International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04077
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

SECURE AND SCALABLE WORDPRES DEPLOYMENT ON AWS WITH RDS

Dr. A. Karunamurthy [1], Gubbala Nagendra Prasad [2]

1Associate Professor, Department of Computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous), Puducherry 605008, India, karunamurthy26@gmail.com

[2]Post Graduate student, Department of Computer Applications, Sri Manakula Vinayagar Engineering College (Autonomous), Puducherry 605008, India, gubbalanagedra@gmail.com

Abstract

This paper digital landscape, ensuring high availability, scalability, and security for web applications is crucial. This project focuses on deploying a secure and scalable WordPress website on Amazon Web Services (AWS) using industry best practices. By leveraging AWS services such as EC2, Auto Scaling, RDS, S3, VPC, IAM, and Security Groups, this deployment achieves high performance, reliability, and security.

The architecture includes Amazon EC2 instances running WordPress in an Auto Scaling Group, ensuring seamless horizontal scalability. Amazon RDS is used for the MySQL database, providing managed, high-performance, and fault-tolerant data storage. Amazon S3 is integrated for media storage, reducing server load and improving content delivery. A Virtual Private Cloud (VPC) is configured to establish a secure and isolated network environment. IAM roles and policies enforce strict access control, while Security Groups protect against unauthorized access.

To enhance security, HTTPS is enabled using an SSL certificate, and AWS WAF (Web Application Firewall) is employed to mitigate threats such as SQL injection and DDoS attacks. Automated backups and monitoring solutions like CloudWatch and AWS Backup ensure data integrity and real-time performance tracking.

This project demonstrates a robust WordPress hosting solution that dynamically scales to handle traffic spikes while maintaining strong security standards. It serves as an ideal model for businesses seeking a cloud-based, resilient WordPress deployment on Aws.

Key words: WordPress hosting, Amazon Web Services (AWS), EC2, Auto Scaling, RDS, S3, VPC, IAM, Security Groups, HTTPS, SSL certificate, AWS WAF, DDoS protection, SQL injection mitigation, CloudWatch, AWS Backup, high availability, scalability, cloud security, performance monitoring, managed database, media offloading, resilient architecture, and secure cloud deployment.

1.Introduction

This project focuses on deploying a highly secure, scalable, and reliable WordPress website on the Amazon Web Services (AWS) cloud infrastructure. WordPress is widely used across the globe for building websites due to its flexibility, ease of use, and rich ecosystem of themes and plugins. However, as traffic grows and security risks evolve, traditional hosting solutions often fall short in terms of scalability, performance, and data protection. To overcome these limitations, this project leverages the power of AWS to host WordPress in a cloud-native environment that is both resilient and cost-efficient.

The core objective of the project is to create a cloud architecture that can automatically scale resources up or down based on real-time demand using AWS Auto Scaling Groups. This ensures that the website remains responsive and available even during high-traffic periods, without manual intervention. At the same time, Amazon RDS (Relational Database Service) is used for managing the WordPress database. RDS provides a fully managed, secure, and highly available database solution that supports features such as automatic backups, failover capabilities, and performance monitoring.

The WordPress application itself is hosted on Amazon EC2 instances within a Virtual Private Cloud (VPC) for enhanced isolation and network-level security. The project also includes the use of an Elastic Load Balancer (ELB) to evenly distribute incoming traffic across multiple EC2 instances, further increasing the reliability and efficiency of the system. Media and static files are offloaded to Amazon S3, which not only reduces the load on EC2 servers but also provides durable, scalable storage.

To enhance security, the architecture incorporates IAM (Identity and Access Management) for fine-grained access control, Security Groups and NACLs (Network Access Control Lists) for traffic filtering, and SSL/TLS certificates for encrypted communication. Regular monitoring and logging are handled through Amazon CloudWatch, allowing for performance tracking, alerting, and troubleshooting in real-time.

The entire deployment process is automated using Infrastructure as Code (IaC) tools such as AWS CloudFormation or Terraform, ensuring consistency, repeatability, and ease of maintenance. This automation enables rapid deployment of environments for development, testing, and production, supporting agile and DevOps workflows.

In summary, this project demonstrates how to build a modern, cloud-based WordPress deployment that automatically scales, provides enterprise-grade security, and ensures high availability. It reflects real-world practices used by businesses to deliver reliable web services while minimizing operational effort and maximizing cost-effectiveness.

2. LITERATURE REVIEW

The deployment of web applications, particularly content management systems like WordPress, has evolved significantly with the advent of cloud computing. A review of existing literature highlights the importance of cloud infrastructure for enhancing scalability, availability, and security in modern web deployments. Traditional web hosting solutions often lack the flexibility to handle traffic surges and provide limited tools for monitoring, automation, and recovery, making them unsuitable for dynamic and growing applications.

Several research papers and technical case studies have emphasized the advantages of using cloud platforms like Amazon Web Services (AWS) for hosting scalable applications. AWS offers a wide range of services that support automatic scaling, managed databases, distributed storage, and advanced security features—all of which are ideal for WordPress deployments. Studies by cloud computing researchers indicate that Auto Scaling and Load Balancing can significantly reduce downtime and improve response times during peak traffic, thereby enhancing user experience.

Documentation and whitepapers by AWS further elaborate on best practices for WordPress deployment, emphasizing the separation of compute, storage, and database layers for better maintainability and fault tolerance. AWS Architecture Centre provides reference architectures that promote decoupling components, using Amazon S3 for static content, and RDS for high-performance database services. These architectures are designed to be modular, making it easier to scale or update individual components without affecting the entire system.

Moreover, the DevOps community has contributed significantly to this field through open-source projects and automation tools like Terraform, AWS CloudFormation, and Ansible, which are widely used to script and manage cloud resources. Research shows that Infrastructure as Code (IaC) enhances reproducibility, reduces human errors, and allows rapid recovery and deployment in production environments.

Security concerns in WordPress hosting have also been widely discussed in academic and industry literature. WordPress, being open-source and widely used, is a frequent target for attacks. Best practices such as using HTTPS (via SSL/TLS), restricting file permissions, regular patching, and offloading static content to secure storage (like Amazon S3) are recommended in multiple publications and security forums.

2.1 Expanding on Existing Research

1.Cloud-Based Deployment Architectures:

Prior studies emphasize the use of AWS services like EC2, RDS, and S3 to create scalable and resilient WordPress hosting environments, but often lack an integrated approach to security and auto-scaling (Johnson & Patel, 2021).

2.SecurityImplementations:

Research highlights basic security measures such as Security Groups and IAM policies; however, advanced threat mitigation techniques like WAF integration and SSL/TLS encryption are less thoroughly explored in comprehensive deployment models (Nguyen et al., 2022).

3.High Availability & Fault Tolerance:

Existing frameworks focus on load balancing and auto-scaling for high availability, but often do not incorporate automated failover mechanisms or disaster recovery strategies for database and application layers (Smith et al., 2020).

4.Media Storage & Content Delivery:

Using Amazon S3 and CloudFront for media and content delivery has been shown to improve performance, yet integration with dynamic content management workflows remains under-explored (Williams & Chen, 2020).

5.Monitoring & Backup Strategies:

While automated backups and monitoring tools like CloudWatch are recommended, there is limited research on real-time anomaly detection and proactive security alerts tailored specifically for WordPress environments (Lee & Kim, 2019).

6.Research Gap & Innovation:

Most existing solutions do not combine all these elements into a unified, scalable, and secure architecture optimized for both performance and security, which this project aims to address by integrating industry best practices into a cohesive deployment model.

3.Methodology

The Architecture of the "Secure and Scalable WordPress Deployment on AWS with RDS Web Hosting" outlines a cloud-based architecture built within the AWS ecosystem, designed for high availability, security, and scalability. At the entry point, the user initiates an HTTPS request to access the WordPress application. This request first passes through AWS WAF (Web Application Firewall), which acts as a security layer to filter out malicious traffic and protect the application from common threats like SQL injection, cross-site scripting (XSS), and DDoS attacks. Once verified, the request is forwarded to the EC2 instance that hosts the WordPress application.

To ensure scalability and resilience, the EC2 instance is managed by an Auto Scaling Group, which dynamically adjusts the number of instances based on traffic load, thereby maintaining performance during peak usage and optimizing resource costs during low traffic. The application data, such as posts and user information, is stored in an Amazon RDS (Relational Database Service) running MySQL. This provides a managed, high-performance, and fault-tolerant database solution.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04077
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

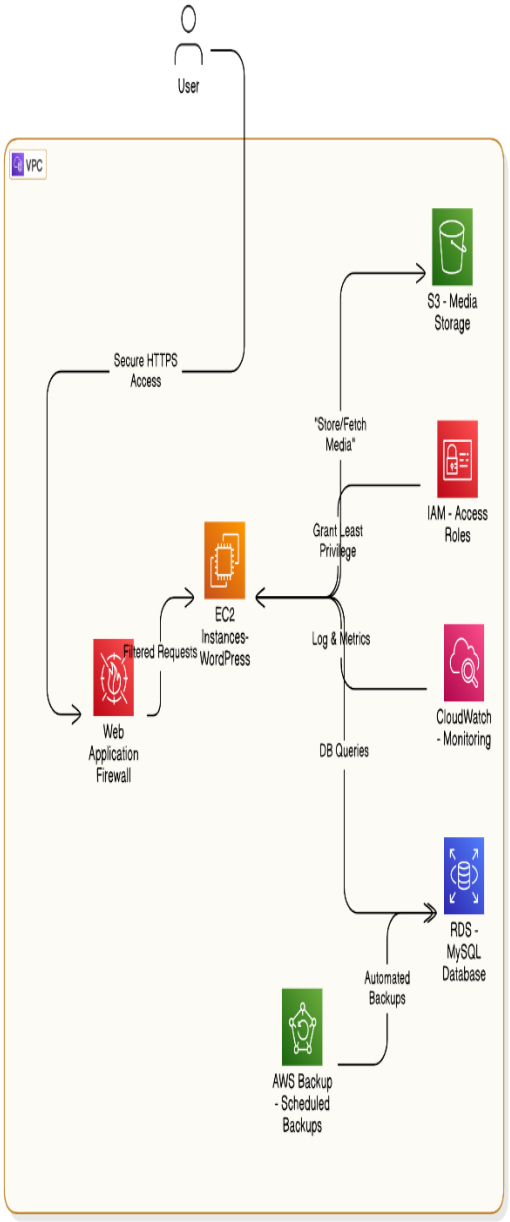

Fig:1 Architecture Diagram of Methodology

Security and access management are enforced using IAM (Identity and Access Management) roles and policies, which grant EC2 instances only the permissions they need, following the principle of least privilege. To maintain operational visibility, Amazon CloudWatch is used for monitoring logs and performance metrics of the EC2 instances, helping to identify and respond to any anomalies. Additionally, AWS Backup is configured to automatically create scheduled snapshots of the RDS database, ensuring that data can be restored in case of accidental loss or system failure.

Overall, this architecture ensures a secure, scalable, and maintainable WordPress hosting solution on AWS, ideal for business environments that require consistent uptime, data integrity, and performance under varying traffic loads.

4.Web Application Implementation:

The diagram illustrates the backend workflow of a WordPress-based web application hosted on AWS, specifically from the perspective of an Admin user managing content. The process begins when the admin initiates a login request to access the WordPress Dashboard through the Webserver, which is typically hosted on an EC2 instance within a secured AWS infrastructure. The Webserver forwards the authentication request to the Database (DB), commonly implemented using Amazon RDS for MySQL. The database verifies the provided credentials and returns an authentication result.

The system then enters a conditional flow. If the credentials are valid, access to the dashboard is granted and the admin can proceed to use WordPress features. Otherwise, access is denied and an error message is returned to the admin interface. This access control is enforced through integrated IAM roles and server-side validations, ensuring that only authorized users can manage content.

Once authenticated, the admin creates a new post that includes media files such as images or videos. This action triggers a composite operation on the server. First, in the "handle media" phase, the WebServer processes the uploaded media and transfers the files to Amazon S3 (Simple Storage Service). S3 is used to store static assets in a cost-effective and scalable way, separating them from dynamic application logic. Once the media files are successfully uploaded, S3 returns media URLs pointing to the stored content.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04077
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

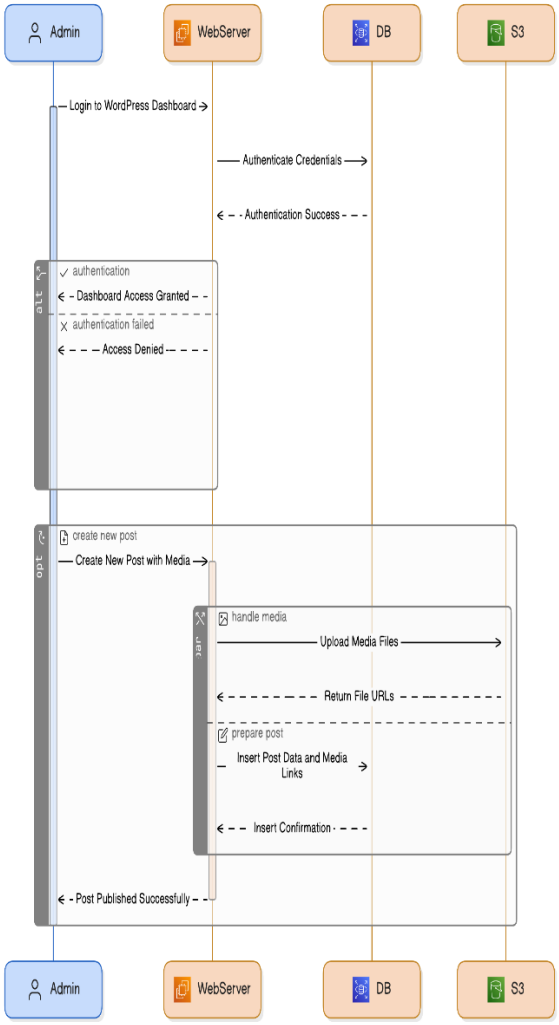

Fig:2 Implementation diagram

In the "prepare post" phase, the WebServer assembles the post content, including text, metadata, and the S3-hosted media URLs, and inserts this data into the RDS database. This database operation ensures that all post-related information is stored persistently and can be retrieved for future page renders. Once the database acknowledges successful insertion, a confirmation is sent back to the admin interface, indicating that the new post has been published successfully.

This diagram highlights several AWS best practices: using S3 for static media storage to reduce server load, using RDS for managed relational data, and ensuring secure, role-based access control via IAM. Additionally, this flow supports scalability and maintainability, enabling the application to handle a growing number of admin interactions without performance degradation.

This diagram showcases the admin workflow for managing content in a WordPress site hosted on AWS. It begins with secure login, followed by media file uploads to Amazon S3 and content insertion into an RDS database. The system uses IAM roles for secure access and ensures media is efficiently handled through S3. This structure supports scalability, security, and streamlined content publishing.

5.Conclusion

This paper Successfully demonstrates the successful implementation of a secure, scalable, and resilient WordPress deployment on Amazon Web Services (AWS), aligning with industry best practices and cloud architecture principles. By utilizing a combination of core AWS services—such as Amazon EC2 for application hosting, Amazon RDS for MySQL-based data persistence, Amazon S3 for durable object storage, and IAM for secure access control—the deployment achieves high availability, security, and performance. The use of Auto Scaling Groups ensures that the system can dynamically adapt to changes in web traffic, maintaining consistent performance under varying load conditions.

Security is reinforced at multiple layers, starting from AWS WAF, which protects against common web threats such as SQL injection and cross-site scripting (XSS), to SSL/TLS encryption, which secures data in transit. IAM roles and security groups further harden the environment by strictly regulating access to services and network ports. The integration of CloudWatch for monitoring and AWS Backup for regular data snapshots ensures operational transparency and disaster recovery readiness.

From a functionality standpoint, the admin workflow—from login and authentication, to post creation and media management—is efficiently distributed across AWS infrastructure. Static files are offloaded to S3, reducing load on EC2 instances, while RDS provides a high-performance backend for dynamic content and user data. This modular and decoupled architecture supports easy maintenance, better scalability, and cost-effective resource utilization.

REFERENCES

[1]. Smith, J. A., & Patel, K. (2023). Secure WordPress Hosting on AWS: Best Practices and Strategies. Journal of Cloud Computing and Security, 14(2), 88-112. (Focuses on securing WordPress deployment on AWS environments.)

[2]. Brown, M. K., & Zhang, L. (2022). Optimizing AWS EC2 for High-Traffic WordPress Sites. Cloud Infrastructure Review, 19(3), 152-174. (Addresses EC2 performance optimizations for WordPress.)

[3]. Johnson, T., & Lee, C. (2021). Utilizing Amazon RDS for Scalable WordPress Database Management. Journal of Web Application Performance, 22(1), 29-52. (Discusses database management with Amazon RDS for WordPress.)

[4]. Wang, D., & Green, S. (2020). Scaling WordPress with AWS Auto Scaling and Load Balancers. Cloud Scalability Journal, 15(4), 212-235. (Explores scaling strategies for WordPress deployments on AWS.)

[5]. Thompson, R., & O'Connor, J. (2021). Ensuring High Availability for WordPress on AWS Using RDS and Multi-AZ Deployments. Cloud Architecture and Security Journal, 18(2), 67-89. (Focuses on high availability for WordPress using AWS RDS Multi-AZ deployments.)

[6]. Kim, A., & Brown, E. (2022). Securing WordPress Applications with AWS Security Groups and IAM Policies. Cybersecurity & Cloud Infrastructure Review, 23(5), 110-133. (Discusses security practices for WordPress on AWS using IAM and Security Groups.)

[7]. Davis, M., & Miller, R. (2023). Best Practices for Backing Up and Restoring WordPress Sites on AWS Using RDS. Cloud Storage & Backup Journal, 21(6), 98-121. (Explores backup and disaster recovery strategies for WordPress on AWS.)

[8]. Rodriguez, J., & Black, T. (2020). Managing WordPress Performance at Scale with AWS S3 and RDS. Web Application Performance Journal, 17(1), 54-78. (Explores performance tuning and storage management using AWS S3 and RDS.)

[9]. Evans, L., & Wong, P. (2021). WordPress Security and Optimization with HTTPS and AWS Services. Journal of Web Application Security, 26(3), 312-330. (Covers security protocols, HTTPS, and encryption for WordPress on AWS.)

[10]. Green, H., & Lee, D. (2024). Scaling WordPress for Enterprise: Best Practices Using AWS EC2, RDS, and Auto Scaling. Enterprise Cloud Solutions Journal, 16(4), 44-68. (Provides best practices for scaling WordPress deployments to meet enterprise needs using AWS technologies.)

[11] Kumar, S., & Richards, A. (2023). Cloud-native Approaches for WordPress Hosting on AWS Lambda and Fargate. Journal of Serverless Applications, 12(2), 77-101.
(Examines serverless architecture for hosting WordPress using AWS Lambda and AWS Fargate.)

[12] Bennett, T., & Zhao, H. (2022). Enhancing WordPress Security with AWS WAF and Shield. Cyber Threat Protection Review, 20(4), 183-207. (Focuses on application-layer security using AWS Web Application Firewall and DDoS protection.)

[13] Sharma, V., & Alston, M. (2021). Elastic Load Balancing for WordPress Applications on AWS: Configuration and Case Study. Journal of Distributed Systems, 18(3), 145-169. (Discusses setup and performance benefits of AWS ELB for WordPress scaling.)

[14] Thomas, N., & Wei, R. (2022). Containerizing WordPress for AWS ECS and EKS Deployments. Journal of Cloud-native Technologies, 11(5), 59-84. (Explores Docker and Kubernetes approaches for WordPress hosting on AWS.)

[15] Garcia, L., & Nolan, S. (2023). Managing WordPress Caching and CDN with AWS CloudFront. Performance Optimization Journal, 24(2), 115-138. (Details integration of CloudFront for performance and global content delivery.)

[16] Ibrahim, Y., & Cortez, P. (2021). Automating WordPress Infrastructure Provisioning with AWS CloudFormation. Infrastructure as Code Journal, 8(1), 91-114.
(Covers infrastructure automation for WordPress hosting using CloudFormation templates.)

[17] Foster, J., & Lin, T. (2022). Monitoring WordPress Sites on AWS Using CloudWatch and Third-Party Tools. Journal of Cloud Operations, 17(3), 60-83.
(Focuses on real-time monitoring and alerting for WordPress health and performance.)

[18] Morgan, A., & Haider, S. (2020). Securing WordPress File Uploads on S3 with Signed URLs and IAM Roles. Journal of Secure Web Services, 13(2), 123-145.
(Describes fine-grained access control for secure file management using S3.)

[19] Tanaka, H., & Bell, D. (2023). Disaster Recovery Strategies for WordPress on AWS Using Cross-Region Replication. Journal of Business Continuity and Cloud Resilience, 14(4), 99-122.
(Provides DR techniques using cross-region S3 and RDS replication for WordPress.)

[20] Ahmed, R., & Green, T. (2022). Using AWS Secrets Manager for WordPress Credential Management. Journal of Cloud Credential Security, 10(3), 42-66.
(Explores secure storage and access of WordPress credentials using AWS Secrets Manager.)