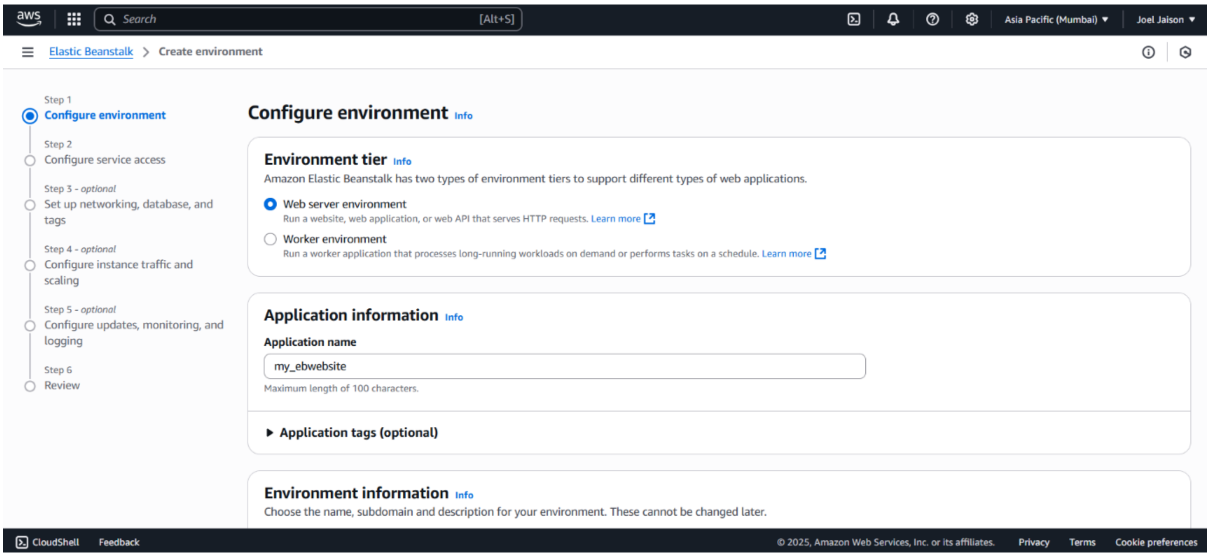
Task 1

The Project



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

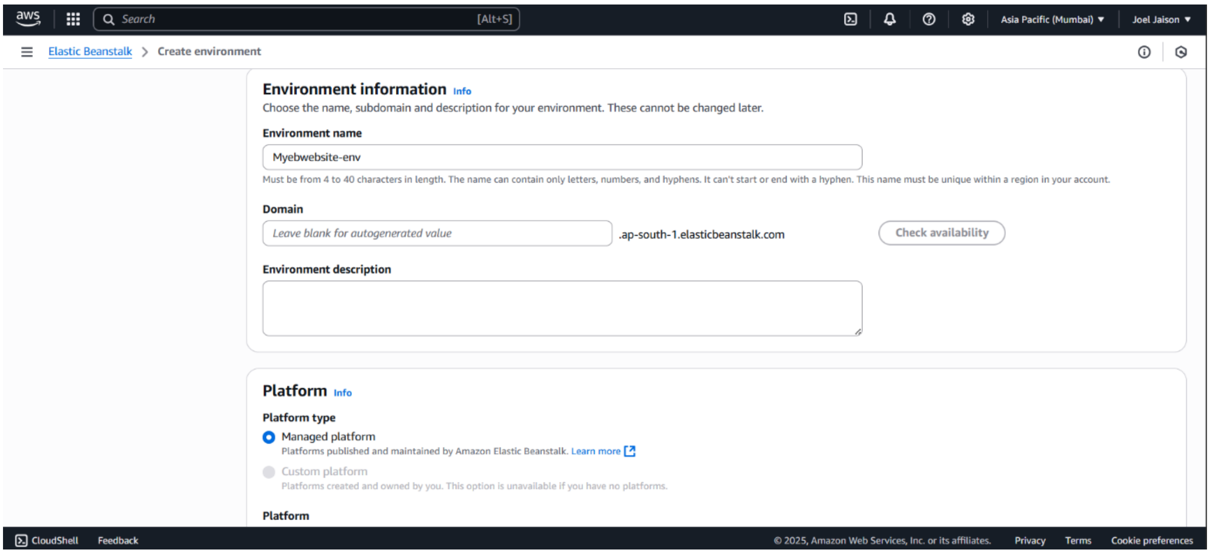
Application information Info

Application name
my_ebwebsite
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 information Info

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

Environment name
Myebwebsite-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

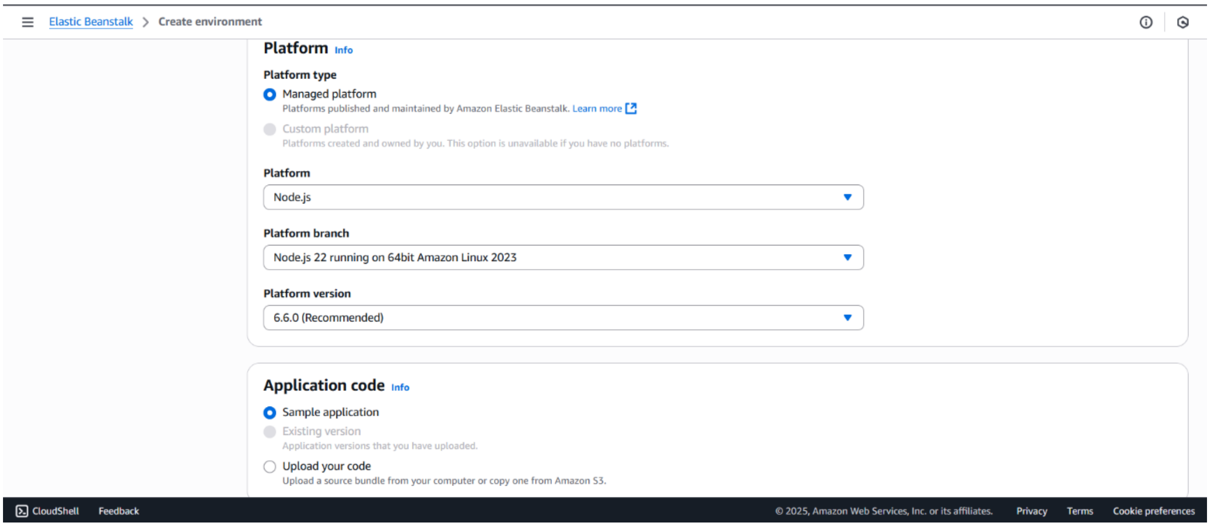
Environment description

Platform 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


Elastic Beanstalk > Create environment
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

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
Node.js

Platform branch
Node.js 22 running on 64bit Amazon Linux 2023

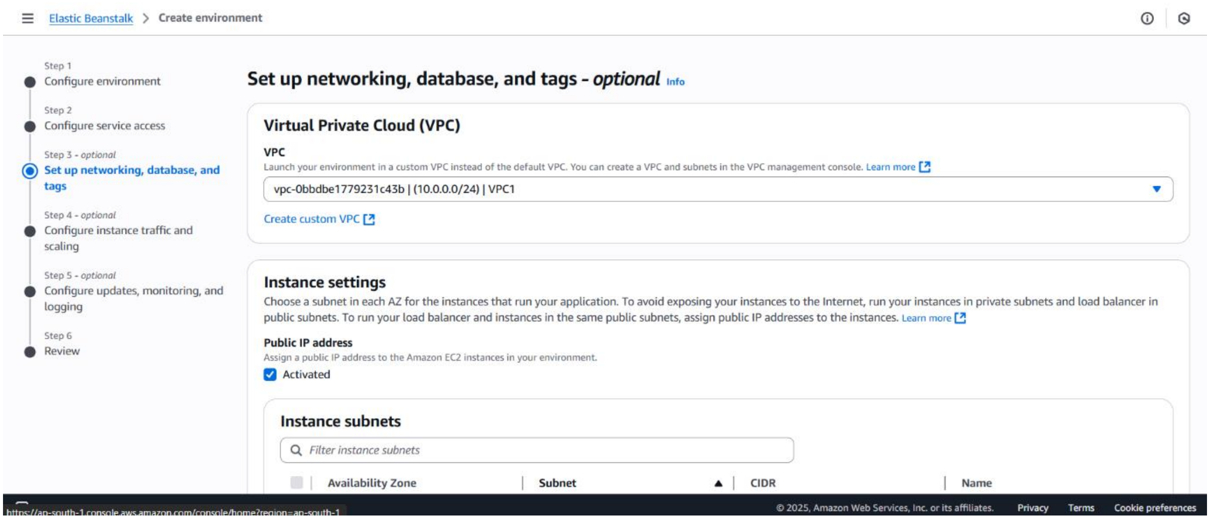
Platform version
6.6.0 (Recommended)

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.


Elastic Beanstalk > Create environment
Step 1
Configure environment
Step 2
Configure service access
Step 3 - optional
Set up networking, database, and tags
Step 4 - optional
Configure instance traffic and scaling
Step 5 - optional
Configure updates, monitoring, and logging
Step 6
Review
https://ap-south-1.console.aws.amazon.com/console/home?region=ap-south-1
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

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-0bbdbe1779231c43b | (10.0.0.0/24) | VPC1

Create custom VPC

Instance settings

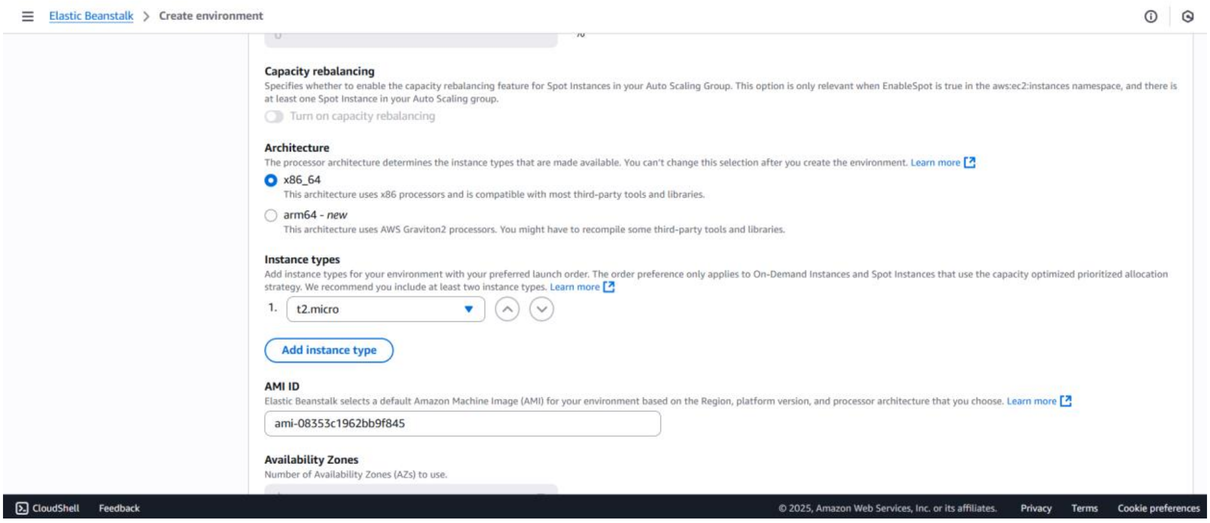
Choose a subnet in each AZ for the instances that run your application. To avoid exposing your instances to the Internet, run your instances in private subnets and load balancer in public subnets. To run your load balancer and instances in the same public subnets, assign public IP addresses to the instances.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


Elastic Beanstalk > Create environment
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

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.

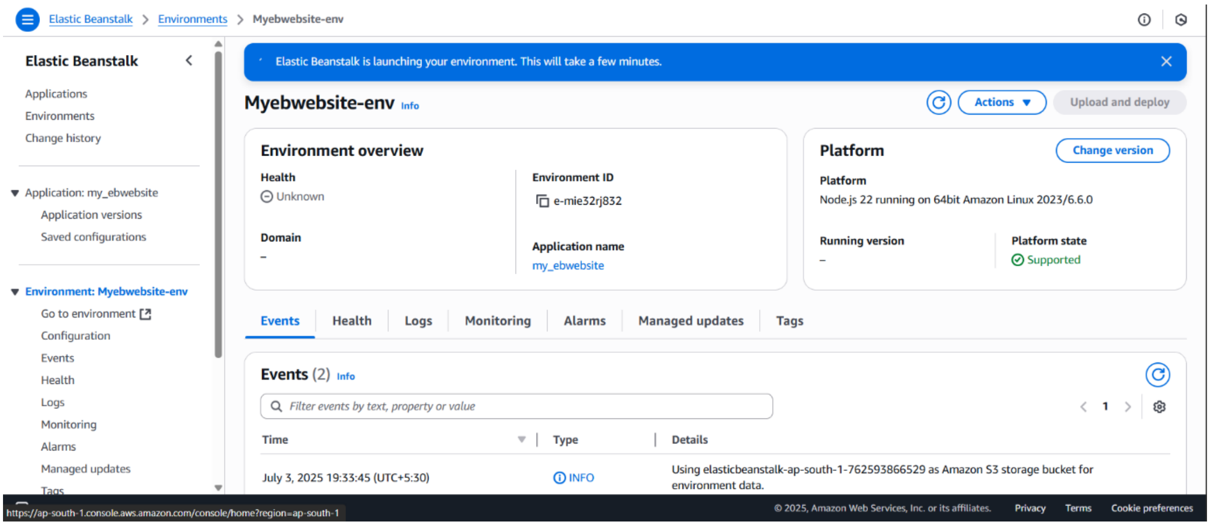
Instance types
Add instance types for your environment with your preferred launch order. The order preference only applies to On-Demand Instances and Spot Instances that use the capacity optimized prioritized allocation strategy. We recommend you include at least two instance types. Learn more

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-08353c1962bb9f845

Availability Zones
Number of Availability Zones (AZs) to use.



Elastic Beanstalk > Environments > Myebwebsite-env

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

Myebwebsite-env Info

Environment overview

Health: Unknown

Environment ID: e-mie32rj832

Domain: –

Application name: my_ebwebsite

Platform

Platform: Node.js 22 running on 64bit Amazon Linux 2023/6.6.0

Running version: –

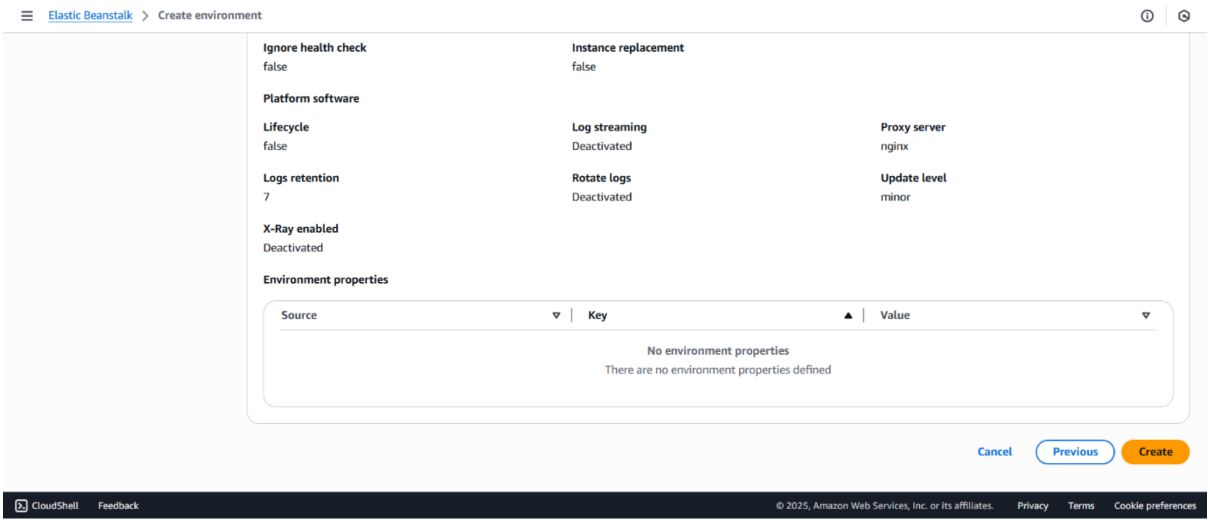
Platform state: Supported

Events (2) Info

Time	Type	Details
July 3, 2025 19:33:45 (UTC+5:30)	INFO	Using elasticbeanstalk-ap-south-1-762593866529 as Amazon S3 storage bucket for environment data.

https://ap-south-1.console.aws.amazon.com/console/home?region=ap-south-1

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



Elastic Beanstalk > Create environment

Ignore health check	Instance replacement	
false	false	

Platform software

Lifecycle	Log streaming	Proxy server
false	Deactivated	nginx
Logs retention	**Rotate logs**	**Update level**
7	Deactivated	minor
X-Ray enabled		
Deactivated		

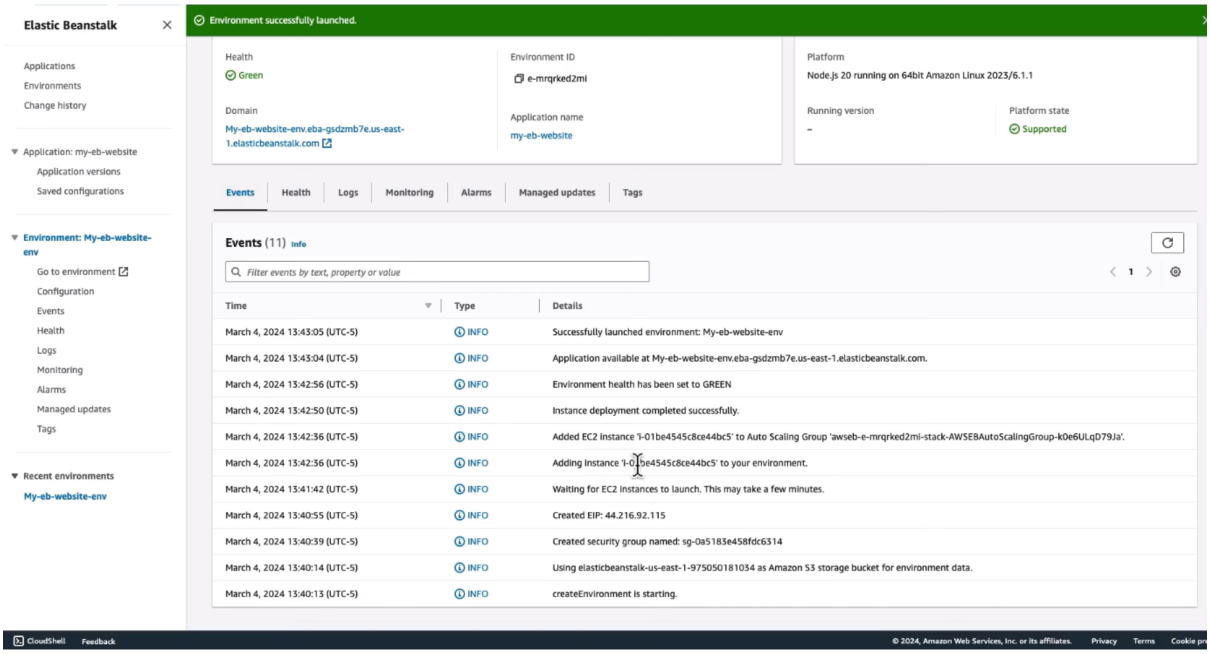
Environment properties

Source	Key	Value

No environment properties

There are no environment properties defined

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



Environment successfully launched.

Health: Green

Environment ID: e-mrqrked2mi

Domain: My-eb-website-env.eba-gsdzmb7e.us-east-1.elasticbeanstalk.com

Application name: my-eb-website

Platform: Node.js 20 running on 64bit Amazon Linux 2023/6.1.1

Running version: –

Platform state: Supported

Events (11) Info

Time	Type	Details
March 4, 2024 13:43:05 (UTC-5)	INFO	Successfully launched environment: My-eb-website-env
March 4, 2024 13:43:04 (UTC-5)	INFO	Application available at My-eb-website-env.eba-gsdzmb7e.us-east-1.elasticbeanstalk.com.
March 4, 2024 13:42:56 (UTC-5)	INFO	Environment health has been set to GREEN
March 4, 2024 13:42:50 (UTC-5)	INFO	Instance deployment completed successfully.
March 4, 2024 13:42:36 (UTC-5)	INFO	Added EC2 instance 'i-01be4545c8ce44bc5' to Auto Scaling Group 'awseb-e-mrqrked2mi-stack-AWSEBAutoScalingGroup-k0e6ULqD79Ja'.
March 4, 2024 13:42:36 (UTC-5)	INFO	Adding instance 'i-01be4545c8ce44bc5' to your environment.
March 4, 2024 13:41:42 (UTC-5)	INFO	Waiting for EC2 instances to launch. This may take a few minutes.
March 4, 2024 13:40:55 (UTC-5)	INFO	Created EIP: 44.216.92.115
March 4, 2024 13:40:39 (UTC-5)	INFO	Created security group named: sg-0a5183e458fdc6314
March 4, 2024 13:40:14 (UTC-5)	INFO	Using elasticbeanstalk-us-east-1-975050181034 as Amazon S3 storage bucket for environment data.
March 4, 2024 13:40:13 (UTC-5)	INFO	createEnvironment is starting.

© 2024, Amazon Web Services, Inc. or its affiliates.

Uses and Application of the Task

1. **Testing Deployment Pipelines**

   Before deploying a real application, using the sample application allows developers to test the entire pipeline without any code.
   This is helpful to ensure that all configurations, permissions, and environment settings are correctly in place.
   It acts as a sandbox environment where one can practice deploying, scaling, and terminating environments with zero risk to actual applications.

2. **Preparing for Real Application Hosting**

   Once familiar with Elastic Beanstalk through the default application, users can move on to deploying their own code.
   The same environment setup and deployment steps can be reused for Flask, Node.js, Java, or .NET applications.
   This transition is smooth because the user already understands how deployment works on a structural level.