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.2026.15330

DEPLOY A DOCKERIZED DJANGO APPLICATION ON AWS LIGHTSAIL

Kavya J[1], Dr.C.Daniel Nesakumar[2]

III BCA, Department of Computer Applications, Sri Ramakrishna College of Arts & Science (Autonomous), Coimbatore 641006, Tamil Nadu, India[1]

Associate Professor, Department of Computer Applications, Sri Ramakrishna College of Arts & Science (Autonomous), Coimbatore-641006, Tamil Nadu, India[2]

Abstract: In recent years, containerization and cloud computing have become important technologies for developing and deploying modern web applications. This project presents the deployment of a Dockerized Django web application called Finance Checker on AWS Lightsail. The main objective of this system is to help users manage and analyze their financial activities by tracking income and expenses through a web-based interface.

The application is developed using the Django framework, which provides a secure and scalable environment for building web applications. Docker is used to containerize the application along with all its dependencies, ensuring consistency across different environments and simplifying the deployment process. The Docker image is then deployed on AWS Lightsail, a cloud computing platform that offers easy and cost-effective hosting for applications.

Keywords: Docker, Django Framework, AWS Lightsail, Cloud Computing, Containerization, Web Application Deployment, Finance Checker, Docker Containers, Cloud Hosting, Financial Data Management.

1. INTRODUCTION

In modern software development, cloud computing and containerization technologies play an important role in building and deploying web applications efficiently. Traditional deployment methods often face problems such as dependency conflicts, scalability issues, and complex configuration processes. To overcome these challenges, container technologies like Docker and cloud platforms such as AWS have become widely used.

Docker is a containerization platform that allows developers to package applications along with all required dependencies into a single container. This ensures that the application runs consistently across different environments. Django is a powerful Python-based web framework that helps developers build secure, scalable, and high-performance web applications quickly.

In this project, a web application called Finance Checker is developed using the Django framework. The application allows users to record income and expenses, manage financial data, and generate reports for better financial analysis. Docker is used to containerize the Django application, making it easier to deploy and manage.

The Dockerized application is deployed on AWS Lightsail, a cloud service provided by Amazon that simplifies the process of hosting applications. By combining Django, Docker, and AWS Lightsail, this project demonstrates an efficient and reliable method for deploying web applications in a cloud environment.

Table 1: Advantages of finance checker application

Feature	Description
User authentication	Allows users to securely register and log in to the finance checker system.
Income management	Enables users to add and track different sources of income.
Expense tacking	Allows users to record and manaage daily expenses.
Dashboard analysis	Displays total income,expenses,and financial summary in a dashboard.
Report generation	Generates financial reports to help users understand their financial activities.

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.2026.15330

2. LITERATURE SURVEY

A literature survey is an important step in understanding the existing research and technologies related to the proposed system. Many researchers have studied the use of containerization and cloud computing for deploying web applications efficiently. Container technologies such as Docker have become popular because they allow applications to run consistently across different environments without dependency issues.

Several studies highlight the advantages of using the Django framework for web application development. Django provides built-in security features, rapid development capabilities, and a structured architecture that helps developers build scalable applications. Many financial management systems are developed using web frameworks like Django because they provide reliable database integration and secure user authentication.

Research on cloud computing platforms such as AWS shows that cloud services provide scalable and cost-effective infrastructure for hosting applications. AWS Lightsail, in particular, offers a simple way to deploy and manage web applications without complex configurations. It provides virtual servers, networking, and storage resources that simplify application hosting.

Previous studies also emphasize that combining Docker with cloud platforms improves application portability and simplifies deployment. Docker containers ensure that applications behave the same way in development and production environments. Based on these concepts, the proposed project deploys a Dockerized Django application called Finance Checker on AWS Lightsail to provide an efficient and scalable financial management system.

3. METHODOLOGY

The methodology of this project focuses on developing and deploying a web application called Finance Checker using modern technologies such as Django, Docker, and AWS Lightsail. The main aim is to build a system that helps users manage their financial records and deploy the application efficiently in a cloud environment.

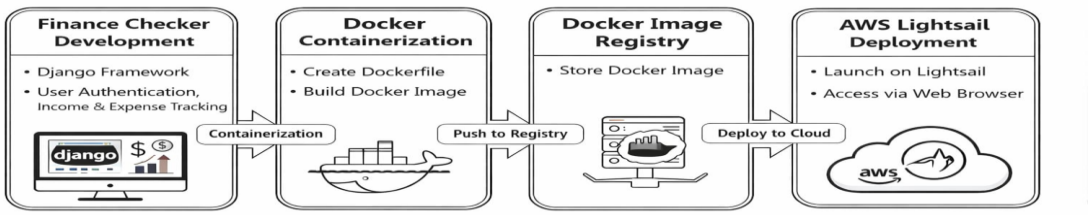
First, the Finance Checker application is developed using the Django framework. The application provides features such as user authentication, income tracking, expense management, and report generation. The system stores financial data in a database, allowing users to easily manage and analyze their financial activities.

After developing the application, Docker is used to containerize the system. A Dockerfile is created to define the environment and dependencies required to run the Django application. This allows the application to run consistently across different systems without configuration issues.

Next, a Docker image is built and stored in a container registry. The Docker image contains the complete application along with all required dependencies.

Finally, the Dockerized application is deployed on AWS Lightsail, which provides a cloud platform for hosting web applications. The Lightsail instance runs the Docker container, allowing users to access the Finance Checker application through a web browser.

3.1 Architectural Design



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.2026.15330

The architectural design of the Finance Checker application explains how different components of the system work together. The system mainly consists of three layers: the user interface layer, the application layer, and the deployment layer.

In the user interface layer, users access the application through a web browser where they can log in, add income and expenses, and view financial reports. The application layer is developed using the Django framework, which handles the business logic and manages the financial data stored in the database.

The application is then containerized using Docker, which packages the application and its dependencies into a container. Finally, the Dockerized application is deployed on AWS Lightsail, allowing users to access the Finance Checker system online through the cloud.

3.2 Workflow / System Flow

The workflow of the Finance Checker application describes the step-by-step process from application development to deployment and user interaction. Initially, the application is developed using the Django framework, where features such as user authentication, income tracking, and expense management are implemented.

After development, the application is containerized using Docker by creating a Dockerfile and building a Docker image. This Docker image contains the complete application along with its required dependencies.

Next, the Docker container is deployed on AWS Lightsail, which provides a cloud environment for hosting the application. Once deployed, users can access the Finance Checker system through a web browser to manage and monitor their financial data efficiently.

This workflow ensures that the system runs smoothly, is easy to deploy, and can be accessed from anywhere through the internet.

4. RESULTS AND EXPLANATION

The implementation of the proposed system demonstrates the successful deployment of the Finance Checker application using Docker and AWS Lightsail. The Django-based application was containerized using Docker, which ensured that all dependencies and configurations were packaged together, allowing the application to run consistently in different environments.

After deployment on AWS Lightsail, the application was able to run smoothly and users could access it through a web browser. The system allows users to add income details, record expenses, and monitor their financial data efficiently.

The results show that using Docker for containerization and AWS Lightsail for cloud deployment improves the reliability, portability, and scalability of the application. The Finance Checker system helps users manage their finances easily and provides a simple platform for tracking financial activities online.

5. CONCLUSION AND FUTURE SCOPE

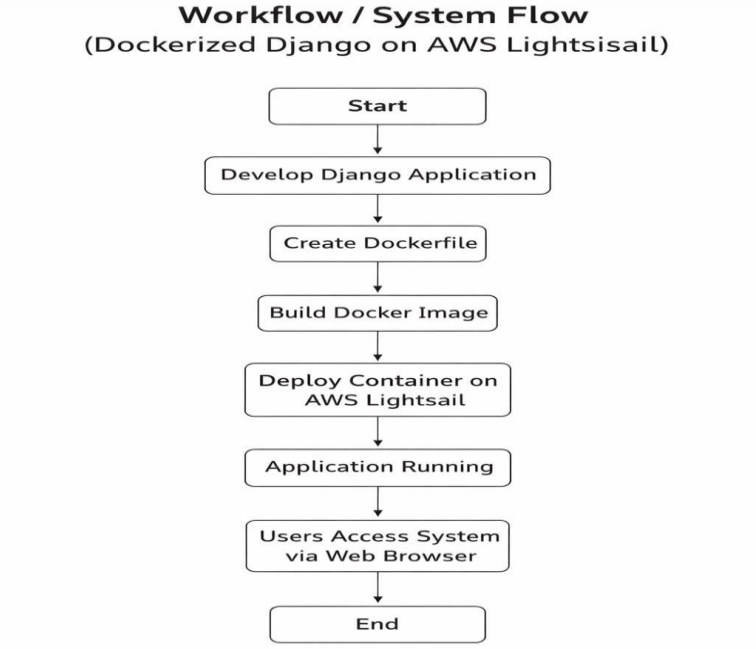
The Finance Checker application successfully demonstrates how a Django web application can be containerized using Docker and deployed on AWS Lightsail. The system allows users to easily manage their income and expenses through a simple web interface. By using Docker, the application becomes portable and easy to deploy, while AWS Lightsail provides a reliable cloud platform for hosting the system. The project proves that containerization and cloud deployment can improve application performance, scalability, and accessibility.

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.2026.15330



In the future, the application can be enhanced by adding advanced features such as data visualization with charts, automatic expense categorization, and mobile application support. Additional security features and database improvements can also be implemented to handle larger amounts of financial data and provide a better user experience.

REFERENCES

[1]. Adrian Holovaty and Jacob Kaplan-Moss, The Django Book, Apress Publications, New York, 2009. This book provides a comprehensive introduction to the Django web framework and explains how to build web applications using Python and Django.
[2]. William S. Vincent, Django for Beginners: Build Websites with Python and Django, Lean Publishing, 2022. This book explains the basic concepts of Django development and demonstrates how to create secure and scalable web applications.
[3]. Karl Matthias and Sean Kane, Docker: Up and Running – Shipping Reliable Containers in Production, O'Reilly Media, 2015. This book explains the concepts of containerization and how Docker can be used to package and deploy applications efficiently.
[4]. Nigel Poulton, Docker Deep Dive, Lean Publishing, 2023. This reference provides a detailed explanation of Docker architecture, container management, and deployment strategies.
[5]. Amazon Web Services (AWS), AWS Lightsail Documentation. Available at:
[6]. https://aws.amazon.com/lightsail/
[7]. Martin Fowler, Microservices Architecture, O'Reilly Media, 2016. This reference discusses modern application deployment methods and explains how microservices and container technologies improve scalability.
[8]. Kelsey Hightower, Brendan Burns, and Joe Beda, Kubernetes: Up and Running, O'Reilly Media, 2017. This book explains container orchestration and modern cloud deployment practices related to containerized applications.
[9]. Official Django Documentation, Django Software Foundation. Available at:
[10]. https://docs.djangoproject.com/
[11]. Docker Inc., Docker Official Documentation. Available at:
[12]. https://docs.docker.com/
[13]. Amazon Web Services, Cloud Computing Concepts and Services, AWS Whitepapers. Available at:
[14]. https://aws.amazon.com/whitepapers/

