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.2026.15355

Deploy a Dockerized E-commerce Platform on AWS ECH

Dinesh Kumar[1], Dr. C. Daneil 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: The increasing demand for scalable and reliable web applications has encouraged the adoption of cloud computing and containerization technologies. Traditional deployment methods often involve complex server management and dependency issues that affect system performance and scalability. This research presents the design and deployment of a Dockerized e-commerce platform using Amazon Web Services (AWS) Elastic Container Service (ECS). The proposed system demonstrates how containerization can simplify application deployment and improve consistency across different environments. The e-commerce application is developed using web technologies such as HTML, CSS, JavaScript, and Node.js, and is packaged into a Docker container. The container image is stored in Amazon Elastic Container Registry (ECR) and deployed using AWS ECS with the Fargate launch type.

The implementation highlights the benefits of cloud-based container orchestration, including scalability, reduced infrastructure management, and improved application availability. The results show that the proposed system provides efficient deployment, simplified maintenance, and reliable access through public internet connectivity. This study emphasizes the importance of modern DevOps practices in building scalable web applications.

I. INTRODUCTION

The rapid growth of digital technologies has transformed the way businesses operate, particularly in the field of electronic commerce. E-commerce platforms enable organizations to offer products and services online, allowing customers to browse, select, and purchase items through web applications. As user demand increases, deploying such applications in a scalable and reliable manner becomes a critical challenge.

Traditional deployment approaches rely on physical servers or virtual machines, which require manual configuration, software installation, and continuous maintenance. These processes increase operational complexity and may lead to inconsistencies between development and production environments. As a result, organizations are increasingly adopting containerization and cloud computing technologies to streamline application deployment.

Containerization, implemented through platforms such as Docker, allows applications and their dependencies to be packaged into lightweight containers. This ensures consistent performance across different environments and simplifies deployment. Cloud platforms such as Amazon Web Services provide scalable infrastructure and managed services that eliminate the need for manual server management.

This research focuses on deploying a Dockerized e-commerce application on AWS ECS. The study demonstrates how modern deployment practices improve system scalability, reduce infrastructure overhead, and enhance application reliability.

II. OBJECTIVES AND CHALLENGES

Objectives

- To develop a web-based e-commerce application.
- To containerize the application using Docker.
- To deploy the containerized application on AWS ECS.
- To provide public internet accessibility through cloud infrastructure.
- To demonstrate scalable and reliable cloud deployment practices.

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.2026.15355

Challenges

- Managing application dependencies across environments.
- Configuring container networking and security settings.
- Ensuring reliable cloud deployment with minimal downtime.
- Handling scalability requirements for increased user traffic.
- Monitoring container performance and resource utilization.

III. SYSTEM ARCHITECTURE

The proposed system architecture consists of several components including the user interface, backend server, containerization layer, and cloud deployment infrastructure.

Users access the application through a web browser over the internet. The application is hosted inside a Docker container that runs on AWS ECS. The Docker image is stored in Amazon ECR, which acts as a secure container registry. ECS manages container execution using the Fargate launch type, allowing the application to run without managing servers.
The architecture ensures that incoming user requests are processed by the containerized Node.js application and appropriate responses are delivered through web pages. Networking and security configurations enable public access through a designated IP address and port mapping.

IV. IMPLEMENTATION

The implementation process begins with the development of the e-commerce application using Node.js and frontend technologies. The application includes modules such as login, product browsing, cart management, and checkout functionality.

After development, a Dockerfile is created to define the container environment. The Docker image is built locally and tested using Docker commands. Once verified, the image is pushed to Amazon ECR.

An ECS cluster is then created, and a task definition is configured to specify container settings such as CPU allocation, memory usage, and port mapping. An ECS service is created to run and manage the container continuously. Security group rules are configured to allow inbound traffic, enabling users to access the application through the public IP address.

V. RESULTS AND DISCUSSION

The deployment results demonstrate that containerization simplifies the application deployment process and ensures consistent execution across environments. The use of AWS ECS provides automatic scaling and improved system reliability.

The application was successfully accessed through the internet using the assigned public IP address. Performance testing indicated that the containerized system handled user requests efficiently while maintaining stable operation. Cloud monitoring tools provided insights into resource utilization and system performance.
The study confirms that integrating Docker with AWS ECS offers a practical solution for deploying scalable web applications.

VI. CONCLUSION

This research presents a cloud-based deployment approach for an e-commerce platform using Docker and AWS ECS. The proposed system eliminates infrastructure management complexity while ensuring scalable and reliable application hosting.

The implementation highlights the effectiveness of container orchestration and serverless infrastructure in modern web development. The study demonstrates that cloud-based deployment practices improve application availability and operational efficiency.

VII. FUTURE ENHANCEMENT

Future improvements may include integrating payment gateways, implementing database management systems, enabling load balancing, and deploying custom domain names with HTTPS security. The system can also incorporate CI/CD pipelines for automated deployment and monitoring tools for advanced performance analysis.

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.2026.15355

REFERENCES

[1]. Amazon ECS Documentation – https://docs.aws.amazon.com/ecs
[2]. Docker Documentation – https://docs.docker.com
[3]. Node.js Documentation – https://nodejs.org
[4]. AWS Cloud Architecture Guide

