SVKM'S NMIMS 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 (PaaS):**

Platform as a Service (PaaS) is a category of cloud computing services that provides a runtime environment for web applications. It allows developers to create, test, run, and manage applications without the complexity of building and maintaining the infrastructure typically associated with these tasks. PaaS includes infrastructure (servers, storage, and networking) and platform (middleware, development tools, database management systems, business intelligence, and more) to support the web application life cycle. Examples of PaaS providers include Google App Engine, Force.com, Joyent, and Azure.

PaaS offers several advantages such as simplified development, lower risk, prebuilt business functionality, instant community, and scalability. It allows developers to focus on development and innovation without worrying about infrastructure management. However, there are also disadvantages such as vendor lock-in and data privacy concerns. One has to write the applications according to the platform provided by the PaaS vendor, so the migration of an application to another PaaS vendor could be a problem. Corporate data, whether it can be critical or not, will be private, so if it is not located within the walls of the company, there can be a risk in terms of privacy of data.

- **Elastic Beanstalk:**

AWS Elastic Beanstalk is an orchestration service offered by Amazon Web Services for deploying applications. It allows developers to quickly deploy and manage applications in the AWS Cloud without having to learn about the infrastructure that runs those applications. Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring. It supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby. When you deploy your application, Elastic Beanstalk builds the selected supported platform version and provides one or more AWS resources, such as Amazon EC2 instances, to run your application.

Why Elastic Beanstalk -

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

- **Components of Beanstalk:**

AWS Elastic Beanstalk consists of several key components:

1. **Application**: A logical collection of Elastic Beanstalk components, including environments, versions, and environment configurations.
2. **Application Version**: Refers to a specific, labeled iteration of deployable code for a web application.
3. **Environment**: A collection of AWS resources running an application version.
4. **Environment Tier**: Designates the type of application that the environment runs, and determines what resources Elastic Beanstalk provisions to support it.
5. **Environment Configuration**: Identifies a collection of parameters and settings that define how an environment and its associated resources behave.
6. **Saved Configuration**: A template that you can use as a starting point for creating unique environment configurations.
7. **Platform**: A combination of an operating system, programming language runtime, web server, application server, and Elastic Beanstalk components.

- **IAM:**

Identity and Access Management (IAM) is a combination of policies, processes, and technologies that enable organizations to manage digital identities and control user access to critical corporate information. It is a vital part of modern IT as it manages access so that the right people can do their jobs and the wrong

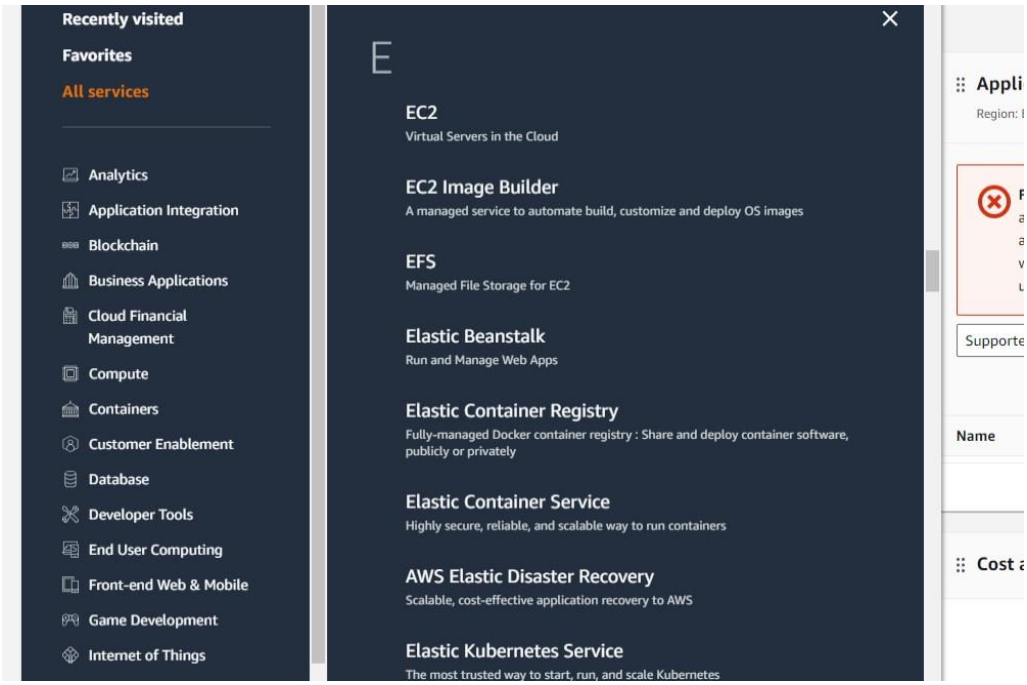
people, like hackers, are denied entry. IAM systems keep hackers out while ensuring that each individual user has the exact permissions they need to do their job. The components of IAM include Users, Roles, Groups, and Policies. IAM can be used for many purposes such as controlling access of individual and group access for your AWS resources. With IAM policies, managing permissions to your workforce and systems to ensure least-privilege permissions becomes easier. The AWS IAM is a global service.

- Implement PAAS using elastic beanstalk for the following.

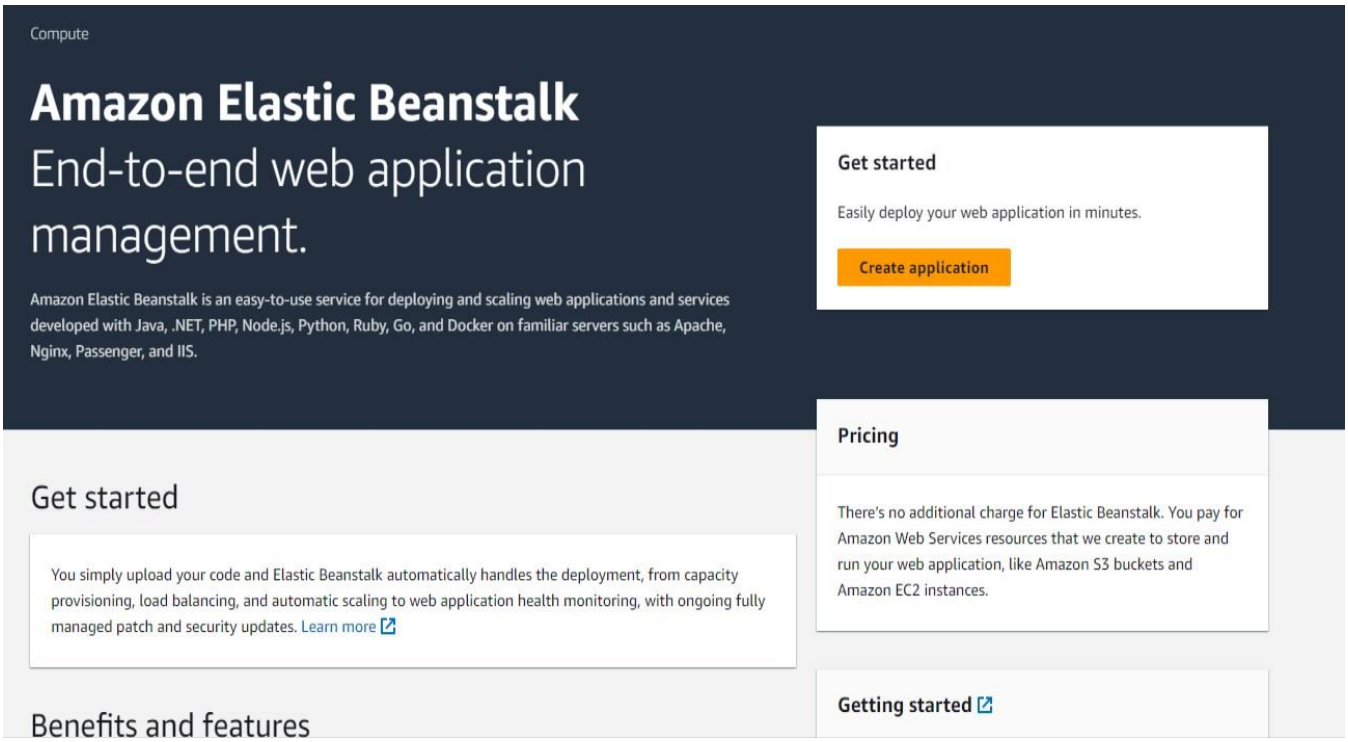
  1. Server 2. Java 3. Python 4. Node.js

FOR PYTHON

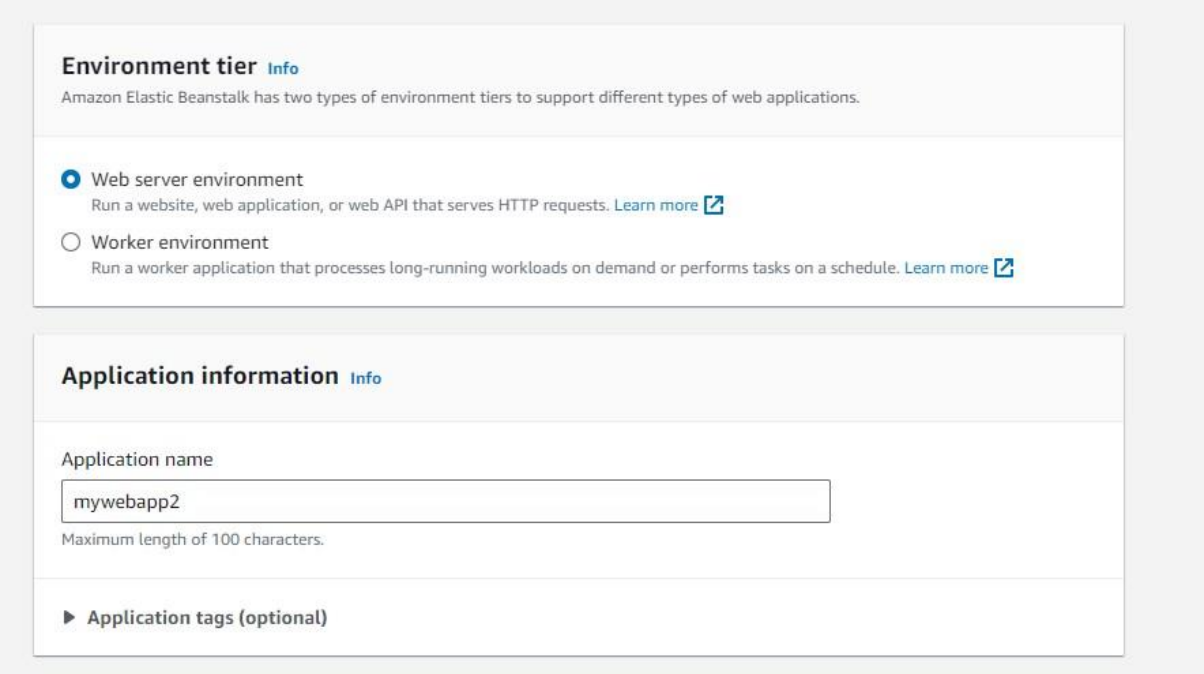
Step 1:- Select Elastic Beanstalk



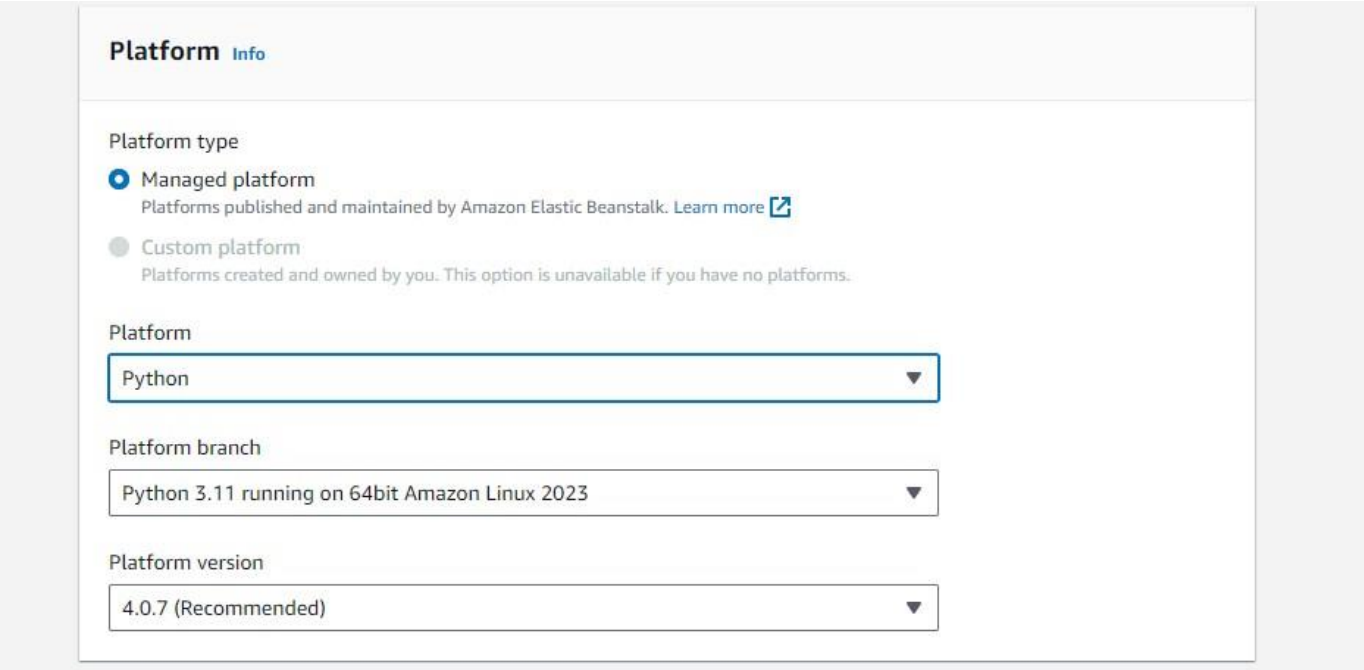
Step 2- Create a new application



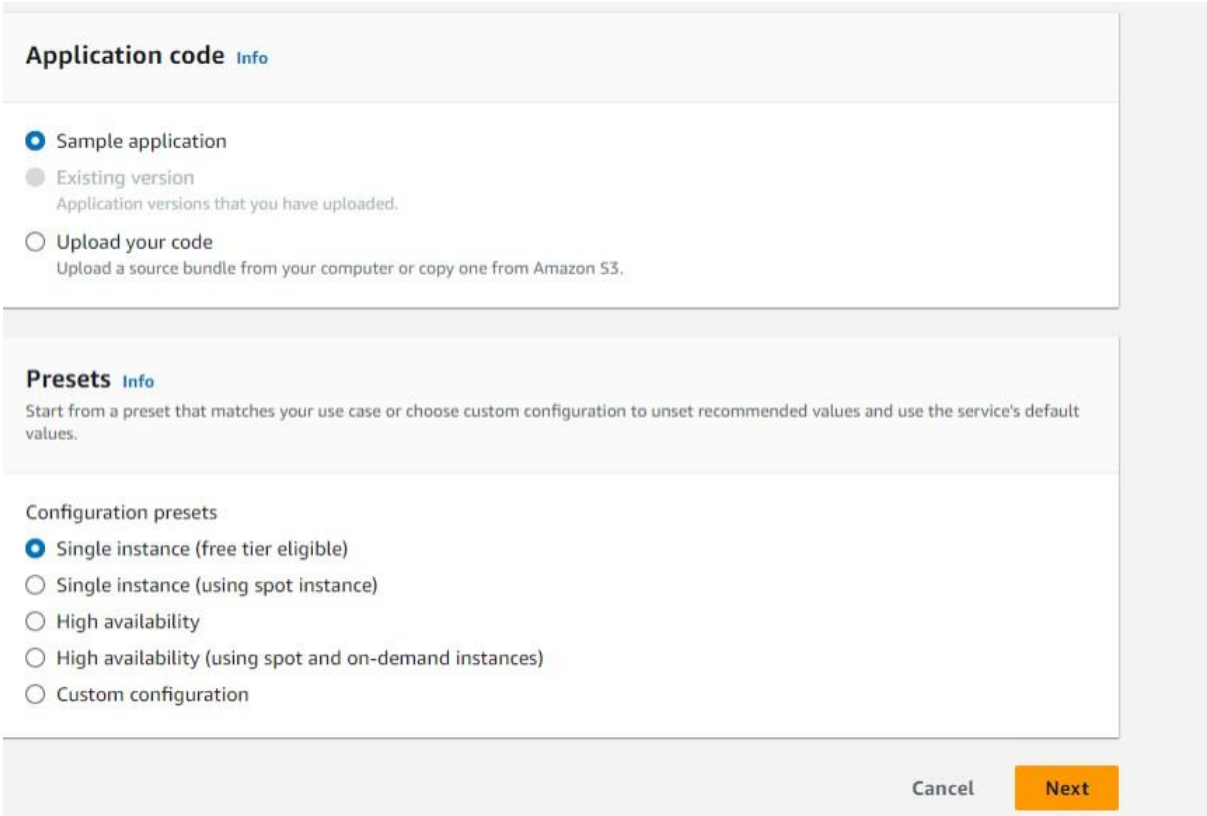
Step 3- Provide a new Name for the Application



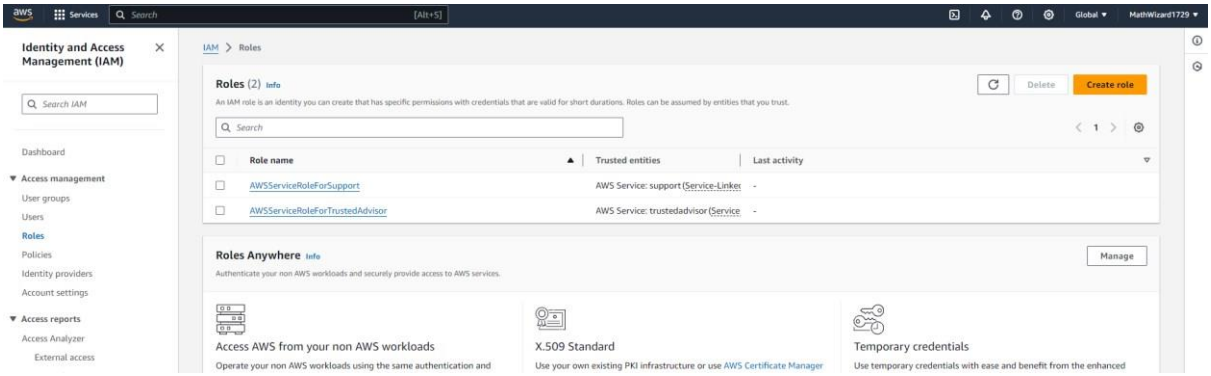
Step 4- Choose the Platform as Python



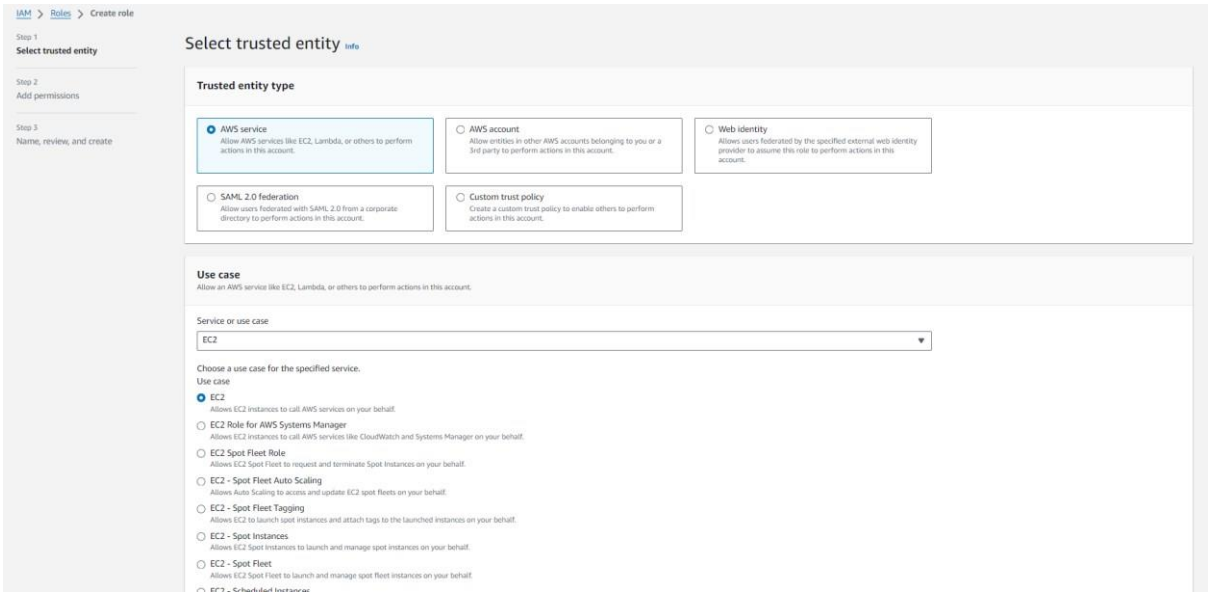
Step 5- Keep it as Single Instance



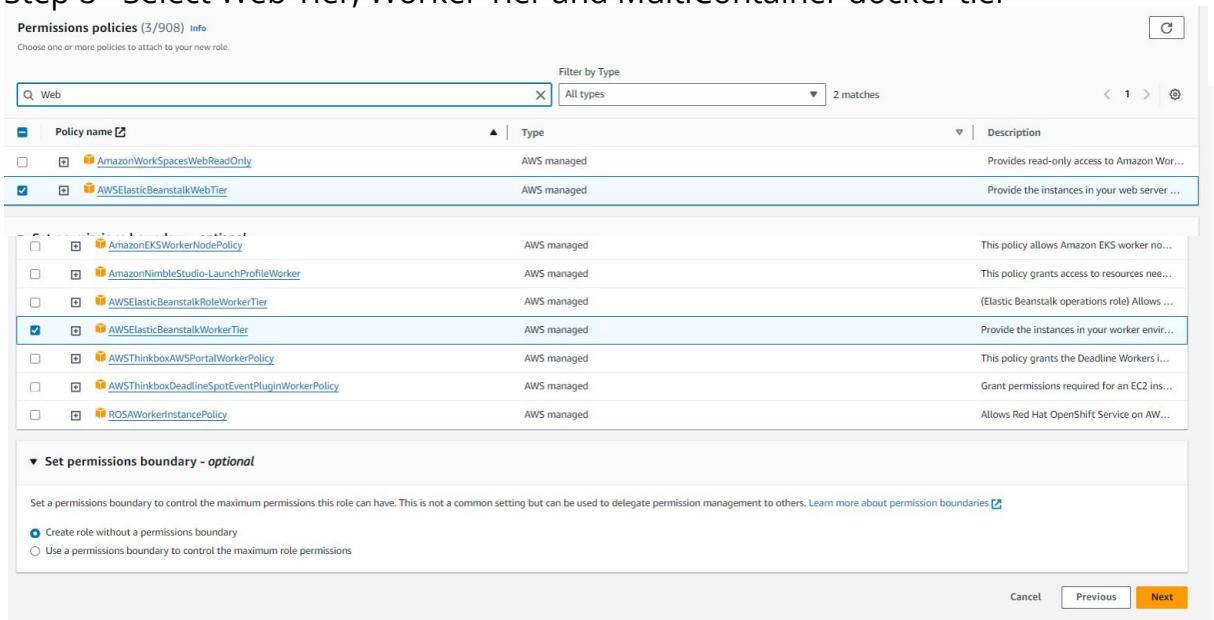
Step 6- Create a new Role under IAM(Identity Access Management)

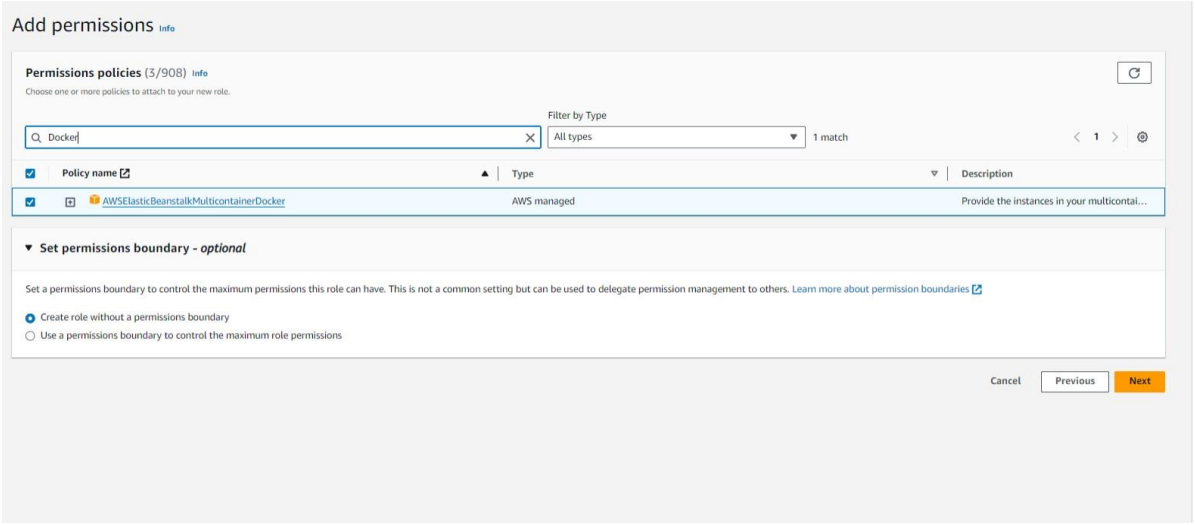


Step 7- Create a new role and select the Usecase as EC2

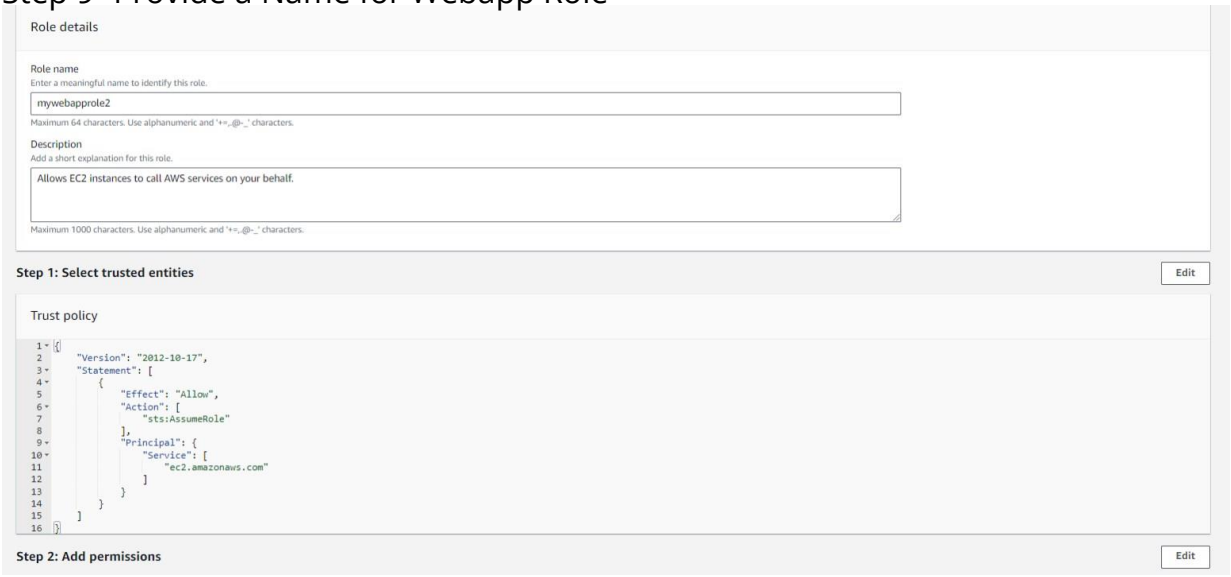


Step 8- Select Web Tier, Worker Tier and MultiContainer docker tier

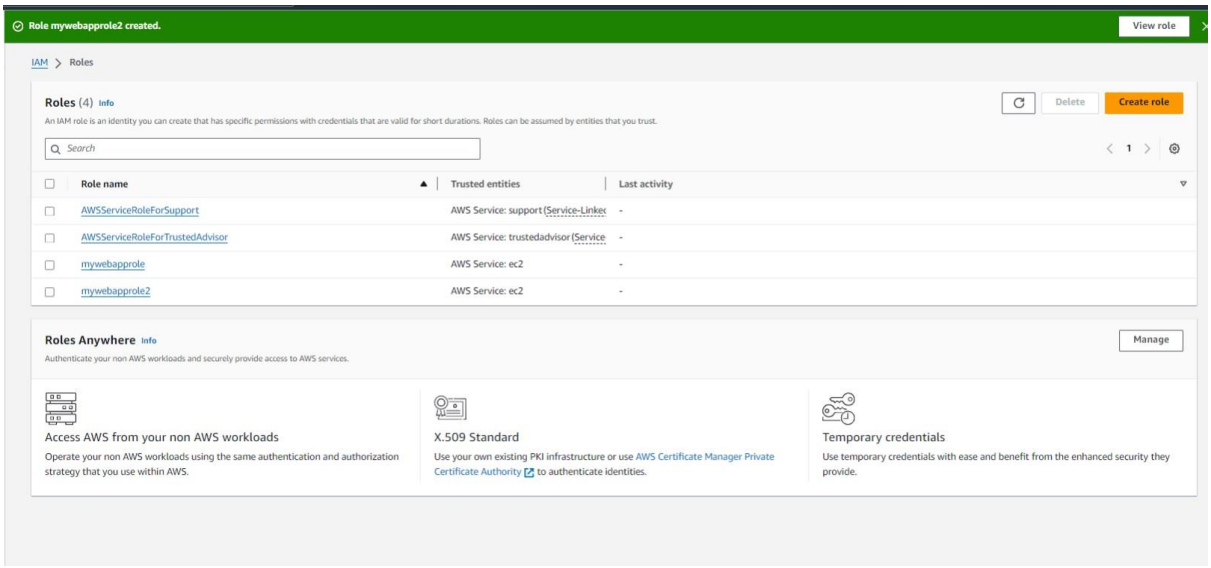




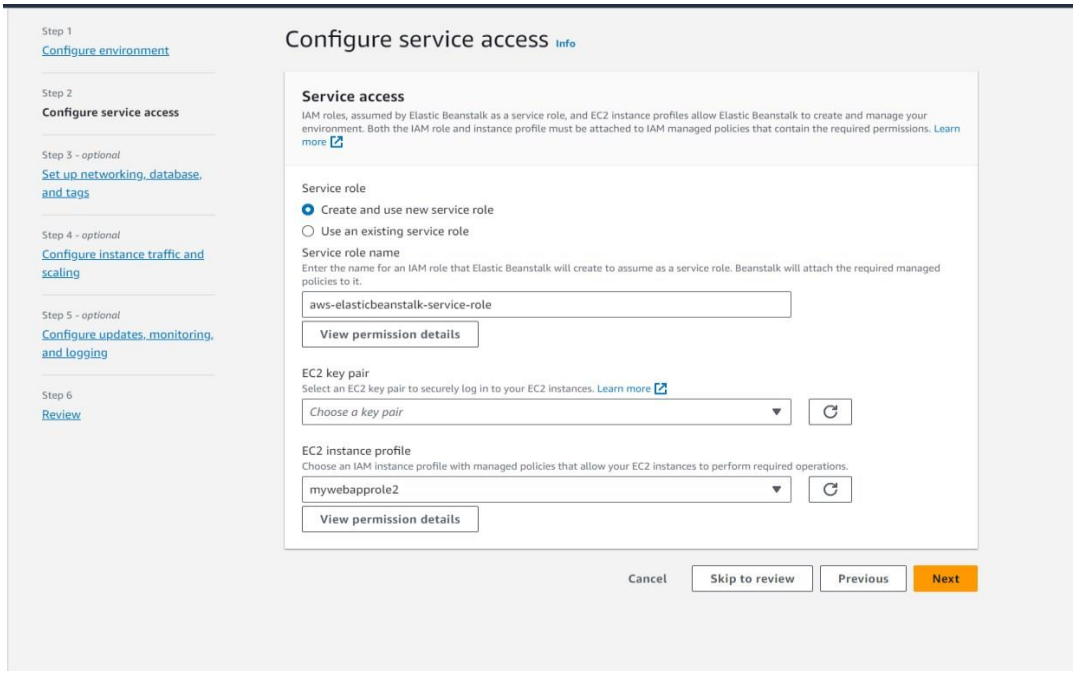
Step 9- Provide a Name for Webapp Role



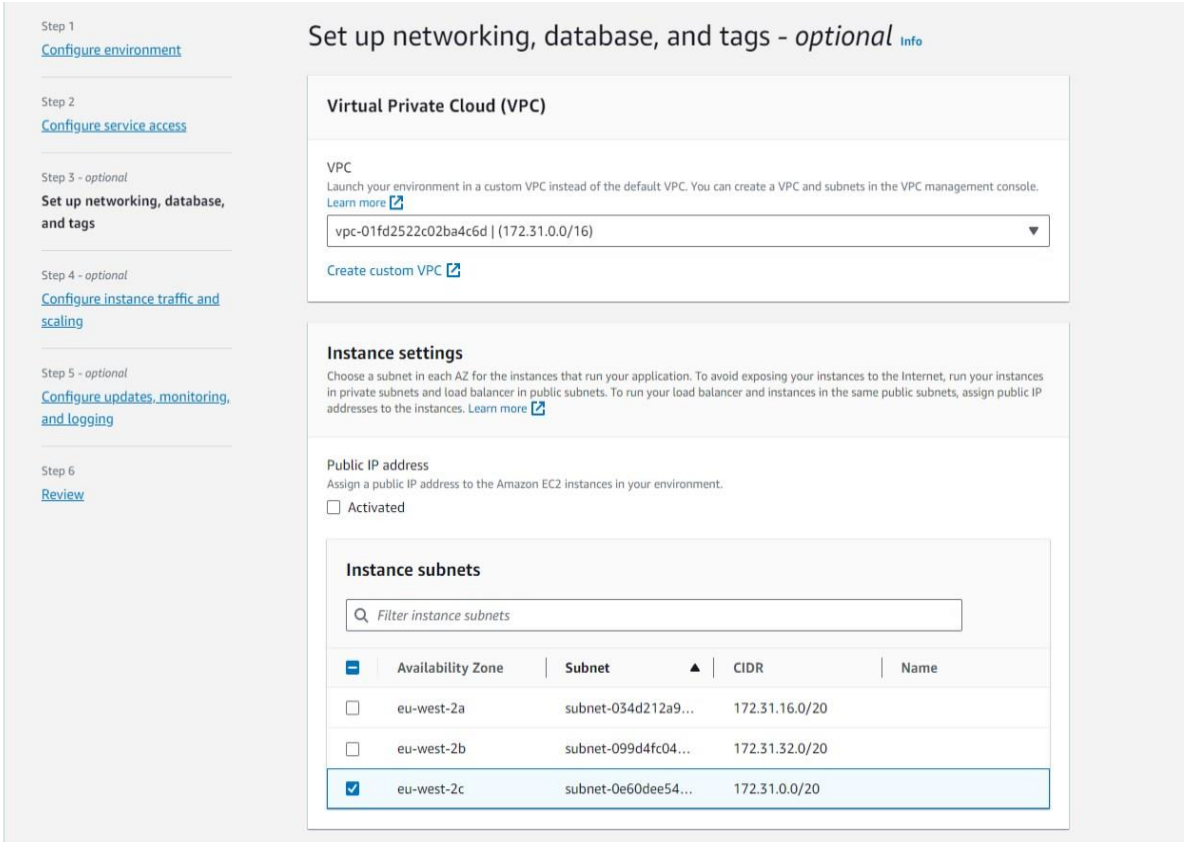
Step 10- role was Created Successfully



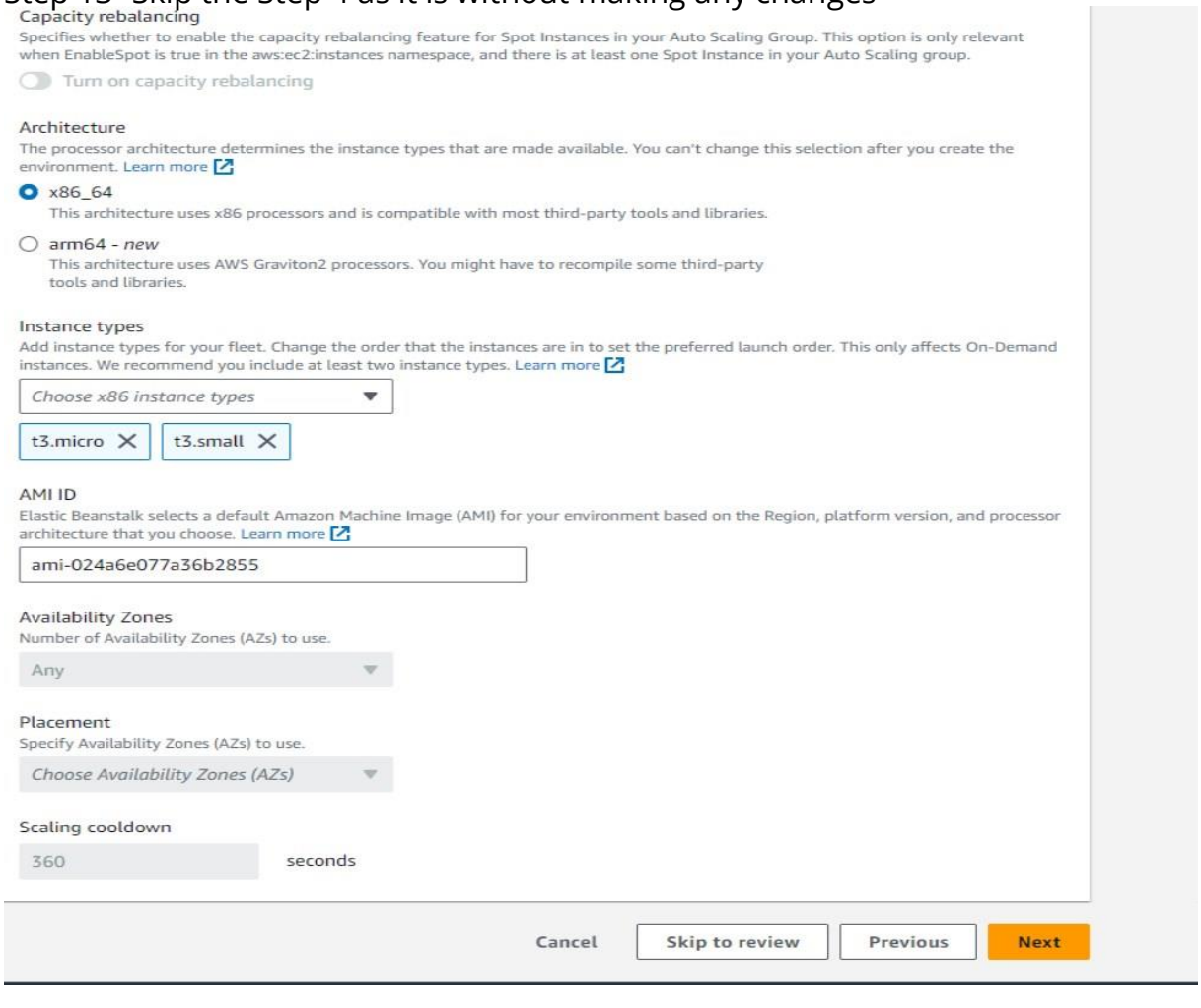
Step 11- For this Access Create a new use case and service role and for Instance profile select as mywebapprole2



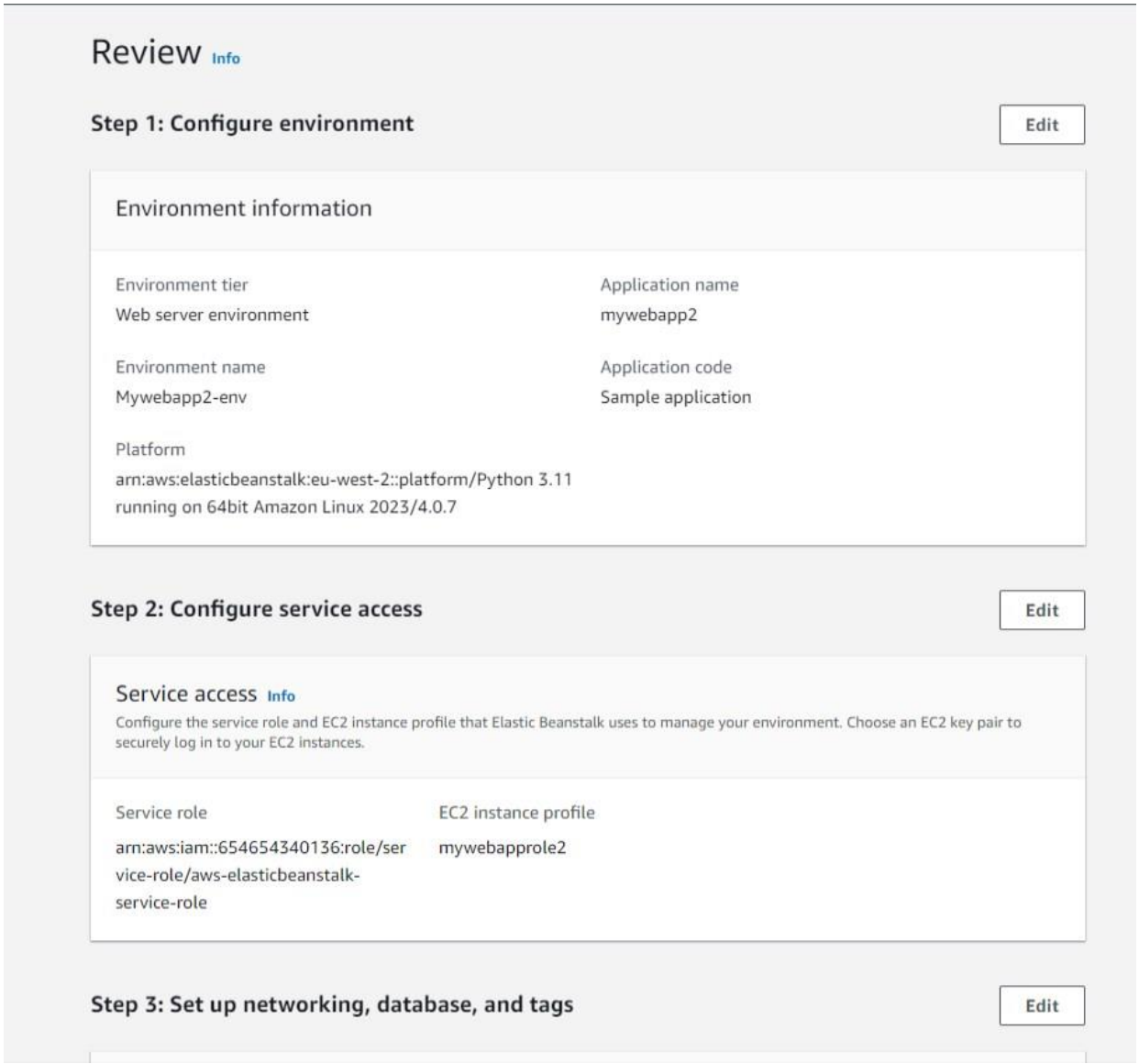
Step 12- Select the VP name as given and from the IP address provided select the same down below for Instance Settings and Database



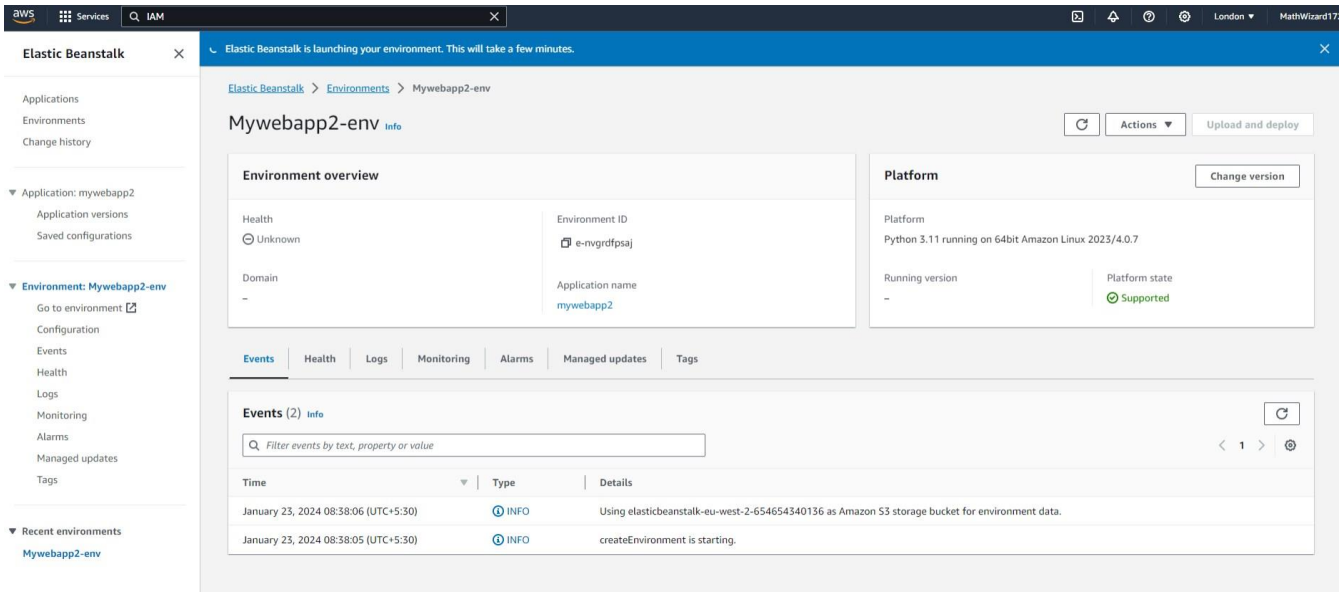
Step 13- Skip the Step 4 as it is without making any changes



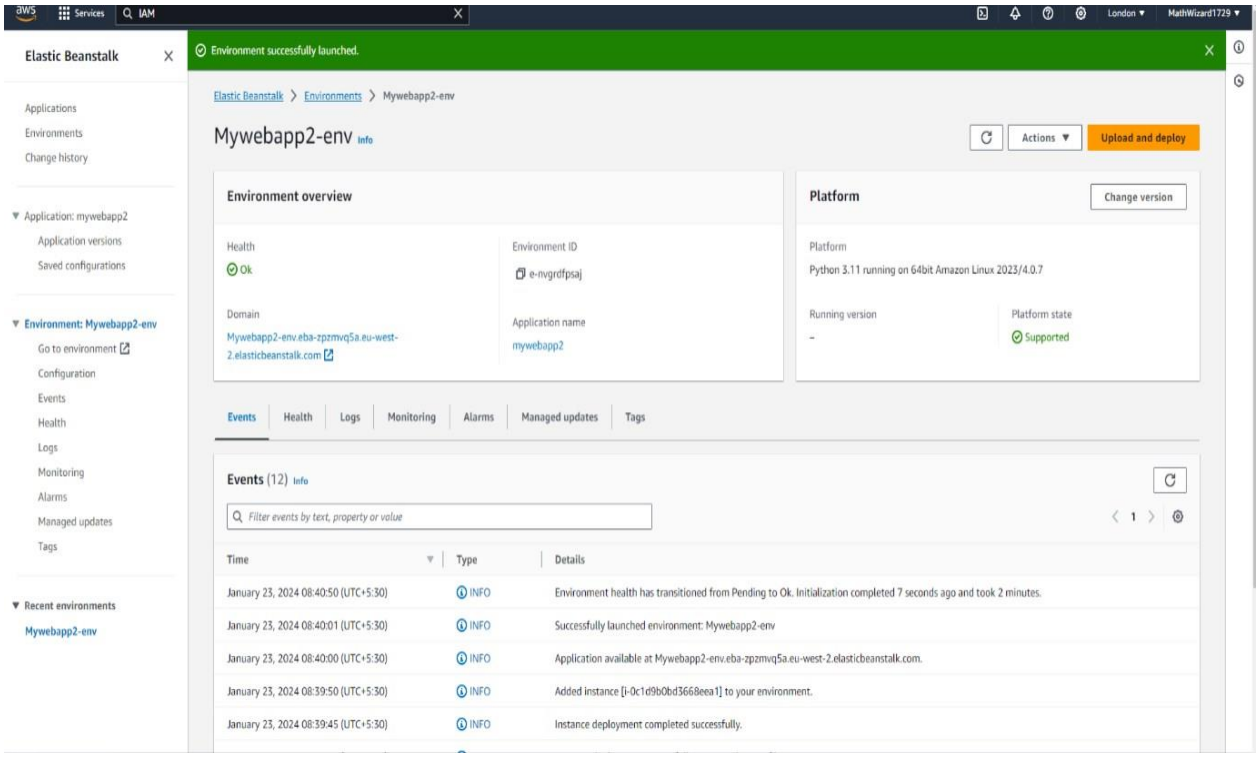
Step 14- Skip the Step 5 Section and directly go to Review Page where you can view the complete Changes made.and Submit it



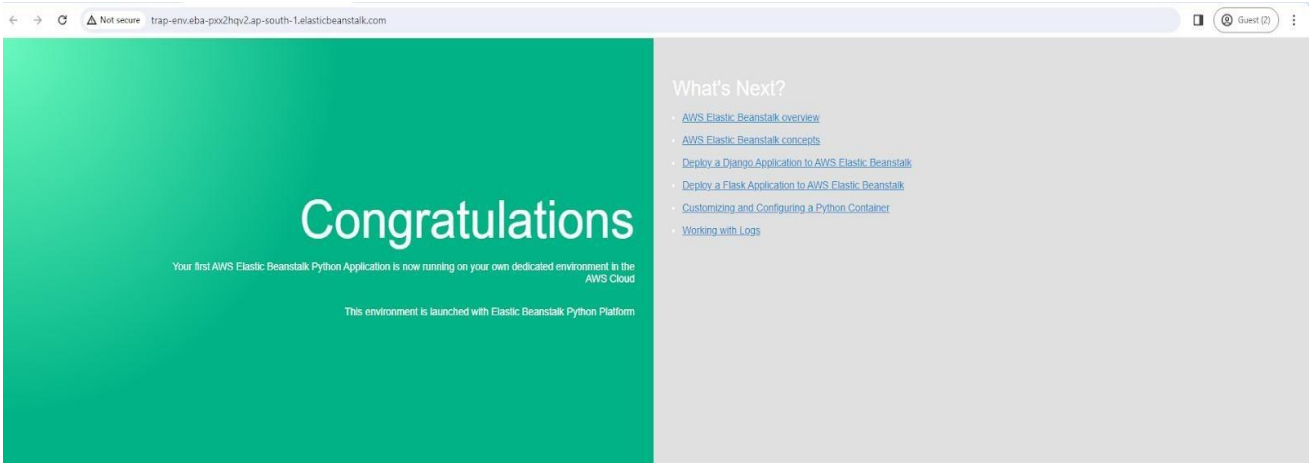
Step 15- After the review the Elastic Beanstalk will run environment and it will take some few minutes



Step 16- Environment is successfully launched



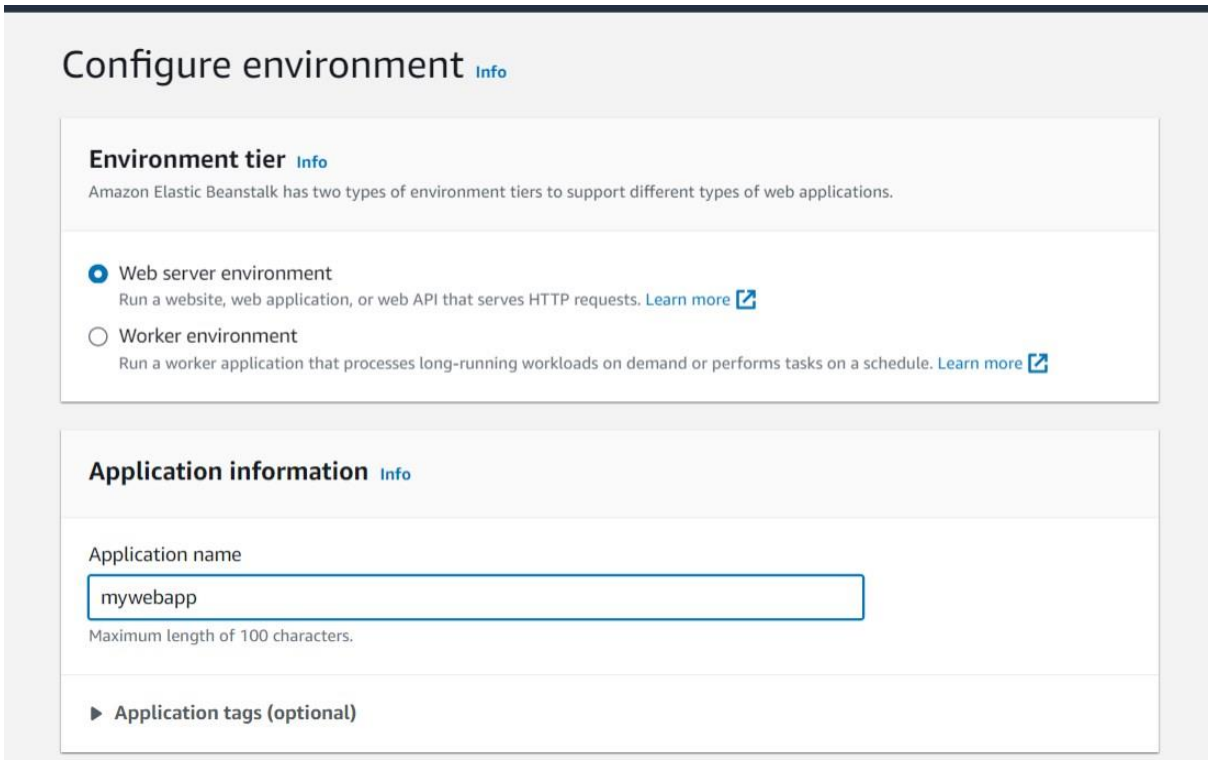
Step 17- The Output is shown as like this



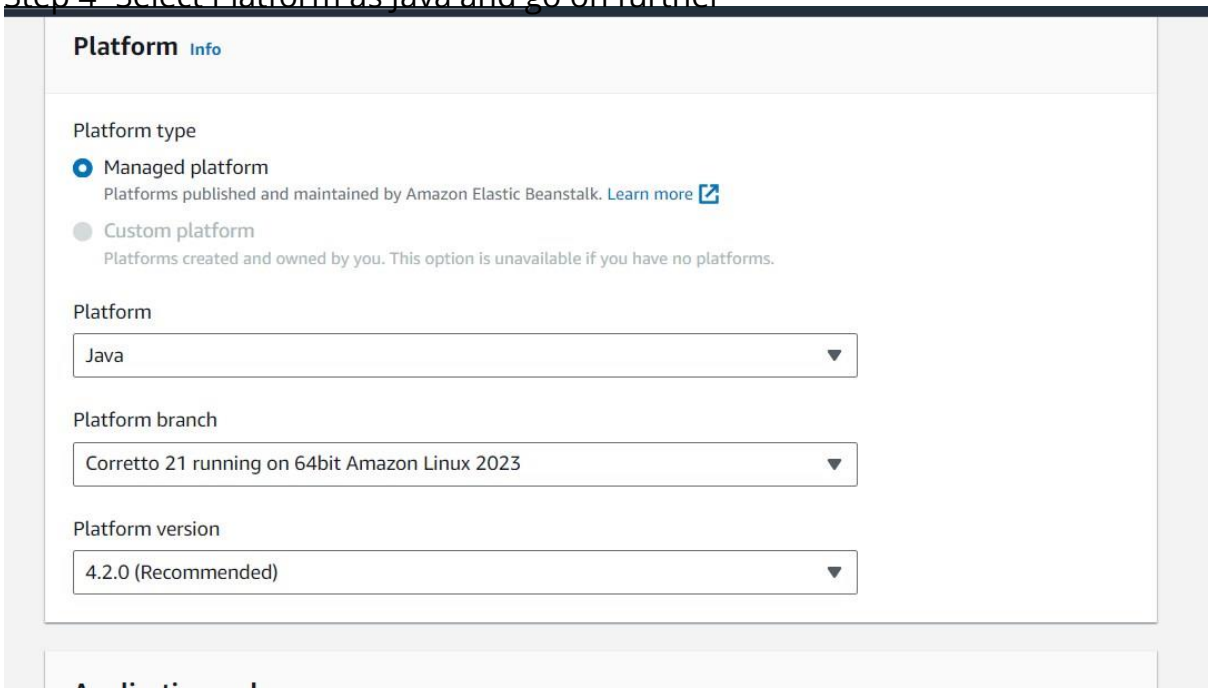
FOR JAVA

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.

Application code Info

- (•) Sample application
- ( ) Existing version
  Application versions that you have uploaded.
- ( ) Upload your code
  Upload a source bundle from your computer or copy one from Amazon S3.

Presets Info

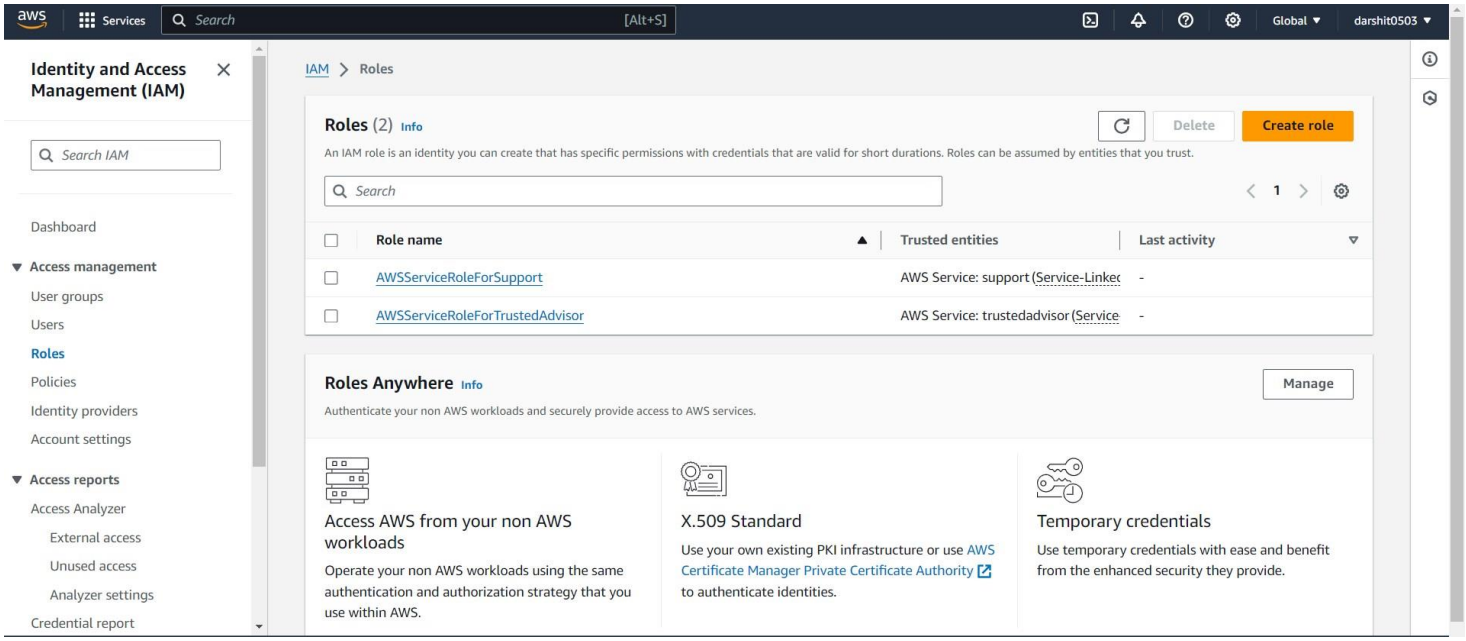
Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

Configuration presets

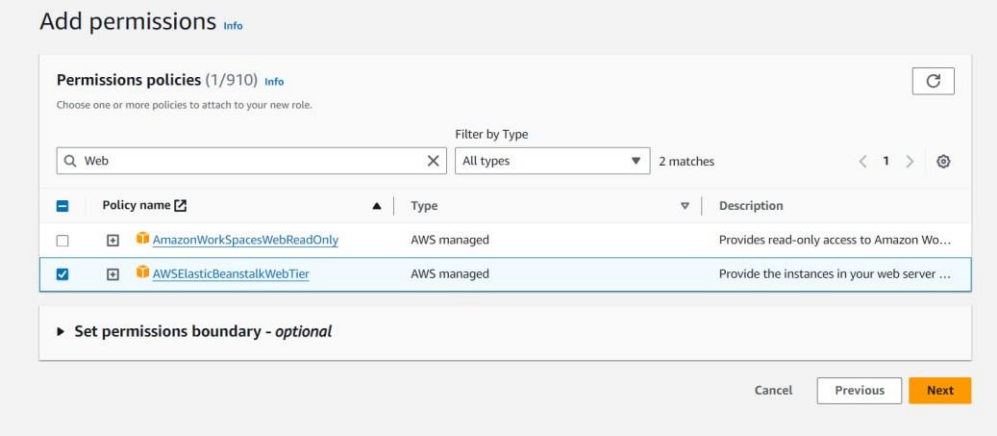
- (•) Single instance (free tier eligible)
- ( ) Single instance (using spot instance)
- ( ) High availability
- ( ) High availability (using spot and on-demand instances)
- ( ) Custom configuration

Cancel **Next**

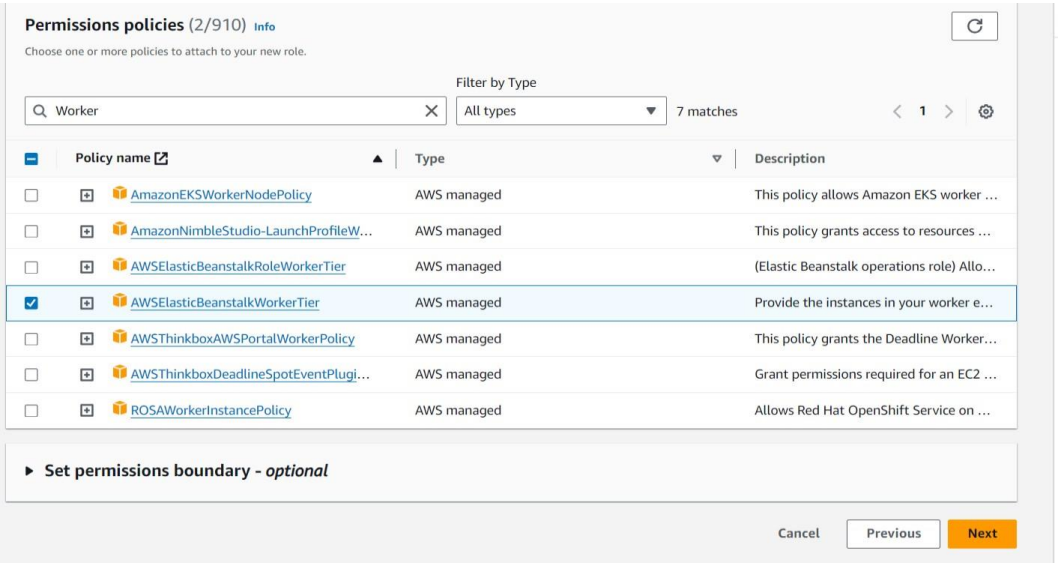
Step 6- Create Role under IAM User



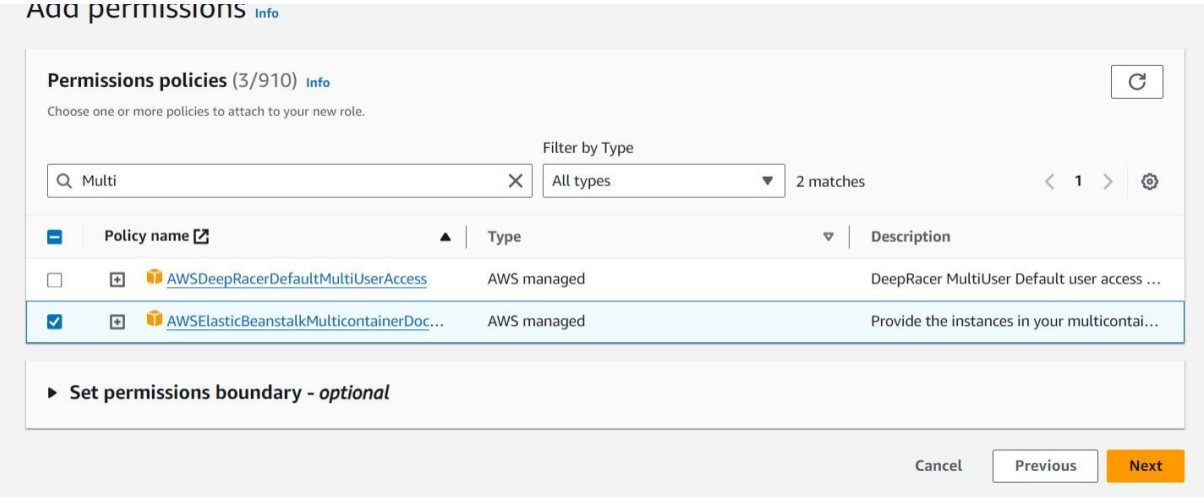
Step 7- Select Usecase as EC2 and select Next and from the following list
Select3 options a) WebTier



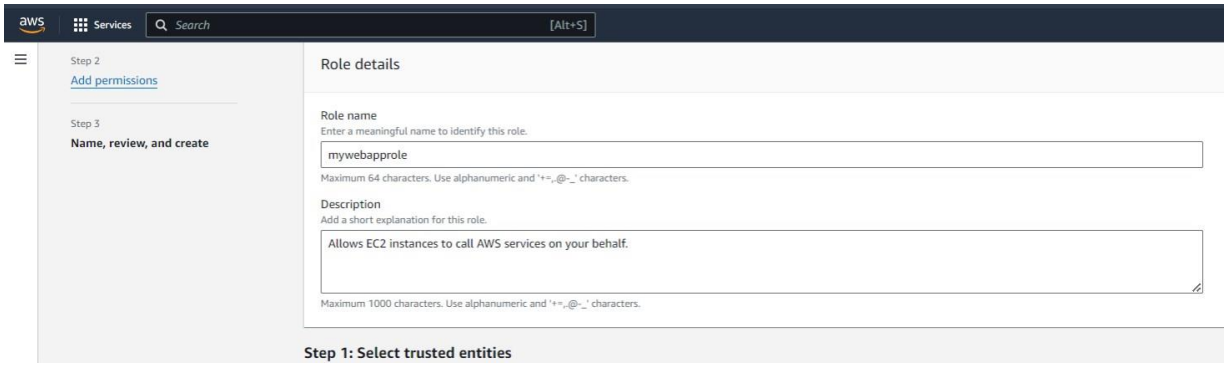
b) WorkerTier



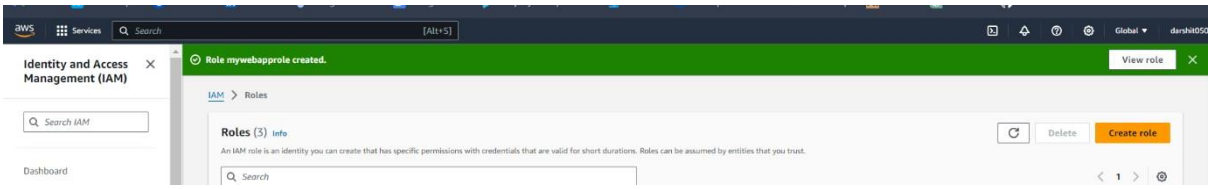
c) MultiContainerDocker list



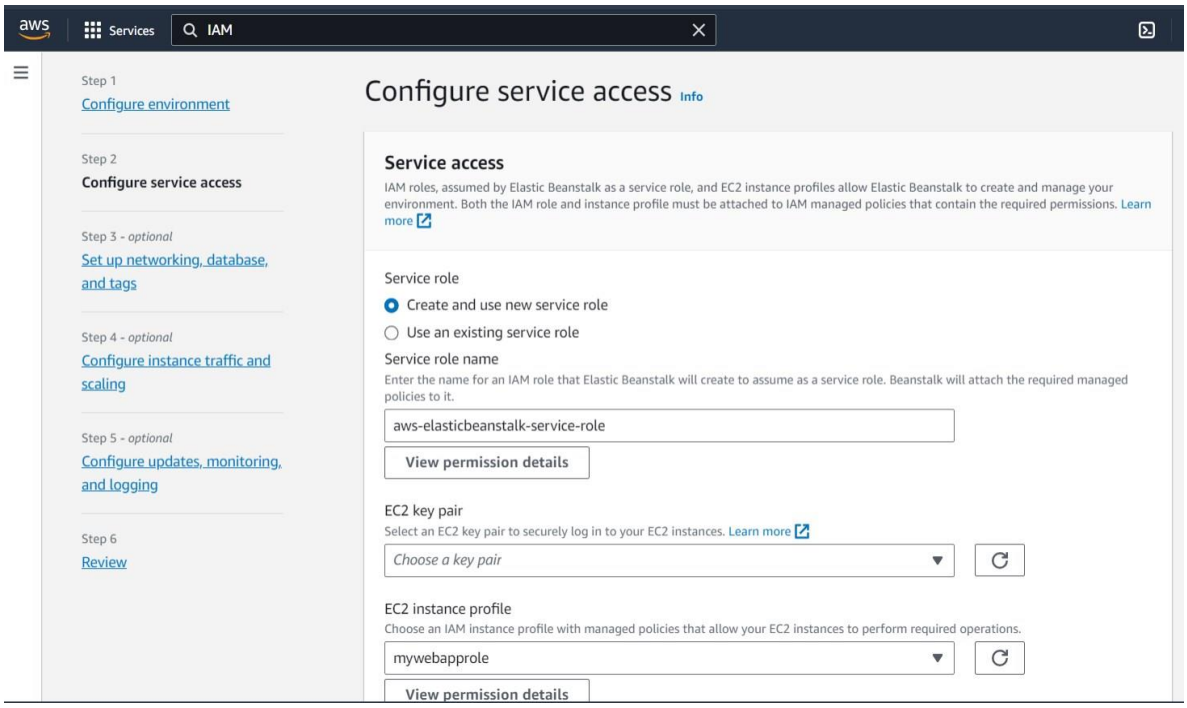
Step 8 - Provide the name for the Role



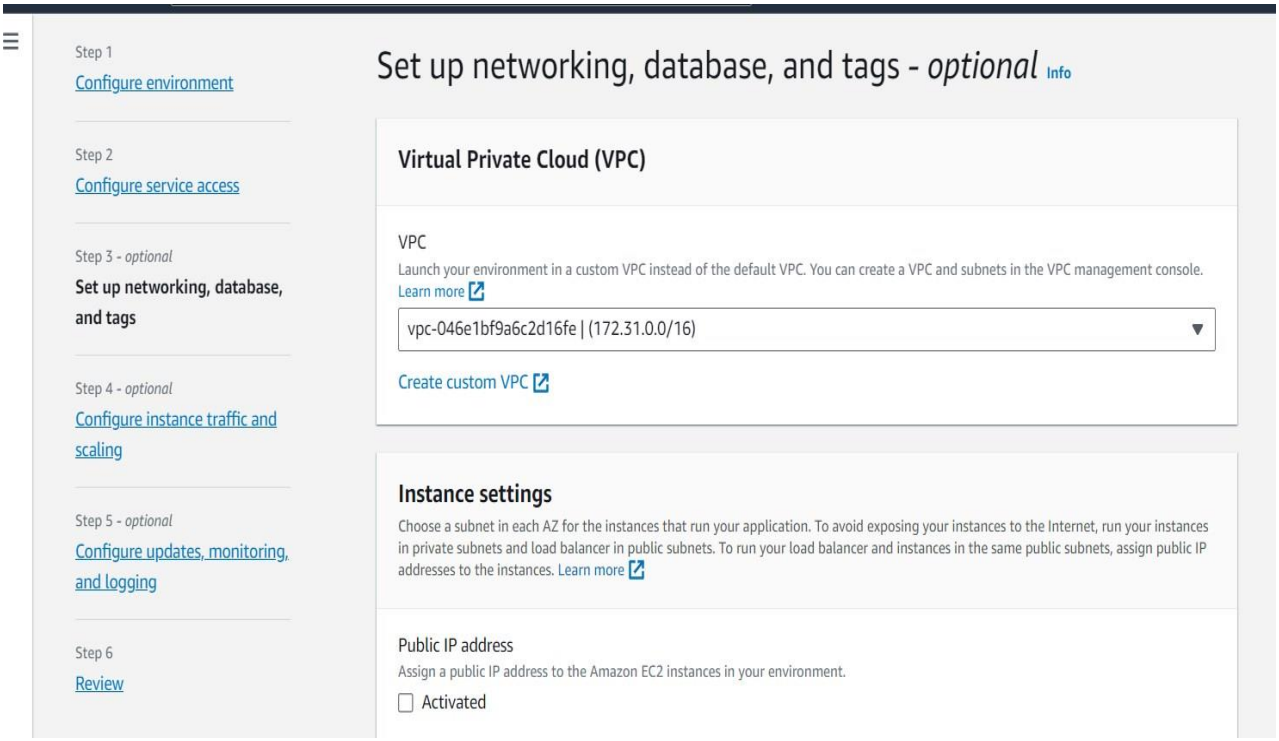
Step 9- The Role was created Successfully



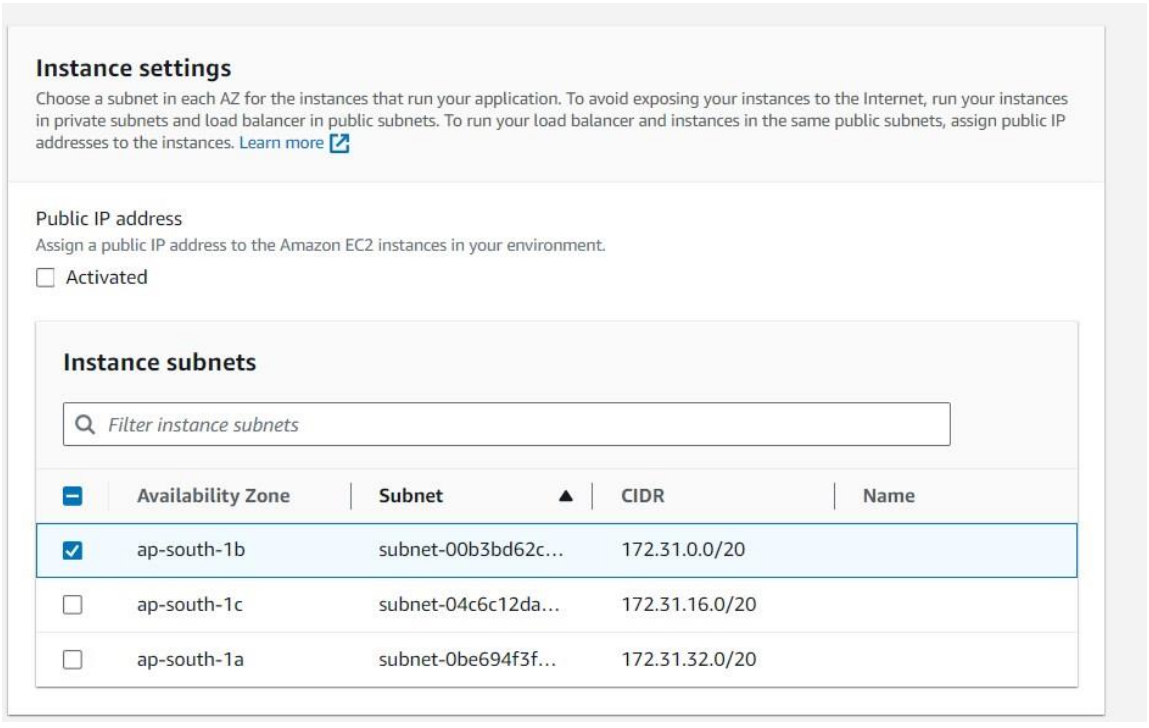
Step 10- For the Configure Service Access select new service Role and under domain EC2 Instance Role select mywebapprole which was created above



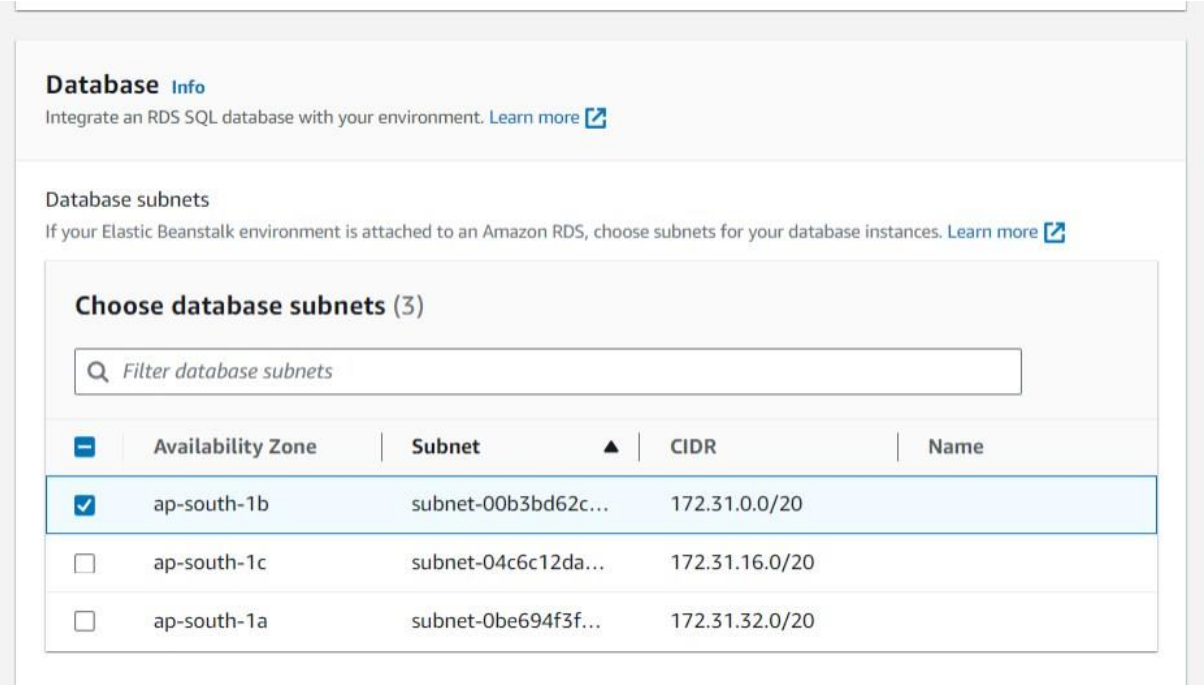
Step 11- Select VPC which was been provided there



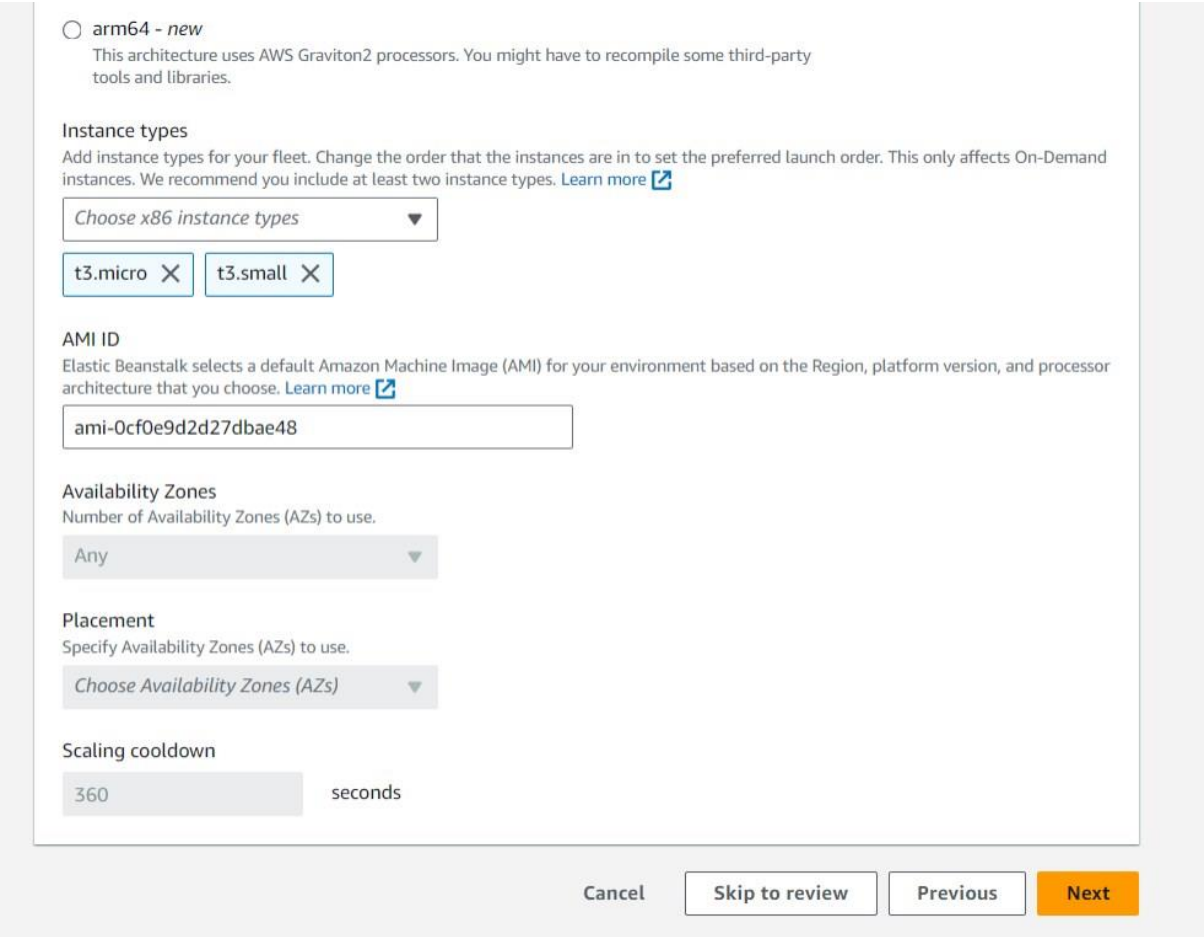
Step 12- Select the IP Address Same as above which was been provided in VPC in Instance and Database



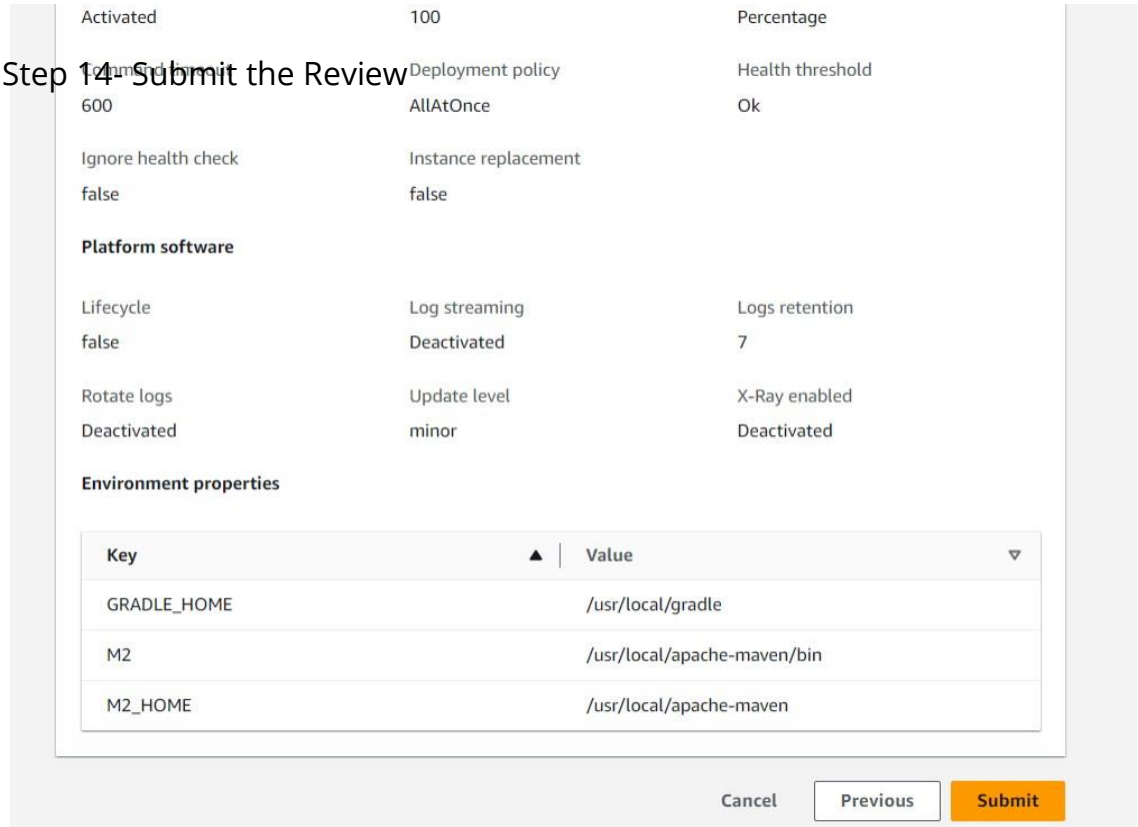
FOR DATABASE



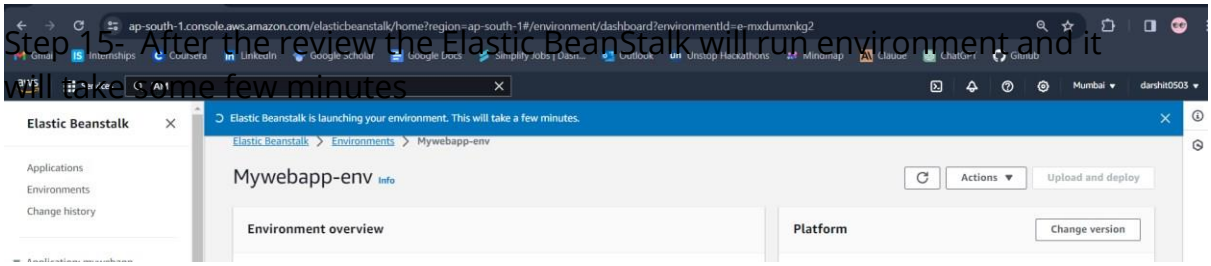
Step 13- Skip the next steps and go to the review Section



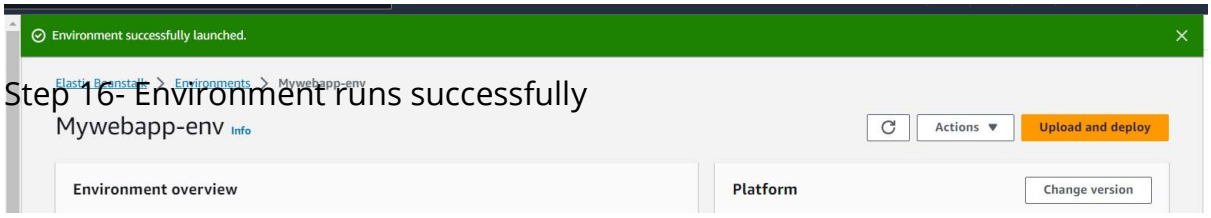
Step 14- Submit the Review



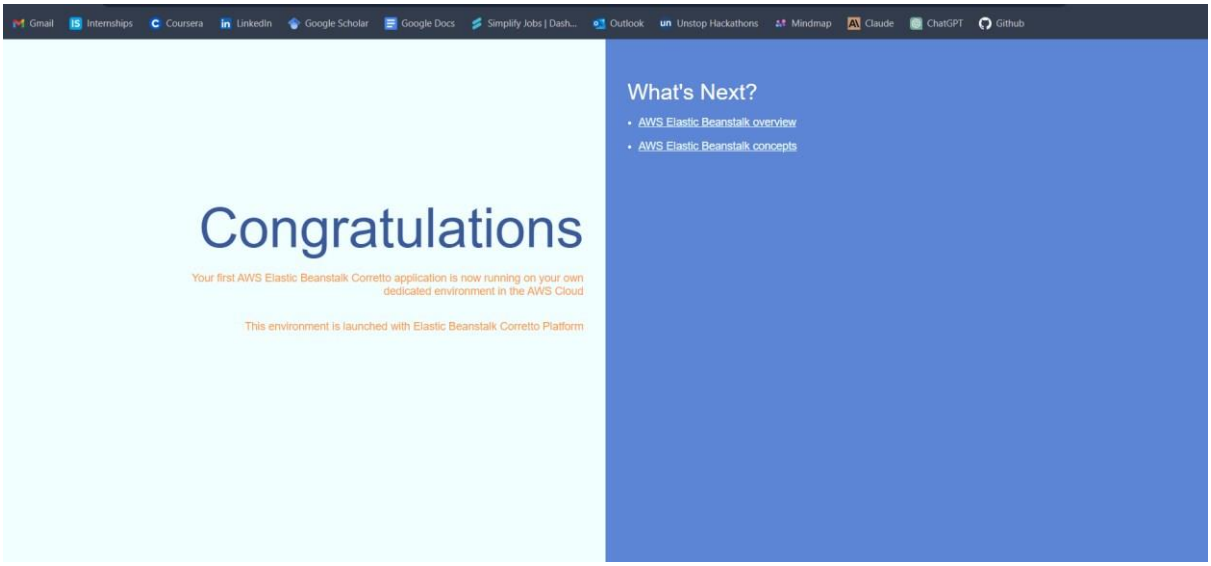
Step 15- After the review the Elastic BeanStalk will run environment and it will take some few minutes



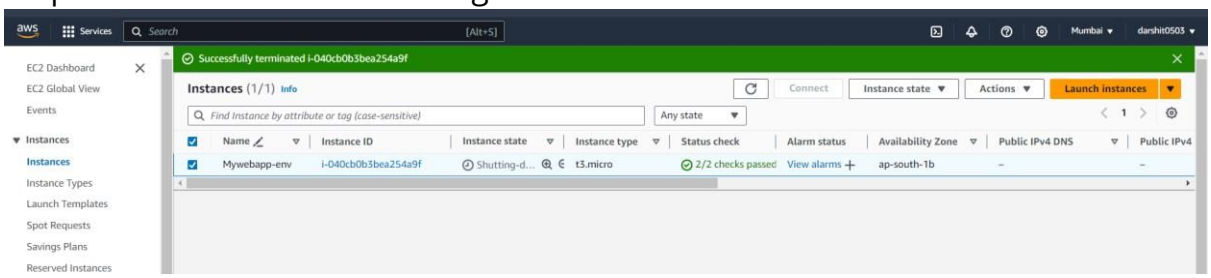
Step 16- Environment runs successfully



Step 17- Output of the following

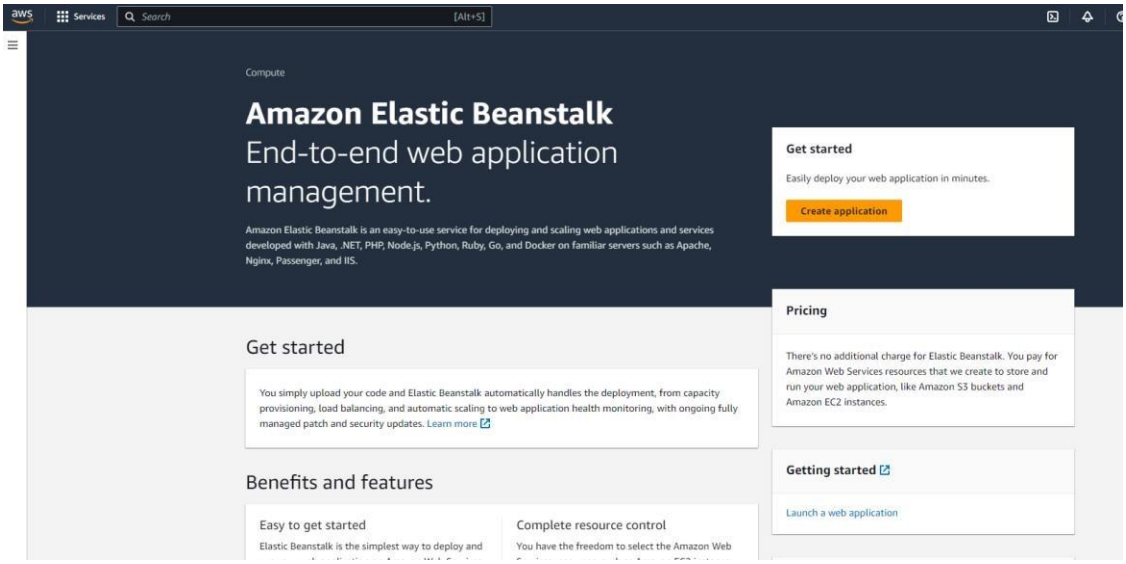


Step 18- Terminate the following Instance

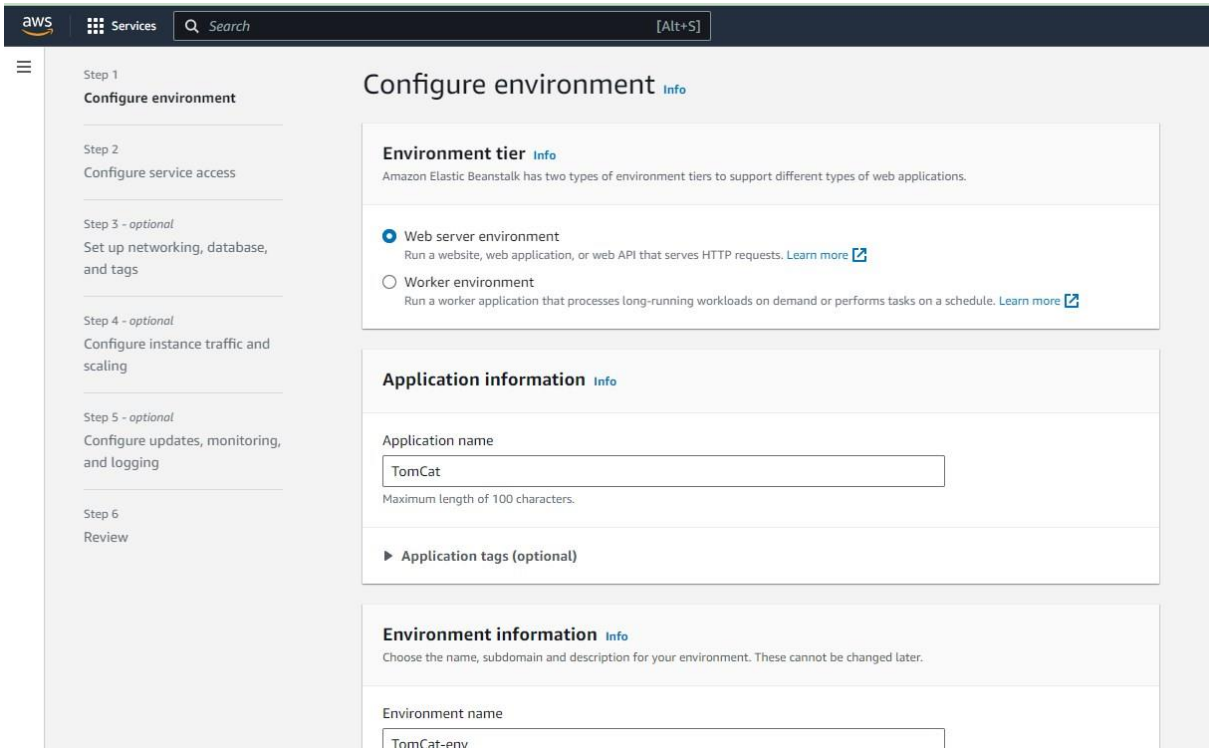


FOR TOMCAT

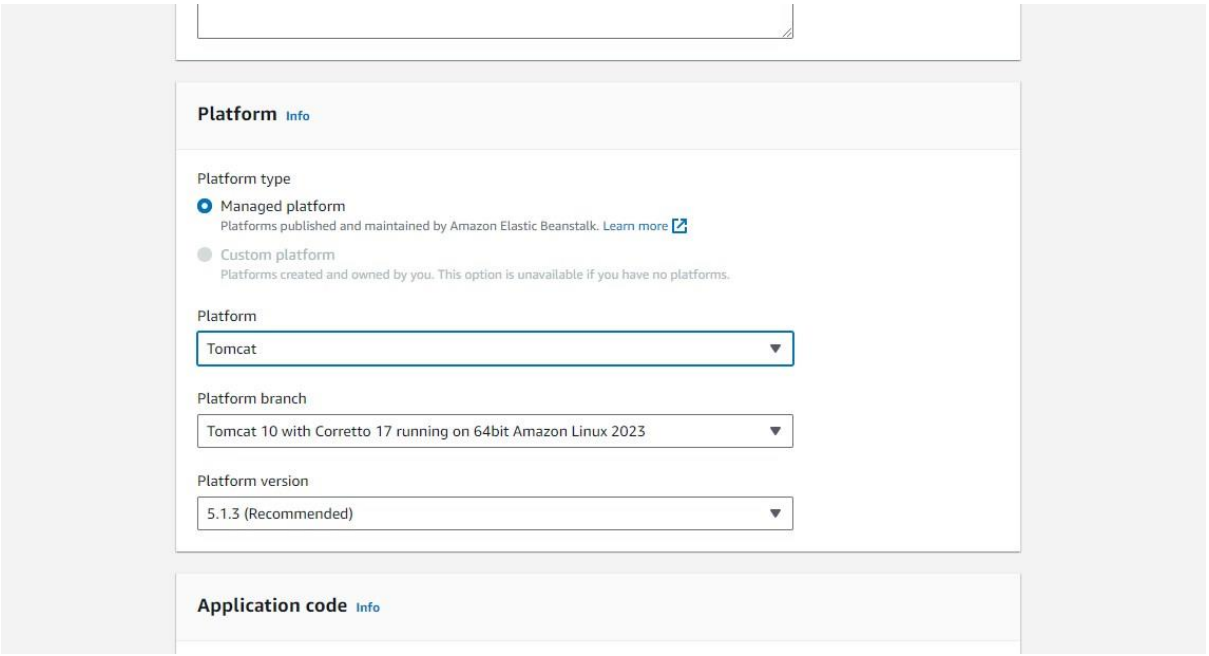
1. Go to your Elastic Beanstalk



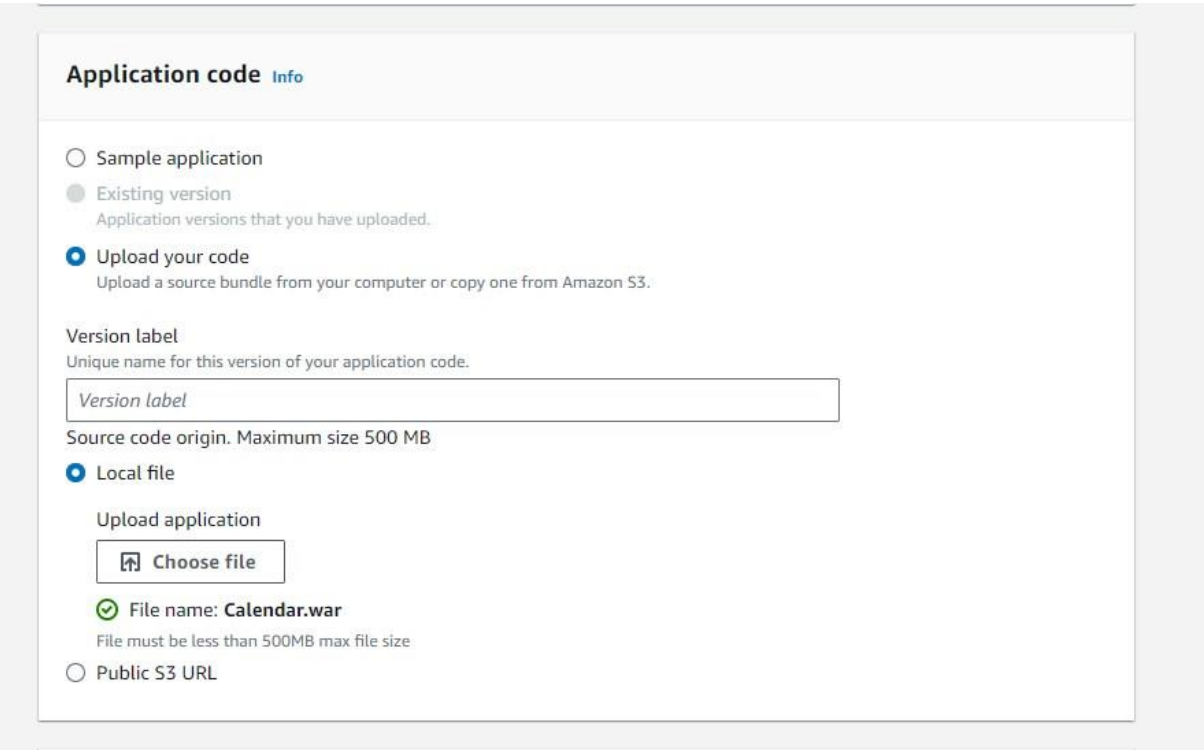
2. Provide the name of the Application



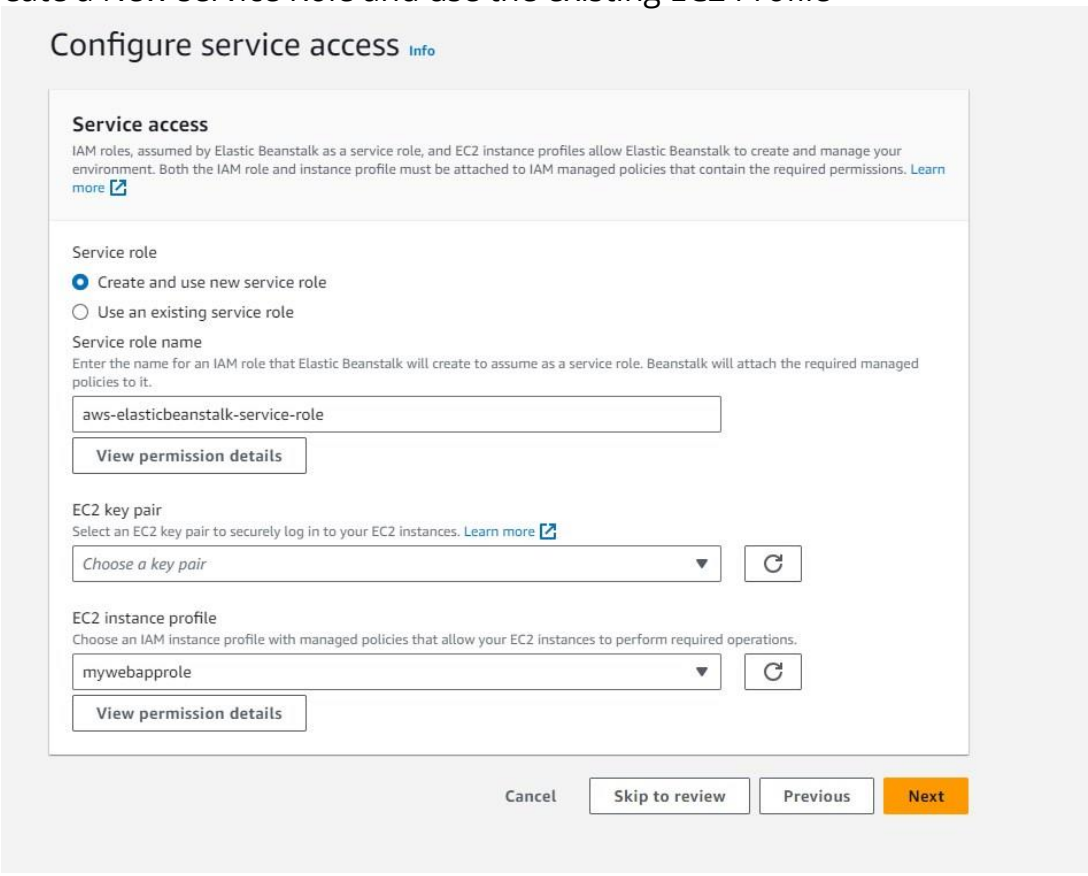
3. Choose the Platform as TomCat



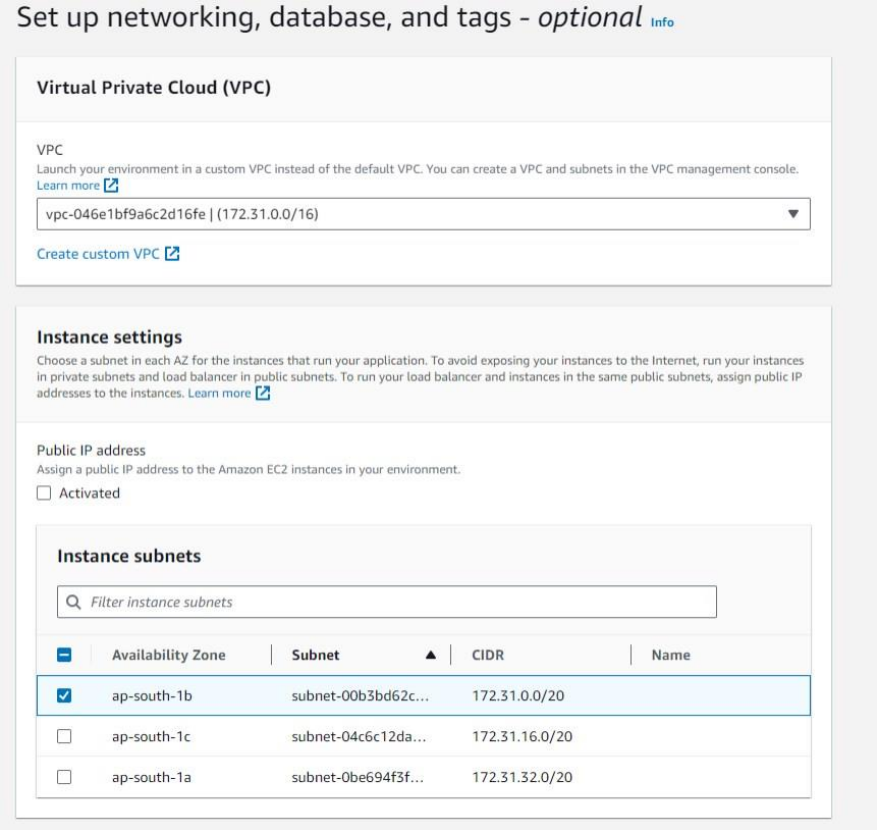
4. Download “calendar.war” File and Upload the Code



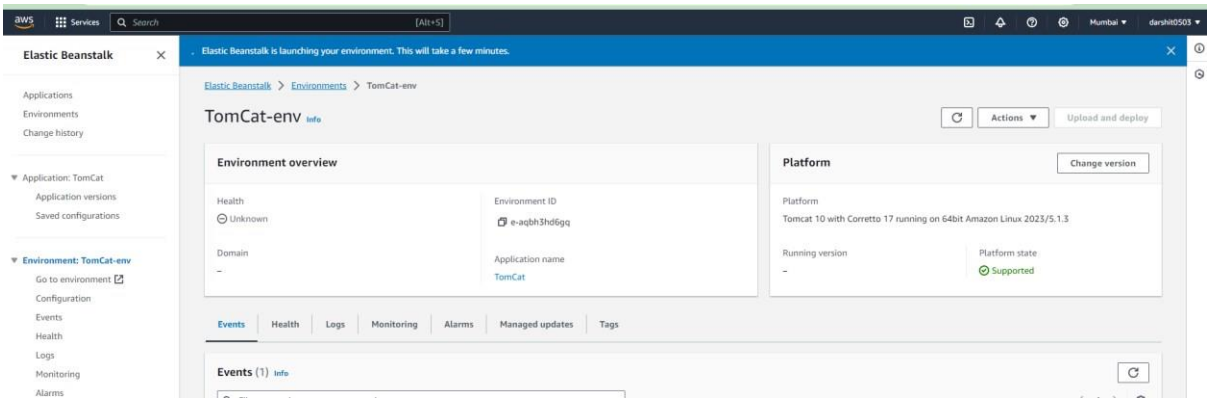
5. Create a New service Role and use the existing EC2 Profile



6. Use the VPC and select the Same IP Address in Instance and Database



7. Skip to the Review Section and Submit it and you will see the Environment will start running



8. The Output is seen as follows