Q1) Writeup:-

Practical-2 Platform as a service using AWS.

 Platform as a service:

Platform as a Service (PaaS) is a cloud computing model that provides a platform allowing customers to develop, run, and manage applications without dealing with the complexities of building and maintaining the infrastructure typically associated with software development. In a PaaS model, the underlying infrastructure, including servers, storage, and

networking, is managed by the service provider, allowing developers to focus solely on their applications and business logic.

PaaS offerings typically include development tools, middleware, database management systems, and other resources needed to support the complete lifecycle of application development and deployment. This enables developers to rapidly develop, test, deploy, and scale applications, reducing time-to-market and operational overhead.

 Elastic Beanstalk

Elastic Beanstalk is a Platform as a Service (PaaS) offering from Amazon Web Services (AWS) that simplifies the process of deploying and managing applications in the cloud. It supports multiple programming languages and frameworks, including Java, .NET, Node.js, Python, Ruby, Go, and Docker, allowing developers to choose the tools and technologies that best suit their needs.

One of the most popular PaaS offerings is Elastic Beanstalk, provided by the cloud giant Amazon Web Services (AWS). It simplifies application deployment and management, allowing you to focus on your code rather than the underlying infrastructure.

 Components of beanstalk

**Application**: An application in Elastic Beanstalk represents the logical container for the various components of your application. It can contain one or more environments.

**Environment**: An environment is an instance of your application running in Elastic Beanstalk. Each environment consists of resources such as Amazon EC2 instances, load balancers, databases, and networking configurations.

**Versions and Deployments**: Elastic Beanstalk allows you to deploy different versions of your application, making it easy to roll back to previous versions if needed. It supports both single and multiple deployment strategies, allowing for continuous deployment and integration workflows.

**Configuration**: Elastic Beanstalk provides configuration options that allow you to customize various aspects of your environment, including instance types, scaling settings, load balancer configurations, security settings, and environment variables.

**Monitoring and Logging**: Elastic Beanstalk integrates with Amazon CloudWatch, allowing you to monitor the health and performance of your applications in real-time. It also provides access to logs, metrics, and events to help you

troubleshoot issues and optimize performance.

 IAM: (The Gatekeeper of Security)

IAM (Identity and Access Management) is a web service provided by AWS that helps you securely control access to AWS resources. IAM allows you to manage users, groups, roles, and permissions, enabling you to grant or deny access to specific resources or actions within your AWS account.

IAM enables you to create and manage user identities, assign permissions using policies, and set up multi-factor authentication (MFA) for added security. With IAM, you can follow the principle of least privilege, ensuring that users have only the permissions they need to perform their tasks, thereby reducing the risk of unauthorized access or accidental misuse of resources.

IAM, adds another layer of security to your Elastic Beanstalk applications. It lets you control who can access your resources and what actions they can perform. Think of it as a bouncer at a nightclub, ensuring only authorized users get in and preventing unwanted guests from causing trouble.

With IAM, you can:

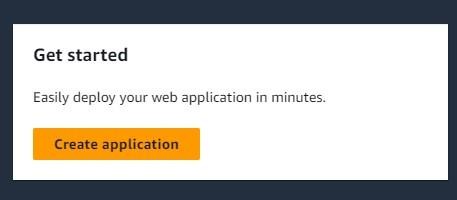
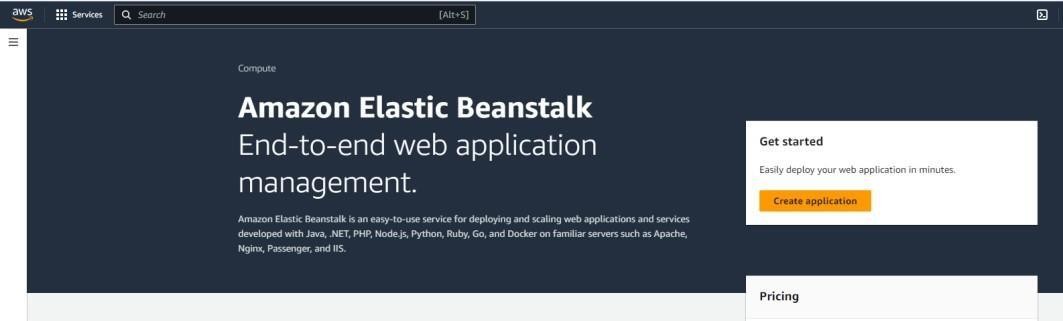
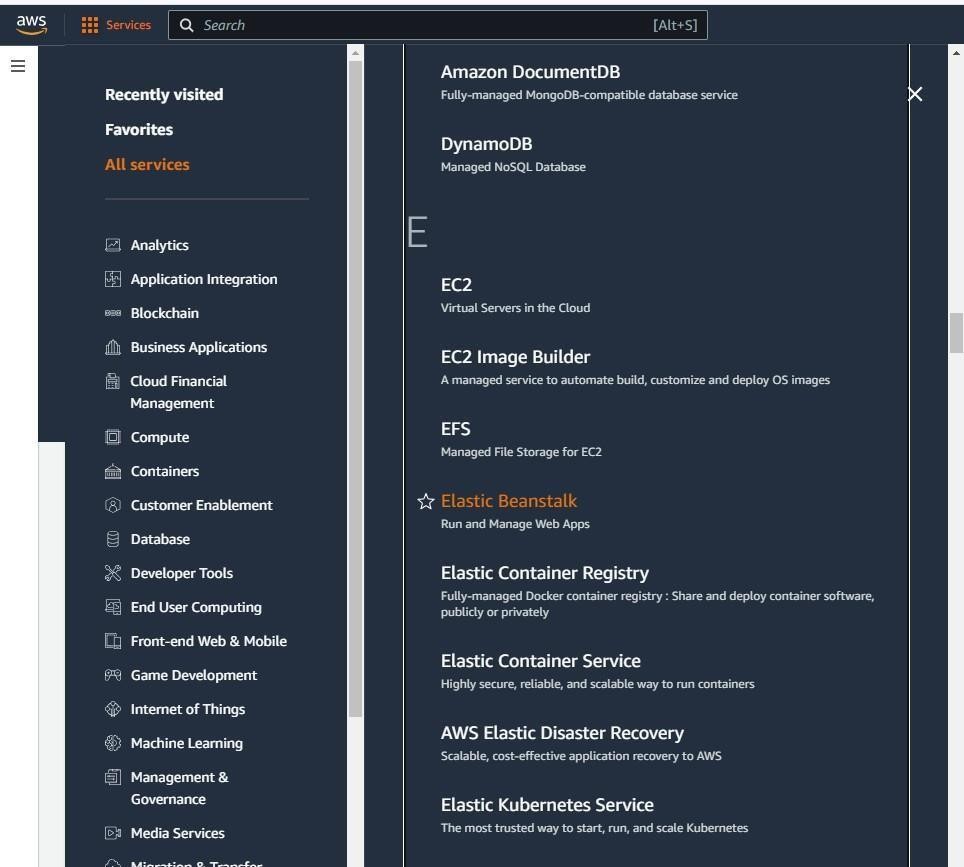
* Create users and groups with specific permissions.
* Define roles that grant access to specific resources and actions.
* Use temporary credentials for short-lived tasks.
* Monitor user activity and identify potential security threats.

By integrating IAM with Elastic Beanstalk, you can ensure that your applications are secure and only authorized users can access and modify them.

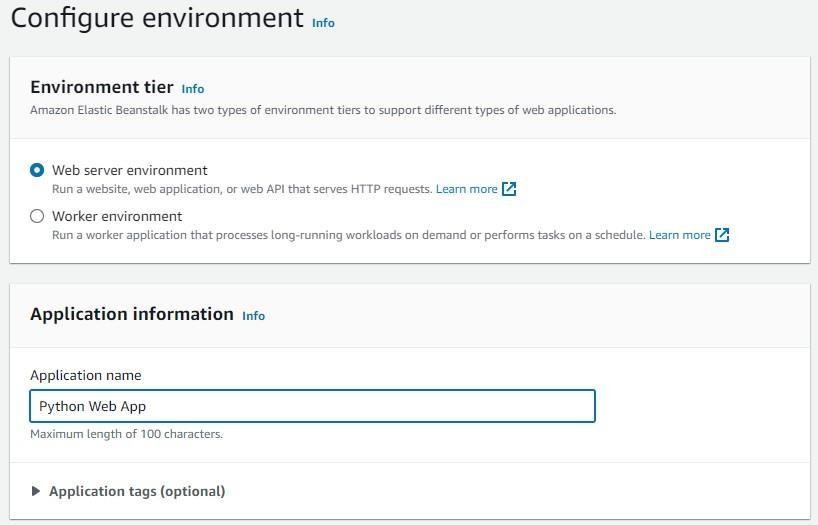
Q2) Implement paas using elastic beanstalk for the following.

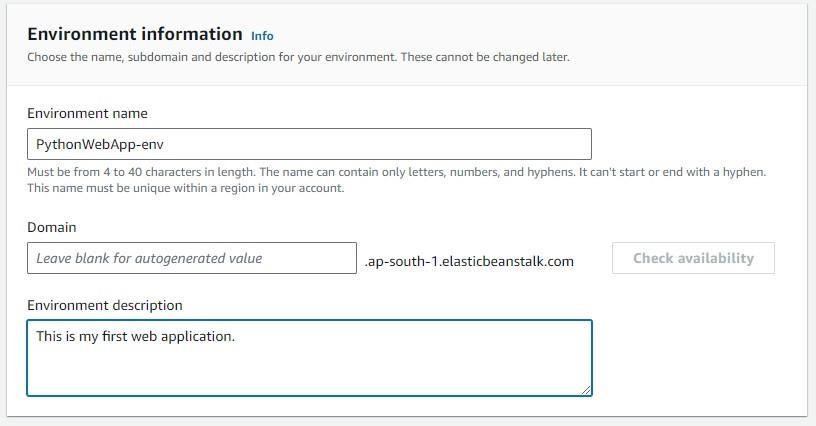
1. Server
2. Java
3. Python
4. Node.js

Step

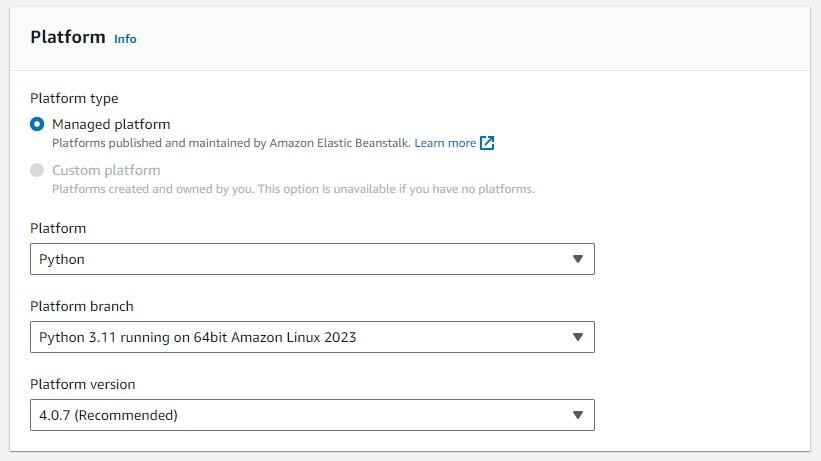
1: create web app

Step

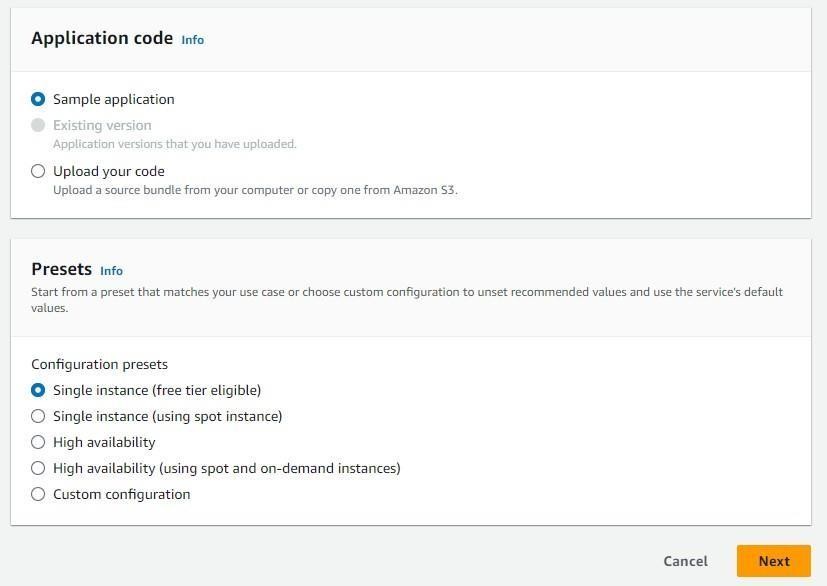
2: Name your application

Step 3:Give environment name and add a short description for example 4:Select Platform on which you want your application and the versions

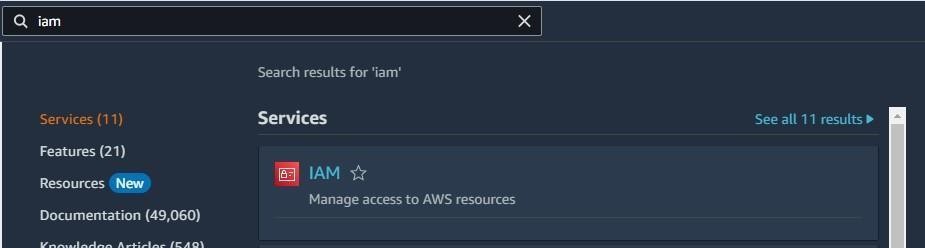
Step



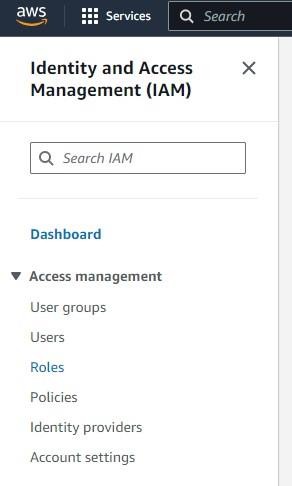
Step 5:



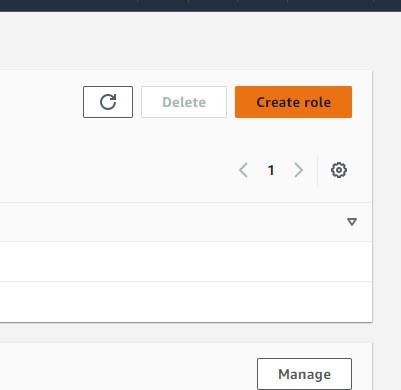
Step 6:Create a role. ( open new tab and then perform , keep previous tab as it is we want to work on it later )



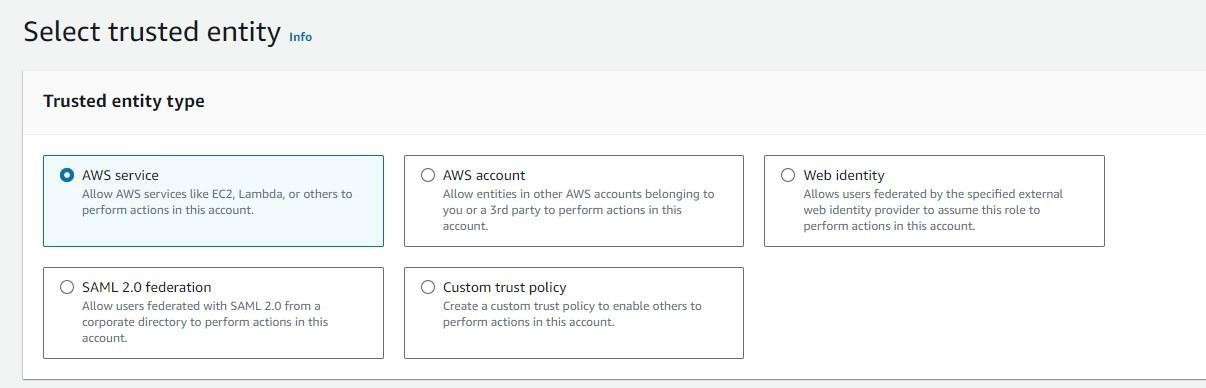
Step 7:



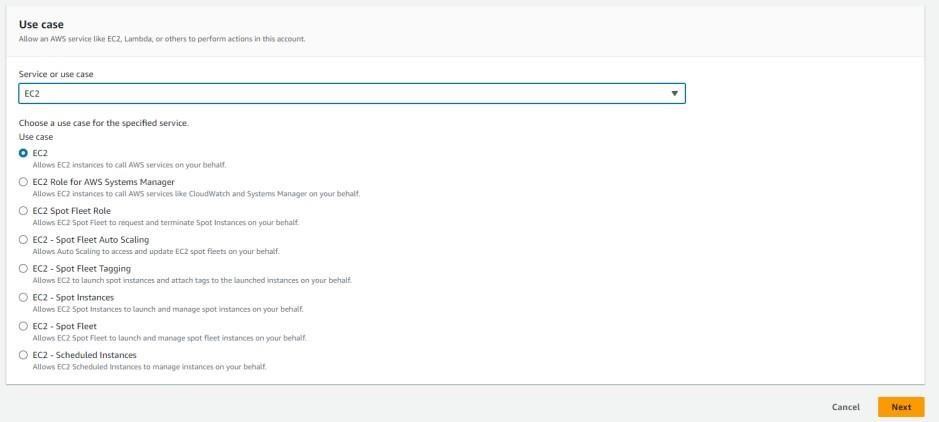
Step 8:



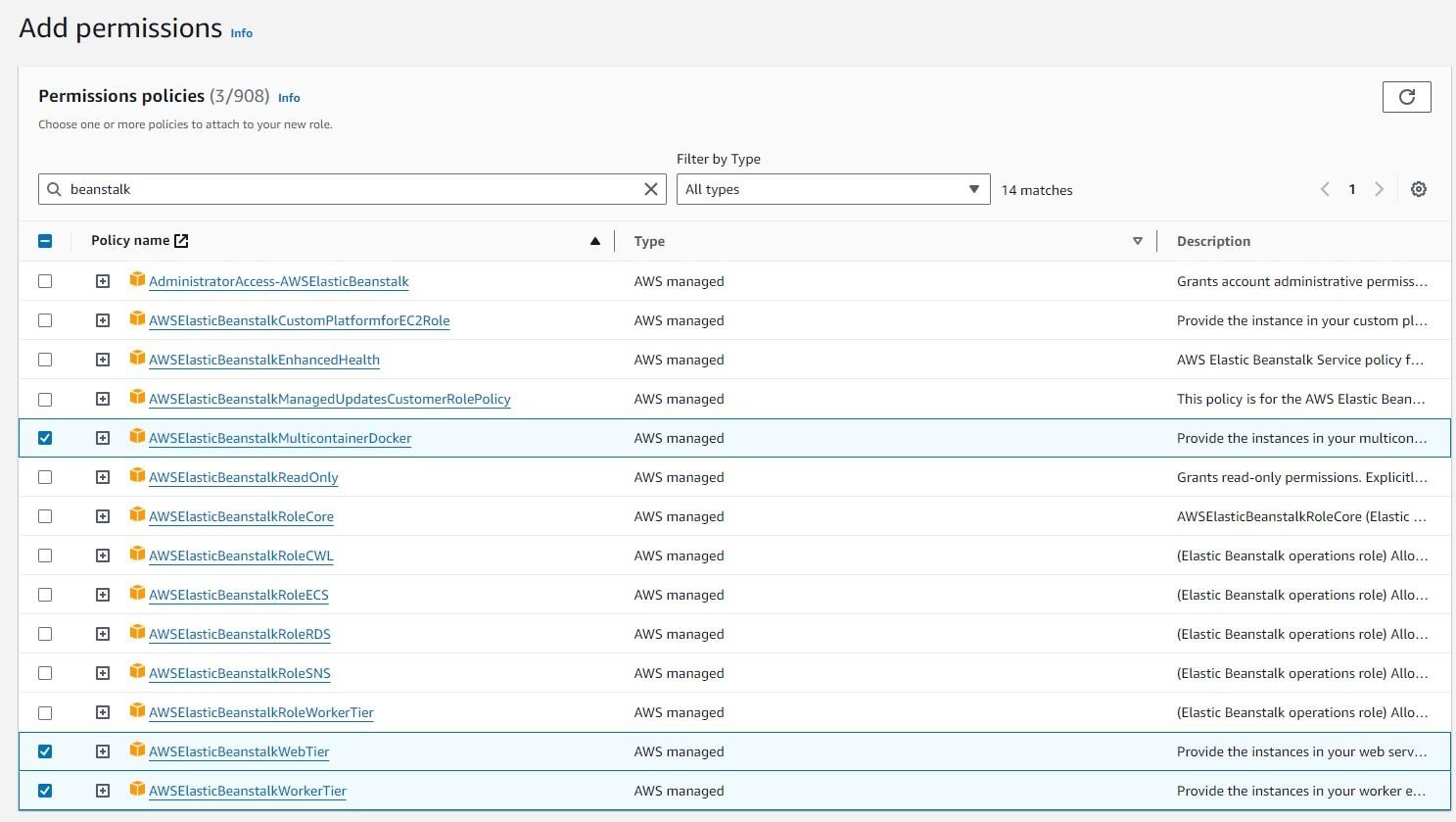
Step 9:Select entity



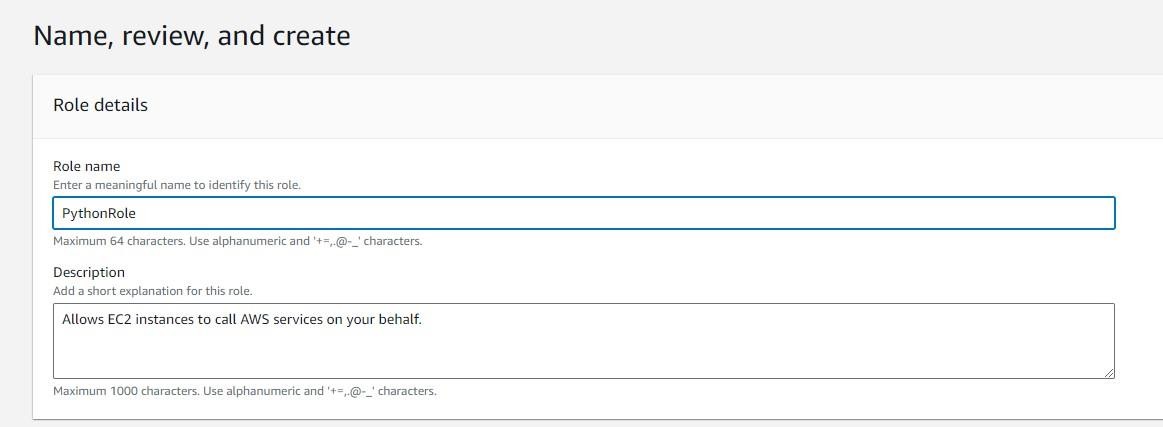
Step 10:Select use case



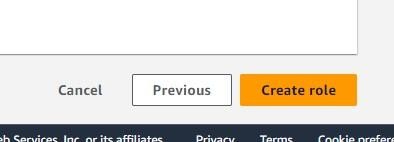
Step 11:Toggle permission’s



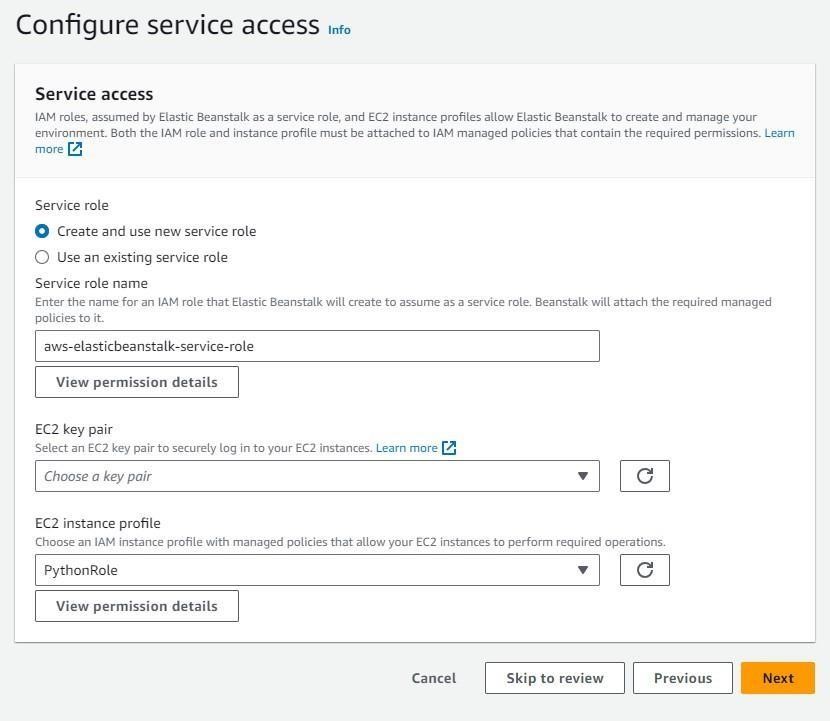
Step 12: Give name to role

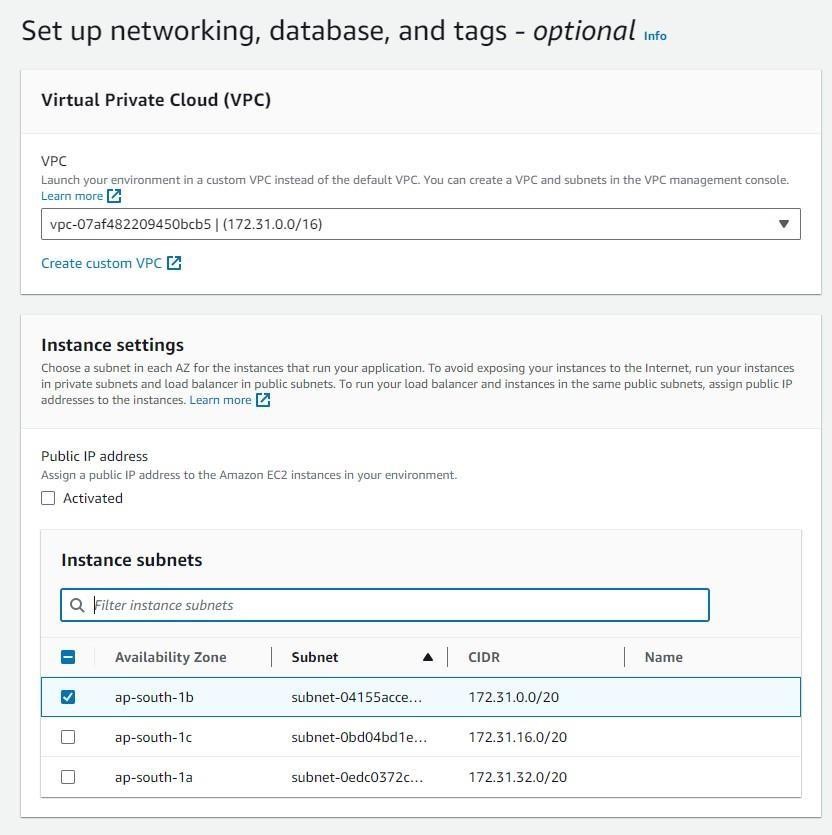


Step 13:

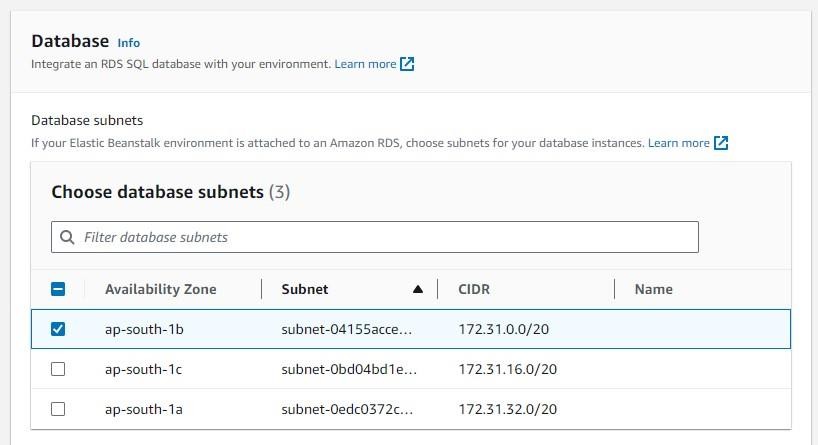


Step 14: Select role in previous working page

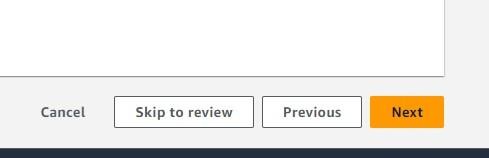


Step 15:Select database

Step 16:

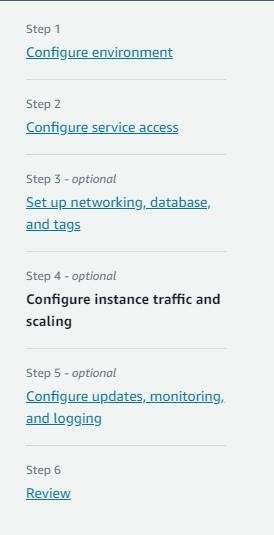


Step 17:

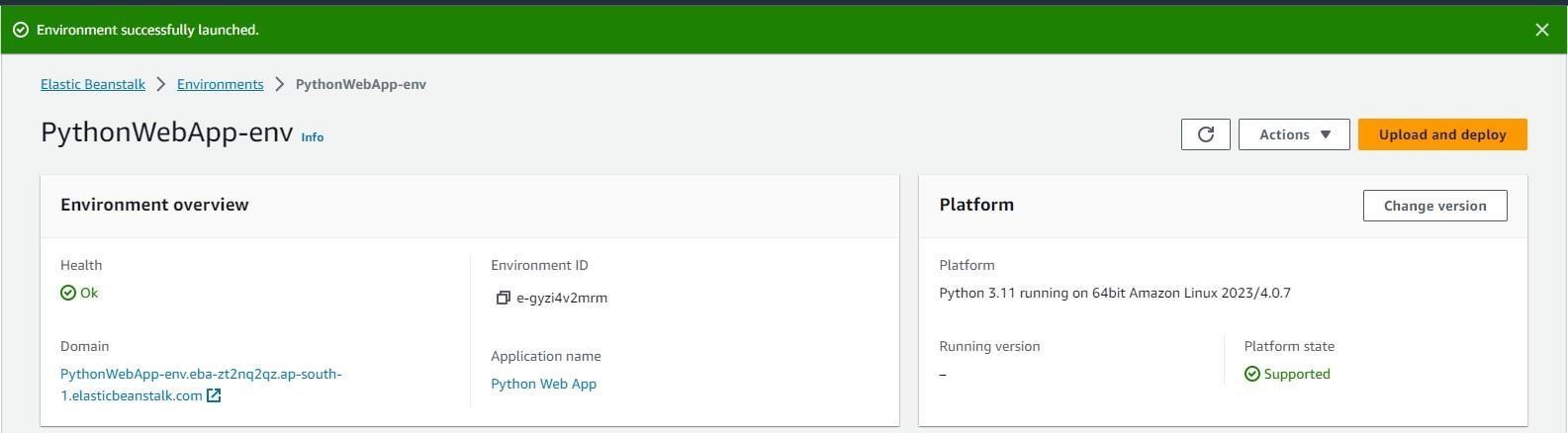
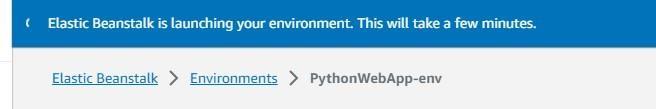


Step 18:

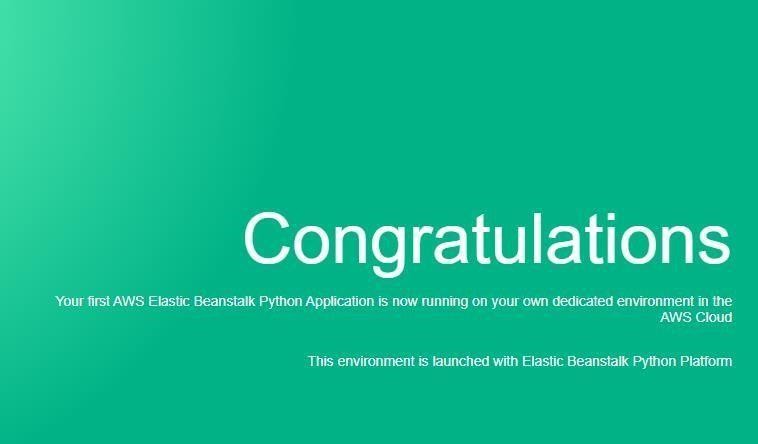
Skip Step 4: Keep as it is

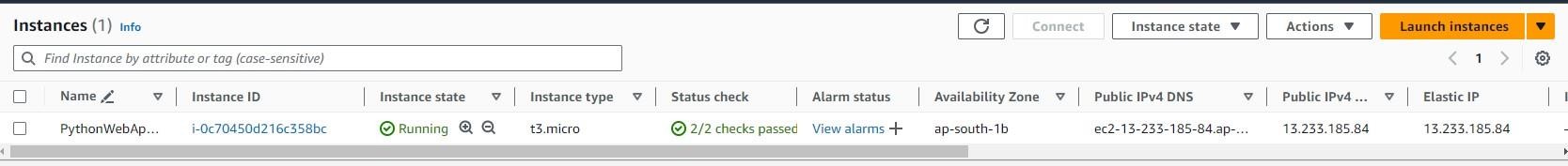
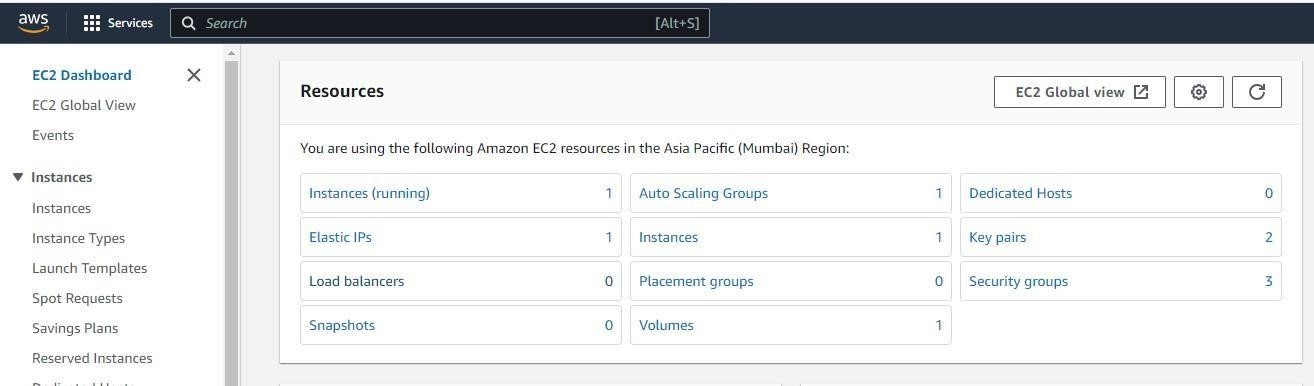


Step 19:

Step 5 : Keep as it is and submit Step 20: Launching environment

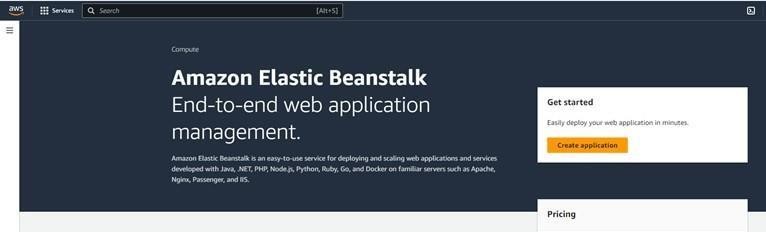
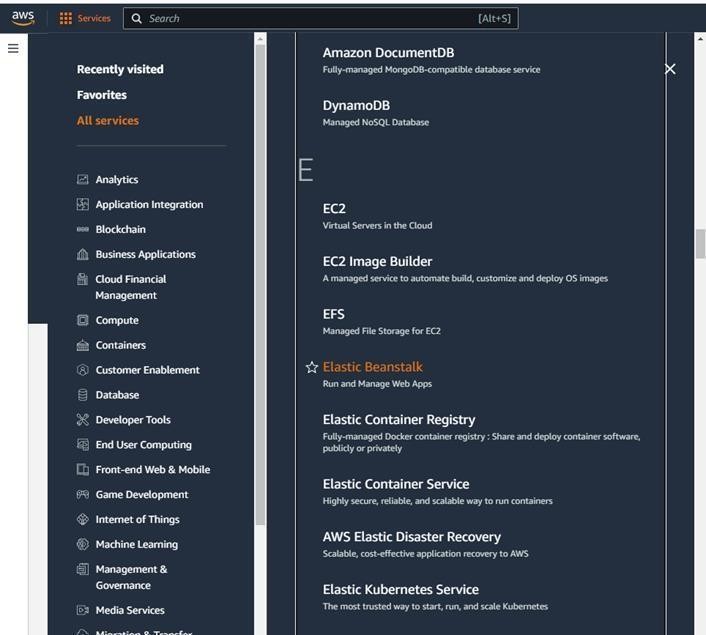
Step 21: Successfully launched our application

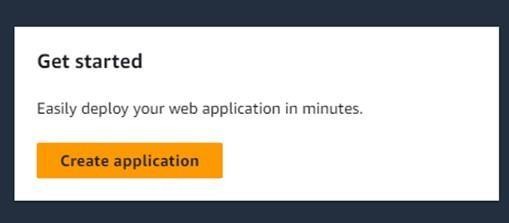


Step 22: If successfully done you will se instances running (1)

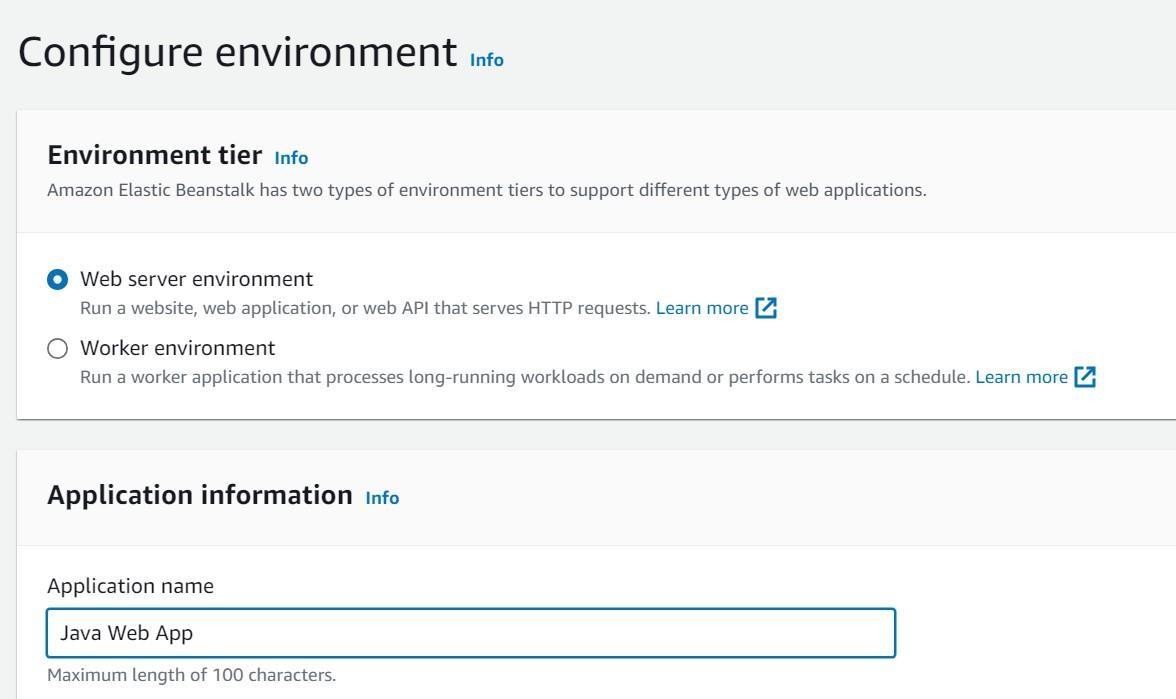
**For Java:**

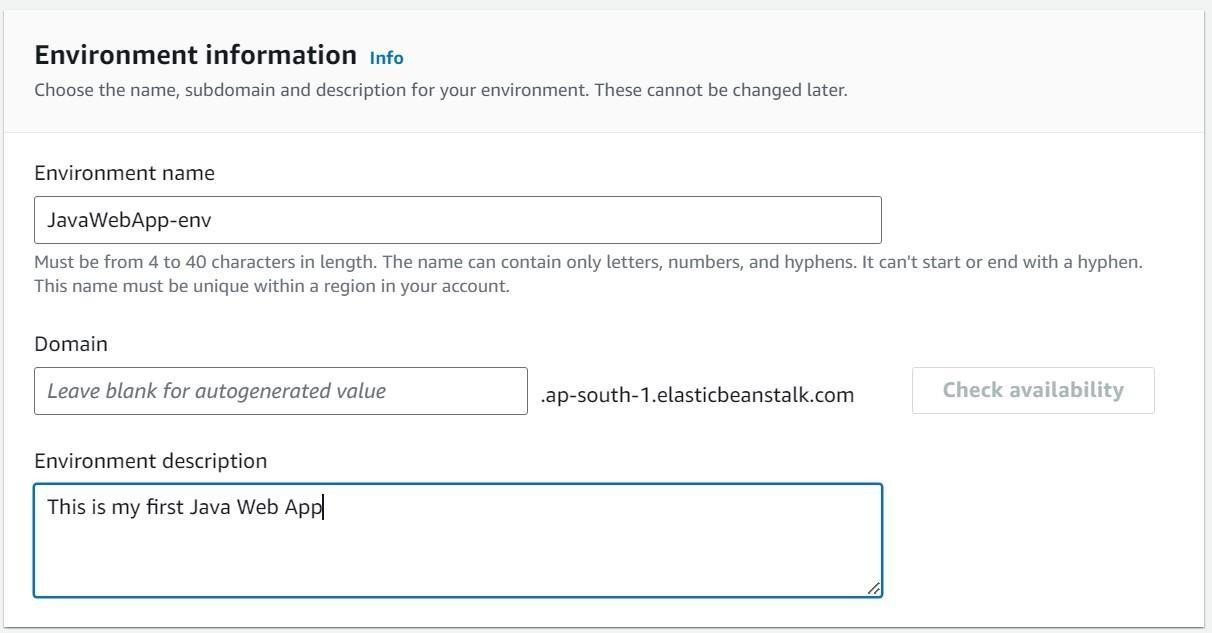
Step 1: create web app



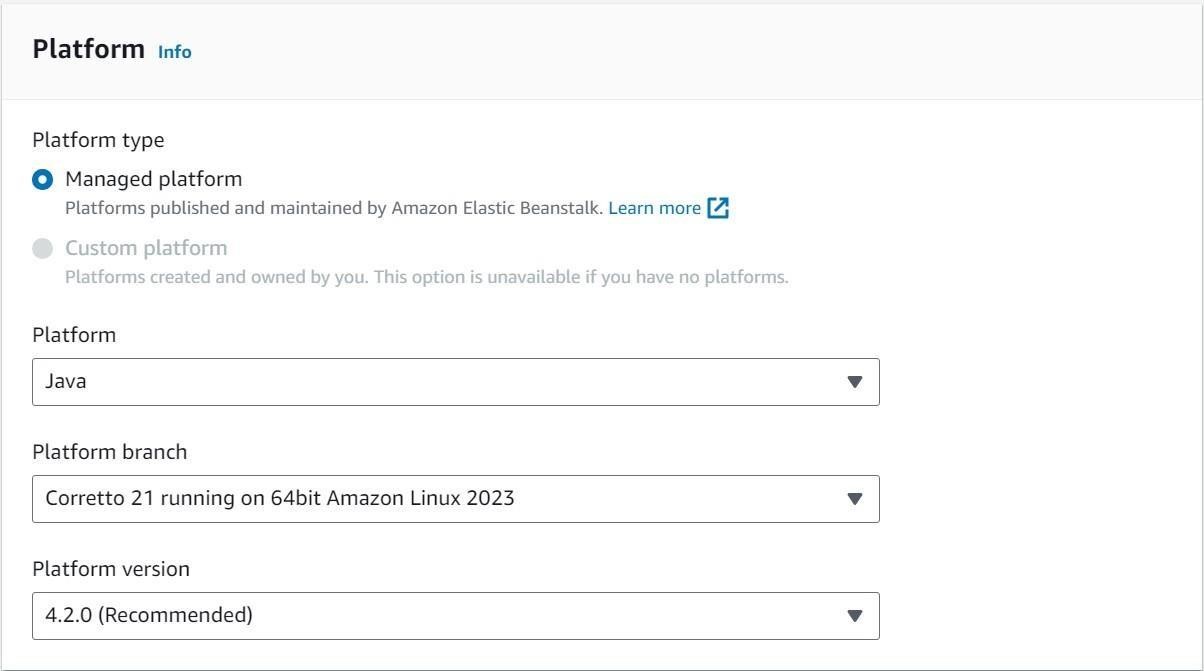


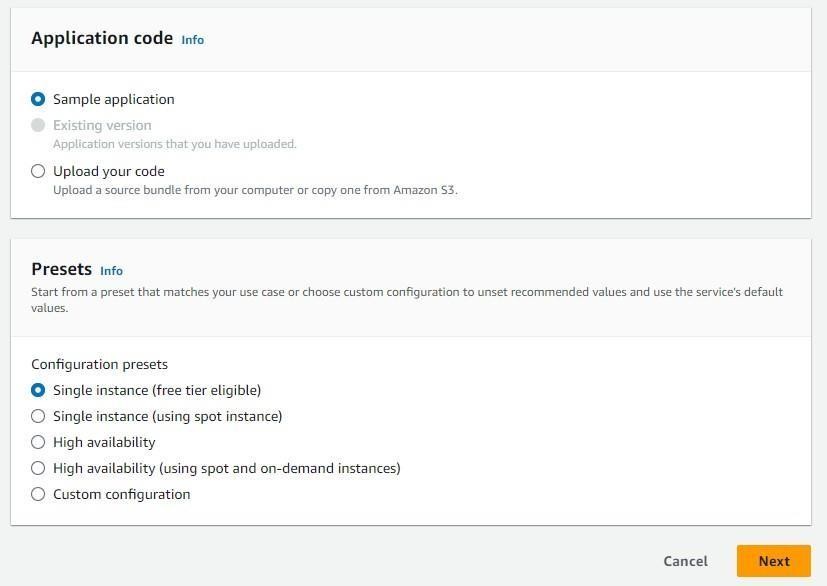
Step 2: Name your application



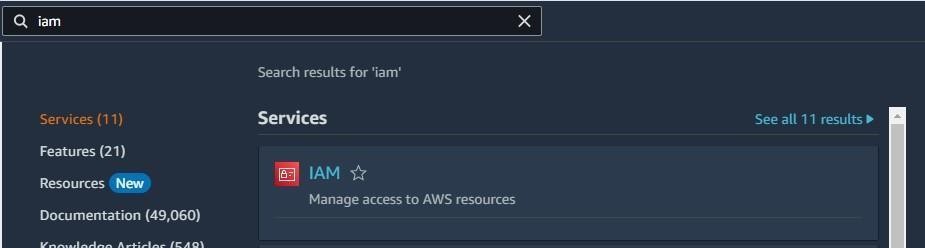
Step 3

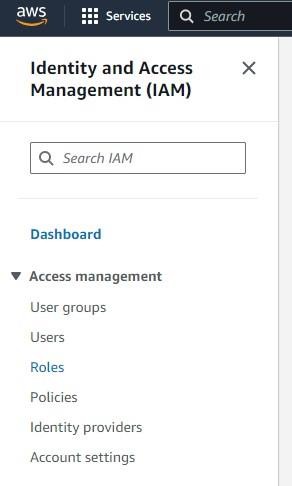
Step 4:Select Platform on which you want your application and the versions



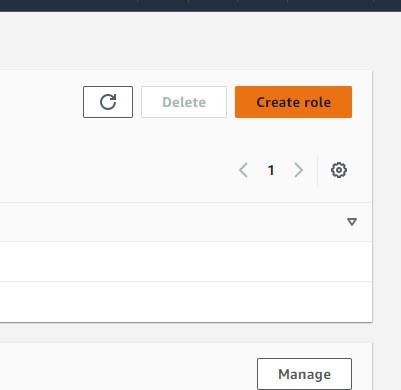
Step 5:

Step 6:Create a role. ( open new tab and then perform , keep previous tab as it is we want to work on it later )

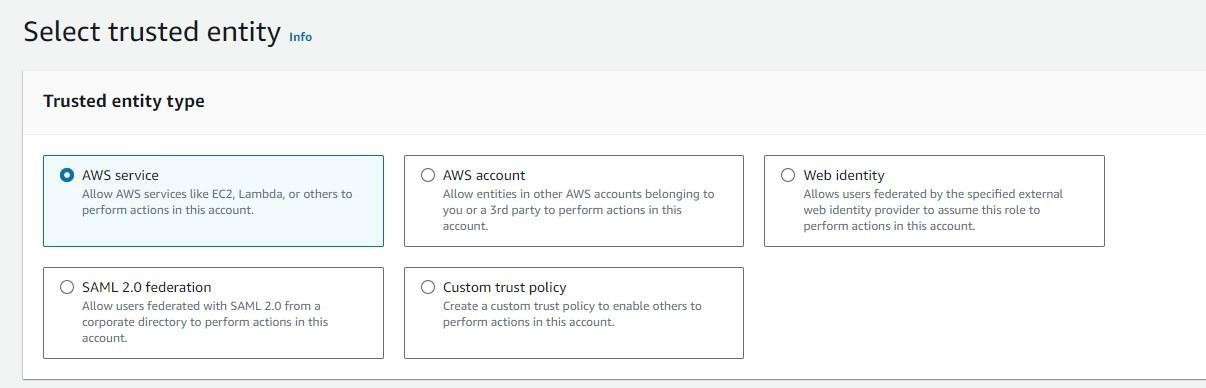


Step 7:

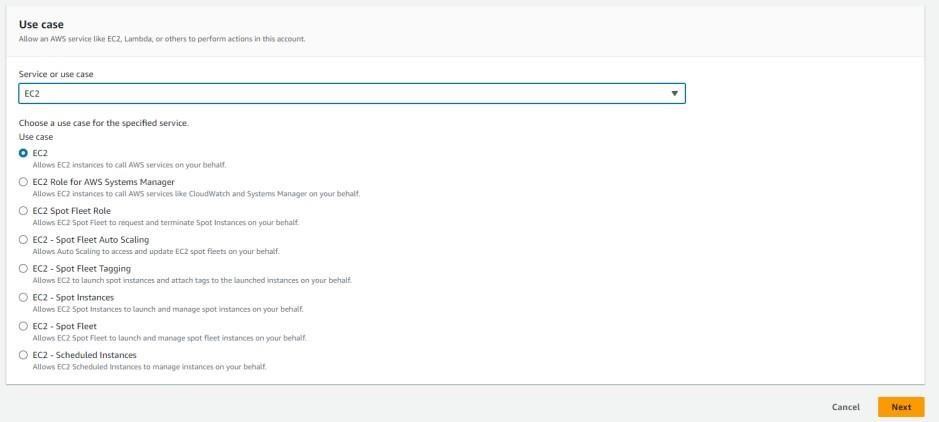
Step 8:



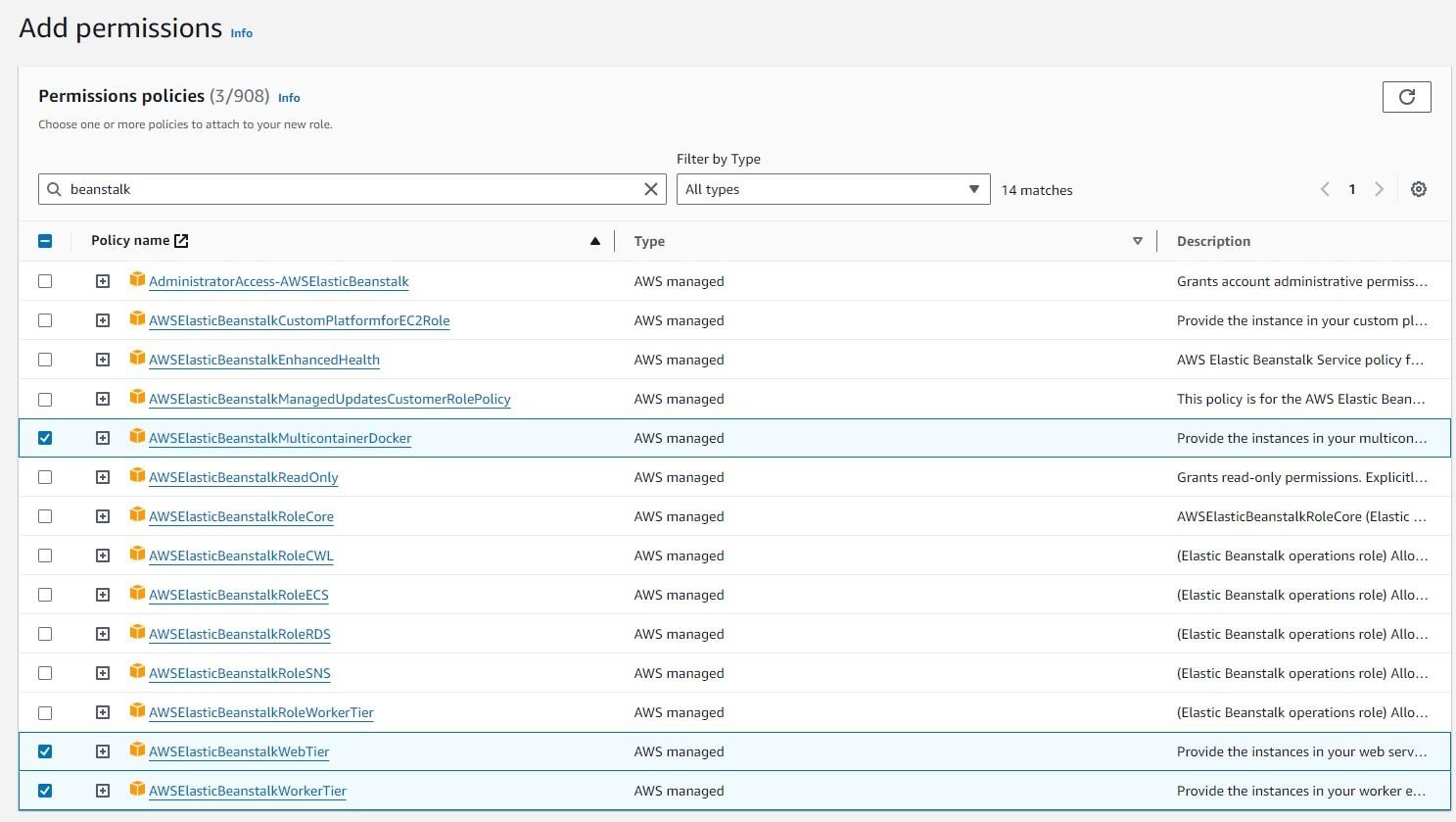
Step 9:Select entity



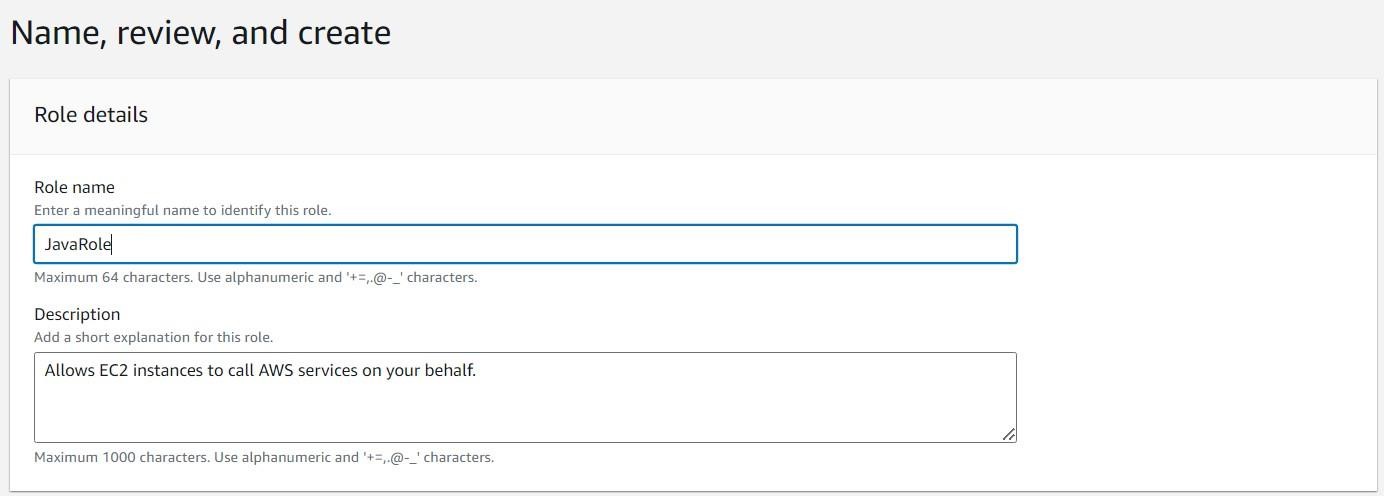
Step 10:Select use case



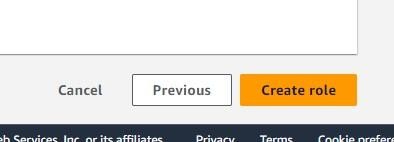
Step 11:Toggle permission’s



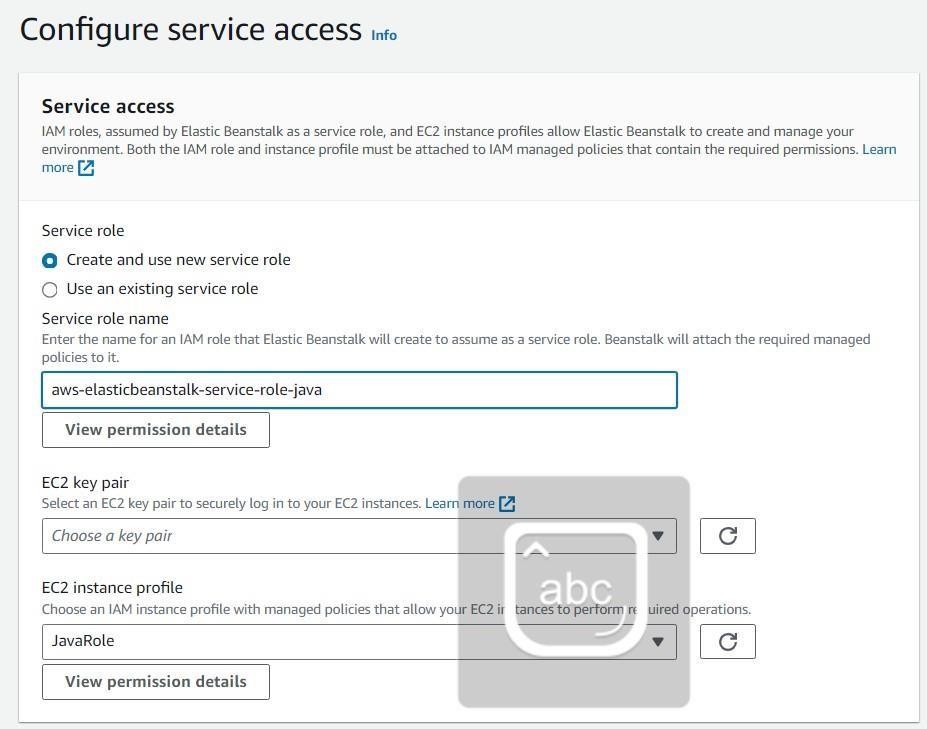
Step 12: Give name to role



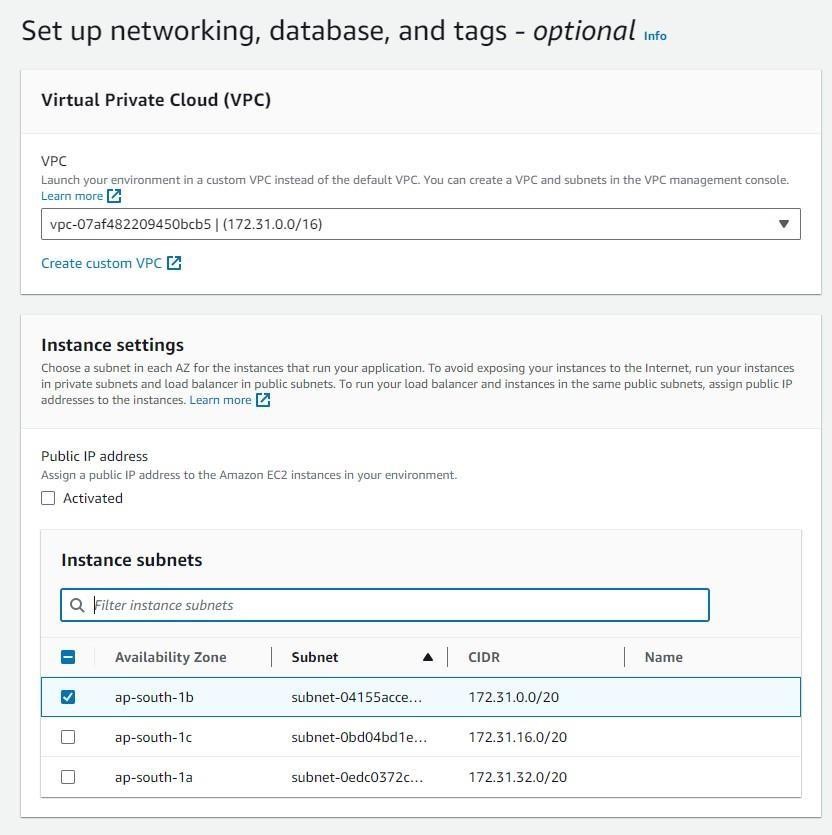
Step 13:



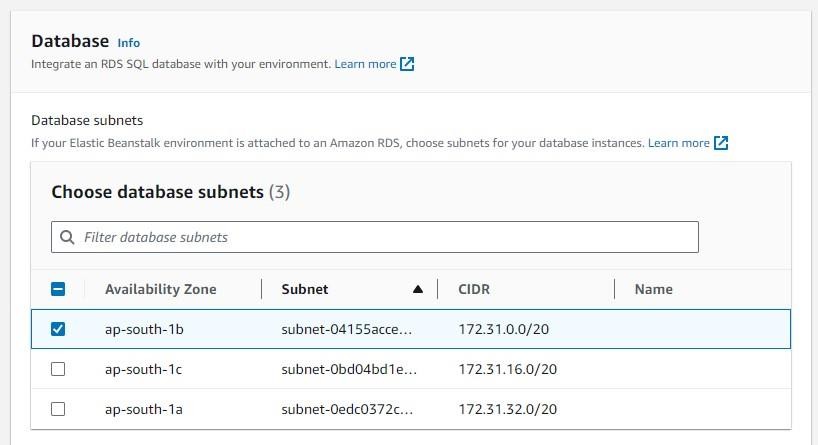
Step 14: Select role in previous working page



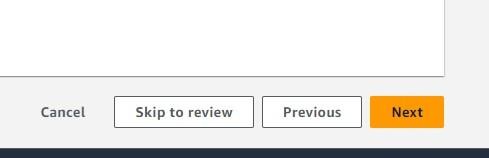
Step 15:Select database



Step 16:

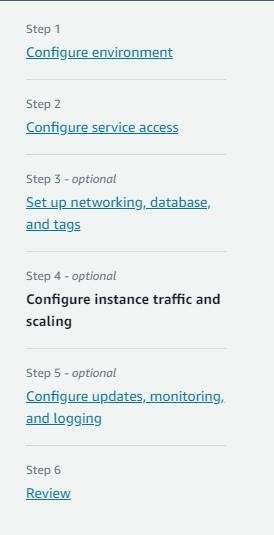


Step 17:



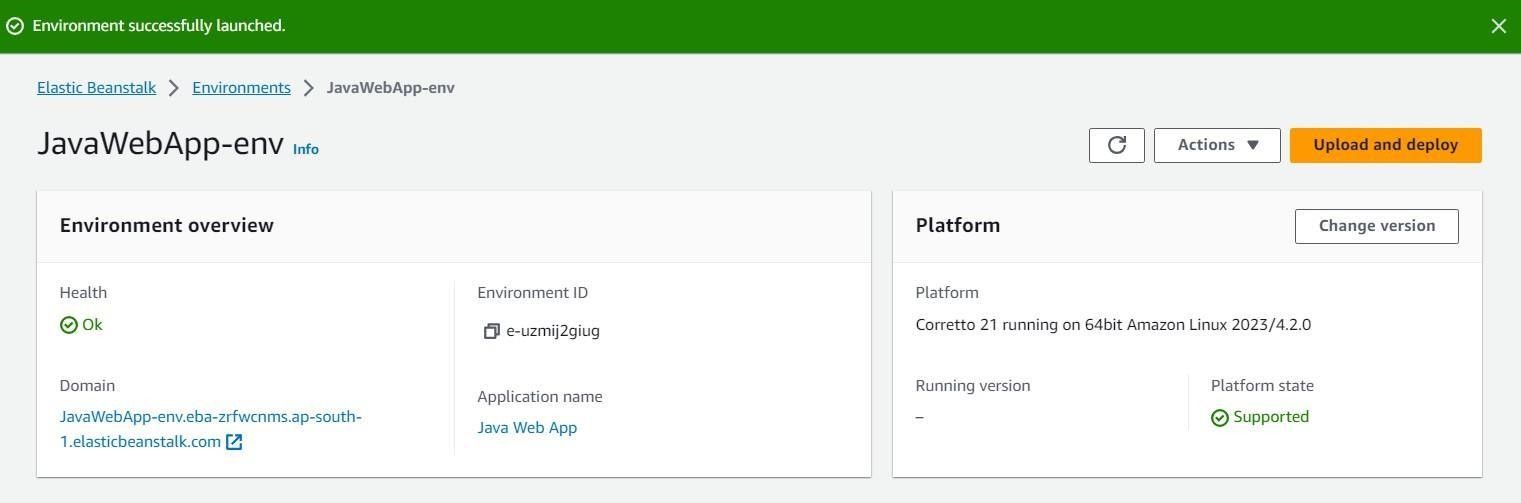
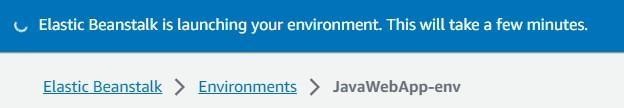
Step 18:

Skip Step 4: Keep as it is

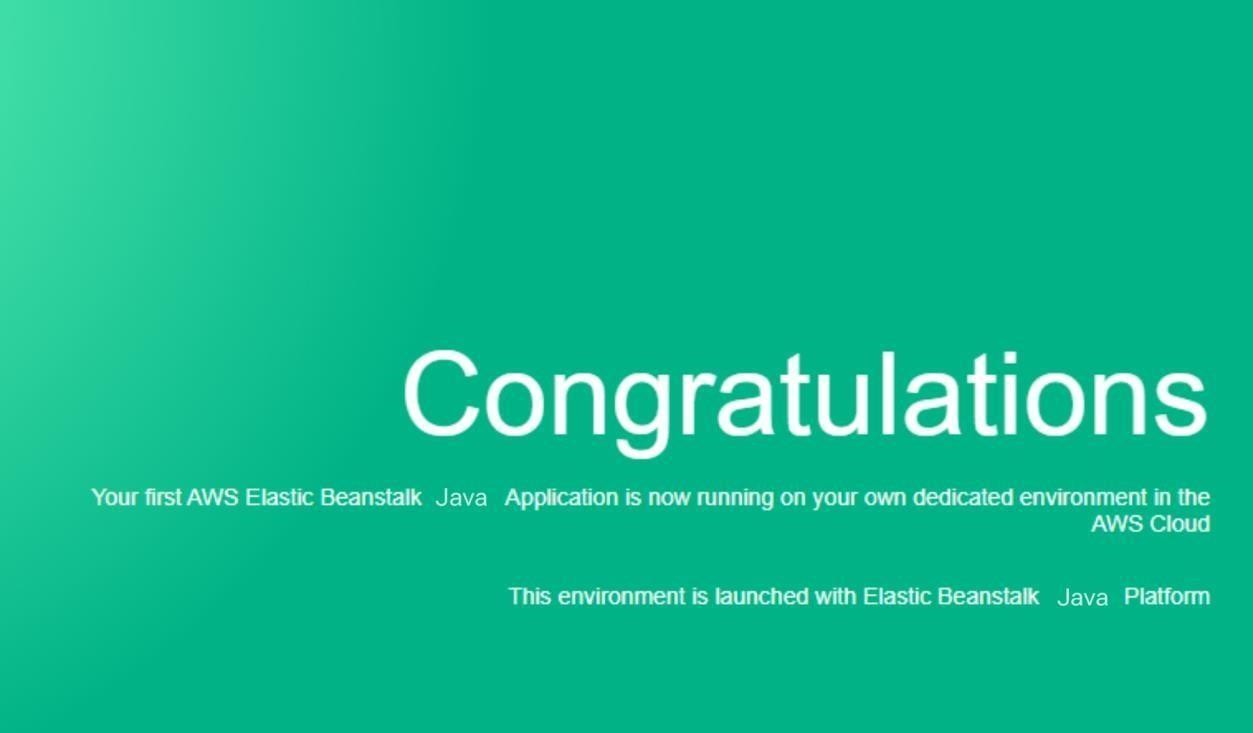


Step 19:

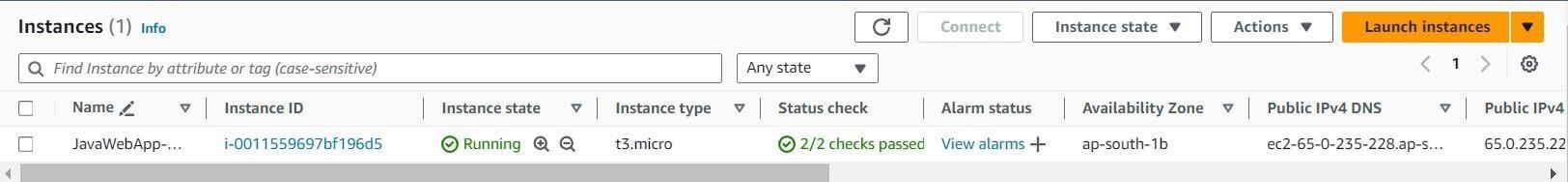
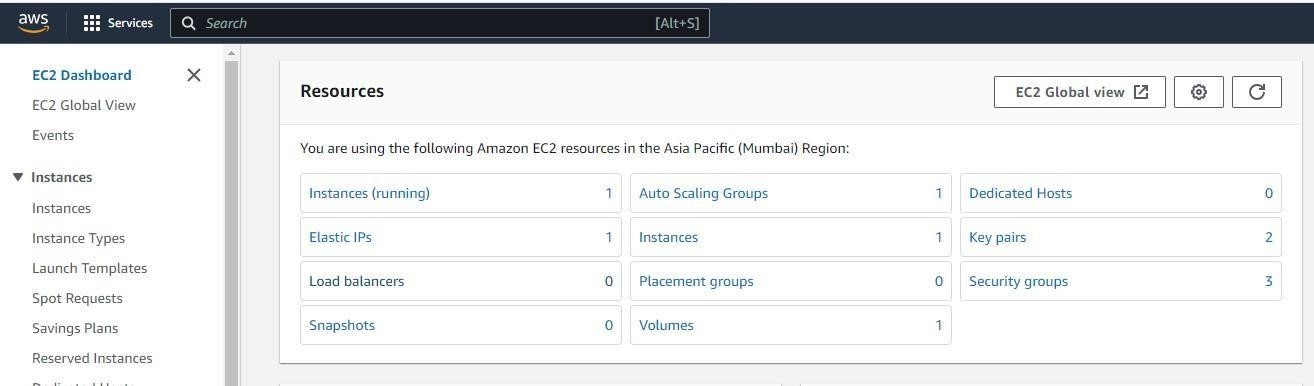
Step 5 : Keep as it is and submit Step 20: Launching environment



Step 21: Successfully launched our application

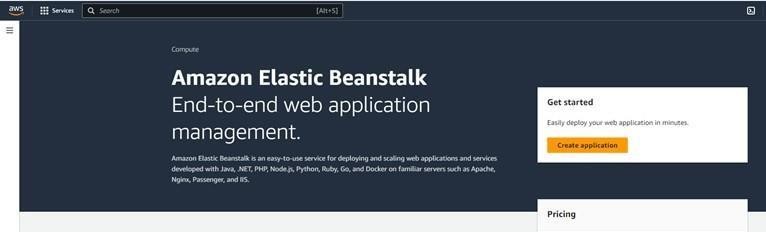
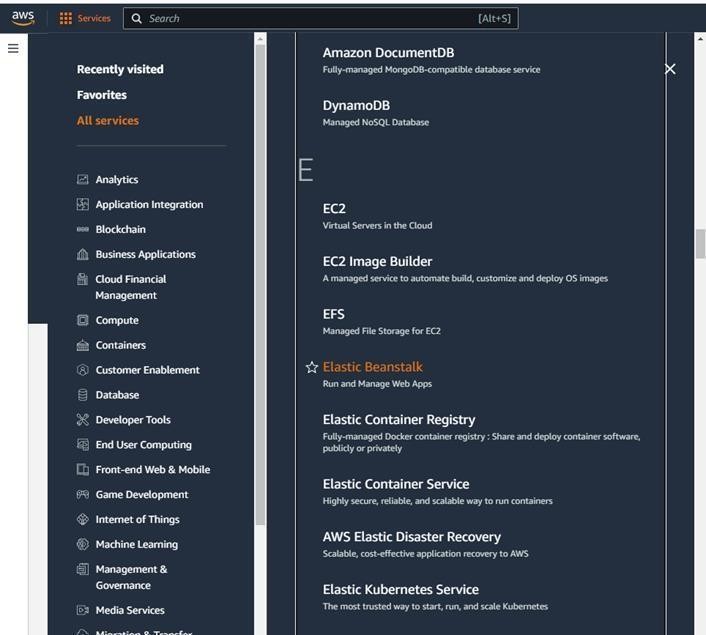


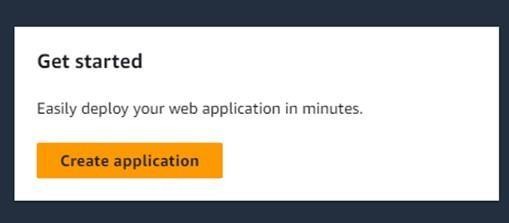
Step 22: If successfully done you will se instances running (1)



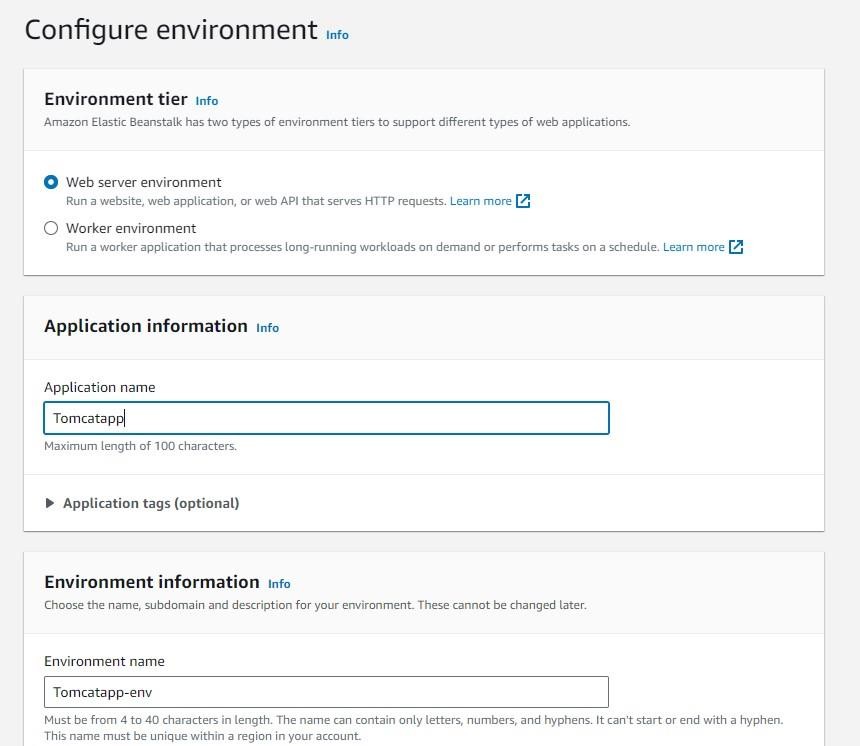
**Implement PaaS using elastic beanstalk with Tomcat Applicatio**

Step 1: create web app

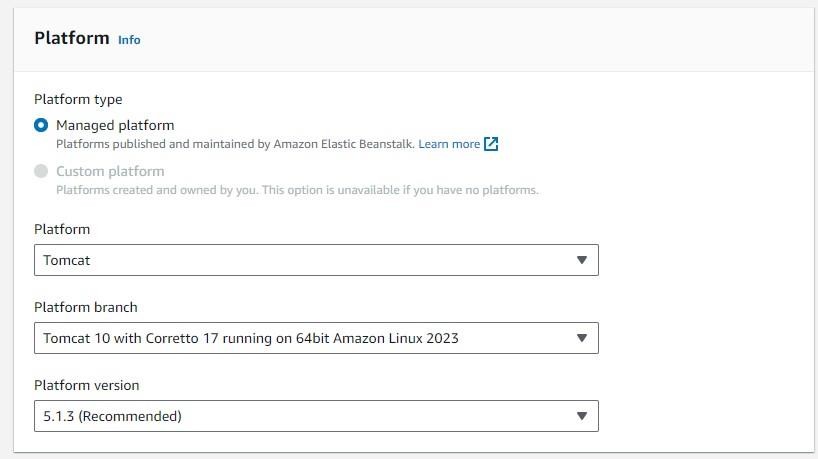




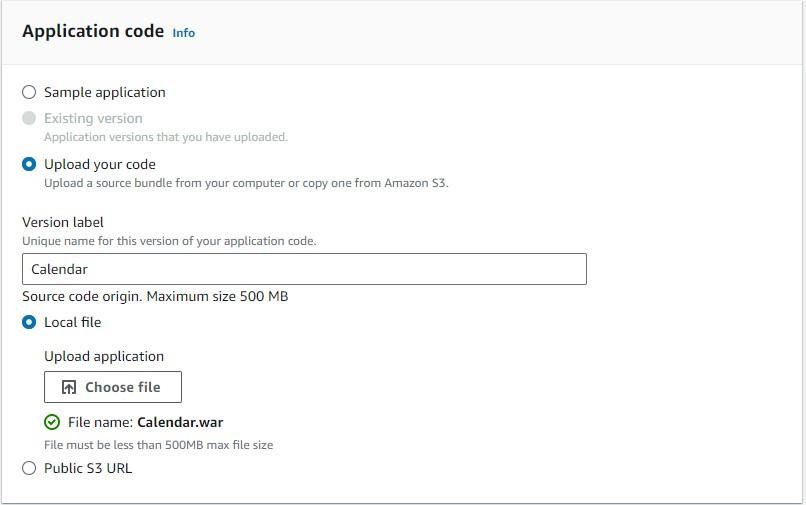
Step 2: Name your application



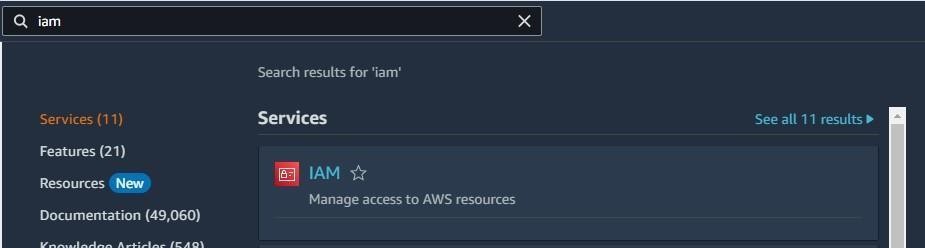
Step 4: Select Platform on which you want your application and the versions



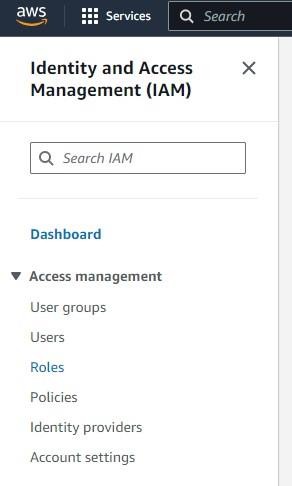
Step 5: Upload the Calendar.war file.



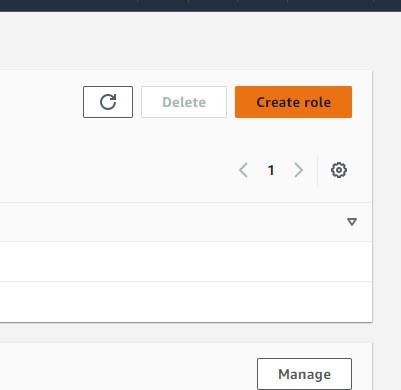
Step 6: Create a role. (open new tab and then perform, keep previous tab as it is we want to work on it later)



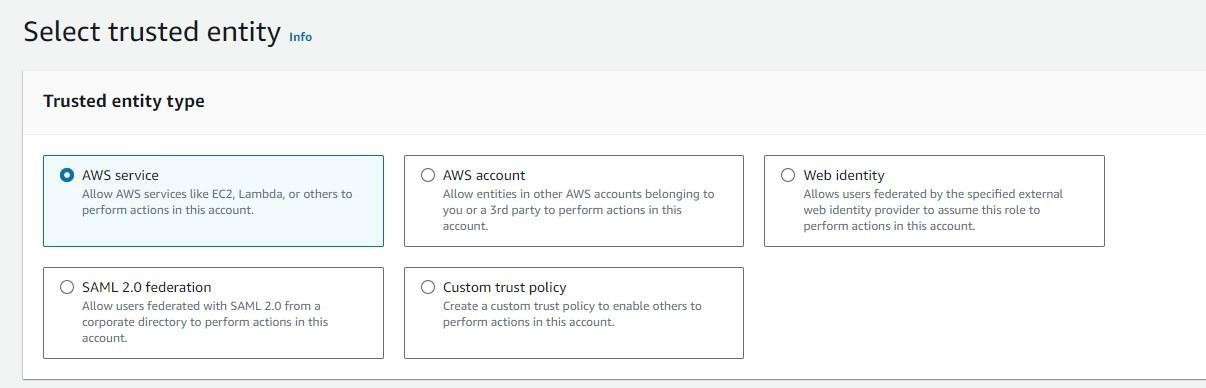
Step 7:



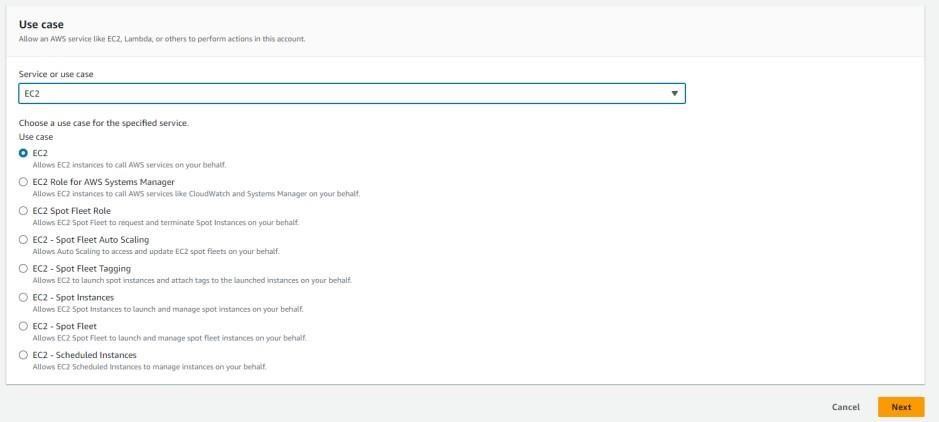
Step 8:



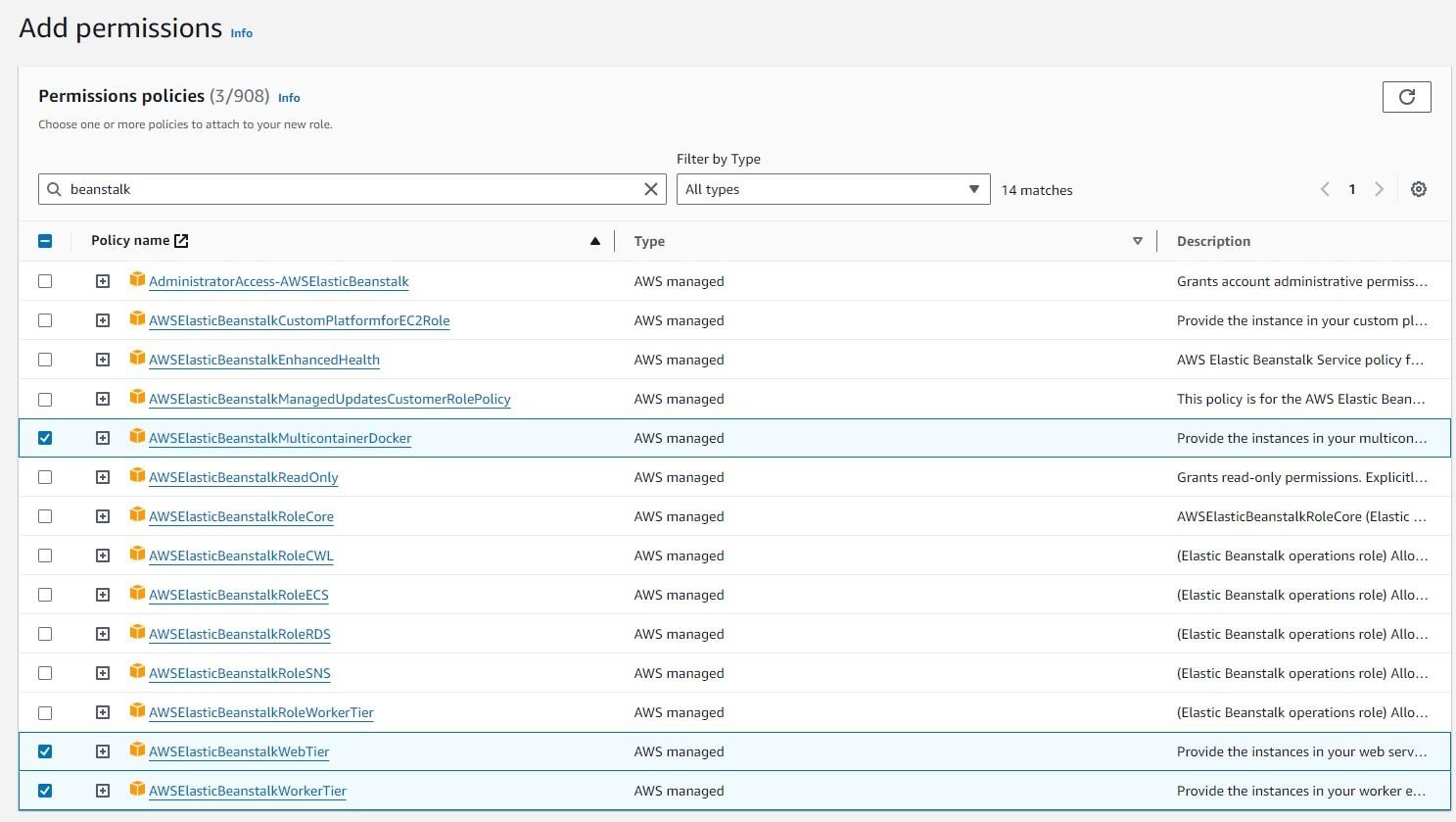
Step 9: Select entity



Step 10: Select use case



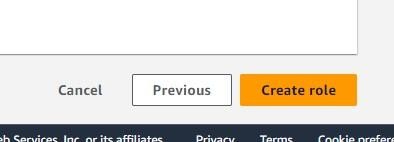
Step 11: Toggle permission’s



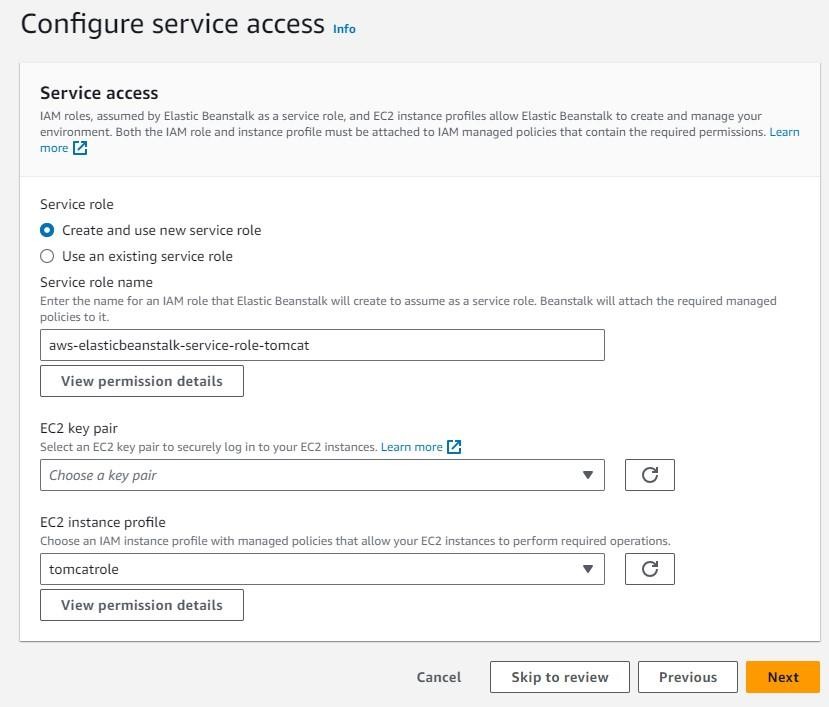
Step 12: Give name to role



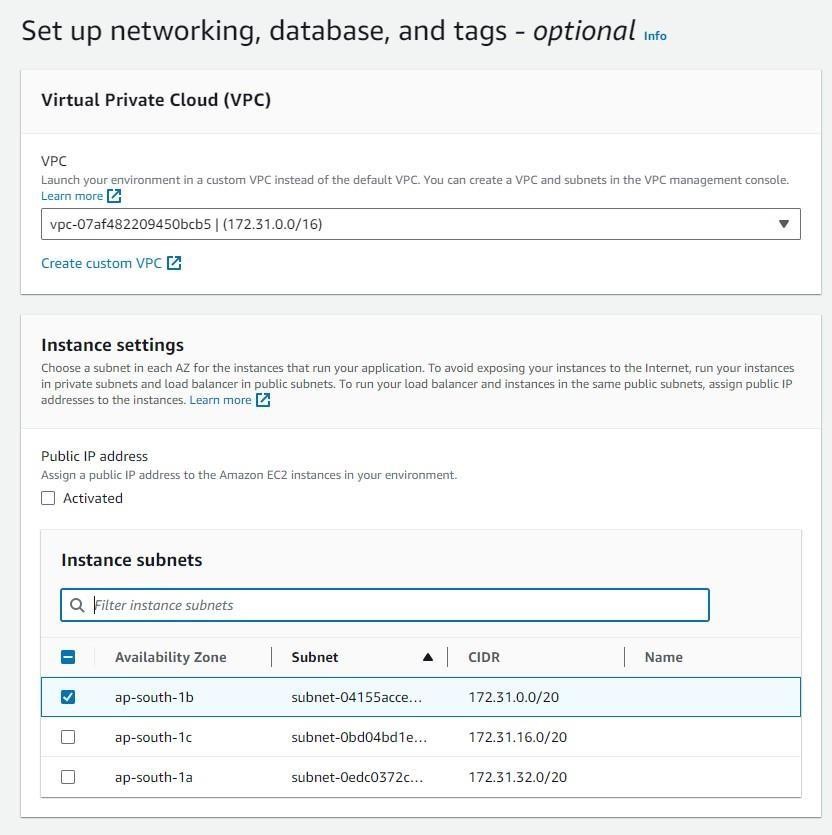
Step 13:



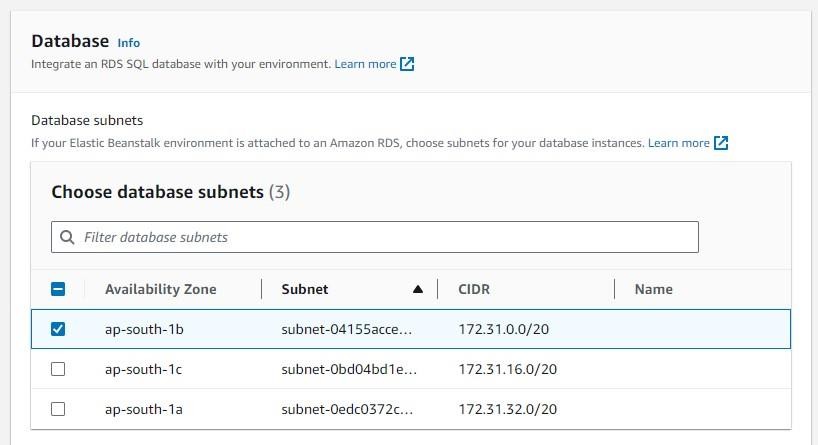
Step 14: Select role in previous working page



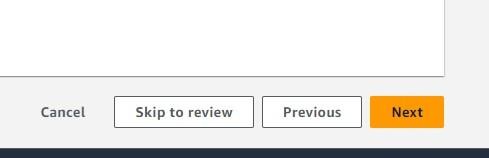
Step 15: Select database



Step 16:

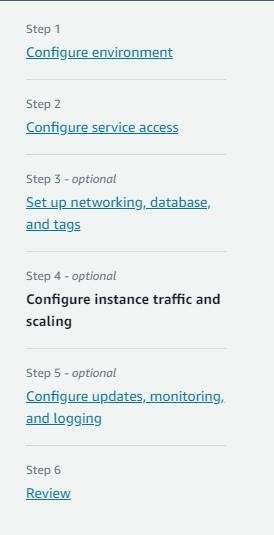


Step 17: Click Next

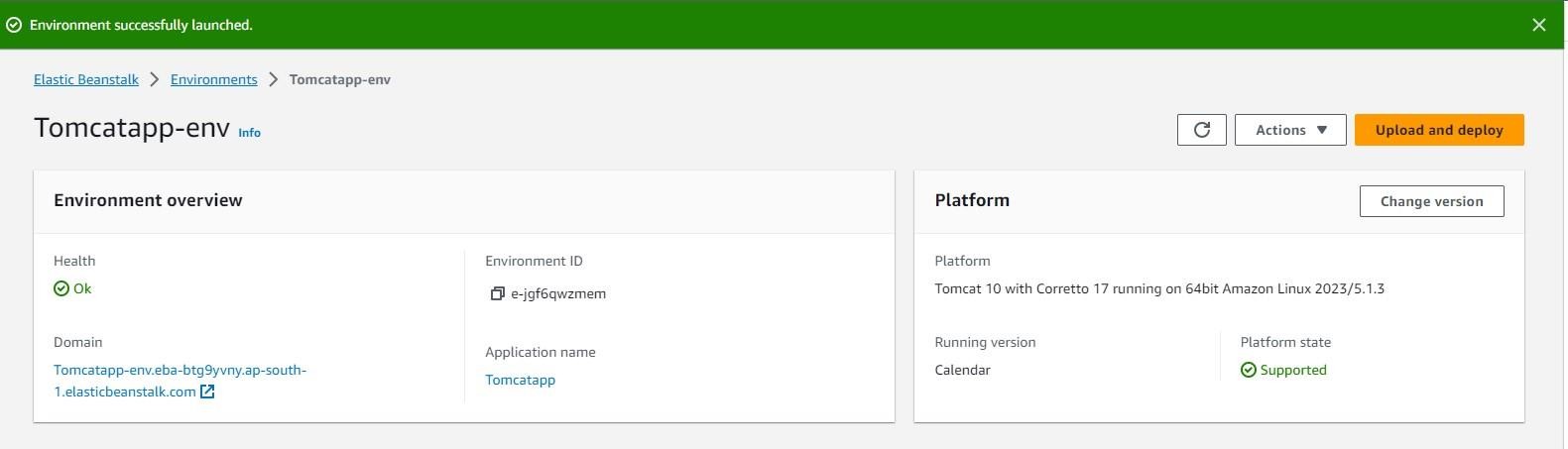


Step 18:

Skip Step 4: Keep as it is



Step 19:

Step 5: Keep as it is and submit Step 20: Launched

