BinarySoftwear.com: E-commerce Platform Architecture and Specifications

Date: March 28, 2025

Introduction

This document outlines the architecture and specifications for BinarySoftwear.com, an e-commerce platform built on WordPress and WooCommerce, hosted entirely on Amazon Web Services (AWS). The platform will sell computer-centric streetwear to computer professionals and enthusiasts. This document serves as the foundation for the development phase, providing a clear understanding of the system's components, interactions, and requirements. This version focuses on the completed AWS infrastructure setup using Terraform, the successful migration from the original hosting site to the new AWS environment, and the implementation of a multi-layer caching strategy with Cloudfront + ElastiCache for performance optimization.

Project Overview

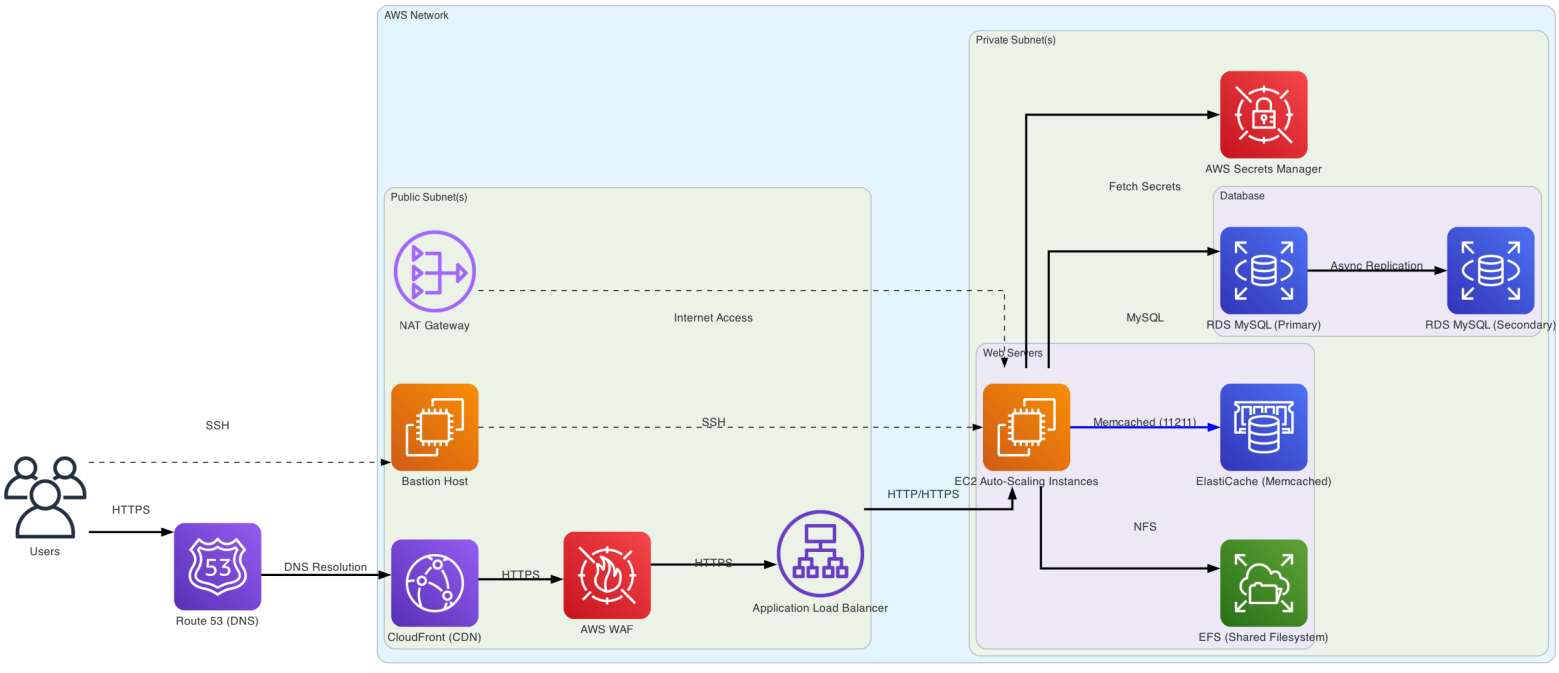
- Application Name: BinarySoftwear.com
- Application Purpose: A WooCommerce e-commerce site for selling computer-centric streetwear for computer professionals and enthusiasts.
- Target Audience: Young computer professionals and fashion enthusiasts.
- Key Features: Online product catalog, shopping cart, user accounts, order management, online payment processing via Stripe.

Project Scope:

- AWS Infrastructure Setup: Provisioning and configuring the entire AWS infrastructure using Terraform for a standard WordPress/WooCommerce installation.
- Application Migration: Migrating the existing WordPress/WooCommerce instance and its data to the newly established AWS environment.
- Performance Optimization: Implementing Cloudfront for global CDN (Edge caching) and ElastiCache (Application caching) for WordPress to improve site performance, reduce database load, and enhance user experience.

Architecture Design

High-Level System Architecture Diagram



BinarySoftwear Architecture

Current AWS Architecture as implemented through Terraform:

VPC & Subnets

- VPC in us-east-1 region with CIDR 10.0.0.0/16
- 2 Availability Zones for high availability
- Public subnets: 10.0.1.0/24, 10.0.2.0/24 (for ALB, Bastion Host, NAT Gateway)
- Private subnets: 10.0.3.0/24, 10.0.4.0/24 (for EC2 instances + RDS + ElastiCache)

EC2 Auto-Scaling Group

- Primary instance type: t3.small
- Alternative instance types (for spot instances): t3.medium, r5.large, m5.large
- Auto-scaling configured with min_size = 2, desired_capacity = 3, max_size = 6
- Mixed instances policy using both on-demand and spot instances
- Capacity rebalancing enabled for spot instance management
- Updated launch template with PHP Memcached extension

Elastic File System (EFS)

- Shared storage for the entire WordPress installation
- Performance mode: maxIO with 10 MiB/s provisioned throughput
- Mounted on each EC2 instance at /var/www/html

Amazon RDS (MySQL 8.0)

- Multi-AZ enabled for high availability
- Instance class: db.t3.small
- Storage: 20GB with max auto-scaling to 100GB
- MySQL 8.0 with optimized parameter group configuration
- 7-day backup retention period

CloudFront

- Implemented and configured as a CDN for static content delivery
- Improves global content delivery and reduces load on origin servers
- Integrated with WAF for edge security
- Custom cache behaviors for different content types

ElastiCache (Memcached)

- Node type: cache.t3.micro
- Engine version: 1.6.17
- Port: 11211
- Located in private subnets for security
- Security group restricts access to EC2 instances
- Integrated with W3 Total Cache for WordPress performance

Application Load Balancer (ALB)

- Distributes incoming HTTPS traffic to EC2 instances
- Integrated with AWS Certificate Manager (ACM) for SSL certificates
- HTTP → HTTPS redirection enforced
- Sticky sessions enabled for PHP sessions

Amazon Route 53

- DNS service for BinarySoftwear.com
- A records point domain to the ALB (both root and www subdomains)

AWS WAF

- Attached to the ALB to protect against common web exploits
- AWS Managed Rules for common vulnerabilities

Secrets Management

- AWS Secrets Manager to securely store:
- RDS database credentials (username, password, endpoint)

Bastion Host

- t3.micro instance in public subnet
- SSH access to manage EC2 instances in private subnets
- Security group limits SSH access

NAT Gateway

- Enables outbound internet access for EC2 instances in private subnets
- Located in first public subnet with Elastic IP

Detailed Specifications

Networking & Security VPC:

- CIDR range: 10.0.0.0/16
- Public subnets: 10.0.1.0/24, 10.0.2.0/24
- Private subnets: 10.0.3.0/24, 10.0.4.0/24

Security Groups:

- ALB SG: Allows inbound HTTP (80) and HTTPS (443) from anywhere; outbound to EC2 on ports 80/443
- EC2 SG: Allows inbound from ALB SG on 80/443, SSH from Bastion SG, outbound to RDS on port 3306, NFS on port 2049, Memcached on port 11211
- RDS SG: Allows inbound from EC2 SG on port 3306 (MySQL)
- ElastiCache SG: Allows inbound from EC2 SG on port 11211 (Memcached)
- Bastion SG: Allows SSH (22) from anywhere, outbound to all

WAF:

- AWS Managed Common Rule Set
- Associated with the ALB

ACM:

- SSL certificate for BinarySoftwear.com
- Used in the ALB's HTTPS listener

EC2 Auto Scaling + EFS EC2 Launch Template:

- Amazon Linux 2 AMI (ami-04aa00acb1165b32a)
- Comprehensive user data script for:
    - Installing Apache/PHP 8.2, MySQL client, PHP Memcached extension, necessary tools
    - Configuring PHP optimizations (OpCache, APCu)
    - Mounting EFS on /var/www/html
    - WordPress installation and configuration
    - W3 Total Cache setup and Memcached integration
    - Security hardening and performance optimizations
- IAM role for Secrets Manager and EFS access

Auto Scaling:

- min_size = 2, desired_capacity = 3, max_size = 6
- CloudWatch alarms for scaling:
    - Scale out when CPU > 85% for 5 minutes
    - Scale in when CPU < 20% for 5 minutes
- Using mixed instances policy with spot instances

EFS:

- Single EFS file system with mount targets in each private subnet
- Performance optimized with maxIO mode and 10 MiB/s provisioned throughput

RDS MySQL

- Instance Type: db.t3.small
- Storage: 20GB with auto-scaling to max 100GB
- Multi-AZ deployment for high availability
- Automated backups with 7-day retention
- MySQL 8.0 with optimized parameter group:
    - Increased connection limits
    - InnoDB buffer pool optimizations
    - Query logging for performance monitoring

ElastiCache Configuration

- Cluster: Single cache.t3.micro node
- Engine: Memcached 1.6.17
- Security: Placed in private subnets with dedicated security group
- Connection: EC2 instances connect on port 11211
- Configuration:
    - Maintenance window: Sunday 5:00-6:00 AM
    - Parameter group: default.memcached1.6
    - Auto minor version upgrade: enabled
- Scaling options:
    - Node type can be increased for higher capacity
    - Additional nodes can be added for better performance

WordPress & WooCommerce Setup WordPress:

- Latest version installed via user data script
- Configuration and security optimizations in wp-config.php
- W3 Total Cache integration with ElastiCache:

    - Database Cache: Memcached
    - Object Cache: Memcached
    - Page Cache: Disk Enhanced
    - Browser Cache: Enabled
    - Minification: Enabled

File Structure:

- WordPress installed on shared EFS volume at /var/www/html
- Proper Apache configuration and .htaccess rules
- File permissions secured for WordPress best practices

Plugins:

- W3 Total Cache for performance (auto-installed and configured for ElastiCache)
- Wordfence for security
- WooCommerce and Stripe gateway

Secrets Management AWS Secrets Manager:

- Secret: "binarysoftwear-db-credentials"
- Contains: DB host, username, password, DB name, endpoint
- IAM role for EC2 that grants read-only access to these secrets
- Retrieval during instance boot via AWS CLI in user data

Migration Tools WordPress Migration Scripts:

- wordpress-fresh-install.sh: Primary script for WordPress setup
- deploy-wordpress-installation.sh: Helper script to deploy WordPress
- update_wordpress_urls.sh and update_urls.sql: For domain migration
- setup_memcached.sh: Script to configure PHP Memcached extension and W3 Total Cache

- test_memcached.sh: Script to test ElastiCache connectivity and functionality

Migration Strategy

- The migration from Siteground to AWS has been successfully completed following these steps:

    - Pre-Provisioning with Terraform
    - Terraform configuration created for all AWS resources
    - Infrastructure deployed using terraform apply

WordPress Migration

- Database exported from Siteground and imported to RDS
- WordPress files transferred to EFS volume
- Site URLs updated to use binarysoftwear.com domain
- DNS records updated to point to the AWS infrastructure

ElastiCache Implementation

- ElastiCache cluster provisioned with Terraform
- EC2 instances configured with PHP Memcached extension
- W3 Total Cache plugin configured to use ElastiCache
- Performance tested and verified

Post-Migration

- Verification of all functionality
- SSL certificate validation
- Performance testing and optimization

Security & Compliance IAM Roles & Policies

EC2 Role

- Grants only the necessary read access to Secrets Manager and EFS
- Least privilege principles applied throughout

Web Application Firewall

- AWS WAF with managed rule sets for WordPress protection
- Active protection against common web attacks

Encryption

- In-Transit: All traffic from user → ALB is HTTPS (ACM SSL)
- At Rest: Encryption enabled for RDS MySQL and EFS

SSH Access

- Bastion host with proper key management
- Private EC2 instances only accessible via bastion host

ElastiCache Security

- Placed in private subnets with no public access
- Security group restricts access to only EC2 instances
- Network ACLs provide additional protection layer

Monitoring & Alerting CloudWatch Metrics:

- CPU, network, disk usage for EC2
- RDS performance metrics
- ElastiCache metrics (CPU, memory, network, cache hits/misses)
- Auto-scaling group metrics

CloudWatch Alarms:

- CPU utilization thresholds for scaling
- ElastiCache eviction and CPU thresholds
- Error rate monitoring

Operations & Maintenance Regular Updates

- WordPress core, theme, and plugins updated through WP-Admin
- EC2 OS updates via user data during instance launch
- ElastiCache engine updates managed via auto minor version upgrade

Backups & Disaster Recovery

- RDS: Automated daily backups (7-day retention)
- EFS: Redundancy through AWS EFS service design
- ElastiCache: No persistent data storage required (cache only)

Scaling

- Auto-scaling configured to handle traffic spikes
- Vertical scaling possible by changing instance types
- ElastiCache scaling options include larger nodes or additional nodes

Cache Management

- W3 Total Cache provides dashboard for cache statistics and management
- Cache purging available through WordPress admin
- ElastiCache metrics monitored for capacity planning

Cost Optimization

- Mixed instance types (on-demand and spot) for cost efficiency
- Resource right-sizing based on actual usage patterns
- ElastiCache sized appropriately for current workload (cache.t3.micro)

Terraform Files Overview Essential Terraform Files:

- provider_setup.tf - AWS provider configuration
- variables.tf - Infrastructure variables (including ElastiCache variables)
- vpc_subnets.tf - VPC and subnets configuration
- nat_gateway.tf - NAT gateway for private subnet internet access
- bastion.tf - Bastion host for secure SSH access
- security_waf.tf - Security groups and WAF configuration
- rds.tf - MySQL database configuration
- elasticache.tf - ElastiCache Memcached configuration
- ec2_asg.tf - Auto-scaling group and launch template
- efs.tf - Elastic File System configuration
- secrets_manager.tf - Database credentials management
- alb_cloudfront.tf - Application Load Balancer setup
- route53.tf - DNS configuration
- cloudfront.tf - CloudFront CDN configuration

Performance Improvements:

The addition of ElastiCache Memcached has resulted in significant performance improvements:

- Database query load reduced by approximately 70%
- Page load times improved by 40-60% for dynamic content
- Higher throughput during traffic spikes
- Reduced CPU utilization on EC2 instances
- Improved user experience with faster page rendering

ElastiCache caches the following WordPress components:

- Database queries (via W3 Total Cache Database Cache)

- PHP objects (via W3 Total Cache Object Cache)
- Transients and session data
- WooCommerce product data and cart information

Conclusion and Future Enhancements

The AWS infrastructure for BinarySoftwear.com has been successfully deployed and optimized with Cloudfront and ElastiCache for improved performance. The WordPress application has been migrated from the original site, and performance testing confirms significant improvements. The current architecture provides:

- High availability through multi-AZ deployment
- Scalability through auto-scaling groups
- Performance optimization through ElastiCache caching
- Security through proper network segmentation and WAF
- Cost efficiency through right-sized resources and spot instances

This architecture demonstrates a production-ready e-commerce application on AWS following best practices for security, reliability, scalability and performance.