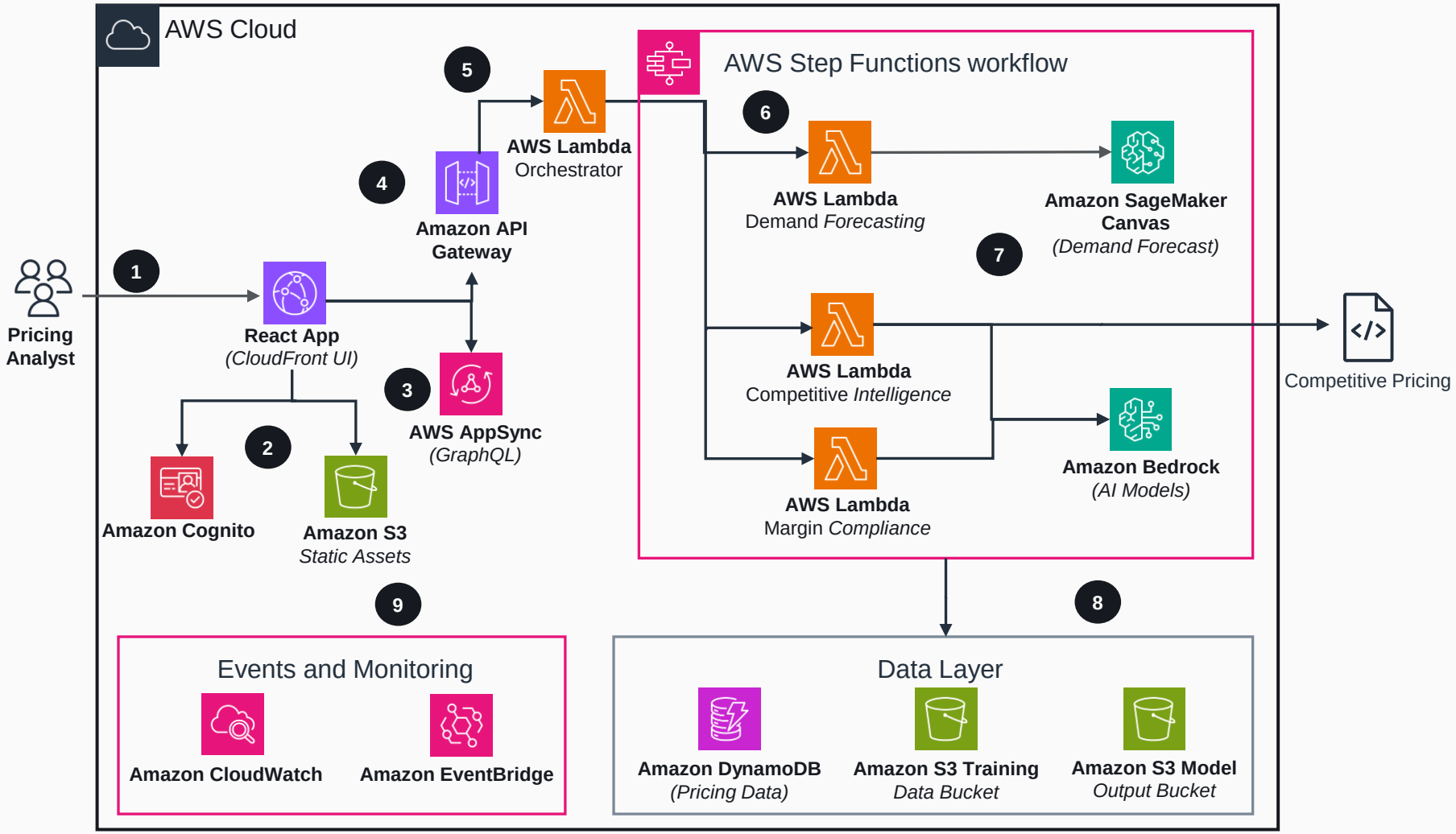
Guidance for a Retail Pricing Agent on AWS

This architecture diagram shows an AI-driven retail pricing system using specialized agents orchestrated by AWS Step Functions and Lambda for demand forecasting, competitive intelligence, and margin compliance.



1. The Pricing Analyst accesses the React frontend application served through **AWS CloudFront**. They select a product from the catalog that they want to price.
2. Authentication is handled by **Amazon Cognito**, providing secure user login and session management. Static web application assets are stored in **Amazon Simple Storage Service (Amazon S3)**.
3. Real-time subscriptions and **GraphQL** queries flow through **AWS AppSync**.
4. API requests route through **Amazon API Gateway** (REST) to backend services.
5. The Orchestrator **Lambda** receives requests and triggers **AWS Step Functions** to coordinate workflows.
6. Step Functions invokes three specialized **AWS Lambda** functions in parallel, each implementing a custom agent using Strands framework: **Demand Forecasting**, **Competitive Intelligence**, and **Margin Compliance**.
7. Agents leverage **Amazon Bedrock,** a fully managed service for generative AI applications with foundation models from leading AI companies, and **SageMaker Canvas** for demand forecasting.
8. Pricing data is persisted in **DynamoDB**; model artifacts and training data are stored in **Amazon S3** buckets.
9. **CloudWatch** provides monitoring and **EventBridge** handles event-driven communication between services.

Reviewed for technical accuracy March 22, 2026

AWS Reference Architecture