VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering (SCOPE)

Fall Semester 2025-26

CBS3005: Cloud, Microservices and Applications

LAB ASSESSMENT 2

Name: Riya Autade
Registration No.: 22BBS0079

Question:

Create a simple web application using your preferred programming language and framework (e.g., Node.js, Python, Java etc.). Ensure the application is fully functional and ready for deployment. Initialize an AWS Elastic Beanstalk environment for your application. Choose the appropriate platform (e.g., Node.js, Python, Java) and configure the environment settings. Package your web application and deploy it to the Elastic Beanstalk environment. Access the deployed web application via the Elastic Beanstalk URL provided. Test its functionality to confirm that the deployment was successful and that the application is accessible and performs as expected.

The Web Application For Deployment

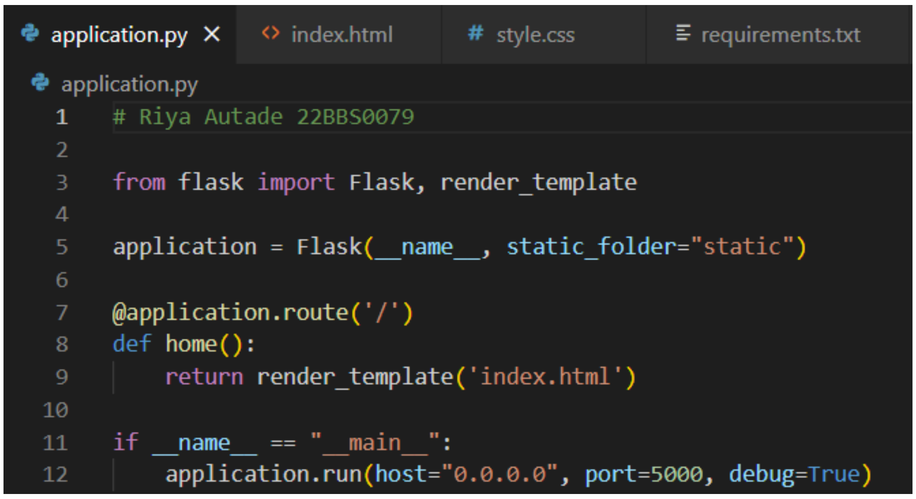
Programming Language: Python
Framework: Flask

Folder Structure:

```
myapp/
├─ application.py
├─ requirements.txt
├─ Procfile
├─ runtime.txt
├─ templates/
│   └─ index.html
├─ static/
  └─ style.css
```

Files & Codes:

1. application.py

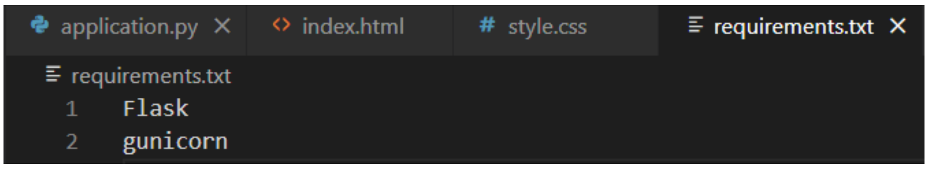


application.py × | index.html | style.css | requirements.txt

application.py

```python
# Riya Autade 22BBS0079

from flask import Flask, render_template

application = Flask(__name__, static_folder="static")

@application.route('/')
def home():
    return render_template('index.html')

if __name__ == "__main__":
    application.run(host="0.0.0.0", port=5000, debug=True)
```

2. requirements.text

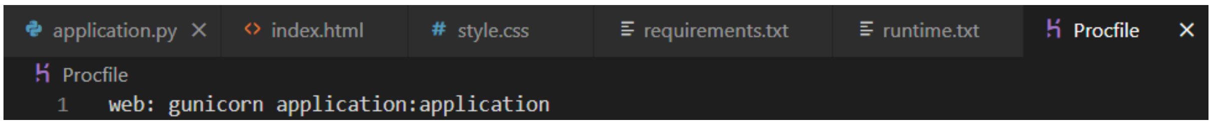


application.py × | index.html | style.css | requirements.txt ×

requirements.txt

```
Flask
gunicorn
```

3. Procfile

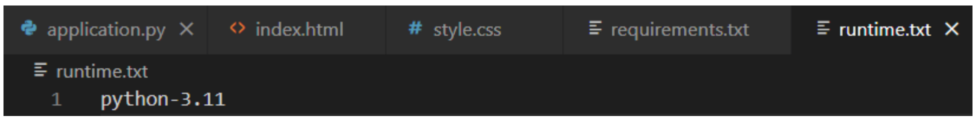


application.py × | index.html | style.css | requirements.txt | runtime.txt | Procfile ×

Procfile

```
web: gunicorn application:application
```

4. runtime.txt

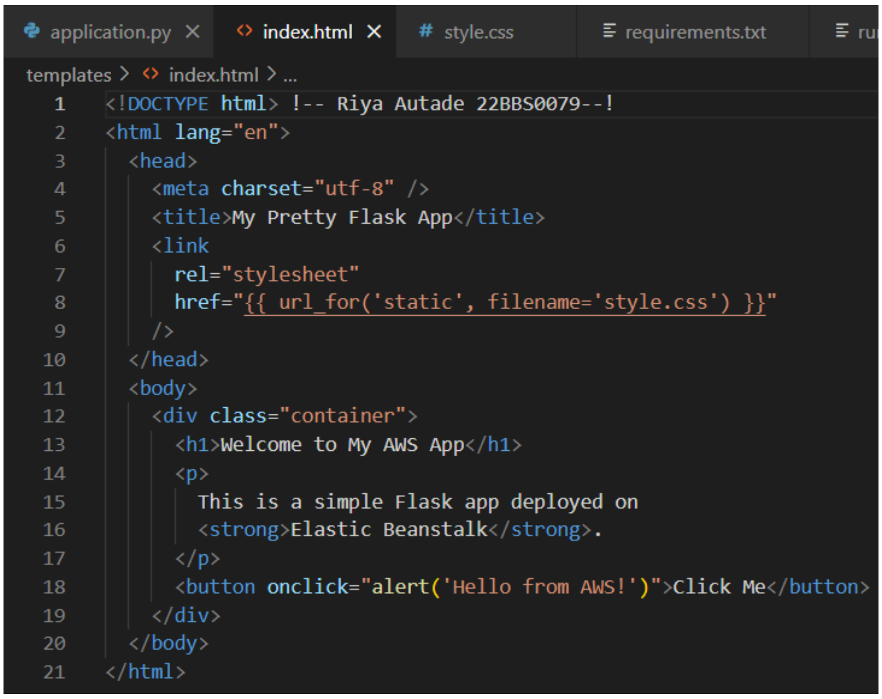


application.py × | index.html | style.css | requirements.txt | runtime.txt ×

runtime.txt

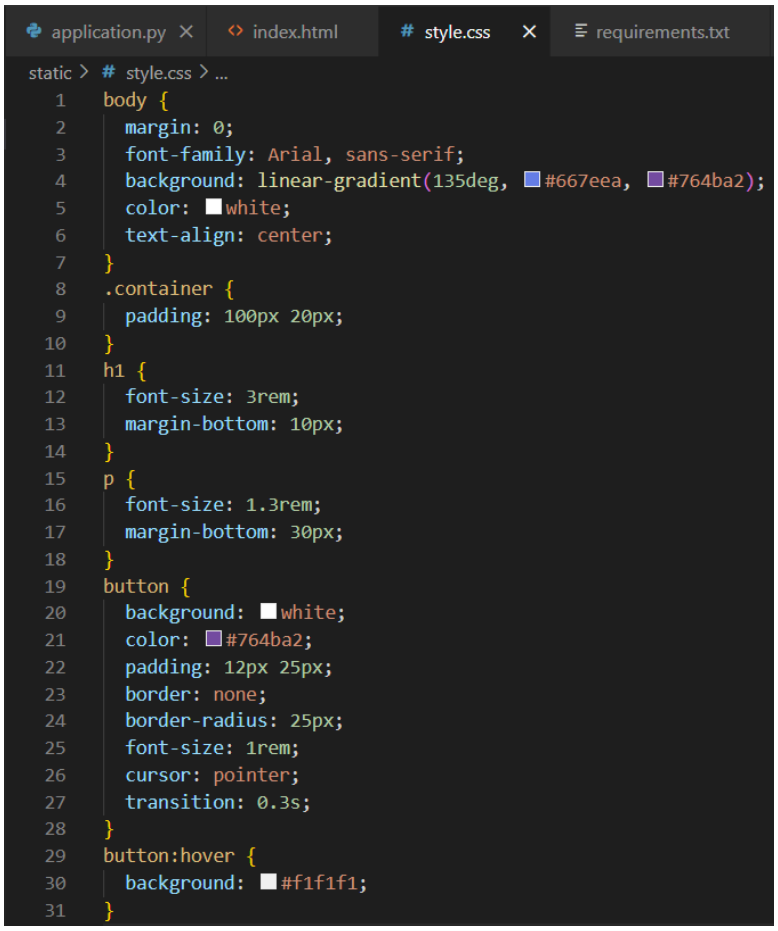
```
python-3.11
```

5. templates> index.html



```html
<!DOCTYPE html> !-- Riya Autade 22BBS0079--!
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>My Pretty Flask App</title>
    <link
      rel="stylesheet"
      href="{{ url_for('static', filename='style.css') }}"
    />
  </head>
  <body>
    <div class="container">
      <h1>Welcome to My AWS App</h1>
      <p>
        This is a simple Flask app deployed on
        <strong>Elastic Beanstalk</strong>.
      </p>
      <button onclick="alert('Hello from AWS!')">Click Me</button>
    </div>
  </body>
</html>
```

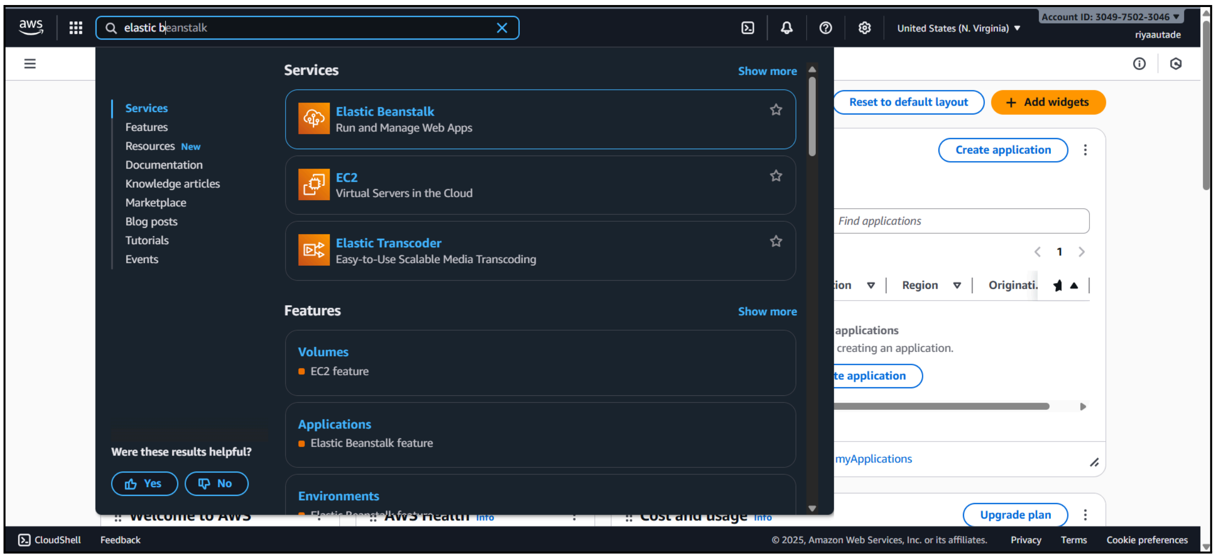
6. static> style.css



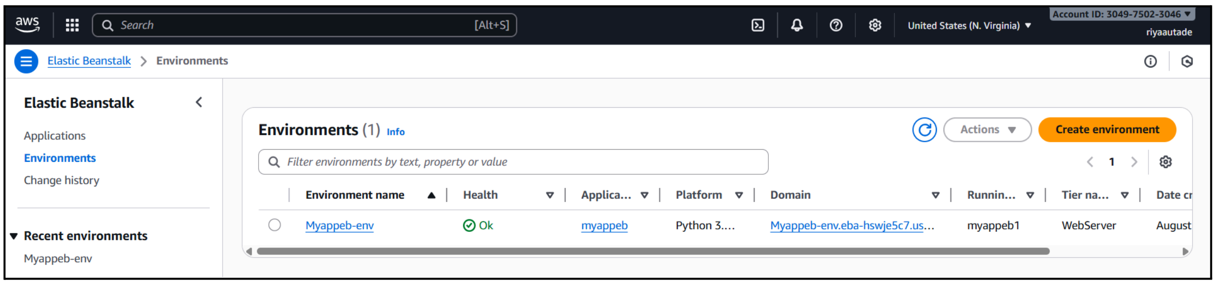
```css
body {
  margin: 0;
  font-family: Arial, sans-serif;
  background: linear-gradient(135deg, #667eea, #764ba2);
  color: white;
  text-align: center;
}
.container {
  padding: 100px 20px;
}
h1 {
  font-size: 3rem;
  margin-bottom: 10px;
}
p {
  font-size: 1.3rem;
  margin-bottom: 30px;
}
button {
  background: white;
  color: #764ba2;
  padding: 12px 25px;
  border: none;
  border-radius: 25px;
  font-size: 1rem;
  cursor: pointer;
  transition: 0.3s;
}
button:hover {
  background: #f1f1f1;
}
```

AWS Elastic Beanstalk Steps to Deploy Web Application

1. Search Elastic Beanstalk in AWS Management Console.

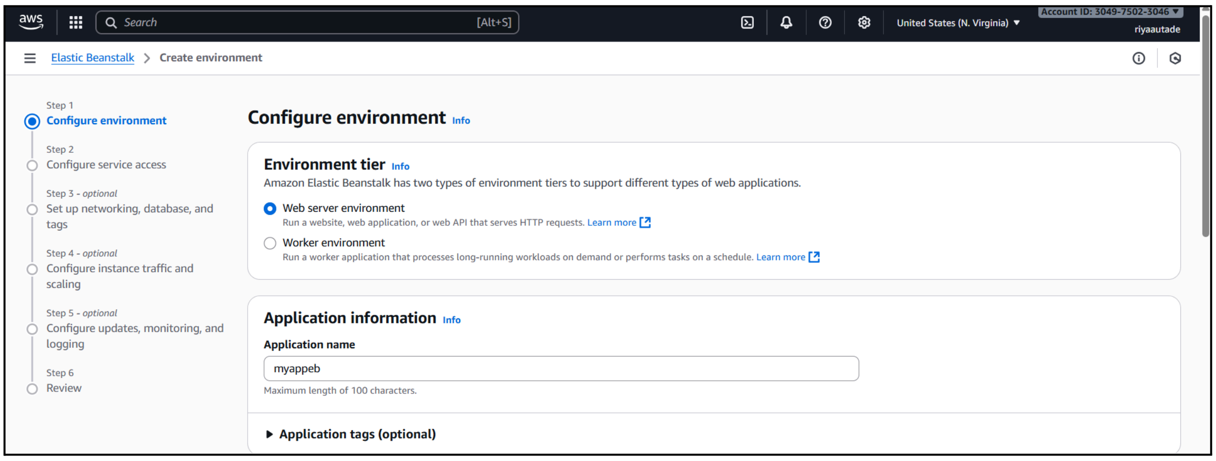


2. Elastic Beanstalk Dashboard opens. Click on 'Create Environment'.



Environment name	Health	Applica...	Platform	Domain	Runnin...	Tier na...	Date cr
Myappeb-env	Ok	myappeb	Python 3....	Myappeb-env.eba-hswje5c7.us...	myappeb1	WebServer	August

3. **Step 1: Configure Environment**: Click on 'Web server environment' & name the application. [See the steps in the panel at the left]


Search [Alt+S]
United States (N. Virginia)
Account ID: 3049-7502-3046
riyaautade
Elastic Beanstalk > Create environment
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

Configure environment Info

Environment tier Info

Amazon Elastic Beanstalk has two types of environment tiers to support different types of web applications.

[✓ Web server environment]
Run a website, web application, or web API that serves HTTP requests. Learn more

Worker environment
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. Learn more

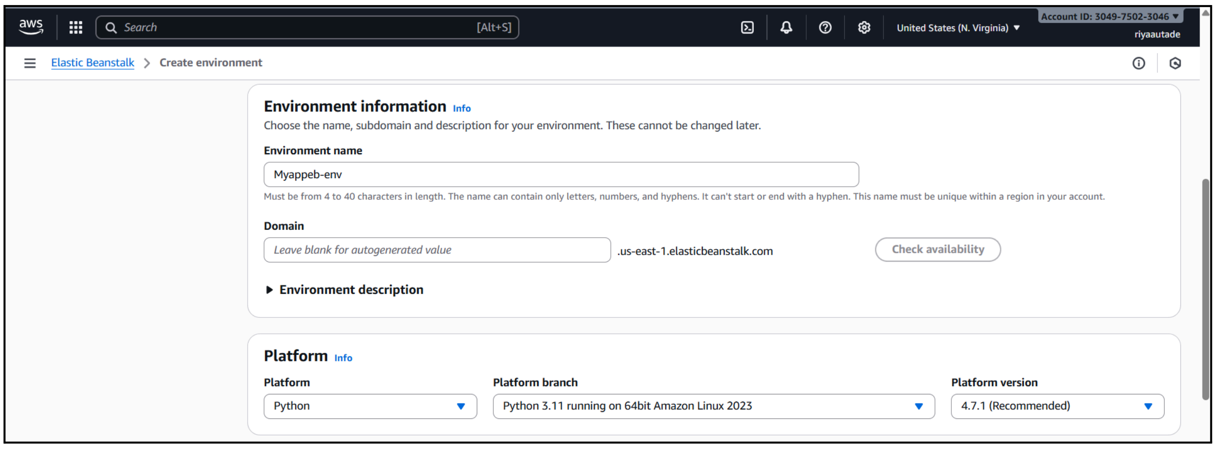
Application information Info

Application name: myappeb

Maximum length of 100 characters.

Application tags (optional)

4. Name your Environment and choose Python for Platform. The Platform branch must be the same as the version in the runtime.txt file.


aws
Search [Alt+S]
United States (N. Virginia)
Account ID: 3049-7502-3046
riyaautade
Elastic Beanstalk > Create environment

Environment information Info

Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name: Myappeb-env

Must be from 4 to 40 characters in length. The name can contain only letters, numbers, and hyphens. It can't start or end with a hyphen. This name must be unique within a region in your account.

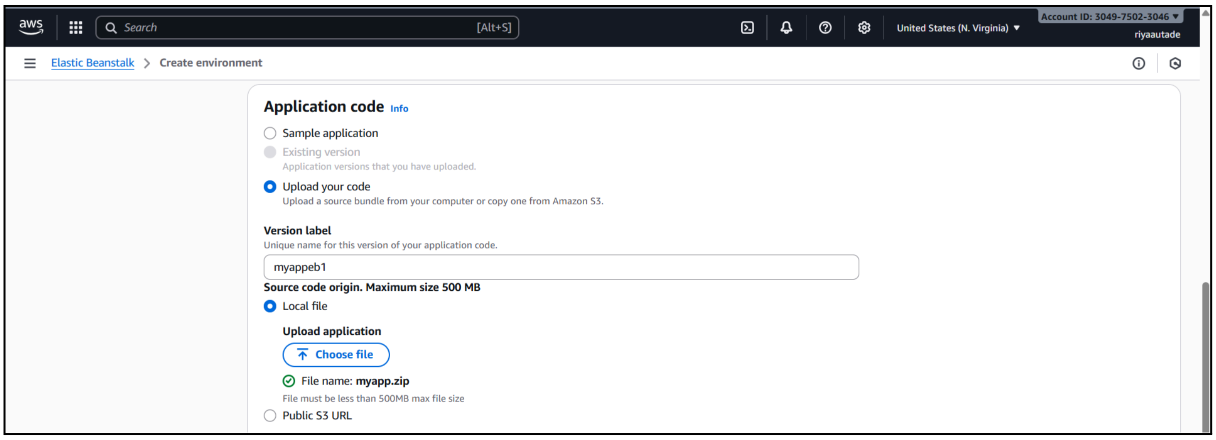
Domain: *Leave blank for autogenerated value* .us-east-1.elasticbeanstalk.com — Check availability

▸ Environment description

Platform Info

Platform	Platform branch	Platform version
Python	Python 3.11 running on 64bit Amazon Linux 2023	4.7.1 (Recommended)

5. Click on 'Upload your code' and 'Local file' to upload your application files as a zip folder (myapp.zip). Give a name to this code version under Version Label.


aws
Search [Alt+S]
United States (N. Virginia)
Account ID: 3049-7502-3046
riyaautade
Elastic Beanstalk > Create environment

Application code Info

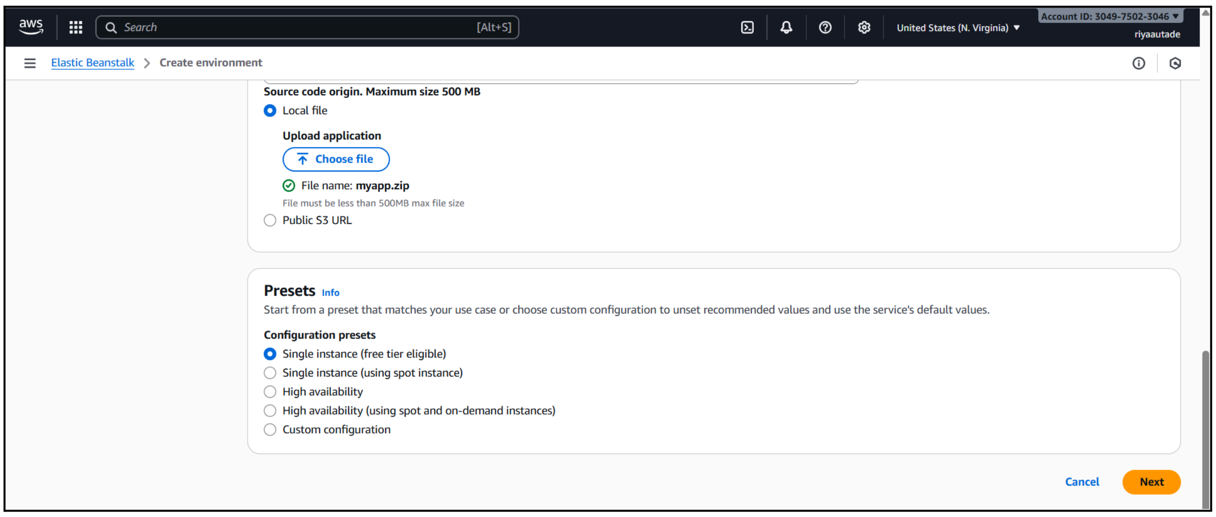
- Sample application
- Existing version — Application versions that you have uploaded.
- [✓ Upload your code] — Upload a source bundle from your computer or copy one from Amazon S3.

Version label — Unique name for this version of your application code.: myappeb1

Source code origin. Maximum size 500 MB

- [✓ Local file]
  - Upload application: Choose file
  - File name: **myapp.zip**
  - File must be less than 500MB max file size
- Public S3 URL

6. Leave the Presets as default and click on 'Next'.


aws
Search [Alt+S]
United States (N. Virginia)
Account ID: 3049-7502-3046
riyaautade
Elastic Beanstalk > Create environment

Source code origin. Maximum size 500 MB

- [✓ Local file]
  - Upload application: Choose file
  - File name: **myapp.zip**
  - File must be less than 500MB max file size
- Public S3 URL

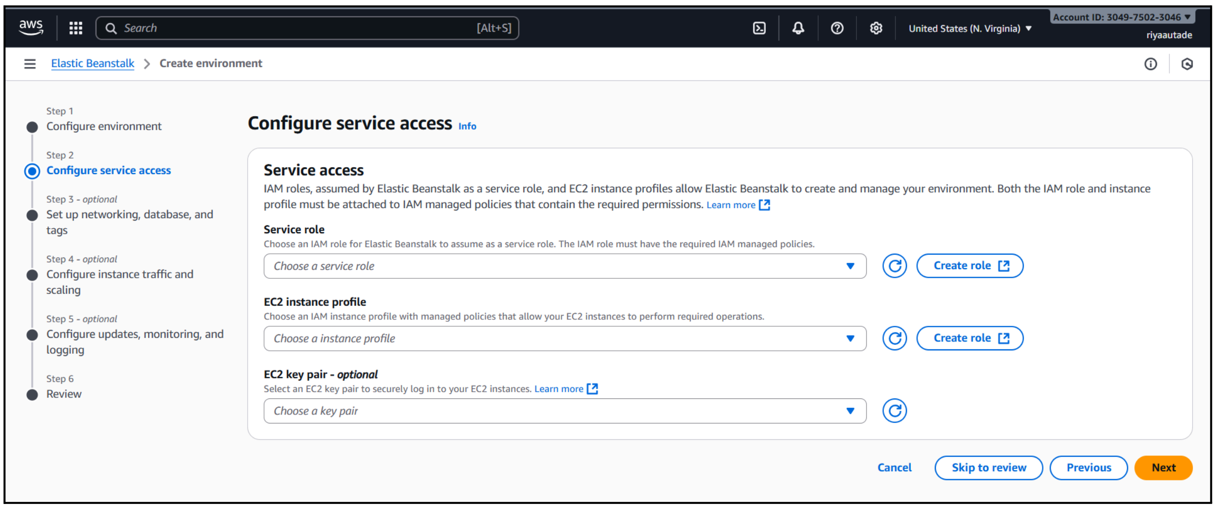
Presets Info

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

7. **Step 2: Configuration service access**: Click on 'Create role' for Service role.


Elastic Beanstalk > Create environment

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

Configure service access Info

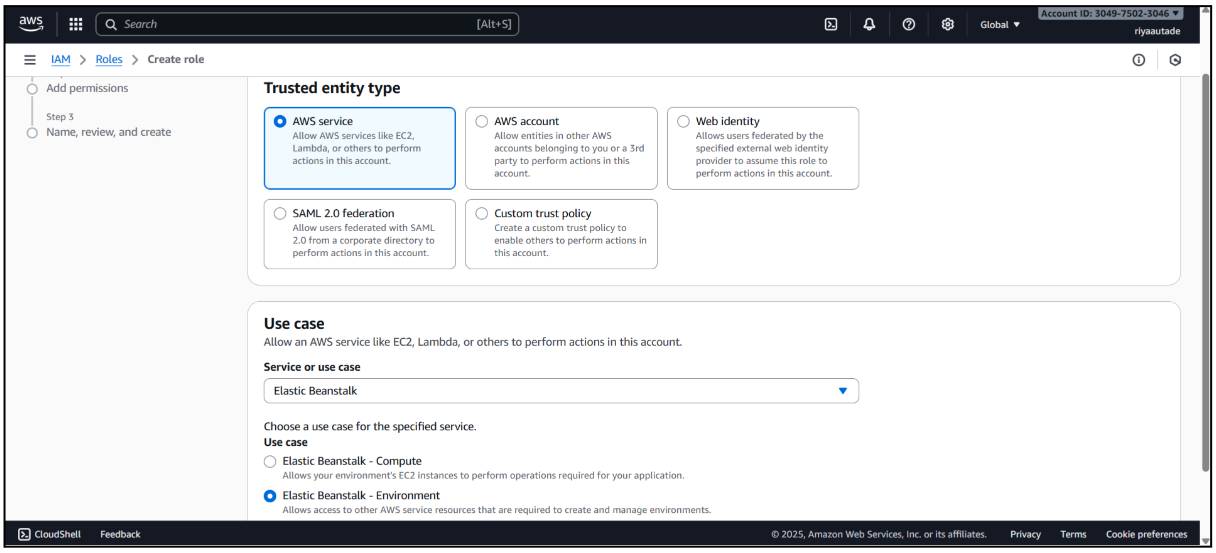
Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. Learn more

Service role
Choose an IAM role for Elastic Beanstalk to assume as a service role. The IAM role must have the required IAM managed policies.
Choose a service role — Create role

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.
Choose a instance profile — Create role

EC2 key pair - *optional*
Select an EC2 key pair to securely log in to your EC2 instances. Learn more
Choose a key pair

8. Let the default selections: 'AWS Service' and 'Elastic Beanstalk- Environment' be.


IAM > Roles > Create role

Add permissions
Step 3
Name, review, and create

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.

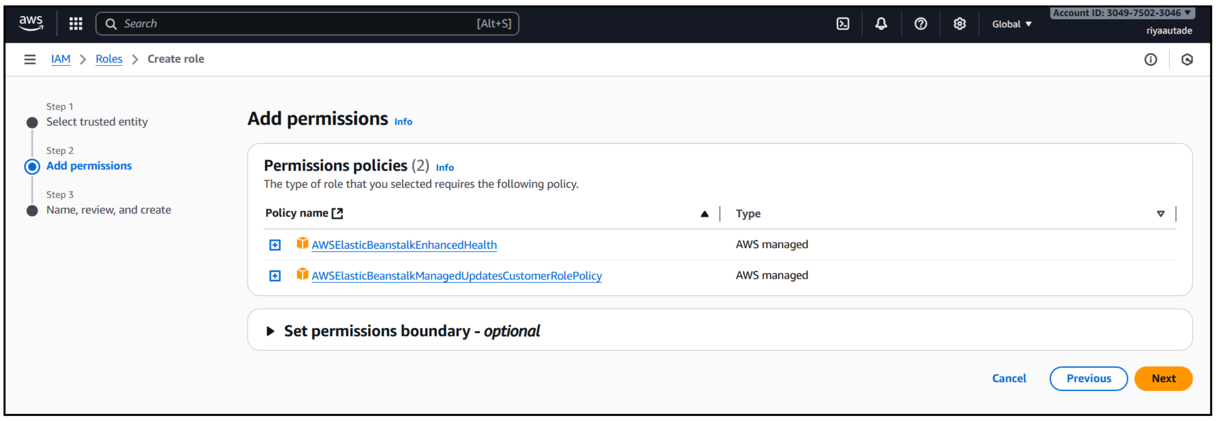
Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case
Elastic Beanstalk

Choose a use case for the specified service.
Use case
Elastic Beanstalk - Compute
Allows your environment's EC2 instances to perform operations required for your application.
[✓ Elastic Beanstalk - Environment]
Allows access to other AWS service resources that are required to create and manage environments.

© 2025, Amazon Web Services, Inc. or its affiliates.

9. Leave the Permissions section as default and click on 'Next'.


IAM > Roles > Create role

Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

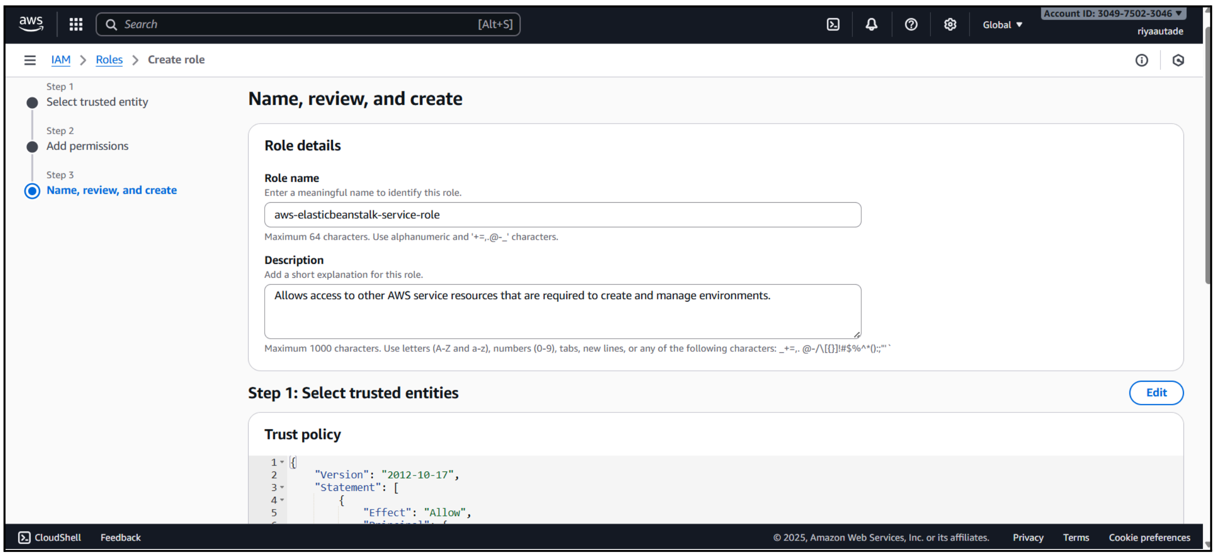
Add permissions Info

Permissions policies (2) Info
The type of role that you selected requires the following policy.

Policy name	Type
AWSElasticBeanstalkEnhancedHealth	AWS managed
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy	AWS managed

Set permissions boundary - *optional*

10. Review all the sections and then click on 'Create Role'.



Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.
aws-elasticbeanstalk-service-role
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

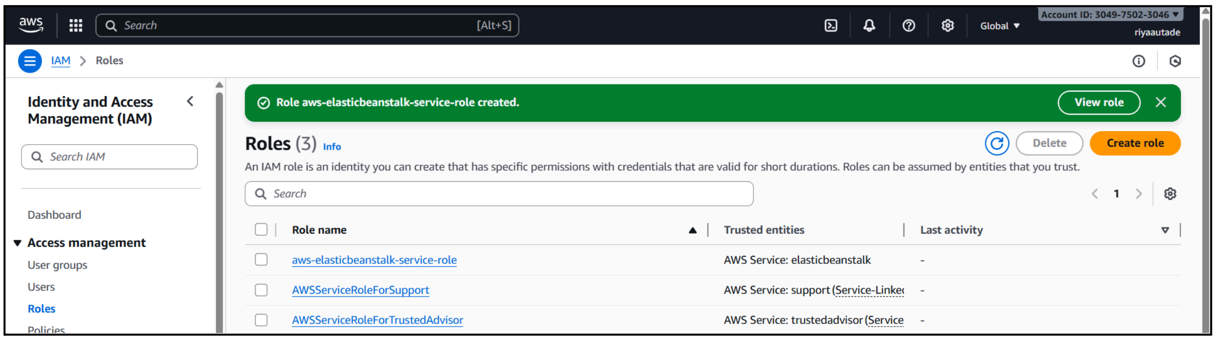
Description
Add a short explanation for this role.
Allows access to other AWS service resources that are required to create and manage environments.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^*();:"'`

Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

11. Service role is successfully created.



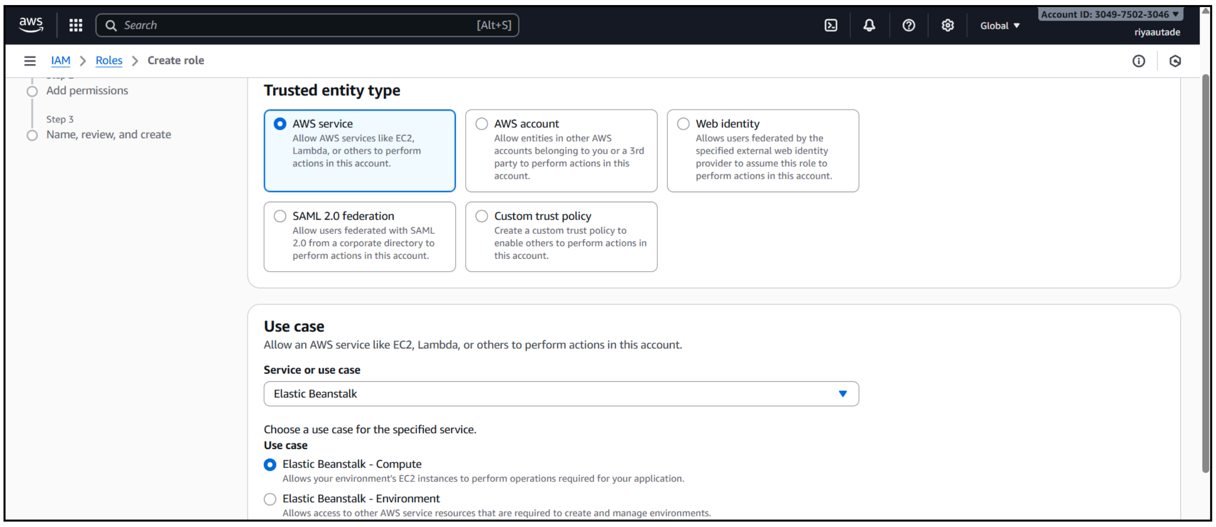
Role aws-elasticbeanstalk-service-role created.

Roles (3) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	-
AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-

12. Go back to the Configuration service access tab. Click on 'Create role' for EC2 instance profile. Now choose 'AWS service' & 'Elastic Beanstalk- Compute'.



Trusted entity type

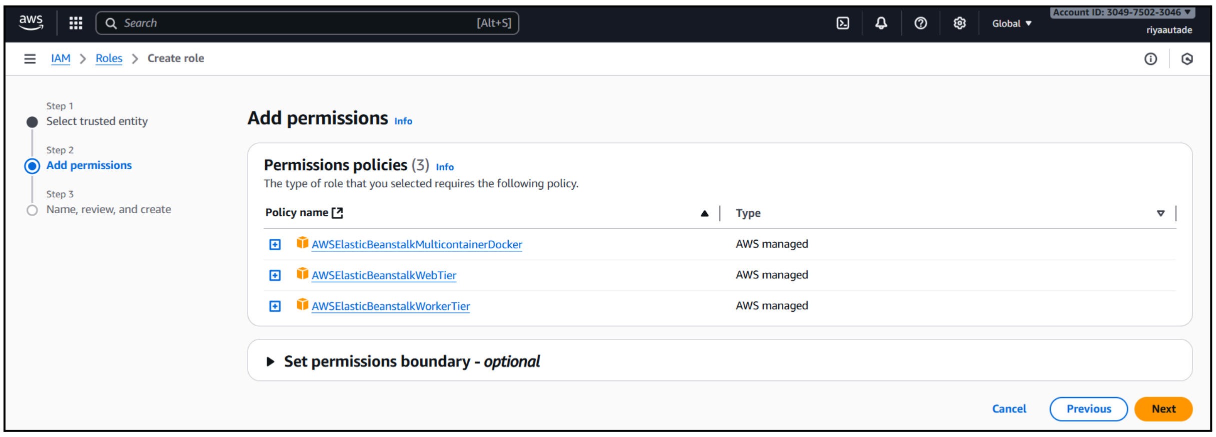
[✓ AWS service] Allow AWS services like EC2, Lambda, or others to perform actions in this account.
AWS account — Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
Web identity — Allows users federated by the specified external web identity provider 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
Elastic Beanstalk

Choose a use case for the specified service.
Use case
[✓ Elastic Beanstalk - Compute] Allows your environment's EC2 instances to perform operations required for your application.
Elastic Beanstalk - Environment — Allows access to other AWS service resources that are required to create and manage environments.

13. Leave the Permissions section as default and click on 'Next'.


IAM > Roles > Create role

Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

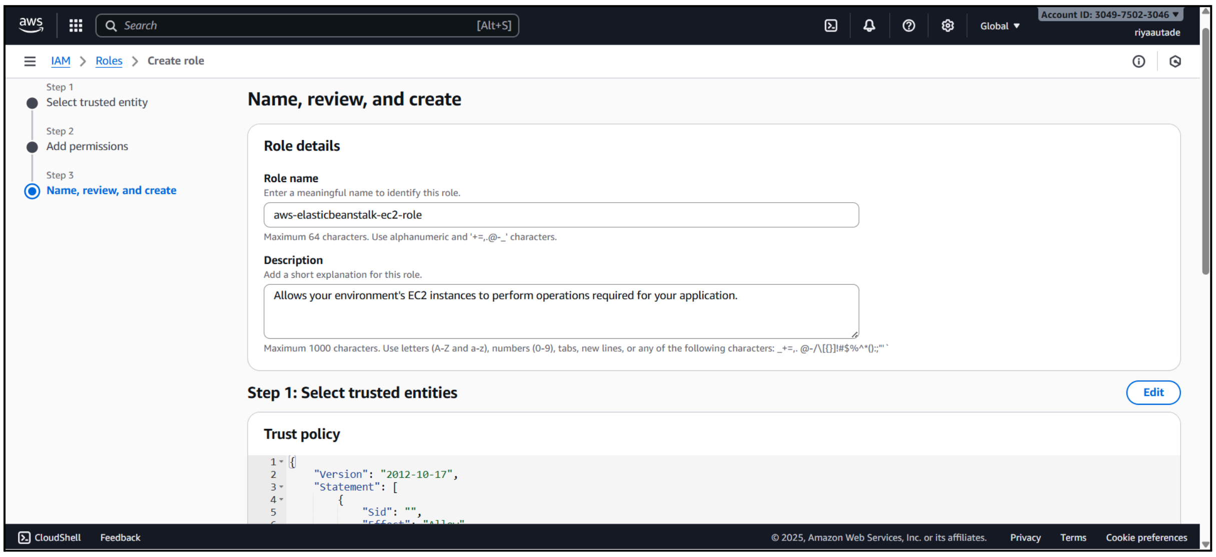
Add permissions Info

Permissions policies (3) Info
The type of role that you selected requires the following policy.

Policy name	Type
AWSElasticBeanstalkMulticontainerDocker	AWS managed
AWSElasticBeanstalkWebTier	AWS managed
AWSElasticBeanstalkWorkerTier	AWS managed

▶ Set permissions boundary - *optional*

14. Review all the sections and then click on 'Create Role'.


IAM > Roles > Create role

Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.
aws-elasticbeanstalk-ec2-role
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.
Allows your environment's EC2 instances to perform operations required for your application.
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: _+=,. @-/\[{}]!#$%^*();"'`

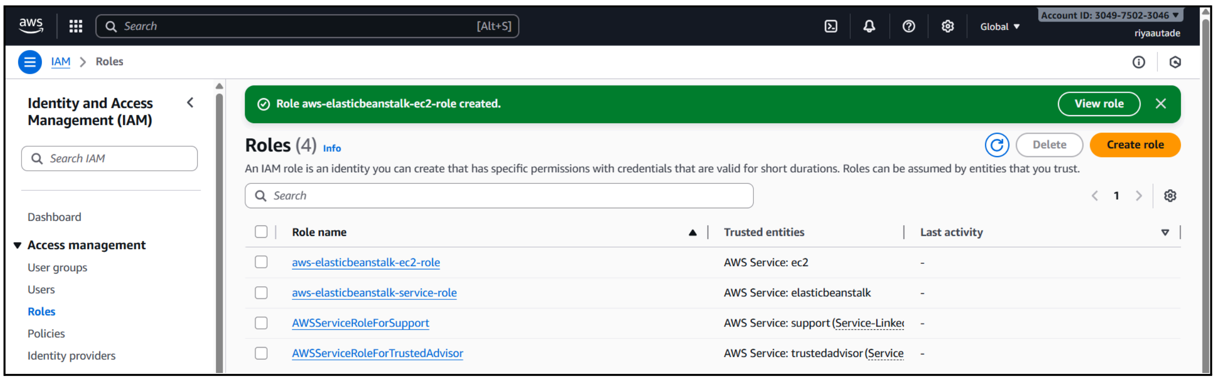
Step 1: Select trusted entities

Trust policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
```

© 2025, Amazon Web Services, Inc. or its affiliates.

15. EC2 instance profile role is successfully created.

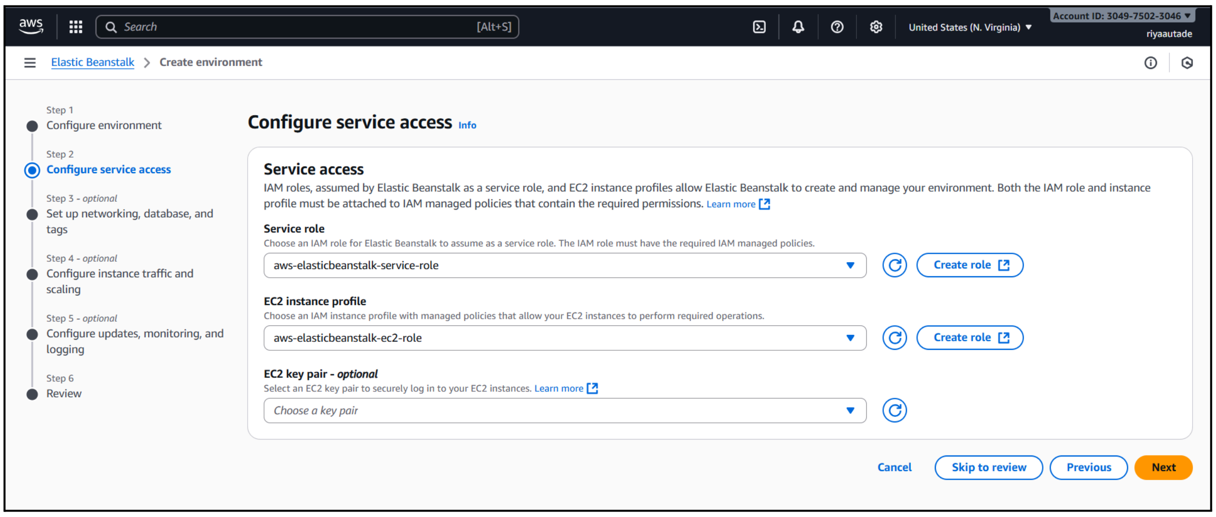

IAM > Roles

Role aws-elasticbeanstalk-ec2-role created.

Roles (4) Info
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
aws-elasticbeanstalk-ec2-role	AWS Service: ec2	-
aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	-
AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-

16. Go back to the Configuration service access tab. Refresh roles and select both the newly created ones: Service & EC2 roles. Leave the EC2 key pair blank and click on 'Next'.


Configure service access

Service access

IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions.

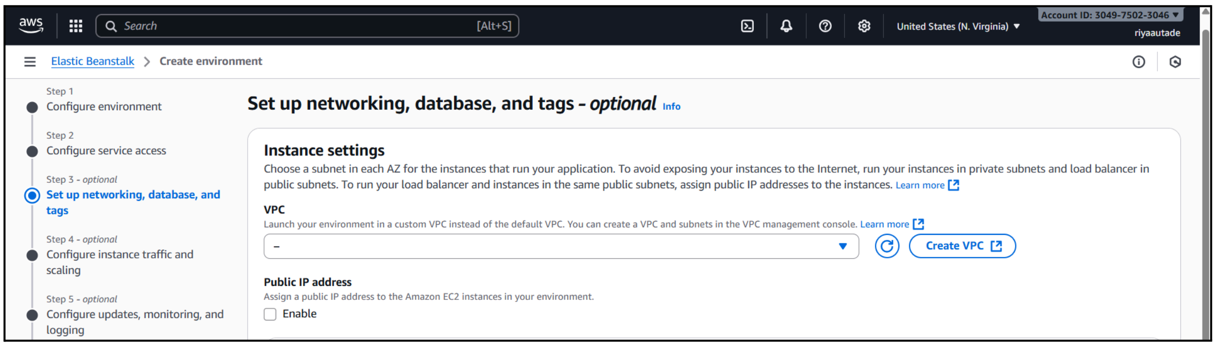
Service role: aws-elasticbeanstalk-service-role

EC2 instance profile: aws-elasticbeanstalk-ec2-role

EC2 key pair - *optional*: Choose a key pair

Elastic Beanstalk > Create environment
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
Create role
Cancel
Skip to review
Previous
Next

17. **Step 3: Networking, database and tags**: Leave the section as default and click on 'Next'.


Set up networking, database, and tags - *optional*

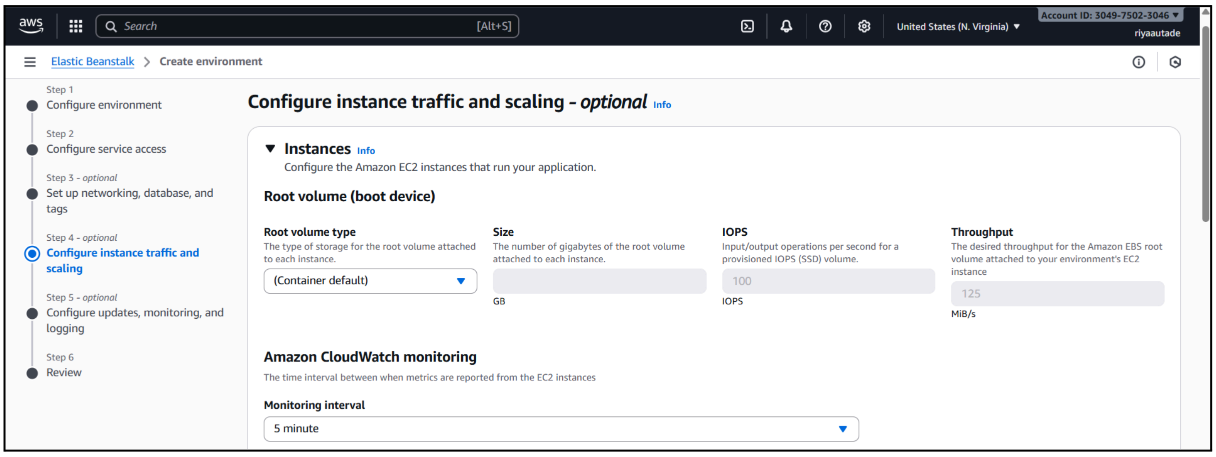
Instance settings

Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances.

VPC: –

Public IP address: Enable ☐

18. **Step 4: Configure instance traffic and scaling**: Leave the section as default and click on 'Next'.


Configure instance traffic and scaling - *optional*

Instances

Configure the Amazon EC2 instances that run your application.

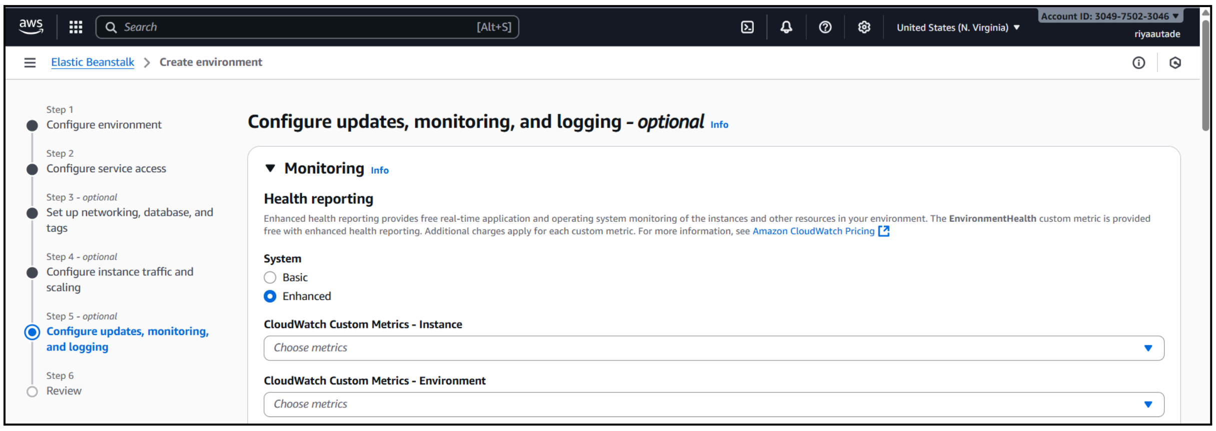
Root volume (boot device)

Root volume type	Size	IOPS	Throughput
(Container default)	GB	100 IOPS	125 MiB/s

Amazon CloudWatch monitoring

Monitoring interval: 5 minute

19. **Step 5: Configure updates, monitoring, and logging:** Leave the section as default & click on 'Next'.



Configure updates, monitoring, and logging - *optional*

Monitoring

Health reporting

Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The **EnvironmentHealth** custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see Amazon CloudWatch Pricing

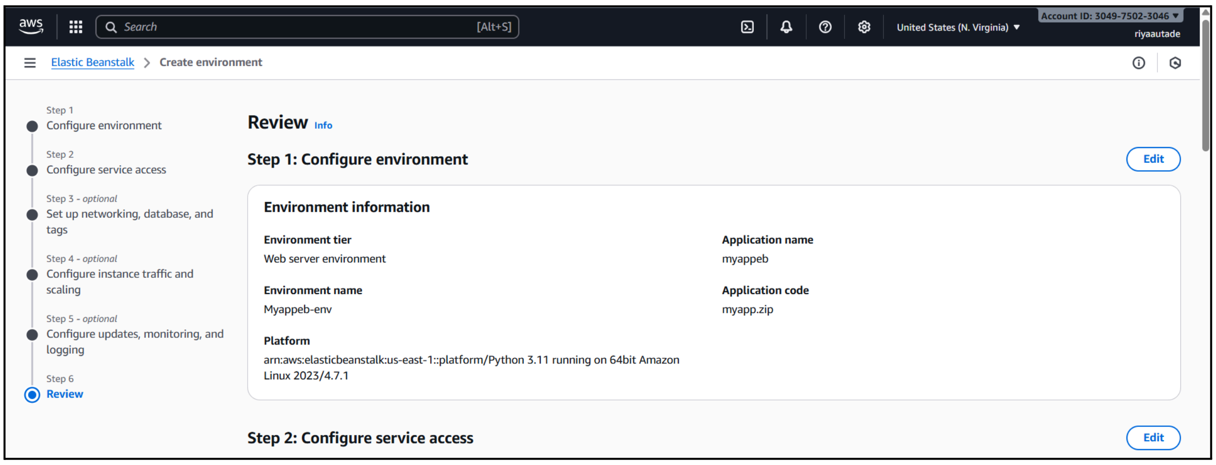
System

Basic [✓ Enhanced]

CloudWatch Custom Metrics - Instance: Choose metrics

CloudWatch Custom Metrics - Environment: Choose metrics

20. **Step 6: Review:** Go through all the Steps from 1 to 5 and review everything.



Review

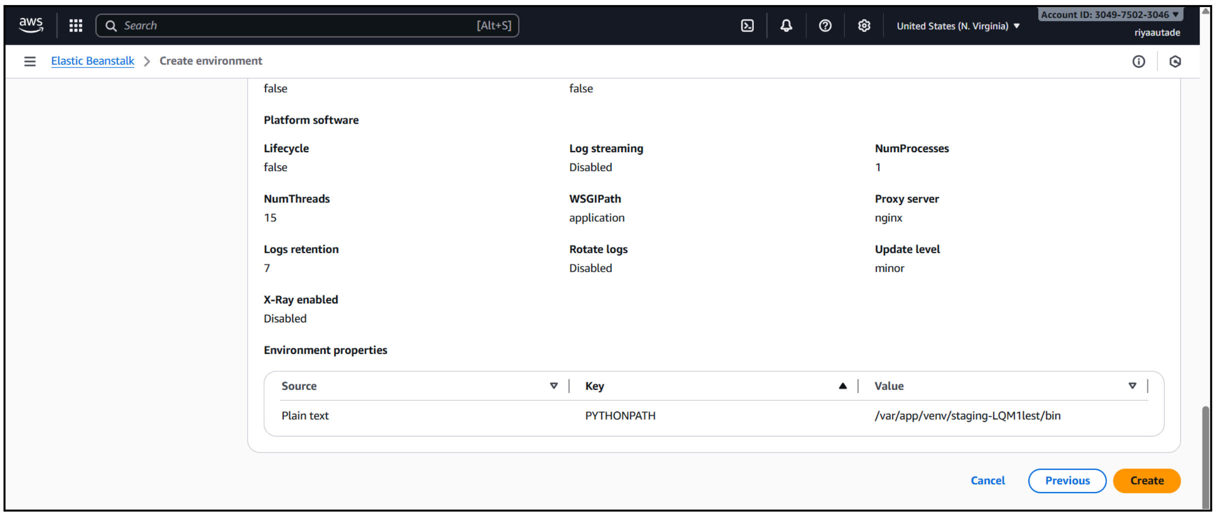
Step 1: Configure environment

Environment information

Environment tier Web server environment	**Application name** myappeb
Environment name Myappeb-env	**Application code** myapp.zip
Platform arn:aws:elasticbeanstalk:us-east-1::platform/Python 3.11 running on 64bit Amazon Linux 2023/4.7.1	

Step 2: Configure service access

21. Click on 'Create'.



false	false	

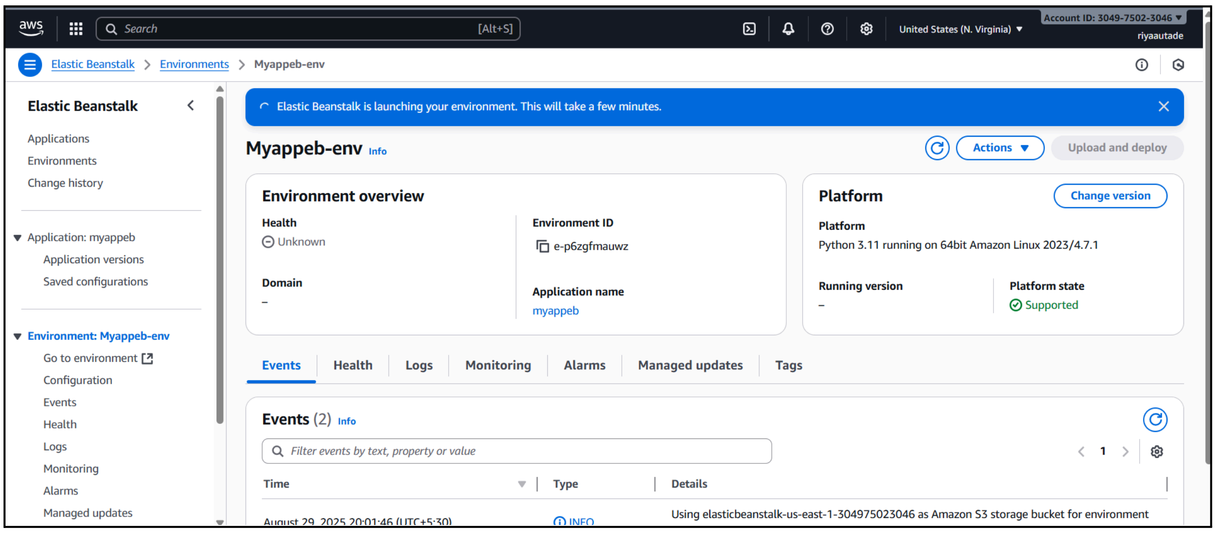
Platform software

Lifecycle false	**Log streaming** Disabled	**NumProcesses** 1
NumThreads 15	**WSGIPath** application	**Proxy server** nginx
Logs retention 7	**Rotate logs** Disabled	**Update level** minor
X-Ray enabled Disabled		

Environment properties

Source	Key	Value
Plain text	PYTHONPATH	/var/app/venv/staging-LQM1lest/bin

22. Elastic Beanstalk takes a few minutes to create your environment.



Elastic Beanstalk is launching your environment. This will take a few minutes.

Myappeb-env Info

Environment overview

Health: Unknown
Environment ID: e-p6zgfmauwz
Domain: –
Application name: myappeb

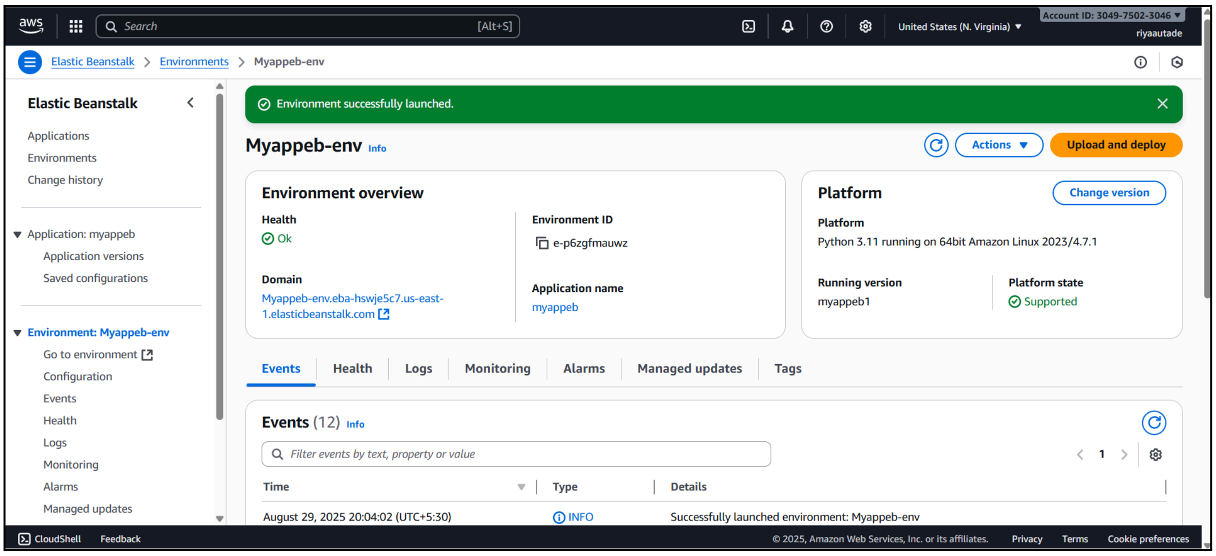
Platform

Platform: Python 3.11 running on 64bit Amazon Linux 2023/4.7.1
Running version: –
Platform state: Supported

Events (2) Info

Time	Type	Details
August 29, 2025 20:01:46 (UTC+5:30)	INFO	Using elasticbeanstalk-us-east-1-304975023046 as Amazon S3 storage bucket for environment

23. **Environment is successfully launched using Elastic Beanstalk.** Check for green 'Health' status and 'Platform State' as well.



Environment successfully launched.

Myappeb-env Info

Environment overview

Health: Ok
Environment ID: e-p6zgfmauwz
Domain: Myappeb-env.eba-hswje5c7.us-east-1.elasticbeanstalk.com
Application name: myappeb

Platform

Platform: Python 3.11 running on 64bit Amazon Linux 2023/4.7.1
Running version: myappeb1
Platform state: Supported

Events (12) Info

Time	Type	Details
August 29, 2025 20:04:02 (UTC+5:30)	INFO	Successfully launched environment: Myappeb-env

24. **Access the application with the URL under 'Domain'.**

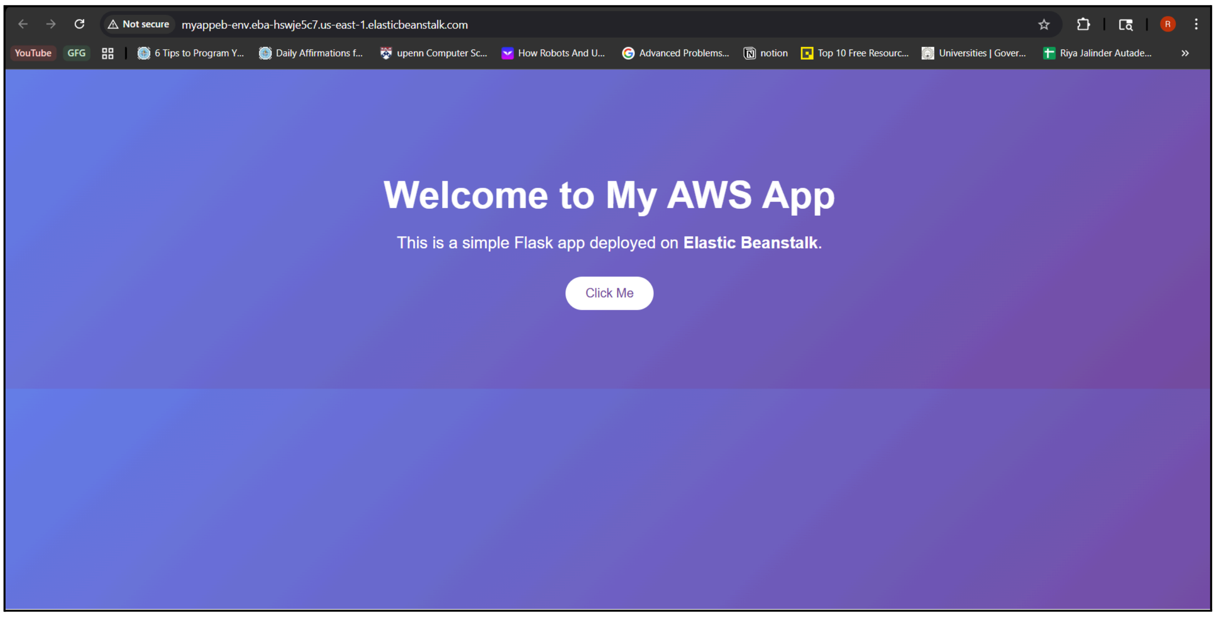
Link: http://myappeb-env.eba-hswje5c7.us-east-1.elasticbeanstalk.com/

The application is now deployed and publicly accessible.

[Note: The environment will be terminated soon to avoid charges, hence the link will be disabled.]

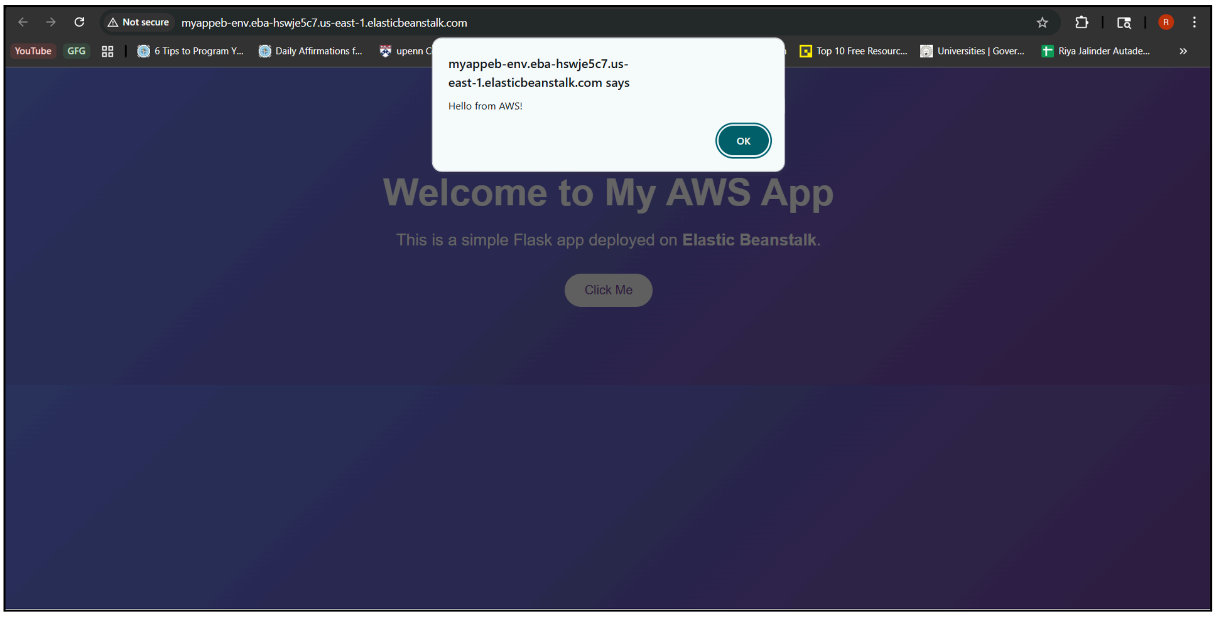
Result:

The Flask application is successfully deployed and running using AWS Elastic Beanstalk.



Welcome to My AWS App

This is a simple Flask app deployed on **Elastic Beanstalk**.



myappeb-env.eba-hswje5c7.us-east-1.elasticbeanstalk.com says

Hello from AWS!

Welcome to My AWS App

This is a simple Flask app deployed on **Elastic Beanstalk**.