Retrieval Augmented Generation options and architectures on AWS

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Retrieval Augmented Generation options and architectures on AWS

Table of Contents

Retrieval Augmented Generation options and architectures on AWS

Mithil Shah, Rajeev Muralidhar, and Natacha Fort, Amazon Web Services

October 2024 (document history)

Generative AI refers to a subset of AI models that can create new content and artifacts, such as images, videos, text, and audio, from a simple text prompt. Generative AI models are trained on vast amounts of data that encompasses a wide range of subjects and tasks. This enables them to demonstrate remarkable versatility in performing various tasks, even those for which they have not been explicitly trained. Due to a single model's ability to perform multiple tasks, these models are often referred to as *foundation models* (FMs).

One of the notable applications of generative AI models is their proficiency in answering questions. However, there are specific challenges that arise when these models are used to answer questions based on custom documents. Custom documents can include proprietary information, internal websites, internal documentation, Confluence pages, SharePoint pages, and others. One option is to use *Retrieval Augmented Generation (RAG)*. With RAG, the foundation model references an authoritative data source that is outside of its training data sources (such as your custom documents) before generating a response.

This guide describes the distinct generative AI options that are available for answering questions from custom documentation, including Retrieval Augmented Generation (RAG) systems. It also provides an overview of building RAG systems on Amazon Web Services (AWS). By reviewing the RAG options and architectures, you can choose between fully managed services on AWS and custom RAG architectures.

Intended audience

The intended audience for this guide is generative AI architects and managers who want to build a RAG solution, to review the available architectures, and to understand the benefits and drawbacks of each option.

Objectives

This guide helps you do the following:

- Understand the generative AI options available for answering questions from custom documents
- Review the architecture options for RAG systems on AWS
- Understand the advantages and disadvantages of each RAG option
- Choose a RAG architecture for your AWS environment

Generative AI options for querying custom documents

Organizations often have various sources of structured and unstructured data. This guide focuses on how you can use generative AI to answer questions from unstructured data.

Unstructured data in your organization can come from various sources. These might be PDFs, text files, internal wikis, technical documents, public facing websites, knowledge bases, or others. If you want a foundation model that can answer questions about unstructured data, the following options are available:

- Train a new foundation model by using your custom documents and other training data
- Fine-tune an existing foundation model by using data from your custom documents
- Use in-context learning to pass a document to the foundation model when you ask a question
- Use a Retrieval Augmented Generation (RAG) approach

Training a new foundation model from scratch that includes your custom data is an ambitious undertaking. A few companies have done it successfully, such as Bloomberg with their BloombergGPT model. Another example is the multimodal EXAONE model by LG AI Research, which was trained by using 600 billion pieces of artwork and 250 million high-resolution images, accompanied with text. According to The Cost of AI: Should You Build or Buy Your Foundation Model (LinkedIn), a model similar to Meta Llama 2 costs around USD $4.8 million to train. There are two primary prerequisites for training a model from scratch: access to resources (financial, technical, time) and a clear return on investment. If this does not seem the right fit, then the next option is to fine-tune an existing foundation model.

Fine-tuning an existing model involves taking a model, such as an Amazon Titan, Mistral, or Llama model, and then adapting the model to your custom data. There are various techniques for fine-tuning, most of which involve modifying only a few parameters instead of modifying all of the parameters in the model. This is called *parameter-efficient fine-tuning*. There are two primary methods for fine-tuning:

- *Supervised fine-tuning* uses labeled data and helps you train the model for a new kind of task. For example, if you wanted to generate a report based on a PDF form, then you might have to teach the model how to do that by providing enough examples.

- *Unsupervised fine-tuning* is task-agnostic and adapts the foundation model to your own data. It trains the model to understand the context of your documents. The fine-tuned model then creates content, such as a report, by using a style that is more custom your organization.

However, fine-tuning may not be ideal for question-answer use cases. For more information, see Comparing RAG and fine-tuning in this guide.

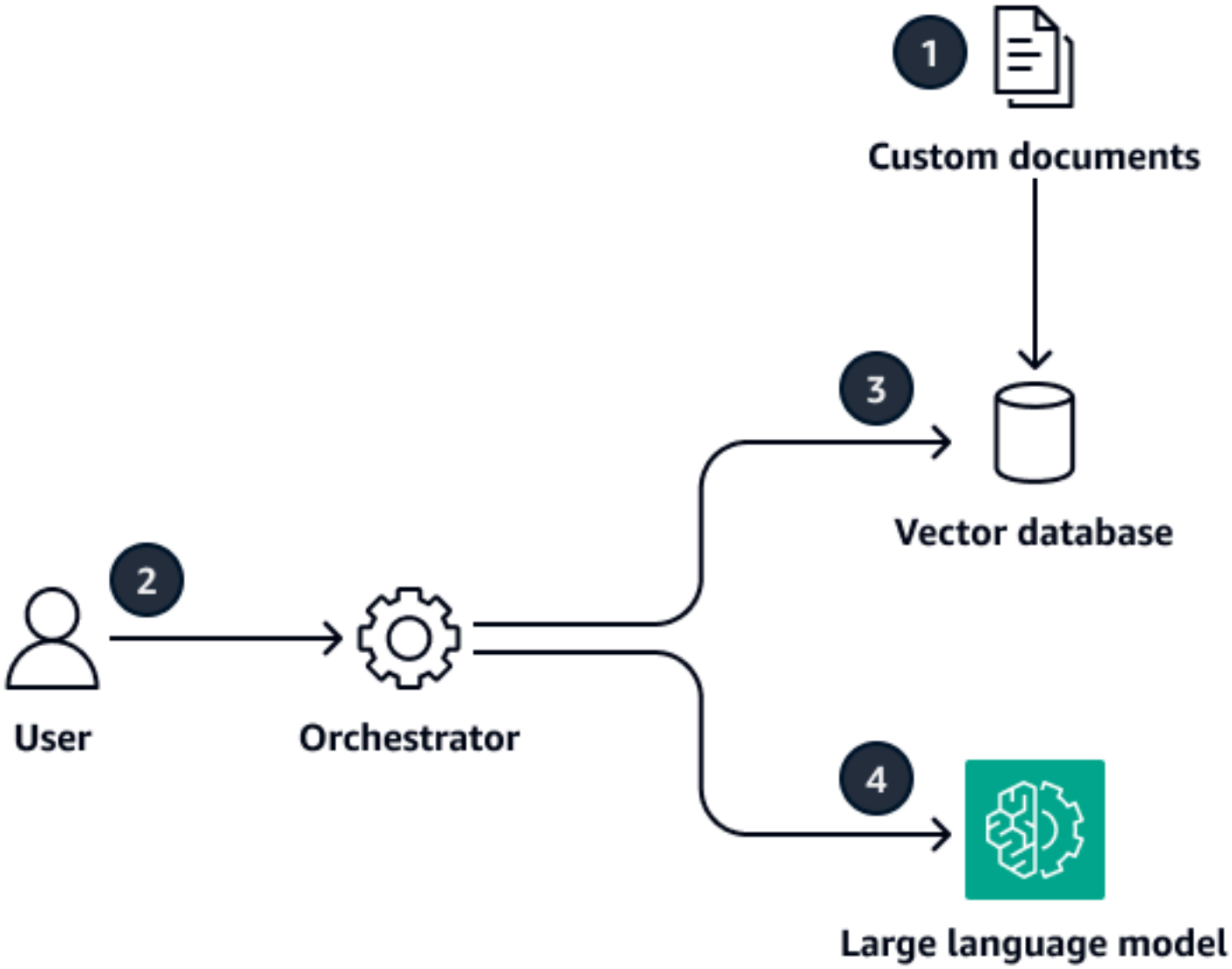
When you ask a question, you can pass a document the foundation model and use the model's in-context learning to return answers from the document. This option is suitable for ad-hoc querying of a single document. However, this solution doesn't work well for querying multiple documents or for querying systems and applications, such as Microsoft SharePoint or Atlassian Confluence.

The final option is to use RAG. With RAG, the foundation model references your custom documents before generating a response. RAG extends the model's capabilities to your organization's internal knowledge base, all without the need to retrain the model. It is a cost-effective approach to improving the model output so that it remains relevant, accurate, and useful in various contexts.

Topics in this section:

Understanding Retrieval Augmented Generation

Retrieval Augmented Generation (RAG) is a technique used to augment a large language model (LLM) with external data, such as a company's internal documents. This provides the model with the context it needs to produce accurate and useful output for your specific use case. RAG is a pragmatic and effective approach to using LLMs in an enterprise. The following diagram shows a high-level overview of how a RAG approach works.



Broadly speaking, the RAG process is four steps. The first step is done once, and the other three steps are done as many times as needed:

1. You create embeddings to ingest the internal documents into a vector database. *Embeddings* are numeric representations of text in the documents that capture the semantic or contextual meaning of the data. A *vector database* is essentially a database of these embeddings, and it is sometimes called a *vector store* or *vector index*. This step requires data cleaning, formatting, and chunking, but this is a one-time, upfront activity.
2. A human submits a query in natural language.
3. An orchestrator performs a similarity search in the vector database and retrieves the relevant data. The orchestrator adds the retrieved data (also known as *context*) to the prompt that contains the query.

4. The orchestrator sends the query and the context to the LLM. The LLM generates a response to the query by using the additional context.

From a user's perspective, RAG looks like interacting with any LLM. However, the system knows much more about the content in question and provides answers that are fine-tuned to the organization's knowledge base.

For more information about how a RAG approach works, see What is RAG on the AWS website.

Components of production-level RAG systems

Building a production-level RAG system requires thinking through several different aspects of the RAG workflow. Conceptually, a production-level RAG workflow requires the following capabilities and components, regardless of the specific implementation:

- **Connectors** — These connect different enterprise data sources with the vector database. Examples of structured data sources include transactional and analytical databases. Examples of unstructured data sources include object stores, code bases, and software as a service (SaaS) platforms. Each data source might require different connectivity patterns, licenses, and configurations.
- **Data processing** — Data comes in many shapes and forms, such as PDFs, scanned images, documents, presentations, and Microsoft SharePoint files. You must use data processing techniques to extract, process, and prepare the data for indexing.
- **Embeddings** — To perform a relevancy search, you must convert your documents and user queries into a compatible format. By using embedding language models, you convert the documents into numerical representation. These are essentially inputs for the underlying foundation model.
- **Vector database** — The vector database is an index of the embeddings, the associated text, and metadata. The index is optimized for search and retrieval.
- **Retriever** — For the user query, the retriever fetches the relevant context from the vector database and ranks the responses based on business requirements.
- **Foundation model** — The foundation model for a RAG system is typically an LLM. By processing the context and the prompt, the foundation model generates and formats a response for the user.
- **Guardrails** — Guardrails are designed to make sure that the query, prompt, retrieved context, and LLM response are accurate, responsible, ethical, and free of hallucinations and bias.

- **Orchestrator** — The orchestrator is responsible for scheduling and managing the end-to-end workflow.
- **User experience** — Typically, the user interacts with a conversational chat interface that has rich features, including displaying chat history and collecting user feedback about responses.
- **Identity and user management** — It is critical to control user access to the application at fine granularity. In the AWS Cloud, policies, roles, and permissions are typically managed through AWS Identity and Access Management (IAM).

Clearly, there is significant amount of work to plan, develop, release, and manage a RAG system. Fully managed services, such as Amazon Bedrock or Amazon Q Business, can help you manage some of the undifferentiated heavy lifting. However, custom RAG architectures can provide more control over the components, such as the retriever or the vector database.

Comparing Retrieval Augmented Generation and fine-tuning

The following table describes the advantages and disadvantages of the fine-tuning and RAG-based approaches.

Approach	Advantages	Disadvantages
Fine-tuning	• If a fine-tuned model is trained using the unsupervised approach, then it is able to create content that more closely matches your organization's style. • A fine-tuned model that is trained on proprietary or regulatory data can help your organization follow in-house or industry-specific data and compliance standards.	• Fine-tuning can take a few hours to days, depending on the size of the model. Therefore, it not be a good solution if your custom documents change frequently. • Fine-tuning requires an understanding of techniques, such as low-rank adaptation (LoRA) and parameter-efficient fine-tuning (PEFT). Fine-tuning might require a data scientist.

Approach	Advantages	Disadvantages
		• Fine-tuning might not be available for all models. • Fine-tuned models do not provide a reference to the source in their responses. • There can be an increased risk of hallucination when using a fine-tuned model to answer questions.
RAG	• RAG allows you to build a question-answering system for your custom documents without fine-tuning. • RAG can incorporate the latest documents in a few minutes. • AWS offers fully managed RAG solutions. Therefore , no data scientist or specialized knowledge of machine learning is required. • In its response, a RAG model provides a reference to the information source. • Because RAG uses the context from the vector search as the basis of its generated answer, there is a reduced risk of hallucina tion.	• RAG does not work well when summarizing information from entire documents.

If you need to build a question-answering solution that references your custom documents, then we recommend that you start from a RAG-based approach. Use fine-tuning if you need the model to perform additional tasks, such as summarization.

You can combine the fine-tuning and RAG approaches in a single model. In the case, the RAG architecture does not change, but the LLM that generates the answer is also fine-tuned with the custom documents. This combines the best of both worlds, and it might be an optimum solution for your use case. For more information about how to combine supervised fine-tuning with RAG, see the RAFT: Adapting Language Model to Domain Specific RAG research from the University of California, Berkeley.

Use cases for Retrieval Augmented Generation

The following are common use cases for using a RAG approach:

- **Search engines** – RAG-enabled search engines can provide more accurate and up-to-date featured snippets in their search results.
- **Question-answering systems** – RAG can improve the quality of responses in question-answering systems. The retrieval-based model uses similarity search to find relevant passages or documents that contain the answer. Then, it generates a concise and relevant response based on that information.
- **Retail or e-commerce** – RAG can enhance the user experience in e-commerce by providing more relevant and personalized product recommendations. By retrieving and incorporating information about user preferences and product details, RAG can generate more accurate and helpful recommendations for customers.
- **Industrial or manufacturing** – In manufacturing, RAG helps you quickly access critical information, such as factory plant operations. It can also help with decision-making processes, troubleshooting, and organizational innovation. For manufacturers who operate within stringent regulatory frameworks, RAG can swiftly retrieve updated regulations and compliance standards from internal and external sources, such as from industry standards or regulatory agencies.
- **Healthcare** – RAG has potential in the healthcare industry, where access to accurate and timely information is crucial. By retrieving and incorporating relevant medical knowledge from external sources, RAG can provide more accurate and context-aware responses in healthcare applications. Such applications augment the information accessible by a human clinician, who ultimately makes the call and not the model.

- **Legal** – RAG can be applied powerfully in legal scenarios, such as mergers and acquisitions, where complex legal documents provide context for queries. This can help legal professionals rapidly navigate complex regulatory issues.

Fully managed Retrieval Augmented Generation options on AWS

To manage Retrieval Augmented Generation (RAG) workflows on AWS, you can use custom RAG pipelines or use some of the fully managed services capabilities that AWS offers. Because they include many of the core components of a RAG-based system, fully managed services can help you manage some of the undifferentiated heavy lifting. However, these services provide less opportunity for customization.

The fully managed AWS services use connectors to ingest data from external data sources, such as websites, Atlassian Confluence, or Microsoft SharePoint. The supported data sources vary by AWS service.

This section explores the following fully managed options for building RAG workflows on AWS:

- Knowledge bases for Amazon Bedrock
- Amazon Q Business
- Amazon SageMaker Canvas

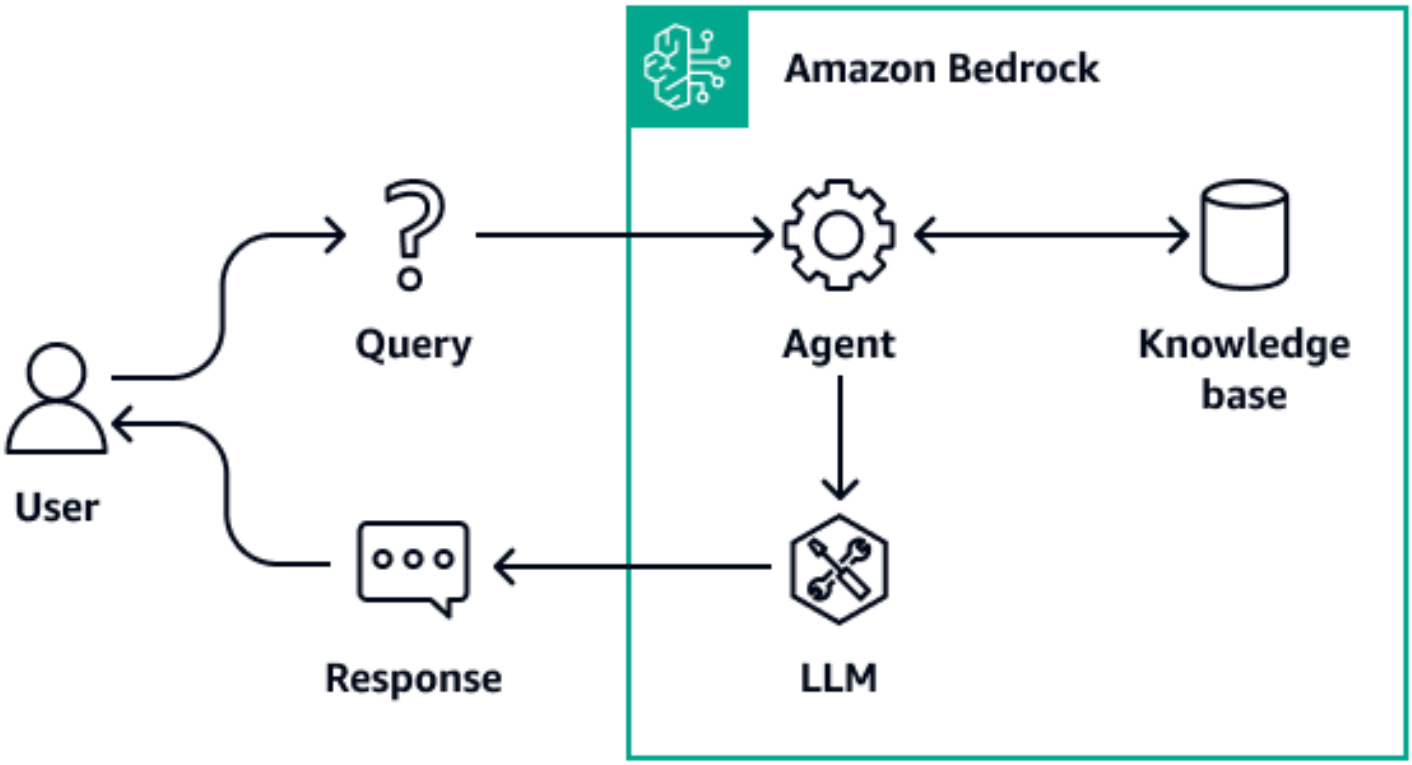
For more information about how to choose between these options, see Choosing a Retrieval Augmented Generation option on AWS in this guide.

Knowledge bases for Amazon Bedrock

Amazon Bedrock is a fully managed service that makes high-performing foundation models (FMs) from leading AI startups and Amazon available for your use through a unified API. Knowledge bases is an Amazon Bedrock capability that helps you implement the entire RAG workflow, from ingestion to retrieval and prompt augmentation. There is no need to build custom integrations to data sources or to manage data flows. Session context management is built in so that your generative AI application can readily support multi-turn conversations.

After you specify the location of your data, knowledge bases for Amazon Bedrock internally fetches the documents, chunks them into blocks of text, converts the text to embeddings, and then stores the embeddings in your choice of vector database. Amazon Bedrock manages and updates the embeddings, keeping the vector database in sync with the data. For more information about how knowledge bases work, see How Amazon Bedrock knowledge bases work.

If you add knowledge bases to an Amazon Bedrock agent, the agent identifies the appropriate knowledge base based on the user input. The agent retrieves the relevant information and adds the information to the input prompt. The updated prompt provides the model with more context information to generate a response. To improve transparency and minimize hallucinations, the information retrieved from the knowledge base is traceable to its source.



Amazon Bedrock supports the following two APIs for RAG:

- RetrieveAndGenerate – You can use this API to query your knowledge base and generate responses from the information it retrieves. Internally, Amazon Bedrock converts the queries into embeddings, queries the knowledge base, augments the prompt with the search results as context information, and returns the LLM-generated response. Amazon Bedrock also manages the short-term memory of the conversation to provide more contextual results.
- Retrieve – You can use this API to query your knowledge base with information retrieved directly from the knowledge base. You can use the information returned from this API to process the retrieved text, evaluate their relevance, or develop a separate workflow for response generation. Internally, Amazon Bedrock converts the queries into embeddings, searches the knowledge base, and returns the relevant results. You can build additional workflows on top of the search

results. For example, you can use the LangChain `AmazonKnowledgeBasesRetriever` plugin to integrate RAG workflows into generative AI applications.

For sample architectural patterns and step-by-step instructions for using the APIs, see Knowledge Bases now delivers fully managed RAG experience in Amazon Bedrock (AWS blog post). For more information about how to use the `RetrieveAndGenerate` API to build a RAG workflow for an intelligent chat-based application, see Build a contextual chatbot application using Amazon Bedrock Knowledge Bases (AWS blog post).

Data sources for knowledge bases

You can connect your proprietary data to a knowledge base. After you've configured a data source connector, you can sync or keep your data up to date with your knowledge base and make your data available for querying. Amazon Bedrock knowledge bases support connections to the following data sources:

- Amazon Simple Storage Service (Amazon S3) – You can connect an Amazon S3 bucket to an Amazon Bedrock knowledge base by using either the console or the API. The knowledge base ingests and indexes the files in the bucket. This type of data source supports the following features:
  - **Document metadata fields** – You can include a separate file to specify the metadata for the files in the Amazon S3 bucket. You can then use these metadata fields to filter and improve the relevancy of responses.
  - **Inclusion or exclusion filters** – You can include or exclude certain content when crawling.
  - **Incremental syncing** – The content changes are tracked, and only content that has changed since the last sync is crawled.
- Confluence – You can connect an Atlassian Confluence instance to an Amazon Bedrock knowledge base by using the console or the API. This type of data source supports the following features:
  - **Auto detection of main document fields** – The metadata fields are automatically detected and crawled. You can use these fields for filtering.
  - **Inclusion or exclusion content filters** – You can include or exclude certain content by using a prefix or a regular expression pattern on the space, page title, blog title, comment, attachment name, or extension.
  - **Incremental syncing** - The content changes are tracked, and only content that has changed since the last sync is crawled.

  - **OAuth 2.0 authentication, authentication with Confluence API token** – The authentication credentials are stored in AWS Secrets Manager.
- Microsoft SharePoint – You can connect a SharePoint instance to a knowledge base by using either the console or the API. This type of data source supports the following features:
  - **Auto detection of main document fields** – The metadata fields are automatically detected and crawled. You can use these fields for filtering.
  - **Inclusion or exclusion content filters** – You can include or exclude certain content by using a prefix or a regular expression pattern on the main page title, event name, and file name (including its extension).
  - **Incremental syncing** - The content changes are tracked, and only content that has changed since the last sync is crawled.
  - **OAuth 2.0 authentication** – The authentication credentials are stored in AWS Secrets Manager.
- Salesforce – You can connect a Salesforce instance to a knowledge base by using either the console or the API. This type of data source supports the following features:
  - **Auto detection of main document fields** – The metadata fields are automatically detected and crawled. You can use these fields for filtering.
  - **Inclusion or exclusion content filters** – You can include or exclude certain content by using a prefix or a regular expression pattern. For a list of content types that you can apply filters to, see *Inclusion/exclusion filters* in the Amazon Bedrock documentation.
  - **Incremental syncing** – The content changes are tracked, and only content that has changed since the last sync is crawled.
  - **OAuth 2.0 authentication** – The authentication credentials are stored in AWS Secrets Manager.
- Web Crawler – An Amazon Bedrock Web Crawler connects to and crawls the URLs that you provide. The following features are supported:
  - Select multiple URLs to crawl
  - Respect standard robots.txt directives, such as `Allow` and `Disallow`
  - Exclude URLs that match a pattern
  - Limit the rate of crawling
  - In Amazon CloudWatch, view the status of each URL crawled

For more information about the data sources that you can connect to your Amazon Bedrock knowledge base, see Create a data source connector for your knowledge base.

Vector databases for knowledge bases

When you set up a connection between the knowledge base and the data source, you must configure a vector database, also known as a *vector store*. A vector database is where Amazon Bedrock stores, updates, and manages the embeddings that represent your data. Each data source supports different types of vector database. To determine which vector database are available for your data source, see the data source types.

If you prefer for Amazon Bedrock to automatically create a vector database in Amazon OpenSearch Serverless for you, you can choose this option when you create the knowledge base. However, you can also choose to set up your own vector database. If you set up your own vector database, see Prerequisites for your own vector store for a knowledge base. Each type of vector database has its own prerequisites.

Depending on your data source type, Amazon Bedrock knowledge bases support the following vector databases:

- Amazon OpenSearch Serverless
- Amazon Aurora PostgreSQL-Compatible Edition
- Pinecone (Pinecone documentation)
- Redis Enterprise Cloud (Redis documentation)
- MongoDB Atlas (MongoDB documentation)

Amazon Q Business

Amazon Q Business is a fully managed, generative-AI powered assistant that you can configure to answer questions, provide summaries, generate content, and complete tasks based on your enterprise data. It allows end users to receive immediate, permissions-aware responses from enterprise data sources with citations.

Key features

The following capabilities of Amazon Q Business can help you build a production-grade RAG-based generative AI application:

- **Built-in connectors** – Amazon Q Business supports more than 40 types of connectors, such as connectors for Adobe Experience Manager (AEM), Salesforce, Jira, and Microsoft SharePoint. For a complete list, see Supported connectors. If you need a connector that is not supported, you can

use Amazon AppFlow to pull data from your data source into Amazon Simple Storage Service (Amazon S3) and then connect Amazon Q Business to the Amazon S3 bucket. For a complete list of data sources that Amazon AppFlow supports, see Supported applications.

- **Built-in indexing pipelines** – Amazon Q Business provides a built-in pipeline for indexing data in a vector database. You can use an AWS Lambda function to add preprocessing logic for your indexing pipeline.
- **Index options** – You can create and provision a native index in Amazon Q Business, and you use an Amazon Q Business retriever to pull data from that index. Alternatively, you can use a preconfigured Amazon Kendra index as a retriever. For more information, see Creating a retriever for an Amazon Q Business application.
- **Foundation models** – Amazon Q Business uses the foundation models that are supported in Amazon Bedrock. For a complete list, see Supported foundation models in Amazon Bedrock.
- **Plugins** – Amazon Q Business provides the capability to use plugins to integrate with target systems, such as an automated way to summarize ticket information and ticket creation in Jira. Once configured, plugins can support read and write actions that can help you boost end user productivity. Amazon Q Business supports two types of plugins: built-in plugins and custom plugins.
- **Guardrails** – Amazon Q Business supports global controls and topic-level controls. For example, these controls can detect personally identifiable information (PII), abuse, or sensitive information in prompts. For more information, see Admin controls and guardrails in Amazon Q Business.
- **Identity management** – With Amazon Q Business, you can manage users and their access to the RAG-based generative AI application. For more information, see Identity and access management for Amazon Q Business. Also, Amazon Q Business connectors index access control list (ACL) information that's attached to a document along with the document itself. Then, Amazon Q Business stores the ACL information it indexes in the Amazon Q Business User Store to create user and group mappings and filter chat responses based on the end user's access to documents. For more information, see Data source connector concepts.
- **Document enrichment** – The document enrichment feature helps you control both **what** documents and document attributes are ingested into your index and also **how** they are ingested. This can be accomplished through two approaches:
  - **Configure basic operations** – Use basic operations to add, update, or delete document attributes from your data. For example, you can scrub PII data by choosing to delete any document attributes related to PII.
  - **Configure Lambda functions** – Use a preconfigured Lambda function to perform more customized, advanced document attribute manipulation logic to your data. For example,

your enterprise data might be stored as scanned images. In that case, you can use a Lambda function to run optical character recognition (OCR) on the scanned documents to extract text from them. Then, each scanned document is treated as a text document during ingestion. Finally, during chat, Amazon Q will factor the textual data extracted from the scanned documents when it generates responses.

When you implement your solution, you can choose to combine both document enrichment approaches. You can use basic operations to do a first parse of your data and then use a Lambda function for more complex operations. For more information, see Document enrichment in Amazon Q Business.

- **Integration** – After you create your Amazon Q Business application, you can integrate it into other applications, such as Slack or Microsoft Teams. For example, see Deploy a Slack gateway forAmazon Q Business and Deploy a Microsoft Teams gateway for Amazon Q Business (AWS blog posts).

End-user customization

Amazon Q Business supports uploading documents that might not be stored in your organization's data sources and index. Uploaded documents are not stored. They are available for use only for the conversation in which the documents are uploaded. Amazon Q Business supports specific document types for upload. For more information, see Upload files and chat in Amazon Q Business.

Amazon Q Business includes a filtering by document attribute feature. Both administrators and end users can use this feature. Administrators can customize and control chat responses for end users by using attributes. For example, if data source type is an attribute attached to your documents, you can specify that chat responses be generated only from a specific data source. Or, you can allow end users to restrict the scope of chat responses by using the attribute filters that you have selected.

End users can create lightweight, purpose-built Amazon Q Apps within your broader Amazon Q Business application environment. Amazon Q apps allow task automation for a specific domain, such as a purpose-built app for marketing team.

Amazon SageMaker Canvas

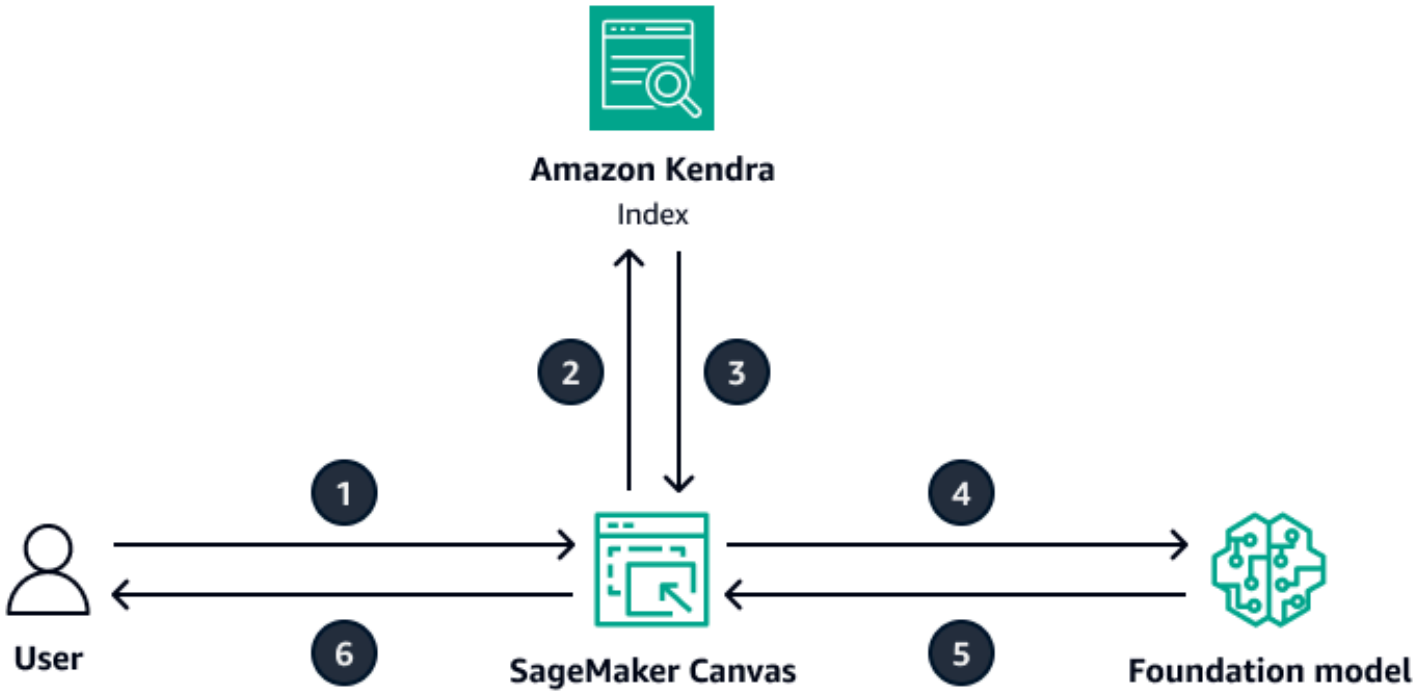
Amazon SageMaker Canvas helps you use machine learning to generate predictions without needing to write any code. It provides a no-code visual interface that empowers you to prepare data, build, and deploy ML models, streamlining the end-to-end ML lifecycle in a unified

environment. The complexities of data preparation, model development, bias detection, explainability, and monitoring are abstracted away behind an intuitive interface. Users don't need to be SageMaker or machine learning operations (MLOps) experts to develop, operationalize, and monitor models with SageMaker Canvas.

With SageMaker Canvas, the RAG functionality is provided through a no-code, document querying feature. You can enrich the chat experience in SageMaker Canvas by using an Amazon Kendra index as the underlying enterprise search. For more information, see Extract information from documents with document querying.

Connecting SageMaker Canvas to the Amazon Kendra index requires a one-time setup. As part of the domain configuration, a cloud administrator can choose one or more Kendra indexes that the user can query when interacting with SageMaker Canvas. For instructions about how to enable the document querying feature, see Getting started with using Amazon SageMaker Canvas.

SageMaker Canvas manages the underlying communication between Amazon Kendra and the selected foundation model. For more information about the foundation models that SageMaker Canvas supports, see Generative AI foundation models in SageMaker Canvas. The following diagram shows how the document querying feature works after the cloud administrator has connected SageMaker Canvas to an Amazon Kendra index.



The diagram shows the following workflow:

1. The user starts a new chat in SageMaker Canvas, turns on **Query documents**, selects the target index, and then submits a question.
2. SageMaker Canvas uses the query to search the Amazon Kendra index for relevant data.
3. SageMaker Canvas retrieves the data and its sources from the Amazon Kendra index.
4. SageMaker Canvas updates the prompt to include the retrieved context from the Amazon Kendra index and submits the prompt to the foundation model.
5. The foundation model uses the original question and the retrieved context to generate an answer.
6. SageMaker Canvas provides the generated answer to the user. It includes references to the data sources, such as documents, that were used to generate the response.

Custom Retrieval Augmented Generation architectures on AWS

The previous section describes how to use a fully managed AWS service for Retrieval Augmented Generation (RAG). However, some use cases require more control over the system components, such as the retriever or the LLM (also called the *generator*). For example, you might need the flexibility to choose your own vector database or access an unsupported data source. For these use cases, you can build a custom RAG architecture.

This section contains the following topics:

- Retrievers for RAG workflows
- Generators for RAG workflows

For more information about how to choose between the retriever and generator options in this section, see Choosing a Retrieval Augmented Generation option on AWS in this guide.

Retrievers for RAG workflows

This section explains how to build a retriever. You can use a fully managed semantic search solution, such as Amazon Kendra, or you can build a custom semantic search by using an AWS vector database.

Before you review the retriever options, make sure that you understand the three steps of the vector search process:

1. You separate the documents that need to be indexed into smaller parts. This is called *chunking*.
2. You use a process called embedding to convert each chunk into a mathematical vector. Then, you index each vector in a vector database. The approach that you use to index the documents influences the speed and accuracy of the search. The indexing approach depends on the vector database and the configuration options that it provides.
3. You convert the user query into a vector by using the same process. The retriever searches the vector database for vectors that are similar to the user's query vector. Similarity is calculated by using metrics such as Euclidean distance, cosine distance, or dot product.

This guide describes how to use the following AWS services or third-party services to build custom retrieval layer on AWS:

- Amazon Kendra
- Amazon OpenSearch Service
- Amazon Aurora PostgreSQL and pgvector
- Amazon Neptune Analytics
- Amazon MemoryDB
- Amazon DocumentDB
- Pinecone
- MongoDB Atlas
- Weaviate

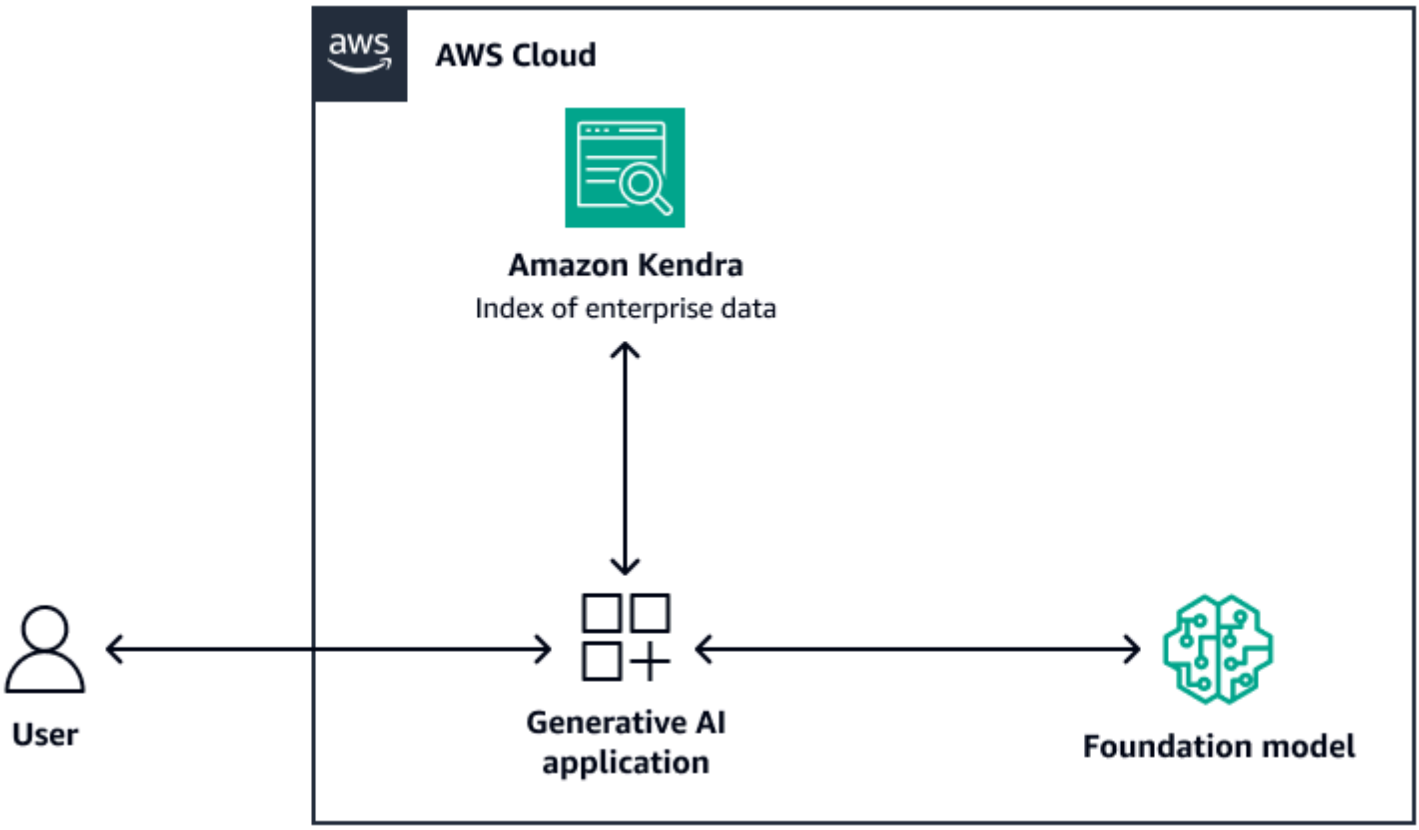
Amazon Kendra

Amazon Kendra is a fully managed, intelligent search service that uses natural language processing and advanced machine learning algorithms to return specific answers to search questions from your data. Amazon Kendra helps you directly ingest documents from multiple sources and query the documents after they have synced successfully. The syncing process creates the necessary infrastructure required to create a vector search on the ingested document. Therefore, Amazon Kendra does not require the traditional three steps of the vector search process. After the initial sync, you can use a defined schedule to handle ongoing ingestion.

The following are the advantages of using Amazon Kendra for RAG:

- You do not have to maintain a vector database because Amazon Kendra handles the entire vector search process.
- Amazon Kendra contains pre-built connectors for popular data sources, such as databases, website crawlers, Amazon S3 buckets, Microsoft SharePoint instances, and Atlassian Confluence instances. Connectors developed by AWS Partners are available, such as connectors for Box and GitLab.
- Amazon Kendra provides access control list (ACL) filtering that returns only documents that the end user has access to.
- Amazon Kendra can boost responses based on metadata, such as date or source repository.

The following image shows a sample architecture that uses Amazon Kendra as the retrieval layer of the RAG system. For more information, see Quickly build high-accuracy Generative AI applications on enterprise data using Amazon Kendra, LangChain, and large language models (AWS blog post).



For the foundation model, you can use Amazon Bedrock or an LLM deployed through Amazon SageMaker JumpStart. You can use AWS Lambda with LangChain to orchestrate the flow between the user, Amazon Kendra, and the LLM. To build a RAG system that uses Amazon Kendra, LangChain, and various LLMs, see the Amazon Kendra LangChain Extensions GitHub repository.

Amazon OpenSearch Service

Amazon OpenSearch Service provides built-in ML algorithms for k-nearest neighbors (k-NN) search in order to perform a vector search. OpenSearch Service also provides a vector engine for Amazon EMR Serverless. You can use this vector engine to build a RAG system that has scalable and high-performing vector storage and search capabilities. For more information about how to build a RAG system by using OpenSearch Serverless, see Build scalable and serverless RAG workflows with a vector engine for Amazon OpenSearch Serverless and Amazon Bedrock Claude models (AWS blog post).

The following are the advantages of using OpenSearch Service for vector search:

- It provides complete control over the vector database, including building a scalable vector search by using OpenSearch Serverless.
- It provides control over the chunking strategy.
- It uses approximate nearest neighbor (ANN) algorithms from the Non-Metric Space Library (NMSLIB), Faiss, and Apache Lucene libraries to power a k-NN search. You can change the algorithm based on the use case. For more information about the options for customizing vector search through OpenSearch Service, see Amazon OpenSearch Service vector database capabilities explained (AWS blog post).
- OpenSearch Serverless integrates with Amazon Bedrock knowledge bases as a vector index.

Amazon Aurora PostgreSQL and pgvector

Amazon Aurora PostgreSQL-Compatible Edition is a fully managed relational database engine that helps you set up, operate, and scale PostgreSQL deployments. pgvector is an open-source PostgreSQL extension that provides vector similarity search capabilities. This extension is available for both Aurora PostgreSQL-Compatible and for Amazon Relational Database Service (Amazon RDS) for PostgreSQL. For more information about how to build a RAG-based system that uses Aurora PostgreSQL-Compatible and pgvector, see the following AWS blog posts:

- Building AI-powered search in PostgreSQL using Amazon SageMaker and pgvector
- Leverage pgvector and Amazon Aurora PostgreSQL for Natural Language Processing, Chatbots, and Sentiment Analysis

The following are the advantages of using pgvector and Aurora PostgreSQL-Compatible:

- It supports exact and approximate nearest neighbor search. It also supports the following similarity metrics: L2 distance, inner product, and cosine distance.
- It supports Inverted File with Flat Compression (IVFFlat) and Hierarchical Navigable Small Worlds (HNSW) indexing.
- You can combine the vector search with queries over domain-specific data that is available in the same PostgreSQL instance.
- Aurora PostgreSQL-Compatible is optimized for I/O and provides tiered caching. For workloads that exceed the available instance memory, pgvector can increase the queries per second for vector search by up to 8 times.

Amazon Neptune Analytics

Amazon Neptune Analytics is a memory-optimized graph database engine for analytics. It supports a library of optimized graph analytic algorithms, low-latency graph queries, and vector search capabilities within graph traversals. It also has built-in vector similarity search. It provides one endpoint to create a graph, load data, invoke queries, and perform vector similarity search. For more information about how to build a RAG-based system that uses Neptune Analytics, see Using knowledge graphs to build GraphRAG applications with Amazon Bedrock and Amazon Neptune (AWS blog post).

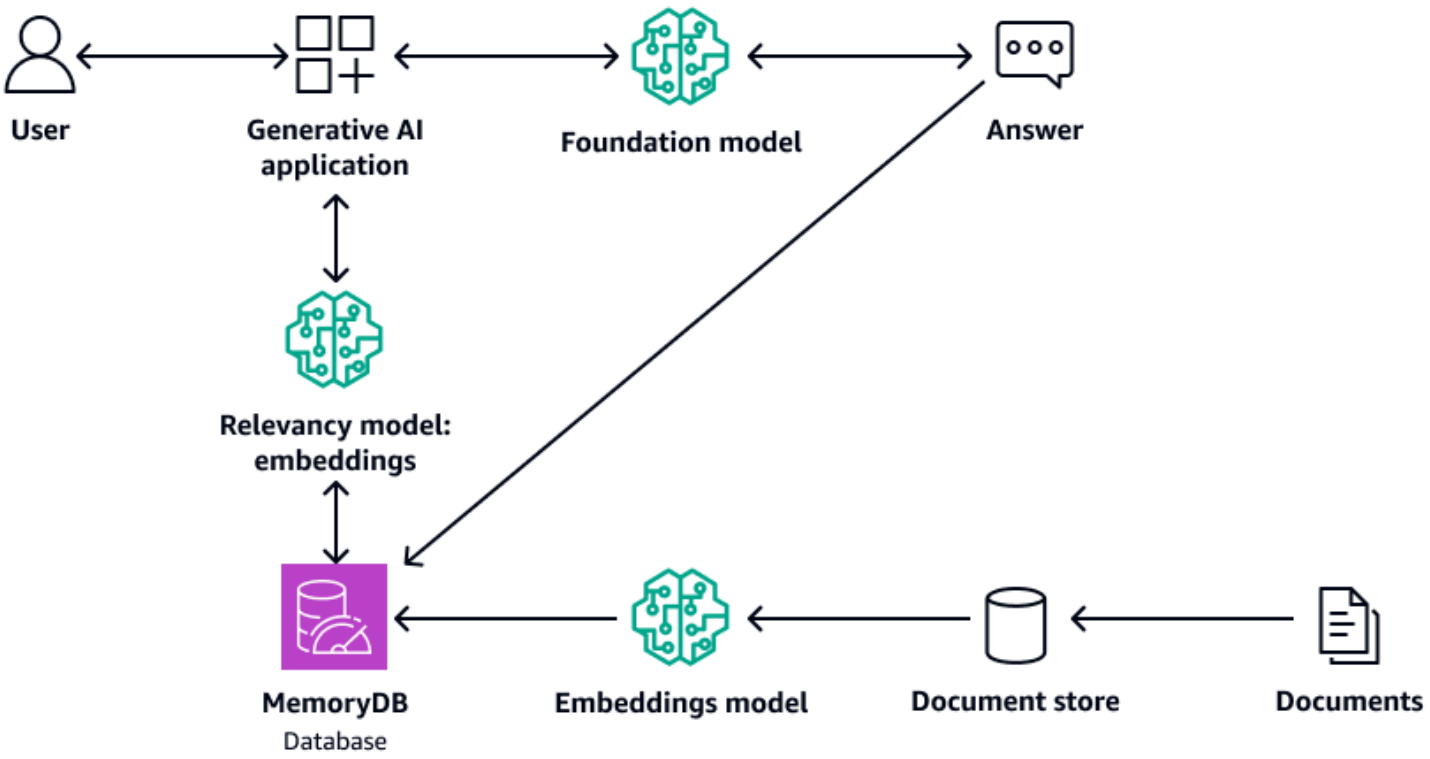
The following are the advantages of using Neptune Analytics:

- You can store and search embeddings in graph queries.
- If you integrate Neptune Analytics with LangChain, this architecture supports natural language graph queries.
- This architecture stores large graph datasets in memory.

Amazon MemoryDB

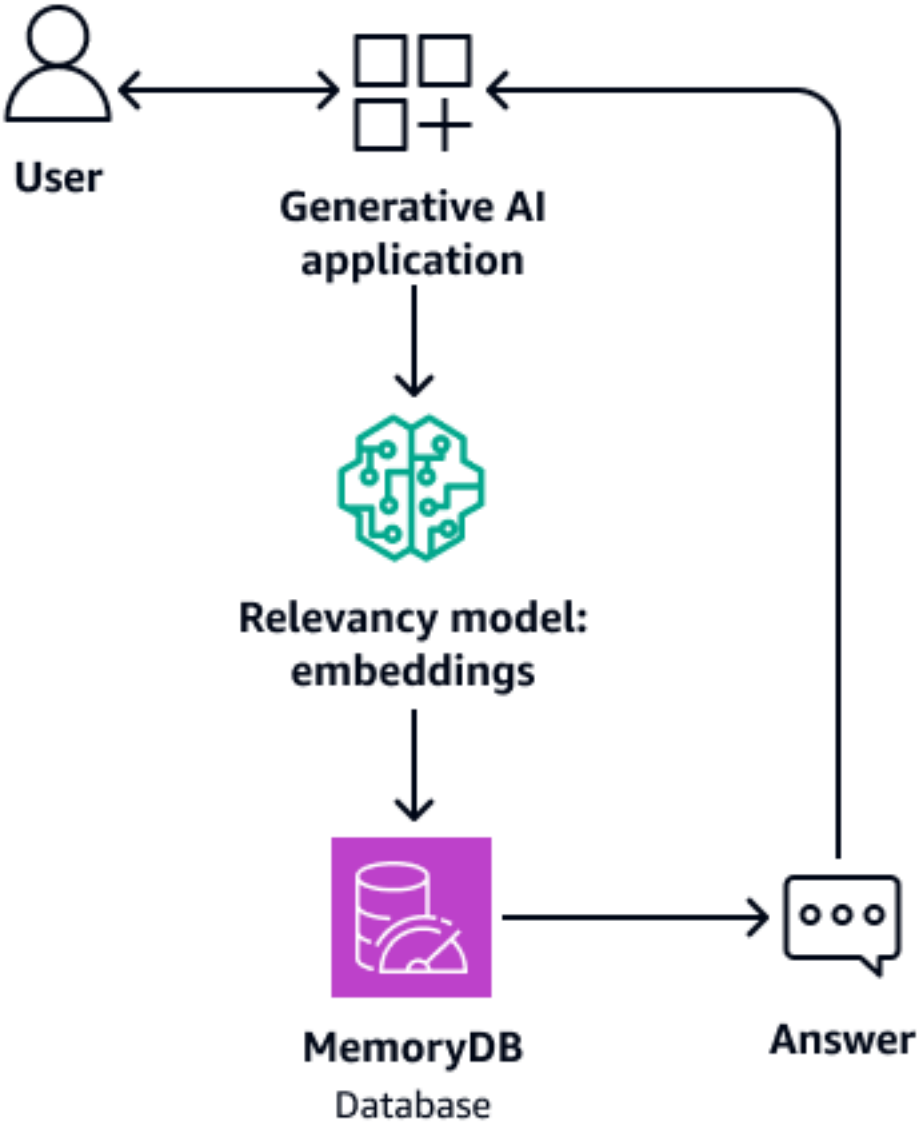
Amazon MemoryDB is a durable, in-memory database service that delivers ultra-fast performance. All of your data is stored in memory, which supports microsecond read, single-digit millisecond write latency, and high throughput. Vector search for MemoryDB extends the functionality of MemoryDB and can be used in conjunction with existing MemoryDB functionality. For more information, see the Question answering with LLM and RAG repository on GitHub.

The following diagram shows a sample architecture that uses MemoryDB as the vector database.



The following are the advantages of using MemoryDB:

- It supports both Flat and HNSW indexing algorithms. For more information, see Vector search for Amazon MemoryDB is now generally available on the AWS News Blog
- It can also act as a buffer memory for the foundation model. This means that previously answered questions are retrieved from the buffer instead of going through the retrieval and generation process again. The following diagram shows this process.



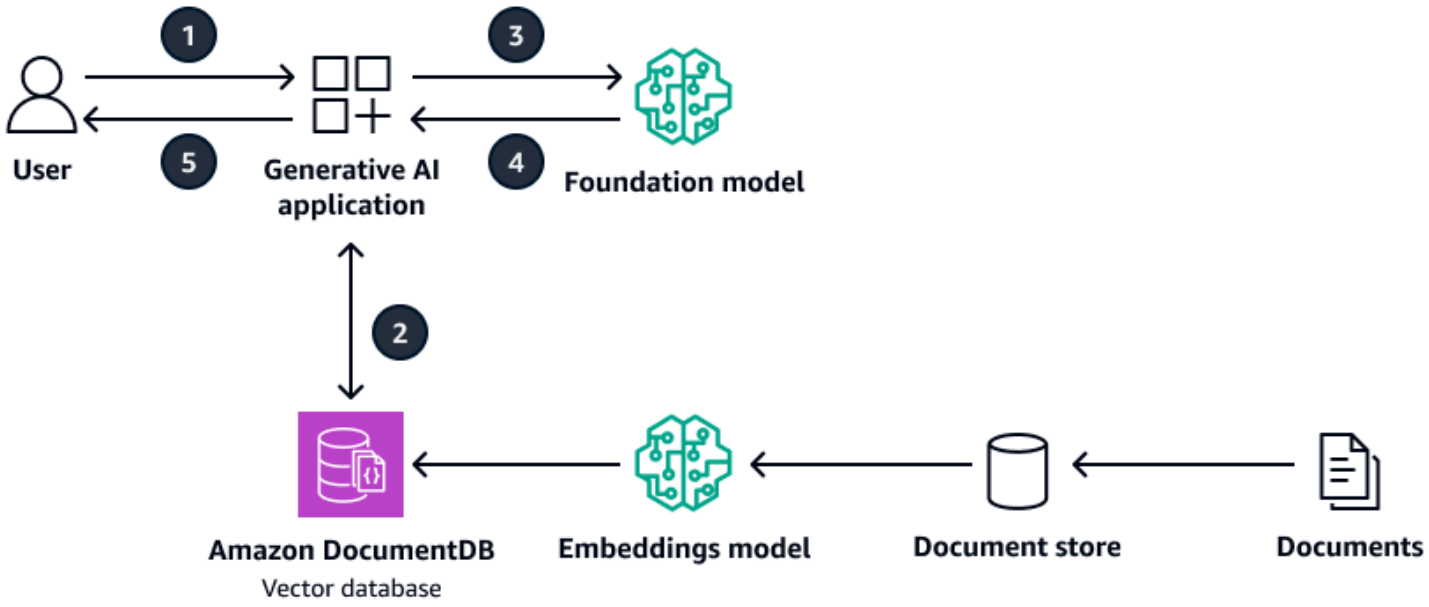
- Because it uses an in-memory database, this architecture provides single-digit millisecond query time for the semantic search.
- It provides up to 33,000 queries per second at 95–99% recall and 26,500 queries per second at greater than 99% recall. For more information, see the AWS re:Invent 2023 - Ultra-low latency vector search for Amazon MemoryDB video on YouTube.

Amazon DocumentDB

Amazon DocumentDB (with MongoDB compatibility) is a fast, reliable, and fully managed database service. It makes it easy to set up, operate, and scale MongoDB-compatible databases in the cloud. Vector search for Amazon DocumentDB combines the flexibility and rich querying capability of a

JSON-based document database with the power of vector search. For more information, see the Question answering with LLM and RAG repository on GitHub.

The following diagram shows a sample architecture that uses Amazon DocumentDB as the vector database.



The diagram shows the following workflow:

1. The user submits a query to the generative AI application.
2. The generative AI application performs a similarity search in the Amazon DocumentDB vector database and retrieves the relevant document extracts.
3. The generative AI application updates the user query with the retrieved context and submits the prompt to the target foundation model.
4. The foundation model uses the context to generate a response to the user's question and returns the response.
5. The generative AI application returns the response to the user.

The following are the advantages of using Amazon DocumentDB:

- It supports both HNSW and IVFFlat indexing methods.
- It supports up to 2,000 dimensions in the vector data and supports the Euclidean, cosine, and dot product distance metrics.

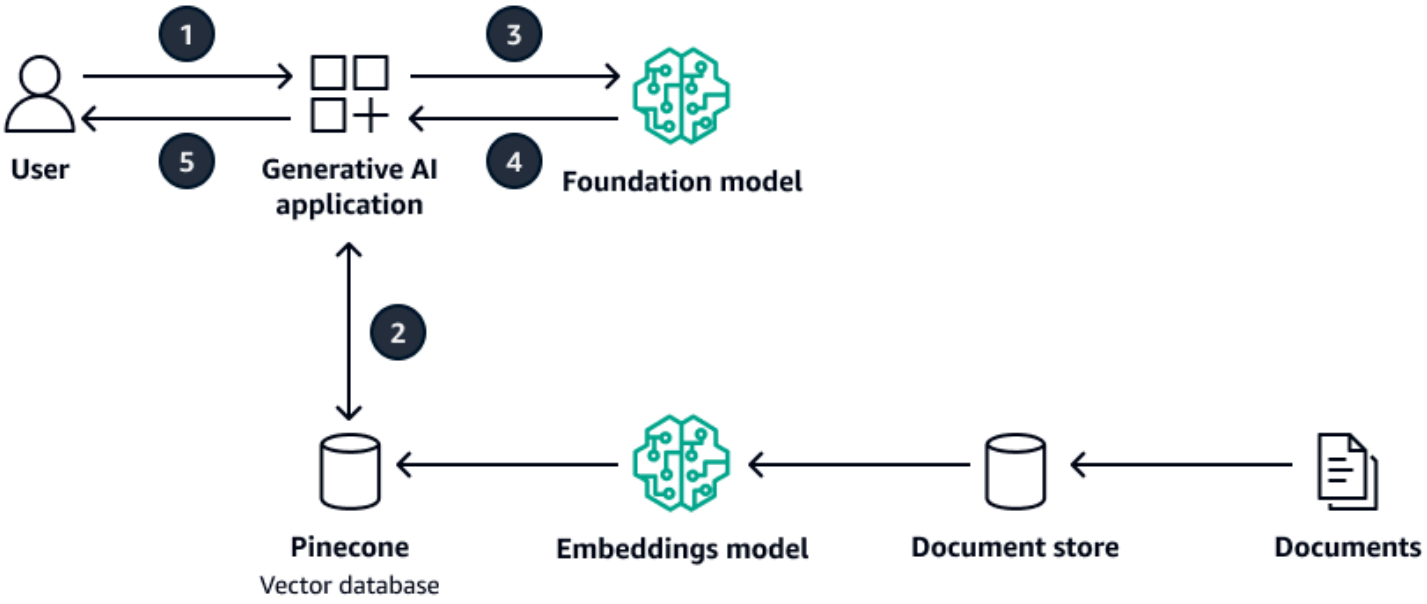
- It provides millisecond response times.

Pinecone

Pinecone is a fully managed vector database that helps you add vector search to production applications. It is available through the AWS Marketplace. Billing is based on usage, and charges are calculated by multiplying the pod price by the pod count. For more information about how to build a RAG-based system that uses Pinecone, see the following AWS blog posts:

- Mitigate hallucinations through RAG using Pinecone vector database & Llama-2 from Amazon SageMaker JumpStart
- Use Amazon SageMaker Studio to build a RAG question answering solution with Llama 2, LangChain, and Pinecone for fast experimentation

The following diagram shows a sample architecture that uses Pinecone as the vector database.



The diagram shows the following workflow:

1. The user submits a query to the generative AI application.
2. The generative AI application performs a similarity search in the Pinecone vector database and retrieves the relevant document extracts.
3. The generative AI application updates the user query with the retrieved context and submits the prompt to the target foundation model.

4. The foundation model uses the context to generate a response to the user's question and returns the response.
5. The generative AI application returns the response to the user.

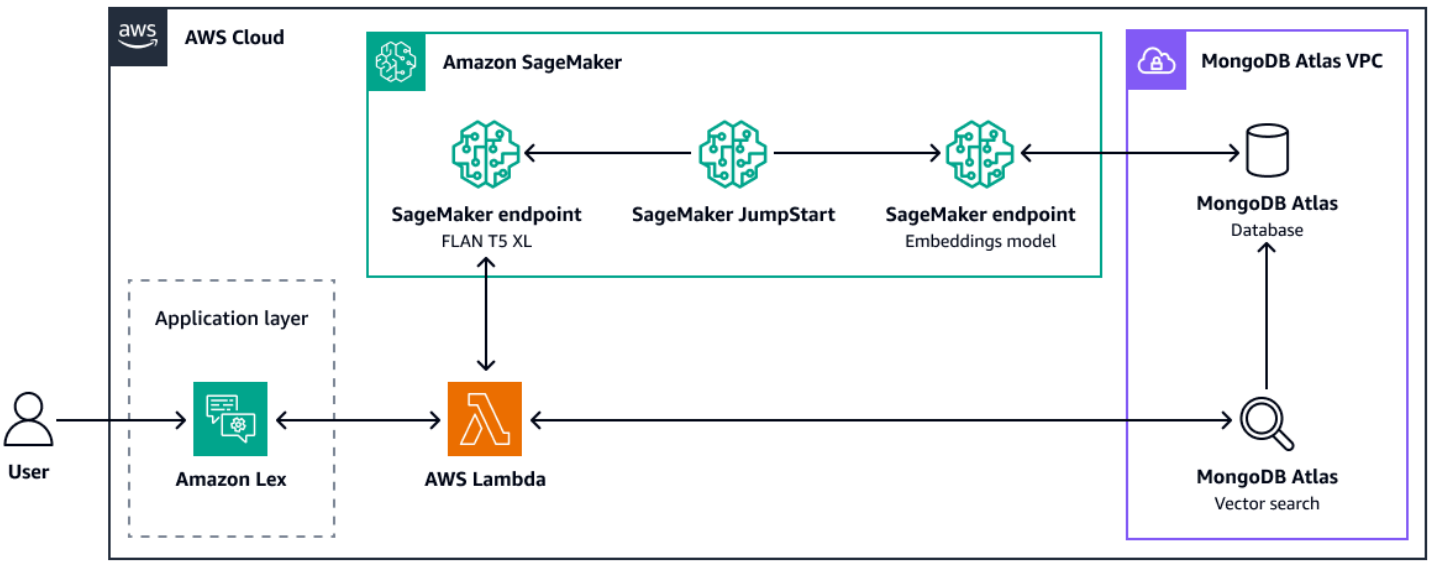
The following are the advantages of using Pinecone:

- It's a fully managed vector database and takes away the overhead of managing your own infrastructure.
- It provides the additional features of filtering, live index updates, and keyword boosting (hybrid search).

MongoDB Atlas

MongoDB Atlas is a fully managed cloud database that handles all the complexity of deploying and managing your deployments on AWS. You can use Vector search for MongoDB Atlas to store vector embeddings in your MongoDB database. Amazon Bedrock knowledge bases supports MongoDB Atlas for vector storage. For more information, see Get Started with the Amazon Bedrock Knowledge Base Integration in the MongoDB documentation.

For more information about how to use MongoDB Atlas vector search for RAG, see Retrieval-Augmented Generation with LangChain, Amazon SageMaker JumpStart, and MongoDB Atlas Semantic Search (AWS blog post). The following diagram shows the solution architecture detailed in this blog post.



The following are the advantages of using MongoDB Atlas vector search:

- You can use your existing implementation of MongoDB Atlas to store and search vector embeddings.
- You can use the MongoDB Query API to query the vector embeddings.
- You can independently scale the vector search and database.
- Vector embeddings are stored near the source data (documents), which improves the indexing performance.

Weaviate

Weaviate is a popular open source, low-latency vector database that supports multimodal media types, such as text and images. The database stores both objects and vectors, which combines vector search with structured filtering. For more information about using Weaviate and Amazon Bedrock to build a RAG workflow, see Build enterprise-ready generative AI solutions with Cohere foundation models in Amazon Bedrock and Weaviate vector database on AWS Marketplace (AWS blog post).

The following are the advantages of using Weaviate:

- It is open source and backed by a strong community.
- It is built for hybrid search (both vectors and keywords).
- You can deploy it on AWS as a managed software as a service (SaaS) offering or as a Kubernetes cluster.

Generators for RAG workflows

Large language models (LLMs) are very large deep learning models that are pretrained on vast amounts of data. They are incredibly flexible. LLMs can perform varied tasks, such as answering questions, summarizing documents, translating languages, and completing sentences. They have the potential to disrupt content creation and the way people use search engines and virtual assistants. While not perfect, LLMs demonstrate a remarkable ability to make predictions based on a relatively small prompt or number of inputs.

LLMs are a critical component of a RAG solution. For custom RAG architectures, there are two AWS services that serve as the primary options:

- Amazon Bedrock is a fully managed service that makes LLMs from leading AI companies and Amazon available for your use through a unified API.
- Amazon SageMaker JumpStart is an ML hub that offers foundation models, built-in algorithms, and prebuilt ML solutions. With SageMaker JumpStart, you can access pretrained models, including foundation models. You can also use your own data to fine-tune the pretrained models.

Amazon Bedrock

Amazon Bedrock offers industry-leading models from Anthropic, Stability AI, Meta, Cohere, AI21 Labs, Mistral AI, and Amazon. For a complete list, see Supported foundation models in Amazon Bedrock. Amazon Bedrock also allows you to customize models with your own data.

You can evaluate the model performance to determine which are best suited for your RAG use case. You can test the latest models and also test to see which capabilities and features provide the best results and for the best price. The Anthropic Claude Sonnet model is a common choice for RAG applications because it excels at a wide range of tasks and provides a high degree of reliability and predictability.

SageMaker JumpStart

SageMaker JumpStart provides pretrained, open source models for a wide range of problem types. You can incrementally train and fine-tune these models before deployment. You can access the pretrained models, solution templates, and examples through the SageMaker JumpStart landing page in Amazon SageMaker Studio or use the SageMaker Python SDK.

SageMaker JumpStart offers state-of-the-art foundation models for use cases such as content writing, code generation, question answering, copywriting, summarization, classification, information retrieval, and more. Use JumpStart foundation models to build your own generative AI solutions and integrate custom solutions with additional SageMaker features. For more information, see Getting started with Amazon SageMaker JumpStart.

SageMaker JumpStart onboards and maintains publicly available foundation models for you to access, customize, and integrate into your ML life cycles. For more information, see Publicly available foundation models. SageMaker JumpStart also includes proprietary foundation models from third-party providers. For more information, see Proprietary foundation models.

Choosing a Retrieval Augmented Generation option on AWS

The Fully managed RAG options and Custom RAG architectures sections of this guide describe various approaches for building a RAG-based search solution on AWS. This section describes how to select between these options based on your use case. In some situations, more than one option might work. In that scenario, the choice depends on the ease of implementation, skills available in your organization, and your company's policies and standards.

We recommend that you consider the fully managed and custom RAG options in the following sequence and choose the first option that fits your use case:

1. Use Amazon Q Business unless:
   - This service is not available in your AWS Region, and your data cannot be moved to a Region where it is available
   - You have a specific reason to customize the RAG workflow
   - You want to use an existing vector database or a specific LLM
2. Use knowledge bases for Amazon Bedrock unless:
   - You have a vector database that is not supported
   - You have a specific reason to customize the RAG workflow
3. Combine Amazon Kendra with your choice of generator unless:
   - You want to choose your own vector database
   - You want to customize the chunking strategy
4. If you want more control over the retriever and want to select your own vector database:
   - If you don't have an existing vector database and don't need low latency or graph queries, consider using Amazon OpenSearch Service.
   - If you have an existing PostgreSQL vector database, consider using the Amazon Aurora PostgreSQL and pgvector option.
   - If you need low latency, consider an in-memory option, such as Amazon MemoryDB or Amazon DocumentDB.
   - If you want to combine vector search with a graph query, consider Amazon Neptune Analytics.
   - If you are already using a third-party vector database or find a specific benefit from one, consider Pinecone, MongoDB Atlas, and Weaviate.

5. If you want to choose an LLM:
   - If you use Amazon Q Business, you can't choose the LLM.
   - If you use Amazon Bedrock, you can choose one of the supported foundation models.
   - If you use Amazon Kendra or a custom vector database, you can use one of the generators described in this guide or use a custom LLM.

   > **Note**
   >
   > You can also use your custom documents to fine-tune an existing LLM to increase the accuracy of its responses. For more information, see Comparing RAG and fine-tuning in this guide.

6. If you have an existing implementation of Amazon SageMaker Canvas that you want to use or if you want to compare RAG responses from different LLMs, consider Amazon SageMaker Canvas.

Conclusion

This guide describes the various options for building a Retrieval Augmented Generation (RAG) system on AWS. You can start with fully managed services, such as Amazon Q Business and Amazon Bedrock knowledge bases. If you want more control over the RAG workflow, you can choose a custom retriever. For a generator, you can use an API to call a supported LLM in Amazon Bedrock, or you can deploy your own LLM by using Amazon SageMaker JumpStart. Review the recommendations in Choosing a RAG option to determine which option is best suited for your use case. After you select the best option for your use case, use the references provided in this guide to start building your RAG-based application.

Document history

The following table describes significant changes to this guide. If you want to be notified about future updates, you can subscribe to an RSS feed.

Change	Description	Date
Initial publication	—	October 28, 2024

AWS Prescriptive Guidance glossary

The following are commonly used terms in strategies, guides, and patterns provided by AWS Prescriptive Guidance. To suggest entries, please use the **Provide feedback** link at the end of the glossary.

Numbers

7 Rs

Seven common migration strategies for moving applications to the cloud. These strategies build upon the 5 Rs that Gartner identified in 2011 and consist of the following:

- Refactor/re-architect – Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability. This typically involves porting the operating system and database. Example: Migrate your on-premises Oracle database to the Amazon Aurora PostgreSQL-Compatible Edition.
- Replatform (lift and reshape) – Move an application to the cloud, and introduce some level of optimization to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Amazon Relational Database Service (Amazon RDS) for Oracle in the AWS Cloud.
- Repurchase (drop and shop) – Switch to a different product, typically by moving from a traditional license to a SaaS model. Example: Migrate your customer relationship management (CRM) system to Salesforce.com.
- Rehost (lift and shift) – Move an application to the cloud without making any changes to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Oracle on an EC2 instance in the AWS Cloud.
- Relocate (hypervisor-level lift and shift) – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations. You migrate servers from an on-premises platform to a cloud service for the same platform. Example: Migrate a Microsoft Hyper-V application to AWS.
- Retain (revisit) – Keep applications in your source environment. These might include applications that require major refactoring, and you want to postpone that work until a later time, and legacy applications that you want to retain, because there's no business justification for migrating them.

- Retire – Decommission or remove applications that are no longer needed in your source environment.

A

ABAC

See attribute-based access control.

abstracted services

See managed services.

ACID

See atomicity, consistency, isolation, durability.

active-active migration

A database migration method in which the source and target databases are kept in sync (by using a bidirectional replication tool or dual write operations), and both databases handle transactions from connecting applications during migration. This method supports migration in small, controlled batches instead of requiring a one-time cutover. It's more flexible but requires more work than active-passive migration.

active-passive migration

A database migration method in which in which the source and target databases are kept in sync, but only the source database handles transactions from connecting applications while data is replicated to the target database. The target database doesn't accept any transactions during migration.

aggregate function

A SQL function that operates on a group of rows and calculates a single return value for the group. Examples of aggregate functions include `SUM` and `MAX`.

AI

See artificial intelligence.

AIOps

See artificial intelligence operations.

anonymization

The process of permanently deleting personal information in a dataset. Anonymization can help protect personal privacy. Anonymized data is no longer considered to be personal data.

anti-pattern

A frequently used solution for a recurring issue where the solution is counter-productive, ineffective, or less effective than an alternative.

application control

A security approach that allows the use of only approved applications in order to help protect a system from malware.

application portfolio

A collection of detailed information about each application used by an organization, including the cost to build and maintain the application, and its business value. This information is key to the portfolio discovery and analysis process and helps identify and prioritize the applications to be migrated, modernized, and optimized.

artificial intelligence (AI)

The field of computer science that is dedicated to using computing technologies to perform cognitive functions that are typically associated with humans, such as learning, solving problems, and recognizing patterns. For more information, see What is Artificial Intelligence?

artificial intelligence operations (AIOps)

The process of using machine learning techniques to solve operational problems, reduce operational incidents and human intervention, and increase service quality. For more information about how AIOps is used in the AWS migration strategy, see the operations integration guide.

asymmetric encryption

An encryption algorithm that uses a pair of keys, a public key for encryption and a private key for decryption. You can share the public key because it isn't used for decryption, but access to the private key should be highly restricted.

atomicity, consistency, isolation, durability (ACID)

A set of software properties that guarantee the data validity and operational reliability of a database, even in the case of errors, power failures, or other problems.

attribute-based access control (ABAC)

The practice of creating fine-grained permissions based on user attributes, such as department, job role, and team name. For more information, see ABAC for AWS in the AWS Identity and Access Management (IAM) documentation.

authoritative data source

A location where you store the primary version of data, which is considered to be the most reliable source of information. You can copy data from the authoritative data source to other locations for the purposes of processing or modifying the data, such as anonymizing, redacting, or pseudonymizing it.

Availability Zone

A distinct location within an AWS Region that is insulated from failures in other Availability Zones and provides inexpensive, low-latency network connectivity to other Availability Zones in the same Region.

AWS Cloud Adoption Framework (AWS CAF)

A framework of guidelines and best practices from AWS to help organizations develop an efficient and effective plan to move successfully to the cloud. AWS CAF organizes guidance into six focus areas called perspectives: business, people, governance, platform, security, and operations. The business, people, and governance perspectives focus on business skills and processes; the platform, security, and operations perspectives focus on technical skills and processes. For example, the people perspective targets stakeholders who handle human resources (HR), staffing functions, and people management. For this perspective, AWS CAF provides guidance for people development, training, and communications to help ready the organization for successful cloud adoption. For more information, see the AWS CAF website and the AWS CAF whitepaper.

AWS Workload Qualification Framework (AWS WQF)

A tool that evaluates database migration workloads, recommends migration strategies, and provides work estimates. AWS WQF is included with AWS Schema Conversion Tool (AWS SCT). It analyzes database schemas and code objects, application code, dependencies, and performance characteristics, and provides assessment reports.

B

bad bot

A [bot](#) that is intended to disrupt or cause harm to individuals or organizations.

BCP

See [business continuity planning](#).

behavior graph

A unified, interactive view of resource behavior and interactions over time. You can use a behavior graph with Amazon Detective to examine failed logon attempts, suspicious API calls, and similar actions. For more information, see [Data in a behavior graph](#) in the Detective documentation.

big-endian system

A system that stores the most significant byte first. See also [endianness](#).

binary classification

A process that predicts a binary outcome (one of two possible classes). For example, your ML model might need to predict problems such as “Is this email spam or not spam?" or "Is this product a book or a car?"

bloom filter

A probabilistic, memory-efficient data structure that is used to test whether an element is a member of a set.

blue/green deployment

A deployment strategy where you create two separate but identical environments. You run the current application version in one environment (blue) and the new application version in the other environment (green). This strategy helps you quickly roll back with minimal impact.

bot

A software application that runs automated tasks over the internet and simulates human activity or interaction. Some bots are useful or beneficial, such as web crawlers that index information on the internet. Some other bots, known as *bad bots*, are intended to disrupt or cause harm to individuals or organizations.

botnet

Networks of bots that are infected by malware and are under the control of a single party, known as a *bot herder* or *bot operator*. Botnets are the best-known mechanism to scale bots and their impact.

branch

A contained area of a code repository. The first branch created in a repository is the *main branch*. You can create a new branch from an existing branch, and you can then develop features or fix bugs in the new branch. A branch you create to build a feature is commonly referred to as a *feature branch*. When the feature is ready for release, you merge the feature branch back into the main branch. For more information, see About branches (GitHub documentation).

break-glass access

In exceptional circumstances and through an approved process, a quick means for a user to gain access to an AWS account that they don't typically have permissions to access. For more information, see the Implement break-glass procedures indicator in the AWS Well-Architected guidance.

brownfield strategy

The existing infrastructure in your environment. When adopting a brownfield strategy for a system architecture, you design the architecture around the constraints of the current systems and infrastructure. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

buffer cache

The memory area where the most frequently accessed data is stored.

business capability

What a business does to generate value (for example, sales, customer service, or marketing). Microservices architectures and development decisions can be driven by business capabilities. For more information, see the Organized around business capabilities section of the Running containerized microservices on AWS whitepaper.

business continuity planning (BCP)

A plan that addresses the potential impact of a disruptive event, such as a large-scale migration, on operations and enables a business to resume operations quickly.

C

CAF

See AWS Cloud Adoption Framework.

canary deployment

The slow and incremental release of a version to end users. When you are confident, you deploy the new version and replace the current version in its entirety.

CCoE

See Cloud Center of Excellence.

CDC

See change data capture.

change data capture (CDC)

The process of tracking changes to a data source, such as a database table, and recording metadata about the change. You can use CDC for various purposes, such as auditing or replicating changes in a target system to maintain synchronization.

chaos engineering

Intentionally introducing failures or disruptive events to test a system's resilience. You can use AWS Fault Injection Service (AWS FIS) to perform experiments that stress your AWS workloads and evaluate their response.

CI/CD

See continuous integration and continuous delivery.

classification

A categorization process that helps generate predictions. ML models for classification problems predict a discrete value. Discrete values are always distinct from one another. For example, a model might need to evaluate whether or not there is a car in an image.

client-side encryption

Encryption of data locally, before the target AWS service receives it.

Cloud Center of Excellence (CCoE)

A multi-disciplinary team that drives cloud adoption efforts across an organization, including developing cloud best practices, mobilizing resources, establishing migration timelines, and leading the organization through large-scale transformations. For more information, see the CCoE posts on the AWS Cloud Enterprise Strategy Blog.

cloud computing

The cloud technology that is typically used for remote data storage and IoT device management. Cloud computing is commonly connected to edge computing technology.

cloud operating model

In an IT organization, the operating model that is used to build, mature, and optimize one or more cloud environments. For more information, see Building your Cloud Operating Model.

cloud stages of adoption

The four phases that organizations typically go through when they migrate to the AWS Cloud:

- Project – Running a few cloud-related projects for proof of concept and learning purposes
- Foundation – Making foundational investments to scale your cloud adoption (e.g., creating a landing zone, defining a CCoE, establishing an operations model)
- Migration – Migrating individual applications
- Re-invention – Optimizing products and services, and innovating in the cloud

These stages were defined by Stephen Orban in the blog post The Journey Toward Cloud-First & the Stages of Adoption on the AWS Cloud Enterprise Strategy blog. For information about how they relate to the AWS migration strategy, see the migration readiness guide.

CMDB

See configuration management database.

code repository

A location where source code and other assets, such as documentation, samples, and scripts, are stored and updated through version control processes. Common cloud repositories include GitHub or Bitbucket Cloud. Each version of the code is called a *branch*. In a microservice structure, each repository is devoted to a single piece of functionality. A single CI/CD pipeline can use multiple repositories.

cold cache

A buffer cache that is empty, not well populated, or contains stale or irrelevant data. This affects performance because the database instance must read from the main memory or disk, which is slower than reading from the buffer cache.

cold data

Data that is rarely accessed and is typically historical. When querying this kind of data, slow queries are typically acceptable. Moving this data to lower-performing and less expensive storage tiers or classes can reduce costs.

computer vision (CV)

A field of AI that uses machine learning to analyze and extract information from visual formats such as digital images and videos. For example, AWS Panorama offers devices that add CV to on-premises camera networks, and Amazon SageMaker provides image processing algorithms for CV.

configuration drift

For a workload, a configuration change from the expected state. It might cause the workload to become noncompliant, and it's typically gradual and unintentional.

configuration management database (CMDB)

A repository that stores and manages information about a database and its IT environment, including both hardware and software components and their configurations. You typically use data from a CMDB in the portfolio discovery and analysis stage of migration.

conformance pack

A collection of AWS Config rules and remediation actions that you can assemble to customize your compliance and security checks. You can deploy a conformance pack as a single entity in an AWS account and Region, or across an organization, by using a YAML template. For more information, see Conformance packs in the AWS Config documentation.

continuous integration and continuous delivery (CI/CD)

The process of automating the source, build, test, staging, and production stages of the software release process. CI/CD is commonly described as a pipeline. CI/CD can help you automate processes, improve productivity, improve code quality, and deliver faster. For more information, see Benefits of continuous delivery. CD can also stand for *continuous deployment*. For more information, see Continuous Delivery vs. Continuous Deployment.

CV

See computer vision.

D

data at rest

Data that is stationary in your network, such as data that is in storage.

data classification

A process for identifying and categorizing the data in your network based on its criticality and sensitivity. It is a critical component of any cybersecurity risk management strategy because it helps you determine the appropriate protection and retention controls for the data. Data classification is a component of the security pillar in the AWS Well-Architected Framework. For more information, see Data classification.

data drift

A meaningful variation between the production data and the data that was used to train an ML model, or a meaningful change in the input data over time. Data drift can reduce the overall quality, accuracy, and fairness in ML model predictions.

data in transit

Data that is actively moving through your network, such as between network resources.

data mesh

An architectural framework that provides distributed, decentralized data ownership with centralized management and governance.

data minimization

The principle of collecting and processing only the data that is strictly necessary. Practicing data minimization in the AWS Cloud can reduce privacy risks, costs, and your analytics carbon footprint.

data perimeter

A set of preventive guardrails in your AWS environment that help make sure that only trusted identities are accessing trusted resources from expected networks. For more information, see Building a data perimeter on AWS.

data preprocessing

To transform raw data into a format that is easily parsed by your ML model. Preprocessing data can mean removing certain columns or rows and addressing missing, inconsistent, or duplicate values.

data provenance

The process of tracking the origin and history of data throughout its lifecycle, such as how the data was generated, transmitted, and stored.

data subject

An individual whose data is being collected and processed.

data warehouse

A data management system that supports business intelligence, such as analytics. Data warehouses commonly contain large amounts of historical data, and they are typically used for queries and analysis.

database definition language (DDL)

Statements or commands for creating or modifying the structure of tables and objects in a database.

database manipulation language (DML)

Statements or commands for modifying (inserting, updating, and deleting) information in a database.

DDL

See database definition language.

deep ensemble

To combine multiple deep learning models for prediction. You can use deep ensembles to obtain a more accurate prediction or for estimating uncertainty in predictions.

deep learning

An ML subfield that uses multiple layers of artificial neural networks to identify mapping between input data and target variables of interest.

defense-in-depth

An information security approach in which a series of security mechanisms and controls are thoughtfully layered throughout a computer network to protect the confidentiality, integrity, and availability of the network and the data within. When you adopt this strategy on AWS, you add multiple controls at different layers of the AWS Organizations structure to help secure resources. For example, a defense-in-depth approach might combine multi-factor authentication, network segmentation, and encryption.

delegated administrator

In AWS Organizations, a compatible service can register an AWS member account to administer the organization's accounts and manage permissions for that service. This account is called the *delegated administrator* for that service. For more information and a list of compatible services, see Services that work with AWS Organizations in the AWS Organizations documentation.

deployment

The process of making an application, new features, or code fixes available in the target environment. Deployment involves implementing changes in a code base and then building and running that code base in the application's environments.

development environment

See environment.

detective control

A security control that is designed to detect, log, and alert after an event has occurred. These controls are a second line of defense, alerting you to security events that bypassed the preventative controls in place. For more information, see Detective controls in *Implementing security controls on AWS*.

development value stream mapping (DVSM)

A process used to identify and prioritize constraints that adversely affect speed and quality in a software development lifecycle. DVSM extends the value stream mapping process originally designed for lean manufacturing practices. It focuses on the steps and teams required to create and move value through the software development process.

digital twin

A virtual representation of a real-world system, such as a building, factory, industrial equipment, or production line. Digital twins support predictive maintenance, remote monitoring, and production optimization.

dimension table

In a star schema, a smaller table that contains data attributes about quantitative data in a fact table. Dimension table attributes are typically text fields or discrete numbers that behave like text. These attributes are commonly used for query constraining, filtering, and result set labeling.

disaster

An event that prevents a workload or system from fulfilling its business objectives in its primary deployed location. These events can be natural disasters, technical failures, or the result of human actions, such as unintentional misconfiguration or a malware attack.

disaster recovery (DR)

The strategy and process you use to minimize downtime and data loss caused by a disaster. For more information, see Disaster Recovery of Workloads on AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML

See database manipulation language.

domain-driven design

An approach to developing a complex software system by connecting its components to evolving domains, or core business goals, that each component serves. This concept was introduced by Eric Evans in his book, *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003). For information about how you can use domain-driven design with the strangler fig pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

DR

See disaster recovery.

drift detection

Tracking deviations from a baselined configuration. For example, you can use AWS CloudFormation to detect drift in system resources, or you can use AWS Control Tower to detect changes in your landing zone that might affect compliance with governance requirements.

DVSM

See development value stream mapping.

E

EDA

See exploratory data analysis.

EDI

See electronic data interchange.

edge computing

The technology that increases the computing power for smart devices at the edges of an IoT network. When compared with cloud computing, edge computing can reduce communication latency and improve response time.

electronic data interchange (EDI)

The automated exchange of business documents between organizations. For more information, see What is Electronic Data Interchange.

encryption

A computing process that transforms plaintext data, which is human-readable, into ciphertext.

encryption key

A cryptographic string of randomized bits that is generated by an encryption algorithm. Keys can vary in length, and each key is designed to be unpredictable and unique.

endianness

The order in which bytes are stored in computer memory. Big-endian systems store the most significant byte first. Little-endian systems store the least significant byte first.

endpoint

See service endpoint.

endpoint service

A service that you can host in a virtual private cloud (VPC) to share with other users. You can create an endpoint service with AWS PrivateLink and grant permissions to other AWS accounts or to AWS Identity and Access Management (IAM) principals. These accounts or principals can connect to your endpoint service privately by creating interface VPC endpoints. For more

information, see Create an endpoint service in the Amazon Virtual Private Cloud (Amazon VPC) documentation.

enterprise resource planning (ERP)

A system that automates and manages key business processes (such as accounting, MES, and project management) for an enterprise.

envelope encryption

The process of encrypting an encryption key with another encryption key. For more information, see Envelope encryption in the AWS Key Management Service (AWS KMS) documentation.

environment

An instance of a running application. The following are common types of environments in cloud computing:

- development environment – An instance of a running application that is available only to the core team responsible for maintaining the application. Development environments are used to test changes before promoting them to upper environments. This type of environment is sometimes referred to as a *test environment*.
- lower environments – All development environments for an application, such as those used for initial builds and tests.
- production environment – An instance of a running application that end users can access. In a CI/CD pipeline, the production environment is the last deployment environment.
- upper environments – All environments that can be accessed by users other than the core development team. This can include a production environment, preproduction environments, and environments for user acceptance testing.

epic

In agile methodologies, functional categories that help organize and prioritize your work. Epics provide a high-level description of requirements and implementation tasks. For example, AWS CAF security epics include identity and access management, detective controls, infrastructure security, data protection, and incident response. For more information about epics in the AWS migration strategy, see the program implementation guide.

ERP

See enterprise resource planning.

exploratory data analysis (EDA)

The process of analyzing a dataset to understand its main characteristics. You collect or aggregate data and then perform initial investigations to find patterns, detect anomalies, and check assumptions. EDA is performed by calculating summary statistics and creating data visualizations.

F

fact table

The central table in a star schema. It stores quantitative data about business operations. Typically, a fact table contains two types of columns: those that contain measures and those that contain a foreign key to a dimension table.

fail fast

A philosophy that uses frequent and incremental testing to reduce the development lifecycle. It is a critical part of an agile approach.

fault isolation boundary

In the AWS Cloud, a boundary such as an Availability Zone, AWS Region, control plane, or data plane that limits the effect of a failure and helps improve the resilience of workloads. For more information, see AWS Fault Isolation Boundaries.

feature branch

See branch.

features

The input data that you use to make a prediction. For example, in a manufacturing context, features could be images that are periodically captured from the manufacturing line.

feature importance

How significant a feature is for a model's predictions. This is usually expressed as a numerical score that can be calculated through various techniques, such as Shapley Additive Explanations (SHAP) and integrated gradients. For more information, see Machine learning model interpretability with :AWS.

feature transformation

To optimize data for the ML process, including enriching data with additional sources, scaling values, or extracting multiple sets of information from a single data field. This enables the ML model to benefit from the data. For example, if you break down the “2021-05-27 00:15:37” date into “2021”, “May”, “Thu”, and “15”, you can help the learning algorithm learn nuanced patterns associated with different data components.

few-shot prompting

Providing an LLM with a small number of examples that demonstrate the task and desired output before asking it to perform a similar task. This technique is an application of in-context learning, where models learn from examples (*shots*) that are embedded in prompts. Few-shot prompting can be effective for tasks that require specific formatting, reasoning, or domain knowledge. See also zero-shot prompting.

FGAC

See fine-grained access control.

fine-grained access control (FGAC)

The use of multiple conditions to allow or deny an access request.

flash-cut migration

A database migration method that uses continuous data replication through change data capture to migrate data in the shortest time possible, instead of using a phased approach. The objective is to keep downtime to a minimum.

FM

See foundation model.

foundation model (FM)

A large deep-learning neural network that has been training on massive datasets of generalized and unlabeled data. FMs are capable of performing a wide variety of general tasks, such as understanding language, generating text and images, and conversing in natural language. For more information, see What are Foundation Models.

G

generative AI

A subset of AI models that have been trained on large amounts of data and that can use a simple text prompt to create new content and artifacts, such as images, videos, text, and audio. For more information, see What is Generative AI.

geo blocking

See geographic restrictions.

geographic restrictions (geo blocking)

In Amazon CloudFront, an option to prevent users in specific countries from accessing content distributions. You can use an allow list or block list to specify approved and banned countries. For more information, see Restricting the geographic distribution of your content in the CloudFront documentation.

Gitflow workflow

An approach in which lower and upper environments use different branches in a source code repository. The Gitflow workflow is considered legacy, and the trunk-based workflow is the modern, preferred approach.

golden image

A snapshot of a system or software that is used as a template to deploy new instances of that system or software. For example, in manufacturing, a golden image can be used to provision software on multiple devices and helps improve speed, scalability, and productivity in device manufacturing operations.

greenfield strategy

The absence of existing infrastructure in a new environment. When adopting a greenfield strategy for a system architecture, you can select all new technologies without the restriction of compatibility with existing infrastructure, also known as brownfield. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

guardrail

A high-level rule that helps govern resources, policies, and compliance across organizational units (OUs). *Preventive guardrails* enforce policies to ensure alignment to compliance standards. They are implemented by using service control policies and IAM permissions boundaries.

Detective guardrails detect policy violations and compliance issues, and generate alerts for remediation. They are implemented by using AWS Config, AWS Security Hub, Amazon GuardDuty, AWS Trusted Advisor, Amazon Inspector, and custom AWS Lambda checks.

H

HA

See high availability.

heterogeneous database migration

Migrating your source database to a target database that uses a different database engine (for example, Oracle to Amazon Aurora). Heterogeneous migration is typically part of a re-architecting effort, and converting the schema can be a complex task. AWS provides AWS SCT that helps with schema conversions.

high availability (HA)

The ability of a workload to operate continuously, without intervention, in the event of challenges or disasters. HA systems are designed to automatically fail over, consistently deliver high-quality performance, and handle different loads and failures with minimal performance impact.

historian modernization

An approach used to modernize and upgrade operational technology (OT) systems to better serve the needs of the manufacturing industry. A *historian* is a type of database that is used to collect and store data from various sources in a factory.

holdout data

A portion of historical, labeled data that is withheld from a dataset that is used to train a machine learning model. You can use holdout data to evaluate the model performance by comparing the model predictions against the holdout data.

homogeneous database migration

Migrating your source database to a target database that shares the same database engine (for example, Microsoft SQL Server to Amazon RDS for SQL Server). Homogeneous migration is typically part of a rehosting or replatforming effort. You can use native database utilities to migrate the schema.

hot data

Data that is frequently accessed, such as real-time data or recent translational data. This data typically requires a high-performance storage tier or class to provide fast query responses.

hotfix

An urgent fix for a critical issue in a production environment. Due to its urgency, a hotfix is usually made outside of the typical DevOps release workflow.

hypercare period

Immediately following cutover, the period of time when a migration team manages and monitors the migrated applications in the cloud in order to address any issues. Typically, this period is 1–4 days in length. At the end of the hypercare period, the migration team typically transfers responsibility for the applications to the cloud operations team.

I

IaC

See infrastructure as code.

identity-based policy

A policy attached to one or more IAM principals that defines their permissions within the AWS Cloud environment.

idle application

An application that has an average CPU and memory usage between 5 and 20 percent over a period of 90 days. In a migration project, it is common to retire these applications or retain them on premises.

IIoT

See industrial Internet of Things.

immutable infrastructure

A model that deploys new infrastructure for production workloads instead of updating, patching, or modifying the existing infrastructure. Immutable infrastructures are inherently more consistent, reliable, and predictable than mutable infrastructure. For more information, see the Deploy using immutable infrastructure best practice in the AWS Well-Architected Framework.

inbound (ingress) VPC

In an AWS multi-account architecture, a VPC that accepts, inspects, and routes network connections from outside an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

incremental migration

A cutover strategy in which you migrate your application in small parts instead of performing a single, full cutover. For example, you might move only a few microservices or users to the new system initially. After you verify that everything is working properly, you can incrementally move additional microservices or users until you can decommission your legacy system. This strategy reduces the risks associated with large migrations.

Industry 4.0

A term that was introduced by Klaus Schwab in 2016 to refer to the modernization of manufacturing processes through advances in connectivity, real-time data, automation, analytics, and AI/ML.

infrastructure

All of the resources and assets contained within an application's environment.

infrastructure as code (IaC)

The process of provisioning and managing an application's infrastructure through a set of configuration files. IaC is designed to help you centralize infrastructure management, standardize resources, and scale quickly so that new environments are repeatable, reliable, and consistent.

industrial Internet of Things (IIoT)

The use of internet-connected sensors and devices in the industrial sectors, such as manufacturing, energy, automotive, healthcare, life sciences, and agriculture. For more information, see Building an industrial Internet of Things (IIoT) digital transformation strategy.

inspection VPC

In an AWS multi-account architecture, a centralized VPC that manages inspections of network traffic between VPCs (in the same or different AWS Regions), the internet, and on-premises networks. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

Internet of Things (IoT)

The network of connected physical objects with embedded sensors or processors that communicate with other devices and systems through the internet or over a local communication network. For more information, see What is IoT?

interpretability

A characteristic of a machine learning model that describes the degree to which a human can understand how the model's predictions depend on its inputs. For more information, see Machine learning model interpretability with AWS.

IoT

See Internet of Things.

IT information library (ITIL)

A set of best practices for delivering IT services and aligning these services with business requirements. ITIL provides the foundation for ITSM.

IT service management (ITSM)

Activities associated with designing, implementing, managing, and supporting IT services for an organization. For information about integrating cloud operations with ITSM tools, see the operations integration guide.

ITIL

See IT information library.

ITSM

See IT service management.

L

label-based access control (LBAC)

An implementation of mandatory access control (MAC) where the users and the data itself are each explicitly assigned a security label value. The intersection between the user security label and data security label determines which rows and columns can be seen by the user.

landing zone

A landing zone is a well-architected, multi-account AWS environment that is scalable and secure. This is a starting point from which your organizations can quickly launch and deploy workloads and applications with confidence in their security and infrastructure environment. For more information about landing zones, see Setting up a secure and scalable multi-account AWS environment.

large language model (LLM)

A deep learning AI model that is pretrained on a vast amount of data. An LLM can perform multiple tasks, such as answering questions, summarizing documents, translating text into other languages, and completing sentences. For more information, see What are LLMs.

large migration

A migration of 300 or more servers.

LBAC

See label-based access control.

least privilege

The security best practice of granting the minimum permissions required to perform a task. For more information, see Apply least-privilege permissions in the IAM documentation.

lift and shift

See 7 Rs.

little-endian system

A system that stores the least significant byte first. See also endianness.

LLM

See large language model.

lower environments

See environment.

M

machine learning (ML)

A type of artificial intelligence that uses algorithms and techniques for pattern recognition and learning. ML analyzes and learns from recorded data, such as Internet of Things (IoT) data, to generate a statistical model based on patterns. For more information, see Machine Learning.

main branch

See branch.

malware

Software that is designed to compromise computer security or privacy. Malware might disrupt computer systems, leak sensitive information, or gain unauthorized access. Examples of malware include viruses, worms, ransomware, Trojan horses, spyware, and keyloggers.

managed services

AWS services for which AWS operates the infrastructure layer, the operating system, and platforms, and you access the endpoints to store and retrieve data. Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB are examples of managed services. These are also known as *abstracted services*.

manufacturing execution system (MES)

A software system for tracking, monitoring, documenting, and controlling production processes that convert raw materials to finished products on the shop floor.

MAP

See Migration Acceleration Program.

mechanism

A complete process in which you create a tool, drive adoption of the tool, and then inspect the results in order to make adjustments. A mechanism is a cycle that reinforces and improves itself as it operates. For more information, see Building mechanisms in the AWS Well-Architected Framework.

member account

All AWS accounts other than the management account that are part of an organization in AWS Organizations. An account can be a member of only one organization at a time.

MES

See manufacturing execution system.

Message Queuing Telemetry Transport (MQTT)

A lightweight, machine-to-machine (M2M) communication protocol, based on the publish/subscribe pattern, for resource-constrained IoT devices.

microservice

A small, independent service that communicates over well-defined APIs and is typically owned by small, self-contained teams. For example, an insurance system might include microservices that map to business capabilities, such as sales or marketing, or subdomains, such as purchasing, claims, or analytics. The benefits of microservices include agility, flexible scaling, easy deployment, reusable code, and resilience. For more information, see Integrating microservices by using AWS serverless services.

microservices architecture

An approach to building an application with independent components that run each application process as a microservice. These microservices communicate through a well-defined interface by using lightweight APIs. Each microservice in this architecture can be updated, deployed, and scaled to meet demand for specific functions of an application. For more information, see Implementing microservices on AWS.

Migration Acceleration Program (MAP)

An AWS program that provides consulting support, training, and services to help organizations build a strong operational foundation for moving to the cloud, and to help offset the initial cost of migrations. MAP includes a migration methodology for executing legacy migrations in a methodical way and a set of tools to automate and accelerate common migration scenarios.

migration at scale

The process of moving the majority of the application portfolio to the cloud in waves, with more applications moved at a faster rate in each wave. This phase uses the best practices and lessons learned from the earlier phases to implement a *migration factory* of teams, tools, and processes to streamline the migration of workloads through automation and agile delivery. This is the third phase of the AWS migration strategy.

migration factory

Cross-functional teams that streamline the migration of workloads through automated, agile approaches. Migration factory teams typically include operations, business analysts and owners,

migration engineers, developers, and DevOps professionals working in sprints. Between 20 and 50 percent of an enterprise application portfolio consists of repeated patterns that can be optimized by a factory approach. For more information, see the discussion of migration factories and the Cloud Migration Factory guide in this content set.

migration metadata

The information about the application and server that is needed to complete the migration. Each migration pattern requires a different set of migration metadata. Examples of migration metadata include the target subnet, security group, and AWS account.

migration pattern

A repeatable migration task that details the migration strategy, the migration destination, and the migration application or service used. Example: Rehost migration to Amazon EC2 with AWS Application Migration Service.

Migration Portfolio Assessment (MPA)

An online tool that provides information for validating the business case for migrating to the AWS Cloud. MPA provides detailed portfolio assessment (server right-sizing, pricing, TCO comparisons, migration cost analysis) as well as migration planning (application data analysis and data collection, application grouping, migration prioritization, and wave planning). The MPA tool (requires login) is available free of charge to all AWS consultants and APN Partner consultants.

Migration Readiness Assessment (MRA)

The process of gaining insights about an organization's cloud readiness status, identifying strengths and weaknesses, and building an action plan to close identified gaps, using the AWS CAF. For more information, see the migration readiness guide. MRA is the first phase of the AWS migration strategy.

migration strategy

The approach used to migrate a workload to the AWS Cloud. For more information, see the 7 Rs entry in this glossary and see Mobilize your organization to accelerate large-scale migrations.

ML

See machine learning.

modernization

Transforming an outdated (legacy or monolithic) application and its infrastructure into an agile, elastic, and highly available system in the cloud to reduce costs, gain efficiencies, and take advantage of innovations. For more information, see Strategy for modernizing applications in the AWS Cloud.

modernization readiness assessment

An evaluation that helps determine the modernization readiness of an organization's applications; identifies benefits, risks, and dependencies; and determines how well the organization can support the future state of those applications. The outcome of the assessment is a blueprint of the target architecture, a roadmap that details development phases and milestones for the modernization process, and an action plan for addressing identified gaps. For more information, see Evaluating modernization readiness for applications in the AWS Cloud.

monolithic applications (monoliths)

Applications that run as a single service with tightly coupled processes. Monolithic applications have several drawbacks. If one application feature experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application's features also becomes more complex when the code base grows. To address these issues, you can use a microservices architecture. For more information, see Decomposing monoliths into microservices.

MPA

See Migration Portfolio Assessment.

MQTT

See Message Queuing Telemetry Transport.

multiclass classification

A process that helps generate predictions for multiple classes (predicting one of more than two outcomes). For example, an ML model might ask "Is this product a book, car, or phone?" or "Which product category is most interesting to this customer?"

mutable infrastructure

A model that updates and modifies the existing infrastructure for production workloads. For improved consistency, reliability, and predictability, the AWS Well-Architected Framework recommends the use of immutable infrastructure as a best practice.

O

OAC

See origin access control.

OAI

See origin access identity.

OCM

See organizational change management.

offline migration

A migration method in which the source workload is taken down during the migration process. This method involves extended downtime and is typically used for small, non-critical workloads.

OI

See operations integration.

OLA

See operational-level agreement.

online migration

A migration method in which the source workload is copied to the target system without being taken offline. Applications that are connected to the workload can continue to function during the migration. This method involves zero to minimal downtime and is typically used for critical production workloads.

OPC-UA

See Open Process Communications - Unified Architecture.

Open Process Communications - Unified Architecture (OPC-UA)

A machine-to-machine (M2M) communication protocol for industrial automation. OPC-UA provides an interoperability standard with data encryption, authentication, and authorization schemes.

operational-level agreement (OLA)

An agreement that clarifies what functional IT groups promise to deliver to each other, to support a service-level agreement (SLA).

operational readiness review (ORR)

A checklist of questions and associated best practices that help you understand, evaluate, prevent, or reduce the scope of incidents and possible failures. For more information, see Operational Readiness Reviews (ORR) in the AWS Well-Architected Framework.

operational technology (OT)

Hardware and software systems that work with the physical environment to control industrial operations, equipment, and infrastructure. In manufacturing, the integration of OT and information technology (IT) systems is a key focus for Industry 4.0 transformations.

operations integration (OI)

The process of modernizing operations in the cloud, which involves readiness planning, automation, and integration. For more information, see the operations integration guide.

organization trail

A trail that's created by AWS CloudTrail that logs all events for all AWS accounts in an organization in AWS Organizations. This trail is created in each AWS account that's part of the organization and tracks the activity in each account. For more information, see Creating a trail for an organization in the CloudTrail documentation.

organizational change management (OCM)

A framework for managing major, disruptive business transformations from a people, culture, and leadership perspective. OCM helps organizations prepare for, and transition to, new systems and strategies by accelerating change adoption, addressing transitional issues, and driving cultural and organizational changes. In the AWS migration strategy, this framework is called *people acceleration*, because of the speed of change required in cloud adoption projects. For more information, see the OCM guide.

origin access control (OAC)

In CloudFront, an enhanced option for restricting access to secure your Amazon Simple Storage Service (Amazon S3) content. OAC supports all S3 buckets in all AWS Regions, server-side encryption with AWS KMS (SSE-KMS), and dynamic PUT and DELETE requests to the S3 bucket.

origin access identity (OAI)

In CloudFront, an option for restricting access to secure your Amazon S3 content. When you use OAI, CloudFront creates a principal that Amazon S3 can authenticate with. Authenticated principals can access content in an S3 bucket only through a specific CloudFront distribution. See also OAC, which provides more granular and enhanced access control.

ORR

See operational readiness review.

OT

See operational technology.

outbound (egress) VPC

In an AWS multi-account architecture, a VPC that handles network connections that are initiated from within an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

P

permissions boundary

An IAM management policy that is attached to IAM principals to set the maximum permissions that the user or role can have. For more information, see Permissions boundaries in the IAM documentation.

personally identifiable information (PII)

Information that, when viewed directly or paired with other related data, can be used to reasonably infer the identity of an individual. Examples of PII include names, addresses, and contact information.

PII

See personally identifiable information.

playbook

A set of predefined steps that capture the work associated with migrations, such as delivering core operations functions in the cloud. A playbook can take the form of scripts, automated runbooks, or a summary of processes or steps required to operate your modernized environment.

PLC

See programmable logic controller.

PLM

See product lifecycle management.

policy

An object that can define permissions (see identity-based policy), specify access conditions (see resource-based policy), or define the maximum permissions for all accounts in an organization in AWS Organizations (see service control policy).

polyglot persistence

Independently choosing a microservice's data storage technology based on data access patterns and other requirements. If your microservices have the same data storage technology, they can encounter implementation challenges or experience poor performance. Microservices are more easily implemented and achieve better performance and scalability if they use the data store best adapted to their requirements. For more information, see Enabling data persistence in microservices.

portfolio assessment

A process of discovering, analyzing, and prioritizing the application portfolio in order to plan the migration. For more information, see Evaluating migration readiness.

predicate

A query condition that returns `true` or `false`, commonly located in a `WHERE` clause.

predicate pushdown

A database query optimization technique that filters the data in the query before transfer. This reduces the amount of data that must be retrieved and processed from the relational database, and it improves query performance.

preventative control

A security control that is designed to prevent an event from occurring. These controls are a first line of defense to help prevent unauthorized access or unwanted changes to your network. For more information, see Preventative controls in *Implementing security controls on AWS*.

principal

An entity in AWS that can perform actions and access resources. This entity is typically a root user for an AWS account, an IAM role, or a user. For more information, see *Principal* in Roles terms and concepts in the IAM documentation.

privacy by design

A system engineering approach that takes privacy into account through the whole development process.

private hosted zones

A container that holds information about how you want Amazon Route 53 to respond to DNS queries for a domain and its subdomains within one or more VPCs. For more information, see Working with private hosted zones in the Route 53 documentation.

proactive control

A security control designed to prevent the deployment of noncompliant resources. These controls scan resources before they are provisioned. If the resource is not compliant with the control, then it isn't provisioned. For more information, see the Controls reference guide in the AWS Control Tower documentation and see Proactive controls in *Implementing security controls on AWS*.

product lifecycle management (PLM)

The management of data and processes for a product throughout its entire lifecycle, from design, development, and launch, through growth and maturity, to decline and removal.

production environment

See environment.

programmable logic controller (PLC)

In manufacturing, a highly reliable, adaptable computer that monitors machines and automates manufacturing processes.

prompt chaining

Using the output of one LLM prompt as the input for the next prompt to generate better responses. This technique is used to break down a complex task into subtasks, or to iteratively refine or expand a preliminary response. It helps improve the accuracy and relevance of a model's responses and allows for more granular, personalized results.

pseudonymization

The process of replacing personal identifiers in a dataset with placeholder values. Pseudonymization can help protect personal privacy. Pseudonymized data is still considered to be personal data.

publish/subscribe (pub/sub)

A pattern that enables asynchronous communications among microservices to improve scalability and responsiveness. For example, in a microservices-based MES, a microservice can publish event messages to a channel that other microservices can subscribe to. The system can add new microservices without changing the publishing service.

Q

query plan

A series of steps, like instructions, that are used to access the data in a SQL relational database system.

query plan regression

When a database service optimizer chooses a less optimal plan than it did before a given change to the database environment. This can be caused by changes to statistics, constraints, environment settings, query parameter bindings, and updates to the database engine.

R

RACI matrix

See responsible, accountable, consulted, informed (RACI).

RAG

See Retrieval Augmented Generation.

ransomware

A malicious software that is designed to block access to a computer system or data until a payment is made.

RASCI matrix

See responsible, accountable, consulted, informed (RACI).

RCAC

See row and column access control.

read replica

A copy of a database that's used for read-only purposes. You can route queries to the read replica to reduce the load on your primary database.

re-architect

See 7 Rs.

recovery point objective (RPO)

The maximum acceptable amount of time since the last data recovery point. This determines what is considered an acceptable loss of data between the last recovery point and the interruption of service.

recovery time objective (RTO)

The maximum acceptable delay between the interruption of service and restoration of service.

refactor

See 7 Rs.

Region

A collection of AWS resources in a geographic area. Each AWS Region is isolated and independent of the others to provide fault tolerance, stability, and resilience. For more information, see Specify which AWS Regions your account can use.

regression

An ML technique that predicts a numeric value. For example, to solve the problem of "What price will this house sell for?" an ML model could use a linear regression model to predict a house's sale price based on known facts about the house (for example, the square footage).

rehost

See 7 Rs.

release

In a deployment process, the act of promoting changes to a production environment.

relocate

See 7 Rs.

replatform

See 7 Rs.

repurchase

See 7 Rs.

resiliency

An application's ability to resist or recover from disruptions. High availability and disaster recovery are common considerations when planning for resiliency in the AWS Cloud. For more information, see AWS Cloud Resilience.

resource-based policy

A policy attached to a resource, such as an Amazon S3 bucket, an endpoint, or an encryption key. This type of policy specifies which principals are allowed access, supported actions, and any other conditions that must be met.

responsible, accountable, consulted, informed (RACI) matrix

A matrix that defines the roles and responsibilities for all parties involved in migration activities and cloud operations. The matrix name is derived from the responsibility types defined in the matrix: responsible (R), accountable (A), consulted (C), and informed (I). The support (S) type is optional. If you include support, the matrix is called a *RASCI matrix*, and if you exclude it, it's called a *RACI matrix*.

responsive control

A security control that is designed to drive remediation of adverse events or deviations from your security baseline. For more information, see Responsive controls in *Implementing security controls on AWS*.

retain

See 7 Rs.

retire

See 7 Rs.

Retrieval Augmented Generation (RAG)

A generative AI technology in which an LLM references an authoritative data source that is outside of its training data sources before generating a response. For example, a RAG model might perform a semantic search of an organization's knowledge base or custom data. For more information, see What is RAG.

rotation

The process of periodically updating a secret to make it more difficult for an attacker to access the credentials.

row and column access control (RCAC)

The use of basic, flexible SQL expressions that have defined access rules. RCAC consists of row permissions and column masks.

RPO

See recovery point objective.

RTO

See recovery time objective.

runbook

A set of manual or automated procedures required to perform a specific task. These are typically built to streamline repetitive operations or procedures with high error rates.

S

SAML 2.0

An open standard that many identity providers (IdPs) use. This feature enables federated single sign-on (SSO), so users can log into the AWS Management Console or call the AWS API operations without you having to create user in IAM for everyone in your organization. For more information about SAML 2.0-based federation, see About SAML 2.0-based federation in the IAM documentation.

SCADA

See supervisory control and data acquisition.

SCP

See service control policy.

secret

In AWS Secrets Manager, confidential or restricted information, such as a password or user credentials, that you store in encrypted form. It consists of the secret value and its metadata.

The secret value can be binary, a single string, or multiple strings. For more information, see What's in a Secrets Manager secret? in the Secrets Manager documentation.

security by design

A system engineering approach that takes security into account through the whole development process.

security control

A technical or administrative guardrail that prevents, detects, or reduces the ability of a threat actor to exploit a security vulnerability. There are four primary types of security controls: preventative, detective, responsive, and proactive.

security hardening

The process of reducing the attack surface to make it more resistant to attacks. This can include actions such as removing resources that are no longer needed, implementing the security best practice of granting least privilege, or deactivating unnecessary features in configuration files.

security information and event management (SIEM) system

Tools and services that combine security information management (SIM) and security event management (SEM) systems. A SIEM system collects, monitors, and analyzes data from servers, networks, devices, and other sources to detect threats and security breaches, and to generate alerts.

security response automation

A predefined and programmed action that is designed to automatically respond to or remediate a security event. These automations serve as detective or responsive security controls that help you implement AWS security best practices. Examples of automated response actions include modifying a VPC security group, patching an Amazon EC2 instance, or rotating credentials.

server-side encryption

Encryption of data at its destination, by the AWS service that receives it.

service control policy (SCP)

A policy that provides centralized control over permissions for all accounts in an organization in AWS Organizations. SCPs define guardrails or set limits on actions that an administrator can delegate to users or roles. You can use SCPs as allow lists or deny lists, to specify which services or actions are permitted or prohibited. For more information, see Service control policies in the AWS Organizations documentation.

service endpoint

The URL of the entry point for an AWS service. You can use the endpoint to connect programmatically to the target service. For more information, see AWS service endpoints in *AWS General Reference*.

service-level agreement (SLA)

An agreement that clarifies what an IT team promises to deliver to their customers, such as service uptime and performance.

service-level indicator (SLI)

A measurement of a performance aspect of a service, such as its error rate, availability, or throughput.

service-level objective (SLO)

A target metric that represents the health of a service, as measured by a service-level indicator.

shared responsibility model

A model describing the responsibility you share with AWS for cloud security and compliance. AWS is responsible for security *of* the cloud, whereas you are responsible for security *in* the cloud. For more information, see Shared responsibility model.

SIEM

See security information and event management system.

single point of failure (SPOF)

A failure in a single, critical component of an application that can disrupt the system.

SLA

See service-level agreement.

SLI

See service-level indicator.

SLO

See service-level objective.

split-and-seed model

A pattern for scaling and accelerating modernization projects. As new features and product releases are defined, the core team splits up to create new product teams. This helps scale your

organization's capabilities and services, improves developer productivity, and supports rapid innovation. For more information, see Phased approach to modernizing applications in the AWS Cloud.

SPOF

See single point of failure.

star schema

A database organizational structure that uses one large fact table to store transactional or measured data and uses one or more smaller dimensional tables to store data attributes. This structure is designed for use in a data warehouse or for business intelligence purposes.

strangler fig pattern

An approach to modernizing monolithic systems by incrementally rewriting and replacing system functionality until the legacy system can be decommissioned. This pattern uses the analogy of a fig vine that grows into an established tree and eventually overcomes and replaces its host. The pattern was introduced by Martin Fowler as a way to manage risk when rewriting monolithic systems. For an example of how to apply this pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

subnet

A range of IP addresses in your VPC. A subnet must reside in a single Availability Zone.

supervisory control and data acquisition (SCADA)

In manufacturing, a system that uses hardware and software to monitor physical assets and production operations.

symmetric encryption

An encryption algorithm that uses the same key to encrypt and decrypt the data.

synthetic testing

Testing a system in a way that simulates user interactions to detect potential issues or to monitor performance. You can use Amazon CloudWatch Synthetics to create these tests.

system prompt

A technique for providing context, instructions, or guidelines to an LLM to direct its behavior. System prompts help set context and establish rules for interactions with users.

T

tags

Key-value pairs that act as metadata for organizing your AWS resources. Tags can help you manage, identify, organize, search for, and filter resources. For more information, see Tagging your AWS resources.

target variable

The value that you are trying to predict in supervised ML. This is also referred to as an *outcome variable*. For example, in a manufacturing setting the target variable could be a product defect.

task list

A tool that is used to track progress through a runbook. A task list contains an overview of the runbook and a list of general tasks to be completed. For each general task, it includes the estimated amount of time required, the owner, and the progress.

test environment

See environment.

training

To provide data for your ML model to learn from. The training data must contain the correct answer. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict). It outputs an ML model that captures these patterns. You can then use the ML model to make predictions on new data for which you don't know the target.

transit gateway

A network transit hub that you can use to interconnect your VPCs and on-premises networks. For more information, see What is a transit gateway in the AWS Transit Gateway documentation.

trunk-based workflow

An approach in which developers build and test features locally in a feature branch and then merge those changes into the main branch. The main branch is then built to the development, preproduction, and production environments, sequentially.

trusted access

Granting permissions to a service that you specify to perform tasks in your organization in AWS Organizations and in its accounts on your behalf. The trusted service creates a service-linked role in each account, when that role is needed, to perform management tasks for you. For more information, see Using AWS Organizations with other AWS services in the AWS Organizations documentation.

tuning

To change aspects of your training process to improve the ML model's accuracy. For example, you can train the ML model by generating a labeling set, adding labels, and then repeating these steps several times under different settings to optimize the model.

two-pizza team

A small DevOps team that you can feed with two pizzas. A two-pizza team size ensures the best possible opportunity for collaboration in software development.

U

uncertainty

A concept that refers to imprecise, incomplete, or unknown information that can undermine the reliability of predictive ML models. There are two types of uncertainty: *Epistemic uncertainty* is caused by limited, incomplete data, whereas *aleatoric uncertainty* is caused by the noise and randomness inherent in the data. For more information, see the Quantifying uncertainty in deep learning systems guide.

undifferentiated tasks

Also known as *heavy lifting*, work that is necessary to create and operate an application but that doesn't provide direct value to the end user or provide competitive advantage. Examples of undifferentiated tasks include procurement, maintenance, and capacity planning.

upper environments

See environment.

V

vacuuming

A database maintenance operation that involves cleaning up after incremental updates to reclaim storage and improve performance.

version control

Processes and tools that track changes, such as changes to source code in a repository.

VPC peering

A connection between two VPCs that allows you to route traffic by using private IP addresses. For more information, see What is VPC peering in the Amazon VPC documentation.

vulnerability

A software or hardware flaw that compromises the security of the system.

W

warm cache

A buffer cache that contains current, relevant data that is frequently accessed. The database instance can read from the buffer cache, which is faster than reading from the main memory or disk.

warm data

Data that is infrequently accessed. When querying this kind of data, moderately slow queries are typically acceptable.

window function

A SQL function that performs a calculation on a group of rows that relate in some way to the current record. Window functions are useful for processing tasks, such as calculating a moving average or accessing the value of rows based on the relative position of the current row.

workload

A collection of resources and code that delivers business value, such as a customer-facing application or backend process.

workstream

Functional groups in a migration project that are responsible for a specific set of tasks. Each workstream is independent but supports the other workstreams in the project. For example, the portfolio workstream is responsible for prioritizing applications, wave planning, and collecting migration metadata. The portfolio workstream delivers these assets to the migration workstream, which then migrates the servers and applications.

WORM

See write once, read many.

WQF

See AWS Workload Qualification Framework.

write once, read many (WORM)

A storage model that writes data a single time and prevents the data from being deleted or modified. Authorized users can read the data as many times as needed, but they cannot change it. This data storage infrastructure is considered immutable.

Z

zero-day exploit

An attack, typically malware, that takes advantage of a zero-day vulnerability.

zero-day vulnerability

An unmitigated flaw or vulnerability in a production system. Threat actors can use this type of vulnerability to attack the system. Developers frequently become aware of the vulnerability as a result of the attack.

zero-shot prompting

Providing an LLM with instructions for performing a task but no examples (*shots*) that can help guide it. The LLM must use its pre-trained knowledge to handle the task. The effectiveness of zero-shot prompting depends on the complexity of the task and the quality of the prompt. See also few-shot prompting.

zombie application

An application that has an average CPU and memory usage below 5 percent. In a migration project, it is common to retire these applications.