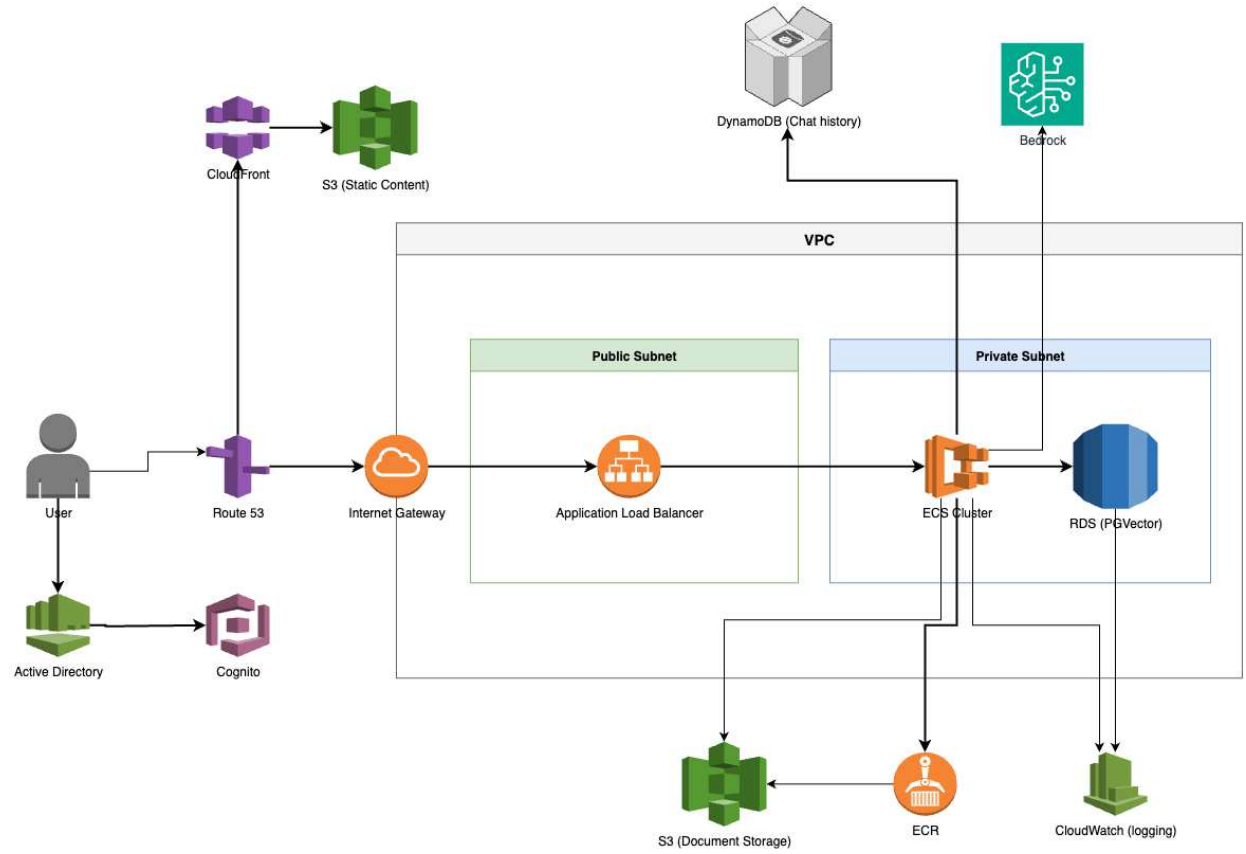

Amazon CloudFront

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 platform components.

Amazon ECS (Elastic Container Service) / EKS

The platform's AI orchestration and reasoning services run as elastic, containerized workloads on AWS ECS or EKS, enabling effortless scaling, automated recovery, and consistent performance across diverse data and user volumes.

Amazon RDS (PostgreSQL with PGVector extension)

Amazon RDS (PostgreSQL with PGVector) stores document and fact embeddings for high-performance vector similarity search. This enables precise context retrieval across

unstructured and structured data sources — a core capability driving the platform's Retrieval-Augmented Generation (RAG) and reasoning accuracy.

AWS Bedrock

Amazon Bedrock serves as the platform's core foundation for language understanding and reasoning, powering the conversational interface and multi-agent orchestration. 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, providing high-performance, auto-scaling persistence for user interactions and ensuring 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 platform 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 diagram above is an example of a Solution created with Well-Architected best practices in mind.

Operational Excellence

Operational Excellence in this solution is achieved through automation, monitoring, and continuous improvement.

By leveraging AWS services like Amazon ECS and RDS, the system ensures scalable management of backend services, automated backups, and failover handling.

Amazon CloudWatch is used for monitoring application performance and logging, providing real-time insights and alerts to quickly resolve issues.

The architecture also supports auditing via DynamoDB, which tracks session history, allowing for detailed analysis and continuous optimization of user interactions and system processes, ensuring high performance and reliability 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.

Encryption is enforced for data at rest and in transit, with detailed audit trails and access controls managed through services like 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.

Amazon ECS provides fault-tolerant and scalable infrastructure for running containerized workloads, with automatic scaling and health checks to ensure uninterrupted operation.

Amazon RDS, supporting PGVector, offers automated backups, failover support, and multi-AZ deployments, ensuring high availability for the vector database.

Amazon CloudWatch logs provide real-time visibility into system performance and health, allowing for proactive monitoring and quick resolution of any issues, further enhancing the system's reliability.

Performance Efficiency

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

Auto Scaling: ECS is configured with auto-scaling policies to dynamically adjust the number of tasks based on load, ensuring resources are provisioned efficiently and only when needed.

Dynamic Load Distribution: ALBs distribute incoming traffic across multiple ECS tasks, improving performance by ensuring that no single instance is overwhelmed.

On-Demand and Provisioned Capacity Modes: Based on workload patterns, DynamoDB can be configured to use either provisioned capacity (for predictable workloads) or on-demand mode (for variable or unpredictable 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 Fargate is used to run containerized services, allowing the application to scale automatically based on demand, ensuring that we only pay for the compute resources we use, without the need to manage underlying infrastructure.

Additionally, 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 only charging for the storage and data 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 tasks are right-sized based on workload requirements to avoid over-provisioning. Auto-scaling policies allow tasks to scale in and out based on demand, reducing the idle capacity and conserving energy by minimizing the number of running containers.

Image Lifecycle Policies: ECR lifecycle policies automatically remove unused or outdated images, minimizing storage consumption. This practice reduces the amount of storage resources needed and lowers the energy required for storing and managing container images.

Log Retention Policies: CloudWatch log retention policies are set to retain logs only as long as needed for compliance and operational needs. By minimizing stored logs, we reduce the energy used for long-term data storage.