Task 1

Objective

Deploy a website on Amazon Web Services (AWS) using AWS Elastic Beanstalk.

Project overview

The goal of this project is to deploy a web application on Amazon Web Services (AWS) using **AWS Elastic Beanstalk**. Elastic Beanstalk is a Platform as a Service (PaaS) that handles deployment, scaling, and monitoring of web applications. In this project, a simple web application (developed using frameworks such as Flask or Node.js) is packaged and uploaded to AWS Elastic Beanstalk for automated deployment and hosting.

By completing this project, users understand how to use cloud platforms to host dynamic websites with high availability and scalability, while AWS handles most of the infrastructure management automatically.

**Key Features**

1. **Simple Deployment Process:**  
Upload code and AWS automatically handles deployment steps like provisioning and load balancing.

2. **Auto Scaling and Load Balancing:**  
Elastic Beanstalk automatically scales the application up or down depending on traffic.

3. **Multi-language Support:**  
Supports platforms such as Python (Flask, Django), Node.js, Java, PHP, Ruby, .NET, and more.

4. **Built-in Monitoring and Logging:**  
Integrated with AWS CloudWatch for real-time monitoring and log analysis.

5. **Easy Configuration Management:**  
Application settings and environment variables can be configured from the dashboard.

**Services Used**

| **Service** | **Purpose** |
| --- | --- |
| **AWS Elastic Beanstalk** | Main service used to deploy and manage the web application. |
| **Amazon EC2** | Underlying compute resources where the app is hosted. |
| **Amazon S3** | (Optional) Used for storing application packages and static files. |
| **Amazon RDS** | (Optional) Used for database integration with the application. |
| **AWS IAM** | Controls access and permissions for AWS resources. |
| **AWS CloudWatch** | Provides logs and monitoring metrics for the deployed application. |