

SVKM'S NMIMS 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 (PaaS):**

Platform as a Service (PaaS) is a category of cloud computing services that provides a runtime environment for web applications. It allows developers to create, test, run, and manage applications without the complexity of building and maintaining the infrastructure typically associated with these tasks. PaaS includes infrastructure (servers, storage, and networking) and platform (middleware, development tools, database management systems, business intelligence, and more) to support the web application life cycle. Examples of PaaS providers include Google App Engine, Force.com, Joyent, and Azure.

PaaS offers several advantages such as simplified development, lower risk, prebuilt business functionality, instant community, and scalability. It allows developers to focus on development and innovation without worrying about infrastructure management. However, there are also disadvantages such as vendor lock-in and data privacy concerns. One has to write the applications according to the platform provided by the PaaS vendor, so the migration of an application to another PaaS vendor could be a problem. Corporate data, whether it can be critical or not, will be private, so if it is not located within the walls of the company, there can be a risk in terms of privacy of data.

- **Elastic Beanstalk:**

AWS Elastic Beanstalk is an orchestration service offered by Amazon Web Services for deploying applications. It allows developers to quickly deploy and manage applications in the AWS Cloud without having to learn about the infrastructure that runs those applications. Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring. It supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby. When you deploy your application, Elastic Beanstalk builds the selected supported platform version and provides one or more AWS resources, such as Amazon EC2 instances, to run your application.

Why Elastic Beanstalk -

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

- **Components of Beanstalk:**

AWS Elastic Beanstalk consists of several key components:

1. **Application:** A logical collection of Elastic Beanstalk components, including environments, versions, and environment configurations.
2. **Application Version:** Refers to a specific, labeled iteration of deployable code for a web application.
3. **Environment:** A collection of AWS resources running an application version.
4. **Environment Tier:** Designates the type of application that the environment runs, and determines what resources Elastic Beanstalk provisions to support it.
5. **Environment Configuration:** Identifies a collection of parameters and settings that define how an environment and its associated resources behave.
6. **Saved Configuration:** A template that you can use as a starting point for creating unique environment configurations.
7. **Platform:** A combination of an operating system, programming language runtime, web server, application server, and Elastic Beanstalk components.

- **IAM:**

Identity and Access Management (IAM) is a combination of policies, processes, and technologies that enable organizations to manage digital identities and control user access to critical corporate information. It is a vital part of modern IT as it manages access so that the right people can do their jobs and the wrong

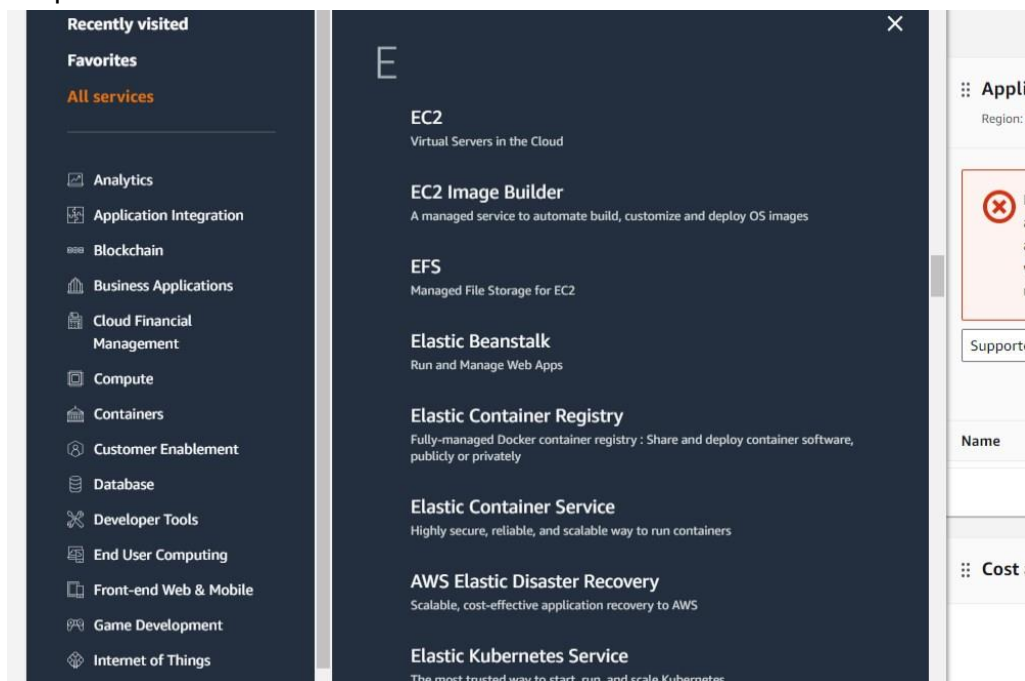
people, like hackers, are denied entry. IAM systems keep hackers out while ensuring that each individual user has the exact permissions they need to do their job. The components of IAM include Users, Roles, Groups, and Policies. IAM can be used for many purposes such as controlling access of individual and group access for your AWS resources. With IAM policies, managing permissions to your workforce and systems to ensure least-privilege permissions becomes easier. The AWS IAM is a global service.

- 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

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)

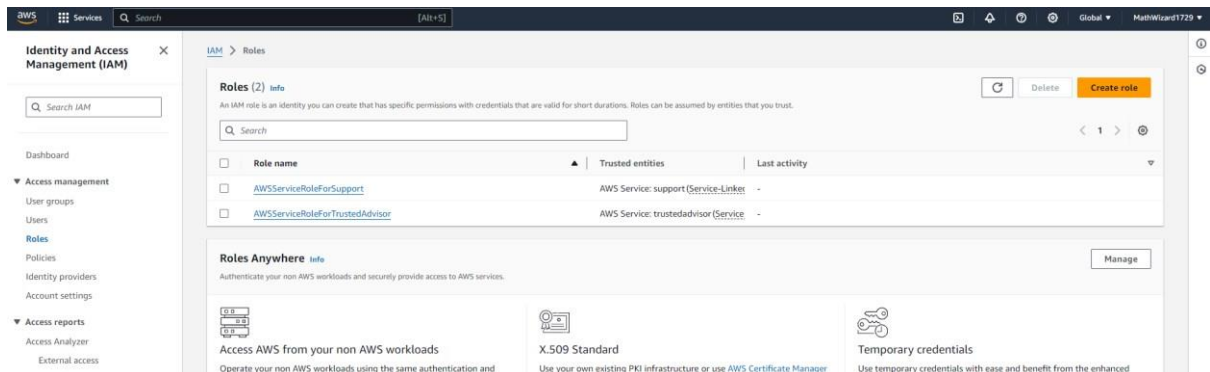
☐ High availability

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

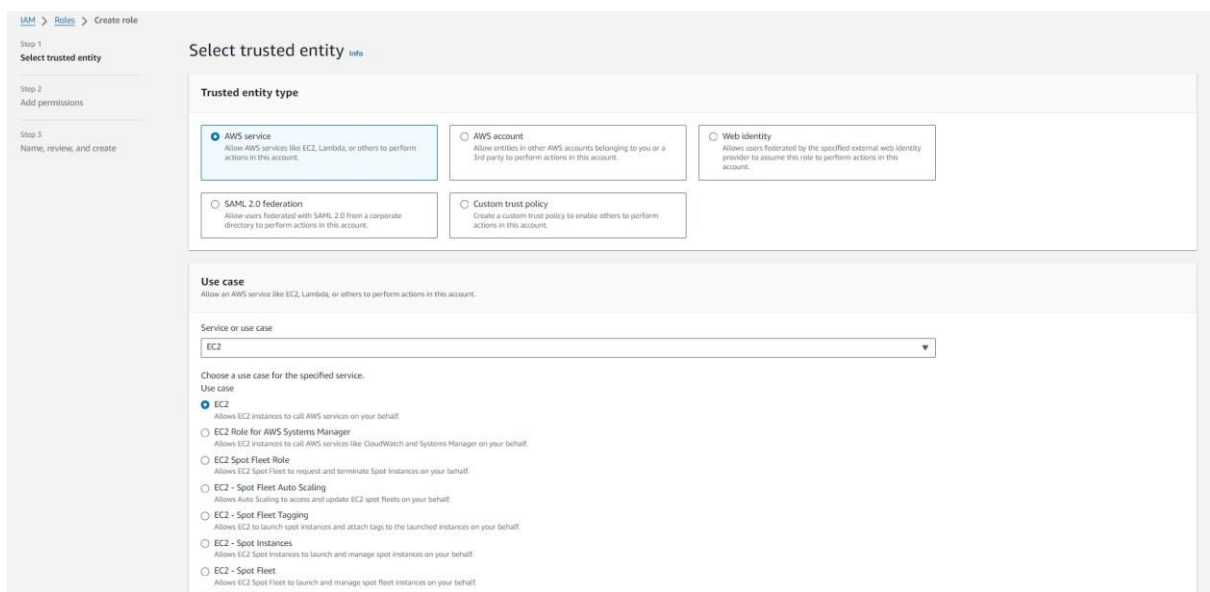
☐ Custom configuration

Cancel **Next**

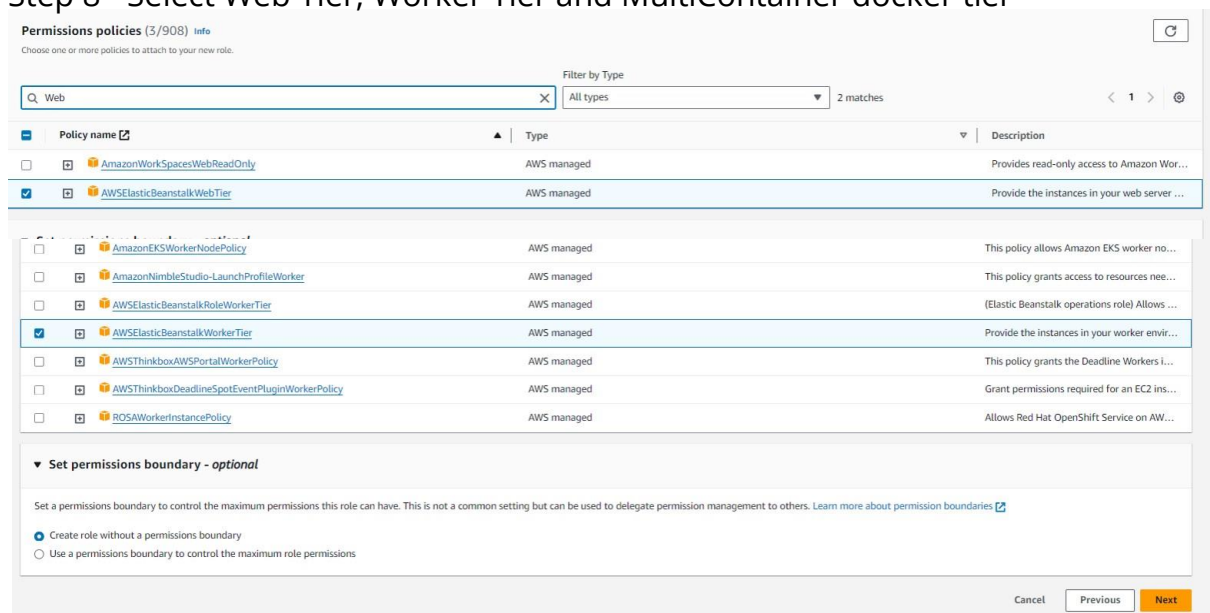
Step 6- Create a new Role under IAM(Identity Access Management)



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



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



Add permissions [Info](#)

Permissions policies (3/908) [Info](#)
Choose one or more policies to attach to your new role.

Filter by Type
All types 1 match

< 1 >

<input checked="" type="checkbox"/>	Policy name Info	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](#) [Info](#)
☒ Create role without a permissions boundary
☐ Use a permissions boundary to control the maximum role permissions

Step 9- Provide a Name for Webapp Role

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.

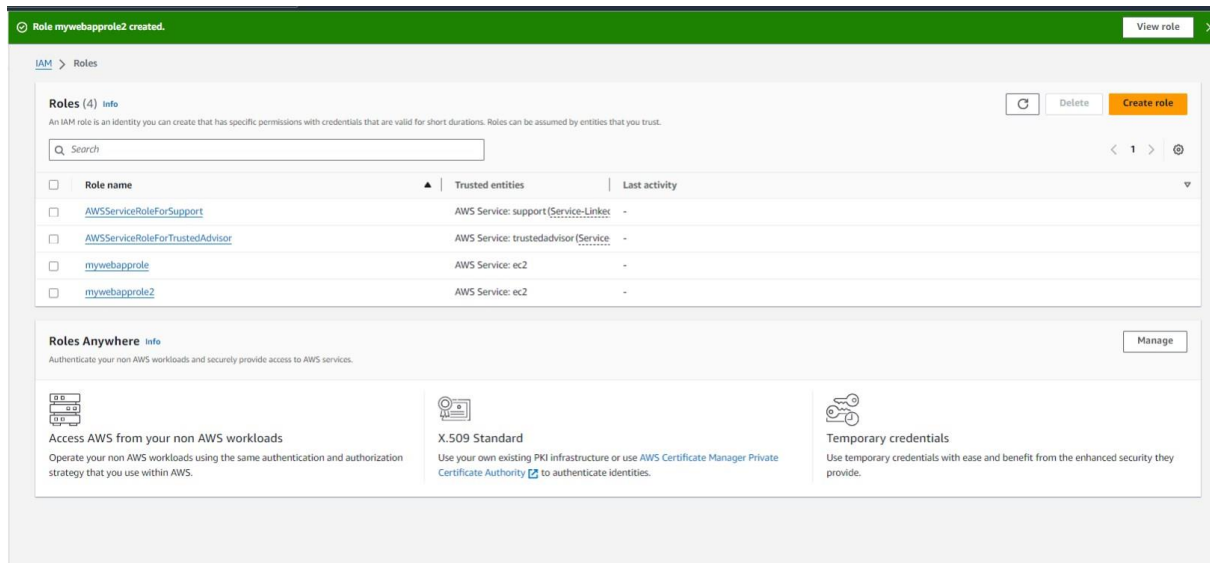
Step 1: Select trusted entities

Trust policy

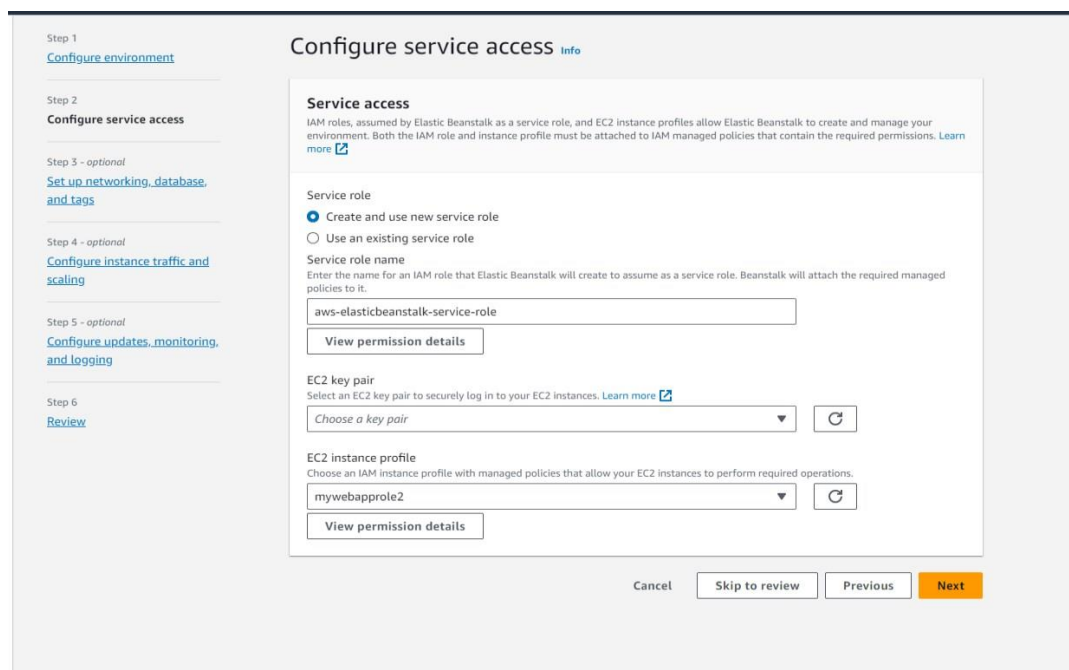
```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ],
9       "Principal": {
10        "Service": {
11          "ec2.amazonaws.com"
12        }
13      }
14    }
15  ]
16 }
```

Step 2: Add permissions

Step 10- role was Created Successfully



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



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*

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

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 X

t3.small X

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	
am: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
am: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

Q 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

Environment overview

Health

Unknown

Environment ID

e-nvgrdfpsaj

Domain

-

Application name

mywebapp2

Platform

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

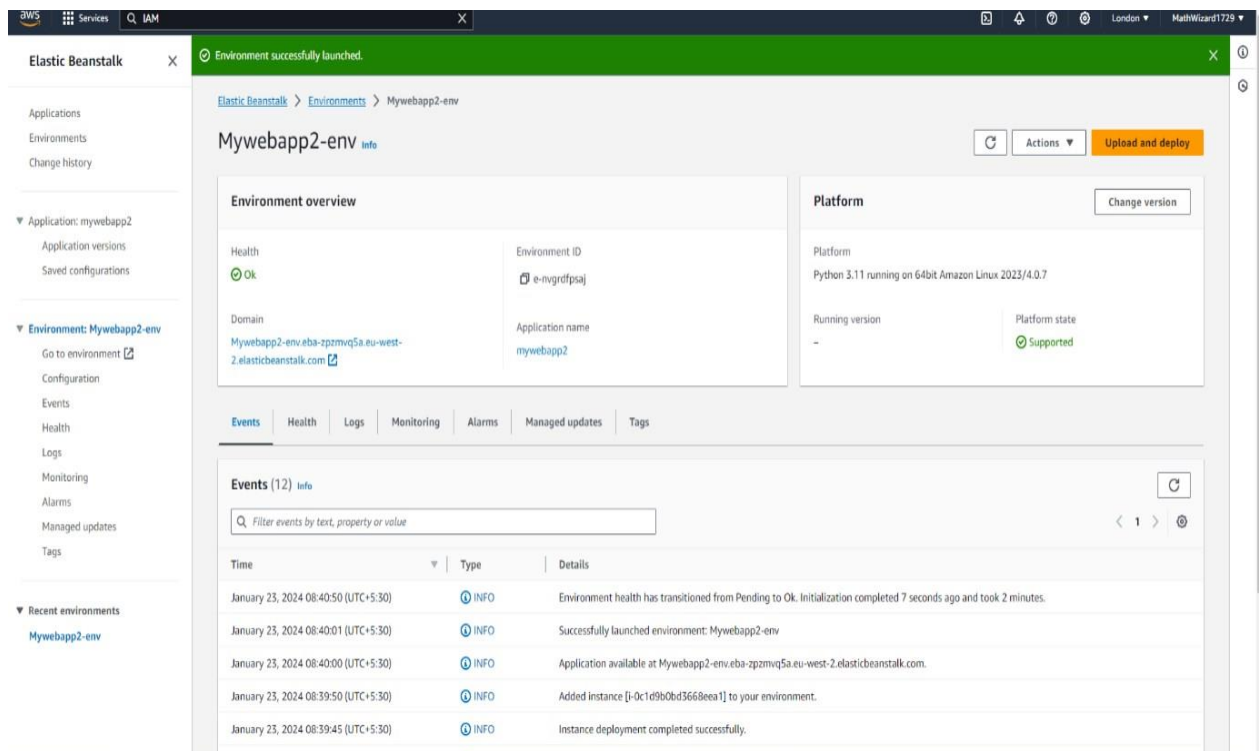
Events (2) Info

Filter events by text, property or value

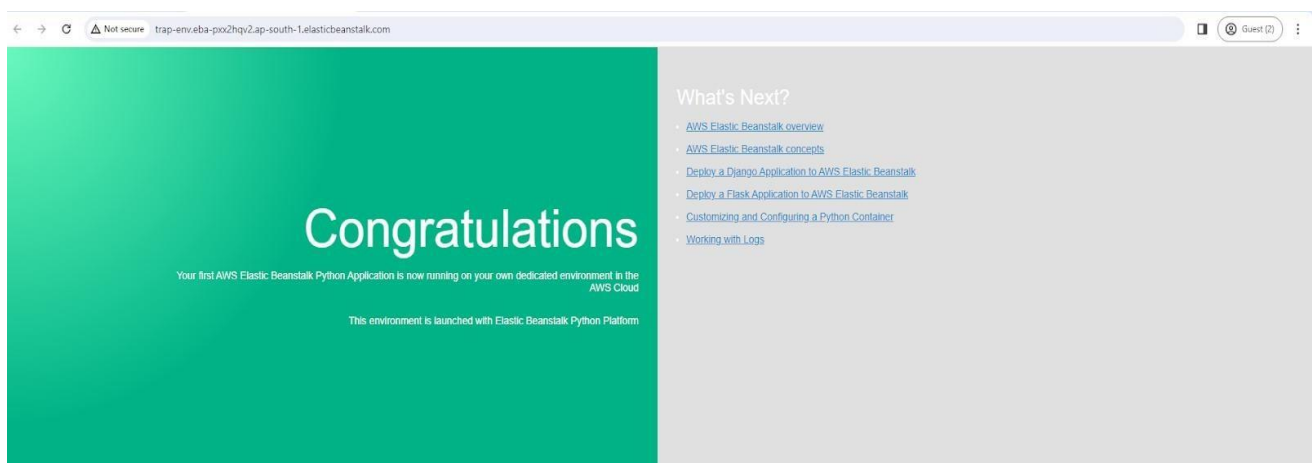
< 1 >

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



Step 17- The Output is shown as like this



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

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


Maximum length of 100 characters.

► Application tags (optional)

Step 4- Select Platform as Java and go on further

Platform [Info](#)

Platform type

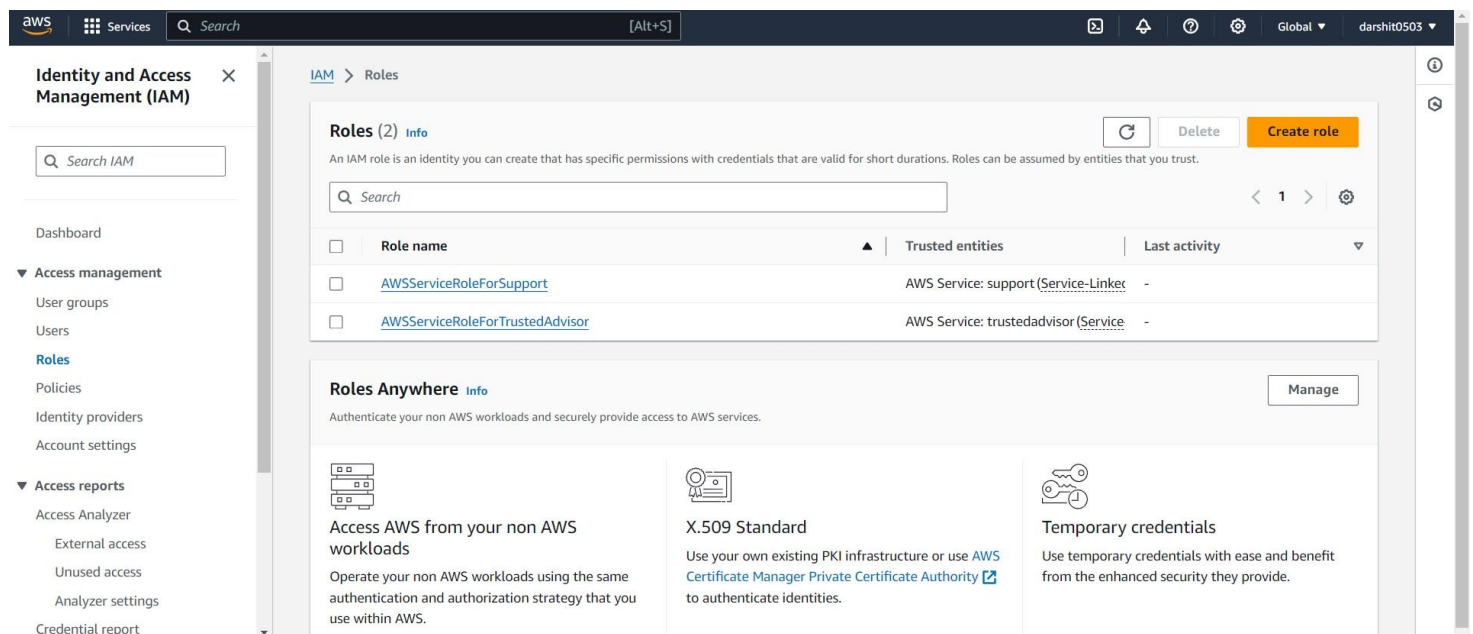
- ☒ **Managed platform**
Platforms published and maintained by Amazon Elastic Beanstalk. [Learn more](#) 
- ☐ **Custom platform**
Platforms created and owned by you. This option is unavailable if you have no platforms.

Platform

Platform branch

Platform version

Step 5-Keep it as Single Instance.



Step 7- Select Usecase as EC2 and select Next and from the following list
Select 3 options a) WebTier

Add permissions [Info](#)

Permissions policies (1/910) [Info](#)

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

Filter by Type

Q Web X All types 2 matches < 1 > ⚙

	Policy name ?	Type	Description
<input type="checkbox"/>	AmazonWorkSpacesWebReadOnly	AWS managed	Provides read-only access to Amazon Wo...
<input checked="" type="checkbox"/>	AWSElasticBeanstalkWebTier	AWS managed	Provide the instances in your web server ...

► Set permissions boundary - optional

Cancel Previous Next

b) WorkerTier

Permissions policies (2/910) [Info](#)

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

Filter by Type

Q Worker X All types 7 matches < 1 > ⚙

	Policy name ?	Type	Description
<input type="checkbox"/>	AmazonEKSWorkerNodePolicy	AWS managed	This policy allows Amazon EKS worker ...
<input type="checkbox"/>	AmazonNimbleStudio-LaunchProfileW...	AWS managed	This policy grants access to resources ...
<input type="checkbox"/>	AWSElasticBeanstalkRoleWorkerTier	AWS managed	(Elastic Beanstalk operations role) Allo...
<input checked="" type="checkbox"/>	AWSElasticBeanstalkWorkerTier	AWS managed	Provide the instances in your worker e...
<input type="checkbox"/>	AWSThinkboxAWSPortalWorkerPolicy	AWS managed	This policy grants the Deadline Worker...
<input type="checkbox"/>	AWSThinkboxDeadlineSpotEventPlugi...	AWS managed	Grant permissions required for an EC2 ...
<input type="checkbox"/>	ROSAWorkerInstancePolicy	AWS managed	Allows Red Hat OpenShift Service on ...

► Set permissions boundary - optional

Cancel Previous Next

c) MultiContainerDocker list

Add permissions [Info](#)

Permissions policies (3/910) [Info](#)

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

Filter by Type

Q Multi X All types 2 matches < 1 > ⚙

	Policy name ?	Type	Description
<input type="checkbox"/>	AWSDeepRacerDefaultMultiUserAccess	AWS managed	DeepRacer MultiUser Default user access ...
<input checked="" type="checkbox"/>	AWSElasticBeanstalkMulticontainerDoc...	AWS managed	Provide the instances in your multicontai...

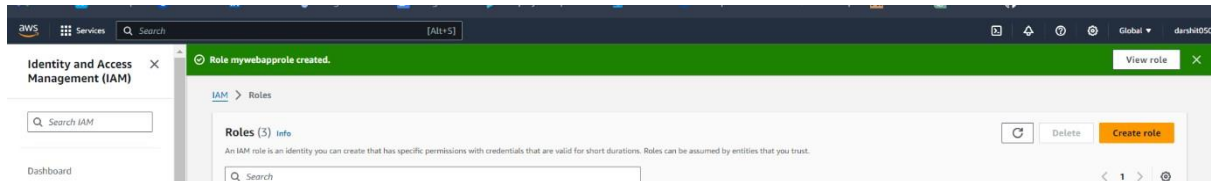
► Set permissions boundary - optional

Cancel Previous Next

Step 8 - Provide the name for the Role

The screenshot shows the AWS IAM console interface. On the left, a sidebar contains a menu with 'Step 2 Add permissions' and 'Step 3 Name, review, and create'. The main area is titled 'Role details'. It has two input fields: 'Role name' with the value 'mywebapprole' and 'Description' with the value 'Allows EC2 instances to call AWS services on your behalf.' Below these fields, a status bar indicates 'Step 1: Select trusted entities'.

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 displays the 'Configure service access' page in the AWS IAM console. The left sidebar shows a sequence of steps: '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', and 'Step 6 Review'. The main area is titled 'Configure service access'. It contains sections for 'Service access', 'Service role' (with 'Create and use new service role' selected), 'Service role name' (set to 'aws-elasticbeanstalk-service-role'), 'EC2 key pair' (with a 'Choose a key pair' dropdown), and 'EC2 instance profile' (set to 'mywebapprole'). Each section has a 'View permission details' button.

Step 11- Select VPC which was been provided there

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

🔍 Filter 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)

<input checked="" type="checkbox"/>	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](#)

t3.micro X

t3.small X

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](#)

Availability Zones

Number of Availability Zones (AZs) to use.

Placement

Specify Availability Zones (AZs) to use.

Scaling cooldown

seconds

Cancel

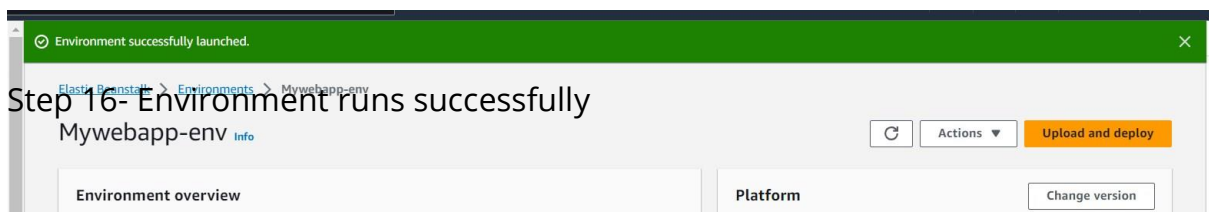
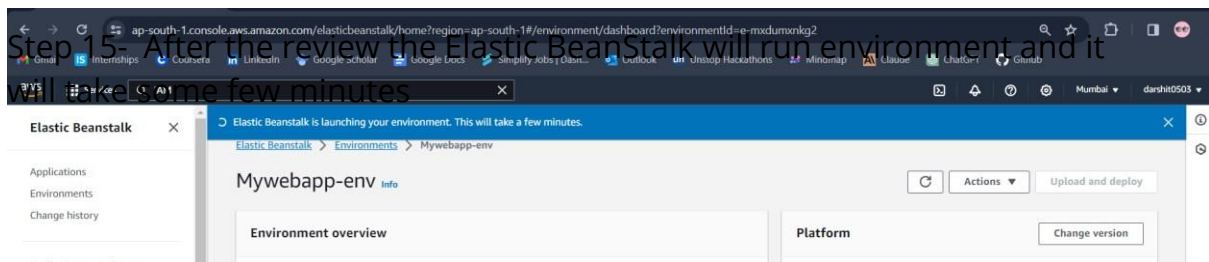
Skip to review

Previous

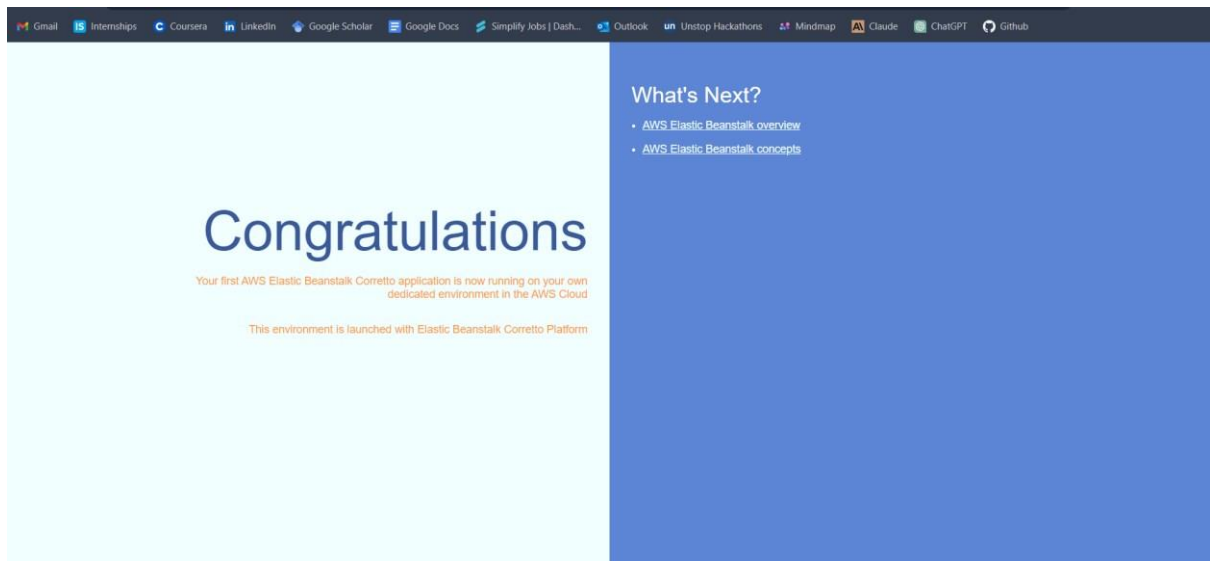
Next

Step 14- Submit the Review

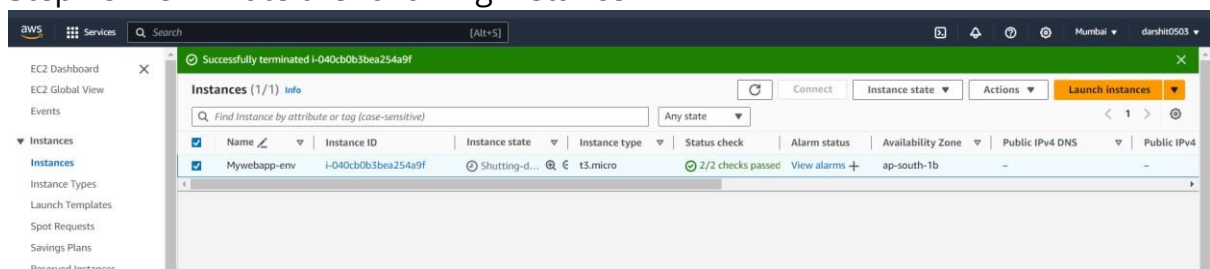
Activated	100	Percentage								
600	Deployment policy	Health threshold								
	AllAtOnce	Ok								
Ignore health check	Instance replacement									
false	false									
Platform software										
Lifecycle	Log streaming	Logs retention								
false	Deactivated	7								
Rotate logs	Update level	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									
<div>Cancel Previous Submit</div>										



Step 17- Output of the following

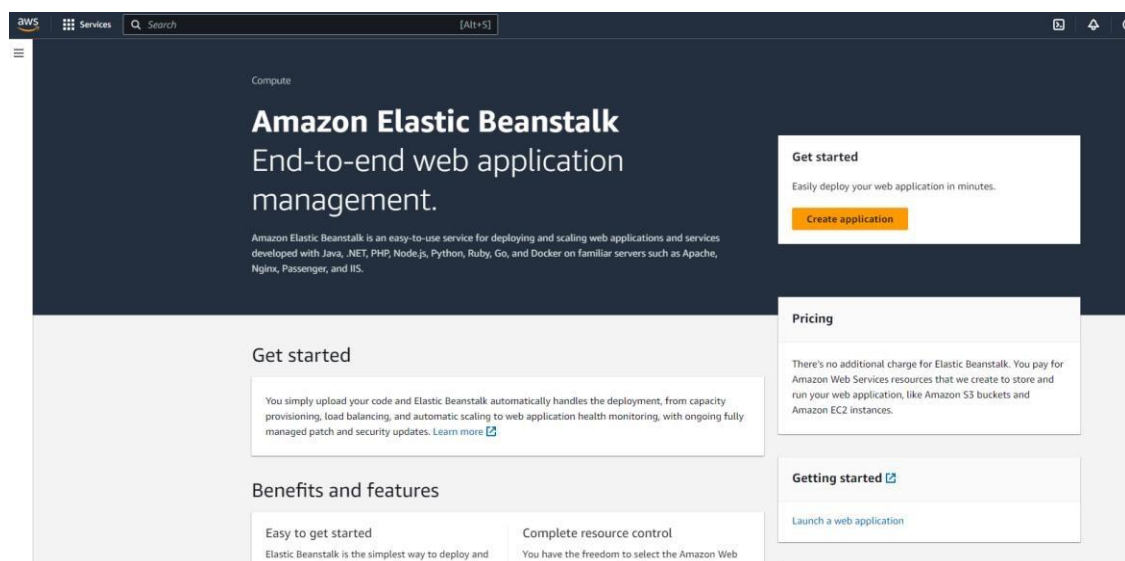


Step 18- Terminate the following Instance



FOR TOMCAT

1. Go to your Elastic Beanstalk



2. Provide the name of the Application

aws Services Search [Alt+S]

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
TomCat
Maximum length of 100 characters.

► Application tags (optional)

Environment information [Info](#)
Choose the name, subdomain and description for your environment. These cannot be changed later.

Environment name
TomCat-env

3. Choose the Platform as TomCat

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
Tomcat

Platform branch
Tomcat 10 with Corretto 17 running on 64bit Amazon Linux 2023

Platform version
5.1.3 (Recommended)

Application code [Info](#)

4. Download “calendar.war” File and Upload the Code

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.

Version label

Source code origin. Maximum size 500 MB

☒ Local file

Upload application

Choose file

☒ File name: **Calendar.war**
File must be less than 500MB max file size

☐ Public S3 URL

5. Create a New service Role and use the existing EC2 Profile

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

Cancel

Skip to review

Previous

Next

6. Use the VPC and select the Same IP Address in Instance and 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-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

Instance subnets

Filter 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	

7. Skip to the Review Section and Submit it and you will see the Environment will start running

The screenshot shows the AWS Elastic Beanstalk console. The left sidebar has a navigation menu with 'Applications', 'Environments', and 'Change history'. Under 'Environments', 'TomCat-env' is selected. The main content area shows the 'TomCat-env' environment overview. It includes a 'Health' section with 'Unknown' status, a 'Domain' section with a '-' sign, and a 'Platform' section with 'Tomcat 10 with Corretto 17 running on 64bit Amazon Linux 2023/5.1.3'. The 'Running version' is '-' and the 'Platform state' is 'Supported'. The 'Events' tab is selected, showing a list of events.

GWT Calendar

8. The Output is seen as follows

Click on day to get date popup. Example Datepicker. Built with the tomcat war builder. <http://code.google.com/p/gwt-examples/>

< January >						
< 2024 >						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			