

# CloudCanvas: A Comprehensive Guide to WordPress Deployment on AWS

By Nguyen Huu Truong Giang



Fig. 1. Deploy WordPress Website on AWS

**Abstract**—This report serves as a comprehensive guide for installing the popular content management system WordPress on Amazon Web Services (AWS). The study outlines the exact steps involved in installing WordPress on AWS infrastructure, making use of services such as Auto Scaling Group, VPC, EC2, RDS, S3, and Load Balancer. This guide covers best practices and considerations for a solid WordPress deployment on AWS, giving companies a scalable and dependable platform for hosting their websites. It covers everything from basic setup to optimization and scaling techniques.

## INTRODUCTION

WordPress is an open-source, free content management system (CMS) that makes it simple to build and maintain websites. Although blogs are the most common application for it, it can also power e-commerce sites, message boards, and a host of other well-known applications. WordPress is utilized by approximately 30% of all websites on the internet, making it the most popular CMS in the world.

The purpose of this project is to give a complete guide on establishing WordPress on Amazon Web Services (AWS), a premier cloud computing platform famous for its scalability, flexibility, and sturdy architecture. Through the utilization of AWS services like EC2, RDS, and S3, companies can establish a robust WordPress environment that can manage fluctuating workloads while preserving peak performance.

This project attempts to demystify the intricacies of AWS deployment for WordPress, motivated by the growing need for scalable web hosting options and the need to expedite the deployment process. Through the provision of an extensive road map, optimal methodologies, and pragmatic perspectives, this manual enables users to effectively utilize AWS and optimize the capabilities of their WordPress websites.

In-depth coverage of every phase of WordPress deployment on AWS is provided in this study, ranging from basic setup to continuous administration and optimization. By clarifying the deployment procedure and emphasizing best practices, I hope to provide readers with the information and resources they need to successfully start their WordPress journey powered by AWS.

## DESIGN DIAGRAM

### 1. Project Networking Service:

- Amazon VPC
- Public Subnets
- Private Subnets
- Route Tables
- + NAT Gateway
- + Internet Gateway

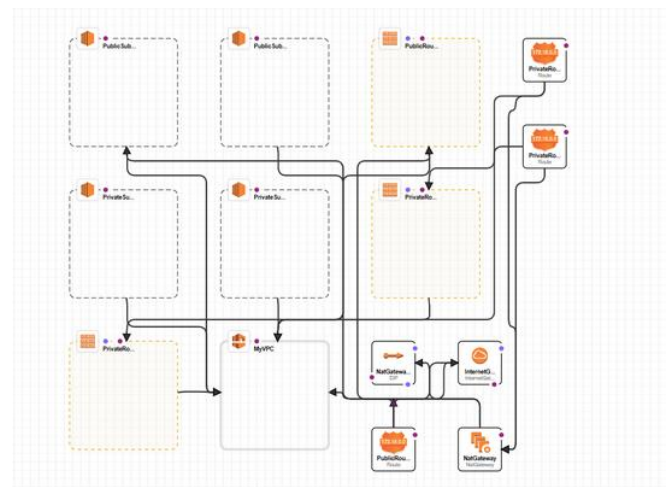


Fig 2. Project Network Related Services Structure

## 2. Application Base Infrastructure:

- Amazon S3 (Media)
- EC2 Instance
- Amazon RDS Instance (MySQL)  
+ Instance Read-Replica (Secondary)

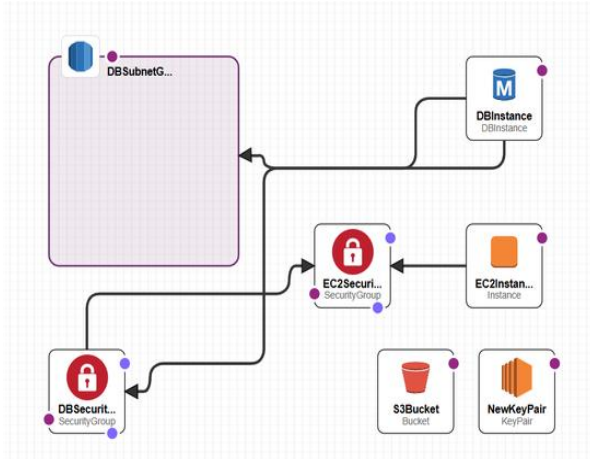


Fig 3. Project Application Infrastructure (EC2, S3, and RDS)

## 3. AWS Handle Infrastructure:

- Elastic Load Balancer  
+ Target Group
- Instance Launch Template  
+ AMIs
- Auto Scaling Group  
+ Scaling Policies

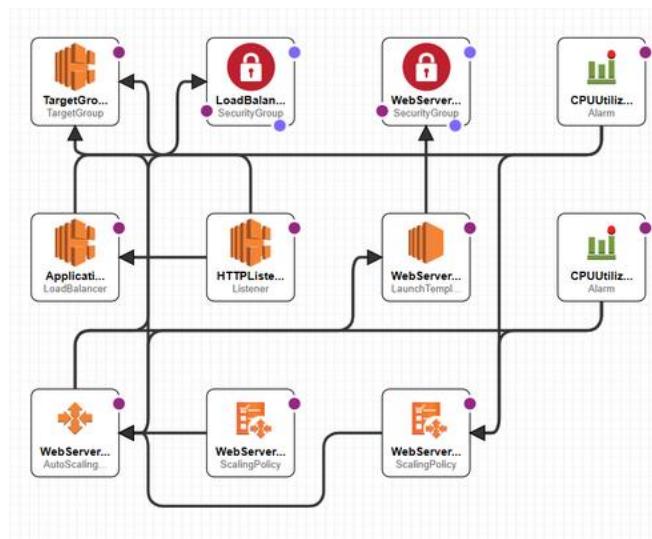


Fig 4. AWS Infrastructure and Scaling Policies

## 4. Workflow:

- The Application Load Balancer receives user data through Internet Gateway and distributes it among EC2 instances.
- To ensure peak availability and performance,

traffic is managed via the Application Load Balancer.

- Based on predetermined policies, the Auto Scaling Group scales up or down the number of EC2 instances while keeping an eye on traffic.
- Within the Amazon VPC, EC2 instances are set up to offer security and seclusion. The Amazon RDS MySQL database is connected to WordPress application instances in order to store data.
- Amazon S3 offers robust and scalable storage for media files and backups.

## 5. Scalability and Reliability:

- Scalability and availability are guaranteed by the Auto Scaling Group, which dynamically modifies the number of EC2 instances to accommodate fluctuating traffic volumes.
- By distributing traffic equally among several EC2 instances, elastic load balancing enhances fault tolerance and dependability.
- Amazon RDS enhances data durability and dependability with its scalability, multi-AZ deployments, and automatic backups.
- Scalable storage for backups and media files is offered by Amazon S3, lowering the possibility of data loss and guaranteeing high availability.

## 6. Security

- Network isolation and security group configurations are made possible by Amazon VPC, allowing for the management of incoming and outgoing traffic to EC2 instances.
- Securing safe communication between components is ensured by IAM roles and policies, which limit access to AWS services and resources.

## IMPLEMENTATION

### 1. VPC components

A customized VPC with private and public subnets and appropriate routing to NAT and IGW is used to host this application.

- Create a new VPC with a CIDR block (e.g., 10.0.0.0/16).
- Divide the VPC into public and private subnets across multiple Availability Zones (AZs).

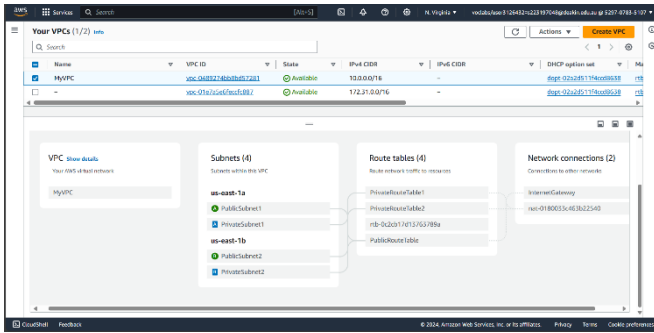


Fig 5. AWS Virtual Private Cloud (VPC)

## 2. Public Subnets

- Across AZs, distribute public subnets (such as 10.0.0.0/24 and 10.0.1.0/24).
- To direct traffic headed for the internet to the internet gateway, configure routing tables.

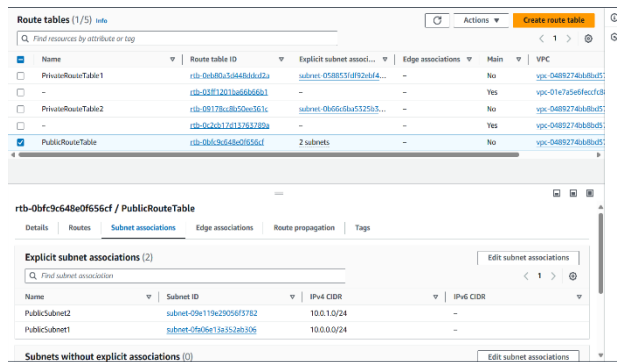


Fig 6. Public Route Tables associated with public subnets.

## 3. Private Subnets

- Distribute private subnets throughout AZs (for example, 10.0.3.0/24 and 10.0.4.0/24).
- To direct outgoing traffic via a NAT gateway, configure routing tables.

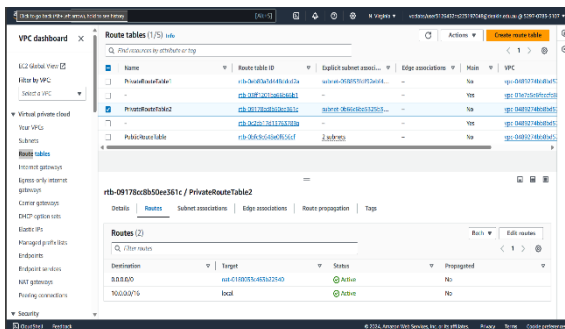


Fig 7. Private Route Tables associated with private subnets.

## 4. Amazon RDS (MySQL)

- Set up an RDS instance for MySQL database storage with MultiAZs enable.
- Create a read-replica in a different AZ than the primary RDS.

- Configure security groups to allow inbound traffic from EC2 instances.

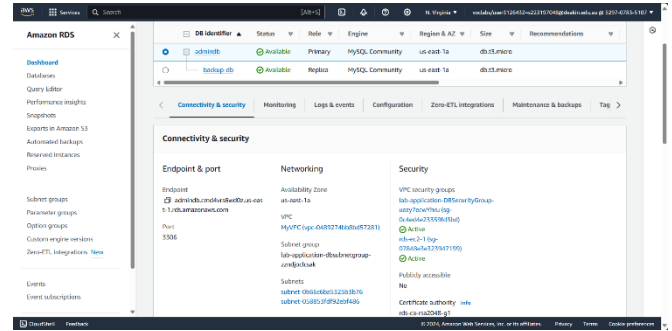


Fig 8. RDS Instance with Replica in the different AZ

## 5. Elastic Load Balancer (ELB)

- Deploy an Application Load Balancer in the public subnets to distribute incoming traffic across EC2 instances.
- Configure health checks and listeners to ensure high availability and fault tolerance.

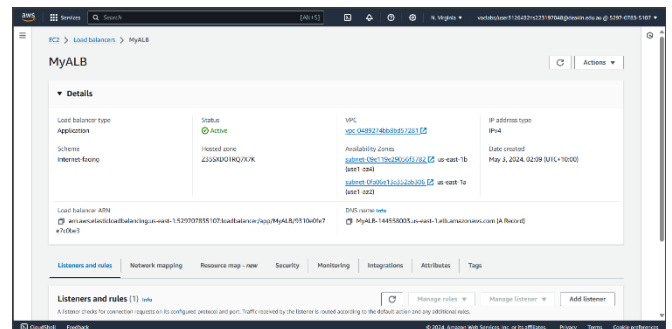


Fig 9. Application Load Balancer, Target Group and HTTP Listener

## 6. Amazon S3 (Media Storage)

- Store media files, backups, and other static assets related to the WordPress website using Amazon S3.
- In order to control object retention and storage expenses, use versioning and lifecycle controls.

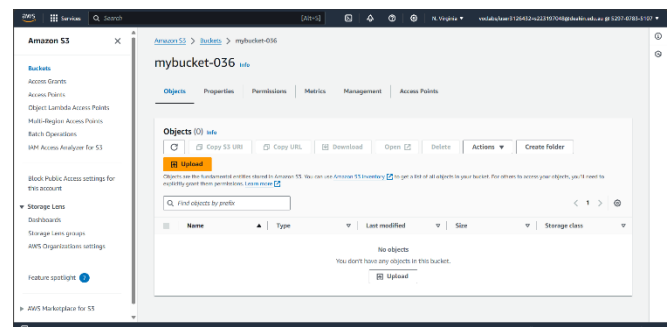


Fig 10. Amazon S3 bucket with public access disabled.

## 7. WordPress configuration via EC2 instance

- “AMI Server” instance in Public Subnet 2 with WordPress configured to connect to the RDS.
- Setup a connection between RDS and AMI Server

Connection details

Connection setup between RDS [admindb](#) and EC2 instance [i-0b5f68aa6508f578](#)

Event	Resource ID	Status
Create security group for RDS database <a href="#">admindb</a>	<a href="#">rds-ec2-1</a>	Succeeded
Create security group for EC2 instance <a href="#">i-0b5f68aa6508f578</a>	<a href="#">ec2-rds-1</a>	Succeeded
Remove default egress rule	<a href="#">rds-ec2-1</a>	Succeeded
Remove default egress rule	<a href="#">ec2-rds-1</a>	Succeeded
Create security group rules for RDS database security group	<a href="#">rds-ec2-1</a>	Succeeded
Create security group rules for EC2 instance security group	<a href="#">ec2-rds-1</a>	Succeeded
Attach security group to EC2 instance <a href="#">i-0b5f68aa6508f578</a>	<a href="#">ec2-rds-1</a>	Succeeded
Attach security group to RDS database <a href="#">admindb</a>	<a href="#">rds-ec2-1</a>	Succeeded

Close

Fig 11. Connected compute resources.

- Using the credentials to start configuring WordPress website.
- Run the installation to check if the database successfully connects.

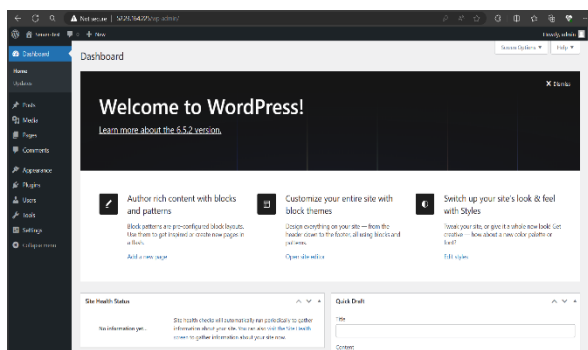


Fig 12. WordPress configuration.

- Now you have an “AMI Server” instance with WordPress configured to connect to the RDS.
- Create an AMI from this server.

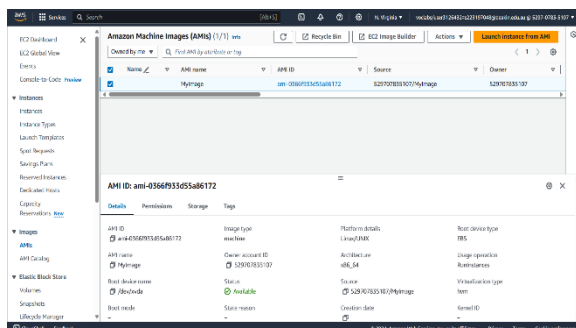


Fig 13. AMI created from “AMI Server” instance.

## 8. S3 Integration

- The WordPress site to an S3 bucket for

storing media files using WordPress plugin via **Offload Media by Acoweb**s

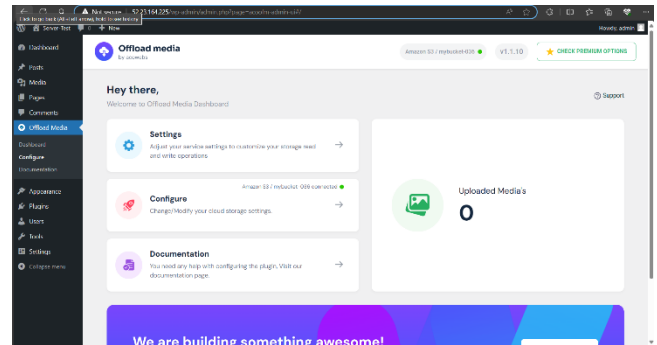


Fig 14. Offload Media by Acoweb's

- Create a Launch Template using AMI create in last step.

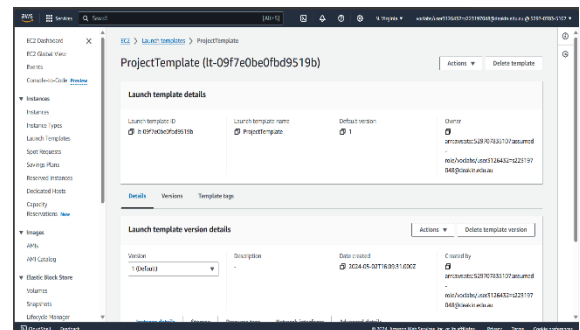


Fig 15. Launch Template via EC2

## 9. Auto Scaling Group

- To automatically scale EC2 instances depending on preset metrics (such as CPU consumption), create an ASG.
- Set up launch settings using the instance type, desired AMI, and additional parameters.

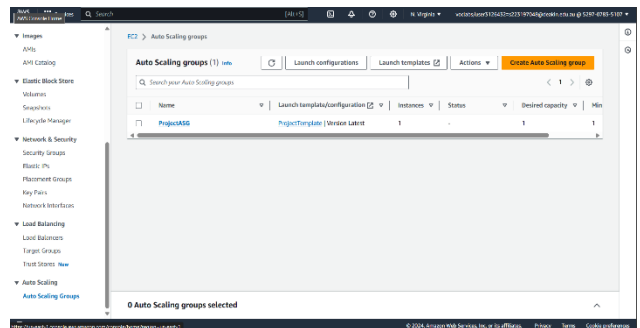


Fig 16. Auto Scaling Group via EC2

## 10. Cloudformation Infrastructure

- With the help of AWS's robust Infrastructure as Code (IaC) tool, CloudFormation, you can specify and supply AWS infrastructure resources using declarative JSON or YAML



template syntax.

- Cloudformation Infrastructure Code including successful automated deployments  
screenshot: <https://github.com/gn213245/SIT-233-Final-Project.git>

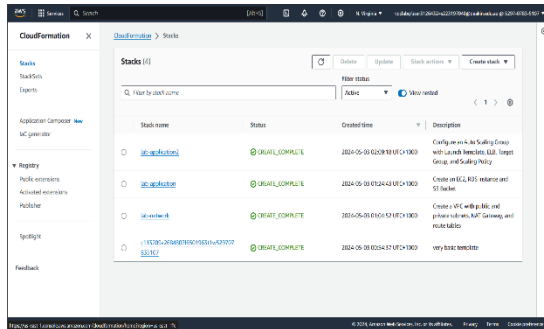


Fig 17. Application and Network Infrastructure via Cloudformation.

## DISCUSSION AND REFLECTION

### 1. Testing Outcomes:

#### a) Access WordPress page via the ALB.

- Yes, you can access a WordPress page via an Application Load Balancer (ALB) in AWS.
- The ALB will route incoming traffic to the WordPress EC2 instances via the configured target group.

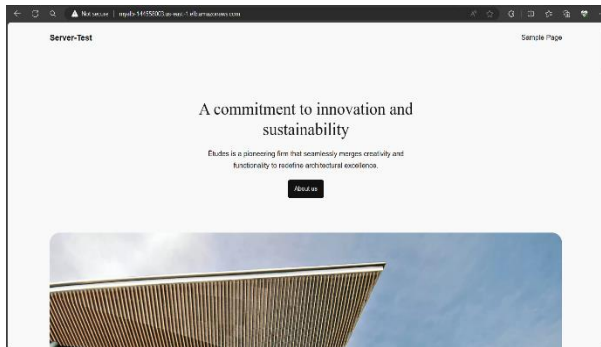


Fig 18. Access WordPress page via ALB

#### b) Terminate the Web Server and check if the ASG creates a new one.

- The Auto Scaling Group (ASG) should detect the termination and initiate the creation of a new instance to maintain the desired capacity.
- Yes, the ASG already created a new instance after terminating.

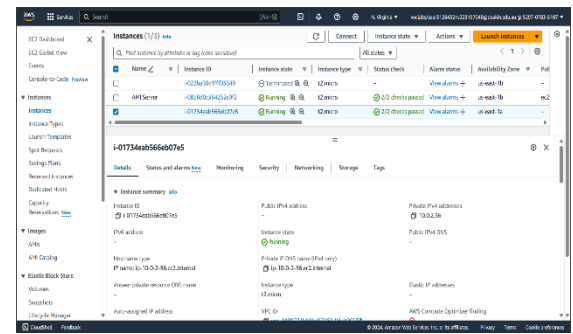


Fig 19. ASG created a new instance.

#### c) Check you can connect to the Web Server via the Session Manager

- Make that your EC2 instance has the necessary IAM role attached to it and that the AWS Systems Manager Agent (SSM Agent) has been established. Communication between the instance and AWS Systems Manager services is made possible by this agent.
- However, Learner Lab does not have IAM Role Permissions to communicate with AWS Systems Manager and create Session Manager access.
- This testing case cannot be successful

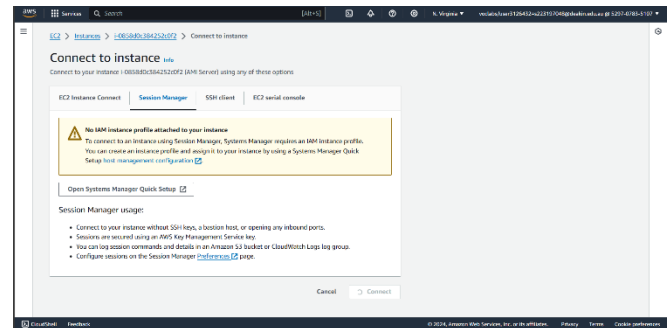


Fig 20. Access EC2 instance via Session Manager failed.

#### d) Upload a photo to the WordPress media library and make sure it has been uploaded to the S3 bucket.

- From the sidebar, go to "Media" -> "Library." Click on the "Add New" button at the top of the page. Select the photo file from your local computer and upload it. Make sure file path is pointing to S3 bucket.

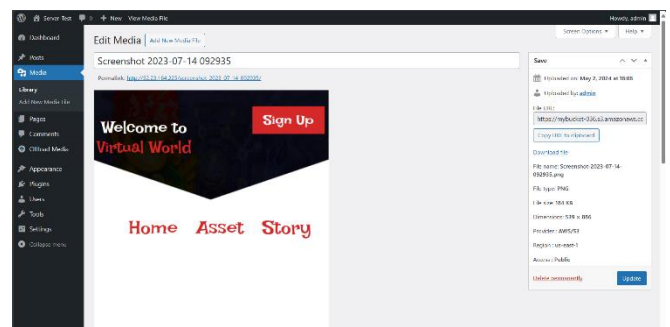
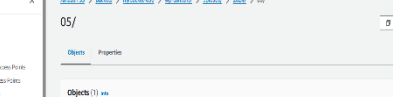


Fig 21. URL path is pointing to S3 bucket.

- 
- The screenshot shows the AWS IAM console interface. At the top, there's a navigation bar with 'Users' selected. Below it, the 'Users' page is displayed. The 'Users' list table shows a user named 'james@redhat.com' with a status of 'Enabled' and a last modified date of 'Mar 1, 2024, 04:06:12'. The 'Groups' section shows a group named 'james@redhat.com' with a status of 'Enabled' and a last modified date of 'Mar 1, 2024, 04:06:12'.

e) Stop RDS instance and observe the change of Secondary Zone

- 
- The screenshot shows the Amazon RDS console's 'Databases' page. At the top, there's a navigation bar with the Amazon RDS logo and a search bar. Below this, a left sidebar contains navigation links: 'Instances', 'Subnet groups', 'Events', and 'Event subscriptions'. The main content area is titled 'Databases (2)' and includes a 'Group resources' button, a 'Modify' button, an 'Actions' dropdown, and a 'Restore from S3' button. A search bar labeled 'Filter by database' is present. The database list table has columns for 'DB identifier', 'Status', 'Role', 'Engine', 'Region & AZ', 'Size', and 'Recommendations'. Two databases are listed: 'db-ids' with a 'Deleting' status and 'db-ids-2' with an 'Available' status. The 'db-ids' database is highlighted in blue. At the bottom of the console, there's a footer with the text '© 2024 Amazon Web Services, Inc. or its affiliates. All rights reserved.' and a 'Help' link.

## 2. Areas for Improvement:

- improve system responsiveness and user experience even further.

- ### 3. Real-World Implementation and Benefits:

- ## PRESENTATION

### 1) Architecture design visual representation:

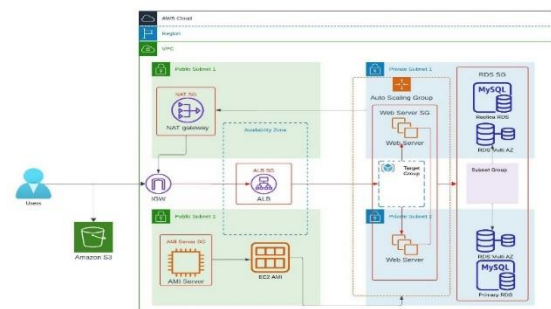


Fig 24. Architecture Design diagram (Created by Lucid Chart)

- **Amazon VPC:** Provides isolated networking environments for enhanced security and control over network traffic.
- **Amazon RDS:** Offers managed database services for high availability, automated backups, and scalability of the WordPress database.
- **Amazon EC2:** Provides resizable compute capacity for hosting the WordPress application, with Auto Scaling for scalability and cost-efficiency.
- **Amazon S3:** Offers scalable object storage for media files, backups, and static assets, ensuring durability and integration with other AWS services.
- **Application Load Balancer (ALB):** Based on predefined criteria, ALB divides up incoming traffic among many EC2 instances that host the WordPress application. It ensures high availability and scalability by evenly distributing traffic across instances and supporting content-based routing.
- **Auto Scaling Group (ASG):** ASG dynamically scales the capacity of the WordPress application to handle fluctuating traffic levels, minimizing operational overhead and costs.
- **AWS CloudFormation:** Using templates, CloudFormation streamlines AWS infrastructure deployment and administration. By specifying resources and configurations in a template format, it streamlines and expedites the provisioning of infrastructure. This lowers operational costs and manual intervention while ensuring consistency, reproducibility, and scalability.

## 2) Technical Implementation video presentation:

Link: <https://youtu.be/RvGYhCV6R24>

## CONCLUSION

This project report concludes by describing the implementation of a WordPress application on Amazon Web Services (AWS) through the use of a range of services, including Amazon VPC, Elastic Load Balancer (ELB), Auto Scaling Group (ASG), and Amazon EC2. We have developed a scalable, dependable, and secure infrastructure for hosting WordPress websites that is appropriate for use in practical commercial applications by utilizing these AWS services.

We succeeded in achieving the following goals during the project:

- **Scalability:** We were able to dynamically scale resources in response to changing traffic levels by using Auto Scaling Groups and Elastic Load Balancers, which

ensured peak performance and economical use.

- **Reliability:** Within Amazon VPC, distributing resources among several Availability Zones improved fault tolerance and resilience, reducing the effect of any single point of failure, and guaranteeing high availability.

- **Security:** Encryption, network isolation, security groups, and other best practices for security helped safeguard the infrastructure against illegal access and data breaches while guaranteeing adherence to laws and industry standards.

- **Cost Optimization:** Infrastructure expenses were minimized while optimizing return on investment by utilizing AWS's pay-as-you-go pricing model and cost management techniques like Reserved Instances and instance right-sizing.

- **Flexibility and Agility:** Rapid deployment and configuration experimentation were made possible by the ease of replication and change of the infrastructure stack made possible using CloudFormation templates.

WordPress deployment on AWS has several advantages in real-world business settings, including scalability, stability, security, cost-effectiveness, and agility. Businesses may concentrate on their primary goals by utilizing cloud computing and AWS services, as AWS takes care of the underlying infrastructure and offers a dependable and expandable platform for hosting websites and apps.

All things considered, this project shows how well AWS services can be used to install and manage WordPress applications, highlighting the potential of cloud-based solutions to fulfill the changing demands of contemporary enterprises in an increasingly digital environment.

## ACKNOWLEDGEMENT

I am grateful to the staff at school for offering a wide range of cloud computing services (AWS) and resources that help companies operate more efficiently and innovate. Many companies all across the world have benefited greatly from their dedication to quality in customer delivery and ongoing innovation.

Furthermore, I would like to express our gratitude to the WordPress open-source community for creating and maintaining such a strong and adaptable content management system (CMS). WordPress has made it easier for people and companies of all sizes to create websites and build an online presence.

I also want to express my gratitude for all of the help and advice that our mentors and coworkers have given us during the project's preparation and execution. Their suggestions and support have been helpful in guiding our strategy and producing significant results.

## REFERENCES

- [1] Amazon Web Services et al. “How-To Guide: Deploy WordPress with Amazon RDS”. Available at: [Deploy WordPress with Amazon RDS](#)
- [2] Ajit Fawade et al. “How to Deploy WordPress Website on AWS: A Step-by-Step Guide for Beginners”, Nov 22, 2023 [Online]. Available at: [How to Deploy WordPress Website on AWS: A Step-by-Step Guide for Beginners | by Ajit Fawade | Medium](#)
- [3] Prajwal Thaware. “Building a WordPress Website on AWS: A Step-by-Step Guide”, Sep 19, 2023 [Online]. Available at: [Building a WordPress Website on AWS: A Step-by-Step Guide | by Prajwal Thaware | Medium](#)
- [4] Learn Online with EduBridge. “Deployment of WordPress Website using AWS Cloud - EC2”, 2023 [Online]. Available at: [\(49\) Deployment of WordPress Website using AWS Cloud - EC2 - YouTube](#)
- [5] Unknow et al. “How to Install WordPress on AWS: Detail, Step-by-Step Guide”, Sep 30, 2023. Available at: [How to Install WordPress on AWS: Detailed, Step-by-Step Guide \(themeisle.com\)](#)
- [6] Aman Pathak. “Deploying WordPress on AWS EC2: A Beginner’s Guide”, Jul 22, 2023 [Online]. Available at: [Deploying WordPress on AWS EC2: A Beginner’s Guide | by Aman Pathak | AWS Tip](#)

## APPENDIX

- CloudFormation Infrastructure as Code accessible link (include successful screenshot each part):  
<https://github.com/gn213245/SIT-233-Final-Project.git>
- Visual representation of the architecture using Lucid Chart:  
[https://lucid.app/lucidchart/invitations/accept/inv\\_75bc43c7-44df-4a7d-8166-8ebb365e43a3](https://lucid.app/lucidchart/invitations/accept/inv_75bc43c7-44df-4a7d-8166-8ebb365e43a3)



