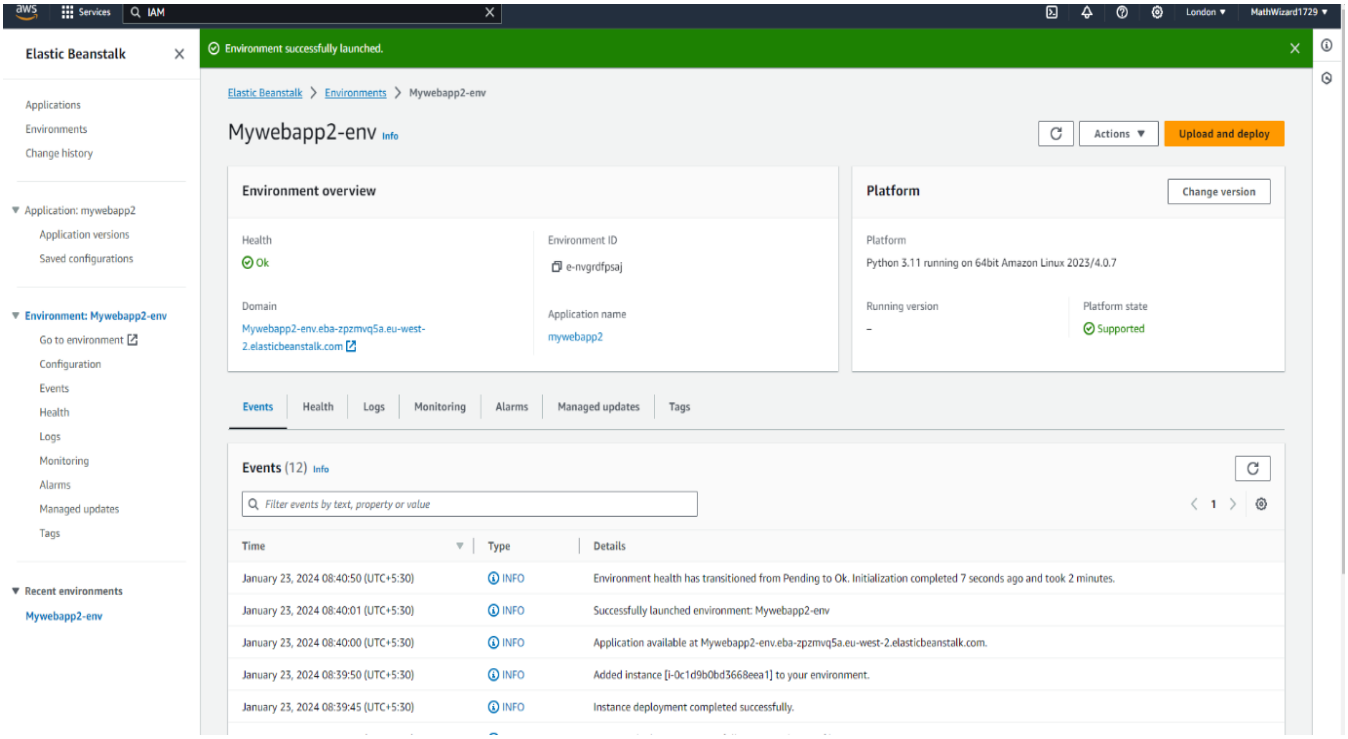
Step 14- Skip the Step 5 Section and directly go to Review Page where you can view the complete Changes made.and Submit it



Step 15- After the review the Elastic BeanStalk will run environment and it will take some few minutes



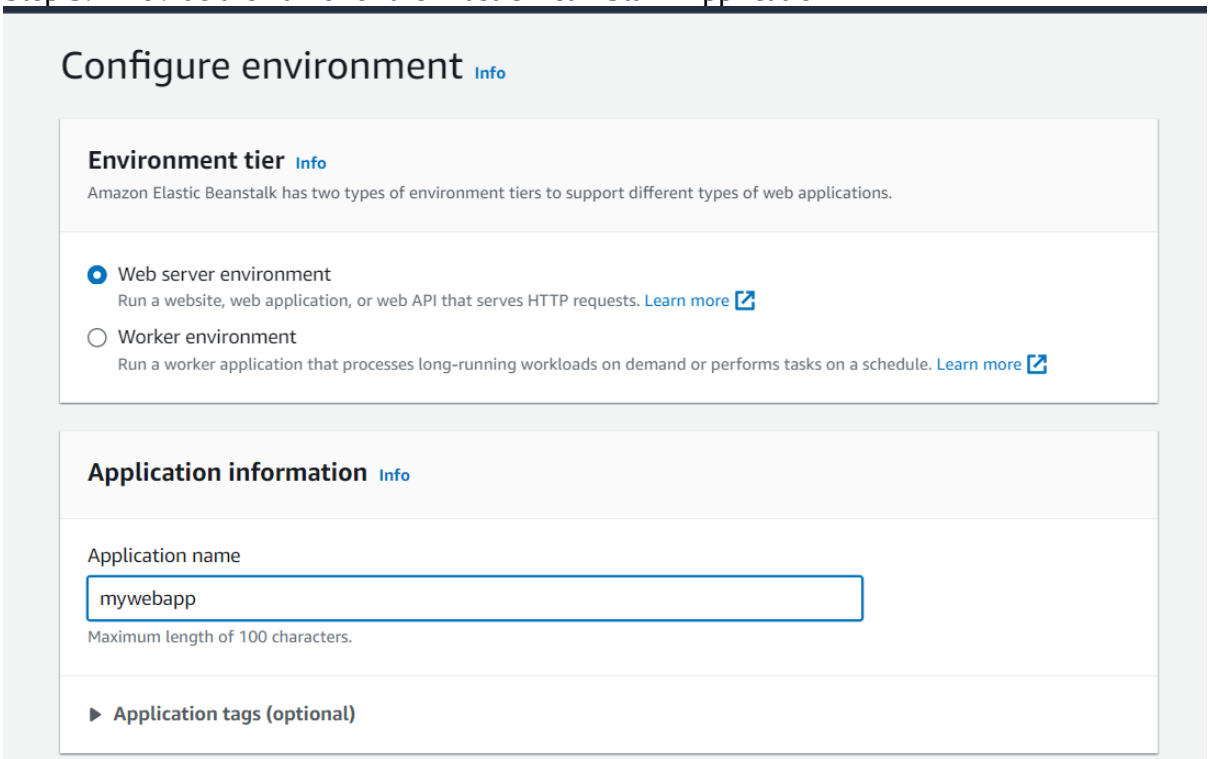
Step 16- Environment is successfully launched



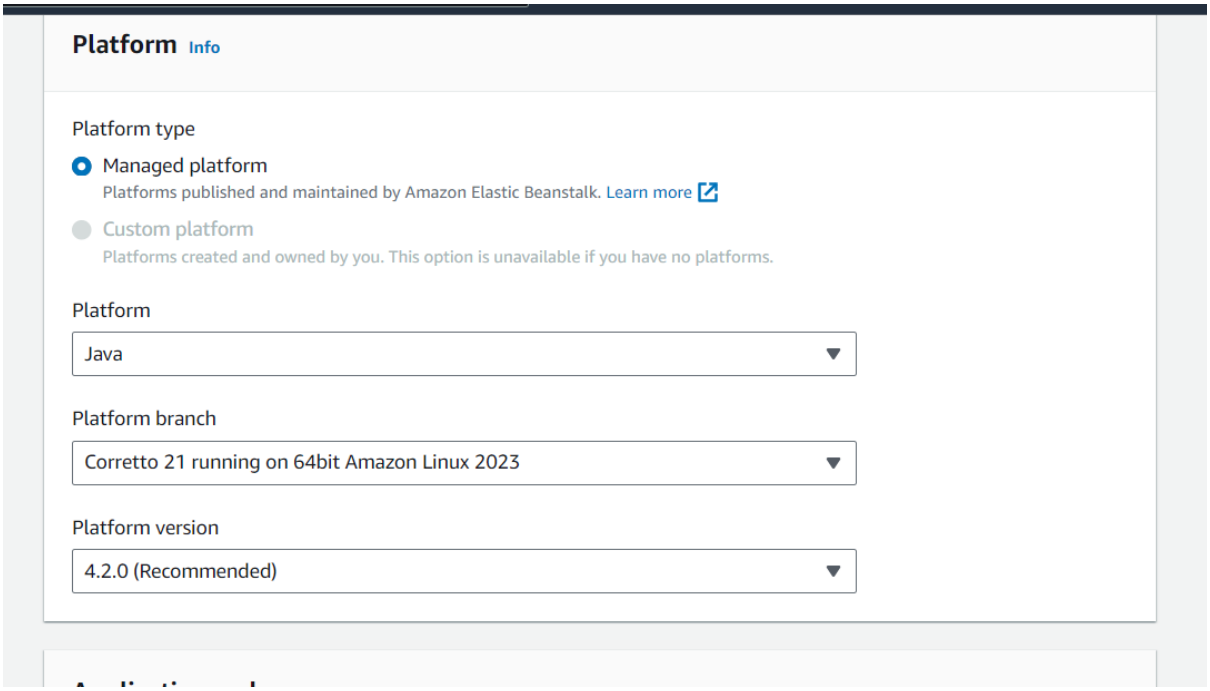
For Java

From Step 1 to Step 4 we repeat the process and from Step 4 we select Java and continue the steps

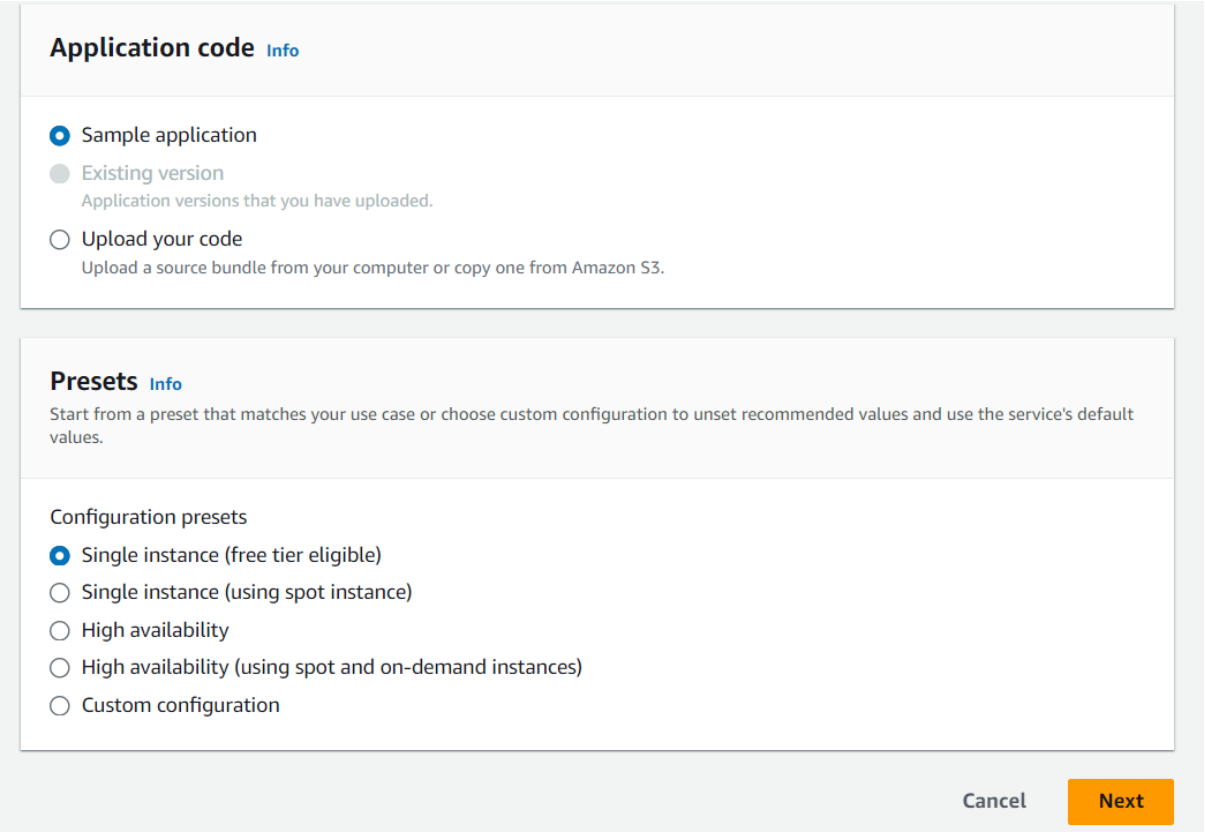
Step 3:- Provide the name for the Elastic Bean Stalk Application



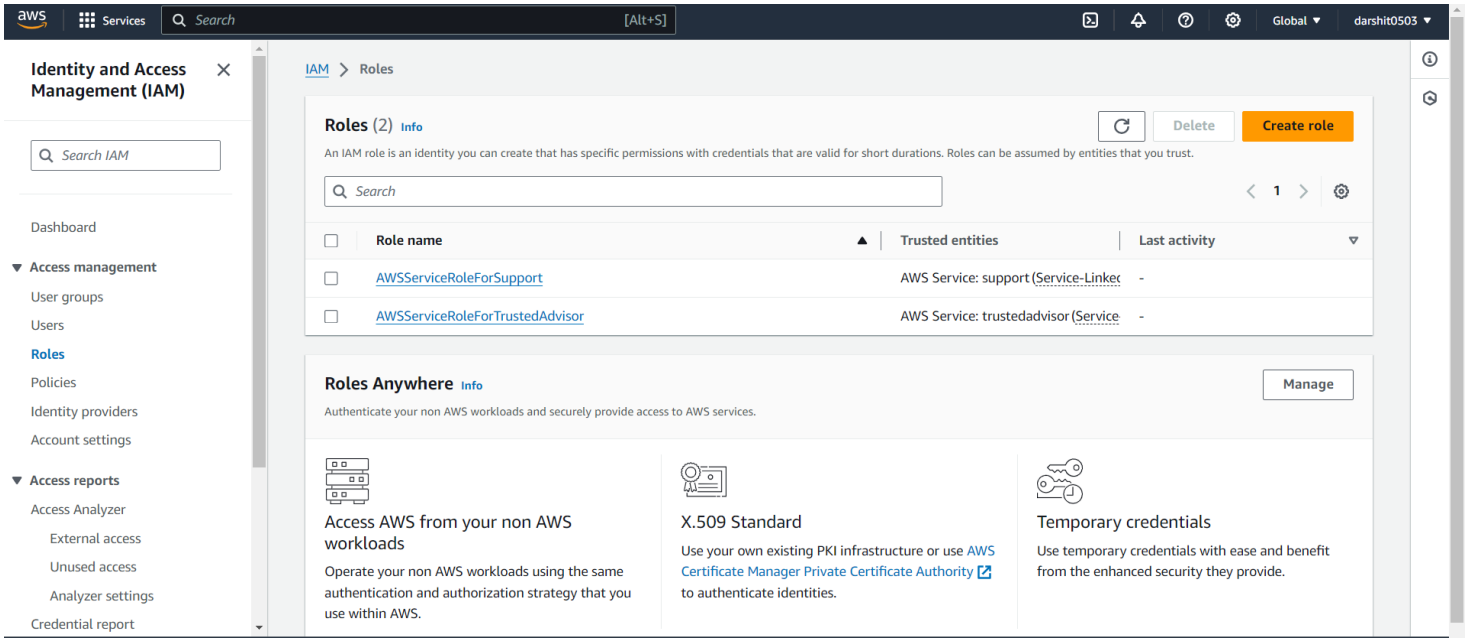
Step 4- Select Platform as Java and go on further



Step 5-Keep it as Single Instance.

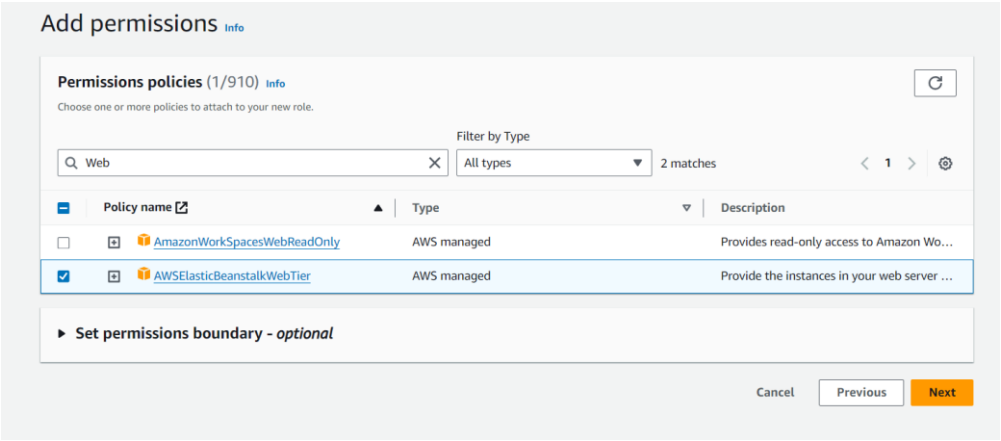


Step 6- Create Role under IAM User

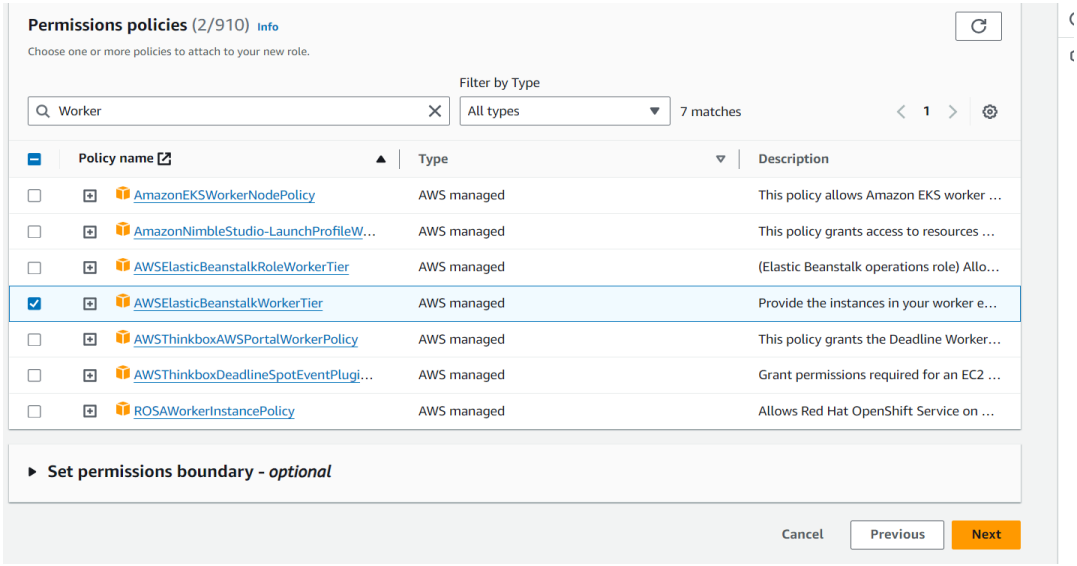


Step 7- Select Usecase as EC2 and select Next and from the following list Select3 options

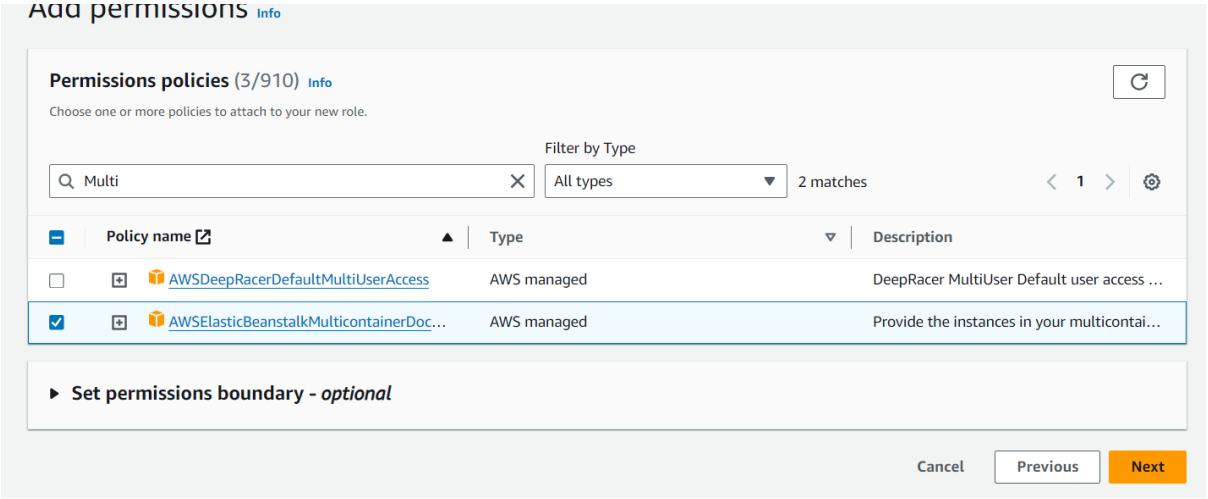
a) WebTier



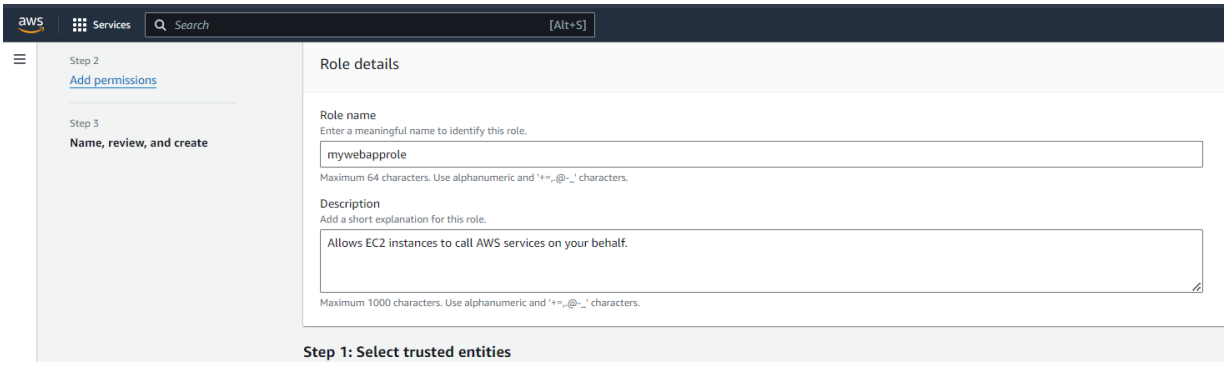
b) WorkerTier



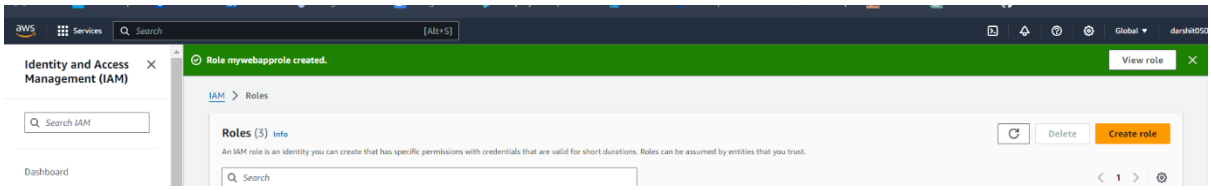
c) MultiContainerDocker list



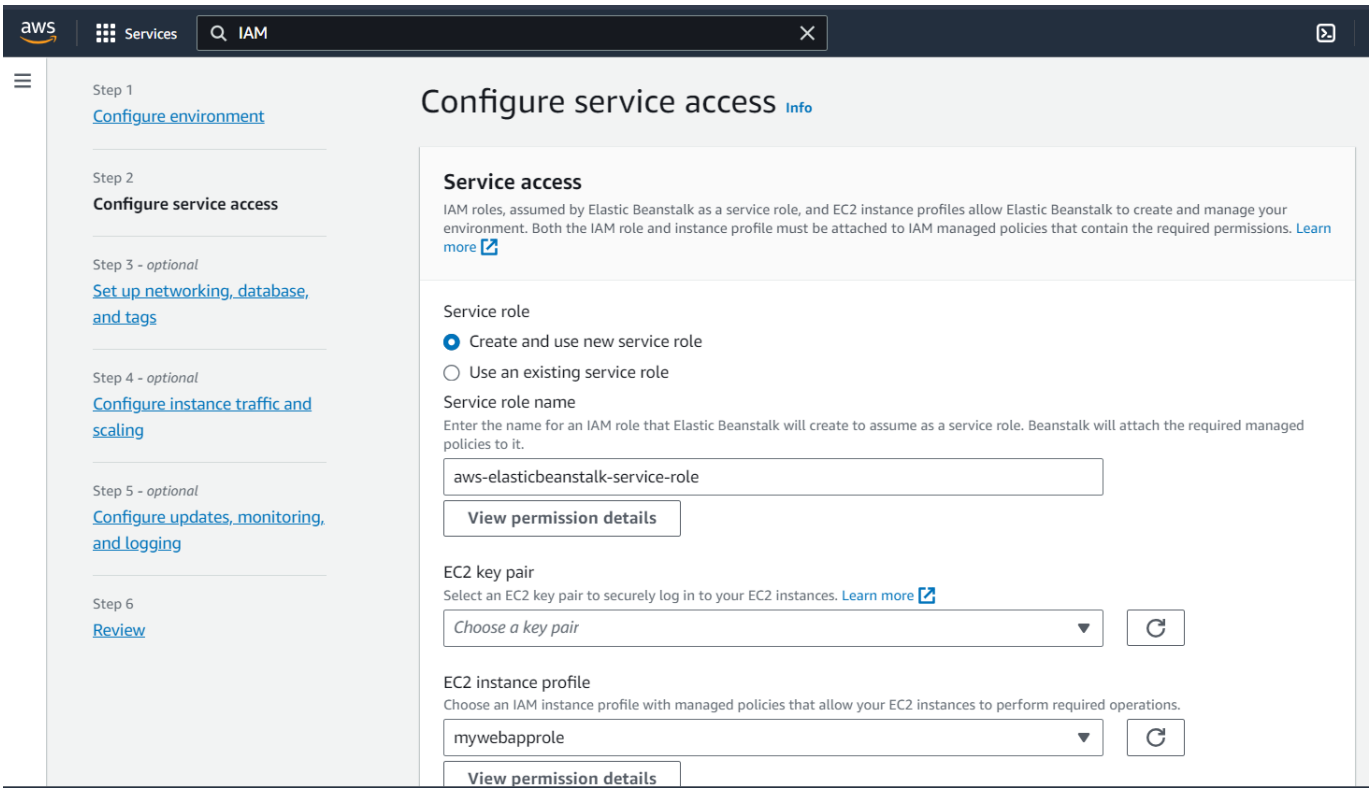
Step 8 - Provide the name for the Role



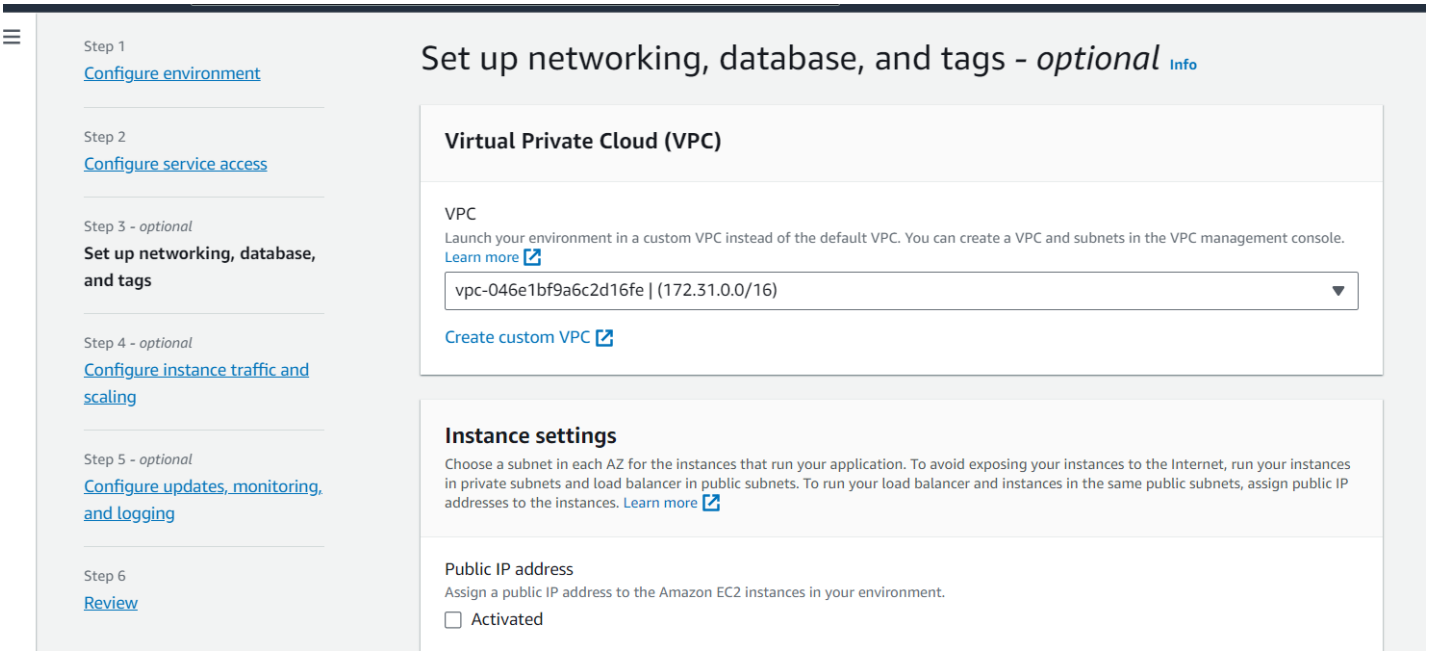
Step 9- The Role was created Successfully



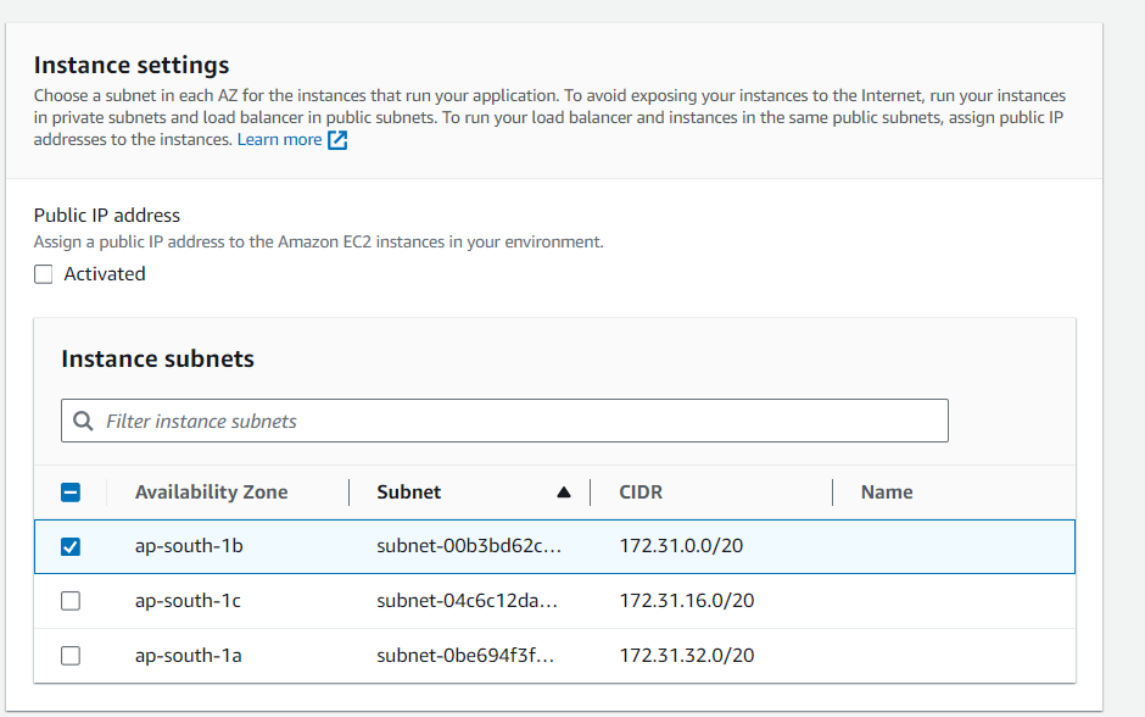
Step 10- For the Configure Service Access select new service Role and under domain EC2 Instance Role select mywebapprole which was created above



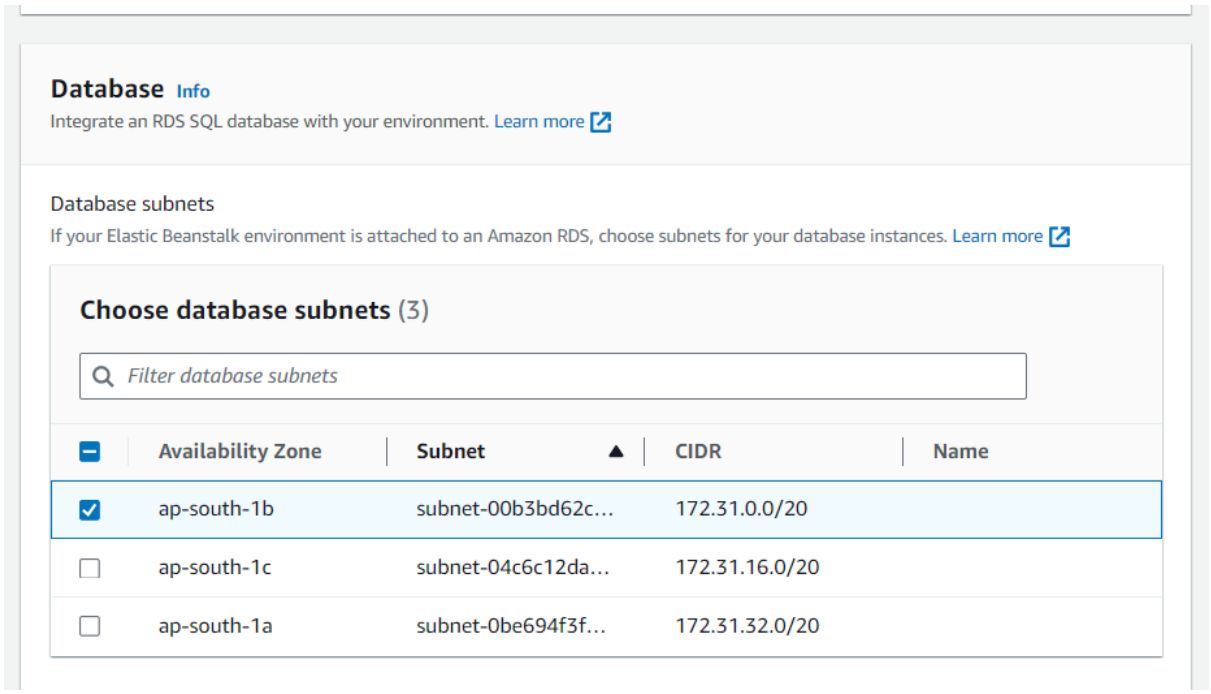
Step 11- Select VPC which was been provided there



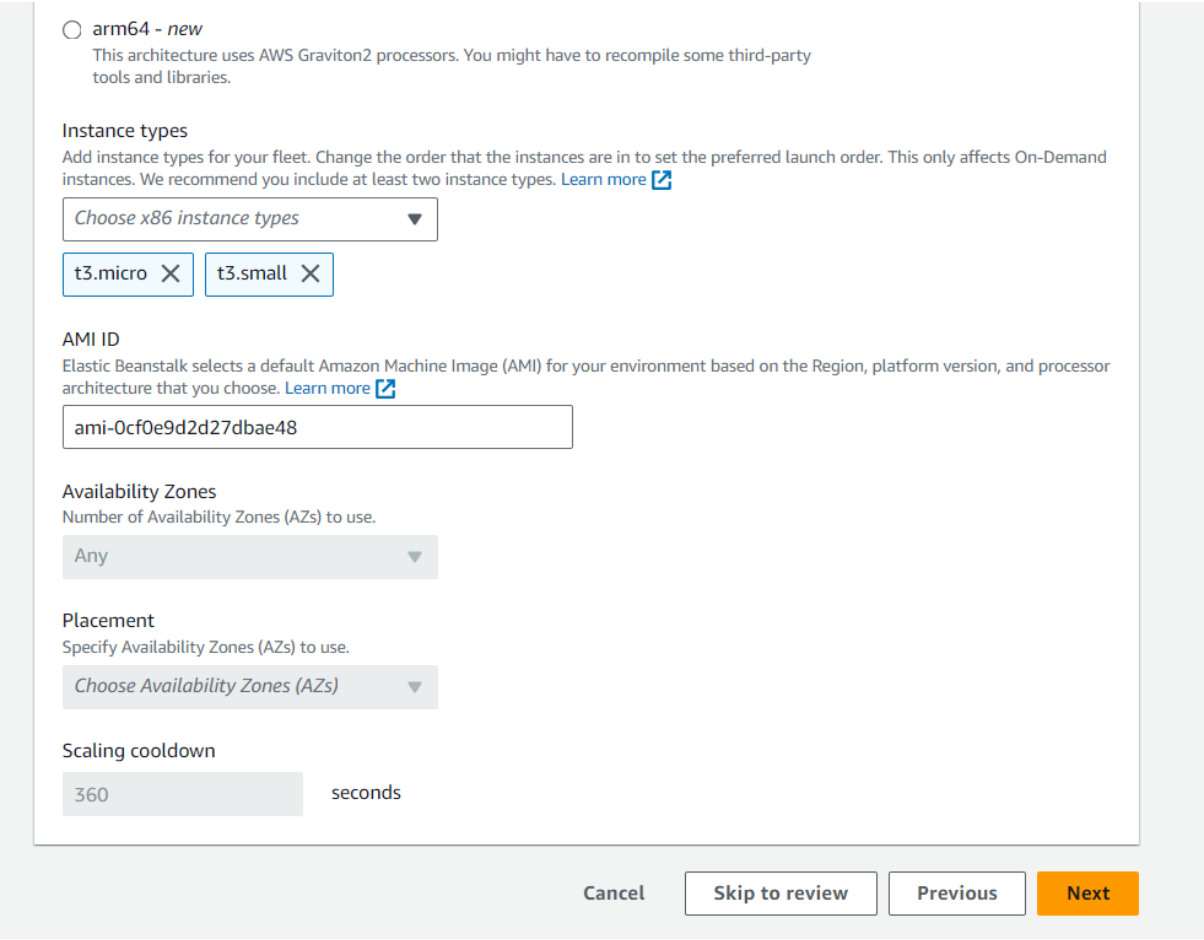
Step 12- Select the IP Address Same as above which was been provided in VPC in Instance and Database



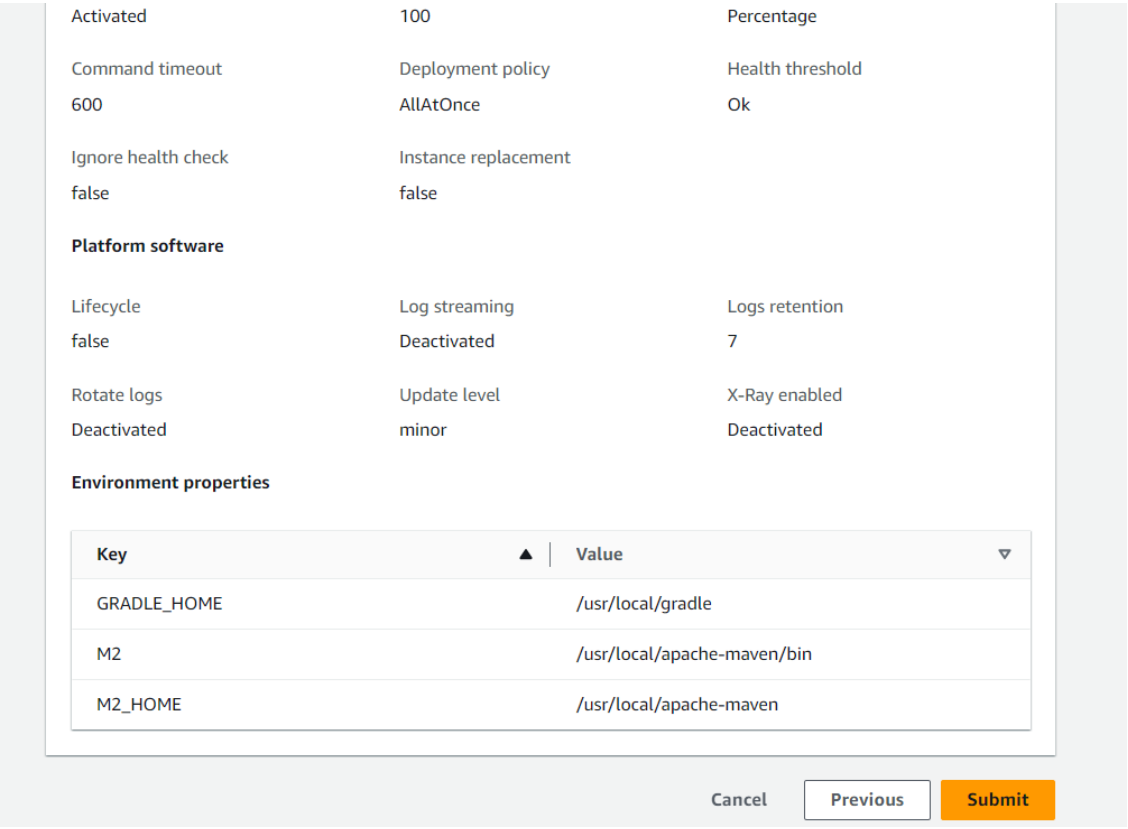
For Database



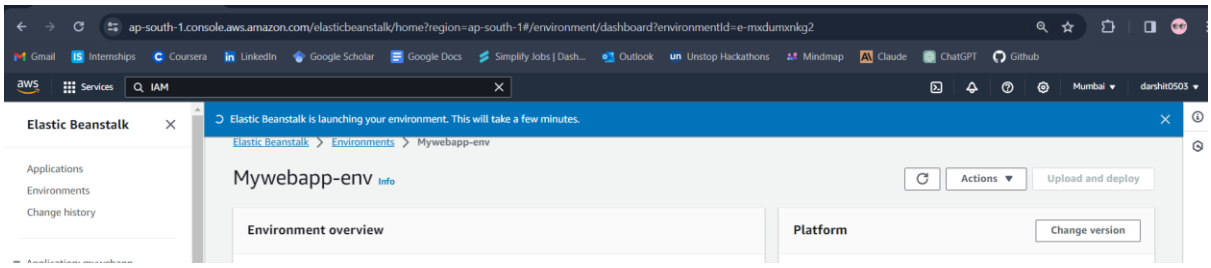
Step 13- Skip the next steps and go to the review Section



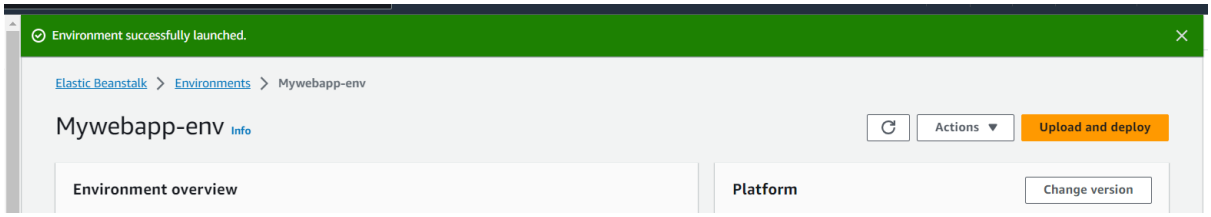
Step 14- Submit the Review



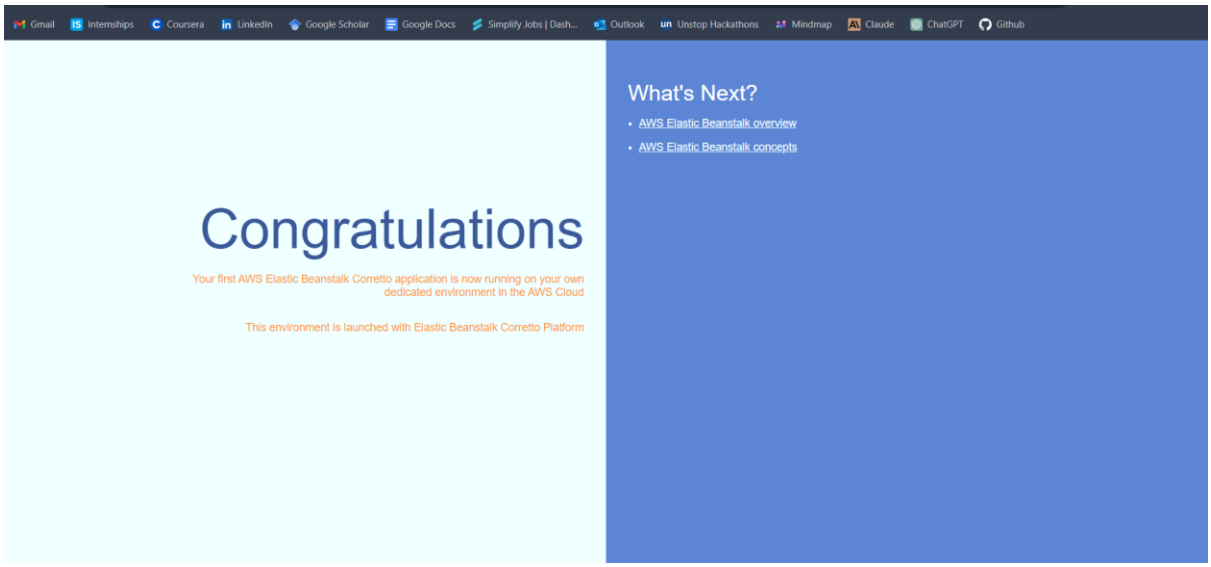
Step 15- After the review the Elastic BeanStalk will run environment and it will take some few minutes



Step 16- Environment runs successfully



Step 17- Output of the following



Step 18- Terminate the following Instance