Technical Implementation Deep-Dive Report

GlobalTech Financial Services

Industry:	Financial Technology
Assessment Type:	Technical Implementation
Assessment Date:	2025-10-08
Current State:	Pilot Phase
Tech Stack:	

Prepared for:

GlobalTech Financial Services

Prepared by:

Cloud202 Technical Architecture Team

CONFIDENTIAL - GlobalTech Financial Services Technical Assessment

Report Generated: October 17, 2025

Current State Assessment

GlobalTech Financial Services operates a hybrid infrastructure with on-premises data centers hosting core portfolio management systems (BlackRock Aladdin) and client PII, while leveraging AWS for non-production workloads and limited cloud-native services. The current architecture consists of a three-tier monolithic application stack running on VMware ESXi clusters with Oracle RAC databases for transactional workloads and a nascent Snowflake data warehouse on AWS for analytics. Network topology includes MPLS circuits connecting regional offices to dual data centers in New York and Chicago, with AWS Direct Connect (10 Gbps) providing hybrid connectivity. The existing system processes approximately 15,000 daily transactions with peak loads during market open/close, achieving 99.5% availability but experiencing degraded performance during volatility spikes when concurrent database connections exceed 2,000. Current SLAs specify 99.9% uptime for trading hours with P95 response times under 5 seconds, though actual measurements show P95 at 7.2 seconds and P99 at 15+ seconds during peak periods.

Performance bottlenecks manifest primarily in three areas: database contention on the Oracle RAC cluster during concurrent portfolio valuation runs, network latency for Bloomberg Terminal data feeds averaging 800ms due to inefficient polling mechanisms, and batch processing windows extending beyond allocated 4-hour maintenance windows due to linear scaling limitations. The portfolio optimization engine, written in legacy C++ with Python wrappers, exhibits CPU saturation at 85%+ utilization during market hours, with memory pressure causing occasional OOM kills on analysis nodes. Storage I/O patterns show 40% random reads on spinning disk arrays (12K RPM SAS), creating 25ms average latency that cascades through the application stack. The pilot GenAI implementation using OpenAI GPT-4 API operates outside the core infrastructure on a segregated AWS account with basic VPC configuration, processing 1,200 requests daily for 10 advisors with average latency of 3.4 seconds and monthly costs of $12,000, indicating a linear cost scaling challenge that would reach $600K monthly at full deployment without optimization.

Security posture reflects traditional perimeter-based controls with Palo Alto firewalls, Cisco ISE for network access control, and Active Directory for identity management. Encryption at rest uses native Oracle TDE with AES-256, while TLS 1.2 secures data in transit, though certificate management remains manual and error-prone. Audit logging captures database access via Oracle Audit Vault and application logs in Splunk with 90-day hot retention, but lacks comprehensive API-level tracing and correlation across distributed systems. Compliance controls for SEC 17a-4 and FINRA requirements rely heavily on manual processes and quarterly attestations, with evidence collection consuming 160 person-hours per audit cycle. The current secrets management approach uses CyberArk for privileged credentials but lacks dynamic secret rotation and fine-grained access policies. IAM follows a coarse-grained RBAC model with 15 predefined roles, creating excessive privilege accumulation and audit findings around least-privilege violations.

Environmental constraints include significant technical debt in the 12-year-old codebase with 2.3M lines of Java/C++ code, limited containerization experience among the 45-person engineering team, and organizational resistance to cloud migration driven by perceived regulatory barriers. The operations team follows ITIL-based change management with 2-week lead times for production changes, creating friction for agile AI/ML experimentation. Vendor dependencies on Bloomberg, BlackRock, and Oracle create integration complexity and licensing costs exceeding $8M annually. The data engineering team of 8 personnel manages 200TB across disparate systems with inconsistent data quality (87% accuracy), lacking unified governance and lineage tracking. Current cloud spending of $180K monthly across AWS and Snowflake shows poor resource utilization with 40% idle compute capacity and unoptimized storage classes. Skills gaps exist in modern cloud-native architectures, MLOps practices, and GenAI engineering, with only 3 engineers having production LLM experience. The pilot phase revealed critical gaps in prompt engineering discipline, lack of systematic evaluation frameworks, and absence of guardrails for financial advice compliance, creating regulatory risk that must be addressed before scaling.

Target Architecture Design

The target architecture implements a cloud-first, event-driven design on AWS using a multi-account strategy with AWS Organizations: a management account, security/audit account, shared services account, and separate accounts for dev, staging, and production environments. The production account deploys across three availability zones in us-east-1 (primary) and us-west-2 (DR) regions, with AWS Control Tower providing guardrails and Service Control Policies enforcing compliance boundaries. Network architecture utilizes a hub-and-spoke topology with AWS Transit Gateway connecting VPCs, maintaining segregated subnets for web tier (public), application tier (private), data tier (isolated), and AI/ML workloads (private with VPC endpoints). Application Load Balancers in public subnets terminate TLS 1.3 with ACM certificates, routing traffic through AWS WAF with managed rule groups for OWASP Top 10 and rate limiting (10,000 requests per 5 minutes per IP). AWS Shield Advanced provides DDoS protection with 24/7 response team engagement. Security groups implement zero-trust principles with explicit allow rules, while NACLs provide subnet-level defense-in-depth. VPC Flow Logs stream to S3 for security analysis and compliance evidence.

The data plane architecture separates real-time and batch processing paths. Real-time market data ingestion uses Amazon MSK (Managed Streaming for Apache Kafka) with 6-broker clusters processing 50MB/s throughput, feeding into Amazon Kinesis Data Analytics for stream processing and anomaly detection. Application tier deploys containerized microservices on Amazon EKS (Kubernetes 1.28+) with Fargate for serverless compute, running advisor-facing APIs, portfolio analysis engines, and AI orchestration services. EKS clusters span three AZs with node groups using c6i.4xlarge instances (16 vCPU, 32GB RAM) for compute-intensive workloads and m6i.2xlarge for general services, auto-scaling from 15 to 150 nodes based on CPU/memory utilization and custom metrics. Amazon Aurora PostgreSQL Serverless v2 serves as the primary transactional database with 2 reader instances for read scaling, supporting 10,000 concurrent connections with query performance insights enabled. Amazon DynamoDB handles high-velocity writes for user sessions, real-time notifications, and event sourcing with on-demand capacity mode and global tables for multi-region replication. Amazon ElastiCache for Redis (cluster mode enabled) provides distributed caching with 5-node clusters, achieving sub-millisecond latency for frequently accessed portfolio data, market quotes, and LLM response caching with 85% hit rates.

GenAI components leverage Amazon Bedrock as the primary inference platform, utilizing Claude 3.5 Sonnet for financial analysis and GPT-4 Turbo via Bedrock's model access for conversational interfaces. Custom SageMaker endpoints host fine-tuned FinBERT models for domain-specific embeddings and proprietary portfolio optimization models, deployed behind SageMaker multi-model endpoints for cost efficiency. The RAG architecture implements a three-tier retrieval system: Amazon OpenSearch Service (3-node r6g.2xlarge cluster) stores 25TB of vectorized research documents with 1536-dimensional embeddings, Amazon Kendra provides intelligent enterprise search over structured documents with ML-powered relevance tuning, and Amazon

Neptune graph database models relationships between securities, sectors, and economic indicators for contextual retrieval. LangChain orchestration runs on AWS Lambda for stateless operations and ECS Fargate for long-running agent workflows, with AWS Step Functions coordinating multi-step analysis pipelines. Prompt templates and guardrails deploy through Amazon Bedrock Guardrails with content filtering policies, PII detection/redaction, and topic-based denial rules preventing unauthorized investment advice.

The control plane implements comprehensive observability with Amazon CloudWatch for metrics/logs/alarms, AWS X-Ray for distributed tracing across microservices and LLM calls, and Amazon Managed Grafana for unified dashboards. CloudWatch Logs Insights queries analyze 500GB daily log volume with automated anomaly detection using CloudWatch Anomaly Detection. AWS Systems Manager Parameter Store (encrypted with KMS) manages application configuration, while AWS Secrets Manager handles database credentials and API keys with automatic 30-day rotation. CI/CD pipelines use AWS CodePipeline orchestrating CodeBuild for containerization, CodeDeploy for EKS deployments, and Terraform Cloud for infrastructure provisioning. The deployment strategy implements blue/green deployments for EKS services using AWS App Mesh for traffic shifting, with automated rollback triggered by CloudWatch alarms monitoring error rates >1% or P99 latency >5 seconds. Feature flags via AWS AppConfig enable progressive rollouts and A/B testing with real-time configuration updates. Disaster recovery architecture achieves RTO of 4 hours and RPO of 15 minutes through continuous Aurora replication to us-west-2, S3 cross-region replication for data lake objects, and automated failover orchestration via Route 53 health checks and AWS Backup for point-in-time recovery. Monthly DR drills validate runbooks and recovery procedures, with automated testing in the DR region consuming <5% of production costs.

Data Strategy

Data governance implements a federated model using AWS Lake Formation as the central authorization layer, with data domains owned by business units (Wealth Management, Research, Compliance) and a centralized Data Platform team providing shared services. Data classification follows a four-tier taxonomy: Public, Internal, Confidential, and Restricted (PII/financial data), with automated tagging via AWS Macie scanning S3 buckets and identifying sensitive data patterns. Lake Formation tag-based access control (TBAC) enforces fine-grained permissions, allowing data scientists to access anonymized datasets while restricting PII to authorized advisors and compliance officers. AWS Glue Data Catalog serves as the unified metadata repository with 2,500+ registered tables, capturing schema, lineage, and data quality metrics. Data lineage tracking uses AWS Glue DataBrew and custom Apache Atlas integration, providing end-to-end visibility from source systems (Bloomberg, Aladdin, Salesforce) through transformation pipelines to consumption by AI models and dashboards.

Ingestion architecture supports three patterns: batch, streaming, and CDC. Batch ingestion uses AWS Glue ETL jobs (Python/Spark) running on G.2X workers, processing nightly feeds from portfolio management systems with incremental loads based on watermark timestamps. AWS Database Migration Service (DMS) implements CDC from Oracle and SQL Server sources, capturing transaction logs and streaming changes to Amazon MSK with sub-second latency. Real-time market data flows through AWS IoT Core (10,000 messages/second) from Bloomberg feeds, with AWS Lambda functions enriching and routing events to appropriate Kinesis streams. Data quality framework leverages AWS Glue DataBrew for profiling and Great Expectations for validation rules, enforcing 95%+ completeness, uniqueness constraints on client IDs, and referential integrity across datasets. Schema evolution follows a backward-compatible approach using AWS Glue Schema Registry with Avro serialization, versioning schemas and validating producer/consumer compatibility before deployment.

Storage architecture implements a medallion pattern with bronze (raw), silver (cleansed), and gold (curated) layers in Amazon S3. Bronze layer uses S3 Standard storage class with lifecycle policies transitioning objects to S3 Intelligent-Tiering after 30 days, storing raw ingestion data partitioned by date (s3://data-lake-prod/bronze/market-data/year=2025/month=10/day=08/). Silver layer applies data quality rules, deduplication, and standardization, storing Parquet files with Snappy compression achieving 6:1 compression ratios. Gold layer contains business-ready datasets optimized for analytics and ML training, partitioned by use case and access patterns. Encryption uses S3-SSE with AWS KMS customer-managed keys (CMK), with separate keys per data classification tier and automatic key rotation enabled. Cross-region replication to us-west-2 provides disaster recovery for critical datasets (Restricted and Confidential tiers), while S3 Object Lock enforces WORM compliance for regulatory data with 7-year retention. Total storage footprint of 200TB in bronze, 120TB in silver, and 80TB in gold layers costs $4,200 monthly with Intelligent-Tiering optimization.

Retrieval and RAG implementation uses a hybrid approach combining dense vector search and sparse keyword matching. Amazon OpenSearch Service stores document embeddings generated by SageMaker endpoints running FinBERT models, with k-NN plugin enabling approximate nearest neighbor search across 15M document chunks (512 tokens each). Index strategy partitions by document type (research reports, regulatory filings, client portfolios) with separate indices optimized for query patterns, using 3 primary shards and 2 replicas per index. Embedding pipeline processes 50,000 documents daily via AWS Batch jobs, generating embeddings with dimension 768 and storing metadata (source, timestamp, classification) alongside vectors. Query-time retrieval implements a two-stage approach: initial k-NN search returns top 100 candidates, followed by cross-encoder reranking using SageMaker endpoints to select top 10 most relevant chunks. Amazon Kendra provides complementary semantic search over structured documents (PDFs, Word files) with custom synonyms for financial terminology and entity recognition for tickers, CUSIPs, and company names. Caching layer in ElastiCache stores frequently accessed embeddings and search results with 24-hour TTL, reducing OpenSearch query volume by 60% and improving P95 latency from 450ms to 80ms. Data products expose curated datasets through AWS Data Exchange and internal API Gateway endpoints, with usage metering via AWS CloudWatch and cost allocation tags enabling chargeback to consuming business units.

Model Evaluation Recommendations

Establish a comprehensive evaluation framework with offline and online testing stages, beginning with offline evaluation using curated golden datasets representing 2,000 real-world advisor scenarios across asset classes, client risk profiles, and market conditions. Golden sets include ground truth labels from senior advisor reviews, capturing expected portfolio recommendations, risk assessments, and compliance requirements. Implement automated regression testing via AWS CodePipeline, executing evaluation harness on every model update and blocking deployments if accuracy drops below 93% threshold or hallucination rate exceeds 2%. Evaluation metrics include ROUGE-L scores for report generation (target >0.85), precision/recall for investment recommendations (target >0.90/0.88), and semantic similarity scores using sentence transformers comparing generated advice against expert-validated responses. Store evaluation results in Amazon DynamoDB with versioning, enabling historical comparison and drift detection across model iterations.

Hallucination detection implements multi-layered validation: factual consistency checking via natural language inference models comparing generated statements against source documents, numerical accuracy verification ensuring portfolio calculations match deterministic algorithms within 0.01% tolerance, and citation validation confirming all claims reference retrievable source documents in the RAG corpus. Deploy Amazon Bedrock Guardrails with custom word filters blocking unauthorized investment products, regulatory-prohibited language, and competitor mentions. Implement prompt injection detection using adversarial testing frameworks, running 500+ attack scenarios monthly including jailbreak attempts, context manipulation, and instruction override patterns. Safety guardrails enforce output validation rules: recommendations must include risk disclosures, avoid guarantees of returns, and flag conflicts of interest. Response policies implement content filtering for PII leakage, using AWS Comprehend to detect and redact SSNs, account numbers, and personal identifiers before returning responses to advisors.

Human-in-the-loop workflows trigger mandatory review for high-stakes decisions: portfolio recommendations exceeding $1M require senior advisor approval captured via workflow in AWS Step Functions, risk scores above 7/10 enter escalation queues monitored in real-time dashboards, and model confidence below 80% routes to expert review with feedback captured for continuous learning. Implement A/B testing infrastructure using AWS AppConfig feature flags, randomly assigning 10% of advisors to challenger models while maintaining 90% on champion models, with statistical significance testing ($p<0.05$) required before promoting challengers. Acceptance criteria define deployment gates: user acceptance testing with 50 advisors achieving >85% satisfaction scores, load testing demonstrating <2 second P95 latency under 500 concurrent users, and security penetration testing showing zero critical vulnerabilities. Define SLIs tracking model quality: recommendation acceptance rate (target >75%), advisor override frequency (target <15%), client complaint rate (target <0.5%), and compliance violation rate (target 0%). SLOs specify 99.5% of recommendations meet quality thresholds with monthly error budgets allowing 3.6 hours of degraded performance.

Cost optimization balances performance and efficiency through intelligent model routing: simple queries (<100 tokens) route to Claude 3 Haiku achieving $0.25 per 1K tokens, complex analysis uses Claude 3.5 Sonnet at $3 per 1K tokens, and batch processing leverages self-hosted Llama 3 70B on SageMaker reducing costs by 60% for non-latency-sensitive workloads. Implement aggressive caching strategies storing LLM responses in ElastiCache with semantic similarity matching, achieving 40% cache hit rates and reducing inference costs by $80K monthly. Response streaming via Server-Sent Events improves perceived latency, displaying partial results within 500ms while full analysis completes in background. Failure isolation implements circuit breakers via AWS App Mesh, degrading gracefully to cached responses or simplified models when primary LLM endpoints experience >5% error rates or >10 second latency. Cost anomaly detection via AWS Cost Anomaly Detection alerts when daily inference costs exceed $8K threshold, triggering investigation of potential abuse or inefficient prompt patterns. Establish monthly model performance reviews with cross-functional stakeholders, analyzing evaluation metrics, user feedback, cost trends, and compliance incidents to inform continuous improvement roadmap.

Implementation Plan

Phase 1 (Months 1-2) establishes foundational infrastructure and baseline capabilities. Week 1-2 focuses on AWS account structure provisioning via AWS Control Tower, creating organizational units for production, non-production, and security accounts with Service Control Policies enforcing encryption, region restrictions, and required tagging. Configure AWS Transit Gateway hub-and-spoke network topology with VPC peering to on-premises data centers via Direct Connect, implementing network segmentation and security group baseline. Week 3-4 deploys infrastructure-as-code framework using Terraform Cloud with remote state in S3, establishing modules for VPC, EKS, RDS Aurora, and S3 data lake components. Implement CI/CD pipelines in AWS CodePipeline with automated testing gates and approval workflows. Week 5-6 establishes observability stack deploying CloudWatch dashboards, X-Ray tracing, and Grafana visualization, configuring log aggregation from on-premises systems and initial AWS services. Deploy AWS Config rules for compliance monitoring and AWS Security Hub for centralized security findings. Week 7-8 focuses on data platform foundation, provisioning AWS Glue Data Catalog, Lake Formation permissions model, and initial S3 bucket structure with encryption and lifecycle policies. Migrate 10TB of historical market data and client portfolios to bronze layer, validating data quality and establishing baseline ingestion pipelines. Deliverables include operational AWS environment, IaC repository, observability dashboards, and initial data lake with 10TB migrated data.

Phase 2 (Months 3-5) implements core application services and AI capabilities. Month 3 develops microservices architecture on Amazon EKS, deploying advisor API gateway, portfolio analysis service, and document processing service as containerized applications with Helm charts. Implement service mesh using AWS App Mesh for traffic management and observability. Integrate with existing Salesforce CRM and BlackRock Aladdin via API Gateway and AWS Lambda functions, establishing data synchronization patterns with 15-minute refresh cycles. Month 4 deploys GenAI components including Amazon Bedrock integration with Claude 3.5 Sonnet, implementing prompt templates and guardrails for financial advice generation. Deploy SageMaker endpoints for custom FinBERT embeddings and portfolio optimization models, establishing MLOps pipelines for model versioning and deployment. Configure Amazon OpenSearch Service for vector storage, ingesting 5M document chunks from research database with automated embedding generation. Month 5 implements RAG retrieval pipeline with hybrid search combining vector similarity and keyword matching, achieving <500ms P95 retrieval latency. Deploy LangChain orchestration on ECS Fargate for multi-step analysis workflows, integrating market data feeds from Bloomberg via MSK streaming. Implement caching layer in ElastiCache with intelligent cache warming for frequently accessed portfolios. Deliverables include functional advisor API processing 10,000 daily requests, operational RAG system with 5M documents, and integrated data pipelines from core systems.

Phase 3 (Months 6-7) conducts comprehensive testing and validation. Week 1-2 executes load testing using AWS Distributed Load Testing solution, simulating 500 concurrent advisors with

realistic usage patterns, validating auto-scaling policies trigger at 70% CPU utilization and scale from 15 to 45 EKS nodes within 3 minutes. Conduct chaos engineering experiments using AWS Fault Injection Simulator, validating graceful degradation when Aurora primary fails over to replica (RTO <60 seconds) and application resilience during AZ outages. Week 3-4 performs security testing including penetration testing by third-party firm, vulnerability scanning via Amazon Inspector, and compliance validation against SEC and FINRA requirements. Remediate identified findings with target of zero high-severity vulnerabilities before production deployment. Week 5-6 conducts user acceptance testing with 50 pilot advisors, collecting feedback via surveys and usage analytics, iterating on UI/UX based on advisor workflows. Validate model quality through side-by-side comparison of AI recommendations versus senior advisor analysis, achieving >90% agreement rates. Week 7-8 executes performance tuning optimizing database queries (reducing P95 from 450ms to 180ms), implementing query result caching, and right-sizing compute resources based on observed utilization patterns. Conduct disaster recovery drill failing over to us-west-2 region, validating RTO of 4 hours and RPO of 15 minutes meet requirements. Deliverables include load test reports demonstrating 500 concurrent user capacity, security assessment with remediation completion, UAT sign-off from 50 advisors, and validated DR procedures.

Phase 4 (Month 8) executes production deployment using blue/green strategy. Week 1 deploys green environment in production account, running parallel to existing systems without client traffic, conducting smoke tests and synthetic monitoring validation. Week 2 initiates traffic shifting routing 10% of advisor requests to green environment, monitoring error rates, latency metrics, and user feedback for 48 hours before proceeding. Week 3 progressively increases traffic to 25%, 50%, 75% over 3-day intervals with automated rollback triggers if error rates exceed 1% or P95 latency exceeds 3 seconds. Week 4 completes cutover to 100% green environment, maintaining blue environment in standby for 7 days before decommissioning. Deploy runbooks in AWS Systems Manager Documents covering incident response procedures, rollback steps, and escalation paths. Conduct tabletop exercises with operations team validating incident response for common failure scenarios. Deliverables include production deployment serving 500 advisors, operational runbooks, and validated rollback procedures.

Phase 5 (Months 9-10) focuses on stabilization and optimization. Month 9 establishes SRE practices including error budget policies (99.5% availability allows 3.6 hours monthly downtime), on-call rotation with PagerDuty integration, and blameless postmortem process for incidents. Optimize costs through reserved instance purchases for baseline EKS capacity (40% cost reduction), S3 Intelligent-Tiering migration (25% storage cost reduction), and LLM response caching improvements (45% inference cost reduction). Month 10 conducts knowledge transfer sessions training 15 operations engineers on architecture, troubleshooting procedures, and maintenance tasks. Create comprehensive documentation including architecture decision records, API specifications, and operational procedures in Confluence. Establish continuous improvement process with monthly reviews of KPIs, user feedback, and optimization opportunities. Deliverables include operational handover to support team, optimized cost structure achieving $180K monthly run rate, and complete documentation repository. Resourcing

requires 2 cloud architects, 4 backend engineers, 2 ML engineers, 2 data engineers, 1 security engineer, 1 DevOps engineer, and 1 project manager. RACI matrix assigns CIO as accountable executive, CTO as responsible delivery owner, and cross-functional stakeholders as consulted/informed parties. Key risks include data migration delays (mitigation: parallel run strategy), model quality issues (mitigation: extensive testing with golden datasets), and user adoption challenges (mitigation: change management program with training and support).

Integration & Operations

Integration architecture implements API-first design with OpenAPI 3.0 specifications defining contracts for 25 core endpoints including portfolio analysis, client recommendations, document processing, and market data retrieval. API Gateway serves as the integration hub, enforcing authentication via OAuth 2.0 with JWT tokens issued by Okta identity provider, implementing rate limiting at 1,000 requests per minute per advisor with burst capacity of 2,000 requests. Throttling policies protect backend services from overload, returning HTTP 429 with Retry-After headers when limits exceeded. Integration with Salesforce CRM uses bidirectional synchronization via AWS AppFlow, scheduling hourly incremental syncs of client profile updates and advisor activity logs, with conflict resolution favoring most recent timestamp. BlackRock Aladdin integration implements REST API calls with mutual TLS authentication, retrieving portfolio positions and performance data with 15-minute refresh cycles during market hours and hourly updates after close. Bloomberg Terminal data feeds connect via FIX protocol over dedicated network circuits, streaming real-time quotes and news to MSK topics with exactly-once delivery semantics. Data exchange formats standardize on JSON for API payloads with JSON Schema validation, Avro for Kafka messages with schema registry enforcement, and Parquet for bulk data transfers with Snappy compression.

Observability stack provides comprehensive visibility across distributed systems. CloudWatch collects 500+ custom metrics including LLM inference latency, cache hit rates, API response times, and business KPIs like recommendations generated per hour. Metrics stream to CloudWatch Logs Insights for ad-hoc analysis and Amazon Managed Service for Prometheus for long-term retention and advanced querying. CloudWatch Alarms monitor critical thresholds: P95 latency >3 seconds triggers PagerDuty alerts to on-call engineers, error rate >1% initiates automated runbook execution via Systems Manager, and Aurora CPU >80% scales read replicas automatically. AWS X-Ray provides distributed tracing with 10% sampling rate during normal operations and 100% sampling during incidents, capturing end-to-end request flows across API Gateway, Lambda, EKS services, and external API calls. Trace analysis identifies bottlenecks like slow database queries (>500ms) and inefficient LLM prompts (>5 seconds). Amazon Managed Grafana dashboards visualize golden signals (latency, traffic, errors, saturation) with separate views for executives (business KPIs), engineers (technical metrics), and operations (infrastructure health). Log aggregation centralizes application logs, access logs, and audit logs in CloudWatch Logs with 90-day retention in hot storage and 7-year retention in S3 Glacier for compliance. Structured logging using JSON format enables automated parsing and correlation, with trace IDs linking logs across service boundaries.

SLOs define reliability targets with error budgets enabling data-driven risk decisions. Availability SLO of 99.5% allows 3.6 hours monthly downtime, tracked via CloudWatch Synthetics running canary tests every 5 minutes from multiple regions. Latency SLO specifies P95 <2 seconds for API requests, measured via X-Ray service maps and CloudWatch metrics. Quality SLO targets >95% recommendation accuracy validated through weekly sampling of 200 advisor interactions

with expert review. Error budget policy halts new feature deployments when monthly budget exhausted, redirecting engineering capacity to reliability improvements. Incident response follows tiered severity model: SEV1 (production down) requires <15 minute response with executive notification, SEV2 (degraded performance) requires <1 hour response, SEV3 (minor issues) handled during business hours. Runbooks in Systems Manager Documents provide step-by-step remediation procedures for 30 common scenarios including Aurora failover, EKS node failures, and LLM endpoint timeouts. Post-incident reviews within 48 hours capture timeline, root cause, and action items tracked in Jira with executive summary distributed to stakeholders.

Operational playbooks cover change management, patching, and capacity planning. Change management follows CAB approval process for production changes with 3-business-day lead time for standard changes and emergency change procedures for critical security patches. All changes deploy via CI/CD pipelines with automated testing gates, blue/green deployment strategy, and 24-hour bake time before decommissioning old environment. Patching strategy applies security updates to EKS nodes monthly using managed node groups with rolling updates, Aurora minor version upgrades during maintenance windows (Sunday 2-4 AM ET), and Lambda runtime updates automatically via AWS managed policies. Capacity planning reviews quarterly analyze growth trends in user count, query volume, and data storage, projecting 12-month capacity needs and procuring reserved instances for cost optimization. Auto-scaling policies handle short-term demand spikes: EKS cluster scales based on CPU/memory utilization and custom metrics like queue depth, Aurora read replicas scale based on connection count and replication lag, and Lambda concurrency limits set at 1,000 with reserved concurrency for critical functions.

Performance management implements continuous optimization based on observed load profiles. Load testing executes monthly using realistic traffic patterns: 200 concurrent users baseline, 500 users peak, with burst scenarios simulating market volatility events (1,000 concurrent users for 15 minutes). Performance regression testing compares P50/P95/P99 latency across releases, blocking deployments with >10% latency degradation. Cost management establishes budgets of $180K monthly with AWS Budgets alerts at 80% and 100% thresholds, triggering review of anomalous spending. Cost allocation tags enable chargeback to business units based on usage, with monthly reports showing cost per advisor and cost per recommendation. FinOps practices include rightsizing recommendations from AWS Compute Optimizer, unused resource identification via AWS Trusted Advisor, and commitment-based discounts through Savings Plans achieving 35% cost reduction. Day-2 operations include quarterly DR drills validating failover to us-west-2 region with documented RTO/RPO measurements, monthly backup restore tests verifying Aurora snapshots and S3 object recovery, and continuous compliance evidence collection via AWS Audit Manager automating control assessments for SOC 2 and SEC requirements. Security operations integrate with SIEM platform ingesting CloudTrail logs, GuardDuty findings, and Security Hub alerts, with automated response playbooks remediating common threats like compromised credentials and unauthorized API calls.