



Deploying a WordPress Site Using Docker and AWS RDS

S. Pranesh¹, Dr. S. Thavamani²

III BCA, Department of Computer Applications, Sri Ramakrishna College of Arts & Science (Autonomous),
Coimbatore – 641006, Tamil Nadu, India¹

Associate Professor, Department of Computer Applications, Sri Ramakrishna College of Arts & Science (Autonomous),
Coimbatore – 641006, Tamil Nadu, India²

Abstract: In modern cloud computing environments, deploying scalable and reliable web applications is essential for ensuring performance, availability, and efficient resource management. Traditional web hosting methods often involve manual configuration and dependency management, which can lead to inconsistencies across deployment environments. Containerization technologies and cloud services provide a solution to these challenges by enabling automated deployment, portability, and scalability.

This paper presents the deployment of a WordPress website using Docker containerization on an Amazon EC2 instance integrated with Amazon Relational Database Service (RDS) for database management. WordPress is a widely used Content Management System (CMS) that allows users to create and manage dynamic websites. By deploying WordPress inside a Docker container, the application environment becomes isolated, portable, and easy to maintain. Amazon EC2 provides scalable cloud infrastructure to host the application, while Amazon RDS offers a managed MySQL database service for secure and reliable data storage.

The proposed architecture follows a two-tier model consisting of an application layer and a database layer. The application layer hosts the Docker container running WordPress on an EC2 instance, while the database layer uses Amazon RDS to store website content and user data. Security groups and network configurations ensure secure communication between the two components. Elastic IP is used to provide consistent external access to the hosted website.

I. INTRODUCTION

Cloud computing has transformed the way applications are deployed and managed. Instead of relying on traditional physical servers, organizations now use cloud platforms that offer scalable infrastructure, on-demand resources, and flexible deployment models. One of the key trends in modern cloud computing is the use of containerization technologies such as Docker to package applications and their dependencies into isolated environments.

WordPress is one of the most popular content management systems used for building websites and blogs. It provides a user-friendly interface and supports plugins, themes, and dynamic content management. However, deploying WordPress manually on servers can be complex due to dependency management, configuration issues, and compatibility problems between environments.

Docker simplifies this process by packaging the application and its dependencies into containers. Containers ensure that the application runs consistently regardless of the underlying infrastructure. This makes Docker an ideal tool for deploying web applications in cloud environments.

Amazon Web Services (AWS) is a widely used cloud platform that provides various services for hosting and managing applications. Amazon EC2 offers virtual server instances where applications can run, while Amazon RDS provides managed database services that reduce administrative overhead. By combining Docker with AWS services, it becomes possible to deploy scalable and reliable web applications efficiently.

This project demonstrates how a WordPress website can be deployed using Docker on an EC2 instance while using Amazon RDS as the backend database. The architecture separates the application and database layers, improving scalability and reliability. The system allows users to access the WordPress website through a public IP address while ensuring secure communication between the application and database components.



II. OBJECTIVES AND CHALLENGES

2.1 Objectives

The main objectives of this project include:

- To deploy a WordPress website using Docker containerization.
- To host the containerized WordPress application on an AWS EC2 instance.
- To configure Amazon RDS as the backend MySQL database for WordPress.
- To establish secure communication between EC2 and RDS using AWS security groups.
- To demonstrate the benefits of containerization in cloud-based application deployment.
- To create a scalable and maintainable architecture for hosting dynamic web applications.

2.2 Challenges

During the implementation of the system, several challenges were encountered:

Container Configuration:

Proper configuration of Docker containers was required to ensure that WordPress could communicate with the remote RDS database.

Database Connectivity:

Setting up secure connectivity between the EC2 instance and the RDS database required proper configuration of security groups and network rules.

Dependency Management:

Ensuring compatibility between WordPress, PHP, MySQL, and Docker images required careful selection of container images.

Security Considerations:

Proper access control mechanisms were necessary to protect the database and prevent unauthorized access.

Cloud Infrastructure Setup:

Configuring AWS resources such as EC2 instances, RDS databases, and Elastic IP addresses required careful planning to ensure system reliability.

III. SYSTEM ARCHITECTURE

The system architecture follows a two-tier cloud architecture model consisting of an application layer and a database layer.

Application Layer

The application layer hosts the WordPress website running inside a Docker container. The container is deployed on an AWS EC2 instance that acts as the web server. Docker ensures that the WordPress application runs in an isolated environment with all required dependencies.

Database Layer

The database layer consists of an Amazon RDS MySQL instance that stores WordPress data such as posts, pages, user information, and configuration settings. RDS is a managed database service that automatically handles backups, updates, and scaling.

Communication

Communication between the EC2 instance and the RDS database occurs through private network connections configured using AWS security groups. Only the EC2 instance is allowed to access the database, ensuring data security.

External Access

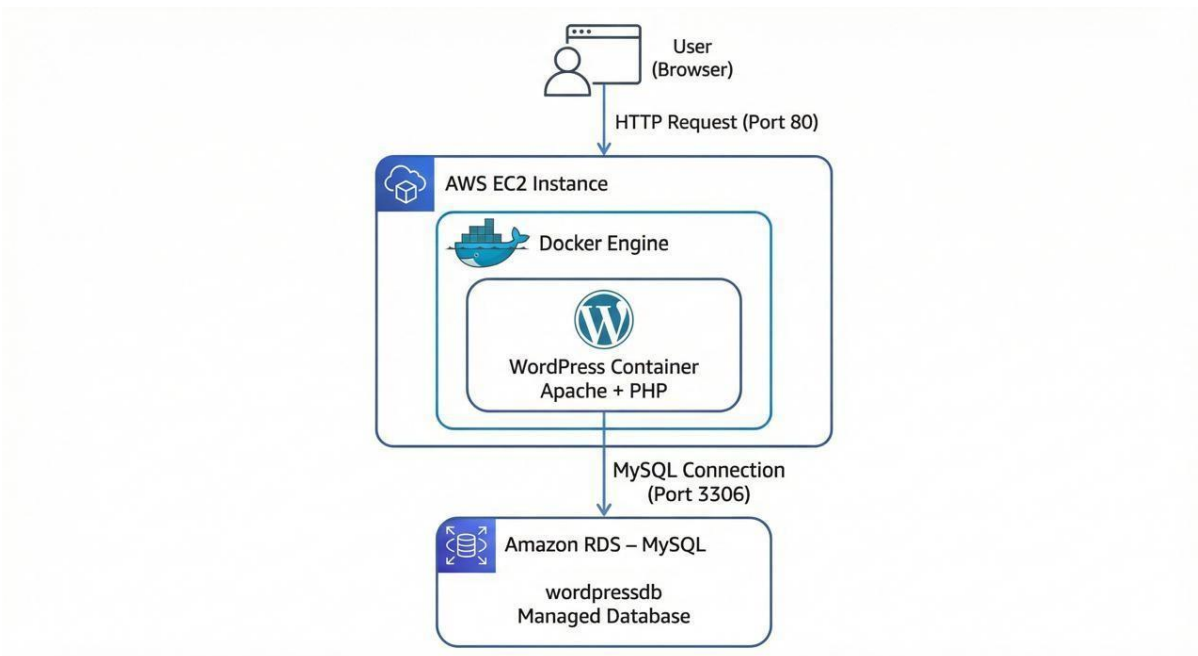
Users access the WordPress website through a web browser using the EC2 public IP address or Elastic IP.

Key Components

- User Browser
- AWS EC2 Instance



- Docker Container
- WordPress Application
- Amazon RDS MySQL Database



IV. IMPLEMENTATION

The implementation process involved several steps.

First, an EC2 instance was launched in the AWS cloud environment. The instance served as the host machine for running Docker containers. The operating system used for the instance was Amazon Linux or Ubuntu.

Next, Docker was installed on the EC2 instance. Docker allows applications to run inside containers, ensuring consistent environments across deployments.

A WordPress Docker image was then pulled from the Docker Hub repository. This image contains all the required components needed to run the WordPress application.

After configuring the container, the WordPress application was connected to an Amazon RDS MySQL database instance. The database was created using AWS RDS service with appropriate security settings.

Security groups were configured to allow communication between the EC2 instance and the RDS database while blocking unauthorized access.

Finally, the Docker container was started, and the WordPress website became accessible through the EC2 public IP address.

V. RESULTS AND DISCUSSION

The implementation successfully deployed a WordPress website using Docker on an AWS EC2 instance with Amazon RDS as the database backend.

The deployed website was accessible through a web browser using the EC2 public IP address. The WordPress application was able to retrieve and store data in the RDS database without issues.



The use of Docker simplified the deployment process by eliminating dependency conflicts and ensuring consistent environments. The containerized application could easily be redeployed or migrated to other cloud environments if required.

The separation of application and database layers improved system scalability and maintainability. Amazon RDS provided reliable database management with automatic backups and monitoring capabilities.

Overall, the system demonstrated how cloud services and containerization technologies can be combined to deploy scalable and reliable web applications.

VI. CONCLUSION

This project successfully demonstrated the deployment of a WordPress website using Docker containerization on an AWS EC2 instance with Amazon RDS as the backend database. The implementation highlights the advantages of using containerization and cloud infrastructure for hosting dynamic web applications.

Docker provided an efficient way to package the application and its dependencies into a portable container environment. AWS EC2 offered scalable infrastructure for running the application, while Amazon RDS simplified database management through its fully managed service.

The architecture ensured secure communication between the application and database layers using AWS security groups. The deployed website was accessible through a public IP address and functioned as expected.

The study confirms that combining Docker with AWS services provides a flexible, scalable, and reliable approach for deploying modern web applications in cloud environments.

VII. FUTURE ENHANCEMENTS

Several improvements can be implemented in the future to enhance the system:

- Implement load balancing using AWS Elastic Load Balancer.
- Deploy multiple EC2 instances for high availability.
- Use AWS Auto Scaling to automatically adjust resources based on traffic.
- Integrate AWS CloudFront for faster content delivery.
- Implement continuous deployment using CI/CD pipelines.
- Add SSL certificates for secure HTTPS communication.
- Monitor system performance using AWS CloudWatch.

REFERENCES

- [1]. Docker Documentation – <https://docs.docker.com>
- [2]. AWS EC2 Documentation – <https://docs.aws.amazon.com/ec2>
- [3]. AWS RDS Documentation – <https://docs.aws.amazon.com/rds>
- [4]. WordPress Official Documentation – <https://wordpress.org/support>
- [5]. Amazon Web Services – <https://aws.amazon.com>
- [6]. Merkel, Dirk. “Docker: Lightweight Linux Containers for Consistent Development and Deployment.” Linux Journal.
- [7]. Pahl, Claus. “Containerization and the PaaS Cloud.” IEEE Cloud Computing.