



Architecture Diagrams

Knowledge Graphs and GraphRAG with AWS and Neo4j



Knowledge Graphs and GraphRAG with AWS and Neo4j: Architecture Diagrams

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

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Knowledge Graphs and GraphRAG with AWS and Neo4j **i**

Knowledge Graphs and GraphRAG with AWS and Neo4j Diagram 1

Further reading 2

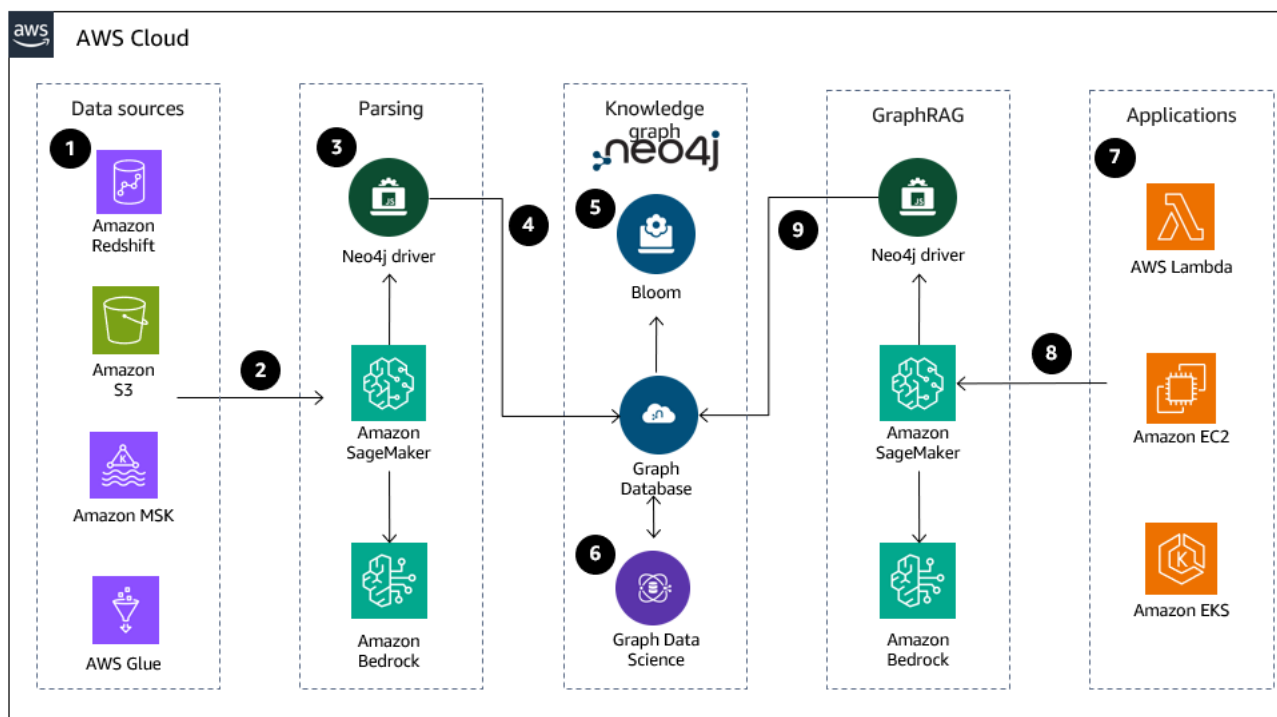
Diagram history 2

Knowledge Graphs and GraphRAG with AWS and Neo4j

Publication date: **November 26, 2024** ([Diagram history](#))

This reference architecture demonstrates how AWS services and Neo4j can be used to create knowledge graphs. Those graphs can then be used in a GraphRAG architecture.

Knowledge Graphs and GraphRAG with AWS and Neo4j Diagram



1. Structured, unstructured, and semi-structured data exists in a wide variety of systems, including **Amazon Redshift**, **Amazon Simple Storage Service (Amazon S3)**, and **Amazon Managed Streaming for Apache Kafka (Amazon MSK)**. Other systems can be accessed through an integration layer such as **AWS Glue**. Some subset of this data is highly connected. Insights can be uncovered by modeling it as a graph.
2. Scripts running in **Amazon SageMaker AI** pull connected data from these source systems.
3. Once in **SageMaker AI**, the data is passed to an **Amazon Bedrock** process managed by LangChain. Large language models (LLMs) extract entities and format the result as Cypher, the Neo4j query language. **SageMaker AI** passes that Cypher to the Neo4j driver.

4. Cypher statements wrapping the extracted data and entities are fed into the Neo4j graph database to create a knowledge graph. The Graph Database is one component of Neo4j AuraDB, a Neo4j SaaS running on AWS.
5. Users can explore the resulting knowledge graph in Neo4j Bloom, a BI tool designed specifically for graphs.
6. Enrichment can be done with Neo4j Graph Data Science. Entity resolution algorithms resolve duplicate entries. Community detection algorithms can be used to identify clusters and generate local cluster summaries for GraphRAG. Graph embedding algorithms help map and identify topologically similar entities for GraphRAG.
7. With a knowledge graph created, applications can then make use of that graph in a GraphRAG architecture to provide grounded results with fewer hallucinations than alternative approaches. Those applications can run in a variety of AWS platforms, including **AWS Lambda**, **Amazon Elastic Compute Cloud** (Amazon EC2), and **Amazon Elastic Kubernetes Service** (Amazon EKS).
8. Client applications can make calls into functions hosted in **SageMaker AI**. Those functions query **Amazon Bedrock**.
9. **Amazon Bedrock** responses are grounded using Neo4j Graph Database with data gathered from the wider enterprise. Using a graph for grounding provides a vector RAG approach with the additional ability to use the structure of the graph to improve responses.

Further reading

For additional information, refer to

- [AWS Architecture Icons](#)
- [AWS Architecture Center](#)
- [AWS Well-Architected](#)

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Reference architecture diagram first published.	November 26, 2024

Note

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.