IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15358

Build a Python Flask App and Deploy Using AWS Elastic Beanstalk

Vishaal S[1], Mr. S.S. SaravanaKumar[2]

III BCA, Department of Computer Applications, Sri Ramakrishna College of Arts & Science, Coimbatore, India[1]

Assistant Professor, Department of Computer Applications, Sri Ramakrishna College of Arts & Science, Coimbatore, India[2]

Abstract: Web applications have become an essential part of modern digital services, enabling users to access systems and information from anywhere through the internet. With the increasing demand for scalable and efficient applications, cloud computing platforms provide reliable environments for deploying and managing web applications. This project focuses on the development and deployment of a web application using the Python Flask framework and Amazon Web Services (AWS) Elastic Beanstalk. The application is developed using Flask due to its lightweight structure, flexibility, and ease of integration with various web technologies. After development, the application is deployed on AWS Elastic Beanstalk, which simplifies the process of managing infrastructure, scaling resources, and maintaining application availability. The system demonstrates how cloud-based deployment can reduce configuration complexity while improving performance and scalability. Through this project, users can understand the complete workflow of designing, developing, and deploying a web application in a cloud environment. The results show that AWS Elastic Beanstalk provides an efficient platform for hosting Flask applications with minimal manual configuration. This project highlights the practical implementation of cloud deployment techniques and demonstrates the advantages of using modern cloud services for web application hosting and management.

Keywords: Python Flask, Web Application Development, Cloud Computing, AWS Elastic Beanstalk, Cloud Deployment, Scalable Web Applications.

I. INTRODUCTION

In recent years, web applications have become an important component of modern information systems. Businesses, educational institutions, and organizations increasingly rely on web-based platforms to deliver services, manage data, and interact with users across different locations. With the rapid growth of internet technologies, the demand for scalable, efficient, and easily deployable web applications has significantly increased. Developers now require frameworks and platforms that simplify the process of building and deploying applications while ensuring reliability and performance.

Python has emerged as one of the most popular programming languages for web development due to its simplicity, flexibility, and extensive ecosystem of libraries and frameworks. Among these frameworks, Flask has gained significant attention because of its lightweight structure and ease of use. Flask allows developers to quickly build web applications while maintaining full control over the application architecture. It provides essential tools and features required for web development without enforcing complex structures, making it suitable for both small and large-scale applications.

In addition to application development, the deployment of web applications is a crucial step in making them accessible to users. Traditionally, deploying web applications required complex server configuration and manual management of infrastructure. However, the introduction of cloud computing platforms has transformed this process. Cloud platforms provide scalable and flexible environments that allow developers to deploy, manage, and maintain applications efficiently without worrying about hardware or infrastructure management.

Amazon Web Services (AWS) is one of the leading cloud service providers that offers a wide range of services for application development and deployment. AWS Elastic Beanstalk is a platform-as-a-service (PaaS) solution that simplifies the deployment process by automatically handling resource provisioning, load balancing, scaling, and

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15358

application monitoring. Developers can focus on writing application code while the platform manages the underlying infrastructure.

The main objective of this project is to design and develop a web application using the Python Flask framework and deploy it using AWS Elastic Beanstalk. The project demonstrates the practical implementation of modern web development techniques combined with cloud-based deployment solutions. By integrating Flask with AWS Elastic Beanstalk, the system provides a simple and efficient approach for hosting web applications in a scalable cloud environment.

This project also aims to help developers and students understand the complete workflow of web application development, including coding, testing, and cloud deployment. The results highlight the benefits of using cloud platforms such as improved scalability, reduced deployment complexity, and better resource management. Through this implementation, the project illustrates how modern cloud technologies can enhance the efficiency and accessibility of web applications.

II. RELATED WORK

Several studies have explored the use of lightweight web frameworks and cloud platforms for developing and deploying web applications. Python Flask is widely used in modern web development because of its simplicity, flexibility, and ability to quickly build scalable applications. Researchers have highlighted that Flask allows developers to create efficient backend systems while maintaining control over application design.

Cloud computing platforms have also transformed the way web applications are deployed and managed. Services such as **AWS Elastic Beanstalk** provide automated infrastructure management, including resource provisioning, load balancing, and application monitoring. This reduces the complexity of deployment and allows developers to focus mainly on application development.

Previous works demonstrate that integrating Flask with cloud platforms like AWS improves scalability, performance, and ease of deployment. Inspired by these approaches, this project implements a web application using the Flask framework and deploys it on AWS Elastic Beanstalk to demonstrate a practical and efficient cloud-based deployment method.

III. SYSTEM ARCHITECTURE AND DESIGN

The proposed system is designed as a web-based application developed using the Python Flask framework and deployed on a cloud environment using Amazon Web Services (AWS) Elastic Beanstalk. The architecture follows a client–server model where users interact with the application through a web browser while the backend processes requests and manages application logic.

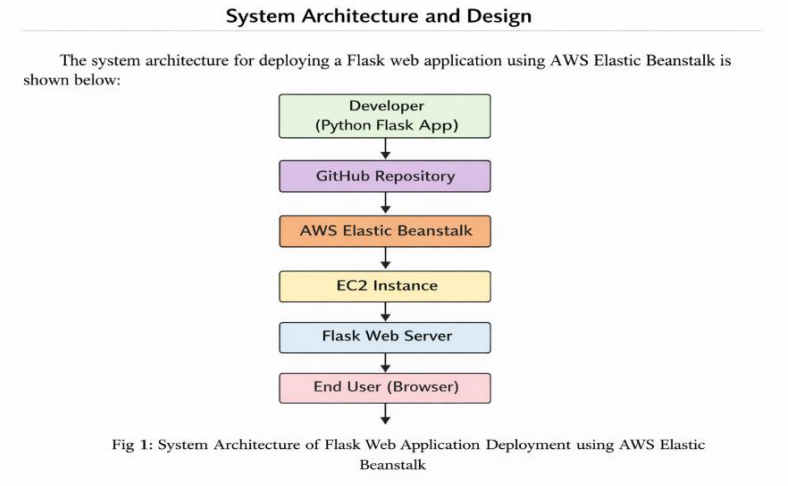

Fig 1: System Architecture of Flask Web Application Deployment using AWS Elastic Beanstalk

In this architecture, the client side represents the user interface accessed through a web browser. Users send requests to the web application through HTTP protocols. These requests are received by the Flask-based web server, which processes the input, performs necessary operations, and generates appropriate responses.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026
DOI: 10.17148/IJARCCE.2026.15358

The backend of the system is implemented using the Flask framework, which handles routing, request processing, and application logic. Flask acts as the intermediary between the user interface and the server environment. The application is structured in a modular manner to ensure maintainability and scalability.

For deployment, AWS Elastic Beanstalk is used as the cloud platform that manages the infrastructure required to run the web application. It automatically handles server provisioning, application deployment, load balancing, and monitoring. This reduces the complexity of manual server configuration and allows the application to run efficiently in a scalable cloud environment.

The overall system architecture ensures smooth communication between the user interface, application logic, and cloud infrastructure. This design enables reliable performance, easy deployment, and efficient management of the web application in a cloud-based environment.

IV. IMPLEMENTATION AND RESULTS

1. Methodology

Explain the overall approach you followed to build and deploy the application. Briefly describe the development steps: designing the Flask application, preparing the project files, configuring dependencies, and deploying through the AWS environment.

2. Implementation

Describe how the system was actually built. Mention the programming language (Python), the Flask framework, and the configuration of the AWS environment. You can also explain how the application routes, templates, and backend logic were implemented.

3. Tools and Technologies Used

List the main technologies involved in the project. For example:

- Python
- Flask Framework
- AWS Elastic Beanstalk
- Amazon EC2
- GitHub
- HTML/CSS for the frontend

4. Results and Discussion

Explain the final outcome of your project. Mention that the Flask web application was successfully deployed using AWS Elastic Beanstalk, making it accessible through a web browser. You can also discuss benefits such as scalability, simplified deployment, and efficient cloud management.

5. Advantages of the Proposed System

Briefly highlight the benefits of your approach—for instance:

- Easy deployment through AWS
- Scalable infrastructure
- Reduced manual server configuration
- Efficient management of web applications

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.15358

6. Future Enhancements

Suggest possible improvements. For example:

- Adding advanced user authentication
- Integrating databases or APIs
- Implementing automated monitoring and logging
- Enhancing the user interface

V. SYSTEM MODULES

The proposed system is divided into several modules to ensure efficient development, deployment, and management of the web application. Each module performs a specific function within the system architecture.

A. User Interface Module

This module represents the front-end of the web application where users interact with the system through a web browser. The interface is designed using standard web technologies such as HTML and CSS. It allows users to access the application, submit requests, and view the responses generated by the server.

B. Application Processing Module

This module is implemented using the Python Flask framework. It handles the core functionality of the application, including routing, request handling, and processing user inputs. Flask processes the requests received from the client and generates appropriate responses based on the application logic.

C. Cloud Deployment Module

This module manages the deployment of the web application on the cloud platform. AWS Elastic Beanstalk is used to deploy and run the Flask application in a scalable environment. It automatically handles server provisioning, load balancing, and application monitoring.

D. Server Management Module

This module manages the server environment where the application is hosted. AWS services such as EC2 instances provide the computing resources required to run the web application. The server processes requests and ensures continuous availability of the system.

E. User Access Module

This module allows end users to access the deployed application through a web browser using the application URL. Users can interact with the system from different devices connected to the internet.

This modular design improves the efficiency, scalability, and maintainability of the web application deployed in the cloud environment.

VI. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

The developed web application was successfully implemented using the Python Flask framework and deployed on the AWS Elastic Beanstalk cloud platform. The deployment process involved configuring the application environment, uploading the project files, and automatically provisioning the required resources through AWS services. After deployment, the application was tested through a web browser to ensure that all functionalities were working correctly.

The system was evaluated based on its performance, accessibility, and deployment efficiency. The results showed that AWS Elastic Beanstalk simplifies the process of deploying web applications by automatically managing server configuration, scaling, and monitoring. The application responded efficiently to user requests, demonstrating reliable performance in the cloud environment.

The experimental results also indicate that the integration of Flask with AWS provides a scalable and flexible solution for hosting web applications. Users were able to access the application through the internet without requiring

DOI: 10.17148/IJARCCE.2026.15358

manual server setup. This demonstrates that cloud-based deployment platforms can significantly reduce infrastructure management efforts while improving application availability and performance.

Overall, the implementation confirms that deploying Flask applications using AWS Elastic Beanstalk is an effective approach for developing modern web applications in a cloud computing environment.

VII. CONCLUSION

This project presented the development and deployment of a web application using the Python Flask framework and AWS Elastic Beanstalk. The system demonstrates how cloud platforms can simplify the deployment process while providing scalability and reliability. The successful implementation shows that integrating Flask with AWS is an efficient approach for hosting modern web applications with minimal infrastructure management.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the faculty members and the project guide of the Department of Computer Applications for their valuable guidance and support throughout the completion of this project. Their suggestions and encouragement greatly helped in the successful development and deployment of the web application. The authors also thank the institution for providing the necessary resources and environment to carry out this work successfully.

REFERENCES

[1] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, 2nd ed., O'Reilly Media, 2018.
[2] Amazon Web Services, "AWS Elastic Beanstalk Developer Guide." Available: https://docs.aws.amazon.com/elasticbeanstalk/
[3] Python Software Foundation, "Python Documentation." Available: https://docs.python.org/
[4] P. Barry, *Head First Python*, 2nd ed., O'Reilly Media, 2016.
[5] M. Lutz, *Learning Python*, 5th ed., O'Reilly Media, 2013.
[6] R. Buyya, J. Broberg, and A. Goscinski, *Cloud Computing: Principles and Paradigms*, Wiley, 2011.
[7] T. Erl, R. Puttini, and Z. Mahmood, *Cloud Computing: Concepts, Technology & Architecture*, Prentice Hall, 2013.
[8] Amazon Web Services, "Getting Started with AWS." Available: https://aws.amazon.com/getting-started/

