What is 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. You can choose from a wide range of foundation models to find the model that is best suited for your use case. Amazon Bedrock also offers a broad set of capabilities to build generative AI applications with security, privacy, and responsible AI. Using Amazon Bedrock, you can easily experiment with and evaluate top foundation models for your use cases, privately customize them with your data using techniques such as fine-tuning and Retrieval Augmented Generation (RAG), and build agents that execute tasks using your enterprise systems and data sources.

With Amazon Bedrock's serverless experience, you can get started quickly, privately customize foundation models with your own data, and easily and securely integrate and deploy them into your applications using AWS tools without having to manage any infrastructure.

Take advantage of Amazon Bedrock foundation models to explore the following capabilities. To see feature limitations by Region, see Model support by AWS Region.

Experiment with prompts and configurations – Run model inference by sending prompts using different configurations and foundation models to generate responses. You can use the API or the text, image, and chat playgrounds in the console to experiment in a graphical interface. When you're ready, set up your application to make requests to the InvokeModel APIs.

Augment response generation with information from your data sources – Create knowledge bases by uploading data sources to be queried in order to augment a foundation model's generation of responses.

Create applications that reason through how to help a customer – Build agents that use foundation models, make API calls, and (optionally) query knowledge bases in order to reason through and carry out tasks for your customers.

Adapt models to specific tasks and domains with training data – Customize an Amazon Bedrock foundation model by providing training data for fine-tuning or continued-pretraining in order to adjust a model's parameters and improve its performance on specific tasks or in certain domains.

Improve your FM-based application's efficiency and output – Purchase Provisioned Throughput for a foundation model in order to run inference on models more efficiently and at discounted rates.

Determine the best model for your use case – Evaluate outputs of different models with built-in or custom prompt datasets to determine the model that is best suited for your application.

Key Definitions

This chapter provides definitions for concepts that will help you understand what Amazon Bedrock offers and how it works. If you are a first-time user, you should first read through the basic concepts. Once you familiarize yourself with the basics of Amazon Bedrock, we recommend for you to explore the advanced concepts and features that Amazon Bedrock has to offer.

Basic concepts

The following list introduces you to the basic concepts of generative AI and Amazon Bedrock's fundamental capabilities.

Foundation model (FM) – An AI model with a large number of parameters and trained on a massive amount of diverse data. A foundation model can generate a variety of responses for a wide range of use cases. Foundation models can generate text or image, and can also convert input into embeddings. Before you can use an Amazon Bedrock foundation model, you must request access. For more information about foundation models, see Supported foundation models in Amazon Bedrock.

Base model – A foundation model that is packaged by a provider and ready to use. Amazon Bedrock offers a variety of industry-leading foundation models from leading providers. For more information, see Supported foundation models in Amazon Bedrock.

Model inference – The process of a foundation model generating an output (response) from a given input (prompt). For more information, see Run model inference.

Prompt – An input provided to a model to guide it to generate an appropriate response or output for the input. For example, a text prompt can consist of a single line for the model to respond to, or it can detail instructions or a task for the model to perform. The prompt can contain the context of the task, examples of outputs, or text for a model to use in its response. Prompts can be used to carry out tasks such as classification, question answering, code generation, creative writing, and more. For more information, see Prompt engineering guidelines.

Token – A sequence of characters that a model can interpret or predict as a single unit of meaning. For example, with text models, a token could correspond not just to a word, but also to a part of a word with grammatical meaning (such as "-ed"), a punctuation mark (such as "?"), or a common phrase (such as "a lot").

Model parameters – Values that define a model and its behavior in interpreting input and generating responses. Model parameters are controlled and updated by providers. You can also update model parameters to create a new model through the process of model customization.

Inference parameters – Values that can be adjusted during model inference to influence a response. Inference parameters can affect how varied responses are and can also limit the length of a response or the occurrence of specified sequences. For more information and definitions of specific inference parameters, see Inference parameters.

Playground – A user-friendly graphical interface in the AWS Management Console in which you can experiment with running model inference to familiarize yourself with Amazon Bedrock. Use the playground to test out the effects of different models, configurations, and inference parameters on the responses generated for different prompts that you enter. For more information, see Playgrounds.

Embedding – The process of condensing information by transforming input into a vector of numerical values, known as the embeddings, in order to compare the similarity between different objects by using a shared numerical representation. For example, sentences can be compared to determine the similarity in meaning, images can be compared to determine visual similarity, or text and image can be compared to see if they're relevant to each other. You can also combine text and image inputs into an averaged embeddings vector if it's relevant to your use case. For more information, see Run model inference and Knowledge base for Amazon Bedrock.

Advanced features

The following list introduces you to more advanced concepts that you can explore through using Amazon Bedrock.

Orchestration – The process of coordinating between foundation models and enterprise data and applications in order to carry out a task. For more information, see Agents for Amazon Bedrock.

Agent – An application that carry out orchestrations through cyclically interpreting inputs and producing outputs by using a foundation model. An agent can be used to carry out customer requests. For more information, see Agents for Amazon Bedrock.

Retrieval augmented generation (RAG) – The process of querying and retrieving information from a data source in order to augment a generated response to a prompt. For more information, see Knowledge base for Amazon Bedrock.

Model customization – The process of using training data to adjust the model parameter values in a base model in order to create a custom model. Examples of model customization include Fine-tuning, which uses labeled data (inputs and corresponding outputs), and Continued Pre-training, which uses unlabeled data (inputs only) to adjust model parameters. For more information about model customization techniques available in Amazon Bedrock, see Custom models.

Hyperparameters – Values that can be adjusted for model customization to control the training process and, consequently, the output custom model. For more information and definitions of specific hyperparameters, see Custom model hyperparameters.

Model evaluation – The process of evaluating and comparing model outputs in order to determine the model that is best suited for a use case. For more information, see Model evaluation.

Provisioned Throughput – A level of throughput that you purchase for a base or custom model in order to increase the amount and/or rate of tokens processed during model inference. When you purchase Provisioned Throughput for a model, a provisioned model is created that can be used to carry out model inference. For more information, see Provisioned Throughput.