

Terna Engineering College

Computer Engineering Department

Program: Sem VI

Course: Cloud Computing Lab (CSL603)

Faculty: PREETI PATIL

Experiment No.4

A.1 Aim:

To study and Implement Platform as a Service using AWS Elastic Beanstalk/ Microsoft Azure App Service.

A.2 Prerequisite:

Understanding of Virtualization, Basics of Networking, Basics of security and privacy.

A.3 Objective:

To demonstrate the steps to deploy Web applications or Web services written in different languages on AWS Elastic Beanstalk/ Microsoft Azure App Service.

A.3 Outcome: (LO2)

After successful completion of this experiment students will be able to deploy the web application using AWS Elastic Beanstalk.

A.4 Theory:

AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with popular programming languages such as Java, .NET, PHP, Node.js, Python and Ruby. You simply upload your application and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling and application health monitoring. At the same time, with Elastic Beanstalk, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

Most existing application containers or platform-as-a-service solutions, while reducing the amount of programming required, significantly diminish developers' flexibility and control. Developers are forced to live with all the decisions pre-determined by the vendor - with little to no opportunity to take back control over various parts of their application's infrastructure. However, with Elastic Beanstalk, you retain full control over the AWS resources powering your application. If you decide you want to take over some (or all) of

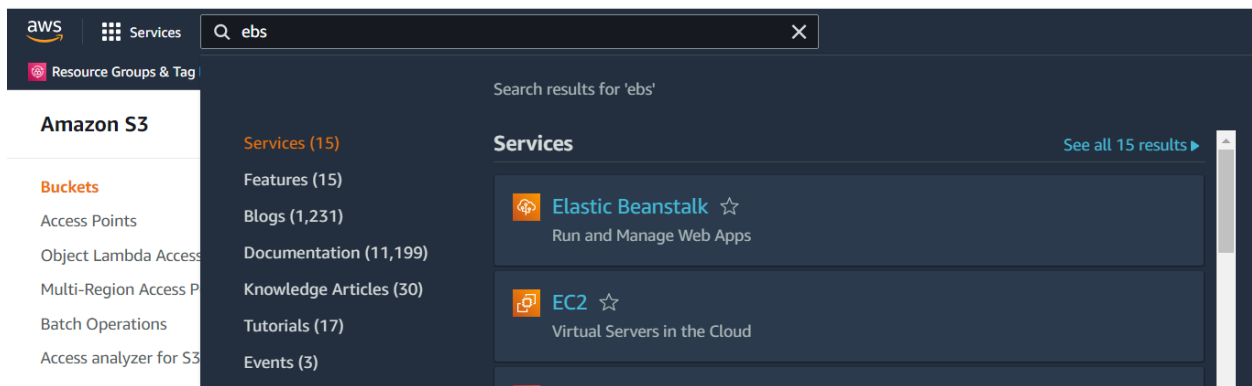
the elements of their infrastructure, you can do so seamlessly by using Elastic Beanstalk's management capabilities.

To ensure easy portability of your application, Elastic Beanstalk is built using familiar application/web servers such as Apache HTTP Server, Apache Tomcat, Nginx, Passenger and IIS 7.5/8.

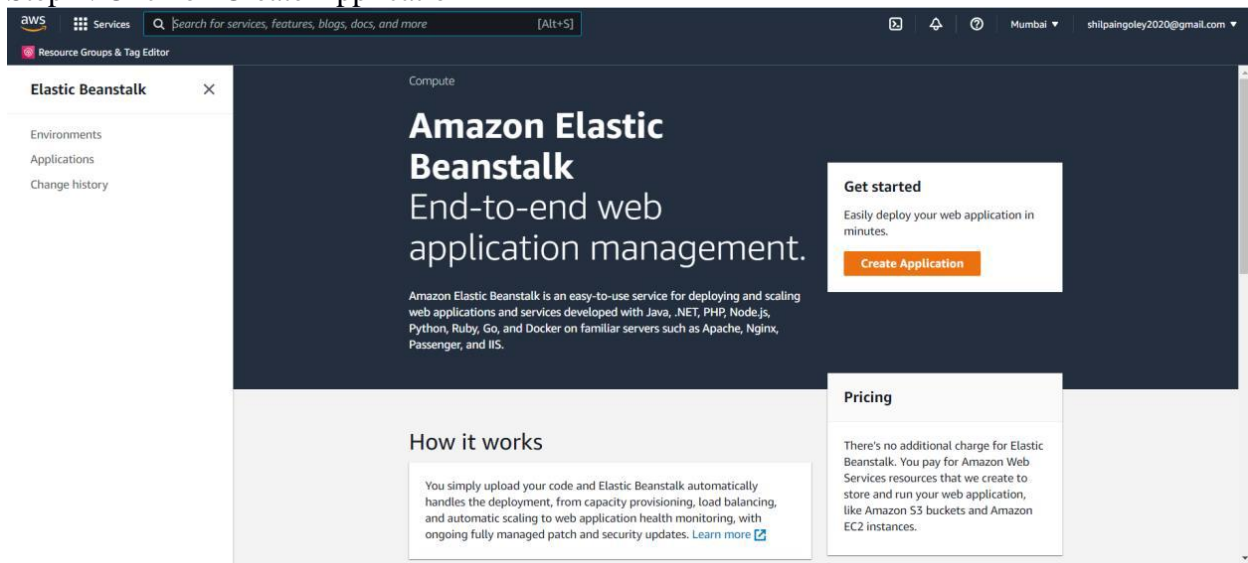
Following are steps to host a website using Elastic Beanstalk:

(Creating Sample Application)

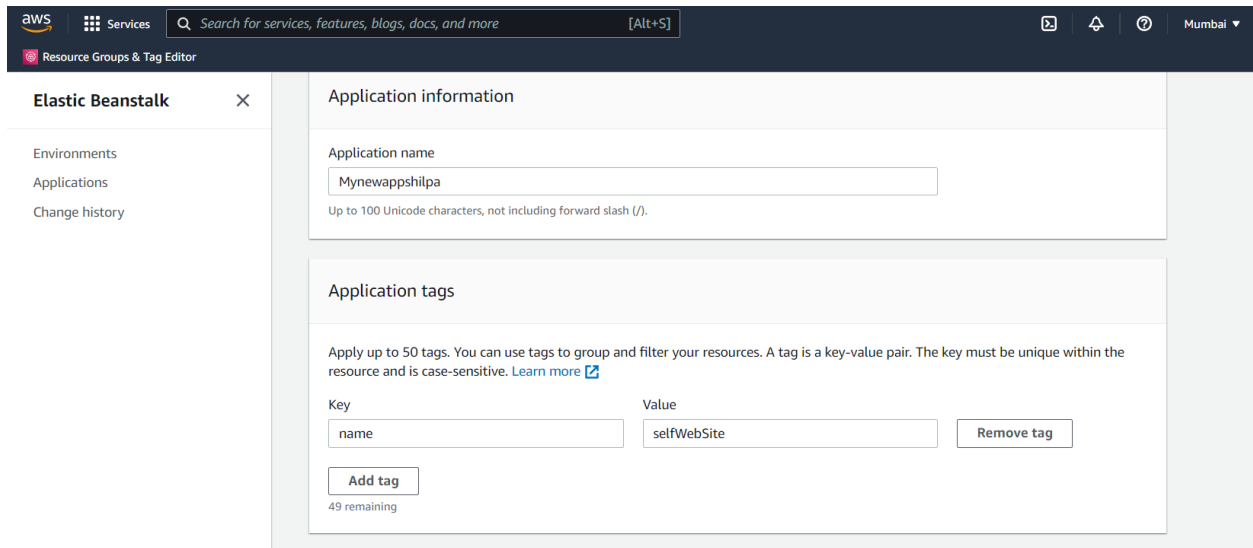
Step1 : Login to AWS console and go to Elastic Beanstalk



Step 2: Click on Create Application



Step 3: Write Application information: Name, Tag,Platform etc.



aws Services Search for services, features, blogs, docs, and more [Alt+S] Mumbai

Resource Groups & Tag Editor

Elastic Beanstalk ✕

Environments
Applications
Change history

Application information

Application name
Mynewappshilpa
Up to 100 Unicode characters, not including forward slash (/).

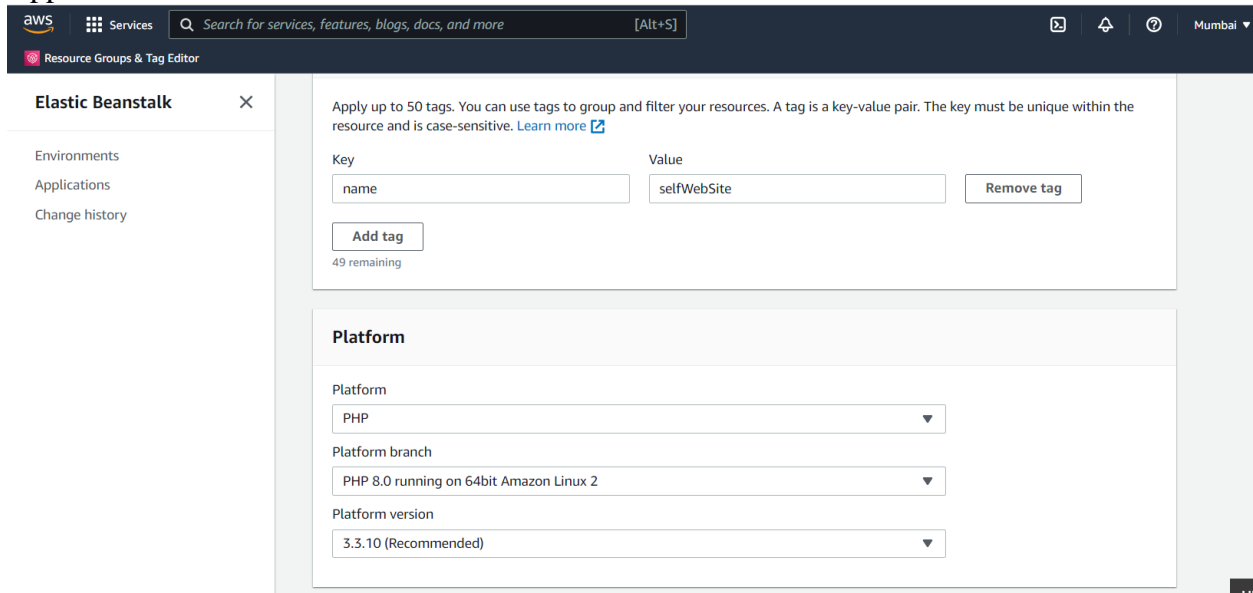
Application tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

Key	Value	
name	selfWebSite	Remove tag

Add tag
49 remaining

Step 4: In Application Code: select sample application and then Click on button Create Application



aws Services Search for services, features, blogs, docs, and more [Alt+S] Mumbai

Resource Groups & Tag Editor

Elastic Beanstalk ✕

Environments
Applications
Change history

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. [Learn more](#)

Key	Value	
name	selfWebSite	Remove tag

Add tag
49 remaining

Platform

Platform
PHP

Platform branch
PHP 8.0 running on 64bit Amazon Linux 2

Platform version
3.3.10 (Recommended)

This will take a few minutes.

Step 5: Click on Environments -> Check the health of Environment wait till it becomes 'OK'

Application code

☒ Sample application
Get started right away with sample code.

☐ Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

Cancel

Configure more options

Create application

AWS

Services

Search for services, features, blogs, docs, and more

[Alt+S]

Resource Groups & Tag Editor

Elastic Beanstalk

Environments

Applications

Change history

▼ apptestshilpa

Application versions

Saved configurations

▶ Apptestshilpa-env

Elastic Beanstalk > Environments > Apptestshilpa-env

Creating Apptestshilpa-env

This will take a few minutes.

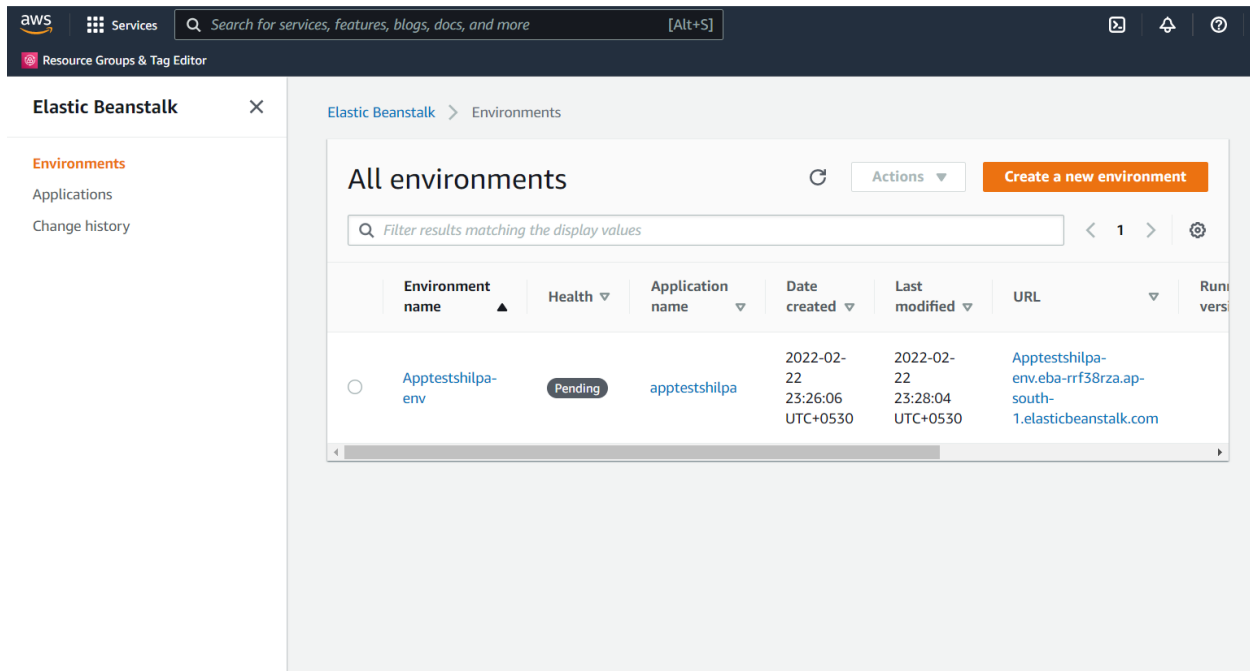
11:26pm Created security group named:
sg-0eb4e480238fb8826

11:26pm Created target group named:
arn:aws:elasticloadbalancing:ap-south-1:580032287469:targetgroup/awseb-AWSEB-ZN1T36EQ9LKB/790652a1d8640215

11:26pm Using elasticbeanstalk-ap-south-1-580032287469 as Amazon S3 storage bucket for environment data.

11:26pm createEnvironment is starting.

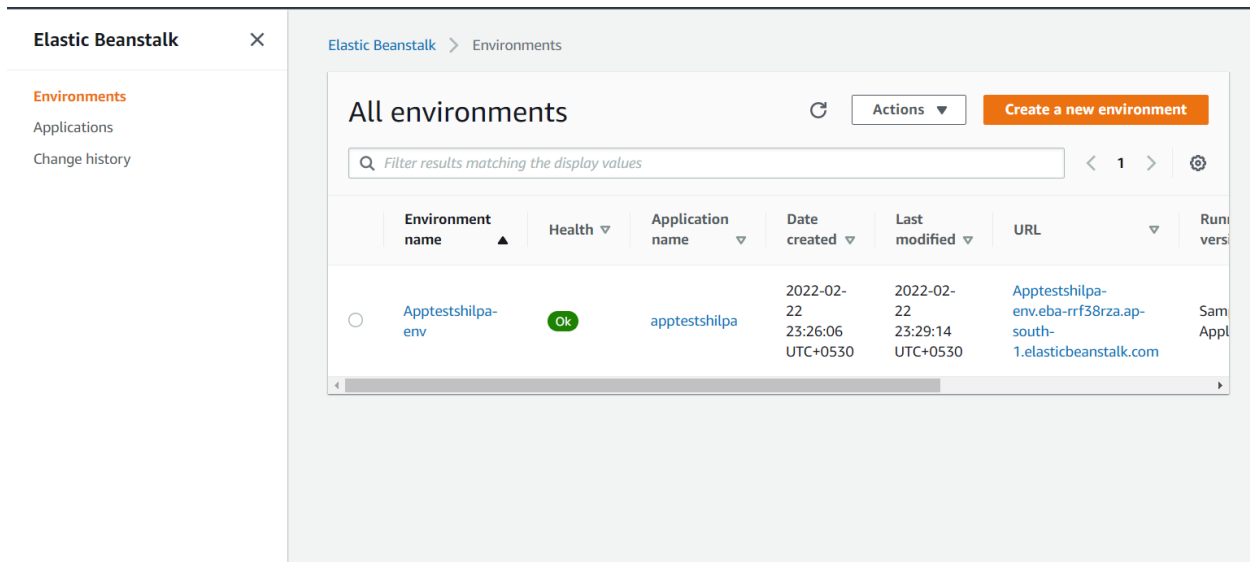
Step 6: Click the URL



The screenshot shows the AWS Elastic Beanstalk console. The left sidebar has 'Elastic Beanstalk' selected. The main content area is titled 'All environments'. A table lists the environments. The first environment, 'Apptestshilpa-env', has a health status of 'Pending'. The URL column shows the endpoint: 'Apptestshilpa-env.eba-rrf38rza.ap-south-1.elasticbeanstalk.com'.

Environment name	Health	Application name	Date created	Last modified	URL	Run version
Apptestshilpa-env	Pending	apptestshilpa	2022-02-22 23:26:06 UTC+0530	2022-02-22 23:28:04 UTC+0530	Apptestshilpa-env.eba-rrf38rza.ap-south-1.elasticbeanstalk.com	

To Delete the application and Environment (Select it and in **Action** -Delete/Terminate : give conformation)



The screenshot shows the AWS Elastic Beanstalk console. The left sidebar has 'Elastic Beanstalk' selected. The main content area is titled 'All environments'. A table lists the environments. The first environment, 'Apptestshilpa-env', has a health status of 'Ok'. The URL column shows the endpoint: 'Apptestshilpa-env.eba-rrf38rza.ap-south-1.elasticbeanstalk.com'.

Environment name	Health	Application name	Date created	Last modified	URL	Run version
Apptestshilpa-env	Ok	apptestshilpa	2022-02-22 23:26:06 UTC+0530	2022-02-22 23:29:14 UTC+0530	Apptestshilpa-env.eba-rrf38rza.ap-south-1.elasticbeanstalk.com	Sam Appl

Congratulations!

Your AWS Elastic Beanstalk *PHP* application is now running on your own dedicated environment in the AWS Cloud

You are running PHP version 8.0.13

This environment is launched with Elastic Beanstalk PHP Platform

What's Next?

- [AWS Elastic Beanstalk overview](#)
- [Deploying AWS Elastic Beanstalk Applications in PHP Using Eb and Git](#)
- [Using Amazon RDS with PHP](#)
- [Customizing the Software on EC2 Instances](#)
- [Customizing Environment Resources](#)

AWS SDK for PHP

- [AWS SDK for PHP home](#)
- [PHP developer center](#)
- [AWS SDK for PHP on GitHub](#)

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the ERP or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no ERP access available)

Roll No.B48	Name: Aryan Unhale
Class :TE COMPS B	Batch :B3
Date of Experiment: 27/2/25	Date of Submission: 27/2/25
Grade :	

Step 1:Login to AWS Console and Create EC2 key pair.

Create key pair | EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateKeyPair:

aws Search [Alt+S]

EC2 > Key pairs > Create key pair

Create key pair [Info](#)

Key pair
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name
IntazaKey
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type [Info](#)
☒ RSA ☐ ED25519

Private key file format
☐ .pem
For use with OpenSSH
☒ .ppk
For use with PuTTY

Tags - optional

Key	Value - optional
Environment	Production

[Add new tag](#)
You can add up to 49 more tags.

us-east-1.console.aws.amazon.com/iam/home?region=ap-south-1#/roles/create

Create role | IAM | AWS

IAM > Roles > Create role

Step 1
Select trusted entity

Step 2
Add permissions

Step 3
Name, review, and create

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.

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

The screenshot shows the AWS IAM console 'Create role' page. The browser address bar shows the URL: `us-east-1.console.amazonaws.com/iam/home?region=ap-south-1#/roles/create?trustedEntityType=AWS_SERVICE&selectedService=EC2&selectedUseCase=EC2&policies=arn%3Aaws%3A...`. The page has a sidebar with navigation links like 'IAM', 'Roles', and 'Create role'. The main content area is titled 'Name, review, and create'. It contains a 'Role details' section with a 'Role name' field (containing 'EC2ReadOnlyS3FullAccess') and a 'Description' field (containing 'Allows EC2 instances to call AWS services on your behalf.'). Below these is a 'Trust policy' section with a 'Trust policy' field (containing a JSON snippet: `{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "s3:ListBucket", "Resource": "*" }] }`). The page is in the 'Step 1: Select trusted entities' phase.

aws [Search] [Alt+S] Global Intara%20

Identity and Access Management (IAM) Search IAM

Dashboard

▼ Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings
- Root access management [New](#)

▼ Access reports

- Access Analyzer
- External access
- Unused access
- Analyzer settings
- Credential report
- Organization activity
- Service control policies

EC2ReadOnlyS3FullAccess

Allows EC2 instances to call AWS services on your behalf.

[Delete](#)

[Edit](#)

Summary

Creation date
February 26, 2025, 22:29 (UTC+05:30)

ARN
[arn:aws:iam::699475959073:role/EC2ReadOnlyS3FullAccess](#)

Instance profile ARN
[arn:aws:iam::699475959073:instance-profile/EC2ReadOnlyS3FullAccess](#)

Last activity
-

Maximum session duration
1 hour

Permissions Trust relationships Tags Last Accessed Revoke sessions

Permissions policies (2) [Info](#)

You can attach up to 10 managed policies.

[Simulate](#) [Remove](#) [Add permissions](#)

Filter by Type
All types

<input type="checkbox"/>	Policy name Info	Type	Attached entities
<input type="checkbox"/>	AmazonEC2ReadOnlyAccess	AWS managed	1
<input type="checkbox"/>	AmazonS3FullAccess	AWS managed	1

Step 3: Navigate to Elastic Beanstalk & Click on "Create Application"

aws [Search] [Alt+S] Asia Pacific (Mumbai) Intara%20

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.

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](#)

Step 4: Provide Application Information(Choose Environment tier,Application Name,Tags,Platform)

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

Environment tier

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

Application name

Intaza-Website

Maximum length of 100 characters.

Application tags (optional)

Environment information

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

Environment name

Intaza-Website-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

Platform

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

Node.js

Platform branch

Node.js 22 running on 64bit Amazon Linux 2023

Platform version

6.4.3 (Recommended)

Application code

☒ 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

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 5 : Configure service access

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](#)

IntazaKey

EC2 instance profile

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

EC2ReadOnlyS3FullAccess

View permission details

Step 6: Set up networking, database, and tags

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

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-0fc116c82609fe67b | (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"/> ap-south-1c	subnet-01166486613ebf5bd	172.31.16.0/20	
<input checked="" type="checkbox"/> ap-south-1a	subnet-01c01a90627fa8eff	172.31.32.0/20	
<input checked="" type="checkbox"/> ap-south-1b	subnet-0cb40017df3d4a6ba	172.31.0.0/20	

Step 7- Configure instance traffic and scaling

[illegible]

aws

Search

[Alt+S]

Asia Pacific (Mumbai) ▾Intaza%20 ▾

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 environment with your preferred launch order. The order preference only applies to On-Demand Instances and Spot Instances that use the capacity optimized prioritized allocation strategy. We recommend you include at least two instance types. [Learn more](#)

1.

t2.micro

Add instance type

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-ObOced806bca5e413

Availability Zones

Number of Availability Zones (AZs) to use.

Any

[illegible]

RMS
[Alt+J]
Asia Pacific (Humboldt)
InstanceID

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

Review Info

Step 1: Configure environment Edit

Environment information

Environment tier Web server environment Environment name Intaza-Website-env Platform arn:aws:elasticbeanstalk-south-1:platform/Node.js 2.2 running on 64bit Amazon Linux 2023/6.4.3	Application name Intaza-Website Application code Sample application
---	--

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 key pair	EC2 instance profile
arn:aws:iam::609475959073:role/service-role/aws-elasticbeanstalk-service-role	IntazaKey	EC2ReadOnlyS3FullAccess

Step 3: Set up networking, database, and tags Edit

Networking, database, and tags Info

Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

Network	
VPC vpc-0fc116d2609fe67b	Public IP address true Instance subnets subnet-0101a90627fa0eff, subnet-0cb40017df3d4a6ba

Tags

Key	Value
No tags There are no tags defined	

Step 4: Configure instance traffic and scaling Edit

Instance traffic and scaling Info

Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic. Configure the software that runs on your environment's instances by setting platform-specific options.

Instances		
IMDSv1 Deactivated	EC2 Security Groups sg-02bd6c5ff9495093	

Capacity	Fleet composition	On-demand base
Environment type Single instance On-demand above base 0	Capacity rebalancing Deactivated	Scaling cooldown 360

Processor type	Instance types	AMI ID
x86_64	t2.micro	ami-0b0ce8f806ca5e413

Step 5: Configure updates, monitoring, and logging

Edit

Updates, monitoring, and logging

info

Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

Monitoring

System

Cloudwatch custom metrics - instance

Cloudwatch custom metrics - environment

Log streaming

Deactivated

Retention

7

Lifecycle

false

Updates

Managed updates

Deactivated

Deployment batch size

100

Deployment batch size type

Percentage

Command timeout

600

Deployment policy

AllAtOnce

Health threshold

OK

Platform software

Lifecycle

false

Log streaming

Deactivated

Proxy server

nginx

Ignore health check

false

Instance replacement

false

Update level

minor

Environment properties

Key

Value

No environment properties

There are no environment properties defined

Step 10: Click the URL

Elastic Beanstalk

Environments

Intataz-Website-env

Applications

Environments

Change history

Application: Intataz-Website

Application versions

Saved configurations

Environment: Intataz-Website-env

Go to environment

Configuration

Events

Health

Logs

Monitoring

Alarms

Managed updates

Tags

Recent environments

Intataz-Website-env

Environment successfully launched.

Intataz-Website-env

info

Actions

Upload and deploy

Environment overview

Health

Green

Environment ID

e-enrwxasdr

Platform

Node.js 22 running on 64bit Amazon Linux 2023/6.4.3

Domain

Intataz-Website-env.eba-jm7kfz7.ap-south-1.elasticbeanstalk.com

Application name

Intataz-Website

Running version

-

Platform state

Supported

Events

Health

Logs

Monitoring

Alarms

Managed updates

Tags

Events (11)

info

Filter events by text, property or value

Time

Type

Details

February 26, 2025 22:55:30 (UTC+5:30)

INFO

Successfully launched environment: Intataz-Website-env

February 26, 2025 22:55:29 (UTC+5:30)

INFO

Application available at Intataz-Website-env.eba-jm7kfz7.ap-south-1.elasticbeanstalk.com.

February 26, 2025 22:55:25 (UTC+5:30)

INFO

Environment health has been set to GREEN

February 26, 2025 22:55:16 (UTC+5:30)

INFO

Instance deployment completed successfully

February 26, 2025 22:55:15 (UTC+5:30)

INFO

Adding instance 'i-016d790db88ab75c' to your environment.

February 26, 2025 22:55:15 (UTC+5:30)

INFO

Added EC2 instance 'i-016d790db88ab75c' to Auto Scaling Group 'aws-eb-e-enrwxasdr-stack-AWSEBAutoScalingGroup-H6F1e8BOGut'.

February 26, 2025 22:54:51 (UTC+5:30)

INFO

Waiting for EC2 instances to launch. This may take a few minutes.

February 26, 2025 22:54:16 (UTC+5:30)

INFO

Created EIP: 15.207.30.178

February 26, 2025 22:54:01 (UTC+5:30)

INFO

Created security group named: sg-03bdf6a7ecf28c900

February 26, 2025 22:53:34 (UTC+5:30)

INFO

Using elasticbeanstalk-ap-south-1-499475959073 as Amazon S3 storage bucket for environment data.

Environment overview - events

Elastic Beanstalk

Not secure

intataz-website-env.eba-jm7kfz7.ap-south-1.elasticbeanstalk.com

Gmail

YouTube

Maps

Terna ERP

Classroom

Congratulations

Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Node.js Platform

What's Next?

AWS Elastic Beanstalk overview

AWS Elastic Beanstalk concepts

Deploying an Express Application to AWS Elastic Beanstalk

Deploying an Express application with clustering to Elastic Beanstalk

Customizing and Configuring a Node.js Container

Working with Logs

aws [Search] [Alt+S] Asia Pacific (Mumbai) Intaza%20

EC2 > Instances

EC2

- Dashboard
- EC2 Global View
- Events
- ▼ Instances
 - Instances
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Capacity Reservations
- ▼ Images
 - AMIs
 - AMI Catalog
- ▼ Elastic Block Store
 - Volumes

Instances (1) Info

Last updated less than a minute ago

Connect Instance state Actions Launch instances

Find instance by attribute or tag (case-sensitive) All states

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	Intaza-Websit...	i-01fd790db98eab75c	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-15-2

Select an instance

Environment overview - events Elastic Beanstalk Connect to instance | EC2 | ap-south-1#ConnectToInstance:instanceId=i-01fd790db98eab75c

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ConnectToInstance:instanceId=i-01fd790db98eab75c

aws [Search] [Alt+S] Asia Pacific (Mumbai) Intaza%20

EC2 > Instances > i-01fd790db98eab75c > Connect to instance

Connect to instance Info

Connect to your instance i-01fd790db98eab75c (Intaza-Website-env) using any of these options

EC2 Instance Connect Session Manager SSH client EC2 serial console

Instance ID i-01fd790db98eab75c (Intaza-Website-env)

Connection Type

☒ Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

☐ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

☒ Public IPv4 address
15.207.30.178

☐ IPv6 address

Username

Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, root.

root

Note: In most cases, the default username, root, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel Connect

aws [Search] [Alt+S] Asia Pacific (Mumbai) Intaza%20

Amazon Linux 2023 AMI

This EC2 instance is managed by AWS Elastic Beanstalk. Changes made via SSH WILL BE LOST if the instance is replaced by auto-scaling. For more information on customizing your Elastic Beanstalk environment, see our documentation here: <http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/customize-containers-ec2.html>

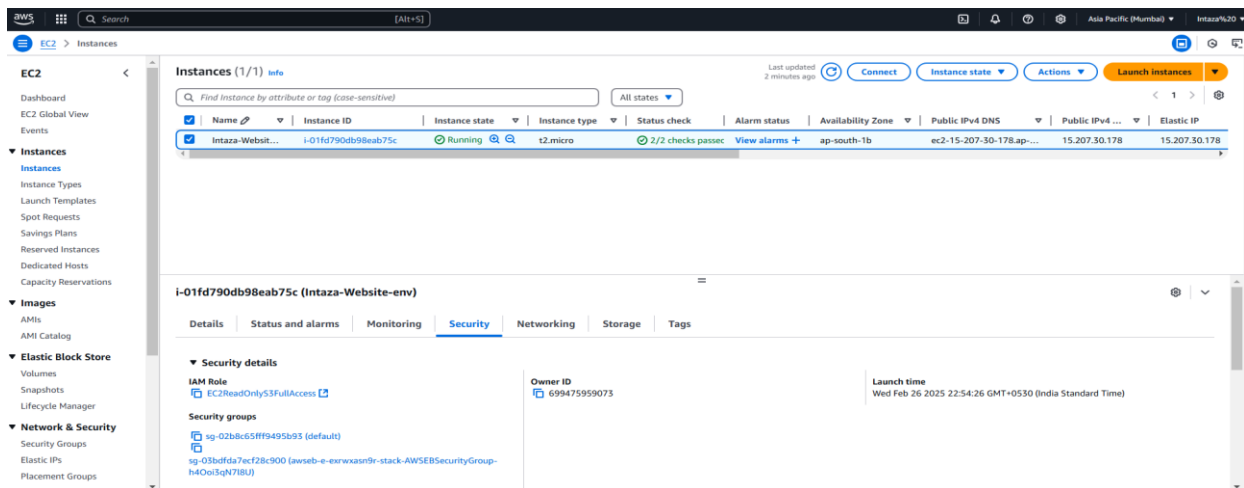
Amazon Linux 2023

<https://aws.amazon.com/linux/amazon-linux-2023>

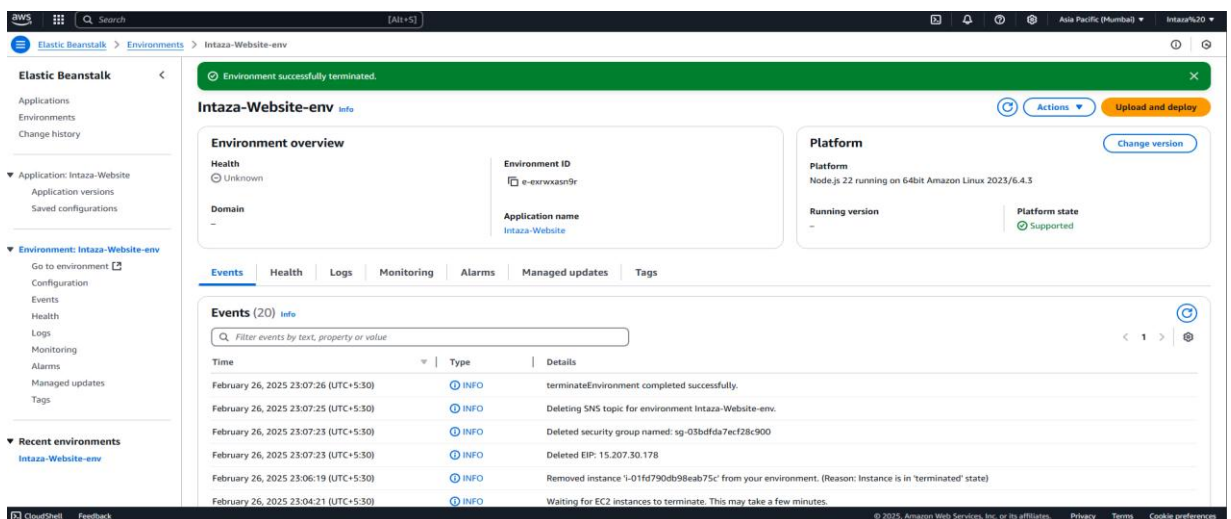
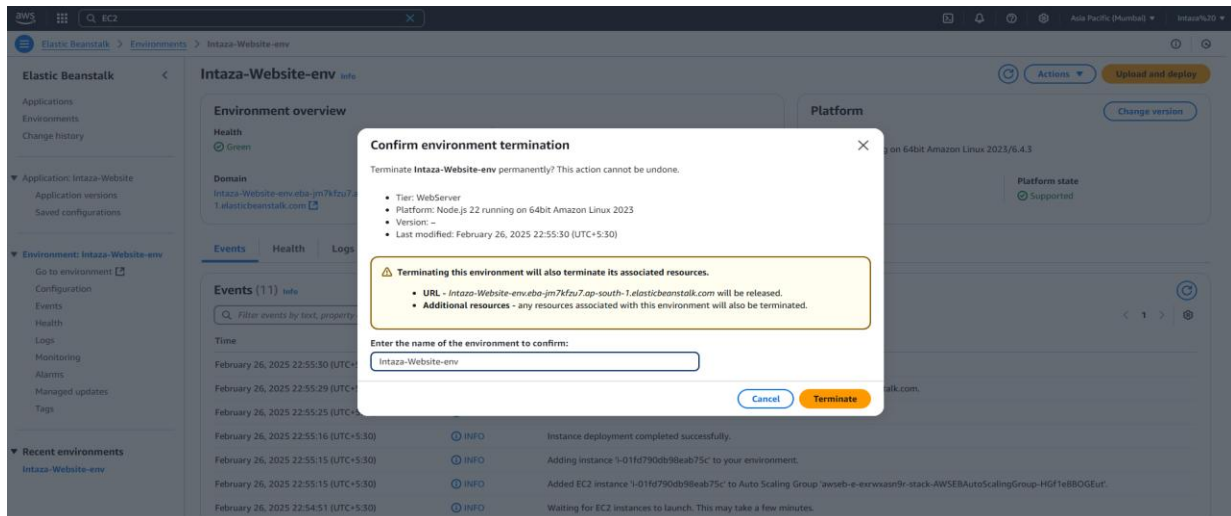
root@ip-172-31-0-174 ~]#

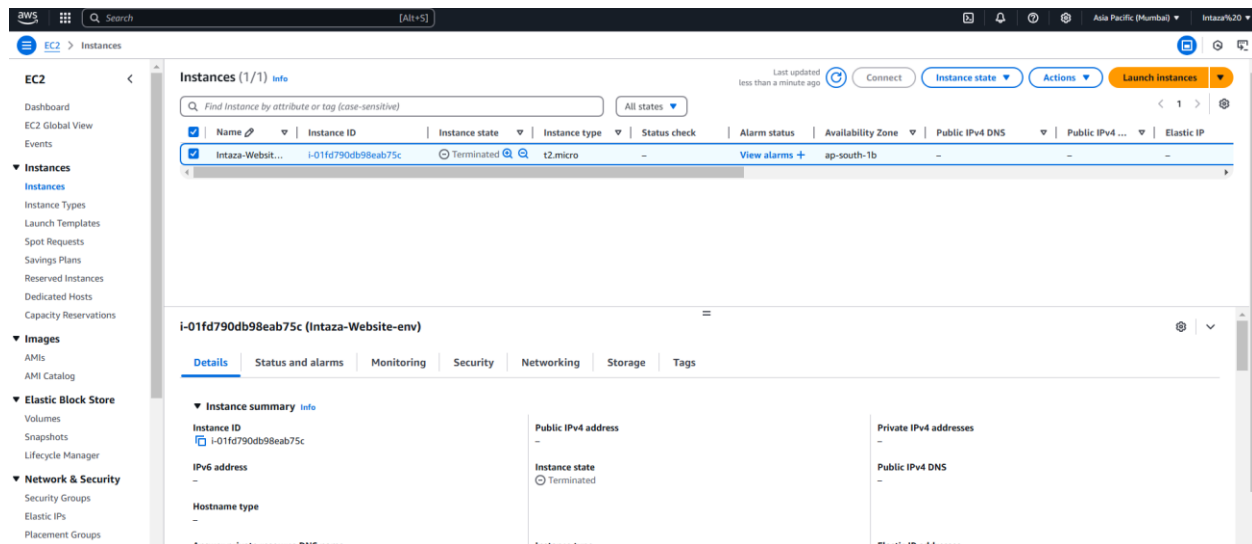
i-01fd790db98eab75c (Intaza-Website-env)

PublicIPs: 15.207.30.178 PrivateIPs: 172.31.0.174



Step 11: Terminate the Environment





B.1 Question of Curiosity:

Q1: What are the benefits of Paas in cloud computing?

Platform as a Service (PaaS) offers several benefits in cloud computing:

1. **Faster Development & Deployment** – Developers can quickly build, test, and deploy applications without managing infrastructure.
2. **Scalability** – Automatically scales resources based on demand.
3. **Cost-Efficient** – Reduces operational costs by eliminating hardware and infrastructure maintenance.
4. **Built-in Security** – Provides automatic updates, security patches, and compliance features.
5. **Multi-Platform Support** – Supports different programming languages, frameworks, and databases.
6. **Automatic Load Balancing** – Ensures high availability and performance.
7. **Easy Integration** – Seamlessly connects with databases, APIs, and third-party services.
8. **Collaboration & Accessibility** – Enables remote teams to collaborate and access cloud resources from anywhere.

Q2: What is PaaS?

Platform as a Service (PaaS) is a cloud computing model where a cloud provider offers a complete development and deployment environment without requiring users to manage the underlying infrastructure.

Key Features of PaaS:

- Provides runtime environments, databases, and development tools.

- Manages servers, networking, and storage for developers.
- Examples: AWS Elastic Beanstalk, Google App Engine, Microsoft Azure App Service.

Use Case:

A developer can deploy a web application on AWS Elastic Beanstalk without worrying about setting up servers, load balancers, or networking.

Q3) what is elastic bean stack ?

AWS Elastic Beanstalk is a Platform as a Service (PaaS) offered by AWS that automates the deployment, scaling, and management of applications.

Key Features:

- Supports multiple languages: Node.js, Python, Java, .NET, PHP, Ruby, Go.
- Automatically provisions EC2 instances, load balancers, databases, and other AWS services.
- Provides a fully managed environment for deploying web applications and APIs.

Allows developers to focus on writing code while AWS handles infrastructure management.

Example Use Case:

A company can deploy a Node.js application on Elastic Beanstalk, and AWS will handle scaling, monitoring, and load balancing automatically.

B.2 Conclusion:

Through this hands-on implementation of Platform as a Service (PaaS) using AWS Elastic Beanstalk, I have gained practical knowledge about deploying, managing, and scaling web applications in a cloud environment. This experiment helped me understand how PaaS simplifies application deployment by handling infrastructure, scaling, and resource management.

I successfully deployed a sample web application, monitored its health, and observed how AWS Elastic Beanstalk automatically manages load balancing, scaling, and provisioning of instances. Additionally, I explored how Elastic Beanstalk integrates with other AWS services like EC2, S3, RDS, and CloudWatch, ensuring seamless cloud operations. The experiment also highlighted how PaaS improves cost efficiency by eliminating the need for manual server management, making application development more efficient and scalable. Overall, this learning experience reinforced my understanding of cloud computing models and how PaaS accelerates deployment while reducing infrastructure complexities.