

# AWS FOR AI ANALYTICS

## MULTI-REGIONALITY, HIGH AVAILABILITY, AND SECURITY WITH AWS

Our client is currently in stealth mode, as are many startups in the AI space. For the purposes of this article, we will say that they are a retail analytics startup that has engaged us to develop software for optimizing supply chain logistics and predicting consumer demand. We worked with the Founder and CEO previously for many years until he ultimately exited his former company after its sale. The new venture's offerings would include a trend analysis and a mobile application using AI to forecast product popularity and inventory needs. Their core clientele consists of mid-sized fashion retailers who manage diverse product lines and require precise demand forecasting for inventory management across multiple store locations.

## THE OBJECTIVE

They aimed to ensure a seamless and intuitive user experience by automatically routing users to their appropriate data storage region. We would facilitate strict data residency requirements demanded by retailers, minimize latency, and improve overall application performance.

To achieve these objectives, we implemented a multi-region architecture utilizing Amazon Web Services (AWS). This approach ensures users interact seamlessly without manual region selection while strictly adhering to data residency regulations. In addition to the expected outcomes of reduced latency and seamless functionality, we needed to meet stringent data residency rules through precise and automated regional data handling for compliance.

Ensuring a seamless and secure user experience across the globe presents a number of challenges. We designed an architecture to tackle these by using Amazon Web Services (AWS) to provide a scalable, multi-region setup that supports a React-based front end and a robust backend with strong authentication protocols. This architecture is particularly suited to professionals developing enterprise-grade software that demands high performance, reliability, and security.

## ARCHITECTURE OVERVIEW

The architecture is built to support applications with global reach and high demand for availability. It leverages AWS's suite of services to ensure that users experience low latency, robust security, and high availability—irrespective of their geographical location.



Key components of this architecture include the following:

- A React front end built with TypeScript
- CloudFront for content delivery and caching
- Amazon Cognito for user authentication and identity management
- Lambda functions for token enrichment and request routing
- Elastic Container Service (ECS) with Virtual Private Clouds (VPCs) for the backend
- ElastiCache for caching and session storage
- Amazon RDS (PostgreSQL) for data storage
- AWS Global Accelerator for optimal traffic routing
- AWS CodeBuild and CodePipeline for deployment management
- AWS Secrets Manager and Systems Manager for secure credential storage

## Front End Architecture: React and CloudFront

The front end is developed using React with TypeScript. This combination ensures a robust and scalable user interface with type safety, aiding developers in maintaining and scaling the application efficiently. The front end application is hosted and served via AWS CloudFront, a content delivery network (CDN) that ensures low latency and global availability.

CloudFront caches content at locations close to the users (edge locations). When a user requests the site, CloudFront quickly checks its cache to provide the file. If unavailable, it fetches it from the origin, optimizing performance and reducing costs.

## User Authentication with Amazon Cognito

Authentication is a critical component of any secure architecture. Here, Amazon Cognito is utilized for managing user identity and handling federated logins. When a user successfully signs in, a custom AWS Lambda function enriches the JSON Web Token (JWT) with user permissions and regional metadata by pulling additional claims from a PostgreSQL database. This enriched token enables secure and authenticated API requests by clients.

## API Requests Handling

All incoming API requests traverse through CloudFront, acting as a global entry point. AWS Lambda@Edge functions play a pivotal role here by intercepting each request, validating the JWT token, and subsequently routing the request to the appropriate regional application load balancer based on the enriched token's information.

This dynamic routing is powered by AWS Global Accelerator, which selects the optimal path for data transfer to ensure requests hit the nearest regional endpoint. It also provides automatic failover; if a regional service fails, requests are rerouted to the next healthy region, ensuring minimal downtime and enhanced availability.

## Backend Support: ECS and ElastiCache

The architecture supports the backend through Amazon ECS, which runs containerized services within isolated Virtual Private Clouds (VPCs). These VPCs contain logic and cache layers, specifically utilizing Amazon ElastiCache to handle workloads efficiently. This setup allows for scaling compute resources horizontally, ensuring the backend can handle varying loads without performance degradation.

## Multi-Region Deployment and Failover

The deployment spans multiple AWS regions, creating identical setups that include ECS clusters, load balancers, and databases in each. This redundancy ensures that, even if one region faces a service interruption, others are available to take over, preventing service downtimes.

AWS Global Accelerator plays a crucial role in managing these multiple regional setups by directing user requests based on the user's location information in the JWT, ensuring optimal request routing and resilience to region-specific failures.

## Deployment, Configuration, and Secrets Management

Deployment processes are automated through AWS CodeBuild and CodePipeline, facilitating continuous integration and delivery (CI/CD). This automation accelerates deployment cycles, reduces manual errors, and enhances the overall efficiency of software updates and releases.

For managing configurations and secrets, AWS Secrets Manager and Systems Manager are employed. These services securely store credentials and configuration data, offering fine-grained permissions to access such sensitive information, thus enhancing the architecture's overall security posture.

## SUMMARY

This architecture represents modern cloud-native service deployments by leveraging AWS's powerful suite of services. It exemplifies best practices for infrastructure design, including content delivery, authentication management, dynamic request routing, and deployment automation. This platform ensures the following:

- **Seamless User Experience:** Quick content delivery and efficient session management
- **High Availability:** Multi-region deployments with automatic failover
- **Robust Security:** End-to-end encryption and secure identity management
- **Smart Routing Mechanisms:** Utilizing user location and access claims to direct traffic efficiently

Professionals in the software development industry can benefit from this type of architecture to solve complex global scaling challenges while maintaining a focus on performance and security to cater to a global audience.

## References

- [AWS CloudFront](#)
- [Amazon Cognito](#)
- [AWS Lambda@Edge](#)
- [Amazon ElastiCache](#)
- [AWS Global Accelerator](#)
- [AWS CodePipeline](#)
- [AWS Secrets Manager](#)
- [AWS Systems Manager](#)



There is no need to waste time navigating complex cloud challenges alone. As an AWS partner with additional experience in Microsoft Azure and Google Cloud, First Factory offers various cloud-based software solutions and services. Last year we worked with over fifty clients across AI, education, FinTech, healthcare, professional services, sports and entertainment, and more to bring efficient cloud solutions to their organizations. We can help identify platforms, set up and manage your infrastructure, and evaluate your setup to maximize utilization and keep costs predictable and under control.

[CONTACT US](#)