ANISH ADNANI

D17B - 01

CCL

(1)

EXPERIMENT 9

Aim: Implementation of PAAS using Google App Engine / AWS / Azure.

Theory:

Google App Engine

- Google App Engine is a Platform as a Service (PaaS) product that provides web app developers and enterprises with access to Google's scalable hosting and tier 1 internet service.
- The App Engine requires that apps be written in Java or Python, store data in Google BigTable and use the Google Query language. Non-compliant applications require modification to use App Engine
- Google App Engine provides more infrastructure than other scalable hosting services such as Amazon Elastic Compute Cloud (EC2). The App Engine also eliminates some system administration and developmental tasks to make it easier to write scalable applications
- Google App Engine is able to dynamically scale up and down to meet the demands of your user traffic regardless of variability or size.
- Google App Engine also integrates a number of additional features that you can take advantage of to ensure your application remains highly available and responsive.

Datastore is a schemaless object store with scalable storage. An SQL like querying language, and an easy-to-use API built into the GAE sdk.

memcache provides lightning quick in-memory caching that not only improves application speed, but also can reduce costs by reducing the number of calls to outside services.

AWS Elastic Beanstalk

- AWS 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.
- You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring. At the same time, you retain full control over AWS resources powering your application and can access the underlying resources at any time.

Benefits of AWS Elastic Beanstalk :

Fast and simple to begin : Elastic beanstalk is the fastest and simplest way to deploy your application on AWS. Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring.

Developer productivity: Elastic Beanstalk provisions and operates the infrastructure and manages the application stack (platform) for you, so you dont have to spend the time or develop the expertise

Complete resource control: You have the freedom to select the AWS resource, such as Amazon EC2 instance type, that are optimal for your application.

Impossible to outgrow: Elastic Beanstalk automatically scales your application up and down based on your application's specific need using easily adjustable Auto Scaling settings.

For example, you can use CPU utilization metrics to trigger Auto scaling actions. With Elastic Beanstalk, your application can handle peaks in workload or traffic while minimizing your costs

Conclusion:

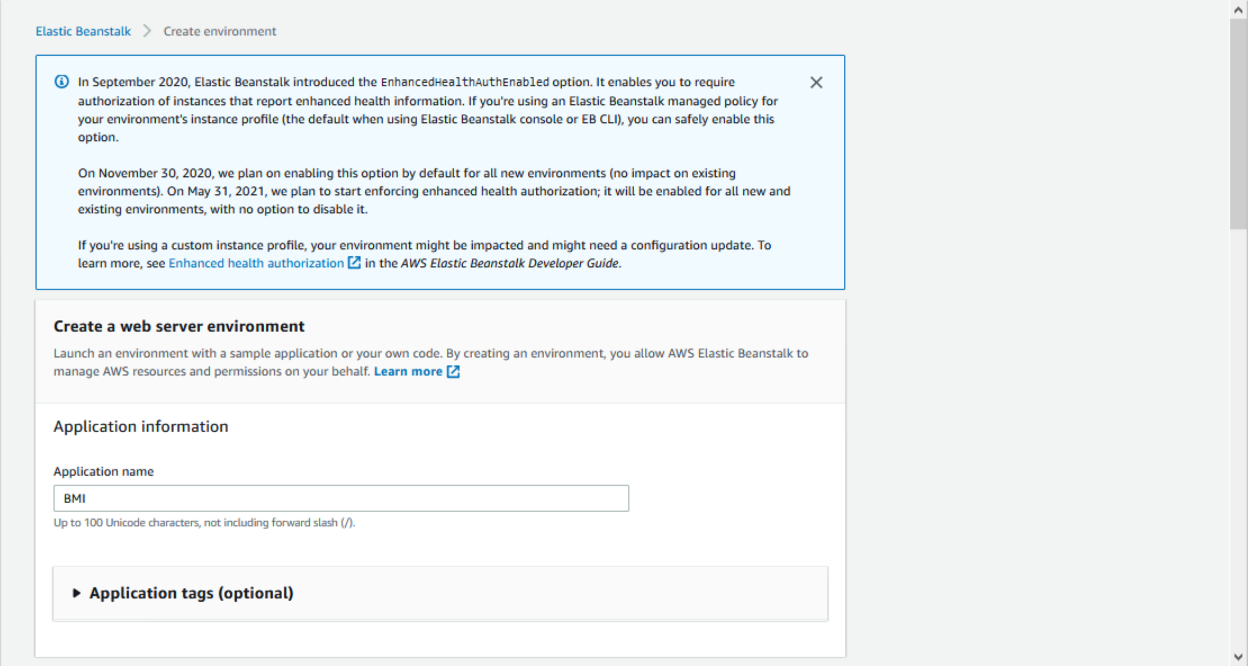
Hence we have successfully developed and deployed a web application on Amazon AWS elastic Beanstalk.

Code:

```
<!DOCTYPE html>
<html>

<head>
    <!-- Include JS files -->
    <script>
            window.onload = () => {
    let button = document.querySelector("#btn");

    // Function for calculating BMI
    button.addEventListener("click", calculateBMI);
};

function calculateBMI() {

    /* Getting input from user into height variable.
    Input is string so typecasting is necessary. */
    let height = parseInt(document
                    .querySelector("#height").value);

    /* Getting input from user into weight variable.
    Input is string so typecasting is necessary.*/
    let weight = parseInt(document
                    .querySelector("#weight").value);

    let result = document.querySelector("#result");

    // Checking the user providing a proper
    // value or not
    if (height === "" || isNaN(height))
            result.innerHTML = "Provide a valid Height!";

    else if (weight === "" || isNaN(weight))
            result.innerHTML = "Provide a valid Weight!";

    // If both input is valid, calculate the bmi
    else {

            // Fixing upto 2 decimal places
            let bmi = (weight / ((height * height)
                                                / 10000)).toFixed(2);

            // Dividing as per the bmi conditions
```

```
          if (bmi < 18.6) result.innerHTML =
                    `Under Weight : <span>${bmi}</span>`;

          else if (bmi >= 18.6 && bmi < 24.9)
                    result.innerHTML =
                              `Normal : <span>${bmi}</span>`;

          else result.innerHTML =
                    `Over Weight : <span>${bmi}</span>`;
   }
}

   </script>
</head>

<body>
   <div class="container">
          <h1>BMI Calculator : Developed by Anish Adnani</h1>

          <!-- Option for providing height
                    and weight to the user-->
          <p>Height (in cm)</p>

          <input type="text" id="height">

          <p>Weight (in kg)</p>

          <input type="text" id="weight">

          <button id="btn">Calculate</button>

          <div id="result"></div>
   </div>
</body>

</html>
```

Steps in deploying code to Elastic Beanstalk



Elastic Beanstalk > Create environment

In September 2020, Elastic Beanstalk introduced the `EnhancedHealthAuthEnabled` option. It enables you to require authorization of instances that report enhanced health information. If you're using an Elastic Beanstalk managed policy for your environment's instance profile (the default when using Elastic Beanstalk console or EB CLI), you can safely enable this option.

On November 30, 2020, we plan on enabling this option by default for all new environments (no impact on existing environments). On May 31, 2021, we plan to start enforcing enhanced health authorization; it will be enabled for all new and existing environments, with no option to disable it.

If you're using a custom instance profile, your environment might be impacted and might need a configuration update. To learn more, see Enhanced health authorization in the *AWS Elastic Beanstalk Developer Guide*.

Create a web server environment

Launch an environment with a sample application or your own code. By creating an environment, you allow AWS Elastic Beanstalk to manage AWS resources and permissions on your behalf. **Learn more**

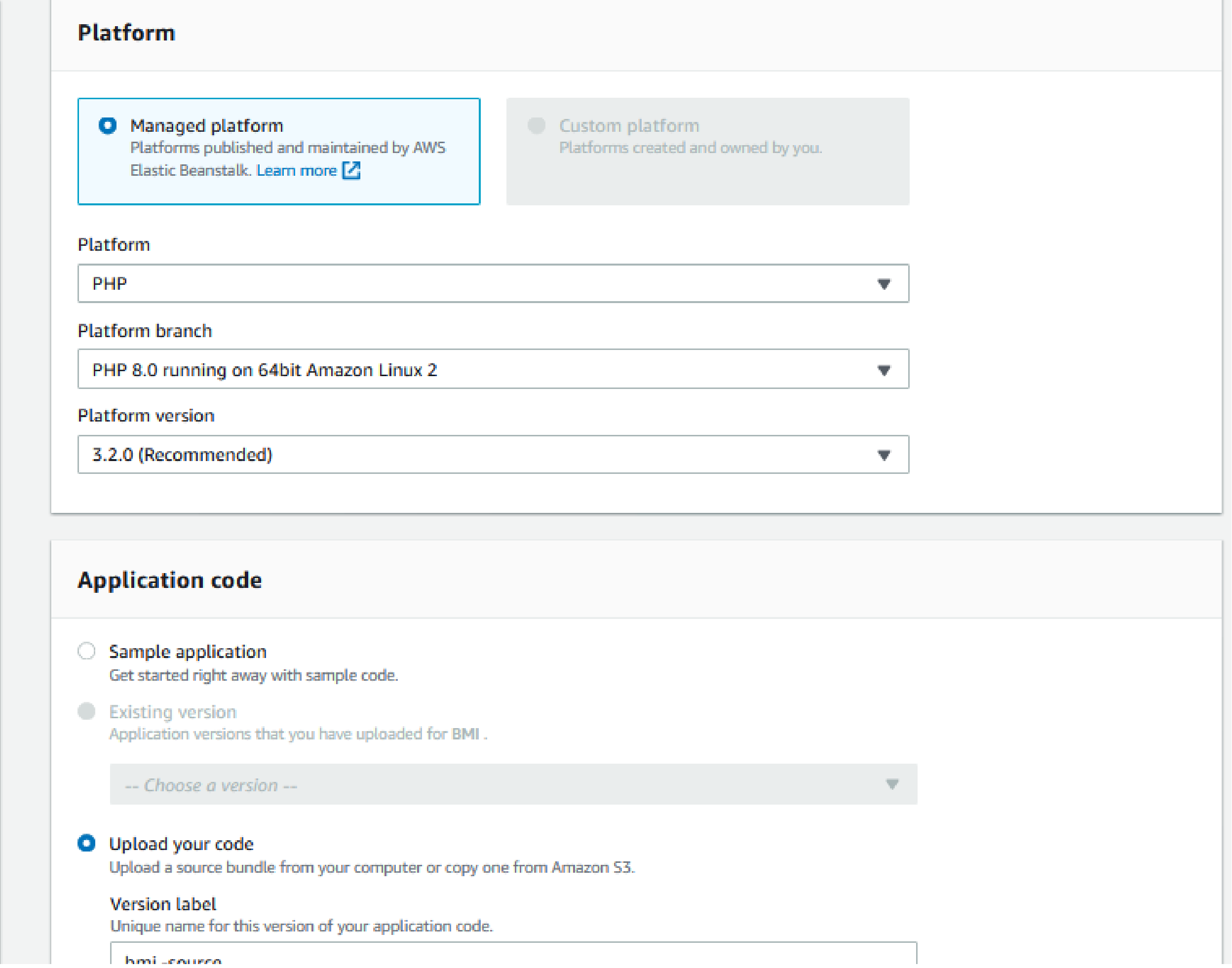
Application information

Application name

BMI

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

▸ **Application tags (optional)**



Platform

● Managed platform
Platforms published and maintained by AWS Elastic Beanstalk. Learn more

○ Custom platform
Platforms created and owned by you.

Platform

PHP ▼

Platform branch

PHP 8.0 running on 64bit Amazon Linux 2 ▼

Platform version

3.2.0 (Recommended) ▼

Application code

○ Sample application
Get started right away with sample code.

○ Existing version
Application versions that you have uploaded for BMI .

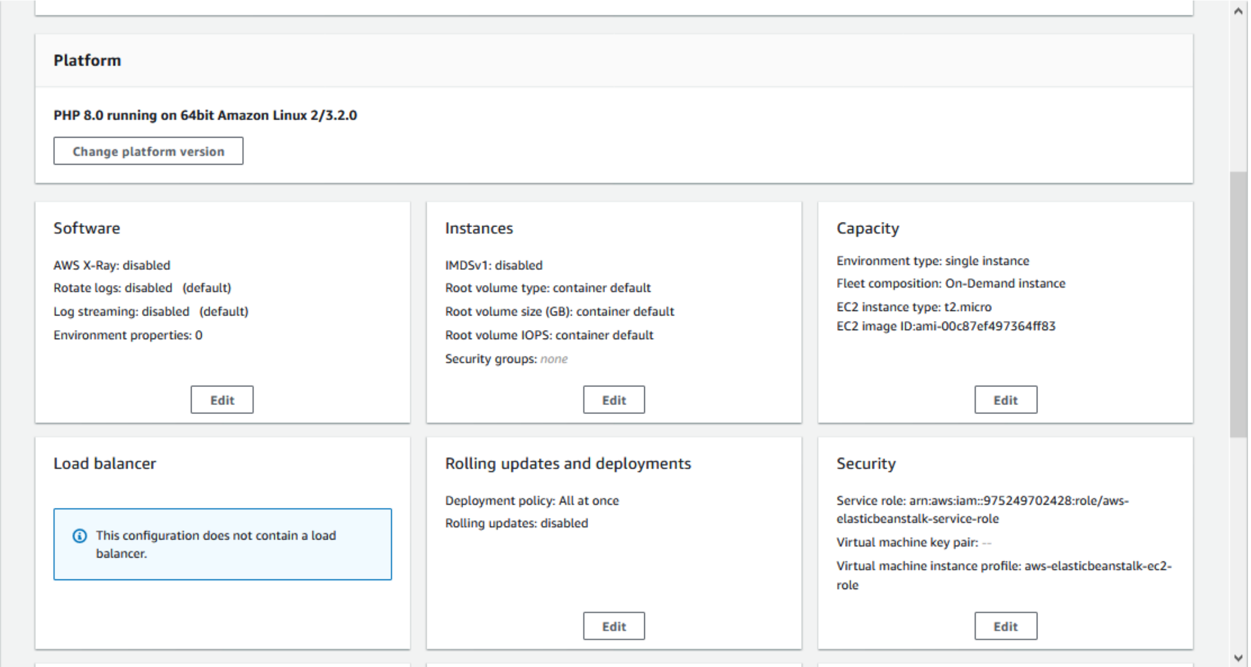
-- Choose a version -- ▼

● 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.

bmi -source

Configuration



Platform

PHP 8.0 running on 64bit Amazon Linux 2/3.2.0

Software

AWS X-Ray: disabled
Rotate logs: disabled (default)
Log streaming: disabled (default)
Environment properties: 0

Instances

IMDSv1: disabled
Root volume type: container default
Root volume size (GB): container default
Root volume IOPS: container default
Security groups: *none*

Capacity

Environment type: single instance
Fleet composition: On-Demand instance
EC2 instance type: t2.micro
EC2 image ID:ami-00c87ef497364ff83

Load balancer

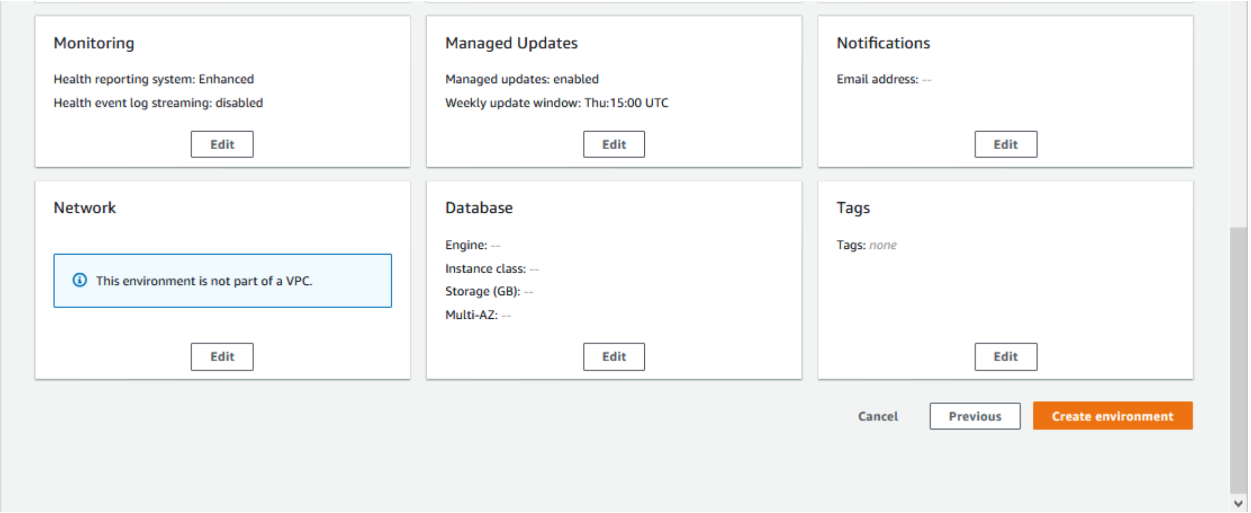
This configuration does not contain a load balancer.

Rolling updates and deployments

Deployment policy: All at once
Rolling updates: disabled

Security

Service role: arn:aws:iam::975249702428:role/aws-elasticbeanstalk-service-role
Virtual machine key pair: --
Virtual machine instance profile: aws-elasticbeanstalk-ec2-role



Monitoring

Health reporting system: Enhanced
Health event log streaming: disabled

Managed Updates

Managed updates: enabled
Weekly update window: Thu:15:00 UTC

Notifications

Email address: --

Network

This environment is not part of a VPC.

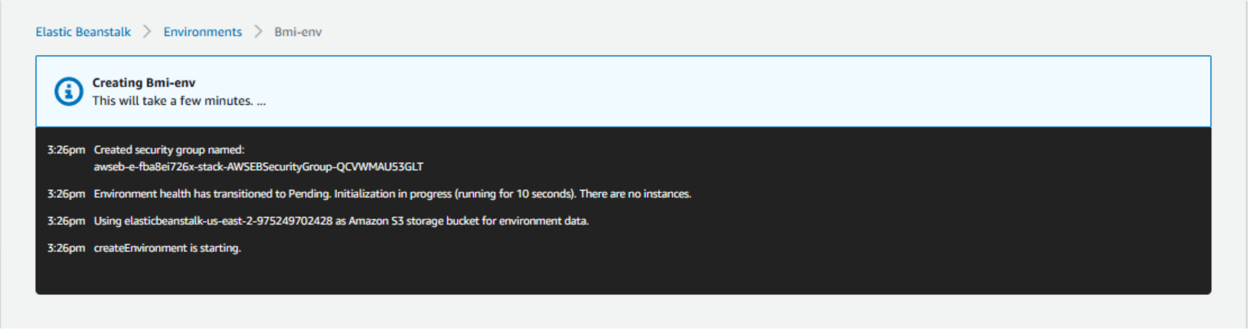
Database

Engine: --
Instance class: --
Storage (GB): --
Multi-AZ: --

Tags

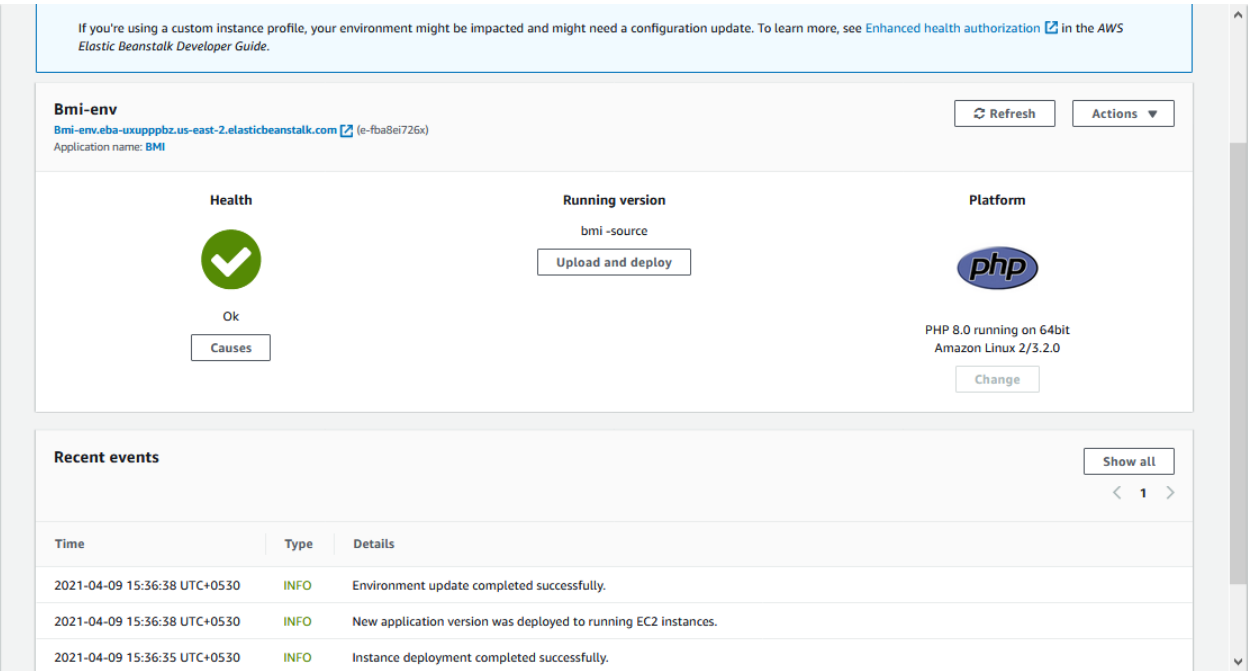
Tags: *none*

Creating and deploying the environment



Creating Bmi-env
This will take a few minutes. ...

```
3:26pm  Created security group named:
        awseb-e-fba8ei726x-stack-AWSEBSecurityGroup-QCVWMAU53GLT

3:26pm  Environment health has transitioned to Pending. Initialization in progress (running for 10 seconds). There are no instances.

3:26pm  Using elasticbeanstalk-us-east-2-975249702428 as Amazon S3 storage bucket for environment data.

3:26pm  createEnvironment is starting.
```



If you're using a custom instance profile, your environment might be impacted and might need a configuration update. To learn more, see Enhanced health authorization in the *AWS Elastic Beanstalk Developer Guide.*

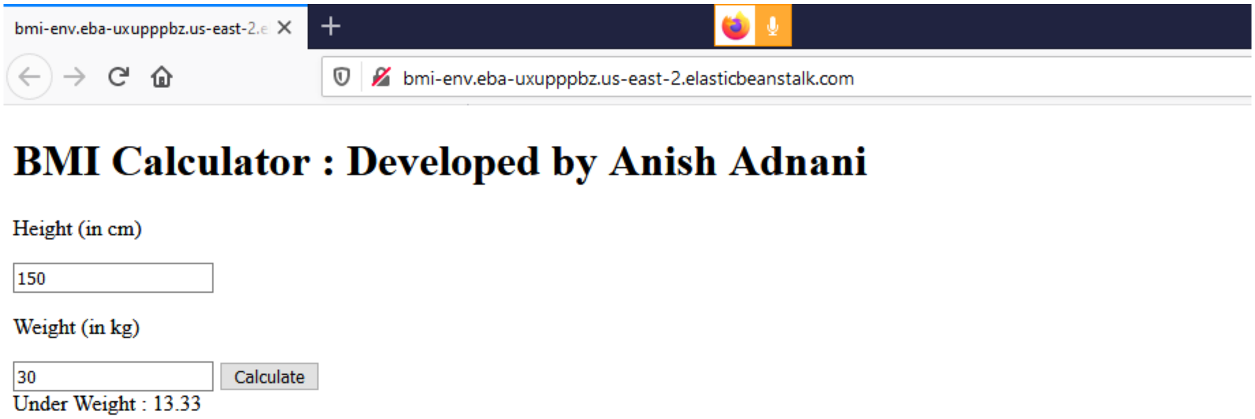
Bmi-env
Bmi-env.eba-uxupppbz.us-east-2.elasticbeanstalk.com (e-fba8ei726x)
Application name: BMI

Health	Running version	Platform
Ok	bmi -source	PHP 8.0 running on 64bit Amazon Linux 2/3.2.0

Recent events

Time	Type	Details
2021-04-09 15:36:38 UTC+0530	INFO	Environment update completed successfully.
2021-04-09 15:36:38 UTC+0530	INFO	New application version was deployed to running EC2 instances.
2021-04-09 15:36:35 UTC+0530	INFO	Instance deployment completed successfully.

Hosted Website:



bmi-env.eba-uxupppbz.us-east-2.elasticbeanstalk.com

BMI Calculator : Developed by Anish Adnani

Height (in cm)

150

Weight (in kg)

30 Calculate

Under Weight : 13.33