

**NAME: Om Muddebihal**

**SAP ID: 86092300033**

**ROLL NO: A032**

## Practical-2 Platform as a service using AWS.

Q1) Writeup:-

### ❖ 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:

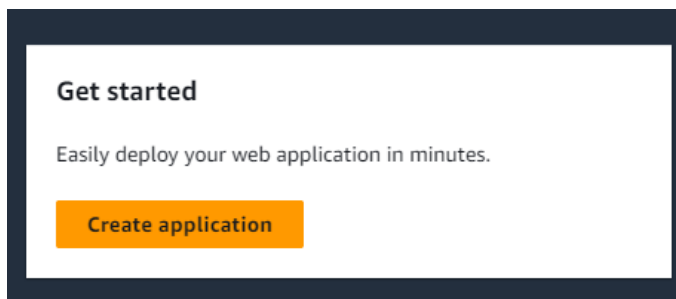
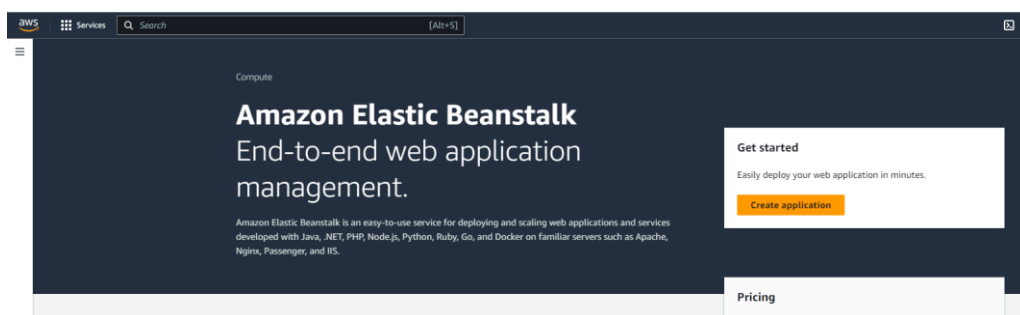
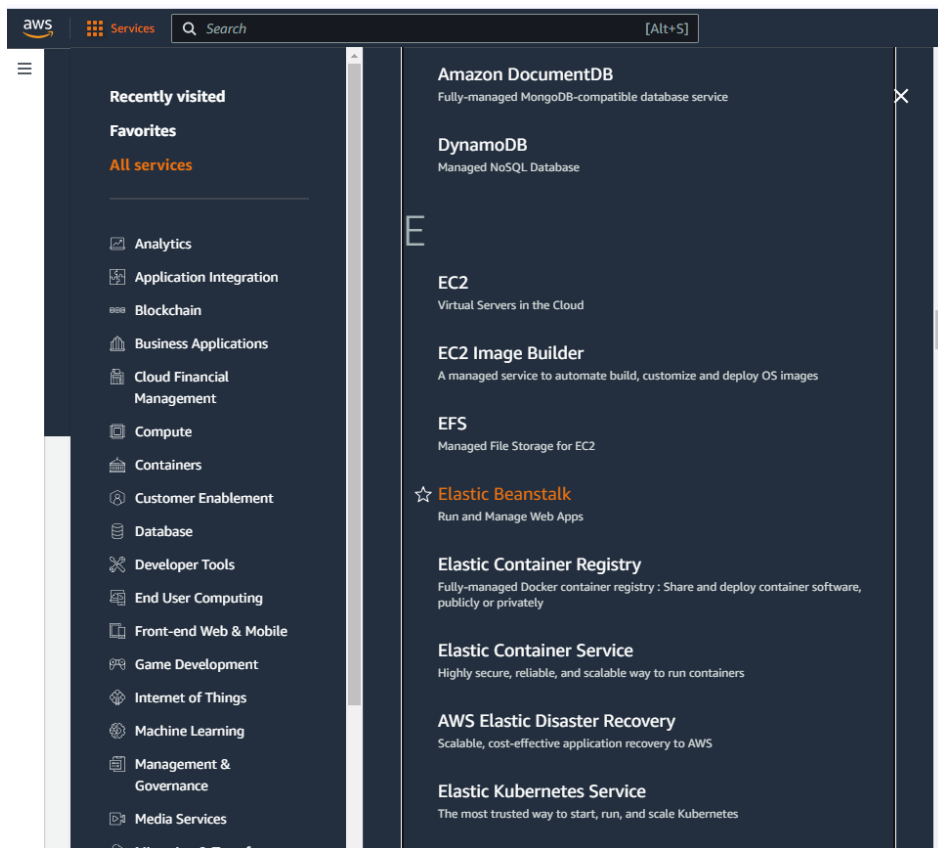
- Create users and groups with specific permissions.
- Define roles that grant access to specific resources and actions.
- Use temporary credentials for short-lived tasks.
- Monitor user activity and identify potential security threats.

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

### 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

Python Web App

Maximum length of 100 characters.

► **Application tags (optional)**

## Step 3: Give environment name and add a short description for example

### Environment information [Info](#)

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

Environment name

PythonWebApp-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

.ap-south-1.elasticbeanstalk.com

Check availability

Environment description

This is my first web application.

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

**Platform** [Info](#)

Platform type

☒ **Managed platform**  
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#)

☐ **Custom platform**  
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Python ▼

Platform branch

Python 3.11 running on 64bit Amazon Linux 2023 ▼

Platform version

4.0.7 (Recommended) ▼

Step 5:

**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)

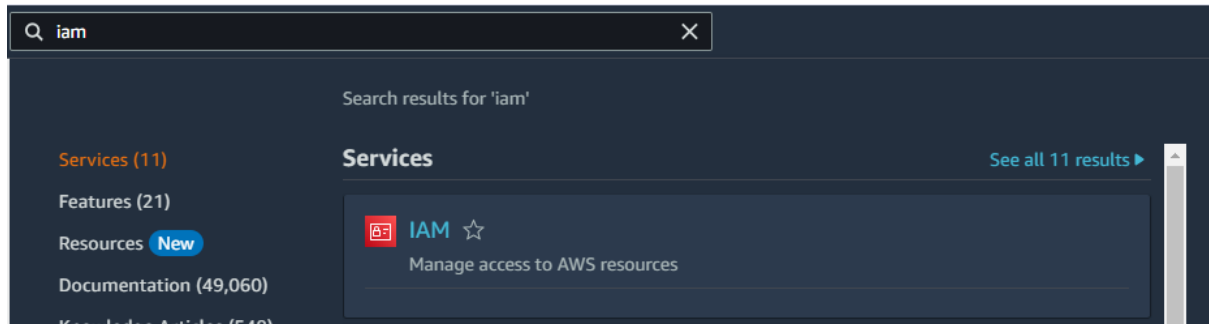
☐ High availability

☐ High availability (using spot and on-demand instances)

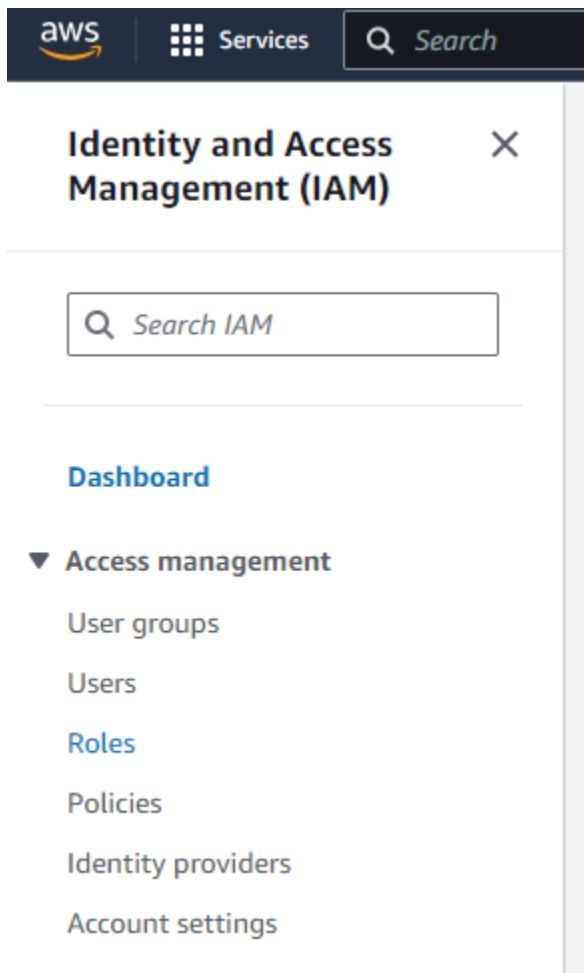
☐ Custom configuration

Cancel **Next**

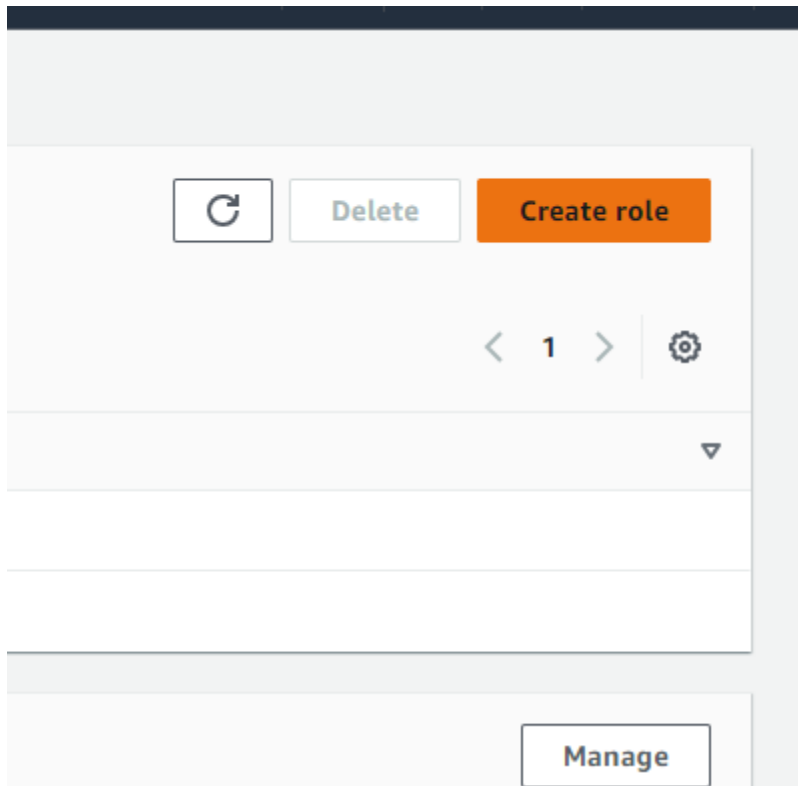
Step 6: Create a role. ( open new tab and then perform , keep previous tab as it is we want to work on it later )



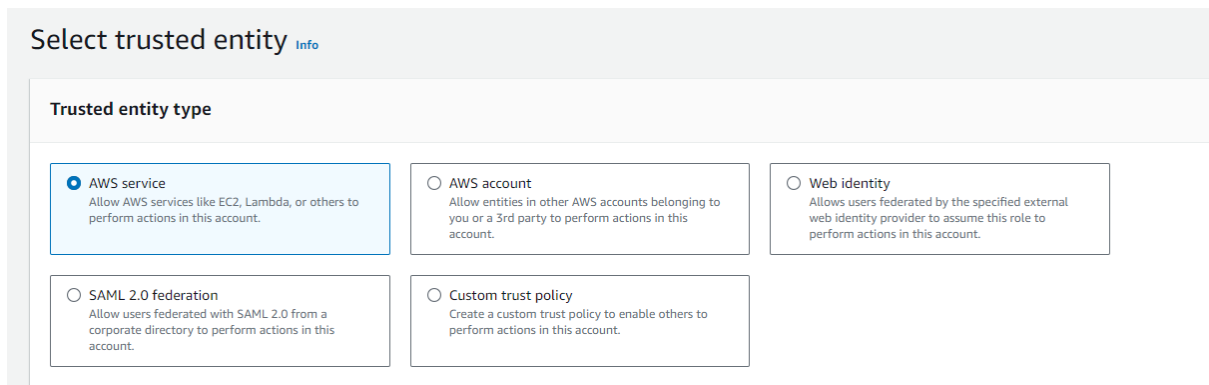
Step 7:



Step 8:



Step 9: Select entity



Step 10: Select use case

**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.
- ☐ **EC2 - Spot Fleet Tagging**  
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.
- ☐ **EC2 - Spot Instances**  
Allows EC2 Spot instances to launch and manage spot instances on your behalf.
- ☐ **EC2 - Spot Fleet**  
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.
- ☐ **EC2 - Scheduled Instances**  
Allows EC2 Scheduled instances to manage instances on your behalf.

Cancel Next

## Step 11: Toggle permission's

**Add permissions** [Info](#)

**Permissions policies** (3/908) [Info](#)

Choose one or more policies to attach to your new role.

Filter by Type  
All types 14 matches

Search: beanstalk

<input type="checkbox"/>	Policy name	Type	Description
<input type="checkbox"/>	<a href="#">AdministratorAccess-AWSElasticBeanstalk</a>	AWS managed	Grants account administrative permis...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkCustomPlatformforEC2Role</a>	AWS managed	Provide the instance in your custom pl...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkEnhancedHealth</a>	AWS managed	AWS Elastic Beanstalk Service policy f...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy</a>	AWS managed	This policy is for the AWS Elastic Bean...
<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalkMulticontainerDocker</a>	AWS managed	Provide the instances in your multicon...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkReadOnly</a>	AWS managed	Grants read-only permissions. Explicitl...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleCore</a>	AWS managed	AWSElasticBeanstalkRoleCore (Elastic ...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleCWL</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleECS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleRDS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleSNS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	<a href="#">AWSElasticBeanstalkRoleWorkerTier</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalkWebTier</a>	AWS managed	Provide the instances in your web serv...
<input checked="" type="checkbox"/>	<a href="#">AWSElasticBeanstalkWorkerTier</a>	AWS managed	Provide the instances in your worker e...

## Step 12: Give name to role

**Name, review, and create**

**Role details**

**Role name**  
Enter a meaningful name to identify this role.

PythonRole

Maximum 64 characters. Use alphanumeric and '+', '@', '-' characters.

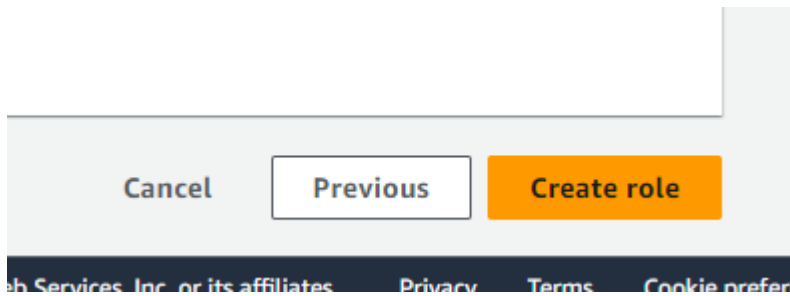
**Description**  
Add a short explanation for this role.

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+', '@', '-' characters.

## Step 13:





Step 14: Select role in previous working page

## 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**

☒ Create and use new service role  
☐ Use an existing service role

**Service role name**  
Enter the name for an IAM role that Elastic Beanstalk will create to assume as a service role. Beanstalk will attach the required managed policies to it.

aws-elasticbeanstalk-service-role

[View permission details](#)

**EC2 key pair**  
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

Choose a key pair ▼ [Refresh](#)

**EC2 instance profile**  
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

PythonRole ▼ [Refresh](#)

[View permission details](#)

[Cancel](#) [Skip to review](#) [Previous](#) [Next](#)

Step 15: Select database

Set up networking, database, and tags - optional [Info](#)

Virtual Private Cloud (VPC)

VPC

Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console. [Learn more](#)

vpc-07af482209450bcb5 | (172.31.0.0/16)

[Create custom VPC](#)

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. [Learn more](#)

Public IP address

Assign a public IP address to the Amazon EC2 instances in your environment.

☐ Activated

Instance subnets

	Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/>	ap-south-1b	subnet-04155acce...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1c	subnet-0bd04bd1e...	172.31.16.0/20	
<input type="checkbox"/>	ap-south-1a	subnet-0edc0372c...	172.31.32.0/20	

Step 16:

Database [Info](#)

Integrate an RDS SQL database with your environment. [Learn more](#)

Database subnets

If your Elastic Beanstalk environment is attached to an Amazon RDS, choose subnets for your database instances. [Learn more](#)

Choose database subnets (3)

	Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/>	ap-south-1b	subnet-04155acce...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1c	subnet-0bd04bd1e...	172.31.16.0/20	
<input type="checkbox"/>	ap-south-1a	subnet-0edc0372c...	172.31.32.0/20	

Step 17:

Cancel

Skip to review

Previous

Next

Step 18:

Skip Step 4: Keep as it is

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](#)

---

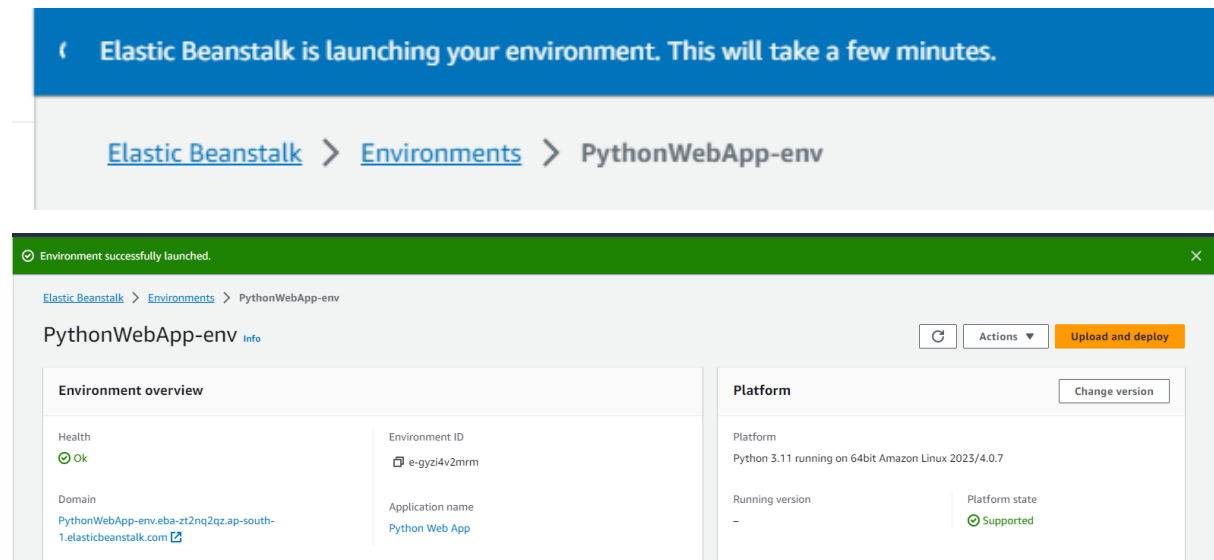
Step 6

[Review](#)

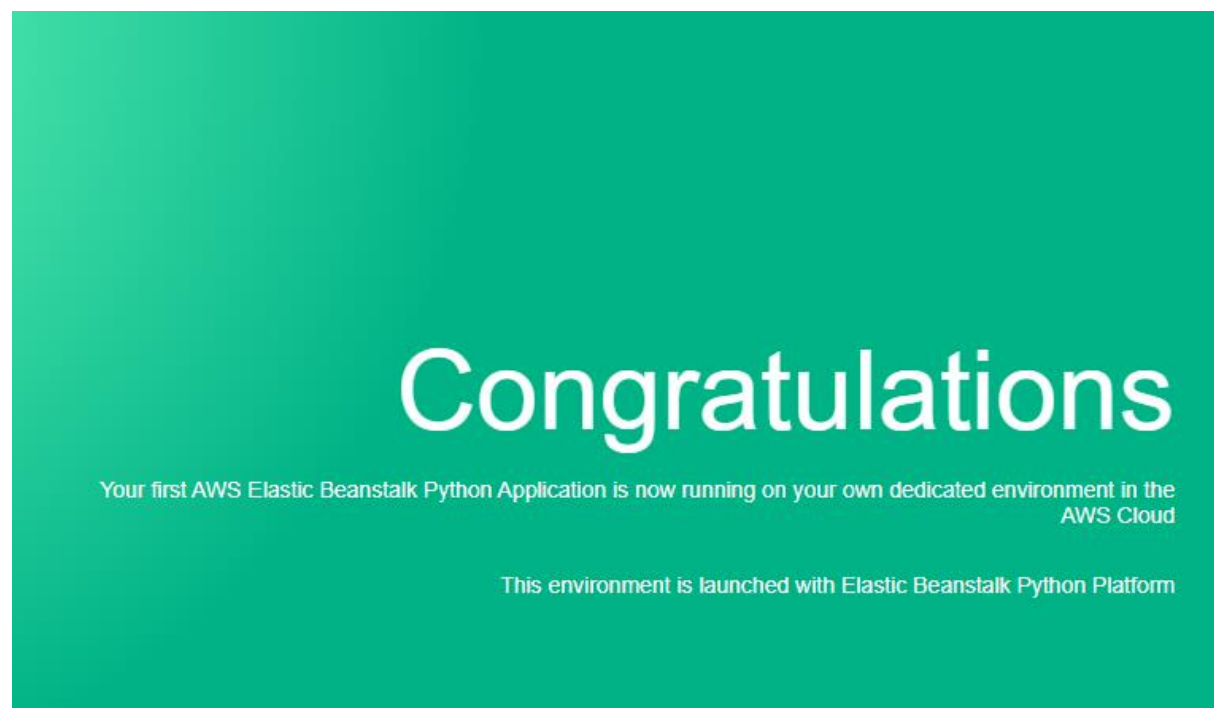
Step 19:

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 see instances running (1)

**Resources**

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Instances (running)	1	Auto Scaling Groups	1	Dedicated Hosts	0
Elastic IPs	1	Instances	1	Key pairs	2
Load balancers	0	Placement groups	0	Security groups	3
Snapshots	0	Volumes	1		

**Instances (1)**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
PythonWebAp...	i-0c70450d216c358bc	Running	t3.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-13-233-185-84-ap...	13.233.185.84	13.233.185.84

Java

Step 1: create web app

**Recently visited**

**Favorites**

**All services**

- Analytics
- Application Integration
- Blockchain
- Business Applications
- Cloud Financial Management
- Compute
- Containers
- Customer Enablement
- Database
- Developer Tools
- End User Computing
- Front-end Web & Mobile
- Game Development
- Internet of Things
- Machine Learning
- Management & Governance
- Media Services
- Migration & Transfer

**Amazon DocumentDB**  
Fully-managed MongoDB-compatible database service

**DynamoDB**  
Managed NoSQL Database

**EC2**  
Virtual Servers in the Cloud

**EC2 Image Builder**  
A managed service to automate build, customize and deploy OS images

**EFS**  
Managed File Storage for EC2

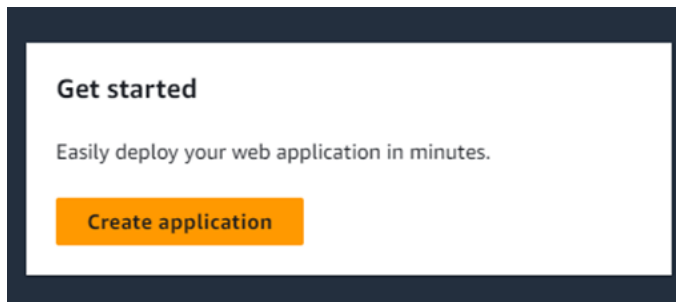
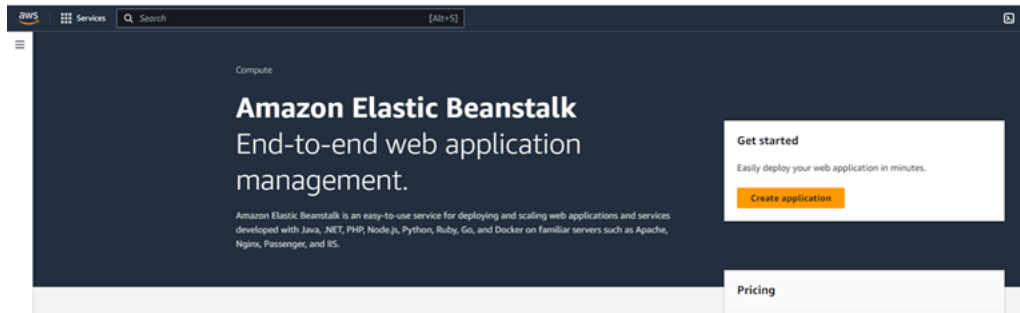
**Elastic Beanstalk**  
Run and Manage Web Apps

**Elastic Container Registry**  
Fully-managed Docker container registry : Share and deploy container software, publicly or privately

**Elastic Container Service**  
Highly secure, reliable, and scalable way to run containers

**AWS Elastic Disaster Recovery**  
Scalable, cost-effective application recovery to AWS

**Elastic Kubernetes Service**  
The most trusted way to start, run, and scale Kubernetes



## Step 2: Name your application

### 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

Java Web App

Maximum length of 100 characters.

## Step 3

## Environment information [Info](#)

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

Environment name

JavaWebApp-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

.ap-south-1.elasticbeanstalk.com

[Check availability](#)

Environment description

This is my first Java Web App

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

## Platform [Info](#)

Platform type

☒ Managed platform

Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#) 

☐ Custom platform

Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Java

Platform branch

Corretto 21 running on 64bit Amazon Linux 2023

Platform version

4.2.0 (Recommended)

Step 5:

**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

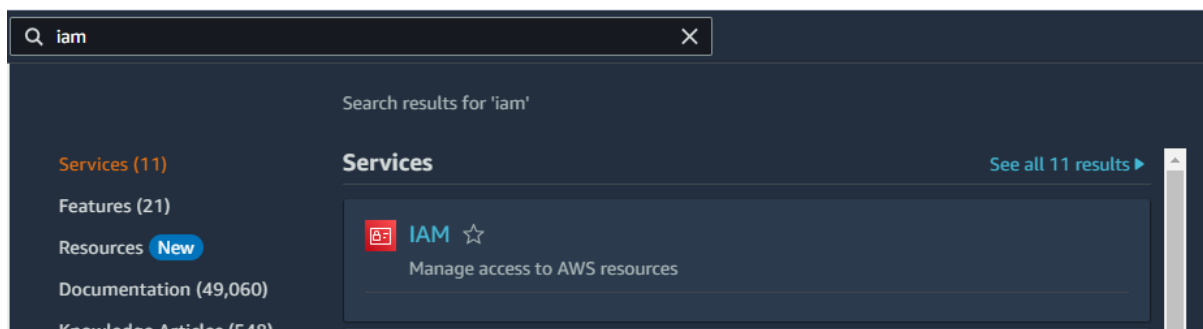
☐ High availability (using spot and on-demand instances)

☐ Custom configuration

Cancel

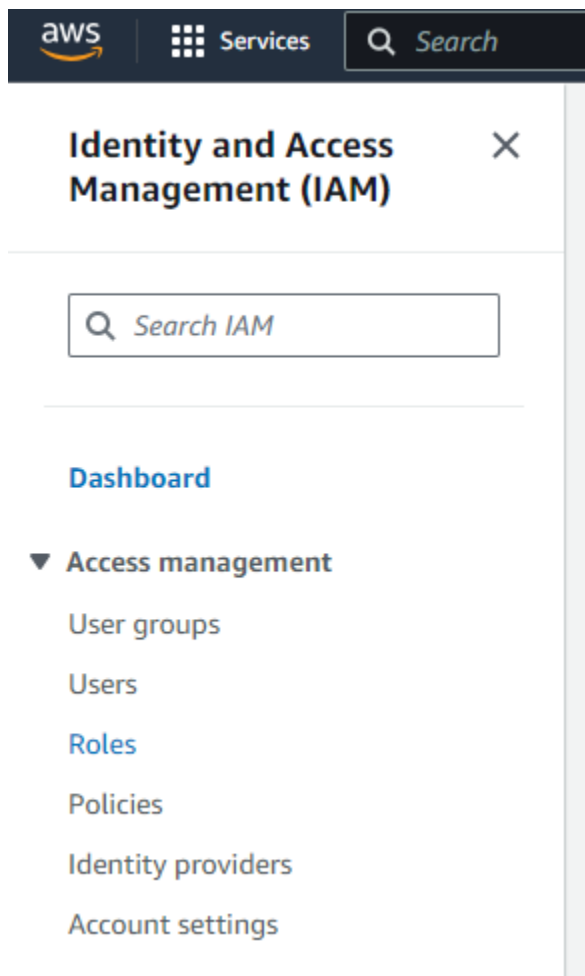
Next

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

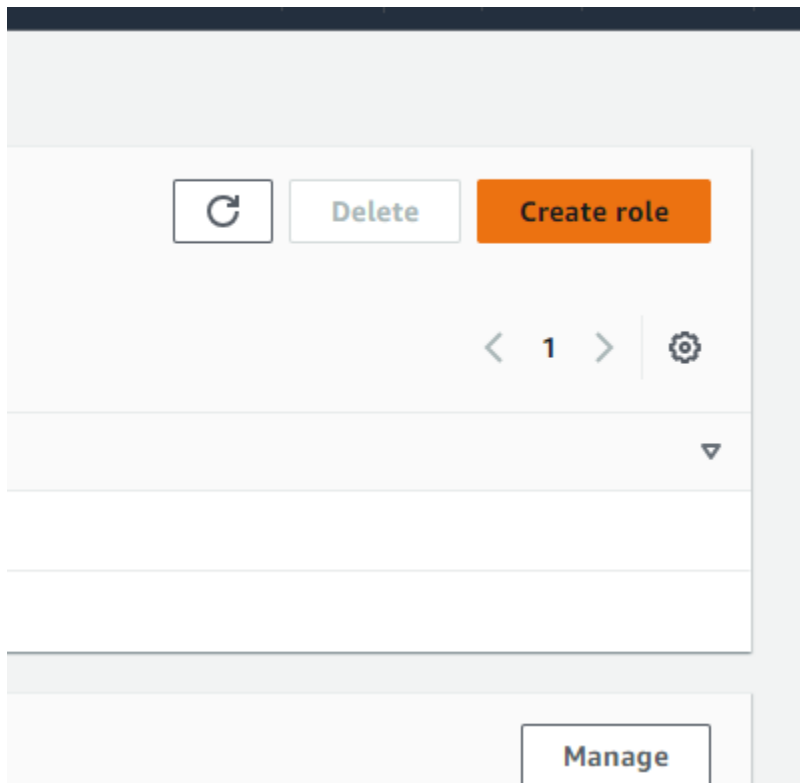


Step 7:





Step 8:



## Step 9: Select entity

**Select trusted entity** [Info](#)

**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.

## Step 10: Select use case

**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.
- ☐ **EC2 - Spot Fleet Tagging**  
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.
- ☐ **EC2 - Spot Instances**  
Allows EC2 Spot instances to launch and manage spot instances on your behalf.
- ☐ **EC2 - Spot Fleet**  
Allows EC2 Spot Fleet to launch and manage spot fleet instances on your behalf.
- ☐ **EC2 - Scheduled Instances**  
Allows EC2 Scheduled instances to manage instances on your behalf.

Cancel **Next**

## Step 11: Toggle permission's

Add permissions Info

Permissions policies (3/908) Info

Choose one or more policies to attach to your new role.















Q beanstalk

Filter by Type

All types

14 matches

< 1 >

	Policy name <small>Info</small>	Type	Description
<input type="checkbox"/>	 <a href="#">AdministratorAccess-AWSElasticBeanstalk</a>	AWS managed	Grants account administrative permis...
<input type="checkbox"/>	 <a href="#">AWSElasticBeanstalkCustomPlatformforEC2Role</a>	AWS managed	Provide the instance in your custom pl...
<input type="checkbox"/>	 <a href="#">AWSElasticBeanstalkEnhancedHealth</a>	AWS managed	AWS Elastic Beanstalk Service policy f...
<input type="checkbox"/>	 <a href="#">AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy</a>	AWS managed	This policy is for the AWS Elastic Bean...
<input checked="" type="checkbox"/>	 <a href="#">AWSElasticBeanstalkMulticontainerDocker</a>	AWS managed	Provide the instances in your multicon...
<input type="checkbox"/>	 <a href="#">AWSElasticBeanstalkReadOnly</a>	AWS managed	Grants read-only permissions. Explicitl...
<input type="checkbox"/>	 <a href="#">AWSElasticBeanstalkRoleCore</a>	AWS managed	AWSElasticBeanstalkRoleCore (Elastic ...
<input type="checkbox"/>	 <a href="#">AWSElasticBeanstalkRoleCWL</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	 <a href="#">AWSElasticBeanstalkRoleECS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	 <a href="#">AWSElasticBeanstalkRoleRDS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	 <a href="#">AWSElasticBeanstalkRoleSNS</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input type="checkbox"/>	 <a href="#">AWSElasticBeanstalkRoleWorkerTier</a>	AWS managed	(Elastic Beanstalk operations role) Allo...
<input checked="" type="checkbox"/>	 <a href="#">AWSElasticBeanstalkWebTier</a>	AWS managed	Provide the instances in your web serv...
<input checked="" type="checkbox"/>	 <a href="#">AWSElasticBeanstalkWorkerTier</a>	AWS managed	Provide the instances in your worker e...

Step 12: Give name to role

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

Maximum 64 characters. Use alphanumeric and '+', '@', '-' characters.

Description

Add a short explanation for this role.

Maximum 1000 characters. Use alphanumeric and '+', '@', '-' characters.

Step 13:

Cancel

Previous

Create role

h Services, Inc. or its affiliates

Privacy

Terms

Cookie prefer

Step 14: Select role in previous working page

## 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

- ☒ Create and use new service role
- ☐ Use an existing service role

#### Service role name

Enter the name for an IAM role that Elastic Beanstalk will create to assume as a service role. Beanstalk will attach the required managed policies to it.

[View permission details](#)

#### EC2 key pair

Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

#### EC2 instance profile

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

[View permission details](#)

## Step 15: Select database

### Set up networking, database, and tags - *optional* [Info](#)

#### Virtual Private Cloud (VPC)

**VPC**  
Launch your environment in a custom VPC instead of the default VPC. You can create a VPC and subnets in the VPC management console. [Learn more](#)

[Create custom VPC](#)

#### 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. [Learn more](#)

#### Public IP address

Assign a public IP address to the Amazon EC2 instances in your environment.

☐ Activated

#### Instance subnets

<input checked="" type="checkbox"/>	Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/>	ap-south-1b	subnet-04155acce...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1c	subnet-0bd04bd1e...	172.31.16.0/20	
<input type="checkbox"/>	ap-south-1a	subnet-0edc0372c...	172.31.32.0/20	

## Step 16:

## Database [Info](#)

Integrate an RDS SQL database with your environment. [Learn more](#)

### Database subnets

If your Elastic Beanstalk environment is attached to an Amazon RDS, choose subnets for your database instances. [Learn more](#)

#### Choose database subnets (3)

<input type="checkbox"/>	Availability Zone	Subnet	CIDR	Name
<input checked="" type="checkbox"/>	ap-south-1b	subnet-04155acce...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1c	subnet-0bd04bd1e...	172.31.16.0/20	
<input type="checkbox"/>	ap-south-1a	subnet-0edc0372c...	172.31.32.0/20	

Step 17:

Cancel

Skip to review

Previous

Next

Step 18:

Skip Step 4: Keep as it is

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](#)

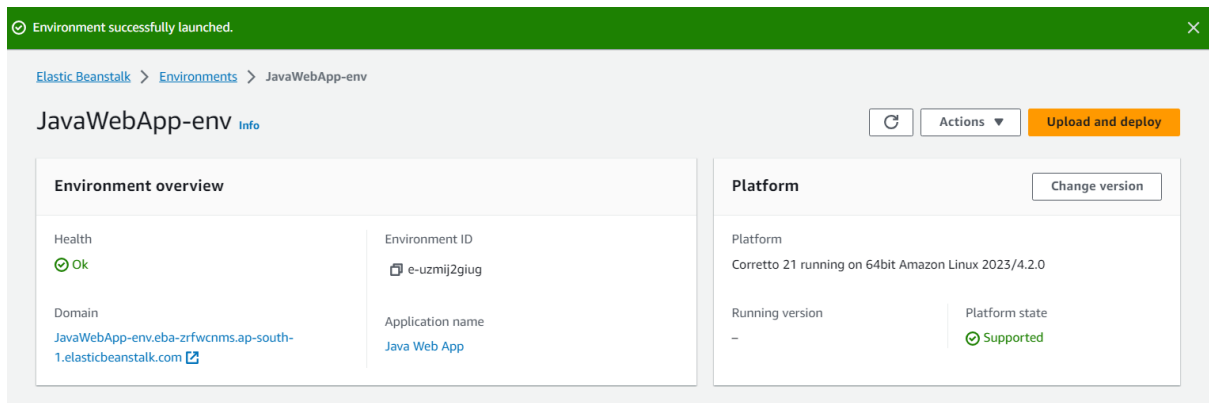
Step 19:

Step 5 : Keep as it is and submit

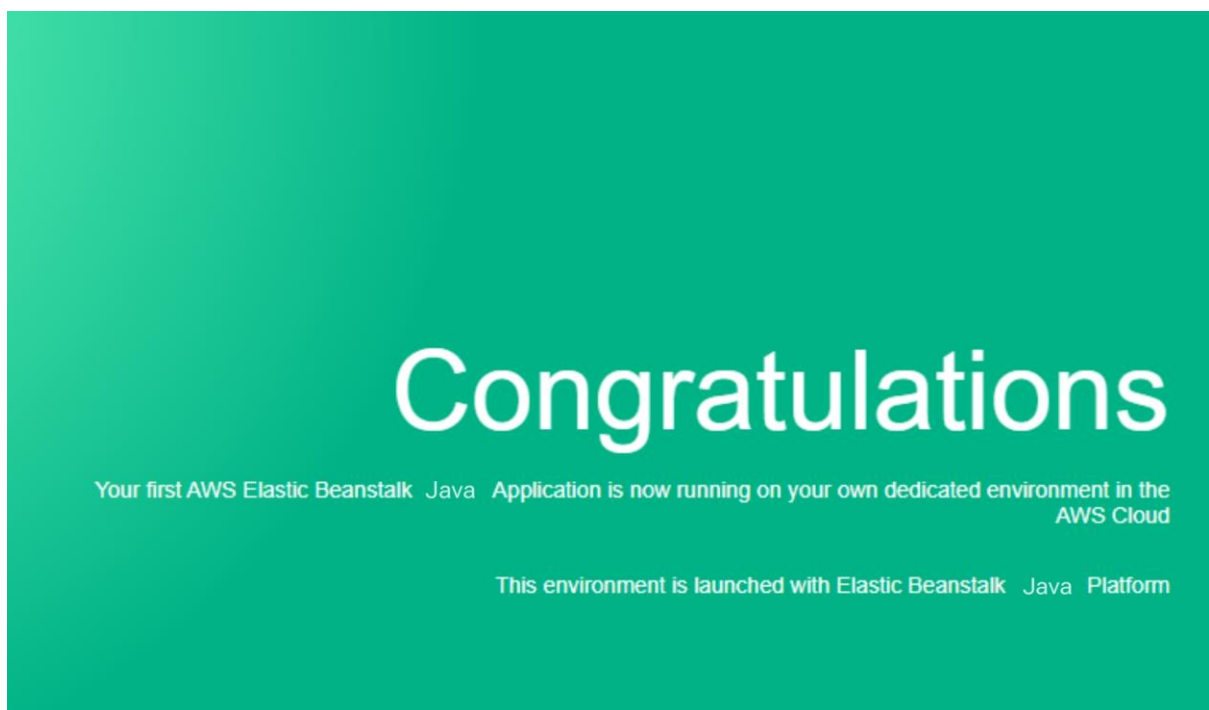
Step 20: Launching environment

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

[Elastic Beanstalk](#) > [Environments](#) > [JavaWebApp-env](#)



Step 21: Successfully launched our application



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

