

Assignment – Elastic Beanstalk

Problem Statement:

You work for XYZ Corporation. Your corporation wants to launch a new web-based application and they do not want their servers to be running all the time. It should also be managed by AWS. Implement suitable solutions.

Tasks To Be Performed:

1. Create an Elastic Beanstalk environment with the runtime as PHP.
2. Upload a simple PHP file to the environment once created.

Create Elastic Beanstalk Environment (PHP Runtime)

Step 1: Login to AWS Console

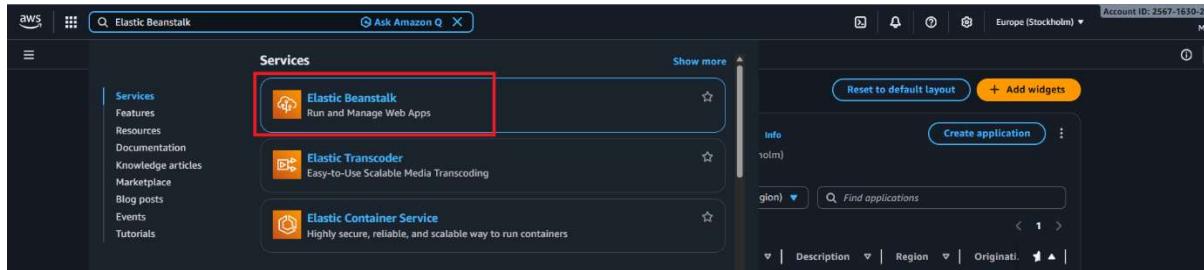
1. Go to - <https://aws.amazon.com>
2. Click **Sign in to the Console**
3. Login with your credentials

The image shows two screenshots of the AWS interface. The top screenshot is the AWS login page at <https://aws.amazon.com>. It features the AWS logo, navigation links like 'Discover AWS', 'Products', 'Solutions', 'Pricing', and 'Resources'. A red box highlights the 'Sign in to console' button. The bottom screenshot is the 'Console Home' page. It shows a sidebar with 'Recently visited' services: EC2, Lambda, CloudWatch, Simple Queue Service, IAM, CloudFormation, Amazon Simple Email Service, and Simple Notification Service. The main area displays an 'Applications' section with a 'Create application' button and a note that no applications have been created yet. The status bar at the bottom right indicates the account ID is 2567-1630-2785 and the region is Europe (Stockholm).

Assignment – Elastic Beanstalk

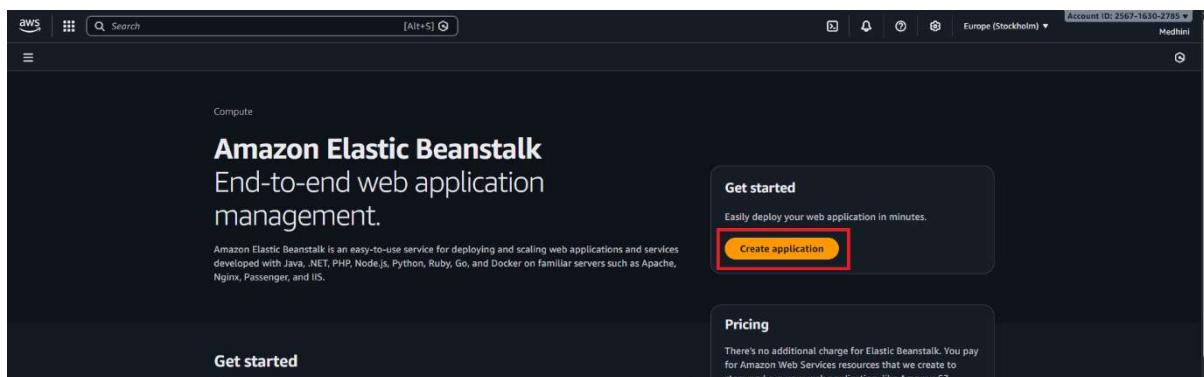
Step 2: Open Elastic Beanstalk

1. In the **Search bar**, type **Elastic Beanstalk**
2. Click **Elastic Beanstalk**



Step 3: Create Application

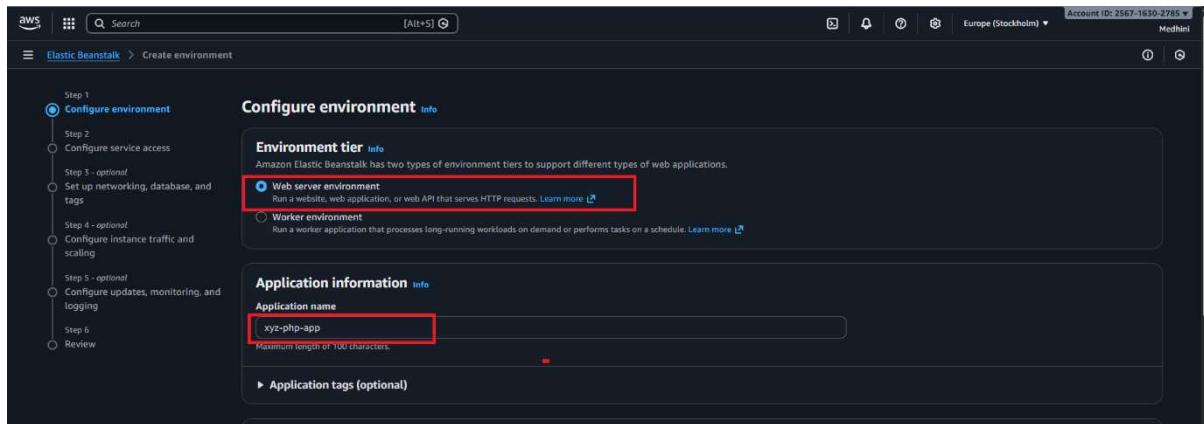
1. Click **Create application**
2. Fill details:
 - **Application name:** xyz-php-app
 - **Description (optional):** PHP Web Application
3. Click **Next**



Step 4: Environment Settings

1. **Environment tier - Select Web server environment**
2. Click **Next**

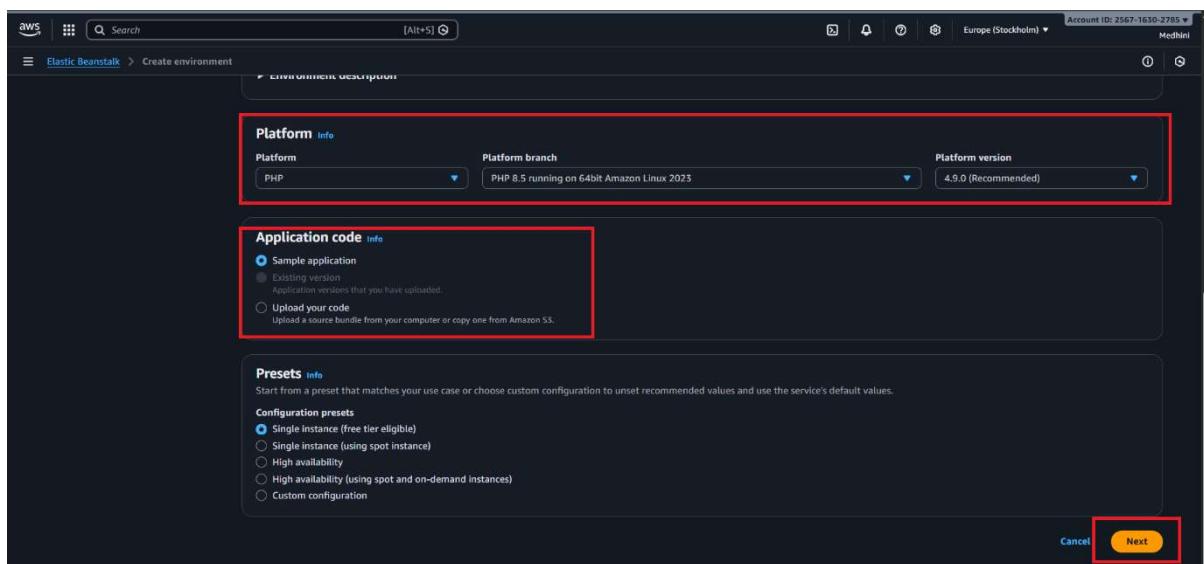
Assignment – Elastic Beanstalk



Step 5: Platform Configuration

- Platform - Select PHP**
- Platform branch - PHP 8.x (default)**
- Platform version - Keep default**
- Application code - Select Sample application**
 - (We will upload our own PHP file later)

Click **Next**



Step 6: Configure Service Access

- Service role - Select Create and use new service role**
- EC2 key pair - Optional (can skip if not required)**
- Click **Next****

Assignment – Elastic Beanstalk

The screenshot shows the 'Configure service access' step of the 'Create environment' wizard. It includes sections for 'Service access', 'EC2 instance profile', and 'EC2 key pair'. A red box highlights the 'Create role' button under 'Service access'.

The screenshot shows the 'Select trusted entity' step of the 'Create role' wizard. It lists options for 'AWS service', 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. The 'AWS service' option is selected. A red box highlights the 'Create role' button at the bottom right.

The screenshot shows the 'Step 2: Add permissions' step of the 'Create role' wizard. It displays a JSON policy document and a table of attached permissions. A red box highlights the 'Edit' button in the top right corner of the permissions summary area.

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

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

Assignment – Elastic Beanstalk

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](#)

EC2 instance profile

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

`aws-elasticbeanstalk-service-role`

EC2 key pair - optional

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

`aws-elasticbeanstalk-ec2-role`

Create role

Choose a key pair

Cancel **Skip to review** **Previous** **Next**

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](#)

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-0ca7f3ca34b87d981 | [10.0.0.16] | project-vpc`

Create VPC

Public IP address

Assign a public IP address to the Amazon EC2 instances in your environment.

Enable

Instance subnets

Filter instance subnets	Availability Zone	Subnet	CIDR	Name
<input type="checkbox"/>	eu-north-1b	subnet-08709ae5a82b99570	10.0.144.0/20	project-subnet-private2-eu-north-1b
<input checked="" type="checkbox"/>	eu-north-1b	subnet-0cdcaf9452578e6	10.0.16.0/20	project-subnet-public2-eu-north-1b
<input checked="" type="checkbox"/>	eu-north-1a	subnet-0c3182213aa113433	10.0.0.0/20	project-subnet-public1-eu-north-1a
<input type="checkbox"/>	eu-north-1a	subnet-0fe39a3df6a385461	10.0.128.0/20	project-subnet-private1-eu-north-1a

Database [Info](#)

Environment successfully launched.

Xyz-php-app-env [Info](#)

Actions [Upload and deploy](#)

Environment overview

Health Pending

Domain -

Environment ID e-jhnqfkdyx

Application name xyz-php-app

Events [Events \(8\)](#) [Info](#)

Platform

Platform PHP 8.5 running on 64bit Amazon Linux 2023/4.9.0

Running version -

Platform state Supported

TASK 2: Upload a Simple PHP File

Step 9: Create PHP File on Your Laptop

1. Open Notepad
2. Paste this code:

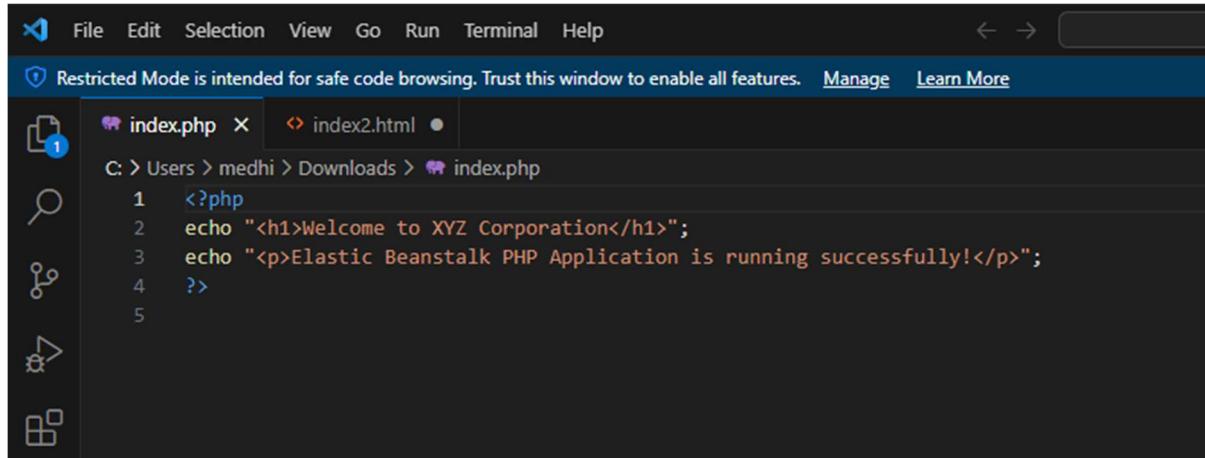
```
<?php
```

```
echo "<h1>Welcome to XYZ Corporation</h1>";
```

Assignment – Elastic Beanstalk

```
echo "<p>Elastic Beanstalk PHP Application is running successfully!</p>";
```

```
?>
```



A screenshot of a code editor window titled "index.php". The window has a dark theme with light-colored syntax highlighting. The code in the editor is:

```
C: > Users > medhi > Downloads > index.php
1 <?php
2 echo "<h1>Welcome to XYZ Corporation</h1>";
3 echo "<p>Elastic Beanstalk PHP Application is running successfully!</p>";
4 ?>
5
```

Step 10: Create ZIP File

1. Right-click index.php
2. Select **Send to - Compressed (zipped) folder**
3. Name it:

Important:

- ZIP should contain **only index.php**
- NOT a folder inside ZIP

Step 11: Upload PHP File to Elastic Beanstalk

1. Go back to **Elastic Beanstalk - Your Environment**
2. Click **Upload and deploy**
3. Click **Choose file**
4. Select php-app.zip
5. **Version label:** v1
6. Click **Deploy**

Assignment – Elastic Beanstalk

The screenshot shows the AWS Elastic Beanstalk console. In the top right corner, there is a green banner that says "Environment successfully launched." Below this, the "Actions" dropdown menu is open, and the "Upload and deploy" button is highlighted with a red box.

This screenshot shows the "Upload and deploy" dialog box. It has a red box around the "Choose file" button where a file named "php-app.zip" is selected. At the bottom right of the dialog box, the "Deploy" button is highlighted with a red box.

The screenshot shows the AWS Elastic Beanstalk console again. A green banner at the top indicates that the file was uploaded to S3 and deployment started. The "Actions" dropdown menu is open, and the "Upload and deploy" button is visible. The "Events" tab is selected in the main interface.

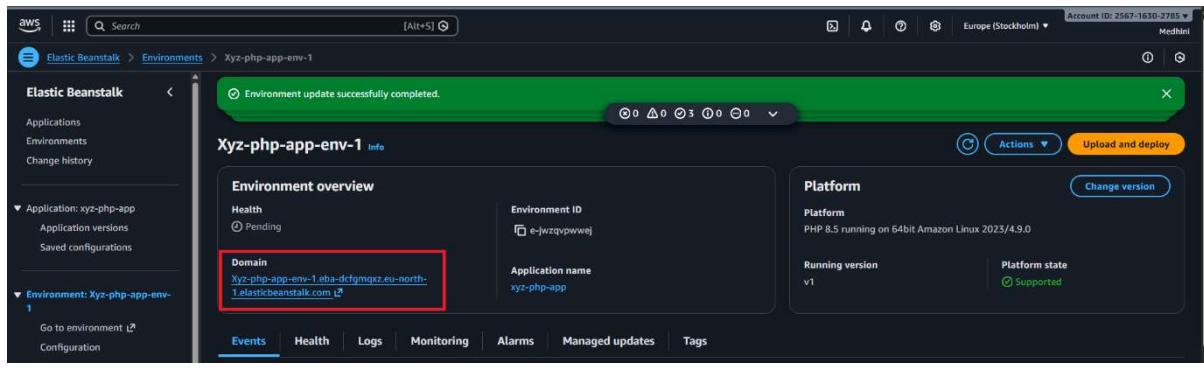
Step 12: Verify Output

1. Wait 1–2 minutes for deployment
2. Open **Environment URL** again
3. You should see:

Welcome to XYZ Corporation

Elastic Beanstalk PHP Application is running successfully!

Assignment – Elastic Beanstalk



Elastic Beanstalk is used to deploy a PHP web application without managing servers. AWS automatically provisions EC2, load balancer, and scaling. A simple PHP file was uploaded to verify successful deployment.