



AWS Decision Guide

# Choosing a generative AI service



# Choosing a generative AI service: AWS Decision Guide

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

# Table of Contents

**Decision Guide**

Introduction

Understand

Consider

Choose

Use

Explore

Resources

1

1

2

10

14

15

21

22

**Document history**

24

# Choosing a generative AI service

## Taking the first step

Purpose	Determine which AWS generative AI services are the best fit for your organization.
Last updated	August 28, 2024
Covered services	<ul style="list-style-type: none"><li><a href="#">Amazon Bedrock</a></li><li><a href="#">Amazon Bedrock Studio</a></li><li><a href="#">Amazon Q Business</a></li><li><a href="#">Amazon Q Developer</a></li><li><a href="#">Amazon SageMaker</a></li><li><a href="#">Amazon Titan foundation models</a></li><li><a href="#">Public foundation models</a></li></ul>

## Introduction

Generative AI is a set of artificial intelligence (AI) systems and models designed to generate content such as code, text, images, music, or other forms of data. These systems can produce new content based on patterns and knowledge learned from existing data. Increasingly, organizations and businesses are using generative AI to:

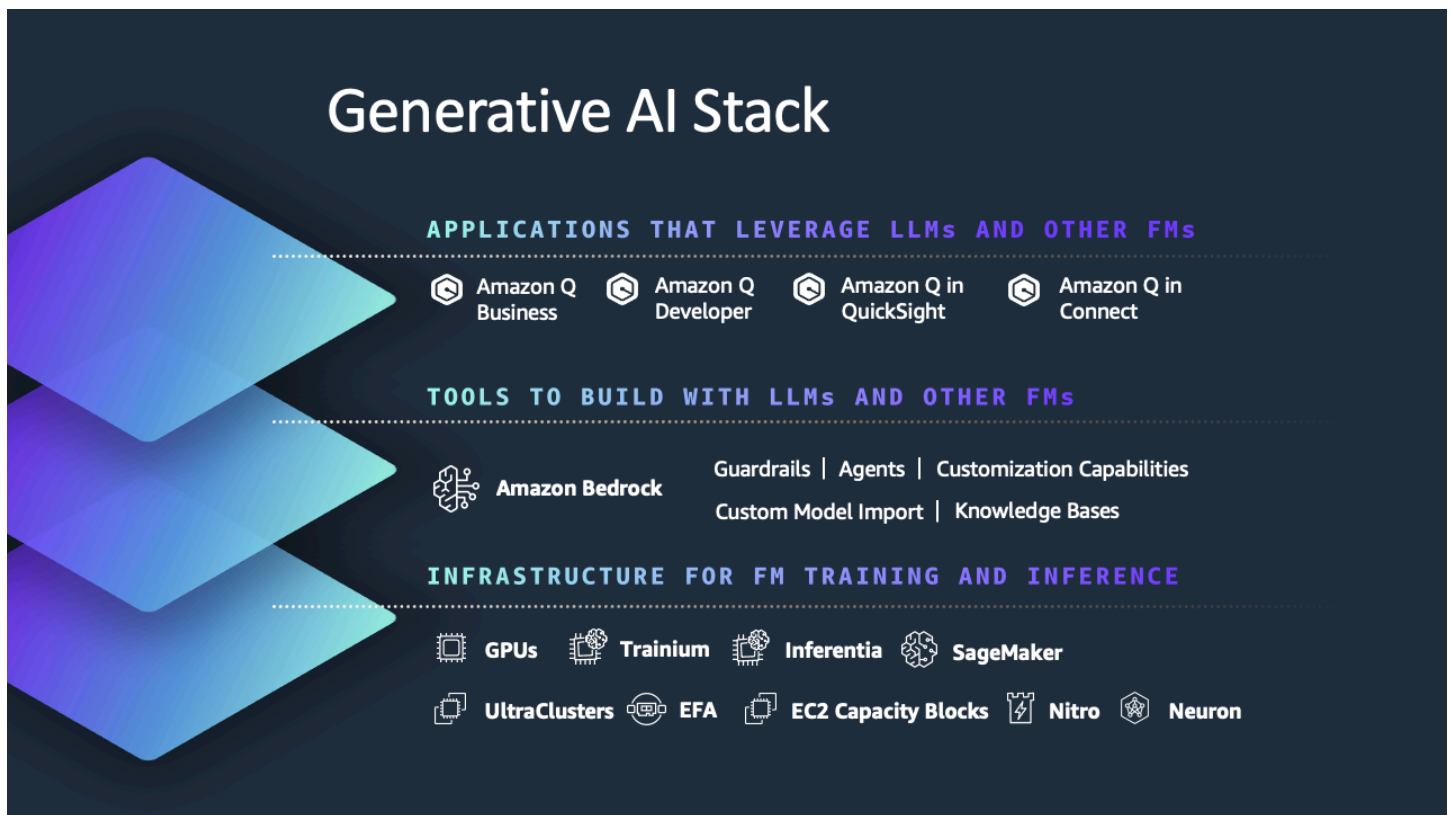
- **Automate creative workflows** — Use generative AI services to automate the workflows of time-consuming creative processes such as writing, image or video creation, and graphic design.
- **Customize and personalize content** — Generate targeted content, product recommendations, and customized offerings for an audience-specific context.
- **Augment data** — Synthesize large training datasets for other ML models to *unlock* scenarios where human-labeled data is scarce.
- **Reduce cost** — Potentially lower costs by using synthesized data, content, and digital assets.
- **Faster experimentation** — Test and iterate on more content variations and creative concepts than would be possible manually.

This guide helps you select the AWS generative AI services and tools that are the best fit for your needs and your organization.

## Understand

Amazon offers a range of generative AI services, applications, tools, and supporting infrastructure. Which of these you use depends a lot on the following factors:

- What you're trying to do
- How much choice you need in the foundation models that you use
- The degree of customization you need in your generative AI applications
- The expertise within your organization



### Amazon Q — Get pre-defined applications for your use case

At the top of Amazon's generative AI stack, Amazon Q generative AI-based applications use large language models (LLMs) and foundation models. However, they don't require that you explicitly choose a model. Each of these applications is aimed at a different use case and all are powered by [Amazon Bedrock](#).

Learn more about the primary Amazon Q generative AI–powered assistants currently available:

## Amazon Q Business

[Amazon Q Business](#) can answer questions, provide summaries, generate content, and securely complete tasks based on the data in your enterprise systems. It supports the general use case of using generative AI to start making the most of the information in your enterprise. With Amazon Q Business, you can make English-language queries about that information. It provides responses in a manner appropriate to your team's needs. In addition, you can create lightweight, purpose-built [Amazon Q Apps](#) within your Amazon Q Business Pro subscription.

## Amazon Q Developer

With [Amazon Q Developer](#), you can understand, build, extend, and operate AWS applications. The supported use cases include tasks that range from coding, testing, and upgrading applications, to diagnosing errors, performing security scanning and fixes, and optimizing AWS resources. The advanced, multistep planning and reasoning capabilities in Amazon Q Developer are aimed at reducing the work involved in common tasks (such as performing Java version upgrades). These capabilities can also help implement new features generated from developer requests.

Amazon Q Developer is also available as a feature in [several other AWS services](#) including AWS Chatbot, Amazon CodeCatalyst, Amazon EC2, AWS Glue, and VPC Reachability Analyzer.

[Chat with Amazon Q Developer](#) to query and explore your AWS infrastructure directly from the AWS Management Console. Using natural language prompts to interact with your AWS account, you can get specific resource details and ask about relationships between resources.

## Amazon Q in QuickSight

[Amazon Q in QuickSight](#) is aimed at meeting the needs of a specific use case: getting actionable insights from your data by connecting Amazon Q to the Amazon Q QuickSight business intelligence (BI) service. You can use it to build visualizations of your data, summarize insights, answer data questions, and build data stories using natural language.

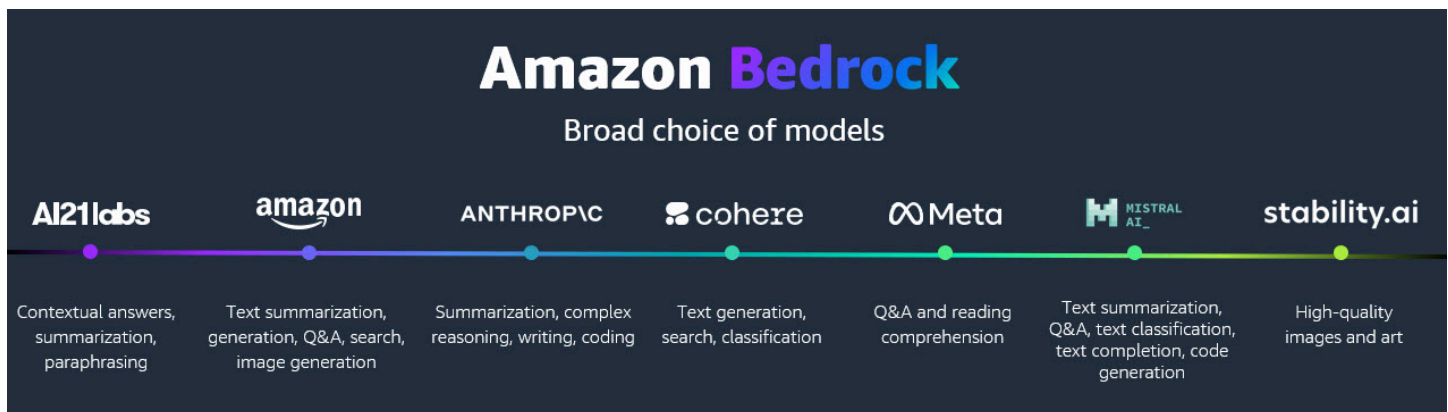
## Amazon Q in Connect

[Amazon Q in Connect](#) can automatically detect customer issues. It provides your customer service agents with contextual customer information along with suggested responses and actions for faster resolution of issues. It combines the capabilities of the [Amazon Connect](#) cloud contact center service with Amazon Q. Amazon Q in Connect can use your real-time

conversations with your customers, along with relevant company content, to recommend what to say or what actions an agent should take to assist customers.

## Amazon Bedrock — Choose your foundation models

If you're developing custom AI applications, need access to multiple foundation models, and want more control over the AI models and outputs, then [Amazon Bedrock](#) could be the service that meets your needs. Amazon Bedrock is a fully managed service, and it supports a choice of popular foundation models, including [Anthropic Claude](#), [Cohere Command & Embed](#), [AI21 Labs Jurassic](#), [Meta Llama](#), [Mistral AI](#), [Stable Diffusion XL](#) and [Amazon Titan](#).



In addition, Amazon Bedrock provides what you need to build generative AI applications with security, privacy, and responsible AI—regardless of the foundation model you choose. It also offers model-independent, single API access and the flexibility to use different foundation models and upgrade to the latest model versions, with minimal code changes.

Learn more about the key features of Amazon Bedrock:

### Model customization

[Model customization](#) can deliver differentiated and personalized user experiences. To customize models for specific tasks, you can privately fine-tune FMs using your own labeled datasets. Custom models include capabilities such as fine-tuning and continued pre-training using unlabeled datasets. The list of FMs for which Amazon Bedrock supports fine-tuning includes Cohere Command, Meta Llama 2, Amazon Titan Text Lite and Express, Amazon Titan Multimodal Embeddings, and Amazon Titan Image Generator. You can [fine-tune Anthropic Claude 3 Haiku](#) in a preview capacity in the US West (Oregon) AWS Region. The list of supported FMs is updated on an ongoing basis.

In addition, you can use [Amazon Bedrock Custom Model Import](#) (currently in preview) to bring your own custom models and use them within Amazon Bedrock.

## Agents

[Amazon Bedrock Agents](#) helps you plan and create multistep tasks using company systems and data sources—from answering customer questions about your product availability to taking their orders. You can create an agent by first selecting an FM and then providing it access to your enterprise systems, knowledge bases, and AWS Lambda functions to run your APIs securely. An agent analyzes the user request, and a Lambda function or your application can automatically call the necessary APIs and data sources to fulfill the request.

Agents [can retain memory](#) across multiple interactions to remember where you last left off and provide better recommendations based on prior interactions. Agents can also interpret code to tackle complex data-driven use cases, such as data analysis, data visualization, text processing, solving equations, and optimization problems.

## Guardrails

[Amazon Bedrock Guardrails](#) evaluates user inputs and FM responses based on use case specific policies, and provides an additional layer of safeguards, regardless of the underlying FM. Using a short natural language description, you can use Amazon Bedrock Guardrails to define a set of topics to avoid within the context of your application. Guardrails detects and blocks user inputs and FM responses that fall into the restricted topics.

Guardrails supports [contextual grounding checks](#), to detect hallucinations in model responses for applications using Retrieval Augmented Generation (RAG) and summarization applications. Contextual grounding checks add to the safety protection in Guardrails to make sure the LLM response is based on the right enterprise source data, and evaluates the LLM response to confirm that it's relevant to the user's query or instruction. Contextual grounding checks can detect and filter over 75% hallucinated responses for RAG and summarization workloads.

## Knowledge Bases

[Amazon Bedrock Knowledge Bases](#) is a fully managed capability that you can use to implement the entire Retrieval Augmented Generation (RAG) workflow—from ingestion to retrieval and prompt augmentation—without having to build custom integrations to data sources, and manage data flows. Session context management is built in, so your application can support multi-turn conversations. You can use the Retrieve API to fetch relevant results for a user query from knowledge bases.



With RAG, you can provide a model with new knowledge or up-to-date info from multiple sources, including document repositories, databases, and APIs. For example, the model might use RAG to retrieve search results from Amazon OpenSearch Service or documents from Amazon Simple Storage Service. Amazon Bedrock Knowledge Bases fully manages this experience by connecting to your private data sources, including [Amazon Aurora](#), [Amazon OpenSearch Serverless](#), MongoDB, Pinecone, and Redis Enterprise Cloud. This list includes connectors for Salesforce, Confluence, and SharePoint (in preview), so you can access more business data to customize models for your specific needs.

## Converse API

Use the [Amazon Bedrock Converse API](#) to create conversational applications that send and receive messages to and from an Amazon Bedrock model. For example, you can create a chatbot that maintains a conversation over many turns and uses a persona or tone customization that is unique to your needs, such as a helpful technical support assistant.

## Tool use (function calling)

[Tool use \(function calling\)](#) gives a model access to tools that can help it generate responses for messages that you send to the model. For example, you might have a chat application that lets users find out the most popular song played on a radio station. To answer a request for the most popular song, a model needs a tool that can query and return the song information.

## Amazon Bedrock Studio

Explore [Amazon Bedrock Studio](#) (in preview), an SSO-enabled web interface that provides a way for developers across your organization to experiment with LLMs and other FMs, collaborate on projects, and iterate on generative AI applications. It offers a rapid prototyping environment and streamlines access to multiple foundation models (FMs) and developer tools in Amazon Bedrock. It also supports Amazon Bedrock Knowledge Bases and Amazon Bedrock Guardrails.

## Prompt management

Use Amazon Bedrock to create and save your own prompts using [Prompt management](#), so that you can save time by applying the same prompt to different workflows. When you create a prompt, you can select a model to run inference on it and modify the inference parameters to use. You can include variables in the prompt so that you can adjust the prompt for different use case.

## Prompt flows

[Prompt flows](#) for Amazon Bedrock offers the ability for you to use supported FMs to build workflows by linking prompts, foundational models, and other AWS services to create comprehensive solutions.

With prompt flows, you can quickly build complex generative AI workflows using a visual builder. You can integrate with Amazon Bedrock offerings such as FMs, knowledge bases, and other AWS services such as AWS Lambda by transferring data between them. You can also deploy immutable workflows to move from testing to production in few clicks.

## **Amazon SageMaker — Build custom models and control the full ML lifecycle, from data preparation to model deployment and monitoring**

With [Amazon SageMaker](#), you can build, train, and deploy machine learning models, including FMs, at scale. Consider this option when you have use cases that can benefit from extensive training, fine-tuning, and customization of foundation models. It also streamlines the sometimes-challenging task of evaluating which FM is the best fit for your use case.

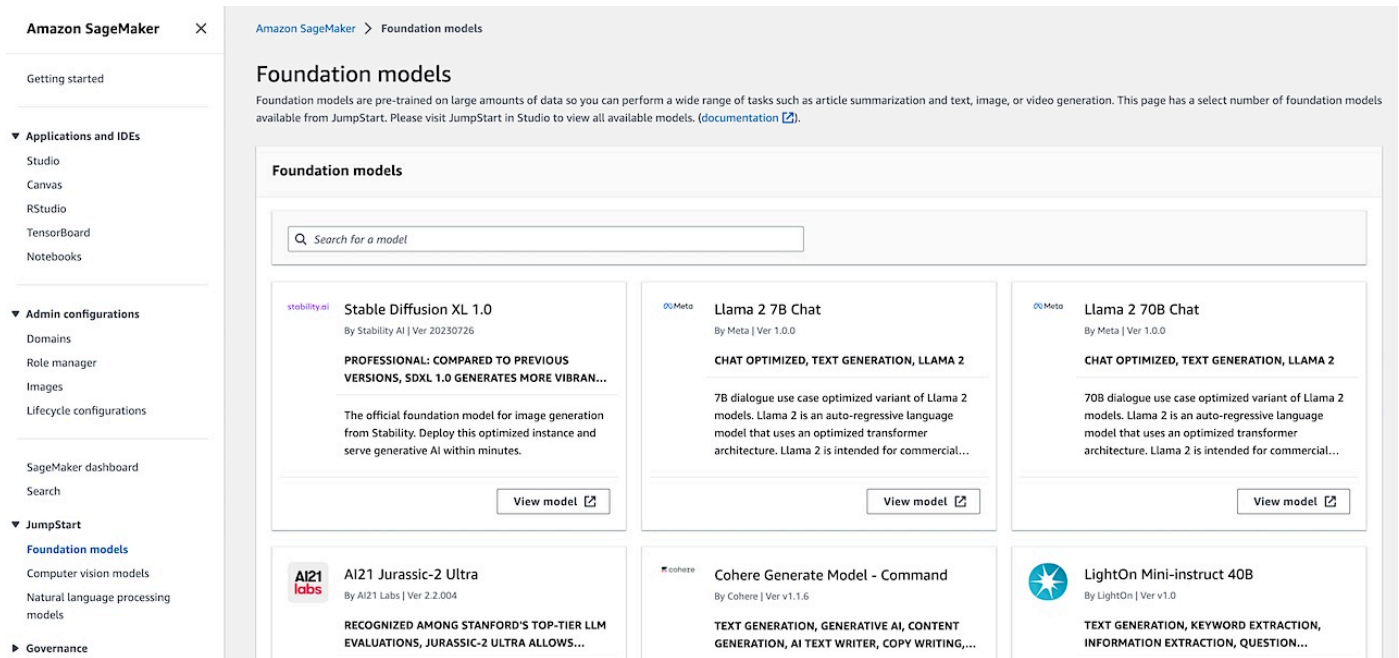
Amazon SageMaker also provides infrastructure and purpose-built tools for use throughout the ML lifecycle, including integrated development environments (IDEs), distributed training infrastructure, governance tools, machine learning operations (MLOps) tools, inference options and recommendations, and model evaluation.

Explore key features of Amazon SageMaker that may help you determine when to use it:

### SageMaker JumpStart

[Amazon SageMaker JumpStart](#) is an ML hub that provides access to publicly available foundation models. Those models include Mistral, Llama 3, CodeLlama, and Falcon 2. They can be customized with advanced fine-tuning and deployment techniques such as Parameter Efficient Fine-Tuning (PEFT) and Low-Rank Adaptation (LoRA).

This following screenshot shows some of the available models in SageMaker JumpStart within the AWS Management Console.



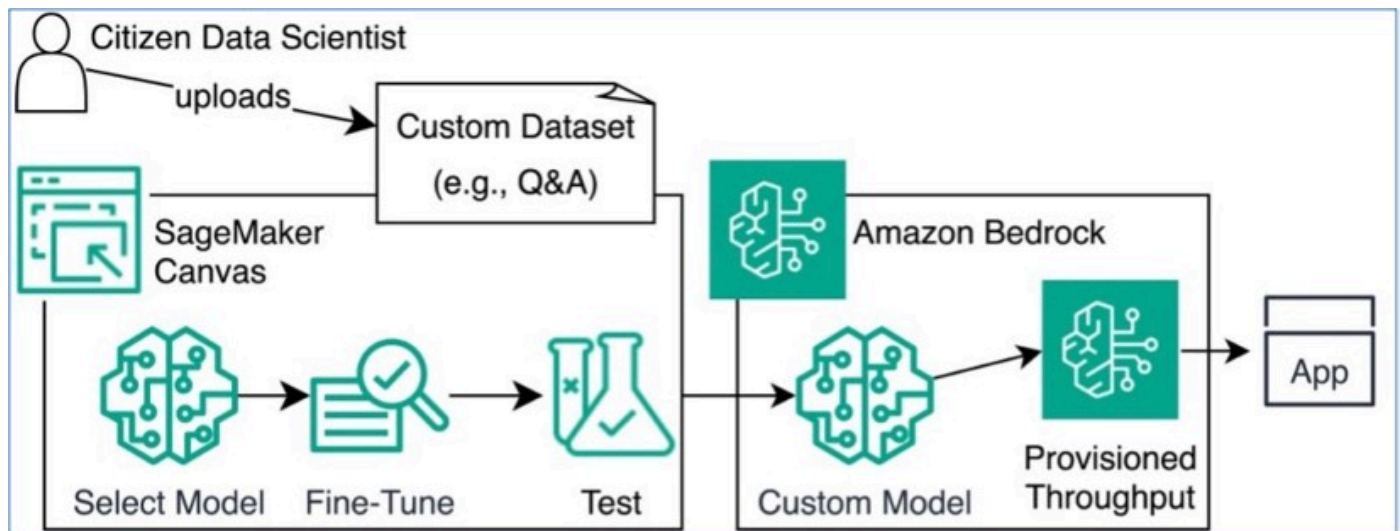
## SageMaker Clarify

[Amazon SageMaker Clarify](#) addresses the all-important decision of which foundation model to use. Use SageMaker Clarify to create model evaluation jobs. A model evaluation job evaluates and compares model quality and responsibility metrics for text-based foundation models from JumpStart. Model evaluation jobs also support the use of JumpStart models that have already been deployed to an endpoint.

## SageMaker Canvas

With [Amazon SageMaker Canvas](#), you can use machine learning to generate predictions without writing any code. You can also use Amazon SageMaker Canvas in collaboration with Amazon Bedrock to fine-tune and deploy language models.

[This blog post](#) describes how you can use them to optimize customer interaction by working with your own datasets (such as your product FAQs) in Amazon Bedrock and Amazon SageMaker JumpStart. The following diagram, from this blog post, demonstrates how SageMaker Canvas and Amazon Bedrock can be used together to fine-tune and deploy language models.



## SageMaker Studio

Amazon SageMaker Studio is a web-based experience for running ML workflows. Studio offers a suite of integrated development environments (IDEs). These include Code Editor, based on Code-OSS, Visual Studio Code - Open Source, a new JupyterLab application, RStudio, and Amazon SageMaker Studio Classic. For more information, see [Applications supported in Amazon SageMaker Studio](#).

The web-based UI in Studio provides access to all SageMaker resources, including jobs and endpoints, in one interface. ML practitioners can also choose their preferred IDE to accelerate ML development. A data scientist can use JupyterLab to explore data and tune models. In addition, a machine learning operations (MLOps) engineer can use Code Editor with the pipelines tool in Studio to deploy and monitor models in production.

SageMaker Studio includes generative AI assistance, powered by Amazon Q Developer right within your JupyterLab Integrated Development Environment (IDE). With Q Developer, you can access expert guidance on SageMaker features, code generation, and troubleshooting.

## Infrastructure for FM training and inference

AWS offers specialized, accelerated hardware for high performance ML training and inference.

- [Amazon EC2 P5](#) instances are equipped with NVIDIA H100 Tensor Core GPUs, which are well-suited for both training and inference tasks in machine learning.

- [Amazon EC2 G5](#) instances feature up to 8 NVIDIA A10G Tensor Core GPUs, and second generation AMD EPYC processors, for a wide range of graphics-intensive and machine learning use cases.
- [AWS Trainium](#) is the second-generation ML accelerator that AWS has purpose-built for deep learning (DL) training of 100B+ parameter models.
- [AWS Inferentia2-based Amazon EC2 Inf2 instances](#) are designed to deliver high performance at the lowest cost in Amazon EC2 for your DL and generative AI inference applications.

## Consider

After you've decided on a generative AI service, choose the foundation model (FM) that gives you the best results for your use case.

Amazon Bedrock has a model evaluation capability that can assist in evaluating, comparing, and selecting the best FMs for your use case. For more details on this capability, see [Amazon Bedrock model evaluation is now generally available](#) on the AWS News Blog.

Here are some critical factors to consider when choosing an appropriate FM for your use case:

### Modality

#### Identify use cases/modality

**What it is:** Modality refers to the type of data the model processes: text, images (vision), or embeddings.

**Why it matters:** The choice of modality should align with the data that you're working with. For example, if your project involves processing natural language, a text-based model like Claude, Llama 3.1, or Titan Text G1 is suitable. If you want to create embeddings, then you might use a model like [Titan Embeddings G1](#). Similarly, for image-related tasks, models such as Stable Diffusion XL, and Titan Image Generator v2, are more appropriate. Your use case might also involve considering your data source and the support for data source connectors, such as [those provided in Amazon Q Business](#).

### Model size

#### Model Size

**What it is:** This criterion refers to the number of parameters in a model. A *parameter* is a configuration variable that is internal to the model. Its values can be estimated (trained)

during the training phase from the given training data. Parameters are crucial as they directly define the model's capability to learn from data. Large models often have more than 50 billion parameters.

**Why it matters:** The number of parameters is a key indicator of the model's complexity. More parameters mean that the model can capture more intricate patterns and nuances in the data, which generally leads to better performance. However, these models are not only expensive to train, but also require more computational resources to operate.

## Inference latency

### Inference latency

**What it is:** *Inference speed*, or *latency*, is the time it takes for a model to process input (often measured in tokens) and return an output. This processing time is crucial when the model's responses are part of an interactive system, like an AWS Chatbot.

**Why it matters:** Quick response times are essential for real-time applications such as interactive chatbots or *instant translation* services. These applications depend on the model's ability to process and respond to prompts rapidly to maintain a smooth user experience. Although larger FMs typically offer more detailed and accurate responses, their complex architectures can lead to slower inference speeds. This slower processing might frustrate users expecting immediate interaction.

To address this challenge, you can choose models optimized for quicker responses, even if it means compromising somewhat on the responses' depth or accuracy.

## Context window

### Maximizing context window

**What it is:** A large language model's *context window* is the amount of text (in tokens) that the model can consider at any one time when generating responses.

**Why it matters:** Larger context windows enable the model to remember and process more information in a single run. This ability is particularly valuable in complex tasks such as understanding long documents, engaging in detailed conversations, or generating contextually accurate text over larger spans.

For example, in a conversation, a model with a larger context window can remember more of the earlier dialogue, and provide responses that are more relevant to the entire conversation.

This leads to a more natural and satisfying user experience, as the model can maintain the thread of discussion without losing context.

## Pricing

### Pricing considerations

**What it is:** The cost of using an FM is influenced by the model's complexity and the model provider's pricing structure.

**Why it matters:** Deploying high-performance models often comes with high costs due to increased computational needs. While these models provide advanced capabilities, their operational expenses can be high, particularly for startups or smaller projects on tight budgets.

Smaller, less resource-intensive models offer a more budget-friendly option without significantly compromising performance. Weigh the model's cost against its benefits to ensure it fits within your project's financial constraints and gets you the best value for your investment.

## Fine-tuning

### Fine-tuning and continuous pre-training capability

**What it is:** Fine-tuning is a specialized training process in which a pre-trained model that has been trained on a large, generic dataset is further trained (or fine-tuned) on a smaller, specific dataset. This process adapts the model to particularities of the new data, improving its performance on related tasks. Continuous pre-training, on the other hand, involves extending the initial pre-training phase with additional training on new, emerging data that wasn't part of the original training set, helping the model stay relevant as data evolves. You can also use [Retrieval Augmented Generation \(RAG\)](#) to retrieve data from outside an FM and augment your prompts by adding the relevant retrieved data in context.

**Why it matters:** With fine-tuning, you can increase model accuracy by providing your own task-specific labeled training dataset and further specialize your FMs. With continued pre-training, you can train models using your own unlabeled data in a secure and managed environment. Continuous pre-training helps models become more domain-specific by accumulating more robust knowledge and adaptability beyond their original training.

## Data quality

### Data quality

Data quality is a critical factor in the success of a generative AI application. Consider the following quality factors:

- **Relevance:** Ensure that the data you use for training your generative AI model is relevant to your application. Irrelevant or noisy data can lead to poor model performance.
- **Accuracy:** The data should be accurate and free from errors. Inaccurate data can mislead your model and result in incorrect outputs.
- **Consistency:** Maintain consistency in your data. Inconsistencies in the data can confuse the model and hinder its ability to learn patterns.
- **Bias and fairness:** Be aware of biases in your data, as they can lead to biased model outputs. Take steps to mitigate bias and help ensure fairness in your generative AI system.
- **Annotation and labeling:** If your application requires labeled data, verify that the annotations or labels are of high quality and created by experts.
- **Data preprocessing:** Prepare your data by cleaning and preprocessing it. This might involve text tokenization, image resizing, or other data-specific transformations to make it suitable for training.

## Data quantity

### Data quantity

Quantity along with quality goes hand in hand. Consider the following quantity factors:

- **Sufficient data:** In most cases, more data is better. Larger datasets allow your model to learn a wider range of patterns and generalize better. However, the required amount of data can vary depending on the complexity of your application.
- **Data augmentation:** If you have limitations on the quantity of available data, consider data augmentation techniques. These techniques involve generating additional training examples by applying transformations to existing data. For example, you can rotate, crop, or flip images or paraphrase text to create more training samples.
- **Balancing data:** Ensure that your dataset is balanced, especially if your generative AI application is expected to produce outputs with equal representation across different categories or classes. Imbalanced datasets can lead to biased model outputs.
- **Transfer learning:** For certain applications, you can use pre-trained models. With transfer learning, you can use models that were trained on massive datasets and fine-tune them with your specific data, often requiring less data for fine-tuning.



It's also important to continuously monitor and update your dataset as your generative AI applications evolve and as new data becomes available.

Quality of response

Quality of response

**What it is:** The most essential criterion is the quality of response. This is where you evaluate the output of a model based on several quality metrics, including accuracy, relevance, toxicity, fairness, and robustness against adversarial attacks.

- *Accuracy* measures how often the model's responses are correct (and you would typically measure this against a pre-configured standard or baseline).
- *Relevance* assesses how appropriate the responses are to the context or question posed.
- *Toxicity* checks for harmful biases or inappropriate content in the model's outputs.
- *Fairness* evaluates whether the model's responses are unbiased across different groups.
- *Robustness* indicates how well the model can handle intentionally misleading or malicious inputs designed to confuse it.

**Why it matters:** The reliability and safety of model outputs are paramount, especially in applications that interact directly with users or make automated decisions that can affect people's lives. High-quality responses ensure user trust and satisfaction, reducing the risk of miscommunication and enhancing the overall user experience, thus earning the trust of your customers.

Choose

Generative AI category	What is it optimized for?	Generative AI services
Amazon Q	Generating code and providing responses to questions across business data by connecting to enterprise data repositories to summarize the data logically,	<a href="#">Amazon Q Business</a> <a href="#">Amazon Q Developer</a>

Generative AI category	What is it optimized for?	Generative AI services
	analyze trends, and engage in dialogue about the data.	
Amazon Bedrock	Offering a choice of foundation models, customizing them with your own data, and building generative AI applications with the builder tools that Amazon Bedrock offers.	<a href="#">Amazon Bedrock</a> <a href="#">Amazon Bedrock Studio</a>
Amazon SageMaker	Building, training, and deploying machine learning models, including foundation models, at scale.	<a href="#">Amazon SageMaker</a>
Amazon FMs	Providing models that support a variety of multi-modal use cases such as text, image, and embeddings.	<a href="#">Amazon Titan</a>
Infrastructure for FM training and inference	Offering services that maximize the price performance benefits in FM training and inference.	<a href="#">AWS Trainium</a> <a href="#">AWS Inferentia</a>

## Use

Now that we've covered the criteria you need to apply in choosing an AWS generative AI service, you can select which services are optimized for your needs and explore how you might get started using each of them.

### Amazon Q

- **Get started with Amazon Q**

Review your options for getting started with Amazon Q Business and Amazon Q Developer in either the AWS Management Console or the IDE.

[Explore the guide](#)

- **Work with Amazon Q**

Use the Amazon Q Business and Amazon Q Developer User Guides, as well as the Amazon Q Business API Reference, to learn how you can tailor Amazon Q to your business needs. Learn how Amazon Q Business and Amazon Q Developer can help you understand, build, extend, and operate applications and workloads on AWS.

[Explore the guides](#)

- **Learn Amazon Q**

Take this short, introductory AWS Skill Builder course to get a high-level overview of Amazon Q (requires registration).

[Start the course](#)

## Amazon Q Business

- **What is Amazon Q Business?**

Get an overview of Amazon Q Business, with explanations of what it is, how it works, and how to get started using it.

[Explore the guide](#)

- **Create a sample Amazon Q Business application**

Learn how to create your first Amazon Q Business application in either the AWS Management Console or using the command line interface (CLI).

[Explore the guide](#)

- **Combine Amazon Q Business and AWS IAM Identity Center to build generative AI apps**

Build private and secure enterprise generative AI apps with Amazon Q Business and AWS IAM Identity Center.

[Read the blog post](#)

## Amazon Q Developer

- **What is Amazon Q Developer?**

Get an overview of Amazon Q Developer, with explanations of what it is, how it works, and how to get started using it.

[Explore the guide](#)

- **Get started with Amazon Q Developer**

Read this blog post to explore some key tasks that you can accomplish with Amazon Q Developer.

[Read the blog post](#)

- **Working with Amazon Q Developer**

Use the Amazon Q Developer Center for fast access to key Amazon Q Developer articles, blog posts, videos, and tips.

[Explore the Amazon Q Developer Center](#)

## Amazon Bedrock

- **What is Amazon Bedrock?**

Learn how to use this fully managed service to make foundation models (FMs) from Amazon and third parties available for your use through a unified API.

[Explore the guide](#)

- **Frequently asked questions about Amazon Bedrock**

Get answers to the most commonly-asked questions about Amazon Bedrock, including how to use agents, security considerations, details on Amazon Bedrock software development kits (SDKs), Retrieval Augmented Generation (RAG), how to use model evaluation, and how billing works.

[Read the FAQs](#)

- **Guidance for generating product descriptions with Amazon Bedrock**

Learn how to use Amazon Bedrock as part of a solution to automate your product review and approval process for an ecommerce marketplace or retail website.

[Explore the solution](#)

## Amazon Bedrock Studio

- **What is Amazon Bedrock Studio?**

Learn how you can use this web application to prototype apps that use Amazon Bedrock models and features, without having to set up and use a developer environment.

[Explore the guide](#)

- **Build generative AI applications with Amazon Bedrock Studio (preview)**

This blog explains how you can build applications using a wide array of top-performing models, as well as how to evaluate and share your generative AI apps within Amazon Bedrock Studio.

[Read the blog post](#)

- **Building an app with Amazon Bedrock Studio**

Use the Build mode in Amazon Bedrock Studio to create prototype apps that use Amazon Bedrock models and features. You can also use the Build mode to try experiments not supported in the Explore mode playground, such as setting inference parameters.

[Explore the guide](#)

## Amazon SageMaker

- **What is Amazon SageMaker?**

Learn how you can use this fully managed machine learning (ML) service to build, train, and deploy ML models into a production-ready hosted environment.

[Explore the guide](#)

- **Get started with Amazon SageMaker**

Learn how to join an Amazon SageMaker domain, giving you access to Amazon SageMaker Studio and RStudio on SageMaker.

[Explore the guide](#)

- **Get started with Amazon SageMaker JumpStart**

Explore SageMaker JumpStart solution templates that set up infrastructure for common use cases, and executable example notebooks for machine learning with SageMaker.

[Explore the guide](#)

## Amazon Titan

- **Amazon Titan in Amazon Bedrock overview**

Get an overview of Amazon Titan foundation models (FMs) to support your use cases.

[Explore the guide](#)

- **Cost-effective document classification using the Amazon Titan Multimodal Embeddings Model**

Learn how you can use this model to categorize and extract insights from high volumes of documents of different formats. This blog explores how you can use it to help determine the next set of actions to take, depending on the type of document.

[Read the blog post](#)

- **Build generative AI applications with Amazon Titan Text Premier, Amazon Bedrock, and AWS CDK**

Explore building and deploying two sample applications powered by Amazon Titan Text Premier in this blog post.

[Read the blog post](#)

## AWS Trainium

- **Overview of AWS Trainium**

Learn about AWS Trainium, the second-generation machine learning (ML) accelerator that AWS purpose built for deep learning training of 100B+ parameter models. Each Amazon EC2 Trn1 instance deploys up to 16 AWS Trainium accelerators to deliver a high-performance, low-cost solution for deep learning (DL) training in the cloud.

[Explore the guide](#)

- **Recommended Trainium Instances**

Explore how AWS Trainium instances are designed to provide high performance and cost efficiency for deep learning model inference workloads.

[Explore the guide](#)

- **Scaling distributed training with AWS Trainium and Amazon EKS**

If you're deploying your deep learning (DL) workloads using Amazon Elastic Kubernetes Service (Amazon EKS), learn how you can benefit from the general availability of Amazon EC2 Trn1 instances powered by AWS Trainium—a purpose-built ML accelerator optimized to provide a high-performance, cost-effective, and massively scalable platform for training DL models in the cloud.

[Read the blog post](#)

## AWS Inferentia

- **Overview of AWS Inferentia**

Understand how AWS designs accelerators to deliver high performance at the lowest cost for your deep learning (DL) inference applications.

[Explore the guide](#)

- **AWS Inferentia2 builds on AWS Inferentia1 by delivering 4x higher throughput and 10x lower latency**

Understand what AWS Inferentia2 is optimized for and how it was designed to deliver higher performance, while lowering the cost of LLMs and generative AI inference.

[Read the blog post](#)

- **Machine learning inference using AWS Inferentia**

Learn how to create an Amazon EKS cluster with nodes running Amazon EC2 Inf1 instances and optionally deploy a sample application. Amazon EC2 Inf1 instances are powered by AWS Inferentia chips, which are custom built by AWS to provide high-performance and low-cost inference in the cloud.

[Explore the guide](#)

## Explore

- **Architecture diagrams**

These reference architecture diagrams show examples of AWS AI and ML services in use.

[Explore architecture diagrams](#)

- **Whitepapers**

Explore whitepapers to help you get started and learn best practices in choosing and using AI and ML services.



[Explore whitepapers](#)

- **AWS solutions**

Explore vetted solutions and architectural guidance for common use cases for AI and ML services.

[Explore solutions](#)

## Resources

### Public foundation models

Supported foundation models are updated on a regular basis, and currently include:

- [Anthropic Claude](#)
- [Cohere Command & Embed](#)
- [AI21 Labs Jurassic](#)
- [Meta Llama](#)
- [Mistral AI](#)
- [Stable Diffusion XL](#)
- [Amazon Titan](#)

Use Amazon Bedrock and Amazon SageMaker to experiment with a variety of foundation models, and privately customize them with your data. To explore generative AI quickly, you also have the option of using [PartyRock, an Amazon Bedrock Playground](#). PartyRock is a generative AI app building playground with which you can experiment hands-on with prompt engineering.

### Associated blog posts

- [Build private and secure enterprise generative AI apps with Amazon Q Business and IAM Identity Center](#)
- [Amazon Q Developer, now generally available, includes previews of new capabilities to reimagine developer experience](#)
- [Chat about your AWS account resources with Amazon Q Developer](#)
- [Build enterprise-grade applications with natural language using AWS App Studio \(preview\)](#)

- [Amazon Bedrock model evaluation is now generally available](#)
- [Build generative AI applications with Amazon Bedrock Studio \(preview\)](#)
- [Fine-tune and deploy language models with Amazon SageMaker](#)

# Document history

The following table describes the important changes to this decision guide. For notifications about updates to this guide, you can subscribe to an RSS feed.

Change	Description	Date
<a href="#">Updated content</a>	Updated Amazon Bedrock feature names, including Amazon Bedrock Agents, Amazon Bedrock Guardrails, Amazon Bedrock Knowledge Bases, and Amazon Bedrock Custom Model Import.	August 28, 2024
<a href="#">Updated content</a>	Minor updates to improve readability.	August 16, 2024
<a href="#">Updated content</a>	Updates for newly released features of Amazon Q, Amazon SageMaker, and Amazon Bedrock.	July 18, 2024
<a href="#">Initial release</a>	Initial release of the decision guide.	July 9, 2024