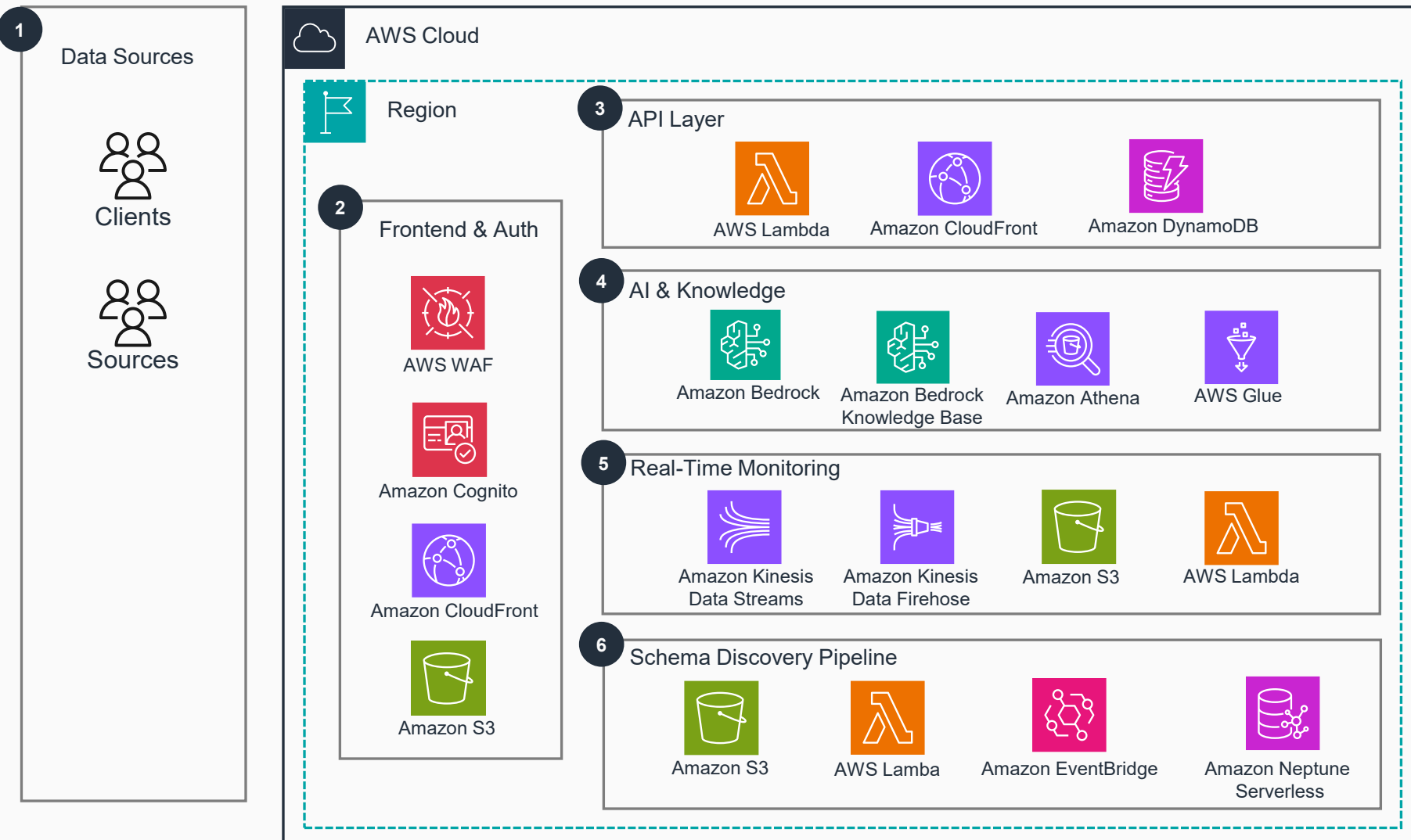


Guidance for Agentic Live Ops on AWS

High Level Overview

Natural language queries routed through a multi-agent system to structured data, knowledge bases, and live metrics.



- 1 Game operators use a React SPA. Game servers emit telemetry events into the monitoring pipeline.
- 2 **Amazon CloudFront** serves the app from **Amazon Simple Storage Service (Amazon S3)** with **AWS Web Application Firewall (WAF)** protection. **Amazon Cognito** authenticates users and provides SigV4 credentials. A regional **AWS WAF** ACL protects the **Amazon Cognito** endpoint.
- 3 Four **AWS Lambda** Function URLs handle Stream, Monitor, Conversations, and Config. Stream invokes **Amazon Bedrock AgentCore** in a private-isolated VPC. An orchestrator routes to eight sub-agents. **Amazon Bedrock Guardrails** filter prompt attacks. **Amazon Bedrock AgentCore Memory** tracks per-user context. **Amazon DynamoDB** stores all operational state.
- 4 Three **Amazon Bedrock Knowledge Bases** on **Amazon S3 Vectors** serve analytics summaries, remediation playbooks, and schema docs. **Amazon Athena** queries tables via **AWS Glue**.
- 5 **Amazon Kinesis Data Streams** ingests game events. A Pattern Detector Lambda evaluates threshold rules against sliding-window metrics in **Amazon DynamoDB**. Breaches trigger an Analysis Runner **AWS Lambda** for AI root cause analysis. **Amazon Data Firehose** archives events to **Amazon S3** for **Amazon Athena** queries.
- 6 **Amazon S3** holds schema definitions. **AWS Lambda** and **Amazon EventBridge** are triggered to summarize schema definitions which are hydrated into **Amazon Neptune Serverless** to assist with NL2SQL via **Amazon Athena**.



Reviewed for technical July 23, 2026

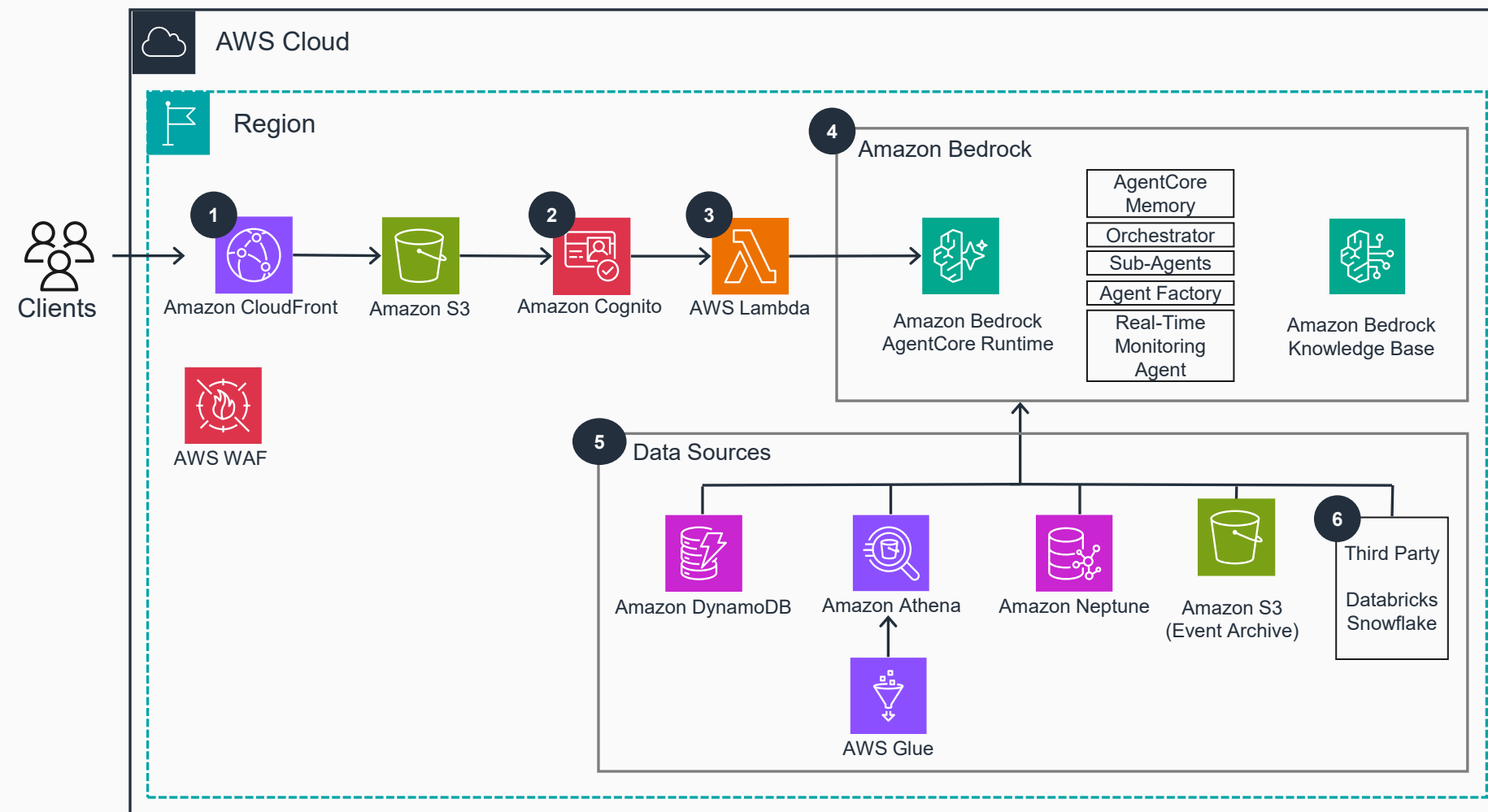
© 2026, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Reference Architecture

Guidance for Agentic Live Ops on AWS

Chat / AI Analytics

Natural language queries routed through a multi-agent system to structured data, knowledge bases, and live metrics.



- 1 Users access the React app via **Amazon CloudFront** with **AWS WAF** protection. Static assets served from **Amazon S3**.
- 2 **Amazon Cognito** authenticates users and provides IAM credentials. Requests signed with AWS Signature V4.
- 3 **AWS Lambda** Function URLs serve as APIs: Chat (SSE streaming), Monitoring, and Conversations.
- 4 **Amazon Bedrock AgentCore** Runtime hosts the orchestrator, routing queries to dynamically loaded sub-agents via Agent Factory. Sub-agents query three **Amazon Bedrock** Knowledge Bases (Analytics, Remediations, Schema) backed by **Amazon S3** vector storage.
- 5 **Amazon DynamoDB** stores live metrics, reports, pattern rules, conversations, and agent config cache. **Amazon Athena** runs SQL queries against structured data in **Amazon S3** via **AWS Glue Data Catalog** (15 tables). **Amazon Neptune** Serverless stores a schema graph of table/column relationships, queried by the orchestrator via Gremlin for join path discovery.
- 6 Third party data source integration is possible by providing schema definitions through the Schema Discovery Pipeline and new agents pertaining to each data source.



Reviewed for technical July 23, 2026

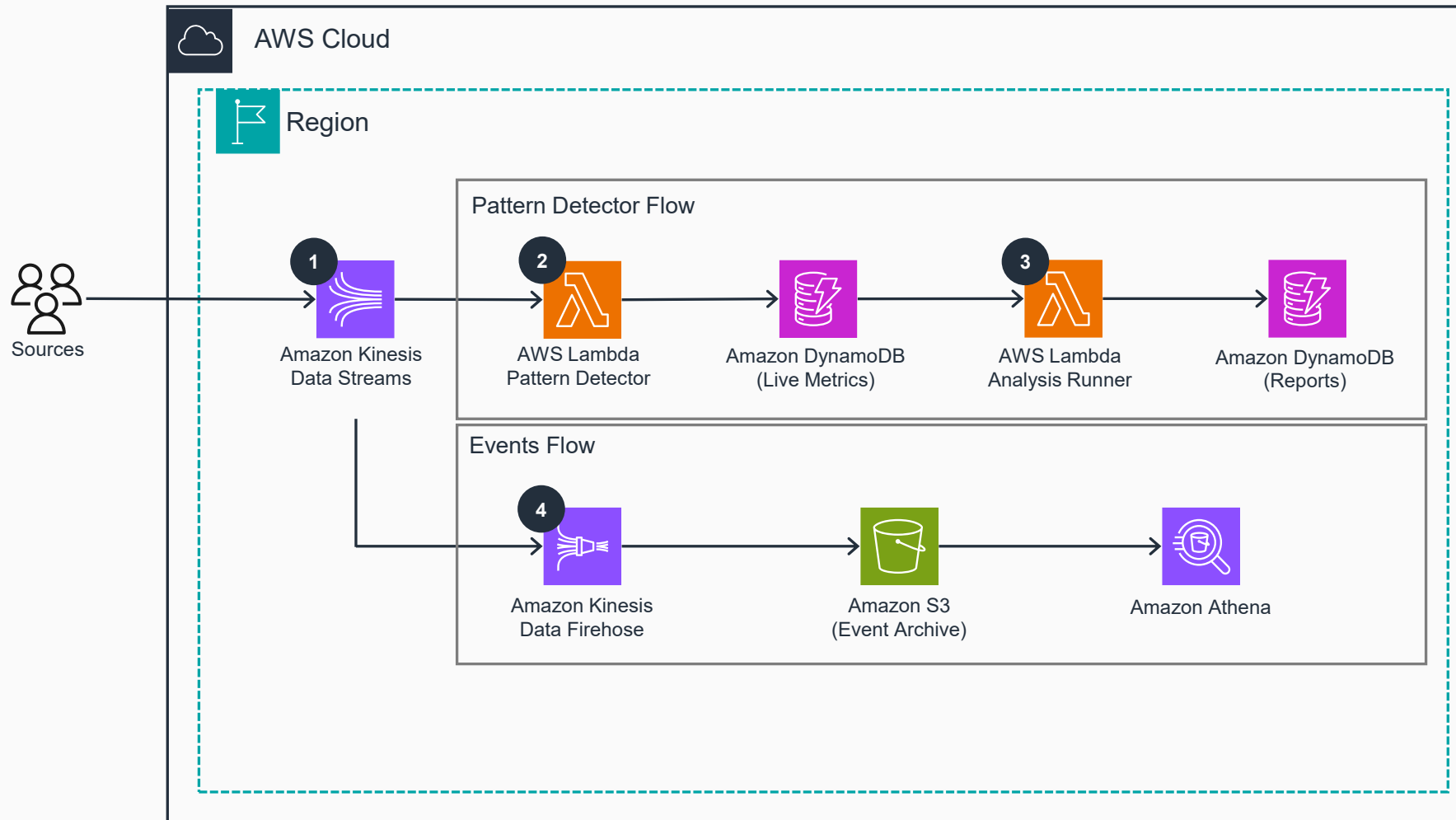
© 2026, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Reference Architecture

Guidance for Agentic Live Ops on AWS

Real-Time Monitoring Pipeline

Game events flow through Kinesis, are evaluated by pattern detection rules, trigger AI-powered analysis, and archive to S3 for historical queries.

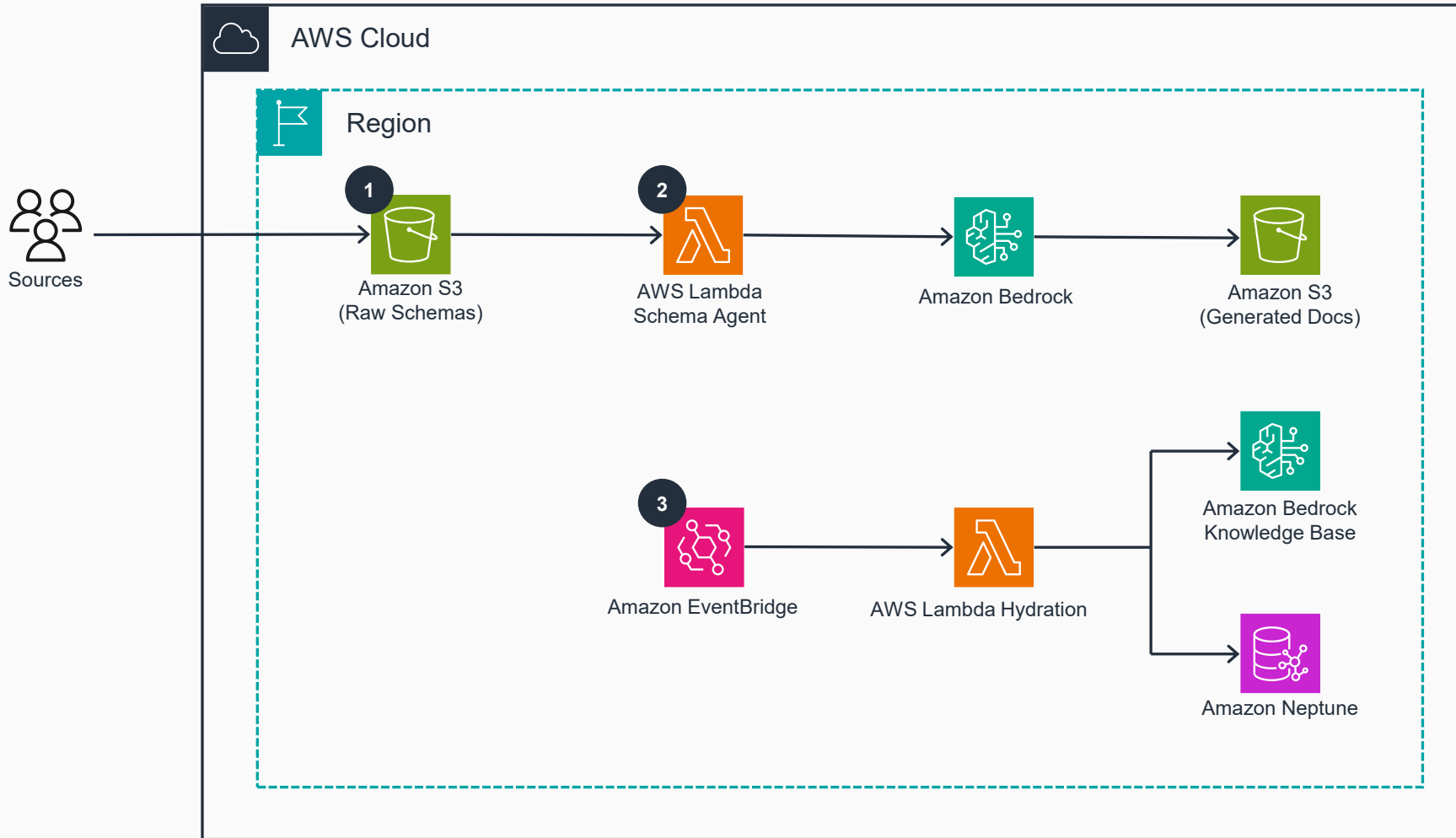


- 1** Game events flow into **Amazon Kinesis** Data Streams from game clients in real time.
- 2** A Pattern Detector **AWS Lambda** evaluates configurable threshold rules against sliding-window metrics in **Amazon DynamoDB**.
- 3** On threshold breach, an Analysis Runner **AWS Lambda** invokes **Amazon Bedrock** for AI-powered root cause analysis. Reports stored in **Amazon DynamoDB**.
- 4** **Amazon Kinesis** Data Firehose archives raw events to **Amazon S3** in 5-minute batches, queryable via **Amazon Athena** for historical analysis.

Guidance for Agentic Live Ops on AWS

Schema Discovery Pipeline

Schema uploads trigger AI-powered documentation generation, knowledge base indexing, and graph relationship syncing.



- 1 Operators upload raw schema files (SQL, JSON, text) to **Amazon S3**. An S3 event notification triggers processing.
- 2 A Schema Agent Lambda uses **Amazon Bedrock** Claude to parse schemas and generate standardized markdown documentation, saved to **Amazon S3**.
- 3 A daily **Amazon EventBridge** rule triggers a Hydration **AWS Lambda** that re-indexes the Schema **Bedrock Knowledge Base** and syncs table/column relationships to **Amazon Neptune**.