RankLab Cloud Architecture on AWS

Overview

RankLab leverages Amazon Web Services (AWS) to provide a reliable, scalable, and secure cloud infrastructure for its data processing and visualization capabilities. The core application is built as a Dash web application deployed on AWS infrastructure with various supporting services. This document outlines the current cloud architecture, highlighting key components and their interactions within the system.

Hosting Environment

RankLab operates on a single Amazon EC2 t2.micro instance, which provides a cost-effective solution for our current workload. The instance runs Amazon Linux 2, offering a stable and well-supported foundation for our application. While this instance type represents an entry-level option within AWS's compute offerings, it has proven adequate for handling our present user traffic and computational needs.

Access to the server for deployment and maintenance is exclusively managed through SSH, employing key-based authentication to ensure secure access. This approach eliminates password-based vulnerabilities and provides a strong access control mechanism. The operations team maintains strict control over the distribution of SSH keys, with access granted only to authorized personnel.

Application Deployment

The Dash application is deployed through a straightforward yet effective process. When updates are ready for production, an engineer connects to the EC2 instance via SSH and pulls the latest code from our version control system. The application itself is executed using Gunicorn, a production-grade WSGI HTTP server.

Gunicorn serves as the backbone of our production environment, providing several critical advantages over Dash's built-in development server. It offers improved reliability through worker process management, automatically restarting workers that fail during execution. Additionally, Gunicorn enables concurrent request handling by distributing incoming traffic across multiple worker processes, significantly enhancing the application's throughput and responsiveness.

The deployment command typically specifies three worker processes, which has been determined as the optimal balance between performance and resource utilization for our

t2.micro instance. This configuration maximizes CPU usage without overloading the system, ensuring responsive user experiences even during periods of moderate traffic.

Storage Architecture

Data persistence within RankLab is primarily handled through Amazon S3, AWS's scalable object storage service. The architecture employs multiple S3 buckets organized according to environment and function. Most critically, we store our machine learning models as serialized Python objects (pickled models) within designated S3 buckets.

The separation between production and development environments extends to our storage architecture, with dedicated buckets for each environment. This segregation prevents development activities from inadvertently affecting production data while maintaining consistent access patterns across environments.

Model versioning is managed through S3's native versioning capabilities. This approach creates a historical record of all models, enabling quick rollbacks if newer versions exhibit unexpected behavior. The application is designed to load these models either during startup or on-demand, depending on configuration settings and operational requirements.

Security Infrastructure

Security permeates every aspect of the RankLab architecture, with AWS Identity and Access Management (IAM) serving as the cornerstone of our access control strategy. The EC2 instance operates with a custom IAM role that defines its permissions within the AWS ecosystem.

The IAM role adheres to the principle of least privilege, granting only the specific permissions required for normal operation. These include the ability to read from designated S3 buckets containing application models, write logs to CloudWatch for monitoring purposes, and access other AWS resources essential to the application's functionality.

Network security is enforced through AWS Security Groups, which function as virtual firewalls controlling inbound and outbound traffic. The security group attached to our EC2 instance allows only necessary connections: HTTP/HTTPS for web traffic and SSH for administrative access. SSH access is further restricted to specific IP ranges, typically including only our corporate network and approved remote working locations.

Additional network protection is provided through Network Access Control Lists (ACLs), which offer stateless filtering of traffic at the subnet level. This dual-layer approach—combining security groups and network ACLs—provides defense in depth against unauthorized access attempts.

Monitoring and Logging

Comprehensive monitoring is essential for maintaining application health and quickly identifying potential issues. RankLab utilizes Amazon CloudWatch to track instance performance metrics, including CPU utilization, memory usage, disk I/O, and network traffic. These metrics provide valuable insights into application behavior and resource consumption patterns.

Application-specific logs are streamed to CloudWatch Logs, creating a centralized repository for log data. This centralization simplifies troubleshooting by providing a single interface for searching and analyzing log entries across the entire application stack. Log retention policies balance the need for historical data against storage costs, typically maintaining logs for 30 days.

Alert mechanisms have been configured within CloudWatch to notify the operations team of critical events. These alerts trigger when predefined thresholds are exceeded, such as sustained high CPU usage, memory pressure, or application errors. Timely notifications enable rapid response to emerging issues, often before they impact end users.

Cost Optimization

RankLab's cloud architecture incorporates several cost optimization strategies. The selection of a t2.micro instance represents a conscious decision to balance performance requirements against infrastructure costs. While this instance type has limitations, it provides sufficient resources for our current needs at a predictable monthly cost.

Storage costs are managed through appropriate selection of S3 storage classes. Frequently accessed models reside in S3 Standard storage, while historical models or backup data may be automatically transitioned to less expensive storage classes like S3 Infrequent Access or S3 Glacier, depending on access patterns and retrieval requirements.

As usage patterns stabilize, we continually evaluate the potential benefits of reserved instances, which could provide significant cost savings compared to on-demand pricing for predictable workloads.

Future Scaling Considerations

While the current architecture effectively serves RankLab's present needs, we have identified several enhancements for future implementation as user adoption grows. The introduction of an Elastic Load Balancer would distribute incoming traffic across multiple application instances, improving both reliability and performance.

Auto Scaling groups would complement the load balancer by dynamically adjusting the number of running instances based on demand. This approach would maintain responsiveness during usage spikes while automatically reducing capacity during periods of low activity, optimizing both performance and cost.

For more advanced scaling and deployment capabilities, we are considering a migration to container-based deployment using Amazon Elastic Container Service (ECS) or Elastic Kubernetes Service (EKS). Containerization would provide greater environment consistency, simplified deployments, and more granular resource allocation.

Conclusion

RankLab's AWS cloud architecture provides a robust foundation for current operations while allowing for future growth. By leveraging AWS services for compute, storage, security, and monitoring, we have created a reliable and cost-effective infrastructure that aligns with both our technical requirements and business objectives. As RankLab continues to evolve, this architecture will adapt and scale to meet changing demands while maintaining our commitments to security, reliability, and performance.