AWS Architecture Documentation

Overview

This document provides a detailed explanation of the AWS architecture depicted in the diagram. The architecture represents a secure, highly available WordPress hosting solution with multiple security layers, redundancy, and performance optimization components.

Architecture Components

1. External Connectivity

Component	Description
GoDaddy	Domain registrar managing DNS records that point to AWS resources
CloudFront	Content delivery network for caching and global distribution of content
Global Accelerator	Improves availability and performance by routing traffic through AWS's global network
Internet Gateway	Provides internet access to resources within the VPC

2. Security Layer

Component	Description
Client VPN	Allows internal users to securely connect to the VPC
VPN Server	Handles VPN connections within the VPC infrastructure
Certificate Manager	Manages SSL/TLS certificates for secure communications
Secrets Manager	Securely stores sensitive credentials and configuration data
IAM Role	Manages access permissions for AWS resources and services
Security Groups	Acts as virtual firewalls controlling inbound and outbound traffic for EC2 instances

3. Network Architecture

Component	Description
VPC	Virtual private cloud containing all resources in isolated network environment
Public Subnets	Host internet-facing resources like ALB and NAT Gateway
Private Subnets	Contain application and database components for enhanced security
Route Tables	Define network traffic paths between subnets and to external networks
NAT Gateway	Allows private subnet resources to access internet while remaining private

4. Application Tier

Component	Description
Application Load Balancer (ALB)	Distributes traffic across multiple EC2 instances and availability zones
EC2 Instances (t2.m)	Hosts the WordPress application in an auto-scaling configuration
Auto Scaling	Automatically adjusts capacity based on demand patterns
Network Access Control Lists (NACLs)	Provides subnet-level security controls

5. Data Storage & Caching

Component	Description
Amazon ElastiCache for Redis	In-memory caching to improve performance and reduce database load
RDS (MySQL)	Managed relational database for WordPress with replication
S3	Object storage for static assets, media files, and backups

Data Flow

User Access Path

1. **External Users**:
   - Connect via the internet through CloudFront and Global Accelerator
   - Traffic is routed to the Application Load Balancer
2. **Internal Users**:
   - Connect via the Client VPN
   - Access internal resources securely through the VPN Server
3. **Common Path**:
   - All traffic routes through the Application Load Balancer
   - ALB distributes requests to EC2 instances across availability zones

Application Processing

1. WordPress instances retrieve:
   - Security certificates from Certificate Manager
   - Sensitive data from Secrets Manager
   - Session and object cache data from ElastiCache
2. Load balancing ensures:
   - Even distribution of traffic

- Failover in case of instance failure
- Scaling based on demand

Database Interactions

1. EC2 instances connect to:
   - RDS MySQL databases located in private subnets
   - ElastiCache clusters for caching
2. Database replication:
   - Primary database in one availability zone
   - Standby replica in second availability zone for redundancy

High-Availability Features

The architecture implements multiple high-availability features:

1. **Multi-AZ Deployment**:
   - Resources spread across multiple availability zones
   - Redundancy for all critical components
2. **Auto-Scaling**:
   - EC2 instances scale based on traffic demands
   - Maintains performance during peak periods
3. **Load Balancing**:
   - Distributes traffic across healthy instances
   - Performs health checks and removes unhealthy instances
4. **Database Redundancy**:
   - RDS with multi-AZ deployment
   - Automatic failover to standby instance
5. **Content Delivery**:
   - CloudFront for global content caching
   - Reduces load on origin servers

Security Measures

The architecture implements a defense-in-depth security approach:

1. **Network Security**:

- VPC isolation with public/private subnet separation
- Security groups for instance-level firewall
- NACLs for subnet-level security
- Route tables controlling traffic flow

2. **Access Control**:
   - VPN access for internal users
   - IAM roles for fine-grained permissions
   - Secrets Manager for credential protection
3. **Data Protection**:
   - Certificate Manager for SSL/TLS
   - Private subnets for database resources
   - Encrypted data in transit and at rest
4. **Monitoring and Compliance**:
   - Security groups audit and logging
   - Controlled routes via custom route tables

Networking Details

1. **Subnet Organization**:
   - Public subnets contain internet-facing resources
   - Private subnets host application and database layers
   - Custom route tables define traffic paths
2. **Traffic Flow**:
   - Inbound: Internet Gateway → ALB → EC2 instances
   - Outbound from private: EC2 → NAT Gateway → Internet
   - Internal: Direct routing between subnets
3. **Security Controls**:
   - Route tables restrict traffic paths
   - NACLs provide stateless packet filtering
   - Security groups provide stateful connection control

Scaling and Performance Optimization

1. **Auto Scaling**:

- EC2 instances scale based on CPU, memory, or custom metrics
- Maintains performance during traffic spikes

2. **Caching Strategy**:
   - ElastiCache reduces database load
   - CloudFront caches content at edge locations
3. **Load Distribution**:
   - ALB spreads traffic across instances
   - Global Accelerator improves global performance

Disaster Recovery

1. **Backup Mechanisms**:
   - RDS automated backups
   - S3 for long-term storage
   - Database replication across availability zones
2. **Failover Procedures**:
   - RDS automatic failover to standby
   - EC2 auto-scaling replaces unhealthy instances
   - ALB routes traffic away from problematic zones

Cost Optimization Considerations

1. **Resource Allocation**:
   - Auto Scaling matches capacity to demand
   - Appropriately sized instance types (t2.m)
2. **Storage Tiering**:
   - S3 for cost-effective object storage
   - ElastiCache for high-performance data access
3. **Network Optimization**:
   - Public/private subnet separation reduces NAT Gateway costs
   - CloudFront reduces origin server load

Conclusion

This AWS architecture provides a robust, secure, and scalable environment for hosting a WordPress application. The design incorporates AWS best practices for high availability, security, and performance optimization. The multi-tier approach with public and private subnets ensures that sensitive components remain protected while still allowing necessary internet access for content delivery.

The use of managed services like RDS, ElastiCache, and CloudFront reduces operational overhead while improving reliability and performance. Auto-scaling capabilities ensure the system can handle variable workloads efficiently while maintaining cost-effectiveness during periods of lower demand.

Overall, this architecture represents a production-ready solution suitable for mission-critical WordPress applications requiring high availability, security, and scalability.