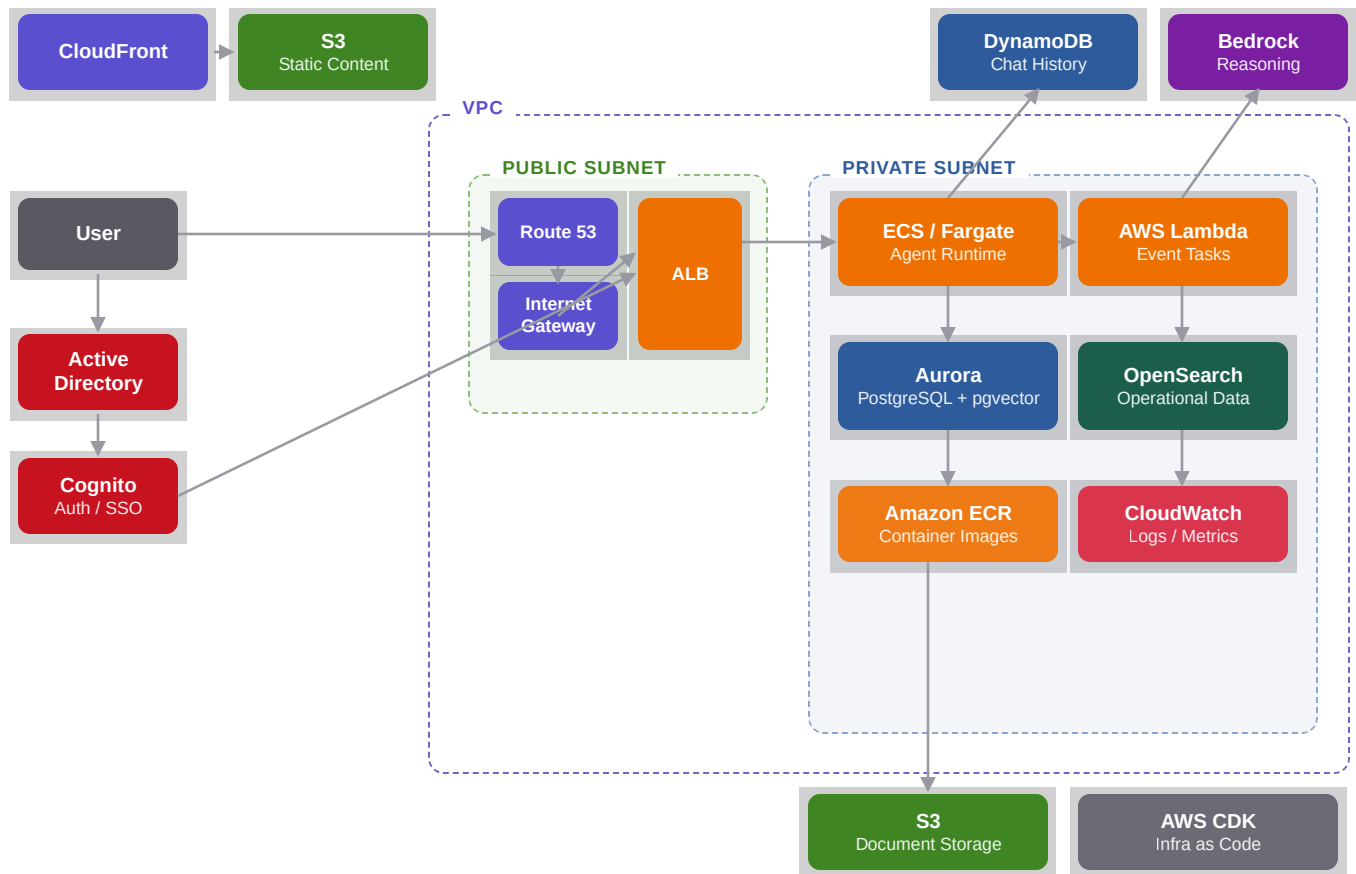


The Agentic AI Accelerator builds you a system that runs entirely inside your AWS environment, built around a central **Planner agent** that orchestrates a team of specialist agents — Operational Data, External APIs, Document Knowledge, Visualization, and Continuous Memory. Because it's custom-built rather than licensed, you own the resulting IP outright. The diagram below shows how each AWS service supports that architecture, provisioned consistently via AWS CDK.



User traffic and internal agent-to-service calls; edge and identity services shown outside the VPC boundary.

Amazon CloudFront

The React-based frontend is delivered globally through Amazon CloudFront, providing secure, low-latency access and a responsive user experience across regions and devices.

Amazon Cognito & SSO Integration

User authentication and identity management are powered by Amazon Cognito, integrating seamlessly with corporate Single Sign-On (SSO) to provide secure, role-based access control across all system components.

ECS / Fargate & AWS Lambda

The Planner agent and every specialist agent run as elastic, containerized workloads on Amazon ECS with Fargate, with AWS Lambda handling event-driven and short-lived orchestration tasks alongside them. Together they enable effortless scaling, automated recovery, and consistent performance across diverse data and user volumes.

Amazon Aurora (PostgreSQL with pgvector)

Amazon Aurora, running PostgreSQL with the pgvector extension, is the knowledge base behind the Document Knowledge agent — storing embeddings for hybrid vector and keyword search with reranking. This enables precise, cited context retrieval across unstructured document sources, a core capability behind the system's reasoning accuracy.

Amazon OpenSearch / Elasticsearch

OpenSearch (Elasticsearch-compatible) indexes operational data — events, telemetry, and logs — giving the Operational Data agent fast, structured search and aggregation across high-volume operational records, kept separate from the document knowledge base for performance and access control.

AWS Bedrock

Amazon Bedrock serves as the system's core foundation for language understanding and reasoning, powering the conversational interface and the Planner agent's orchestration of every specialist. It delivers accurate, context-aware responses through managed access to leading LLMs such as Claude, while ensuring enterprise-grade privacy, governance, and data security.

Amazon DynamoDB

Amazon DynamoDB stores conversation histories and session metadata — the substrate the Continuous Memory agent draws on — providing high-performance, auto-scaling persistence for user interactions and consistent context management across multi-agent workflows.

Amazon CloudWatch

Amazon CloudWatch provides centralized monitoring and observability, capturing application metrics, logs, and agent performance data to ensure reliable operation, proactive alerting, and continuous optimization across all system services.

Well-Architected Pillars

The AWS Well-Architected Framework helps you understand the pros and cons of the decisions you make when building systems in the cloud. The six pillars of the Framework allow you to learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems. The architecture on the earlier page reflects Well-Architected best practices.

Operational Excellence

Operational Excellence in this solution is achieved through automation, monitoring, and continuous improvement. The full stack — networking, ECS/Fargate services, Lambda functions, and data stores — is defined and deployed via AWS CDK, so every environment is provisioned consistently and every change is code-reviewed before it reaches production.

Amazon CloudWatch monitors application performance and logging, providing real-time insights and alerts to quickly resolve issues, while DynamoDB tracks session history for detailed analysis and continuous optimization of agent behavior over time.

Security

Security is a critical component of this solution, with multiple layers of protection built into the architecture.

Amazon Cognito integrates with Microsoft Active Directory using OAuth2 and OIDC, ensuring secure authentication and access control through corporate credentials.

Amazon CloudFront provides additional security by protecting the React client from DDoS attacks, while AWS WAF safeguards the application from web threats.

Data privacy is maintained with Amazon Bedrock, ensuring that sensitive information is processed securely within the AWS environment and never leaves it.

Encryption is enforced for data at rest and in transit, with detailed audit trails and access controls managed through AWS Identity and Access Management (IAM) and DynamoDB for session history storage.

Reliability

Reliability in this solution is ensured through the use of highly available and resilient AWS services.

ECS/Fargate provides fault-tolerant, scalable infrastructure for running the system's agents, with automatic scaling and health checks ensuring uninterrupted operation; Lambda adds resilience for event-driven tasks without managing servers.

Amazon Aurora, running PostgreSQL with pgvector, offers automated backups, failover support, and multi-AZ deployments, ensuring high availability for the document knowledge base. OpenSearch is similarly deployed across multiple nodes for resilient operational-data search.

Amazon CloudWatch logs provide real-time visibility into system performance and health, enabling proactive monitoring and fast issue resolution.

Performance Efficiency

Leveraging AWS's managed services, the system prioritizes optimal resource allocation, scalability, and monitoring to adapt to evolving workload demands effectively.

- **Auto Scaling:** ECS/Fargate services are configured with auto-scaling policies that dynamically adjust running tasks based on load, provisioning resources efficiently and only when needed.
- **Dynamic Load Distribution:** Application Load Balancers distribute incoming traffic across tasks, ensuring no single instance is overwhelmed.
- **On-Demand and Provisioned Capacity:** Based on workload patterns, DynamoDB can be configured for provisioned capacity (predictable workloads) or on-demand mode (variable workloads), ensuring efficient performance without over-provisioning.

Cost Optimization

Cost optimization in this solution is achieved by leveraging AWS services that provide flexible scaling and efficient resource usage.

Amazon ECS with Fargate runs the system's containerized agents, scaling automatically based on demand so the system pays only for the compute resources it uses, without managing underlying infrastructure. Lambda extends this same pay-per-use model to event-driven tasks.

Documents and application data are stored in Amazon S3, which offers elastic storage, allowing the system to scale storage capacity as needed while keeping costs low by charging only for storage and retrieval used. This combination of serverless and scalable services helps minimize overhead and optimize overall costs.

Sustainability

Our approach to sustainability involves optimizing resource usage, minimizing idle capacity, and utilizing AWS's energy-efficient managed services and data centers.

- **Right-Sizing and Task Auto Scaling:** ECS/Fargate tasks are right-sized to workload requirements and scale in and out based on demand, reducing idle capacity and conserving energy by minimizing the number of running containers.
- **Image Lifecycle Policies:** ECR lifecycle policies automatically remove unused or outdated container images, minimizing storage consumption and the energy required to store and manage them.
- **Log Retention Policies:** CloudWatch log retention policies keep logs only as long as needed for compliance and operational needs, reducing the energy used for long-term data storage.

Infrastructure as Code

Every environment — networking, ECS/Fargate services, Lambda functions, Aurora, OpenSearch, and DynamoDB — is defined in AWS CDK, so new environments and per-vertical domain configurations deploy consistently, with every change reviewed as code before it reaches production.