Cloud Infrastructure Modernization with AWS

Introduction:

The Client is a leader in customer-centric marketing and media strategy, combining deep analytics with cross-industry expertise to drive measurable business growth. The company specializes in helping major retail brands, media agencies, and media companies optimize customer connections, increase sales, and improve marketing ROI through data-driven campaigns and technology-enabled solutions.

Their expertise spans increasing store visits for retailers, refining coupon strategies for restaurant chains, and building integrated multi-channel media platforms that enhance both in-store and digital performance, consistently achieving notable cost efficiencies and higher profitability for clients.

Client Details:

Name: Confidential | **Industry**: Software | **Location**: USA

Technologies:

- **AWS Services:** VPC, EC2, CloudFront, WAF, S3, IAM, CodeDeploy, SSM, Aurora MySQL (RDS), CloudWatch, SNS.
- **Application:** Django with Gunicorn (WSGI), NGINX, Supervisor, Celery.
- **DevOps/Monitoring:** AWS CodeDeploy, GitHub Actions (via OIDC), New Relic for APM/infrastructure monitoring.
- **Security:** Custom IAM roles, bastion hosts, private subnets, AWS WAF and Shield, CloudFront edge security, and custom security groups.

Cloud Infrastructure Modernization with AWS

Problem Statement:

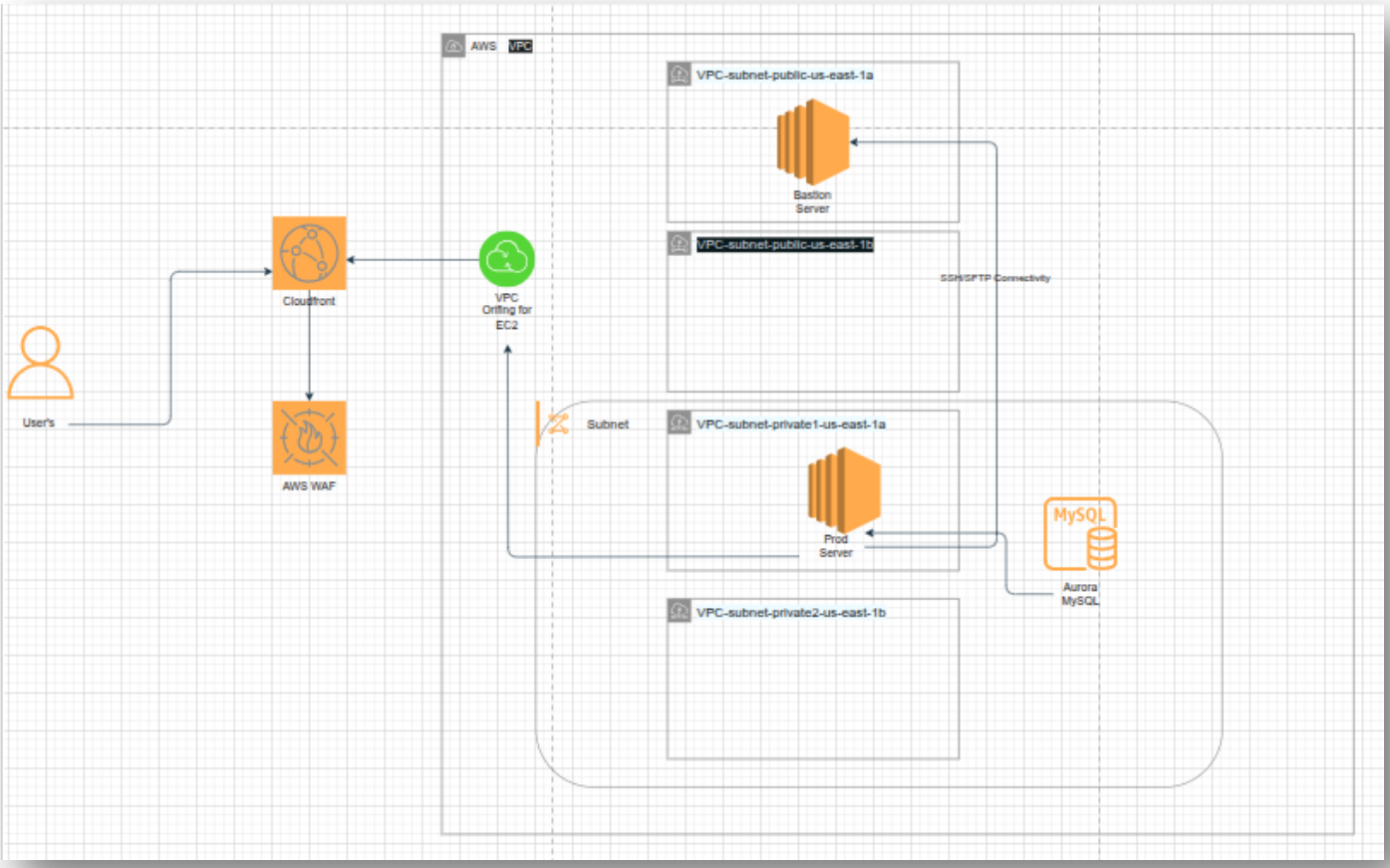
The client's environment was previously hosted on Rackspace but lacked critical security features. Data was not consistently encrypted at rest, and WAF-based threat filtering was absent. Additionally, secure connectivity was limited, with no enforced private subnets or encrypted inter-service connections. The infrastructure ran outdated components—Python 2.7, Ubuntu 14.04, MySQL 5.5—introducing risks and operational inefficiencies. Backup and restore processes were slow and unreliable, while manual deployments and log management led to delays, errors, and limited visibility. Real-time monitoring and alerting were missing, increasing the chance of unnoticed failures or performance issues, and negatively impacting uptime and user experience.

Solution Provided:

We redesigned the client’s infrastructure on AWS, embracing cloud-native security, automation, and high availability best practices. Sensitive data is now safeguarded with multi-layered encryption, private subnet isolation, and automated AWS WAF rules that proactively block threats. Legacy limitations were overcome with fast, reliable backup and restore workflows and fully automated CI/CD pipelines that drastically reduce manual effort and error. Integrated monitoring with CloudWatch, New Relic, and SNS ensures real-time alerts for incidents, enabling rapid remediation. The solution delivers a scalable, resilient, and secure environment that simplifies management and empowers the client to focus on business growth.

Cloud Infrastructure Modernization with AWS

Architecture Diagram:



This architecture leverages CloudFront's global reach and AWS networking best practices to ensure secure, reliable, and performant access to private backend servers.

Infrastructure Design:

- **Virtual Private Cloud (VPC):** The core of the architecture is a dedicated AWS VPC, segmented into separate **public** and **private** subnets across at least two Availability Zones, ensuring both **high availability** and **layered security** for its applications and databases.

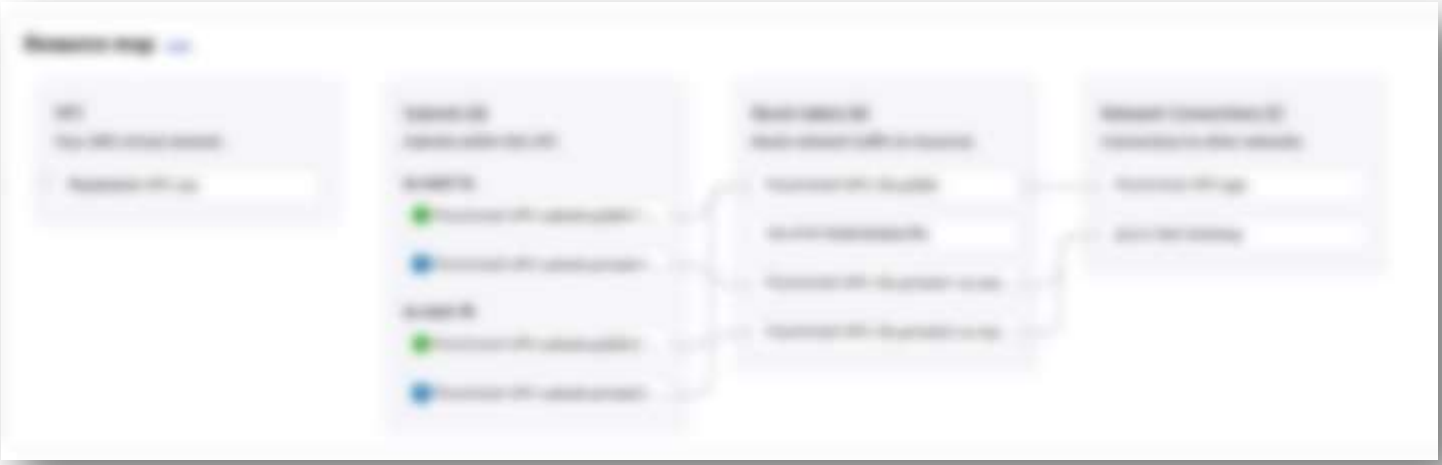
- **CIDR Range Selection**: Choosing the best CIDR range is crucial for scalability and effective IP address management. CIDR is large enough to accommodate current resources and future expansion, covering growth in EC2, RDS, ELB, and other services.

Cloud Infrastructure Modernization with AWS

- **VPC default security group**: The VPC default security group is automatically created and is permissive by default: all members can talk to each other. For secure environments, it's best practice to restrict or avoid the use of the default security group for any public-facing or critical resources. Instead, create custom security groups tailored to each workload and flow requirement.

    - **Public Subnet:** Hosts a secure EC2 bastion server for controlled SSH/SFTP access by administrators. Bastion access is restricted using IAM policies and security groups to allow only specific IPs, ensuring secure perimeter management.

    - **Private Subnet:** Runs the production EC2 server, accessible only through the bastion via SSH. Private subnet houses application workloads (Django app, NGINX, Gunicorn, Supervisor, Celery).
      * Aurora MySQL is deployed in a private subnet group for enhanced data protection.



- **Security Groups & IAM: A** Security group acts as a virtual firewall, controlling inbound and outbound traffic to resources within the VPC based on defined rules. In this architecture, security groups are tightly managed to enforce application-layer isolation and restrict access to only what is necessary for each component.

Cloud Infrastructure Modernization with AWS

- **Bastion security group**: Allows only trusted IPs via SSH/SFTP.
  * Currently, SSH is permitted for the bastion host server, but the SSH service runs on a custom (non-default) port to reduce risk from automated attacks and common scanning tools.

- **Application server security group:** Application server only accepts SSH connections from the bastion host, again using a custom SSH port for extra protection. Inbound HTTP/HTTPS traffic is permitted only from AWS CloudFront, using the AWS-recommended list of CloudFront edge IP ranges. This ensures that only traffic passing through CloudFront (with WAF filtering), not the open internet, ever reaches the application tier.

Cloud Infrastructure Modernization with AWS

- **Database security group:** Port 3306 (MySQL) is exposed only to the production application server within the private subnet. Both the application server and the RDS/Aurora database are isolated in private subnets. No external network or public internet can directly access them.
  * Zero Public Database Exposure: Databases remain isolated, with no direct public or even VPC-external network exposure.

- **CloudFront as a Secure Entry Point:**
  CloudFront and CloudFront's VPC origin integration play a critical role in securing and optimizing access to the application, especially when the EC2 server is placed inside a private subnet.
  - AWS CloudFront acts as a global content delivery network (CDN) that receives all user requests before they reach the application.
  - CloudFront communicates with the private EC2 server using a **VPC endpoint,** making the app available only via CloudFront while keeping it isolated from public IP exposure.
  - Improved performance: Static assets and common content are delivered with minimal latency from CloudFront edge locations, taking load off backend resources.
- **VPC origin access** is enforced in CloudFront's distribution settings: only requests from CloudFront's specific IP address ranges are allowed through the EC2's security group and NGINX/proxy layer.

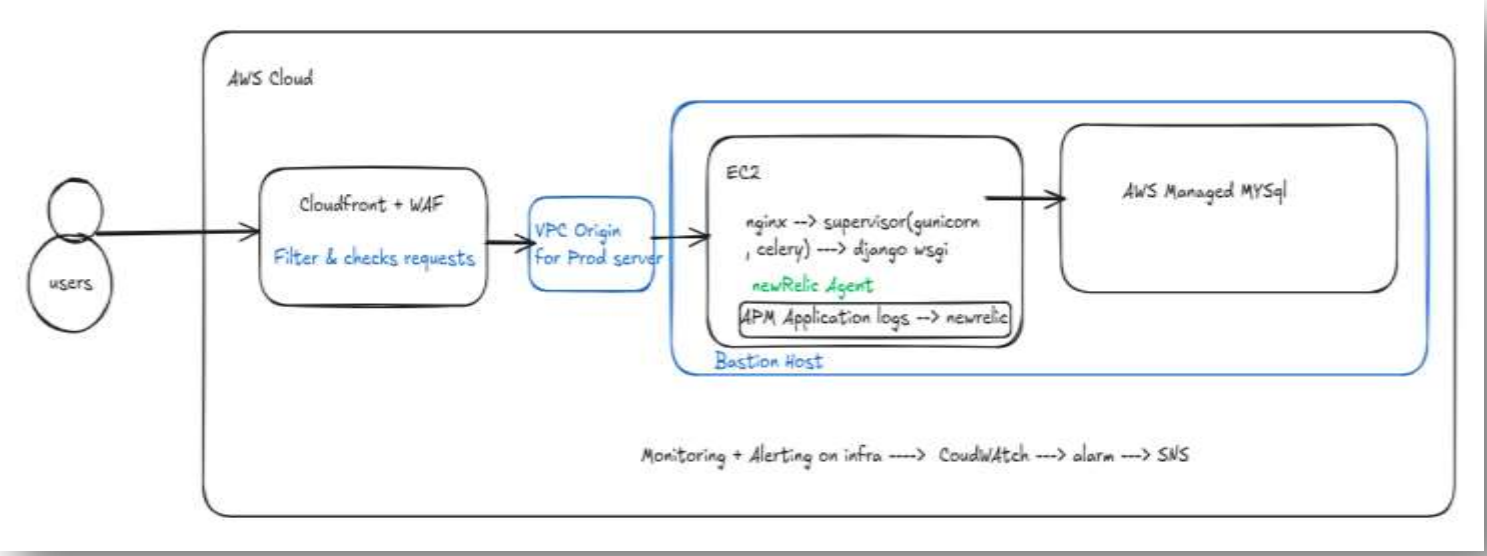
- **Security and Performance Benefits**
  - Complete isolation of the application server in a private subnet drastically reduces the attack surface.

Cloud Infrastructure Modernization with AWS

- CloudFront communicates with the private EC2 origin via CloudFront **VPC origins**, keeping the instance in private subnets with **no public IP** while allowing only CloudFront to reach it.
- Seamless integration with AWS WAF: All traffic to the backend is pre-filtered, preventing exploits and unauthorized access.
- **AWS WAF (Web Application Firewall)** is tightly integrated with CloudFront, inspecting all inbound traffic for common web exploits (SQL injection, XSS, etc.), DDoS attack patterns, and enforcing IP rate limits.
- Traffic filtering is performed at the edge, reducing load on origin servers and blocking threats before they enter the private AWS network.

Cloud Infrastructure Modernization with AWS

Application Flow and Security Controls:



- User Request Flow
  - End-users access applications through CloudFront, which provides edge caching and DDoS mitigation.
  - AWS WAF inspects, filters, and blocks malicious requests before they reach the origin.
  - Only legitimate traffic passes through to the VPC origin for the production server.
- Compute Layer
  - All user-facing workloads run under process managers (Supervisor) for resilience.
  - NGINX acts as a reverse proxy to Gunicorn/Django WSGI for application logic.
  - Celery handles background jobs securely in the same workflow.
- Monitoring & Logging
  - **New Relic agents** are deployed on both application and infrastructure layers, collecting Application Performance Metrics (APM), infrastructure health, and custom logs for end-to-end observability.
  - **AWS CloudWatch** ingests metrics and logs from all critical services. Alerting rules trigger SNS notifications for incident response.
- Database Security
  - Aurora MySQL (AWS RDS) is only accessible from the private subnet. IAM roles and security groups further restrict database connectivity.

Cloud Infrastructure Modernization with AWS

- Data Backups and S3
    - Regular automated backups are stored on S3, following encryption-at-rest best practices and strict access policies.

Security and Monitoring:

- Automated Monitoring & Alerting: Centralized, actionable alerting for operational issues and security anomalies through CloudWatch and third-party integrations.
- Access Control & Auditing: IAM, MFA, and strict boundary access are enforced and logged.
- Continuous Updates & Patch Management: Environments are regularly updated via CI/CD workflows, minimizing vulnerabilities.
- Compliance Readiness: The infrastructure is built with PCI DSS and similar benchmarks in mind, supporting regular audits with detailed logging.

Impact on Cost, Performance, and Other Factors:

AWS offers a wide variety of flexible instance types and pricing models, enabling precise tuning of both compute and storage resources. We select the best-suited servers and database instances based on our application requirements, resulting in significant cost savings. By utilizing Reserved Instances with 1-year or 3-year commitment terms, we further optimize costs, achieving substantial savings compared to on-demand pricing. In contrast, Rackspace provides simpler pricing options but with fewer customization possibilities and generally higher costs for managed services. This granular cost control on AWS allows us to adhere to budget constraints while fulfilling our performance needs.

Performance Improvements:

AWS-managed services and automation contributed to noticeable performance gains. We achieved faster image creation pipelines and implemented automated backups with high availability to ensure data durability. Utilizing AWS CloudFront CDN significantly enhanced **content delivery speed**, improving the end-user experience through **reduced latency**. These improvements combined to provide a robust and responsive infrastructure capable of scaling according to traffic and operational demands.

Cloud Infrastructure Modernization with AWS

Enhanced Security:

The integration of **AWS WAF** strengthened **security** by mitigating **application-layer** attacks. **Role-based IAM policies** ensure precise access control across services. These measures safeguard sensitive data and infrastructure while maintaining.

Conclusion:

Our migration to AWS fundamentally transformed the client's infrastructure, delivering substantial cost savings, enhanced security, faster performance, and streamlined management. The private cloud design, combined with automated deployments, real-time monitoring, and advanced security controls, provides a strong foundation for ongoing scalability and compliance. By moving from Rackspace to AWS, the client now operates in a resilient, secure, and efficient cloud environment that supports accelerated growth and innovation.