Practical-6 Platform as a service using AWS

Name: Sahil Jadhav

Roll no: A021

MSc SDS Batch 1

1. We’ll be using Elastic Beanstalk - Python

2. There will be the steps

3. Fill in the details

Step 1

Configure environment

Step 2

Configure service access

Step 3 - *optional*

Set up networking, database, and tags

Step 4 - *optional*

Configure instance traffic and scaling

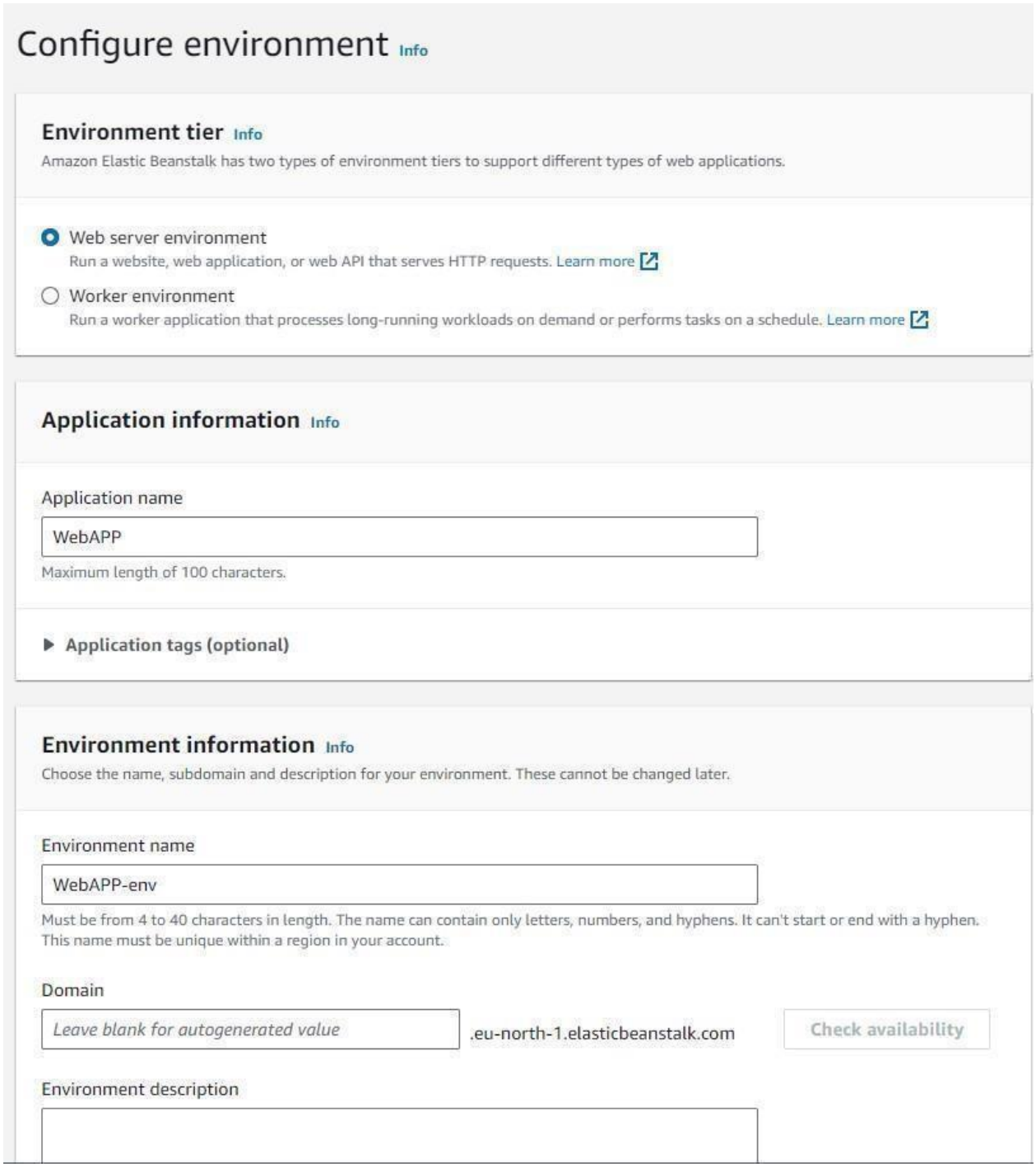
Step 5 - *optional*

Configure updates, monitoring, and logging

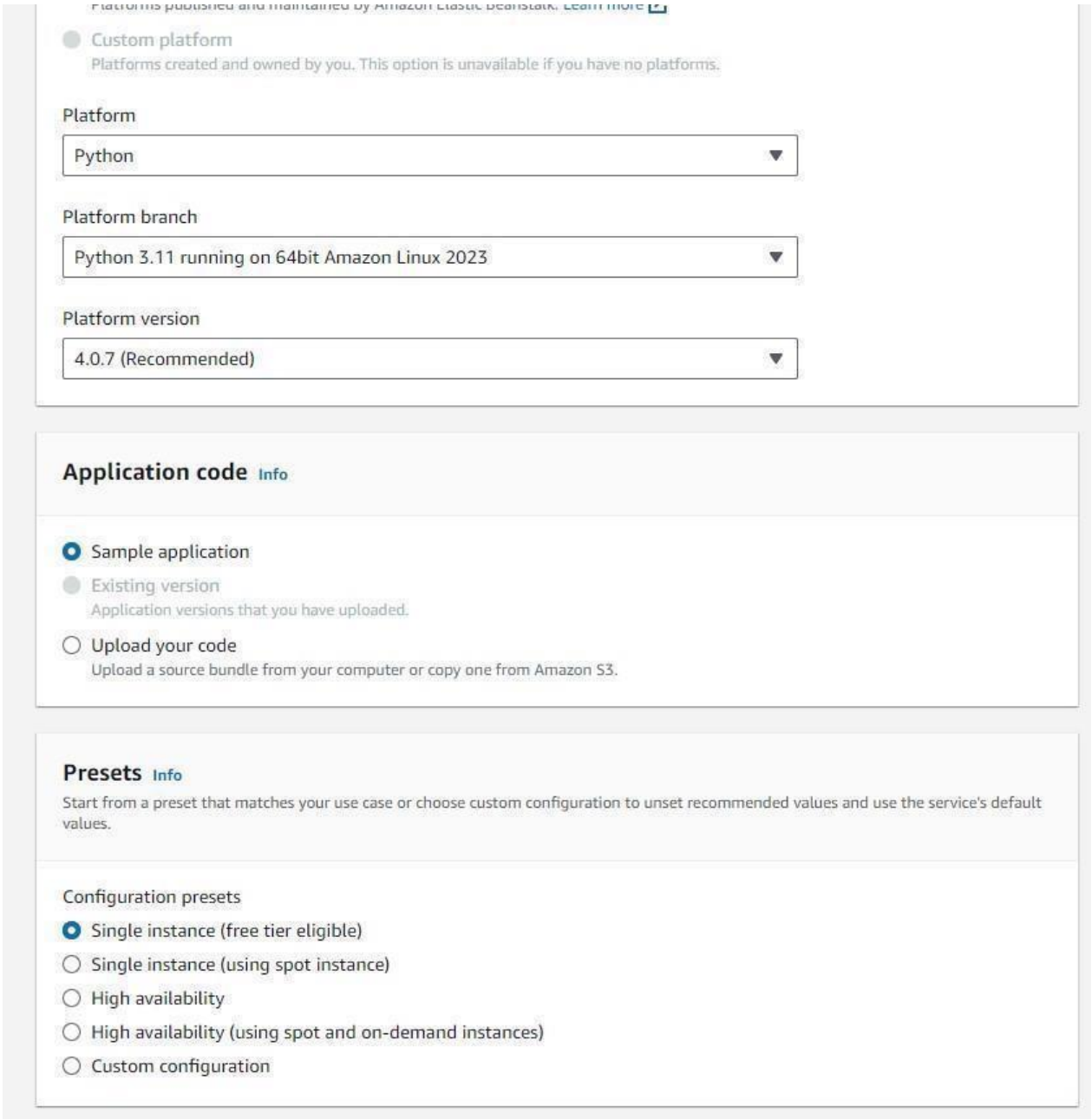
Step 6

Review

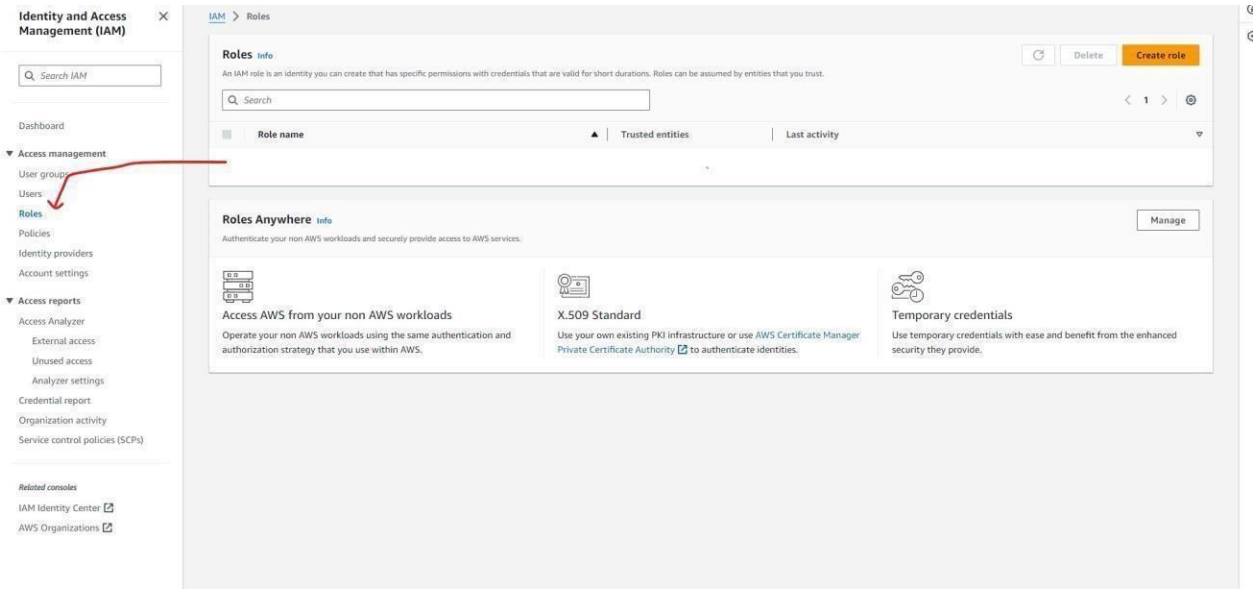
3. Fill in the details



4. Select the platform



5. Go to the IAM services -> Create role



Trusted entity type

- (●) **AWS service**
  Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- ( ) **AWS account**
  Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- ( ) **Web identity**
  Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- ( ) **SAML 2.0 federation**
  Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- ( ) **Custom trust policy**
  Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

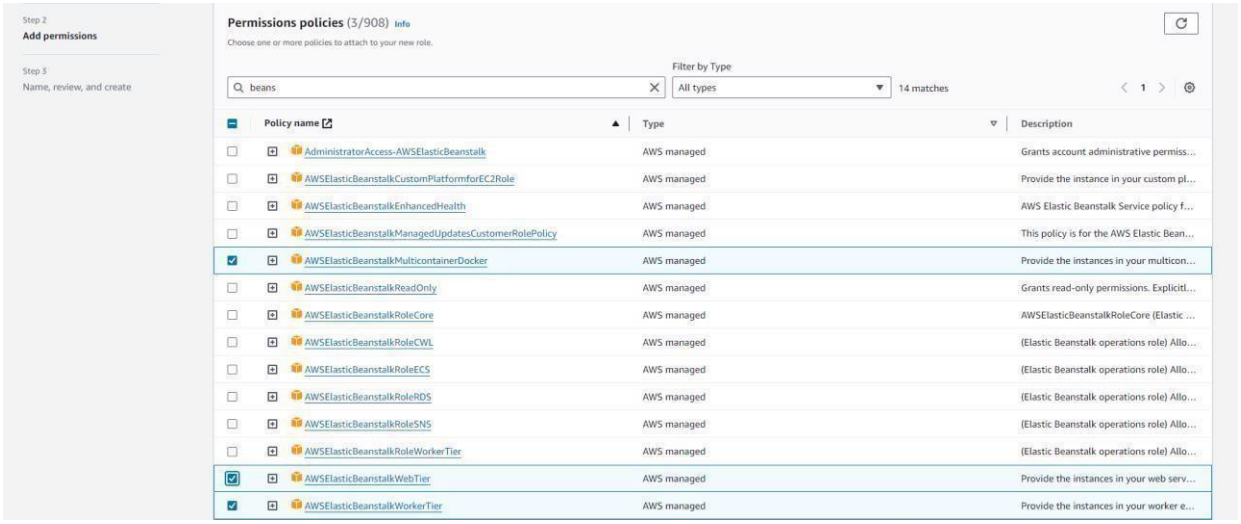
Service or use case

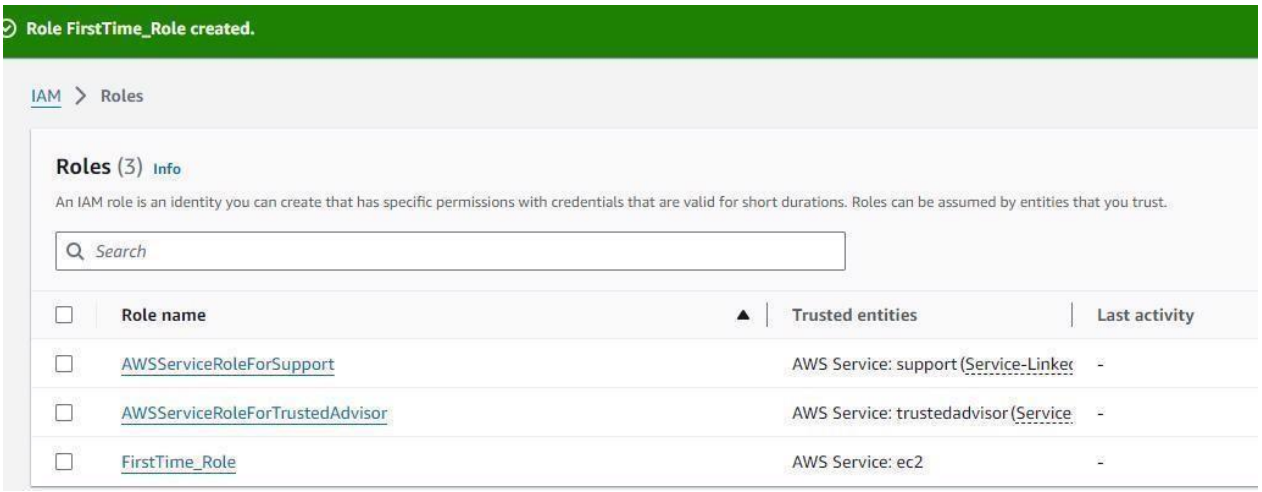
EC2

Choose a use case for the specified service.

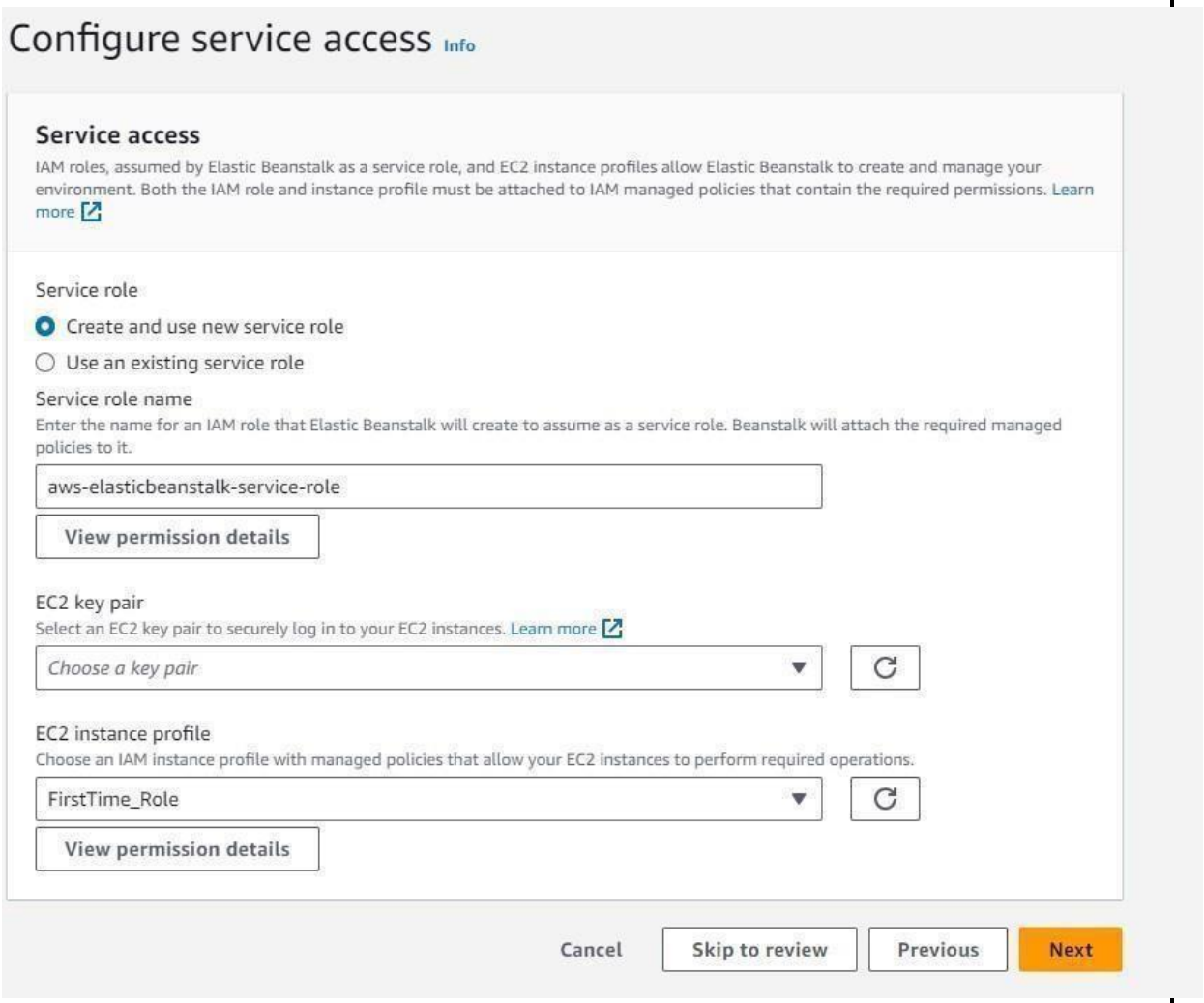
Use case

- (●) **EC2**
  Allows EC2 instances to call AWS services on your behalf.
- ( ) **EC2 Role for AWS Systems Manager**
  Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
- ( ) **EC2 Spot Fleet Role**
  Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
- ( ) **EC2 - Spot Fleet Auto Scaling**
  Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

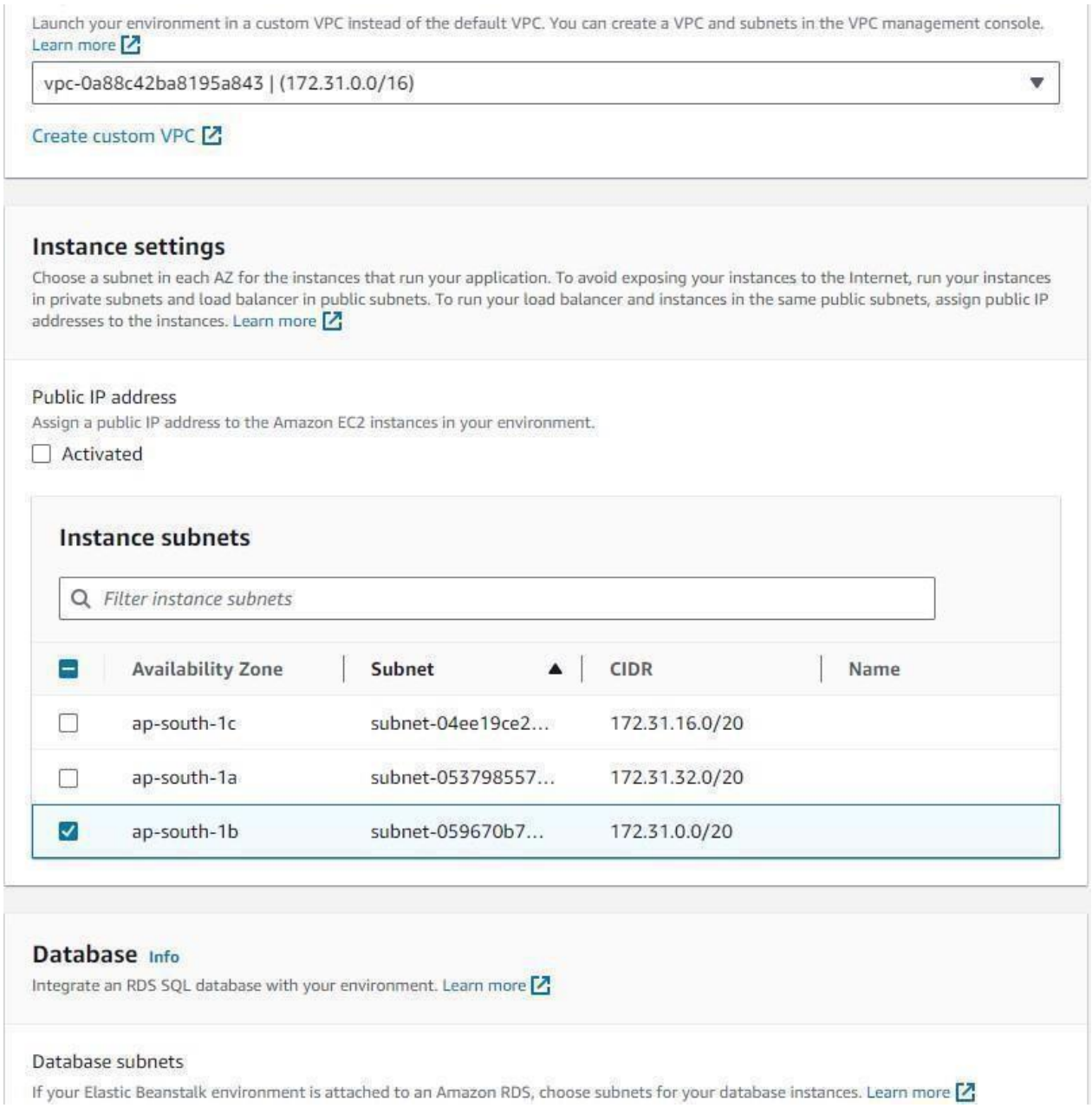




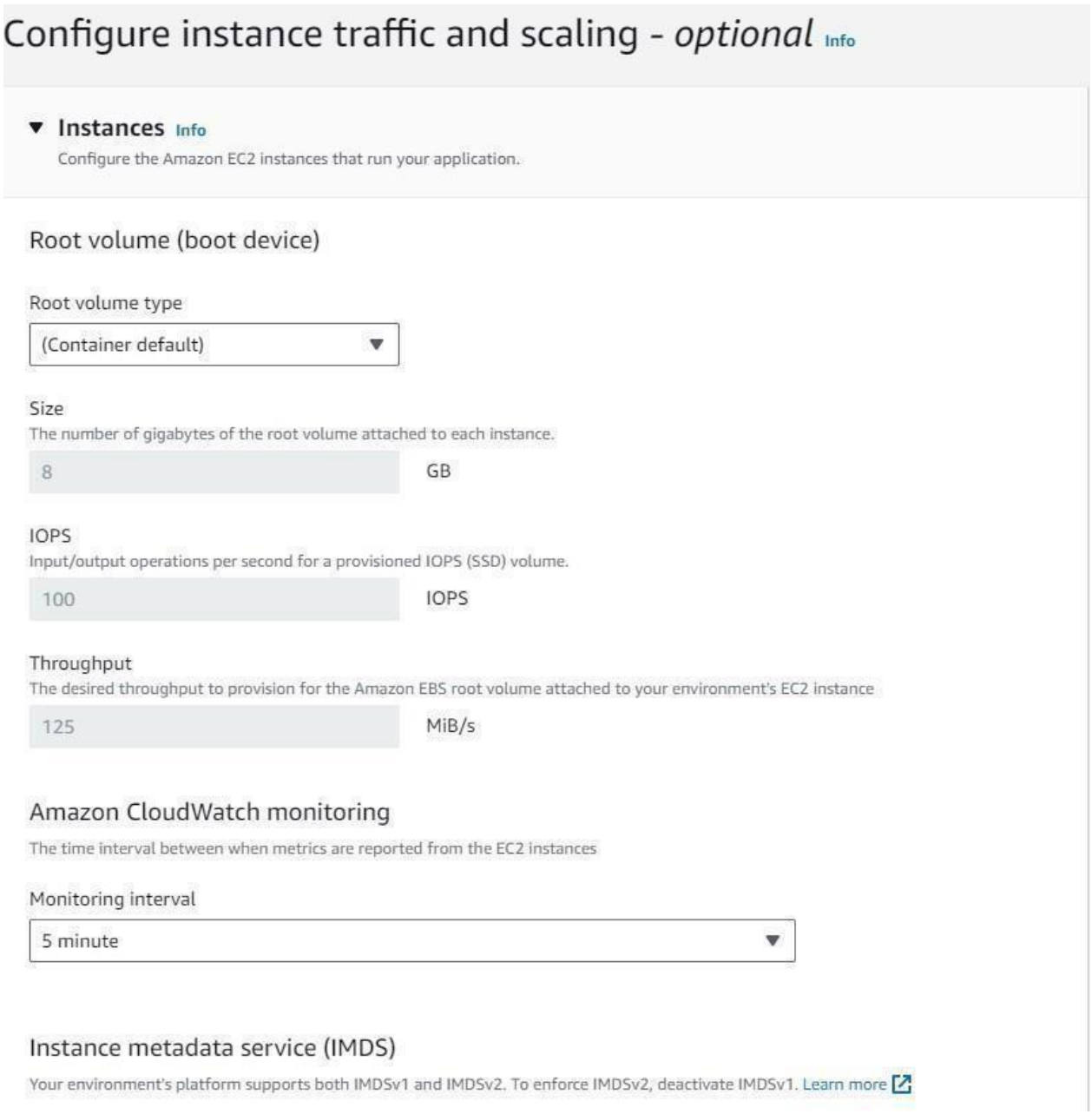
6. Create new service role, and select the EC instance profile



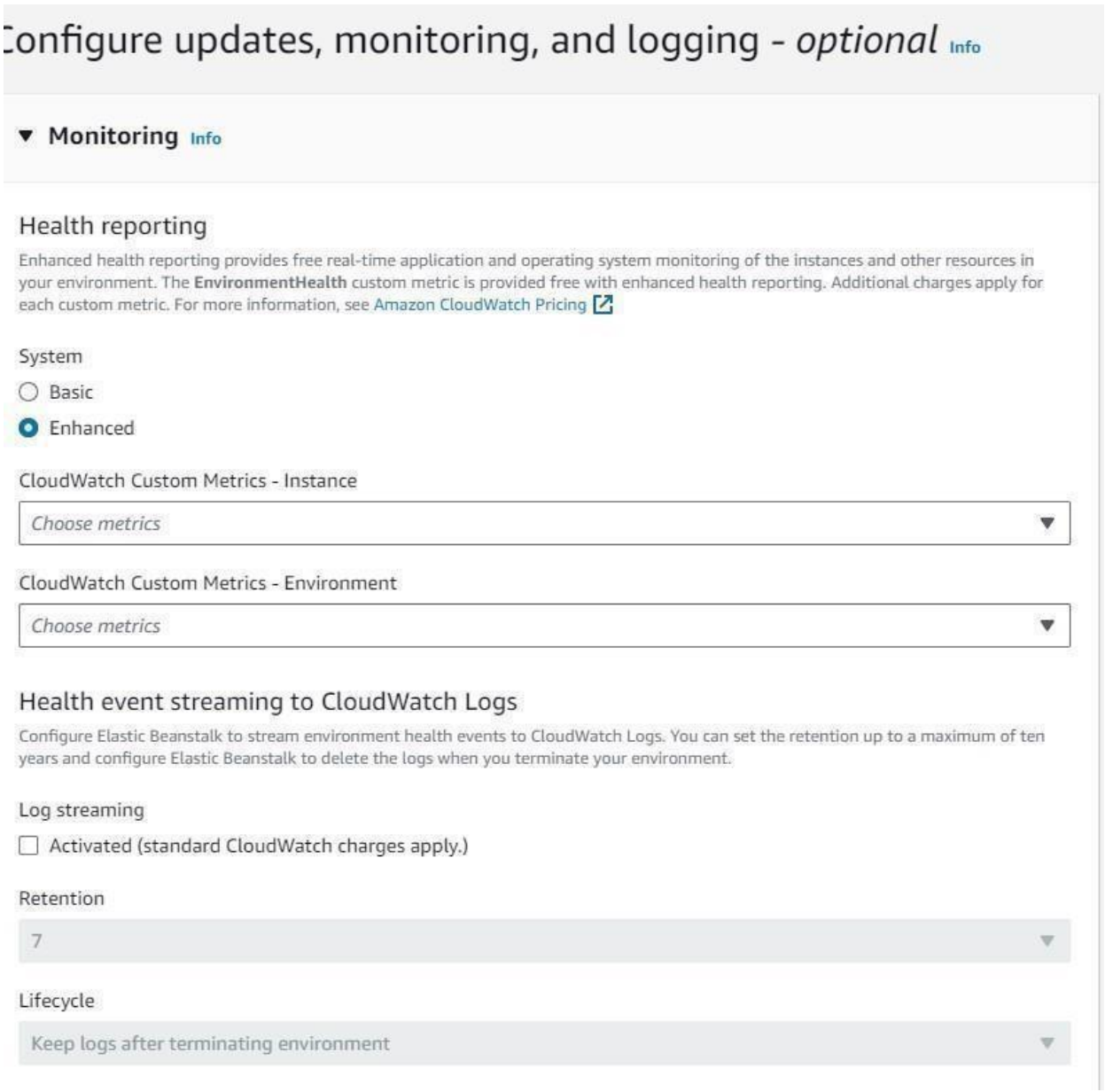
7. Set up network and database



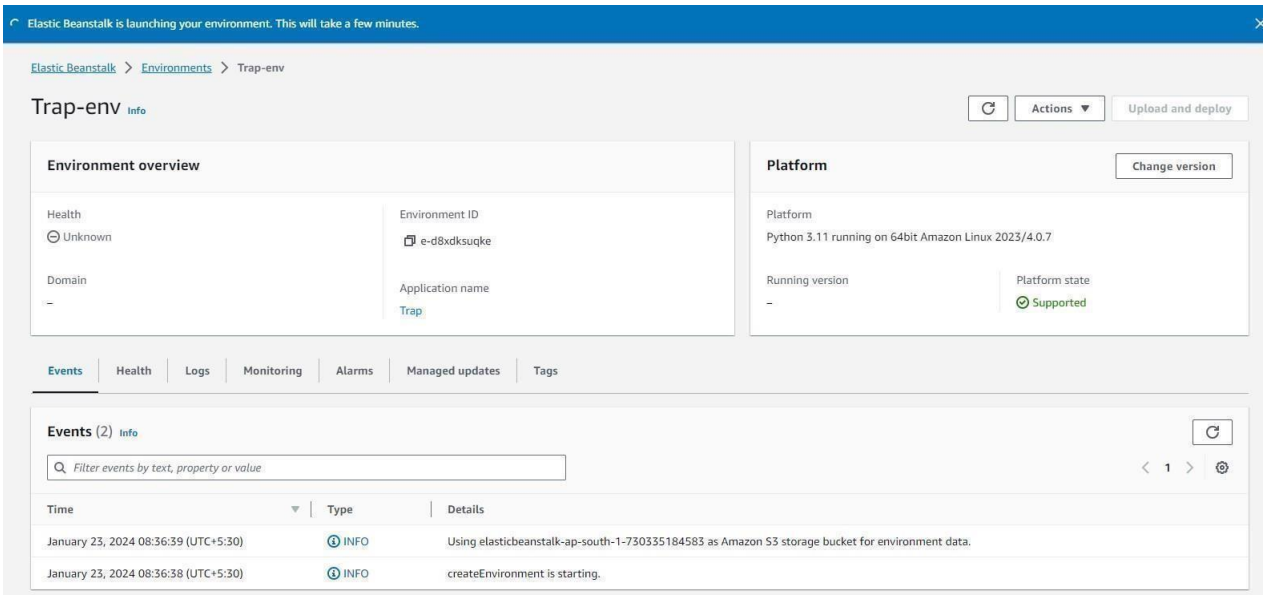
8. Keep the settings default

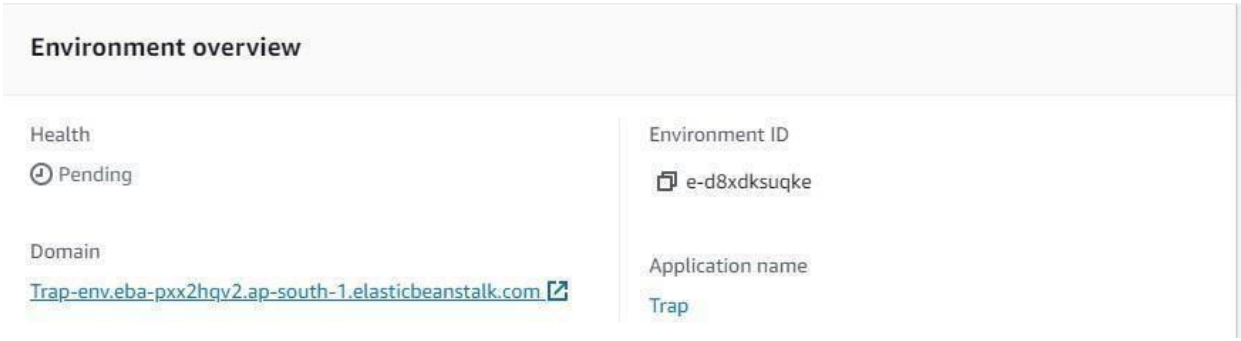


9. Keep the settings default

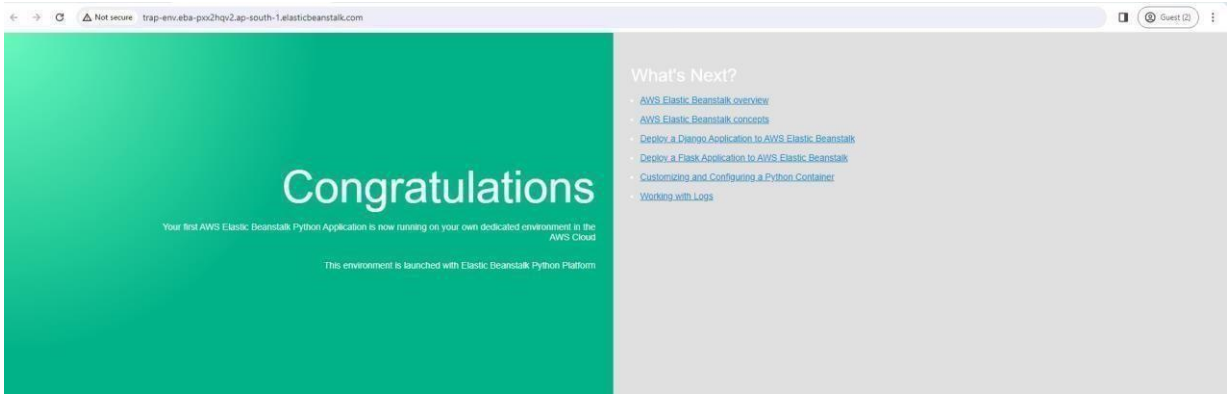


10. Review and submit!
11. The environment will start launching



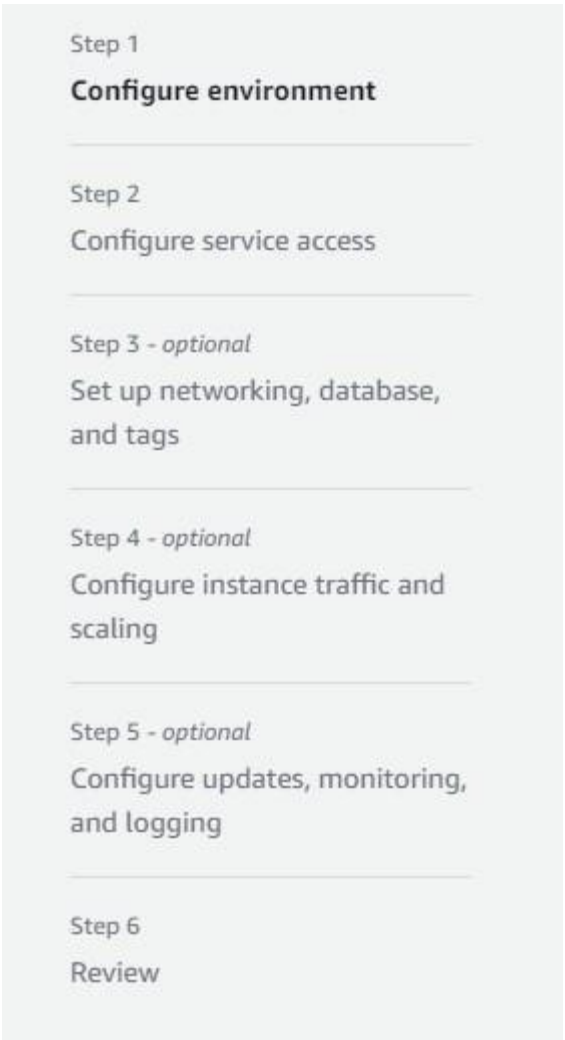


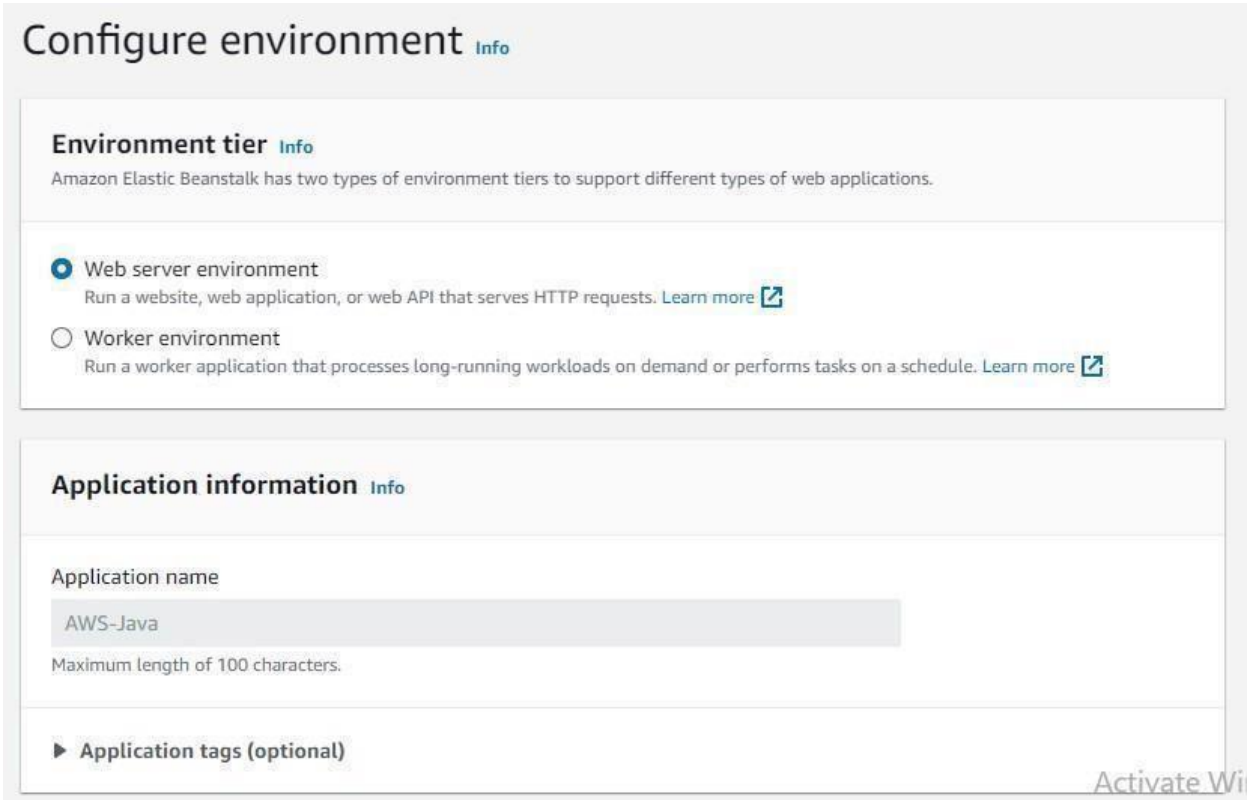
The webapp is launched!



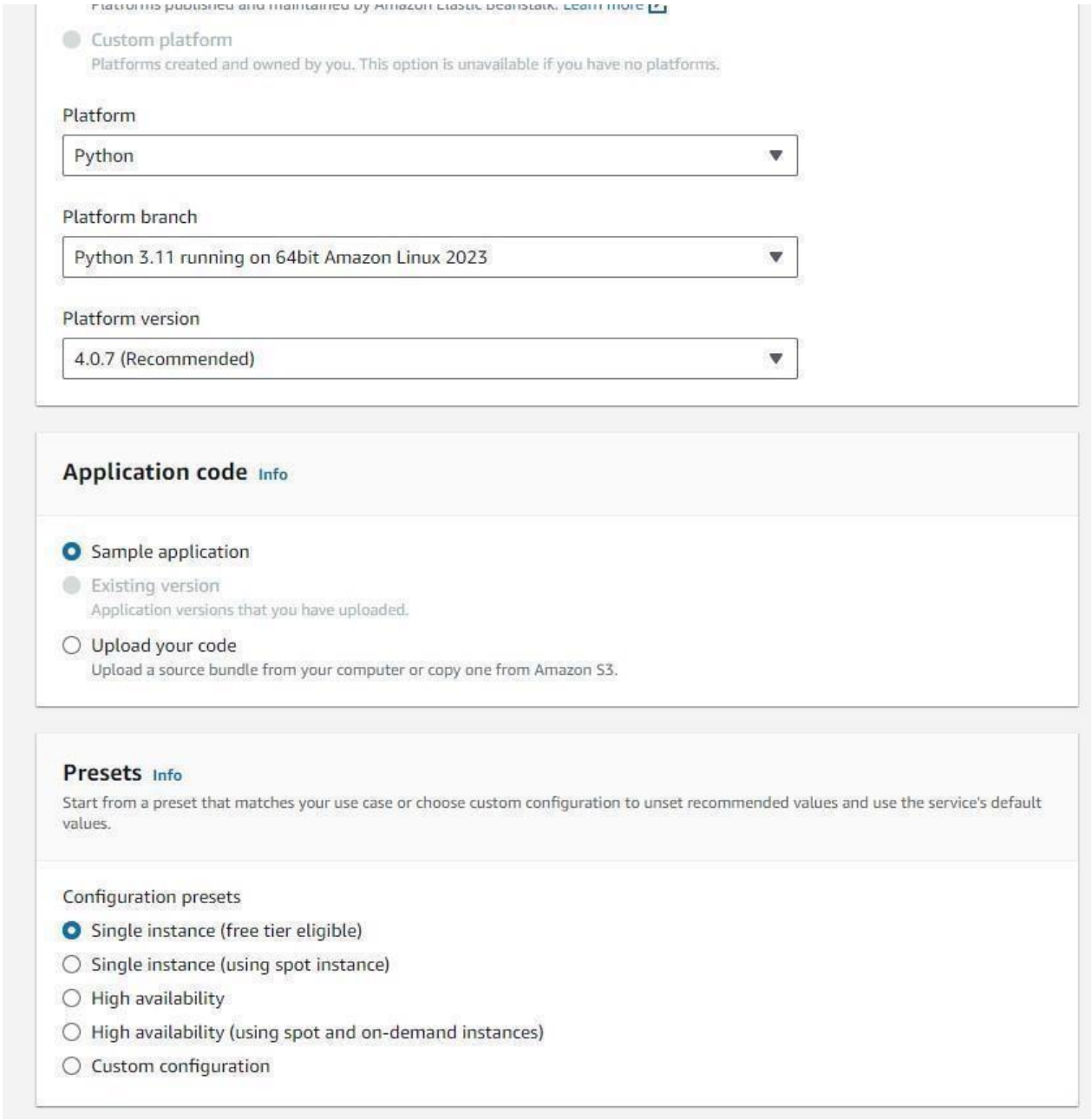
1. We'll be using Elastic Beanstalk - Java

2. There will be the steps

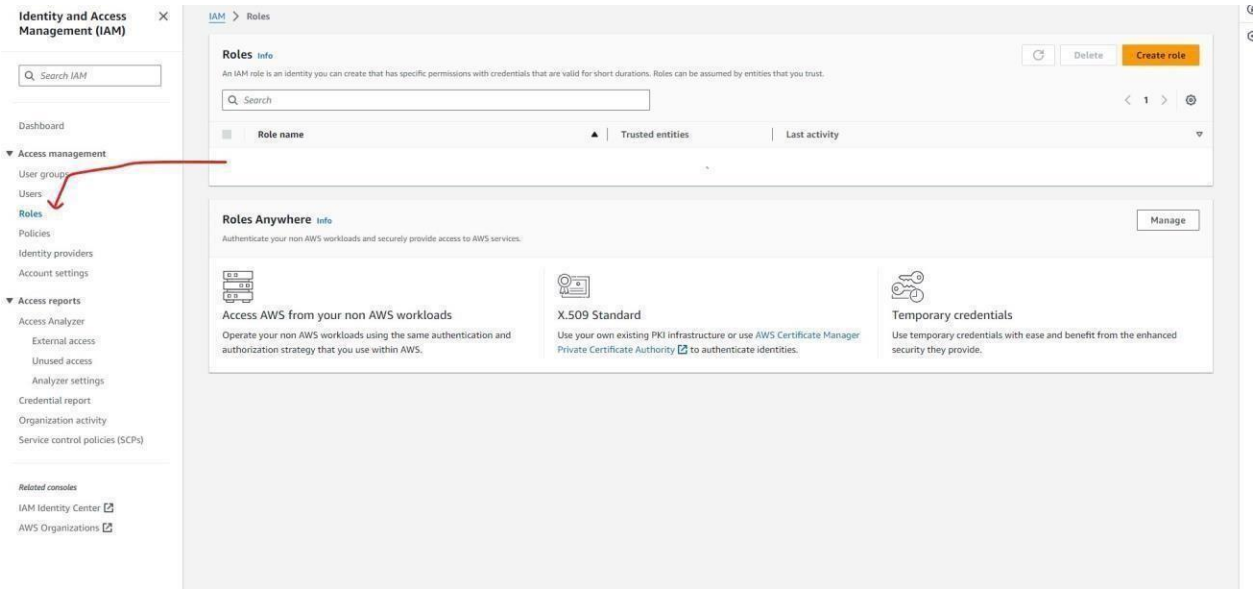




12. Select the platform



13. Go to the IAM services -> Create role



Trusted entity type

- (●) **AWS service**
  Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- ( ) **AWS account**
  Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- ( ) **Web identity**
  Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- ( ) **SAML 2.0 federation**
  Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- ( ) **Custom trust policy**
  Create a custom trust policy to enable others to perform actions in this account.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

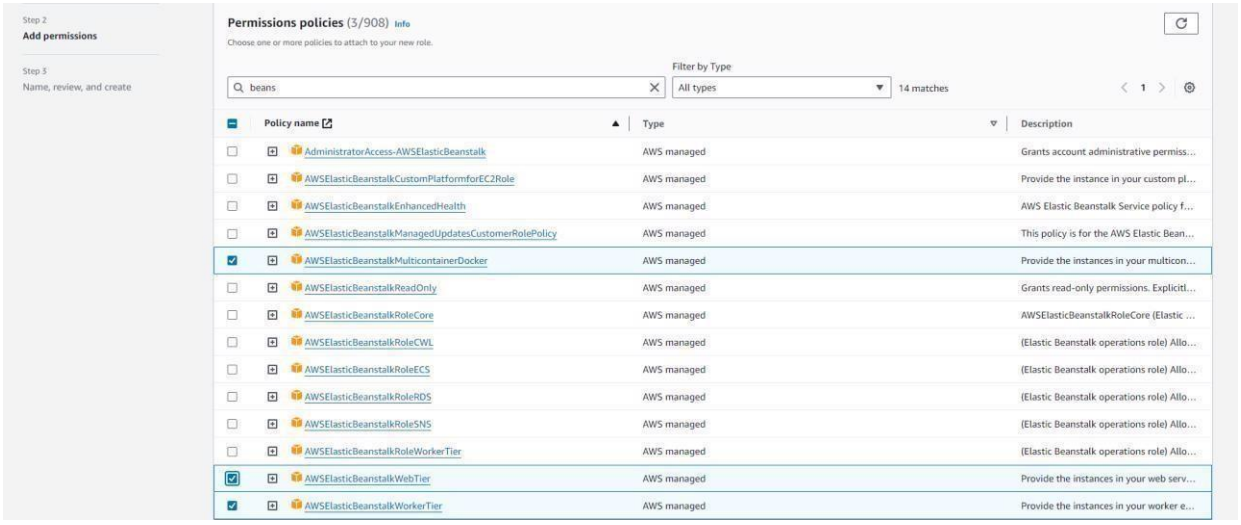
Service or use case

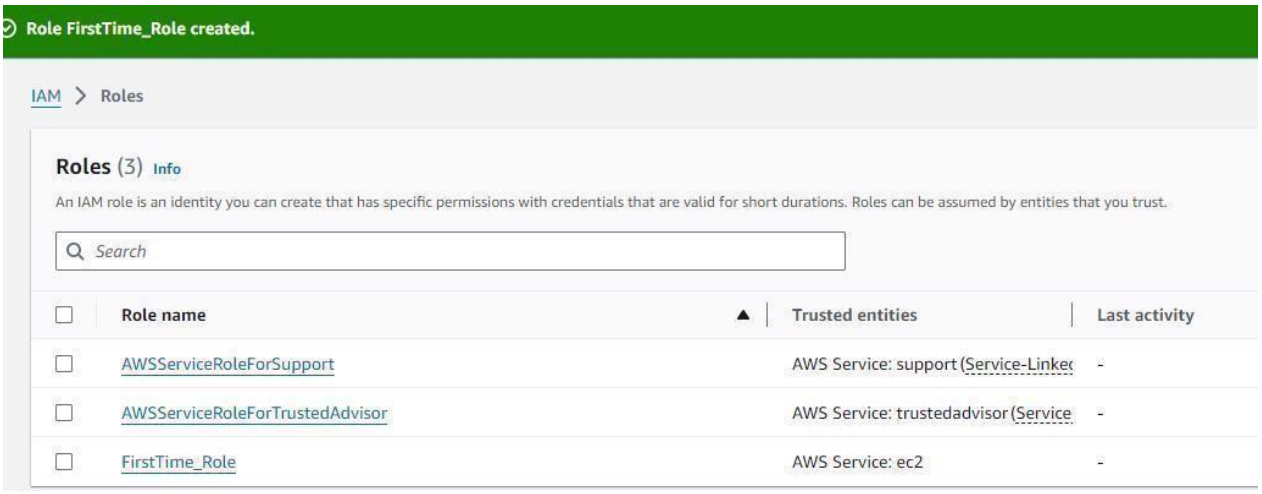
EC2

Choose a use case for the specified service.

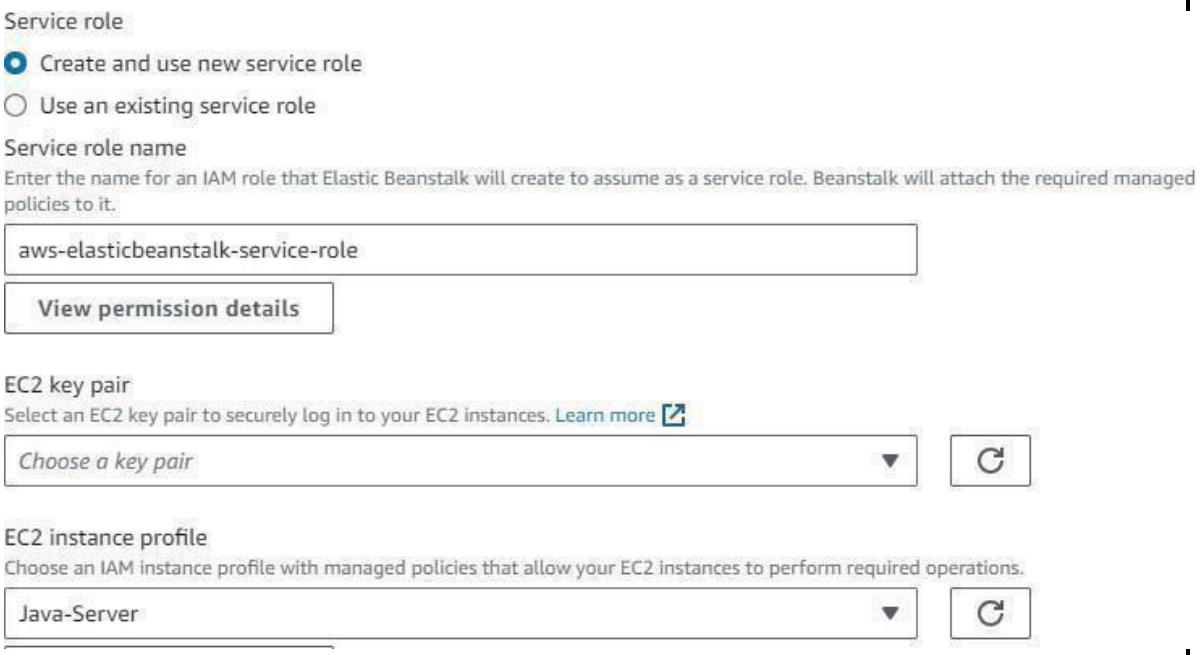
Use case

- (●) **EC2**
  Allows EC2 instances to call AWS services on your behalf.
- ( ) **EC2 Role for AWS Systems Manager**
  Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.
- ( ) **EC2 Spot Fleet Role**
  Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.
- ( ) **EC2 - Spot Fleet Auto Scaling**
  Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

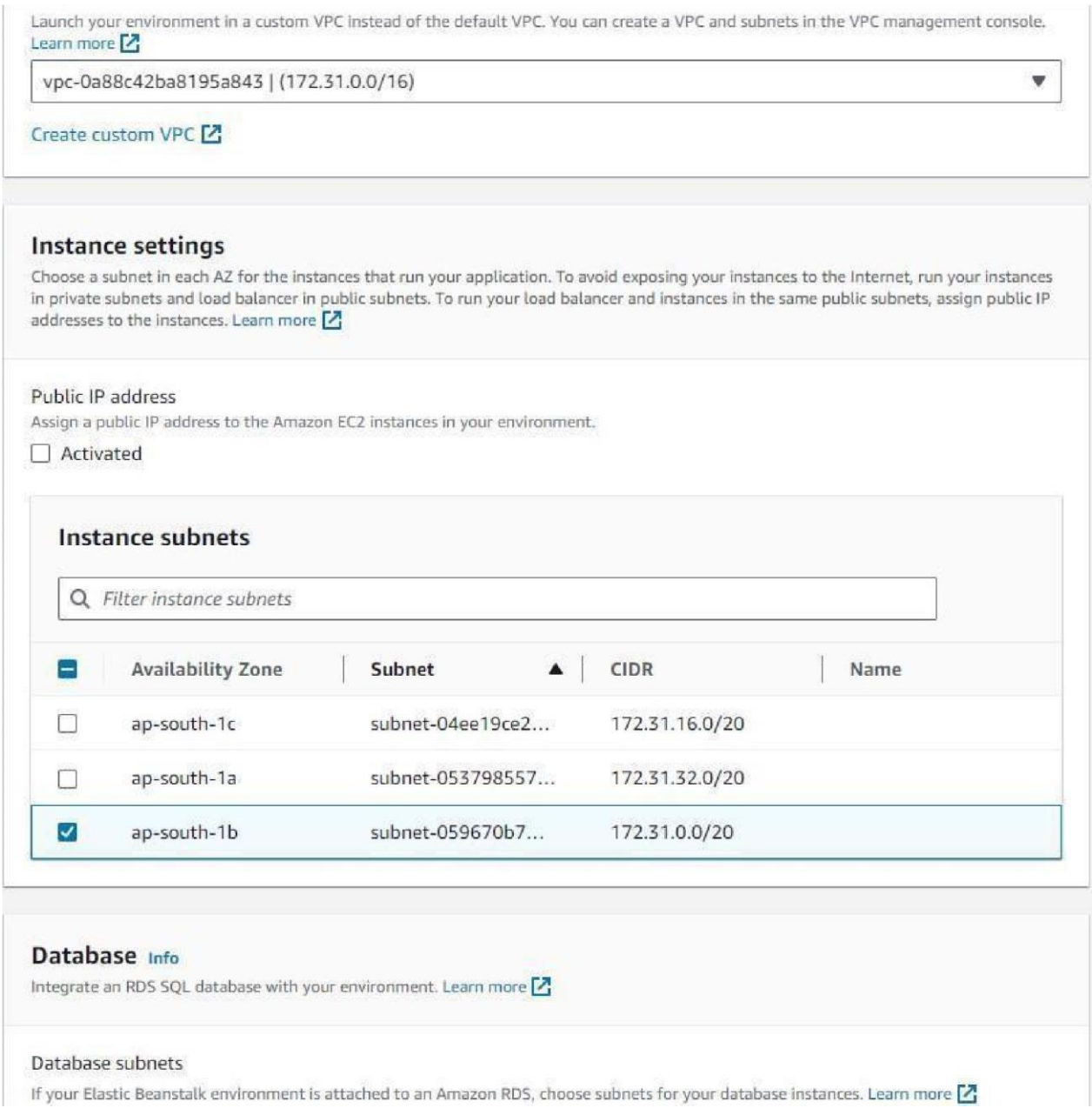




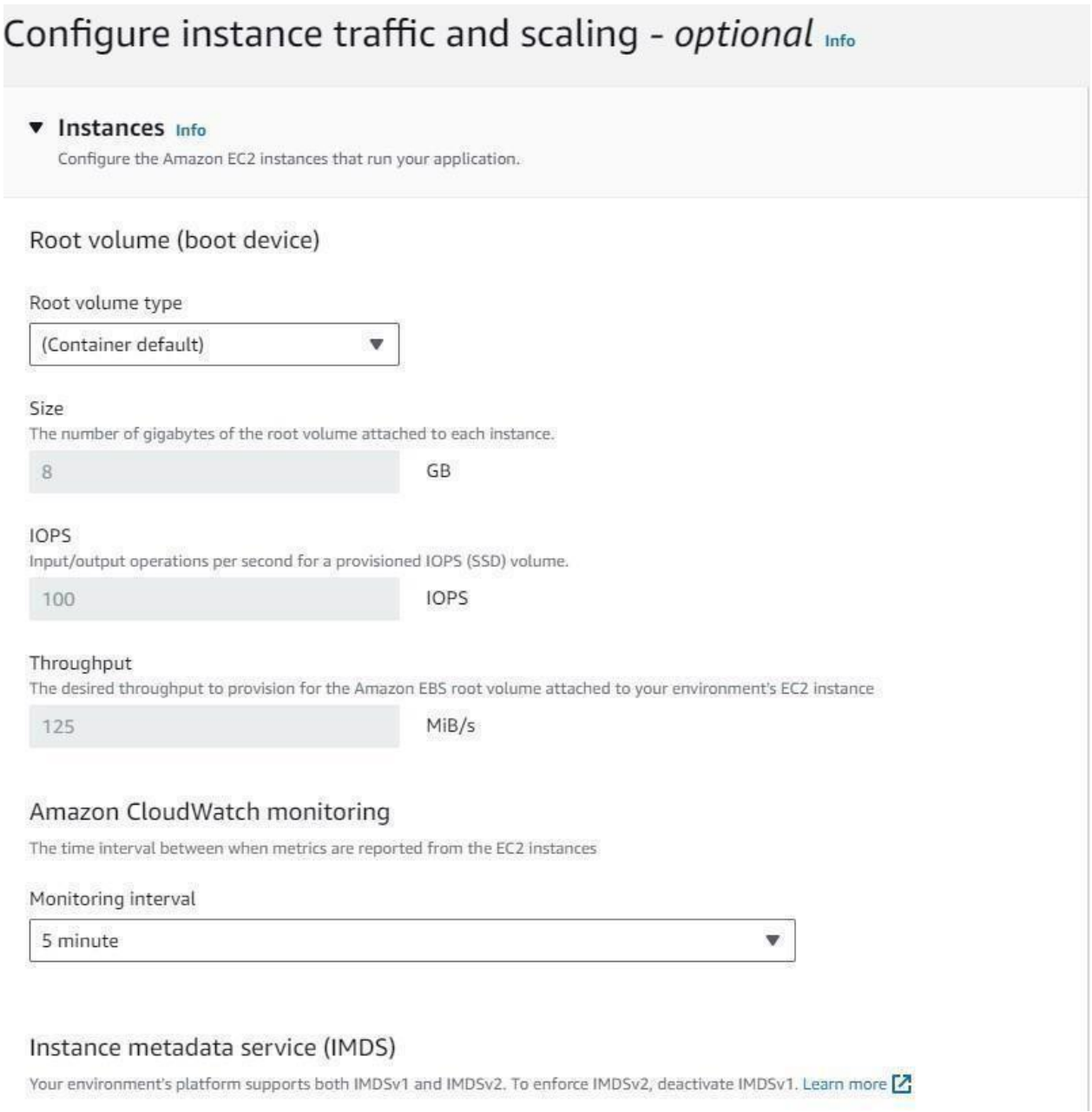
14. Create new service role, and select the EC instance profile



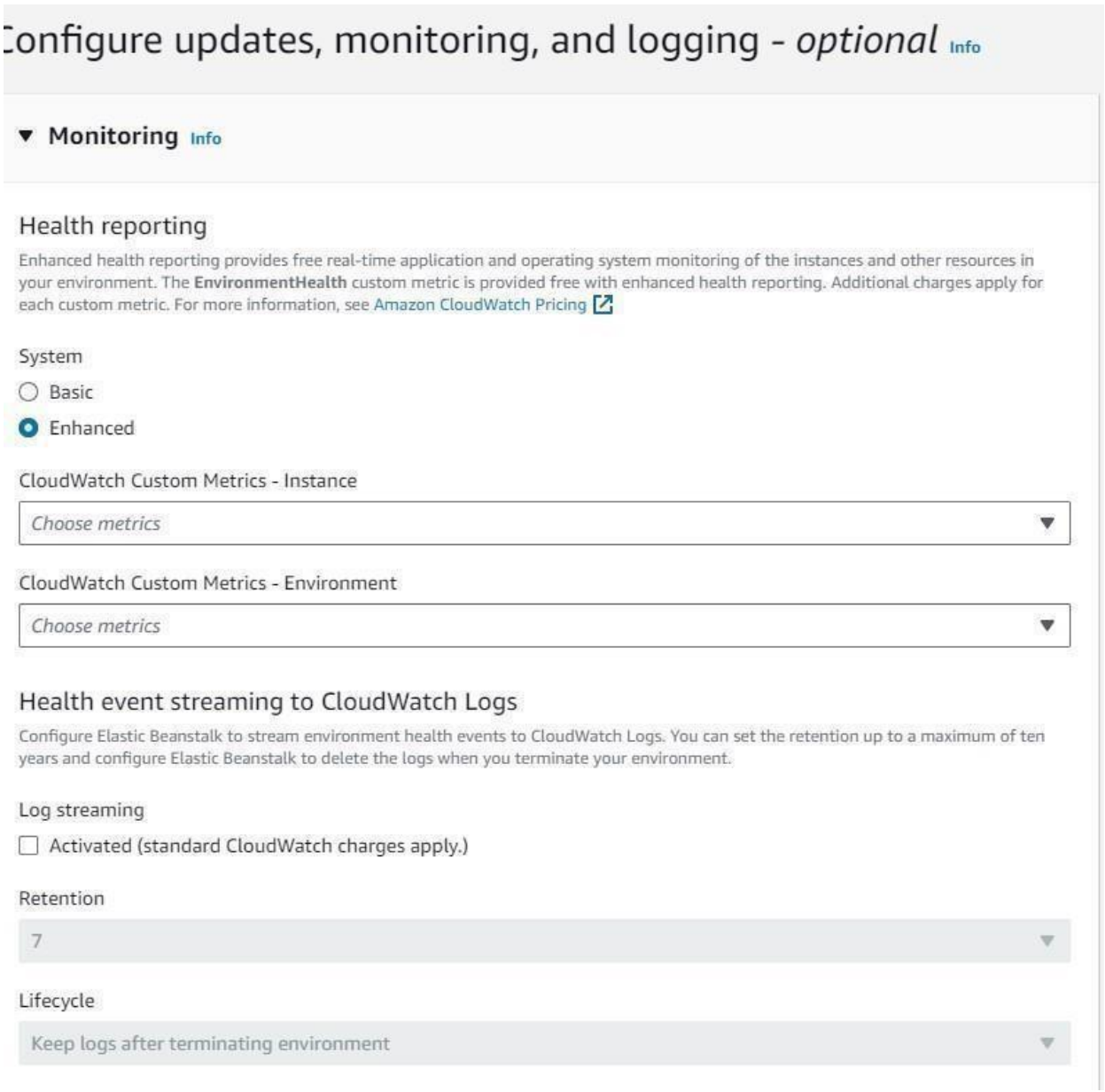
15. Set up network and database



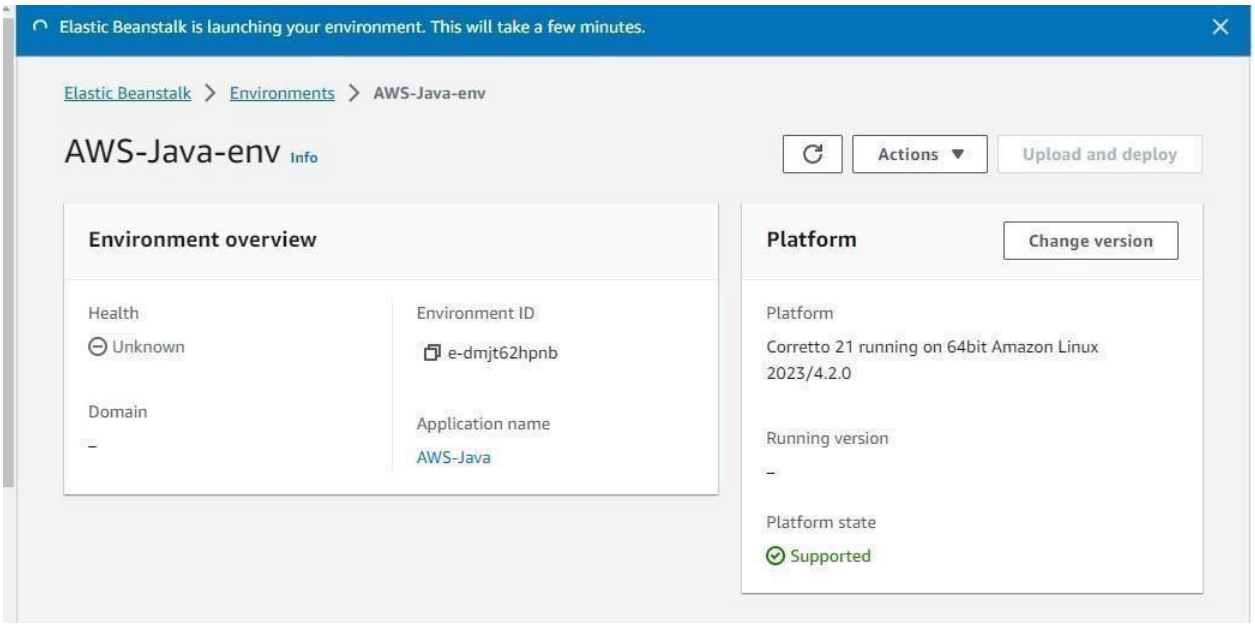
16. Keep the settings default

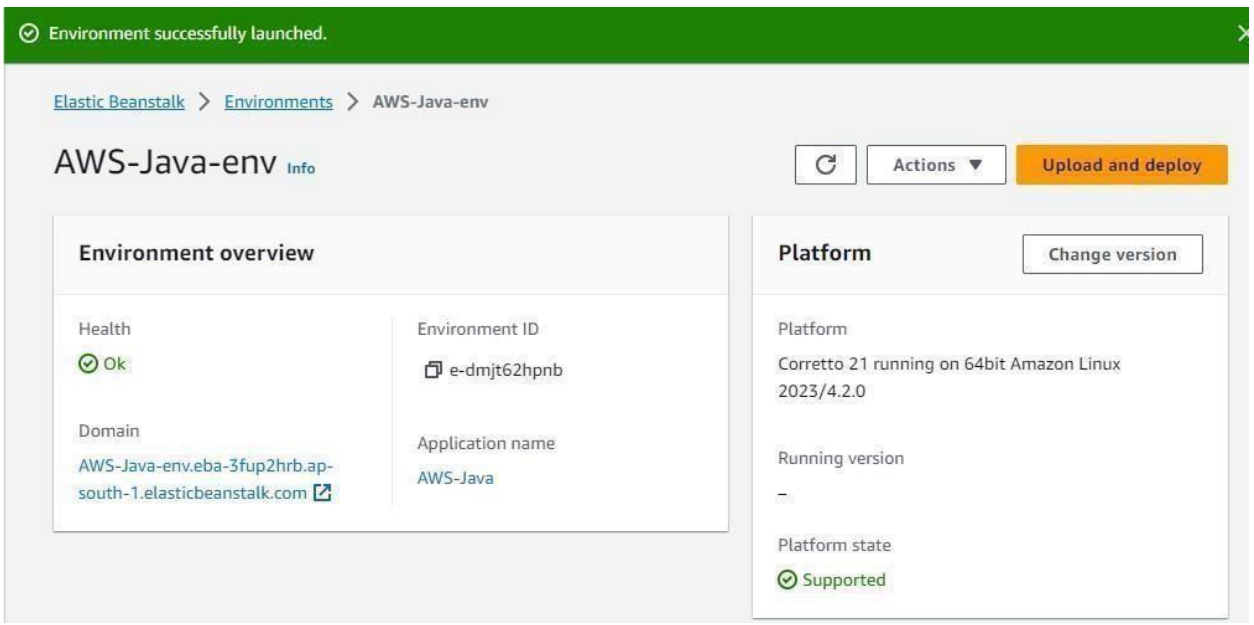


17. Keep the settings default

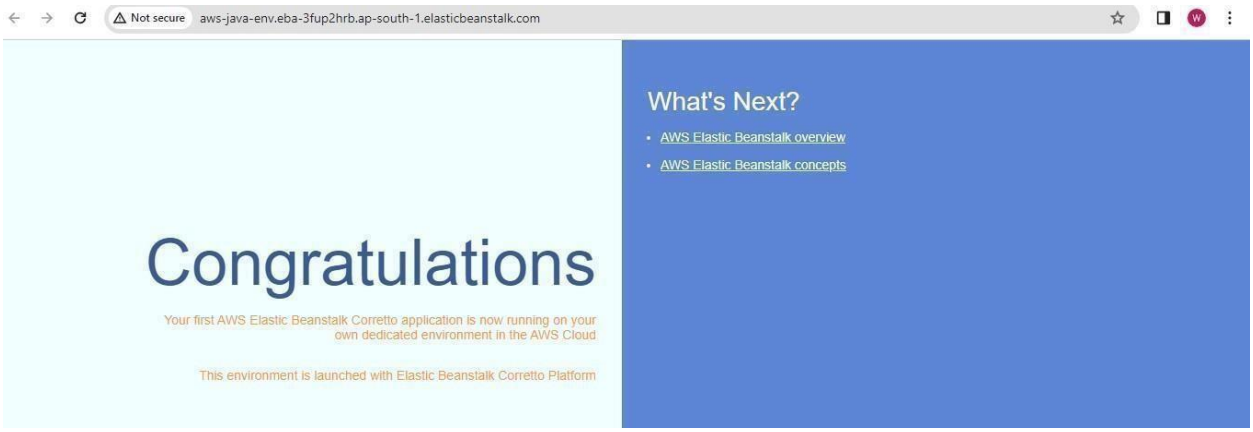


18. Review and submit!
19. The environment will start launching





The webapp is launched!


Not secure
aws-java-env.eba-3fup2hrb.ap-south-1.elasticbeanstalk.com
Congratulations
Your first AWS Elastic Beanstalk Corretto application is now running on your own dedicated environment in the AWS Cloud
This environment is launched with Elastic Beanstalk Corretto Platform
What's Next?
AWS Elastic Beanstalk overview
AWS Elastic Beanstalk concepts