

**SVKM'S NMIM'S 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**

AWS Elastic Beanstalk is the PaaS offering from AWS that makes it easy to deploy and manage applications in the cloud without worrying about the infrastructure. Developers can simply upload their code and Elastic Beanstalk automatically handles provisioning, load balancing, auto-scaling and monitoring.

- **Elastic Beanstalk**

Elastic Beanstalk is a PaaS service offered by AWS to deploy and scale web applications quickly without worrying about the infrastructure. It automatically handles capacity provisioning, load balancing, scaling and application health monitoring. Developers just have to upload their code and Elastic Beanstalk will deploy it on AWS infrastructure like EC2, auto scale it and monitor it.

Why Elastic BeanStalk

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

- **Components of beanstalk**

- i. Application: This is the actual web application code packaged into a zip and uploaded to Elastic Beanstalk.
- ii. Application Version: Each deployment of code is an application version. Rollbacks can be done to previous versions.
- iii. Environment: This is a version of the application running on AWS resources. We can create multiple environments like dev, test, prod etc from the same application.

- iv. Configuration Templates: These allow customization of the AWS resources powering an environment like EC2 instance type, autoscaling settings etc.
- v. Events: Important lifecycle events like deployments, scaling etc are logged for debugging.

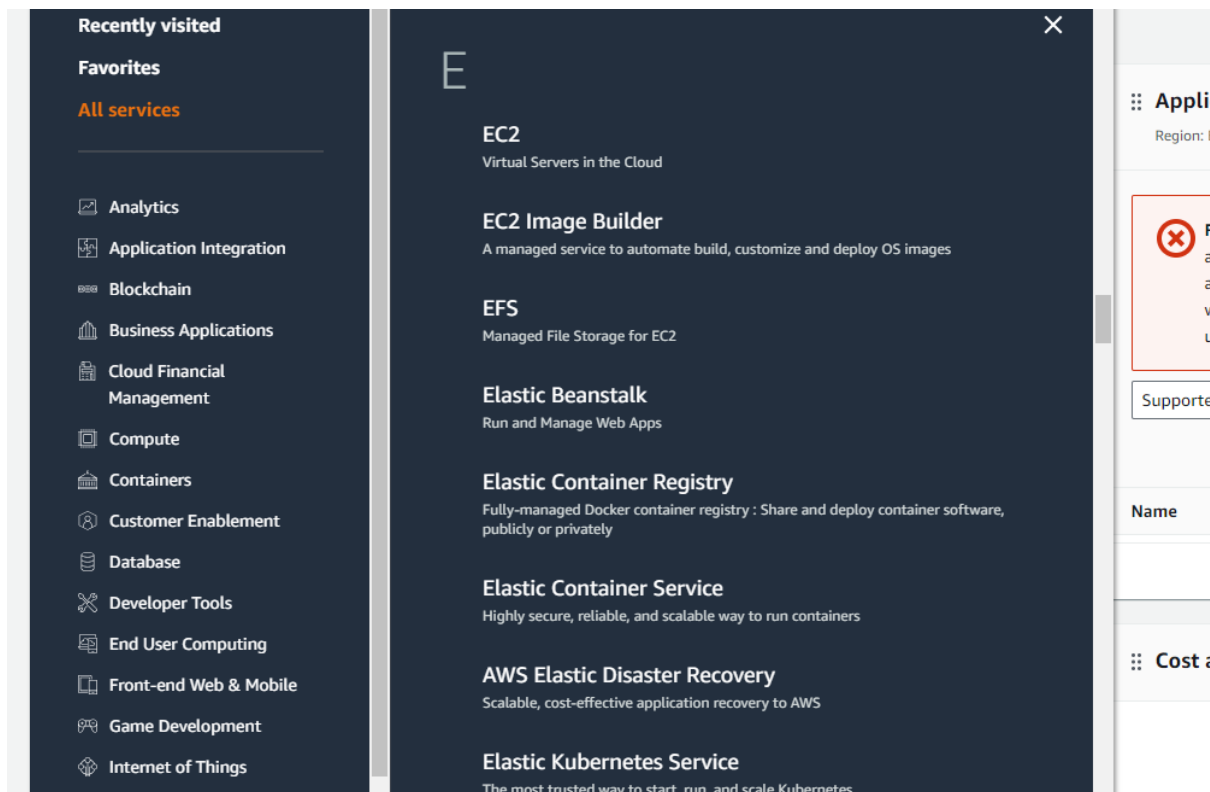
- **IAM**

- i. IAM allows managing users, roles and permissions to access AWS services and resources.
- ii. Users can be created and assigned granular permissions policies.
- iii. Roles can be created with permissions and then assigned to AWS resources like EC2 instances.
- iv. Policies define the permissions like which AWS actions can be performed on which resources.
- v. IAM is important for security, access control and compliance in AWS.

- 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

Compute

# Amazon Elastic Beanstalk

## End-to-end web application management.

Amazon Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

### Get started

Easily deploy your web application in minutes.

Create application

### Pricing

There's no additional charge for Elastic Beanstalk. You pay for Amazon Web Services resources that we create to store and run your web application, like Amazon S3 buckets and Amazon EC2 instances.

### Getting started

### Get started

You simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, and automatic scaling to web application health monitoring, with ongoing fully managed patch and security updates. [Learn more](#)

### Benefits and features

## Step 3- Provide a new Name for the Application

### 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

mywebapp2

Maximum length of 100 characters.

► Application tags (optional)

## Step 4- Choose the Platform as Python

**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- 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)

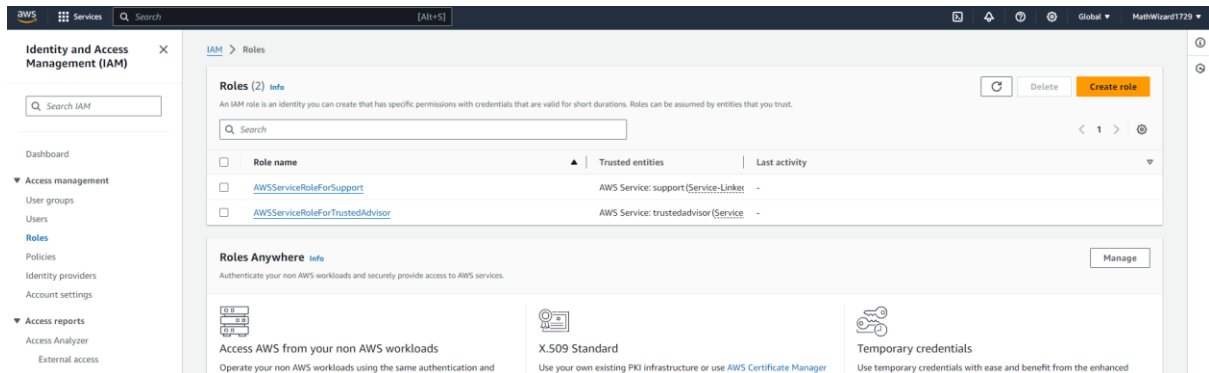
☐ High availability

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

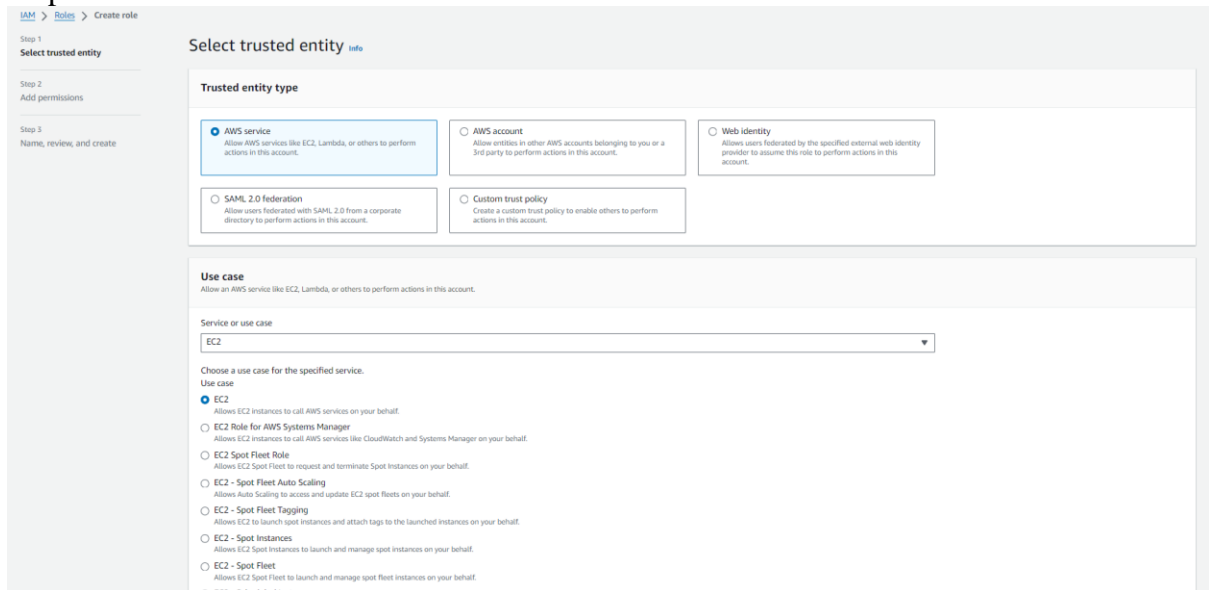
☐ Custom configuration

Cancel **Next**

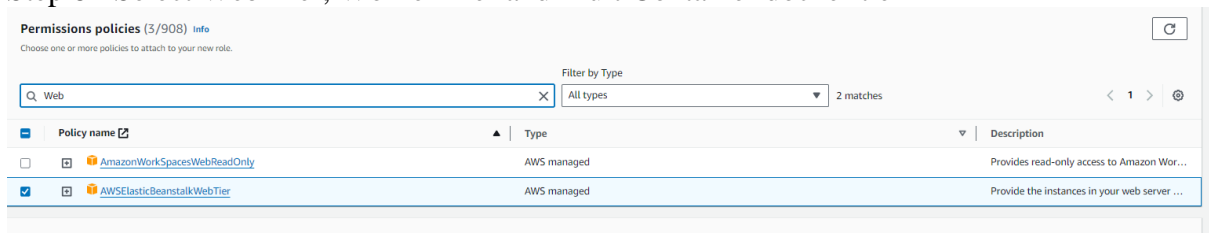
## Step 6- Create a new Role under IAM(Identity Access Management)



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



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



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

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








Q Worker

Filter by Type

All types

7 matches

< 1 >

<input checked="" type="checkbox"/>	Policy name <a href="#">↗</a>	Type	Description
<input type="checkbox"/>	 AmazonEKSWorkerNodePolicy	AWS managed	This policy allows Amazon EKS worker no...
<input type="checkbox"/>	 AmazonNimbleStudio-LaunchProfileWorker	AWS managed	This policy grants access to resources nee...
<input type="checkbox"/>	 AWSElasticBeanstalkRoleWorkerTier	AWS managed	(Elastic Beanstalk operations role) Allows ...
<input checked="" type="checkbox"/>	 AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instances in your worker envir...
<input type="checkbox"/>	 AWSThinkboxAWSPortalWorkerPolicy	AWS managed	This policy grants the Deadline Workers i...
<input type="checkbox"/>	 AWSThinkboxDeadlineSpotEventPluginWorkerPolicy	AWS managed	Grant permissions required for an EC2 ins...
<input type="checkbox"/>	 ROSAWorkerInstancePolicy	AWS managed	Allows Red Hat OpenShift Service on AW...

▼ Set permissions boundary - optional

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others. [Learn more about permission boundaries](#) [↗](#)

☒ Create role without a permissions boundary

☐ Use a permissions boundary to control the maximum role permissions

Cancel

Previous

Next

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

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


Q Docker

Filter by Type

All types

1 match

< 1 >

<input checked="" type="checkbox"/>	Policy name <a href="#">↗</a>	Type	Description
<input checked="" type="checkbox"/>	 AWSElasticBeanstalkMulticontainerDocker	AWS managed	Provide the instances in your multicontai...

▼ Set permissions boundary - optional

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting but can be used to delegate permission management to others. [Learn more about permission boundaries](#) [↗](#)

☒ Create role without a permissions boundary

☐ Use a permissions boundary to control the maximum role permissions

Cancel

Previous

Next

## Step 9- Provide a Name for Webapp Role

Role details

Role name

Enter a meaningful name to identify this role.

mywebapprole2

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 1: Select trusted entities

Edit

Trust policy

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ],
9       "Principal": {
10        "Service": [
11          "ec2.amazonaws.com"
12        ]
13      }
14    ]
15  }
16 }
```

Step 2: Add permissions

Edit

## Step 10- role was Created Successfully

Role mywebapprole2 created.

View role

IAM > Roles

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.


< 1 > ⌂

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	<a href="#">AWSServiceRoleForSupport</a>	AWS Service: support (Service-Linker)	-
<input type="checkbox"/>	<a href="#">AWSServiceRoleForTrustedAdvisor</a>	AWS Service: trustedadvisor (Service-Linker)	-
<input type="checkbox"/>	<a href="#">mywebapprole</a>	AWS Service: ec2	-
<input type="checkbox"/>	<a href="#">mywebapprole2</a>	AWS Service: ec2	-

Roles Anywhere Info


Authenticate your non AWS workloads and securely provide access to AWS services.

Manage




Access AWS from your non AWS workloads

Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.



X.509 Standard

Use your own existing PKI infrastructure or use [AWS Certificate Manager Private Certificate Authority](#) to authenticate identities.



Temporary credentials

Use temporary credentials with ease and benefit from the enhanced security they provide.

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

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

☒ 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](#)

Choose a key pair

EC2 instance profile

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

mywebapprole2

View permission details

Cancel

Skip to review

Previous

Next

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 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](#)

## 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-01fd2522c02ba4c6d | (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

Q Filter instance subnets

	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	eu-west-2a	subnet-034d212a9...	172.31.16.0/20	
<input type="checkbox"/>	eu-west-2b	subnet-099d4fc04...	172.31.32.0/20	
<input checked="" type="checkbox"/>	eu-west-2c	subnet-0e60dee54...	172.31.0.0/20	

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

#### Capacity rebalancing

Specifies whether to enable the capacity rebalancing feature for Spot Instances in your Auto Scaling Group. This option is only relevant when EnableSpot is true in the aws:ec2:instances namespace, and there is at least one Spot Instance in your Auto Scaling group.

☐ Turn on capacity rebalancing

#### Architecture

The processor architecture determines the instance types that are made available. You can't change this selection after you create the environment. [Learn more](#)

☒ x86\_64

This architecture uses x86 processors and is compatible with most third-party tools and libraries.

☐ arm64 - new

This architecture uses AWS Graviton2 processors. You might have to recompile some third-party tools and libraries.

#### Instance types

Add instance types for your fleet. Change the order that the instances are in to set the preferred launch order. This only affects On-Demand instances. We recommend you include at least two instance types. [Learn more](#)

Choose x86 instance types ▼

t3.micro ✕

t3.small ✕

#### AMI ID

Elastic Beanstalk selects a default Amazon Machine Image (AMI) for your environment based on the Region, platform version, and processor architecture that you choose. [Learn more](#)

ami-024a6e077a36b2855

#### Availability Zones

Number of Availability Zones (AZs) to use.

Any ▼

#### Placement

Specify Availability Zones (AZs) to use.

Choose Availability Zones (AZs) ▼

#### Scaling cooldown

360

seconds

Cancel

Skip to review

Previous

Next



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

## Review [Info](#)

### Step 1: Configure environment [Edit](#)

#### Environment information

Environment tier	Application name
Web server environment	mywebapp2
Environment name	Application code
Mywebapp2-env	Sample application
Platform	
arn:aws:elasticbeanstalk:eu-west-2::platform/Python 3.11 running on 64bit Amazon Linux 2023/4.0.7	

### Step 2: Configure service access [Edit](#)

#### Service access [Info](#)

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 instance profile
arn:aws:iam::654654340136:role/service-role/aws-elasticbeanstalk-service-role	mywebapprole2

### Step 3: Set up networking, database, and tags [Edit](#)

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

aws

Services

IAM

Elastic Beanstalk

Applications

Environments

Change history

▼ Application: mywebapp2

Application versions

Saved configurations

▼ Environment: Mywebapp2-env

Go to environment

Configuration

Events

Health

Logs

Monitoring

Alarms

Managed updates

Tags

▼ Recent environments

Mywebapp2-env

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

Elastic Beanstalk > Environments > Mywebapp2-env

Mywebapp2-env [Info](#)

↻

Actions

Upload and deploy

Change version

Environment overview

Health	Environment ID
Unknown	e-nvgrdfpsaj
Domain	Application name
-	mywebapp2

Platform

Change version

Platform	
Python 3.11 running on 64bit Amazon Linux 2023/4.0.7	
Running version	Platform state
-	Supported

Events

Health

Logs

Monitoring

Alarms

Managed updates

Tags

Events (2) [Info](#)

Q

Filter events by text, property or value

◀ 1 ▶

⌕

Time	Type	Details
January 23, 2024 08:38:06 (UTC+5:30)	INFO	Using elasticbeanstalk-eu-west-2-654654340136 as Amazon S3 storage bucket for environment data.
January 23, 2024 08:38:05 (UTC+5:30)	INFO	createEnvironment is starting.

## Step 16- Environment is successfully launched

The screenshot shows the AWS Elastic Beanstalk console. At the top, a green banner states "Environment successfully launched." The left sidebar shows the navigation menu with "Elastic Beanstalk" selected. The main content area displays the "Mywebapp2-env" environment details. The "Environment overview" section shows the health status as "Ok", the environment ID as "e-nvgrdfpsaj", the domain as "Mywebapp2-env.eba-zpzmqv5a.eu-west-2.elasticbeanstalk.com", and the application name as "mywebapp2". The "Platform" section shows the platform as "Python 3.11 running on 64bit Amazon Linux 2023/4.0.7" and the platform state as "Supported". Below these sections, there are tabs for "Events", "Health", "Logs", "Monitoring", "Alarms", "Managed updates", and "Tags". The "Events" tab is selected, showing a list of 12 events. The events list includes the following details:

Time	Type	Details
January 23, 2024 08:40:50 (UTC+5:30)	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 7 seconds ago and took 2 minutes.
January 23, 2024 08:40:01 (UTC+5:30)	INFO	Successfully launched environment: Mywebapp2-env
January 23, 2024 08:40:00 (UTC+5:30)	INFO	Application available at Mywebapp2-env.eba-zpzmqv5a.eu-west-2.elasticbeanstalk.com.
January 23, 2024 08:39:50 (UTC+5:30)	INFO	Added instance [i-0c1d9b0bd3668eea1] to your environment.
January 23, 2024 08:39:45 (UTC+5:30)	INFO	Instance deployment completed successfully.

## 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

The screenshot shows the "Configure environment" page in the AWS Elastic Beanstalk console. The page has a header "Configure environment" with an "Info" link. Below the header, there are two main sections: "Environment tier" and "Application information".

**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

mywebapp


Maximum length of 100 characters.

► **Application tags (optional)**

#### Step 4- Select Platform as Java and go on further

**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-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

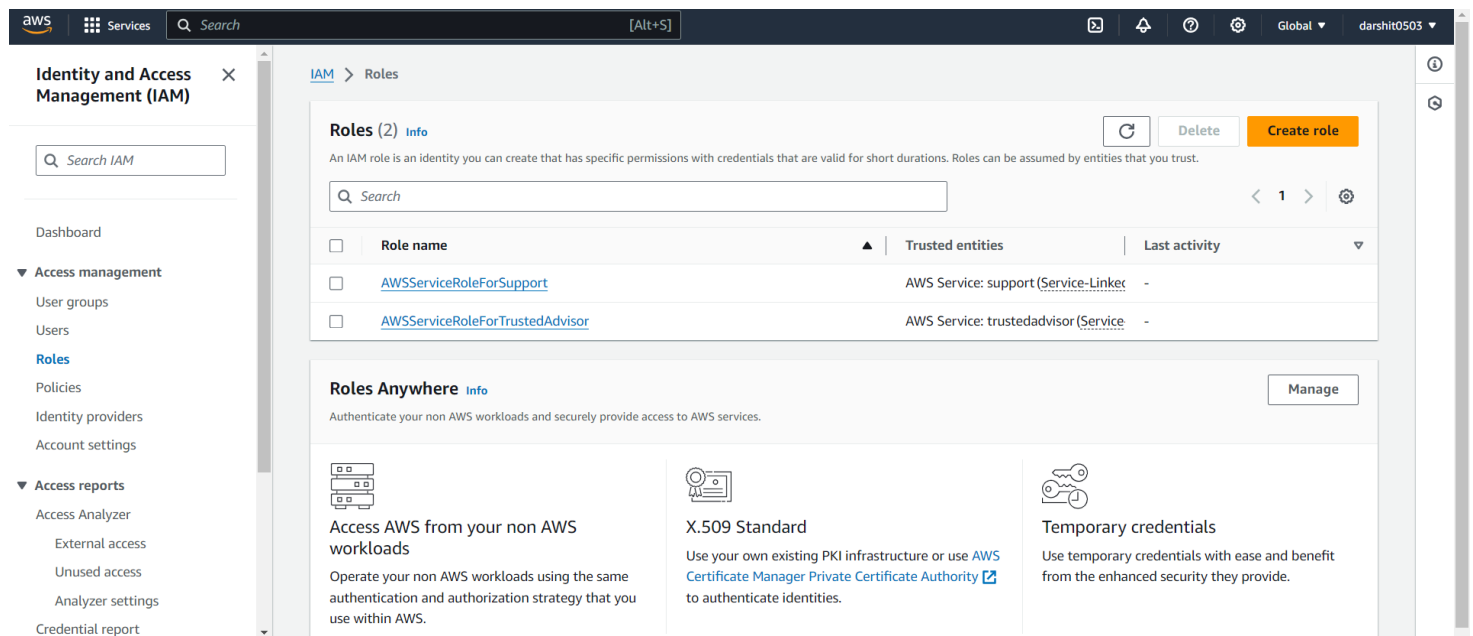
☐ High availability (using spot and on-demand instances)

☐ Custom configuration

Cancel

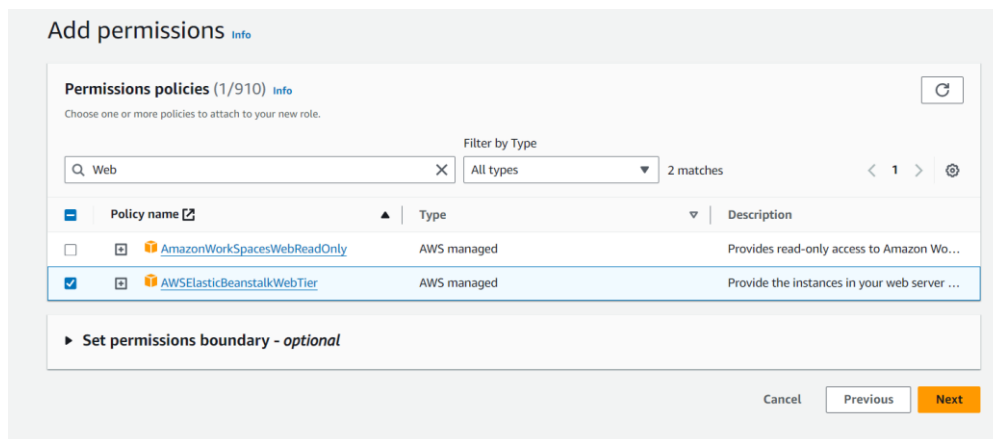
Next

## Step 6- Create Role under IAM User

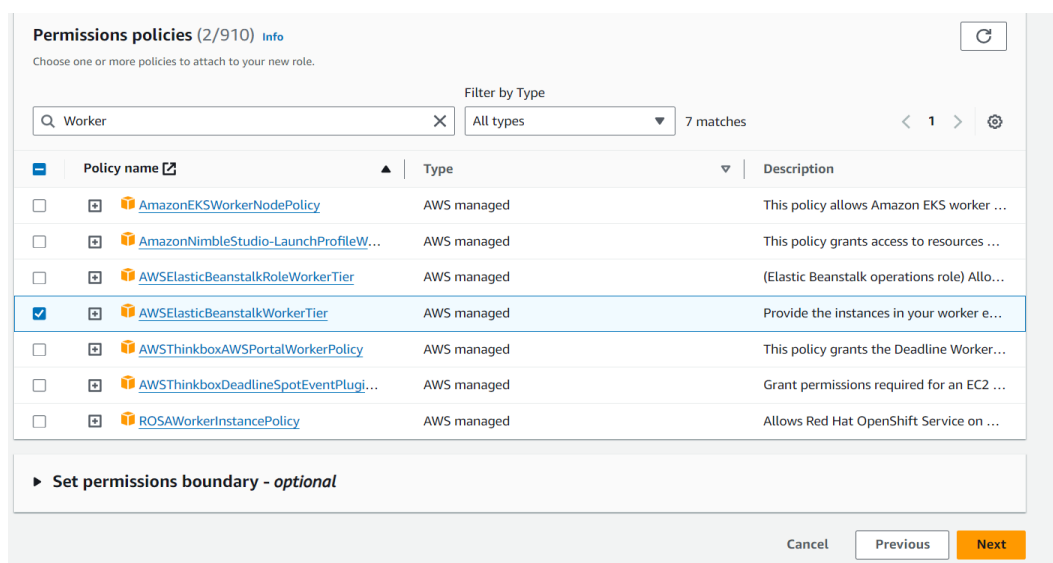


## Step 7- Select Usecase as EC2 and select Next and from the following list Select3 options

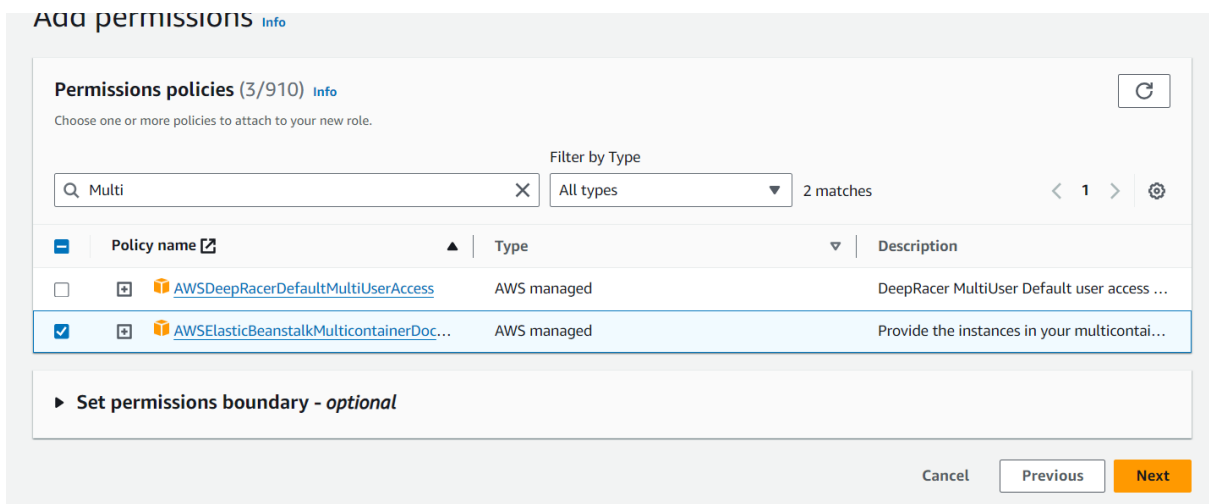
a) WebTier



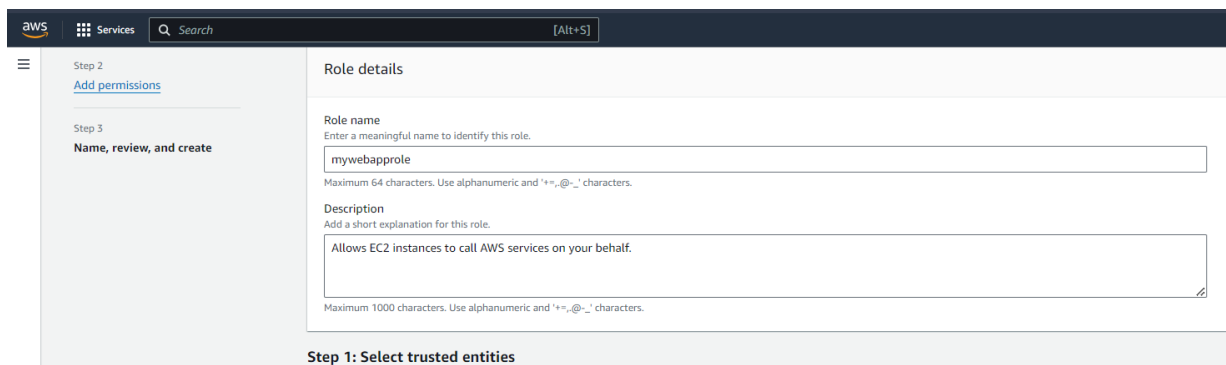
### b) WorkerTier



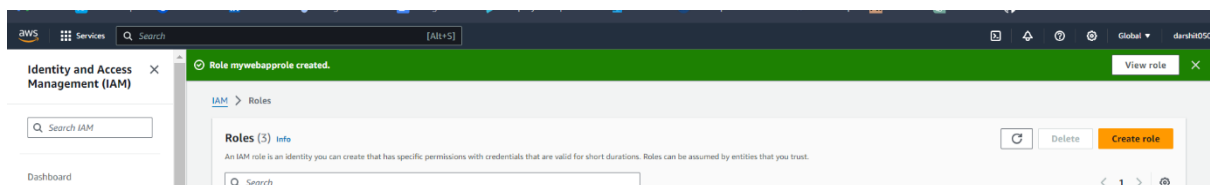
### c) MultiContainerDocker list



## 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

The screenshot shows the AWS IAM console interface for the 'Configure service access' step. The left sidebar contains a navigation menu with steps 1 through 6. Step 2, 'Configure service access', is the active step. The main content area is titled 'Configure service access' and includes an 'Info' link. It contains three sections: 'Service access', 'EC2 key pair', and 'EC2 instance profile'. In the 'Service access' section, the 'Create and use new service role' radio button is selected, and the 'Service role name' is 'aws-elasticbeanstalk-service-role'. In the 'EC2 key pair' section, a dropdown menu shows 'Choose a key pair'. In the 'EC2 instance profile' section, a dropdown menu shows 'mywebapprole'. Each section has a 'View permission details' button.

aws Services IAM

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**

☒ 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

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

mywebapprole

[View permission details](#)

Step 11- Select VPC which was been provided there

The screenshot shows the AWS IAM console interface for the 'Set up networking, database, and tags - optional' step. The left sidebar contains a navigation menu with steps 1 through 6. Step 3, 'Set up networking, database, and tags', is the active step. The main content area is titled 'Set up networking, database, and tags - optional' and includes an 'Info' link. It contains two sections: 'Virtual Private Cloud (VPC)' and 'Instance settings'. In the 'VPC' section, a dropdown menu shows 'vpc-046e1bf9a6c2d16fe | (172.31.0.0/16)'. In the 'Instance settings' section, the 'Public IP address' checkbox is unchecked. Each section has a 'Create custom VPC' link.

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](#)

### 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-046e1bf9a6c2d16fe | (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

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

### 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-00b3bd62c...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1c	subnet-04c6c12da...	172.31.16.0/20	
<input type="checkbox"/>	ap-south-1a	subnet-0be694f3f...	172.31.32.0/20	

For Database

### 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-00b3bd62c...	172.31.0.0/20	
<input type="checkbox"/>	ap-south-1c	subnet-04c6c12da...	172.31.16.0/20	
<input type="checkbox"/>	ap-south-1a	subnet-0be694f3f...	172.31.32.0/20	

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

☐ arm64 - new

This architecture uses AWS Graviton2 processors. You might have to recompile some third-party tools and libraries.

Instance types

Add instance types for your fleet. Change the order that the instances are in to set the preferred launch order. This only affects On-Demand instances. We recommend you include at least two instance types. [Learn more](#)

Choose x86 instance types

t3.micro t3.small

AMI ID

Elastic Beanstalk selects a default Amazon Machine Image (AMI) for your environment based on the Region, platform version, and processor architecture that you choose. [Learn more](#)

ami-0cf0e9d2d27dbae48

Availability Zones

Number of Availability Zones (AZs) to use.

Any

Placement

Specify Availability Zones (AZs) to use.

Choose Availability Zones (AZs)

Scaling cooldown

360 seconds

Cancel

Skip to review

Previous

Next

## Step 14- Submit the Review

Activated	100	Percentage								
Command timeout	Deployment policy	Health threshold								
600	AllAtOnce	Ok								
Ignore health check	Instance replacement									
false	false									
Platform software										
Lifecycle	Log streaming	Logs retention								
false	Deactivated	7								
Rotate logs	Update level	X-Ray enabled								
Deactivated	minor	Deactivated								
Environment properties										
<table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>GRADLE_HOME</td><td>/usr/local/gradle</td></tr><tr><td>M2</td><td>/usr/local/apache-maven/bin</td></tr><tr><td>M2_HOME</td><td>/usr/local/apache-maven</td></tr></tbody></table>			Key	Value	GRADLE_HOME	/usr/local/gradle	M2	/usr/local/apache-maven/bin	M2_HOME	/usr/local/apache-maven
Key	Value									
GRADLE_HOME	/usr/local/gradle									
M2	/usr/local/apache-maven/bin									
M2_HOME	/usr/local/apache-maven									

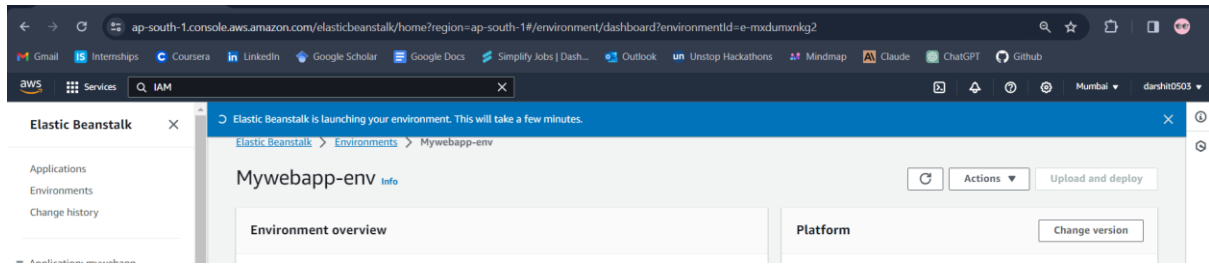
Cancel

Previous

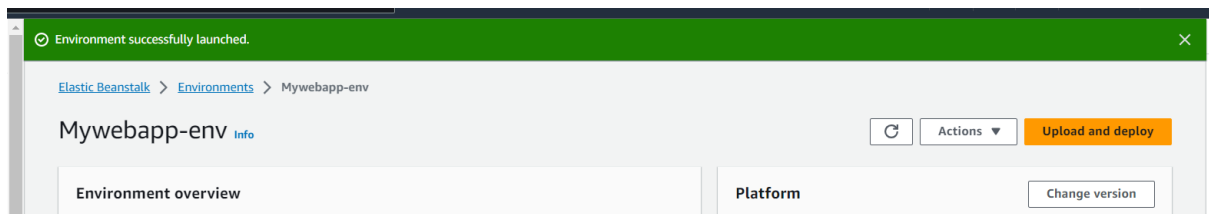
Submit



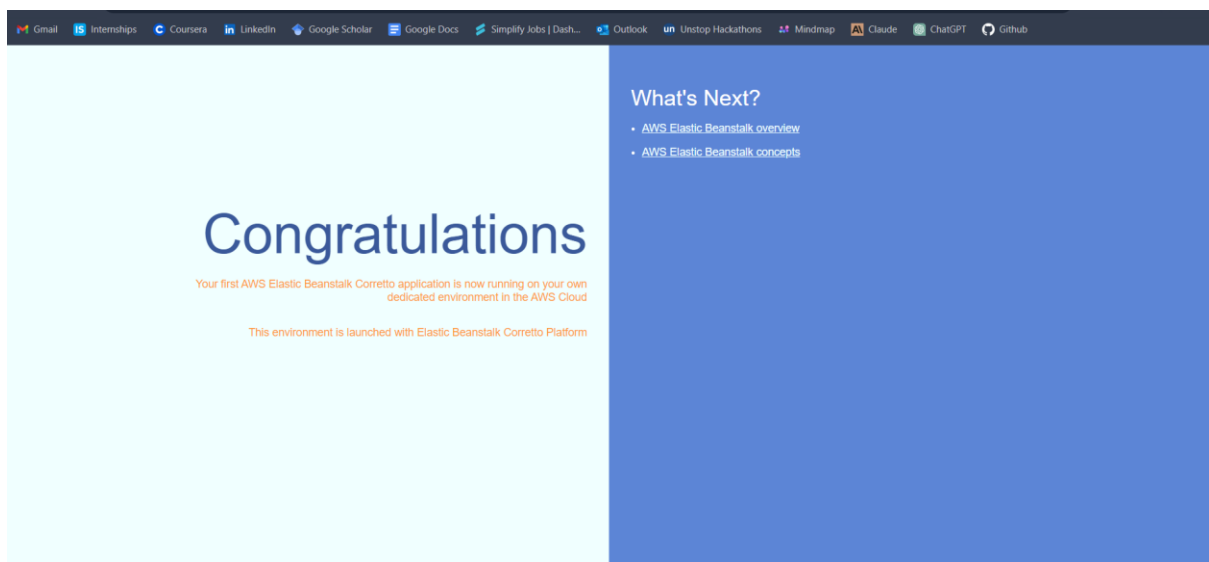
Step 15- After the review the Elastic BeanStalk will run environment and it will take some few minutes



Step 16- Environment runs successfully



Step 17- Output of the following



Step 18- Terminate the following Instance

