

ELASTIC BEANSTALK

ELASTIC BEANSTALK?

A Platform as a Service Solution

AWS Elastic Beanstalk is a PaaS-like service designed for deploying and managing web applications without the need to manage underlying infrastructure, simplifying the development process significantly.

THE PROMISE

You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring.

WHY?

01

Multi-Platform Support:

Native support for Go, Java, .NET, Node.js, PHP, Python, Ruby, and Docker.

02

Monitoring:

Unified user interface to monitor application health (40+ metrics).

03

Deployment Options

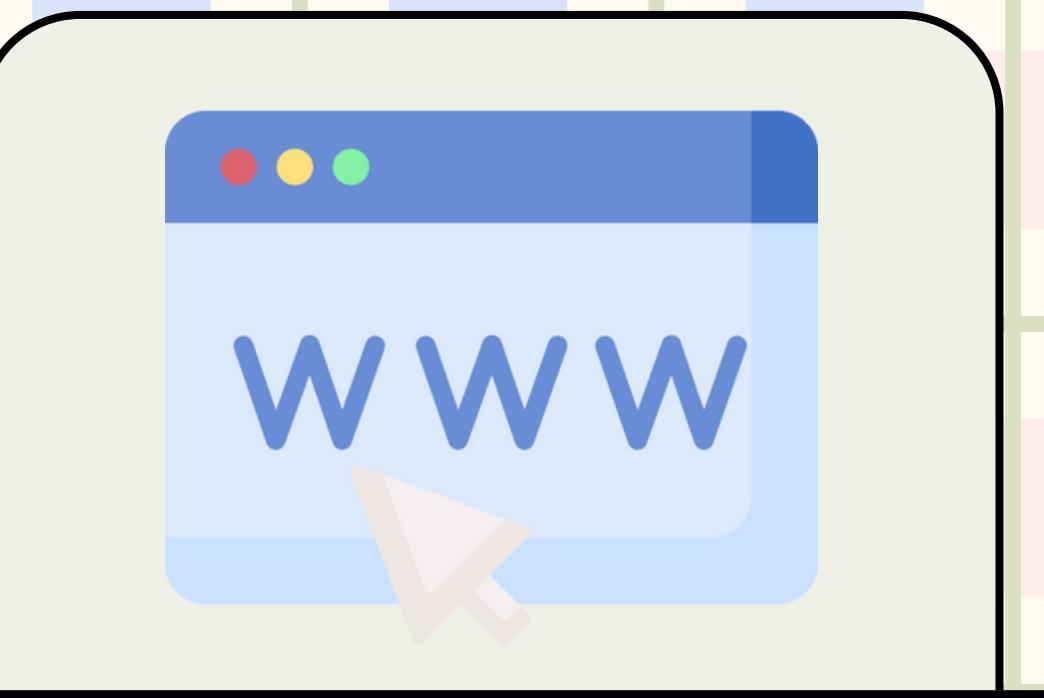
Deploy via AWS Console, EB CLI, Visual Studio, or Eclipse.

04

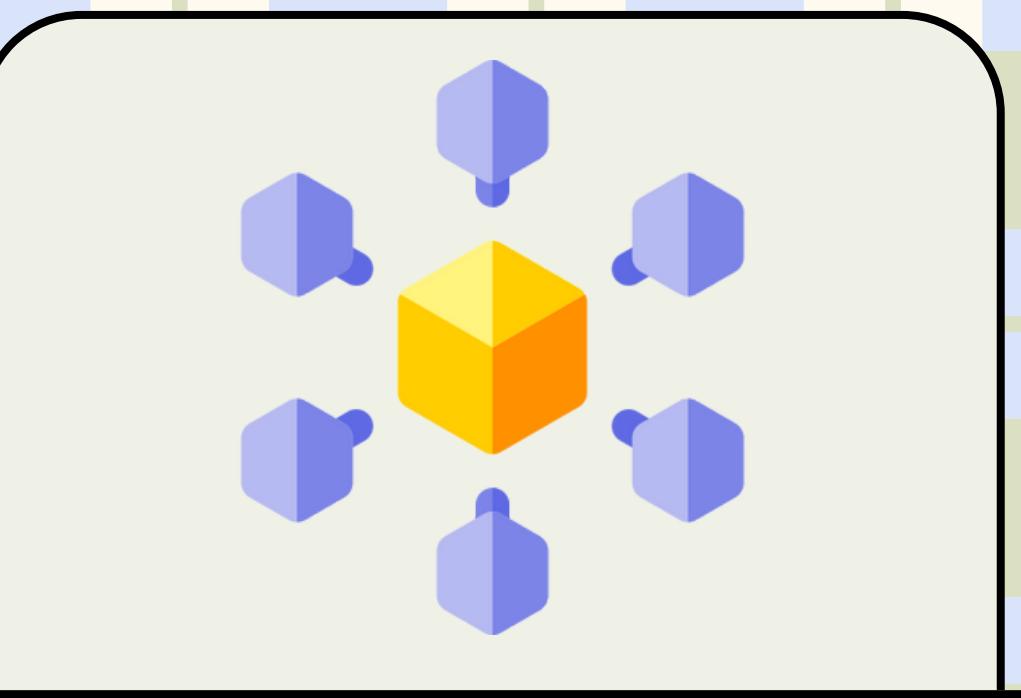
Managed Updates

Automatically applies the latest platform updates and patches.

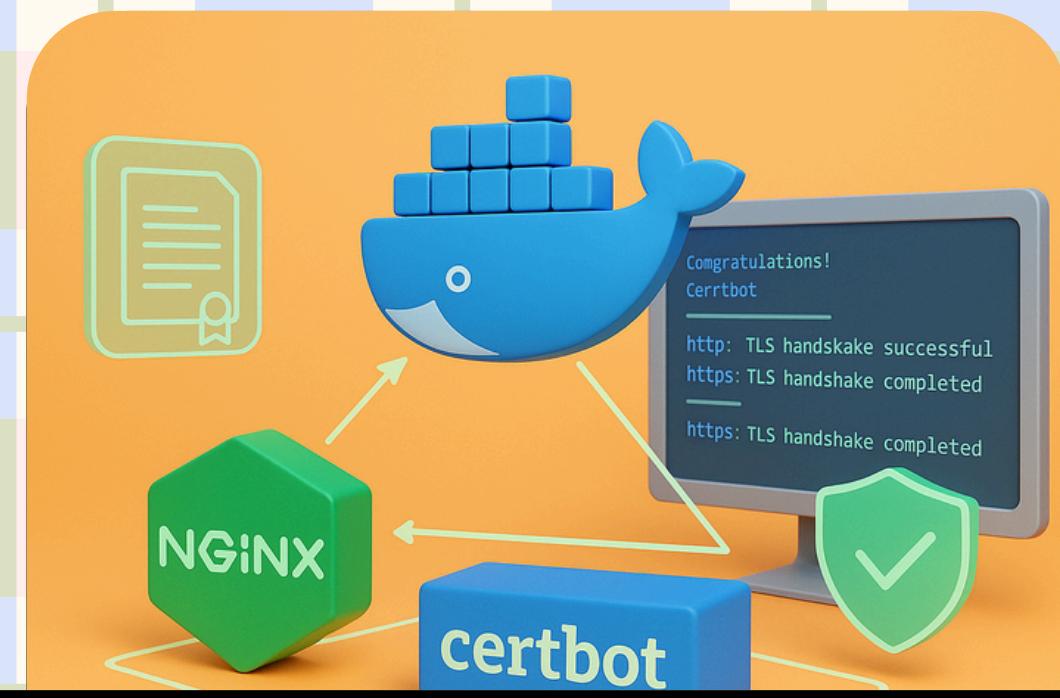
USE CASES



WEB APPS



MICROSERVICES



WORKER ENV

ADVANTAGES

Ease of use

Developers simply upload code for deployment.

Faster time

Reduces operational overhead, accelerating launch times.

Scalability

Automatically adjusts resources to handle traffic spikes.

AWS Integration

Seamlessly connects with various AWS services.



DISADVANTAGES

Limited control

Users may not access full infrastructure details.

Troubleshooting complexity

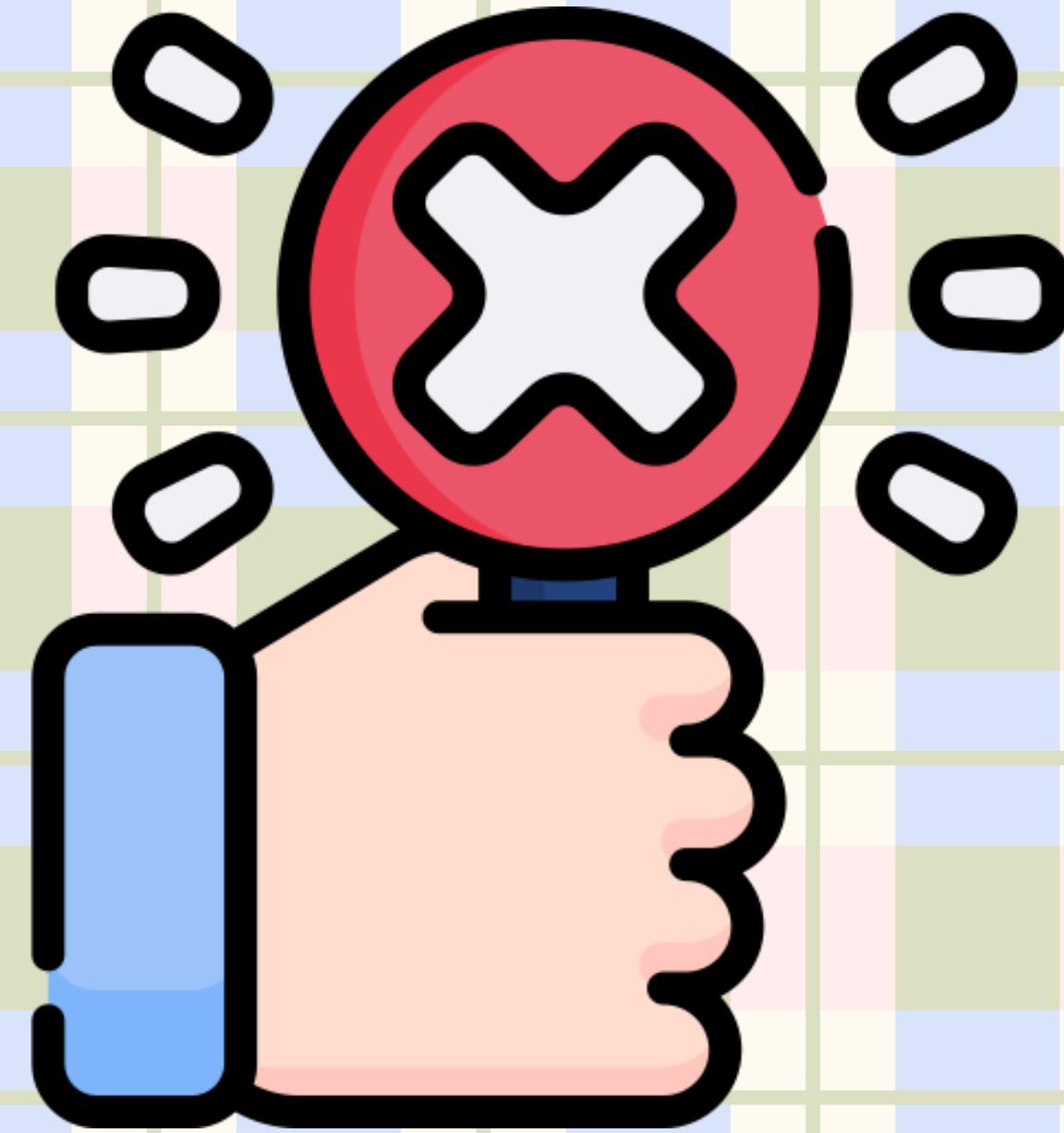
Abstraction can make issue resolution harder.

Vendor lock-in

Deep integration may limit future platform changes.

Performance overhead

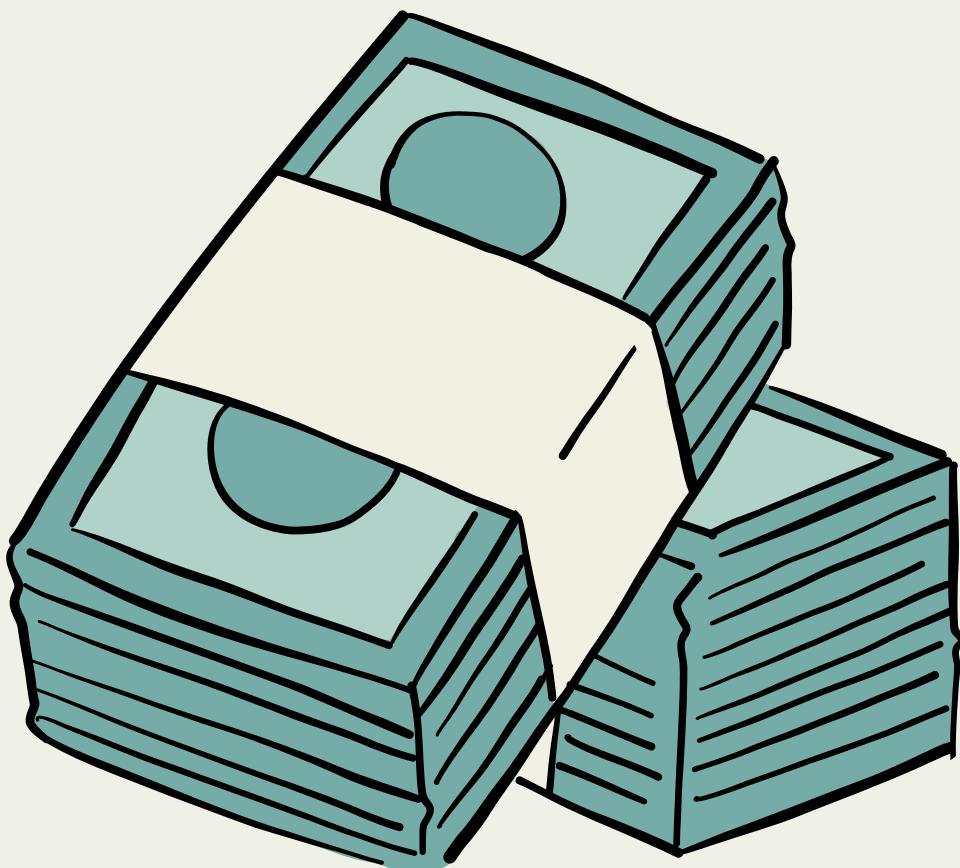
Potential latency compared to custom setups.



PRICING

AWS Elastic Beanstalk itself is free

users are charged only for the underlying AWS resources utilised, such as EC2, S3, and ELB, making it a cost-effective solution.



HOW TO QUICKLY GET STARTED

CREATE AN ELASTIC BEANSTALK APPLICATION

The screenshot shows the Amazon Elastic Beanstalk landing page. At the top, there's a navigation bar with the AWS logo, a search bar, and a [Alt+S] keyboard shortcut. Below the header, the word "Compute" is visible. The main title "Amazon Elastic Beanstalk" is prominently displayed, followed by the subtitle "End-to-end web application management." A descriptive paragraph explains that Elastic Beanstalk is a service for deploying and scaling web applications using various languages and frameworks. To the right, there's a callout box titled "Get started" with the subtext "Easily deploy your web application in minutes." It features a large orange "Create application" button, which is highlighted with a yellow oval. Another callout box titled "Pricing" states that there's no additional charge for the service; users pay for underlying AWS resources like EC2 instances. At the bottom, there are sections for "Get started" (with a note about automatic deployment), "Benefits and features" (with a "Learn more" link), and "Getting started" (with a "Launch a web application" link).

aws | Search [Alt+S]

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 ↗](#)

Benefits and features

Getting started

[Launch a web application](#)

CONFIGURE THE ENVIRONMENT FOR YOUR APPLICATION

The screenshot shows the AWS Elastic Beanstalk 'Create environment' wizard. On the left, a vertical navigation bar lists steps: Step 1 (Configure environment, marked with a blue circle), 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 environment' with an 'Info' link. It contains two sections: 'Environment tier' (with 'Web server environment' selected) and 'Application information'. A large orange arrow points from the text 'We choose a web server since we are going to deploy a website' to the 'Web server environment' option. Another orange arrow points from the text 'Give it a name of your liking; this is the name that will be seen in your application list' to the 'Application name' input field, which contains 'LaravelSocialMediaApp'. A note below the input field states 'Maximum length of 100 characters.'

aws | Search [Alt+S] | Maryam Tarek

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

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

LaravelSocialMediaApp

Maximum length of 100 characters.

▶ Application tags (optional)

We choose a web server since we are going to deploy a website

Give it a name of your liking; this is the name that will be seen in your application list

CONFIGURE THE ENVIRONMENT FOR YOUR APPLICATION

The screenshot shows the AWS Elastic Beanstalk 'Create environment' wizard. The top navigation bar includes the AWS logo, a search bar, and various icons. The main content area is titled 'Create environment' under 'Elastic Beanstalk'. It has three tabs: 'Application tags (optional)', 'Environment information', and 'Environment description'. The 'Environment information' tab is active, showing fields for 'Environment name' (set to 'LaravelSocialMediaApp-env') and 'Domain' (set to 'laravel-social-media-v1.eu-north-1.elasticbeanstalk.com'). A large orange arrow points from the explanatory text on the left to the 'Domain' field. Another orange arrow points from the explanatory text at the bottom to the 'Check availability' button.

The domain name, this is where your application will be deployed, you can leave it and it will get auto generated

► Application tags (optional)

Environment information Info

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

Environment name Your environment name, you can name it whatever you want but it defaults to `ApplicationName-env`

LaravelSocialMediaApp-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

laravel-social-media-v1.eu-north-1.elasticbeanstalk.com

Check availability

► Environment description

Platform Info

Platform

Be sure to check if the domain is available if you decide to give it a name

CONFIGURE THE ENVIRONMENT FOR YOU APPLICATION

The screenshot shows the 'Create environment' step in the AWS Elastic Beanstalk console. The interface has a dark header with the AWS logo, search bar, and various navigation icons.

Environment description: A large orange arrow points to the 'Platform' dropdown, which is set to 'PHP'. The text next to it says: "Choose your platform you can choose from Docker, .NET, Go, PHP, Java, Node.js, and others I will go with PHP".

Platform branch: An orange arrow points to the dropdown menu, which is set to 'PHP 8.2 running on 64bit Amazon Linux 2023'. The text next to it says: "Choose the platform branch, it depends on your app version In my case I will choose PHP 8.2".

Platform version: An orange arrow points to the dropdown menu, which is set to '4.8.0 (Recommended)'. The text next to it says: "Choose the platform version I will choose latest".

Application code: An orange arrow points to the 'Info' link next to the 'Application code' section. The section shows 'Sample application' selected. The text next to it says: "Here you can upload your code to be deployed But I will choose Sample application for demonstration purposes".

CONFIGURE THE ENVIRONMENT FOR YOUR APPLICATION

The screenshot shows the AWS Elastic Beanstalk 'Create environment' wizard. At the top, there's a navigation bar with the AWS logo, search bar, and various icons. The main content area has a breadcrumb trail: 'Elastic Beanstalk > Create environment'. On the left, there's a sidebar with three options: 'Sample application' (selected), 'Existing version', and 'Upload your code'. A callout bubble points to the 'Upload your code' option with the text 'You can upload your code if you want'. Below this, the 'Presets' section is shown with a 'Configuration presets' dropdown. The 'Single instance (free tier eligible)' option is selected. A callout bubble points to this section with the text 'Here you can choose a preset'. Another callout bubble points to the 'Single instance' option with the text 'I will go with Single because it's supported in the free tier and I'm not going to deploy my app into production'. On the far left, another callout bubble points to the 'High availability' option with the text 'While high is recommended for production'. At the bottom right are 'Cancel' and 'Next' buttons.

aws | Search [Alt+S] Maryam Tarek

Elastic Beanstalk > Create environment

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.

You can upload your code if you want

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

Single is primarily recommended for development use cases and will save costs

While high is recommended for production

Here you can choose a preset

I will go with Single because it's supported in the free tier and I'm not going to deploy my app into production

Cancel Next

CONFIGURE SERVICE ACCESS AND IAM USERS

aws | Search [Alt+S] | Maryam Tarek

Elastic Beanstalk > Create 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

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 ↗](#)

Lets start with the Service IAM user (It will open a new tab)

Service role
Choose an IAM role for Elastic Beanstalk to assume as a service role. The IAM role must have the required IAM managed policies.

[Choose a service role](#) [Create role ↗](#)

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

[Choose a instance profile](#) [Create role ↗](#)

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

[Choose a key pair](#) [Create role ↗](#)

We will need to create two IAM users, one for Elastic Beanstalk and the other for EC2 instances

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

CONFIGURE SERVICE ACCESS AND IAM USERS

1 SELECT THE TYPE

- 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.

- 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 from a corporate directory to assume this role to perform actions in this account.

2 THEN THE USE CASE

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

Elastic Beanstalk

Choose a use case for the specified service.

Use case

- Elastic Beanstalk - Compute
Allows your environment's EC2 instances to perform operations required for your application.
- Elastic Beanstalk - Environment
Allows access to other AWS service resources that are required to create and manage environments.

3 THEN THE PERMISSIONS

Add permissions Info

Permissions policies (2) Info

The type of role that you selected requires the following policy.

Policy name ↗

 AWSElasticBeanstalkEnhancedHealth

AWS managed

 AWSElasticBeanstalkManagedUpdatesCustomer...

AWS managed

► Set permissions boundary - optional

Cancel

Previous

Next

Default is fine, click next

REVIEW THE IAM USER THEN CLICK CREATE ROLE

The screenshot shows the AWS IAM Role Creation Wizard with three main steps:

- Step 1: Select trusted entities**

Trust policy

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

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AWSElasticBeanstalkEnhancedHealth	AWS managed	Permissions policy
AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy	AWS managed	Permissions policy

Step 3: Add tags

Add tags - optional Info

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.
- Name, review, and create**

THE ROLE IS NOW CREATED

The screenshot shows the AWS IAM Roles page. At the top, there is a green success message: "Role aws-elasticbeanstalk-service-role created." Below this, the "Roles (3)" section is displayed. The first role listed is "aws-elasticbeanstalk-service-role", which is associated with the "AWS Service: elasticbeanstalk". The other two roles, "AWSServiceRoleForSupport" and "AWSServiceRoleForTrustedAdvisor", are associated with "AWS Service: support (Service-Linked)" and "AWS Service: trustedadvisor (Service-Linked)", respectively. On the left sidebar, under "Access management", the "Roles" option is selected. A large orange arrow points from the top text "THE ROLE IS NOW CREATED" down to the "aws-elasticbeanstalk-service-role" entry in the list.

You can close this tab now and go to the previous tab

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings
- Root access management
- Temporary delegation requests
- New

Role aws-elasticbeanstalk-service-role created.

View role X

Roles (3) 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.

Role name	Trusted entities
aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk
AWSServiceRoleForSupport	AWS Service: support (Service-Linked)
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked)

Search

Roles Anywhere Info

Manage

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

THE ROLE IS NOW CREATED

aws | Search [Alt+S] | Maryam Tarek

Elastic Beanstalk > Create 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

You find the role you just created available for choosing

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
Choose an IAM role for Elastic Beanstalk to assume as a service role. The IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role [C](#) [Create role ↗](#)

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

Choose a instance profile [C](#) [Create role ↗](#)

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

Choose a key pair [C](#)

If not you can refresh here

Now with the second role, same as the previous

Cancel Skip to review Previous Next

The screenshot shows the 'Create environment' wizard for Elastic Beanstalk. The user has just created a service role, which is now listed in the 'Service role' dropdown. The 'EC2 instance profile' section is highlighted with an orange circle around its 'Create role' button. The text 'If not you can refresh here' is overlaid on the page, pointing to this button. The text 'Now with the second role, same as the previous' is also overlaid, indicating that the user can now proceed with the setup.

CONFIGURE SERVICE ACCESS AND IAM USERS

1 SELECT THE TYPE

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 other accounts to assume this account.

2 THEN THE USE CASE

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case
Elastic Beanstalk

Choose a use case for the specified service.

Use case

Elastic Beanstalk - Compute
Allows your environment's EC2 instances to perform operations required for your application.

Elastic Beanstalk - Management
Allows access to other AWS service resources that are required to create and manage environments.

3 THEN THE PERMISSIONS

Add permissions Info

Permissions policies (3) Info
The type of role that you selected requires the following policy.

Policy name	Type
AWS-ElasticBeanstalk-Multicontainer-Docker	AWS managed
AWS-ElasticBeanstalk-Web-Tier	AWS managed
AWS-ElasticBeanstalk-Worker-Tier	AWS managed

▶ Set permissions boundary - optional

Default is fine, click next

Cancel Previous Next

REVIEW THE IAM USER THEN CLICK CREATE ROLE

THE ROLE IS NOW CREATED

You can close this tab now and go to the previous tab

Identity and Access Management (IAM)

Role aws-elasticbeanstalk-ec2-role created.

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.

Role name	Trusted entities
aws-elasticbeanstalk-ec2-role	AWS Service: ec2
aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk
AWSServiceRoleForSupport	AWS Service: support (Service-Linked)
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked)

Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings
- Root access management
- Temporary delegation requests
- New

Roles Anywhere Info

Manage

THE ROLE IS NOW CREATED

aws | Search [Alt+S] | Maryam tarek

Elastic Beanstalk > Create environment

You find the role you just created available for choosing

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

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
Choose an IAM role for Elastic Beanstalk to assume as a service role. The IAM role must have the required IAM managed policies.
aws-elasticbeanstalk-service-role [C](#) [Create role ↗](#)

EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.
aws-elasticbeanstalk-ec2-role [C](#) [Create role ↗](#)

EC2 key pair - optional
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more ↗](#)
Choose a key pair [C](#)

If not you can refresh here

Now click skip so elastic beanstalk will take care of the rest

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

The screenshot shows the 'Create environment' wizard for Elastic Beanstalk. The 'Service role' dropdown contains 'aws-elasticbeanstalk-service-role' and has a 'Create role' button next to it. The 'EC2 instance profile' dropdown contains 'aws-elasticbeanstalk-ec2-role' and also has a 'Create role' button next to it. At the bottom, there are buttons for 'Cancel', 'Skip to review' (which is highlighted with an orange oval), 'Previous', and 'Next'.

REVIEW EVERYTHING WE JUST DID

Step 1: Configure environment

Environment information

- Environment tier: Web server environment
- Environment name: LaravelSocialMedia
- Platform: arn:aws:elasticbeanstalk:2011-08-15:v1:aws-elasticbeanstalk-1.0

Step 2: Configure service access

Service access

Configure the service role and AWS key pair to securely log in to your instances.

Service role

arn:aws:iam::081523437812:role/elasticbeanstalk-service-role

Step 3: Set up networking, database, and tags

Networking, database, and tags

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.

Tags

Key

Step 4: Configure instance traffic and scaling

Instance traffic and scaling

Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic.

Instances

IMDSv1
Disabled

Capacity

Environment type: Single instance

On-demand above base: 70

Processor type: x86_64

Step 5: Configure updates, monitoring, and logging

Updates, monitoring, and logging

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: enhanced

Cloudwatch custom metrics - instance: —

Cloudwatch custom metrics - environment: —

Log streaming

Retention: 7

Lifecycle: false

Updates

Managed updates: Enabled

Deployment batch size: 100

Deployment batch size type: Percentage

Command timeout: 600

Deployment policy: AllAtOnce

Health threshold: Ok

NOW JUST CLICK CREATE

Logs retention

Rotate logs

Update level

Disabled

minor

Play enabled

Enabled

Environment properties

Source

▼ | Key

▲ | Value

▼ |

No environment properties

There are no environment properties defined

Cancel

Previous

Create

NOW JUST WAIT UNTIL IT SAYS HEALTH IS OK

Beanstalk > Environments > LaravelSocialMediaApp-env

beanstalk < Environment successfully launched. X

LaravelSocialMediaApp-env Info

Actions  Upload and deploy

Environment overview

Health  Ok

Domain laravel-social-media-v1.eu-north-1.elasticbeanstalk.com ↗

Environment ID  e-ivzmc36bc3

Application name LaravelSocialMediaApp

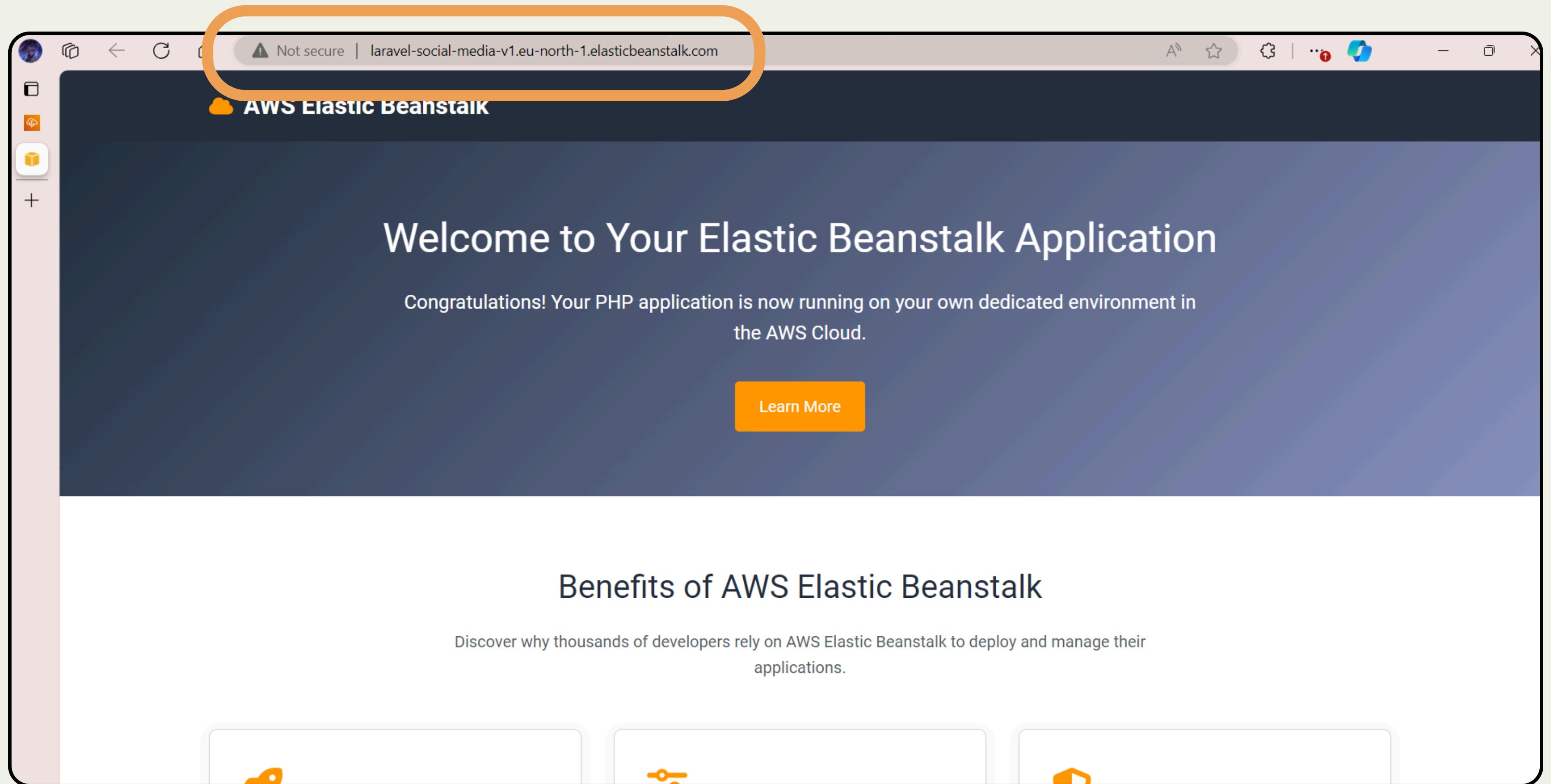
Platform

Platform PHP 8.4 running on 64bit Amazon Linux 2023/4.8.0

Running version 

Platform state

NOW YOU CAN FIND YOUR WEBSITE AT YOUR CHOOSEN DOMAIN



CONCLUSION

The Elastic Beanstalk service makes it easier and faster to deploy your website, but it might not be the best option if you want more control and specialization

**ANY
QUESTIONS?**

THANK YOU