central data TECHNOLOGY
a member of CTI Group

Amazon Bedrock: Building Generative AI Applications on AWS

Topics

What Is Amazon Bedrock?

Amazon Bedrock is a fully managed AWS service that provides access to high-performing foundation models from leading AI companies. It enables developers to build and scale generative AI applications through APIs without having to manage the underlying infrastructure used to host the foundation models. Foundation models are models that have been trained on large amounts of data and can be adapted to perform a wide range of tasks. Amazon Bedrock provides access to foundation models with different capabilities, including models that can work with text, images, embeddings, and other supported modalities.

Amazon Bedrock provides foundation models from multiple providers. Each model can differ in its capabilities, supported APIs, endpoints, Regions, and other characteristics. This allows developers to select a model based on the requirements of a particular application rather than being limited to a single model provider. Model selection should take into account factors such as the model's capabilities, API and endpoint compatibility, Region availability, cost, and throughput. These considerations are important because model capabilities and availability can vary across different models and AWS Regions.

a. Foundation Models

Foundation models provide the core intelligence used by generative AI applications. An application sends input to a selected model, and the model generates an output based on the input and the model's supported capabilities and configuration. The capabilities available to an application depend on the selected foundation model. AWS provides model-specific information covering supported features, APIs, endpoints, and Regions so developers can determine which models are appropriate for their workloads. Amazon Bedrock currently provides access to models from multiple providers, including Amazon and other AI companies. The available model catalog and endpoint support can change over time, so developers should refer to the current AWS documentation when selecting models for production workloads.

b. Managed Model Access

Amazon Bedrock provides managed access to foundation models through AWS endpoints. For new applications, AWS recommends the bedrock-runtime endpoint. Amazon Bedrock also provides the bedrock-mantle endpoint, which supports additional APIs and capabilities. The bedrock-runtime endpoint supports the Bedrock-native InvokeModel and Converse APIs, as well as OpenAI-compatible Responses and Chat Completions APIs and the Anthropic Messages API. The bedrock-mantle endpoint supports Responses, Chat Completions, and Messages APIs. Amazon Bedrock extends beyond direct model inference by providing capabilities for knowledge retrieval, safeguards, model evaluation, and model customization. These capabilities can be incorporated into applications according to their technical and business requirements.

Amazon Bedrock Architecture

A generative AI application built with Amazon Bedrock can be viewed as several interconnected components. An application communicates with Amazon Bedrock through an inference endpoint and API, which connects the application to a selected foundation model. Additional capabilities such as Knowledge Bases and Guardrails can be added when an application requires enterprise data retrieval or additional controls. At the inference layer, Amazon Bedrock provides different APIs for interacting with foundation models. The choice of API depends on the application's requirements, the desired interaction pattern, and the capabilities supported by the selected model.

a. Application Layer

The application layer represents the software or business application that uses generative AI capabilities. Applications can use Amazon Bedrock for conversational interactions, content generation, information retrieval, summarization, and other workloads supported by the selected foundation model. The application communicates with Amazon Bedrock through supported APIs. Before making an inference request, the application's IAM role or user must have the permissions required to perform the relevant model invocation operations.

b. API and Endpoint Layer

Amazon Bedrock currently supports two inference endpoints: bedrock-runtime and bedrock-mantle. AWS recommends bedrock-runtime for new applications. Each endpoint provides a different set of supported APIs and capabilities. The bedrock-runtime endpoint supports InvokeModel, Converse, Chat Completions, Responses, and Messages APIs. The endpoint also supports capabilities such as Guardrails and cross-Region inference.

The bedrock-mantle endpoint provides Responses, Chat Completions, and Messages APIs and supports capabilities such as asynchronous inference and server-side tool use for supported workloads. The two endpoints can also be used together within the same application when different use cases require different capabilities.

c. Foundation Model Layer

The foundation model layer contains the model that processes the application's input and produces an output. Amazon Bedrock supports multiple foundation models, and each model can have different capabilities, APIs, and availability. Because model capabilities vary, developers should verify whether a selected model supports the API, modality, Region, and inference features required by the application. AWS provides endpoint availability information to help developers determine model compatibility.

d. Additional Bedrock Capabilities

Amazon Bedrock provides capabilities beyond direct foundation model inference. These include Knowledge Bases for retrieving information from data sources, Guardrails for applying configurable safeguards, model evaluation for assessing model performance, and model

customization for supported models. These capabilities can be combined to support more complete generative AI applications. For example, an enterprise knowledge assistant can use a foundation model together with a Knowledge Base, while an application that requires content controls can apply Guardrails to user inputs and model responses.

Foundation Models and Query Processing

In Amazon Bedrock, model inference is the process of generating an output from an input provided to a model. Applications can send inference requests through supported APIs, with the required permissions depending on the endpoint and API being used. Amazon Bedrock provides several API patterns for inference. The Converse API provides a unified interface for interacting with supported models, while InvokeModel provides more direct access to model-specific request and response formats.

a. Converse API

The Converse API provides a standardized interface for conversational interactions across supported foundation models in Amazon Bedrock. This allows developers to use a common API structure when working with different supported models. The Converse API can support multi-turn conversations and standardized interactions while still allowing applications to work with models from different providers. Developers should verify the supported capabilities of their selected model before implementation.

b. InvokeModel API

The InvokeModel API provides direct access to supported foundation models through their model-specific request and response formats. This approach provides greater control over the request structure compared with the standardized Converse API. Applications using InvokeModel must provide the appropriate permissions and specify a supported model or inference profile. The exact request format depends on the selected model.

c. Model Evaluation

Model evaluation allows developers to assess and compare foundation models against defined evaluation tasks. Amazon Bedrock supports automatic model evaluation using built-in or custom prompt datasets. Amazon Bedrock also supports evaluation approaches in which an LLM acts as a judge to evaluate responses generated by another model. Evaluation results can provide scores and other information that can help developers assess model performance against their application requirements.

d. Model Customization

Model customization involves providing training data to improve a foundation model's performance for specific use cases. Amazon Bedrock supports customization methods such as supervised fine-tuning and other supported approaches, depending on the model. With supervised fine-tuning, labeled training examples are used to train a model for specific tasks. The model learns from the provided examples and adjusts its parameters to improve performance for the represented tasks. AWS also provides other model customization approaches, with availability depending on the selected foundation model. Developers should therefore check the current customization capabilities and requirements for the model they intend to customize.

Data Integration and Knowledge Bases

Enterprise generative AI applications often need to work with information that is not contained within the general knowledge of a foundation model. Amazon Bedrock Knowledge Bases provides a way to connect applications with data sources and retrieve relevant information that can be used to generate responses.

Knowledge Bases support Retrieval-Augmented Generation (RAG), a technique that retrieves information from a data store and uses that information to augment responses generated by a large language model. This enables applications to incorporate proprietary or enterprise information into generative AI workflows.

a. Retrieval-Augmented Generation

RAG combines information retrieval with generative AI. Instead of relying only on information contained within the foundation model, the application retrieves relevant information from connected data sources and uses that information as context for generating a response. Amazon Bedrock Knowledge Bases provides managed capabilities for implementing RAG. AWS

describes it as an out-of-the-box RAG solution that abstracts much of the work involved in building retrieval pipelines.

b. Knowledge Base Retrieval

After a Knowledge Base has been configured, an application can query its data sources using Amazon Bedrock APIs. The Retrieve operation returns source chunks or images that are relevant to a query. The RetrieveAndGenerate operation combines retrieval with model invocation. It retrieves relevant source information and generates a natural-language response based on the retrieved results. The response can also include citations to the source chunks used during retrieval. Amazon Bedrock Knowledge Bases also supports reranking models for Retrieve and RetrieveAndGenerate. Reranking can be used to reorder retrieved information based on relevance before it is used in the generation process.

c. Data Sources and Vector Stores

A Knowledge Base can use data sources to provide information for retrieval. Amazon Bedrock processes data from the configured sources so that relevant information can be retrieved when an application sends a query. Knowledge Bases use vector representations of data to support semantic retrieval. The architecture includes data sources, embedding models, vector stores, and the foundation model used to generate responses.

d. Structured Data

Amazon Bedrock Knowledge Bases can also work with structured data. The GenerateQuery operation converts a natural-language query into a query suitable for a structured data store. This capability can allow users to interact with structured business information using natural-language questions rather than manually constructing database queries. The specific supported data sources and configuration requirements depend on the Knowledge Base implementation.

central data
TECHNOLOGY
a member of CTI Group

Security and Data Access

Security in Amazon Bedrock follows the AWS Shared Responsibility Model. AWS is responsible for security of the cloud, while customers remain responsible for security in the cloud based on factors such as data sensitivity, organizational requirements, and applicable laws and regulations.Customers are responsible for configuring their Amazon Bedrock resources and applications according to their security requirements. AWS provides security capabilities and documentation that can be used to protect and monitor Amazon Bedrock resources.

a. Identity and Access Management

Applications interacting with Amazon Bedrock require appropriate permissions to perform model inference and other operations. AWS Identity and Access Management (IAM) can be used to control access to Amazon Bedrock APIs and resources. Before an inference request can be made, the IAM role used by the application must have permissions to perform the required model invocation actions. The specific permissions depend on the endpoint and APIs being used.

b. Infrastructure Security

Amazon Bedrock is a managed AWS service protected by the AWS global network security infrastructure. Access to Amazon Bedrock uses published AWS APIs and requires secure network communication using TLS. Requests to Amazon Bedrock must be authenticated using AWS credentials associated with an IAM principal or temporary credentials generated through AWS Security Token Service.

c. Amazon Bedrock Guardrails

Amazon Bedrock Guardrails provides configurable safeguards for generative AI applications. It can evaluate user inputs and model responses and help detect and filter undesirable content or protect sensitive information. Guardrails provides configurable filters including content filters, denied topics, sensitive information filters, word filters, and other supported safeguards.

Content filters can detect categories such as hate, insults, sexual content, violence, misconduct, and prompt attacks. Guardrails can be applied to model inference and can also be integrated with other Amazon Bedrock capabilities. This allows organizations to apply consistent safeguards to different generative AI application workflows.

d. Guardrail Evaluation

Amazon Bedrock Guardrails evaluates both user inputs and model responses. Depending on the configured policies, Guardrails can intervene when content violates the configured requirements. Amazon Bedrock also provides the ApplyGuardrail API, which allows developers to evaluate content independently from foundation model invocation. This enables Guardrails to be incorporated into different points within an application workflow.

Scalability and Performance Management

Generative AI applications can have different throughput and latency requirements depending on their workload. Amazon Bedrock provides multiple inference options that can be considered based on application requirements, including On-Demand inference, cross-Region inference, and Provisioned Throughput. The appropriate inference approach depends on factors such as expected request volume, throughput requirements, model availability, and workload characteristics. Developers should evaluate these factors when designing applications for production.

a. On-Demand Inference

On-Demand inference allows applications to invoke supported foundation models without purchasing Provisioned Throughput. The models available for On-Demand inference depend on the model, endpoint, and AWS Region. On-Demand inference can be appropriate for workloads where applications need access to supported foundation models without provisioning a fixed level of throughput. Applications are still subject to the applicable Amazon Bedrock quotas and model availability.

b. Cross-Region Inference

Amazon Bedrock supports cross-Region inference for supported models through inference profiles. Cross-Region inference can route requests across multiple AWS Regions according to the applicable inference profile. Cross-Region inference can be used to improve access to available model capacity across supported Regions. The models and Regions available for cross-Region inference vary, so developers should verify the current endpoint and model availability before implementation.

c. Provisioned Throughput

Provisioned Throughput allows customers to purchase a higher level of throughput for a model at a fixed cost. Throughput refers to the number and rate of inputs and outputs that a model processes and returns. Provisioned Throughput is billed hourly. The hourly price depends on factors including the selected model, number of Model Units, and commitment duration. AWS provides no-commitment, one-month, and six-month commitment options. A Model Unit provides a defined throughput level for a particular model, including the number of input tokens it can process and output tokens it can generate within a specified period.

d. Performance Considerations

Performance should be evaluated according to the requirements of the application. Factors such as model selection, inference endpoint, API, throughput requirements, quotas, and request characteristics can influence how an application performs in production. Amazon Bedrock provides endpoint-specific quotas and guidance for managing throughput. Developers can use these capabilities and recommendations when preparing applications for workloads with increasing or variable request volumes.

Cost Model and Cost Efficiency

The cost of an Amazon Bedrock implementation depends on the models and capabilities used by the application. Model selection should consider cost alongside capabilities, API and endpoint compatibility, Region availability, and throughput requirements. Different Amazon Bedrock capabilities can introduce different cost considerations. Applications may incur costs associated with model inference, Guardrails, Knowledge Bases, model customization, model evaluation, and Provisioned Throughput depending on the architecture.

a. Model Inference Cost

Inference costs can depend on the amount and type of tokens processed by a model. AWS Cost and Usage Reports for Amazon Bedrock include categories such as input tokens, output tokens, cache read tokens, and cache write tokens for supported usage. Input tokens represent tokens processed from requests, while output tokens represent tokens generated by the model. Applications using supported prompt caching capabilities can also have cache-related token usage. Because different foundation models can have different pricing, selecting the appropriate model can influence the overall cost of an application's inference workload. Developers should compare the model's capabilities and expected usage against its current pricing.

b. Provisioned Throughput Cost

Provisioned Throughput is billed hourly, with pricing determined by factors such as the selected model, number of Model Units, and commitment duration. Longer commitment periods can provide discounted hourly pricing. Billing for Provisioned Throughput continues until the provisioned capacity is deleted or the applicable commitment ends. This makes workload planning important when deciding whether Provisioned Throughput is appropriate for an application.

c. Guardrails Cost

Amazon Bedrock Guardrails introduces charges based on the policies configured for a Guardrail. AWS documents different policy types and how Guardrail evaluation is charged. If Guardrails blocks an input before foundation model inference is performed, the Guardrail evaluation is charged while the discarded foundation model inference is not charged. If the model response is blocked after inference, the relevant model inference and Guardrail evaluation charges can apply.

d. Model Customization Cost

Model customization can introduce costs associated with training and model storage. The specific cost depends on the customization method, model, training workload, and other applicable pricing factors. Customized models can also have additional inference requirements. AWS

provides different options for running inference with custom models, including Provisioned Throughput and, for supported custom model deployments, on-demand inference.

e. Cost Optimization

Cost optimization should begin with choosing a model that provides the capabilities required by the application. Selecting a model with appropriate capabilities, availability, throughput, and pricing can help align the AI architecture with workload requirements. Businesses should also evaluate the cost impact of additional services and capabilities used in the application. Knowledge Bases, Guardrails, model customization, evaluation, and Provisioned Throughput can all become part of the overall cost model depending on the application's architecture.

Conclusion

Amazon Bedrock provides a managed platform for building generative AI applications with foundation models from multiple providers. Its capabilities extend beyond direct inference to include knowledge retrieval, safeguards, model evaluation, customization, and different approaches for managing inference capacity.

The architecture of an Amazon Bedrock application can be designed according to the requirements of the workload. Applications can start with direct foundation model inference and expand with Knowledge Bases when enterprise data is required, Guardrails when additional controls are needed, or customization when supported models need to be adapted for specific use cases.

For enterprise applications, data integration is an important consideration. Amazon Bedrock Knowledge Bases enables RAG-based applications to retrieve relevant information from connected data sources and use that information to generate responses, while also providing source citations for retrieved content.

Security should be incorporated throughout the architecture. Amazon Bedrock follows the AWS Shared Responsibility Model, while IAM, infrastructure security, and Guardrails provide mechanisms for controlling access, protecting infrastructure, and applying configurable safeguards to generative AI interactions.

Scalability and cost should also be evaluated based on the characteristics of each workload. Amazon Bedrock provides On-Demand inference, cross-Region inference, and Provisioned

Throughput, allowing businesses to select an approach according to their throughput and application requirements.

Ultimately, building a production-ready generative AI application requires consideration of more than the foundation model itself. Model capability, application architecture, enterprise data, security, performance, scalability, and cost all need to be evaluated together. Amazon Bedrock provides managed capabilities that allow developers and businesses to address these requirements as they build and scale generative AI applications on AWS.

Start Your Generative AI Journey with Central Data Technology

From exploring use cases to building production-ready solutions, CDT can help your business get more value from AWS and Generative AI. Discover More with CDT by click this **link!**

Credit to: AWS Documents