Getting Started with Generative AI and Foundation Models

Large-language models (LLMs) and multi-modal models like text and image are enabling new capabilities, from code generation to the creation of images based on natural language descriptions. Together these new capabilities are transforming existing machine learning (ML)-enabled capabilities, such as web search and chatbots.

These models are known as foundation models (FMs) because customers can easily customize them for their specific use cases (such as summarization, translation, code generation) without having to build a new ML model from scratch for each use case. To help get you started with generative AI and foundation models, we provide an overview of the FM space, present a broad overview of the landscape, and outline opportunities and risks for LLMs. Finally, we show how you can leverage these LLMs using various AWS technologies to build highly differentiated solutions in a secure manner.

Table of Contents:

We are truly at an exciting inflection point in the widespread adoption of ML, and we believe most customer experiences and applications will be reinvented with generative AI.

PART 1

Introduction to Generative AI and Foundation Models

New Architectures Speed Up Innovation

Recent advancements in ML (specifically the invention of the *transformer-based* neural network architecture) have led to the rise of large-scale models that contain billions of parameters. A parameter is a configuration variable that is internal to the model and whose value can be tuned to optimize model performance. To give a sense for the change in scale, the largest pre-trained model in 2019 (BERT) was 330M parameters while the state-of-the-art LLM in 2022 is 540B parameters – an increase of 1,600x in size. The size of these models plays a big role in what makes them remarkable. To understand why, let's understand the basics of how FMs are built.

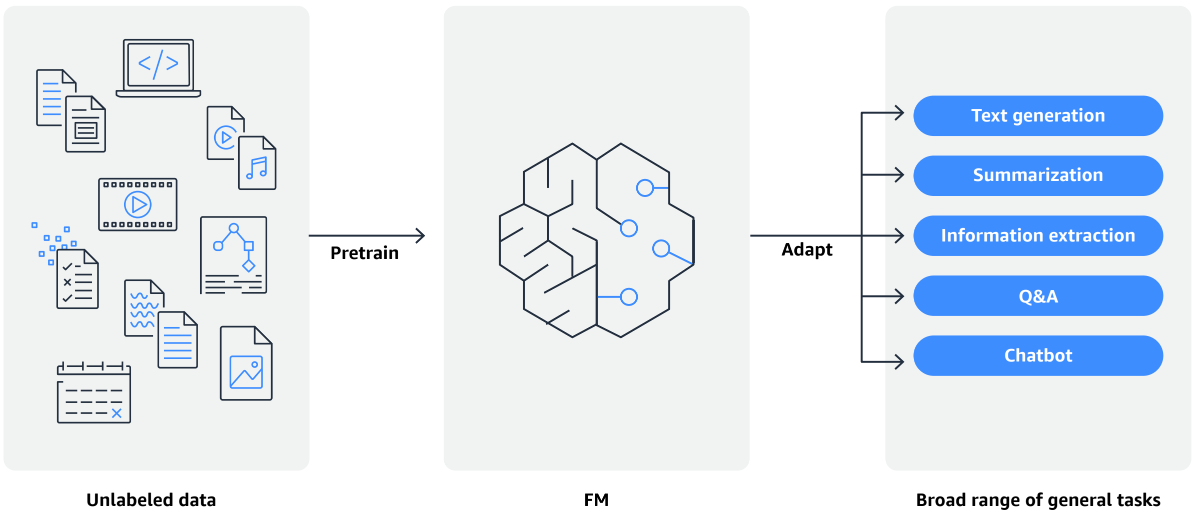
The main step of creating an FM (known as pre-training) involves training a model with terabytes of unlabeled text and/or multi-modal data (such as images, audio, video). This unlabeled data used for *pre-training* is usually obtained by crawling the Web and contains information from publicly crawled sources (such as Wikipedia and other sites) and proprietary data (if available). Unlabeled data can be used at scale for pre-training because it is a lot easier to obtain compared to labeled data which requires a human task force to create laborious annotations (e.g., explicitly adding the tag "dog" to an image that contains one). During the pre-training process, the model automatically takes context into account from all this training data, and tracks relationships in sequential data like the words in this sentence to develop some understanding of the real world.

A large model with billions of parameters can better capture this knowledge as it is able to store richer and deeper context across large amounts of data in its memory compared to a smaller model trained on a smaller data set. Pre-training models of this size requires access to: a) sufficient quantity and quality of training data (this involves collection of the relevant datasets and processing them, e.g., removing duplicates), and b) large-scale training infrastructure. Upon completion of the pre-training step, the resulting model can deliver an impressive out-of-the-box performance on a wide range of tasks

The largest pre-trained model in 2019 (BERT) was 330M parameters while the state-of-the-art LLM in 2022 is 540B parameters – an increase of 1,600x in size.

across multiple domains. For example, an FM can tackle many diverse tasks such as writing blog posts, summarizing documents, solving math problems, engaging in a chat dialogue, answering questions based on a doc, and even composing poetry.

How foundation models work



To understand FMs better, let's first dive deep into *transformers*, a popular model architecture that led to the rise of FMs. A transformer-based model has an encoder component that converts the input text into embeddings (mathematical representations), and a decoder component that consumes these *embeddings* to emit some output text. Compared to its predecessors like recursive neutral nets, transformers are more parallelizable because they do not process words sequentially one at a time; but instead, process the entire input all at once during the learning cycle. This makes the training process highly parallelizable; and as a result, transformers require significantly less time to train as one can apply more computing power to speed up training.

There are three main sub-types of transformer architectures – a) encoder-only, b) encoder-decoder, and c) decoder-only. An example of an *encoder-only* model is **M5 (a model used by the Amazon search in Amazon.com)**. Encoder-only models do not generate any human-interpretable outputs. Instead, they are most useful in applications where you need to efficiently query to find similar items. The items can be documents or any arbitrary entity with billions of different characteristics that are being tracked. The items are encoded into the ML model when the model is created, and when presented with a query or even some other document, the model will return those items most similar to the query or document submitted. For example, in the case of M5, we use it for Amazon Product Search where comparing embeddings from a user's search query and embeddings from the catalog data produces more accurate/ semantically relevant search results. An example of an *encoder-decoder* model is T5 (an open-source model from Google), which is trained to treat every natural language processing (NLP) problem (e.g., translate an input string in one language to another) as a text-to-text conversion problem. Finally, *decoder-only* models like OpenAI's GPT-3 or Amazon Titan's text generation model (more on that later) extend the input text sequence by generating new text (e.g., sentence auto-completion). Models with encoder-decoder and decoder-only architectures are known as *generative models* because they are capable of creating new content. These models are complex and more expensive to train (compared to encoder-only models).

Selecting and Customizing Foundations Models

ML practitioners usually choose a particular type of FM based on their needs and then either use it out of the box or customize it for their specific use case. While typically these FMs are used by ML practitioners, many FM providers offer a Web playground or a chat interface (e.g., ChatGPT) to make interacting with an FM easy, even for non-ML experts. Customers simply provide different natural language commands (known as *prompts*) such as "list all the action items from this meeting transcript," or "translate this doc into German." An FM's ability to handle simple commands like the ones above is known as *zero-shot learning*.

Customers (developers or ML practitioners) can also include a handful of examples as part of their input prompts to improve the relevancy of the model's outputs on the fly (this process is known as *in-context learning*). For example, a customer can provide the prompt: "create a social media ad for the 2023 Polygon Xtrada 7 mountain bike based on its product description" along with a few examples of their company's past social media ads for similar products. In this case, the FM adapts its output based on the given examples. Note that the FM does not go through any additional training in the case of both zero-shot and in-context learning and does not burden the user to create a separate ML model for this task of creating a social media ad (like one would have had to do just two years ago). Instead, the FM is able to generate outputs directly as part of the run-time inference process. This represents a major shift because traditional *task-specific* models were rigid in terms of their expectations for input and output (related to a specific task) whereas an FM can generalize well beyond the tasks that it saw as part of the pre-training process, allowing it to learn during inference (i.e. without additional training) from a new task definition and associated examples in the user prompt. Some FMs can handle prompts containing ~6,000 words, sufficiently long to provide the required context to make the model responses more relevant.

If one compares the out-of-the-box performance (i.e. no customization) for an FM on a particular task (e.g., sentiment analysis), the corresponding task-specific model (built on top of classic NLP algorithms) will likely be more accurate because it uses large volumes of labeled data and custom-designed model architectures. Furthermore, a task-specific model will likely also be faster and less expensive because it's typically smaller (i.e. millions of parameters) than an FM. This begs the question—"*What advantages do FMs have over task-specific models?*"

First, development of task-specific models requires companies to spin up disparate internal efforts that require them to staff dedicated teams of ML experts (who are difficult to hire), collect and annotate large volumes of labeled training data (a slow and laborious process), and fund infrastructure costs related to training individual models from scratch. Companies find that the cumulative costs to build multiple such models across their organization becomes steep and the dependency on data collection/annotation makes it hard to innovate faster. Effectively, developing individual task-specific models for the long tail of use cases does not scale well.

Second, if developers want to connect different tasks together in a single application, they have to build complicated orchestration and glue them together. On the other hand, with FMs, they enable the developer to address multiple tasks using natural language prompts with a single FM that gets better over time. As a result, a developer building this application is no longer burdened with complex orchestration logic across lots of ML models along with having to constantly improve the accuracy of all the individual ML models.

Finally, the use of task-specific models has been mostly limited to analysis tasks (such as document classification and object detection from an image) while FMs are capable of handling tasks that demand more creativity (including generative tasks) such as creating a video based on a natural language description.

Given that FMs can support multiple tasks out of the box, one might think of them as a Swiss Army knife. However, unlike a Swiss Army knife, FMs can also be customized for specific tasks. One customization approach is called *fine-tuning*. Here, customers can further train the FM by using a small number of labeled examples. This approach is efficient and cost-effective because the amount of labeled data required for fine-tuning is orders of magnitude smaller than what is required to develop a task-specific model from scratch. For example, a professional recruiting firm can customize the FM to automatically process new incoming resumes and generate summarized resumes at scale by fine-tuning the model with a few examples of candidate resumes and their corresponding summaries that follow the firm's format and writing style/tone.

To summarize, customers can use an FM in the following ways (ranked in terms of the degree of complexity from simple to advanced)—a zero-shot setting (ability to complete a task without any additional training examples), in-context learning (passing a handful of examples as part of user prompts), and fine-tuning (customizing the underlying FM for a specific task by using a small number of labeled examples).

Running Foundation Models – Inference

Once an FM is trained, customers have to invoke the model so that it can generate results for their production use cases. This step is called *inference*. However, the sheer size of these models makes it expensive to execute the model to get a prediction. Model *compression* techniques include *quantization* (approximating a neural network by using smaller precision 8-bit integers instead of 32-bit floating point numbers) and *distillation* (transferring of knowledge from a larger teacher model to a smaller student model). These can reduce the model size significantly (e.g. 10x+ smaller) while providing similar accuracy (while some loss in accuracy is unavoidable, the goal is to minimize it), improve latency, and make more cost-effective access. While model compression techniques (particularly, for generative FMs) are still evolving, they will likely play a key role in the broader FM adoption. Separately, the original size of the models matters even if they are compressed for inference. Let's imagine two pre-trained models—one with 500B parameters and the other with 100B parameters. If both models were to be compressed down to 10B parameters each, the one that was compressed from the 500B parameter model will usually deliver a higher task-specific accuracy (after fine-tuning) despite their inference costs being the same.

Learn more about how AWS is building generative AI and foundation models ›

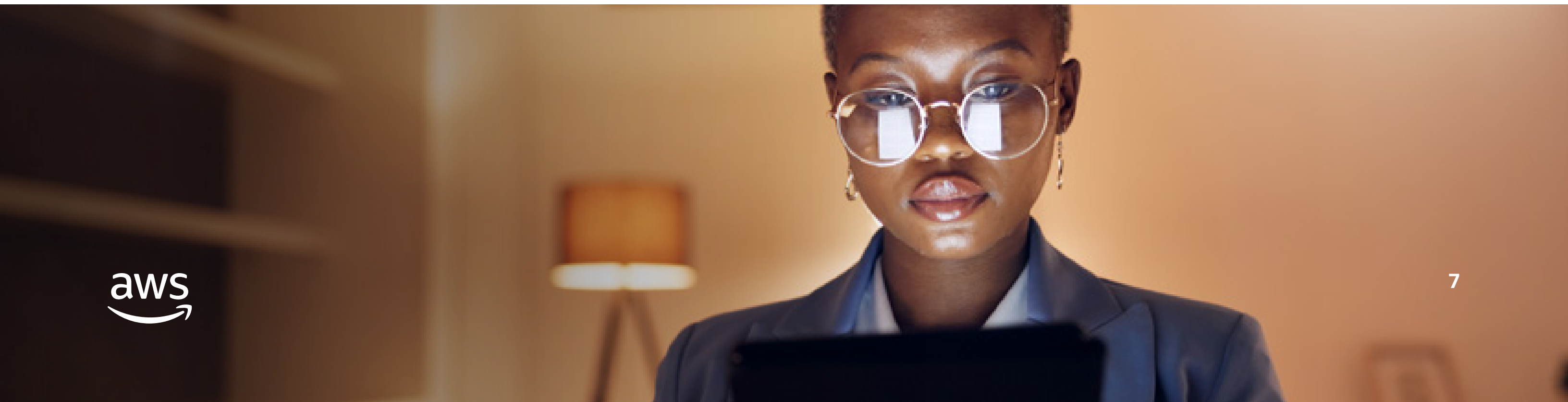
Issues to Consider with Foundation Models

Responsible AI
Foundation models raise new issues in defining, measuring, and mitigating responsible AI concerns across the development cycle including accuracy, fairness, intellectual property considerations, toxicity, and privacy, among others. These new challenges stem from the vast size of foundation models, trained by billions of parameters and their open-ended nature compared to traditional uses of machine learning, which are typically more focused and narrow. For example, look at the issue of fairness, can we ask an LLM to assign male and female pronouns at the same rate in reference to a doctor? Does that still apply if the prompt describes the doctor as having a beard? And should we do the same for other professions like nurses, firefighters, pilots, and attorneys – what about the Women's National Basketball Association (WNBA)? You can see that simply defining fairness in the context of an LLM is harder and requires new approaches and solutions. With foundation models we also see emerging concerns like toxicity, the possibility of generating content (whether it be text, images, or other modalities) that is offensive, disturbing, or otherwise inappropriate, and intellectual property considerations, where LLMs occasionally produce text or code passages that were verbatim regurgitations of parts of their training data. While these challenges may seem daunting, new research, science, and policies are already being created to address these challenges from end user education to filtering to more technical concepts like watermarking and differential privacy.

Hallucinations
LLMs can suffer from hallucination (i.e., they make up inaccurate responses that are not consistent with the training data). They are typically a byproduct of the way these FMs represent their inputs, often causing them not to distinguish among different numeric values or names, to "invent" facts in order to be consistent with the requested output format (e.g., inventing citations and author names when asked to provide evidence to an answer), and to conflate facts that are presented by multiple sources in their input.

To learn more about the specific challenges and solutions being developed across generative AI, foundation models, and responsible AI, read the "**Responsible AI in the generative era**" blog.

Costs
Today, most of the time and money spent on FMs goes into training them. This is because many customers are only just starting to deploy FMs into production. However, in the future, when FMs are deployed at scale, most costs will be associated with running the models and doing inference. While you typically train a model periodically, a production application can be constantly generating predictions, known as inferences, potentially generating millions per hour. And these predictions need to happen in real-time, which requires very low-latency and high-throughput networking. Alexa is a great example with millions of requests coming in every minute, which accounts for 40% of all compute costs. Some of the model compression considerations discussed above will play a key role here. Customizing models to specific use cases can result in smaller, more fine-tuned models that are more accurate and scale better.



PART 2

The Opportunity Ahead of Us

Generative AI has the potential to bring about sweeping changes to the global economy. According to **Goldman Sachs**, generative AI could drive a 7% (or almost $7 trillion) increase in global GDP and lift productivity growth by 1.5 percentage points over a 10-year period. **Goldman Sachs** also estimates that roughly two-thirds of US occupations will be enhanced by AI, and AI will likely drive the creation of new jobs and the emergence of new occupations for the vast majority of long-run employment growth. Much of this growth is driven by spend on generative AI cloud services which are estimated by **Bloomberg** to reach over $109B by 2030, a CAGR of 34.6% from 2022 to 2030.

At AWS, we have played a key role in democratizing ML and making it accessible to anyone who wants to use it, including more than 100,000 customers of all sizes and industries. This is why customers like Intuit, Thomson Reuters, AstraZeneca, Ferrari, Bundesliga, 3M, and BMW, as well as thousands of startups and government agencies around the world, are transforming themselves, their industries, and their missions with ML leveraging AWS capabilities from our infrastructure through our managed services and access to a variety of FMs.

The coming 12-24 months will be a period of intense experimentation. We expect new architectures to arise in the future, and this diversity will set off a wave of innovation. Some immediate applicability is in text generation (creating new pieces of original content), chatbots (conversation interfaces and virtual assistants to enhance user experience), search (search, find and synthesize information from large corpus of data to answer questions), text summarization (to get the gist without having to read full content), image generation (realistic 2D and now 3D images on various subjects from language prompts), personalization (more relevant and contextual product recommendations) and developer productivity (accelerate application development with code suggestions). Generative AI will play a transformational role in every industry and a subset are discussed below.

Healthcare and Life Sciences. One of the most promising use cases of generative AI is to accelerate drug discovery and research by using models to create novel protein sequences with specific properties for design of antibodies, enzymes, and vaccines, as well as gene therapy. According to many leading analysts, more than 30% of new drugs will be discovered using generative AI techniques. Healthcare and life sciences companies can also use FMs to design synthetic gene sequences for applications in synthetic biology and metabolic engineering, such as creating new biosynthetic pathways or optimizing gene expression for biomanufacturing purposes. Lastly, FMs can create synthetic patient and healthcare data, which can be useful for training AI models, simulating clinical trials, or studying rare diseases without access to large real-world datasets. Companies like 3M and Philips are partnering with AWS to leverage Amazon Bedrock (more on this later) to leverage generative AI to improve patient outcomes in clinical setting and medical imaging respectively.

Financial Services. Financial services companies can bring the power and cost-effectiveness of generative AI to serve their customers better while reducing costs. Financial intuitions can use conversational bots powered by FMs to improve customer service, gain customer loyalty, and ultimately boost their profitability. Lending institutions can fast-track loan approvals using FMs for financially underserved markets, especially in developing nations. Investment firms can use the power of FMs to provide personalized financial advice to their clients at low cost.

Media and Entertainment. From animations and scripts to full-length movies, generative AI can produce high-quality, novel content at a fraction of the cost and time it would traditionally take. Artists can complement and enhance their albums with AI-generated music to create whole new genres. Gaming companies can use Generative AI to create new games and allow players to build new avatars.

Education. Across education there are a number of emerging use cases that use generative AI to streamline learning. From creating concise summaries of long documents and lectures to question and flashcard generation to generate practice questions from a piece of content. Customers can generate assessment questions, which are then usually reviewed or adapted by a human to confirm their validity. Conversational interfaces can enhance various aspects of teaching and learning. Over time, these tools will generate other transformative capabilities. These include unique learning environments for each student by combining different sources of data and creating personalized learning pathways, as well as rapidly building simulations and virtual reality learning environments that allow students to learn in a more engaging and interactive way. We will likely see new ways to automate grading and measure student performance and to provide feedback and recommendations to teachers and students in near real time.

Automotive and Manufacturing. Automotive companies can use generative AI for a multitude of use cases, from engineering to in-vehicle experiences and customer service. Generative AI will help automotive companies optimize the design of mechanical parts to reduce drag in vehicle designs. Generative AI will also create new in-vehicle experiences, allowing for the design of personal assistants. Auto companies are using generative AI to deliver better customer service by providing quick responses to most common customer questions. New material, chip, and part designs can be created with generative AI to optimize manufacturing processes and drive down costs. Generative AI can also be used for synthetic data generation to test applications, especially for data not often included in testing datasets, such as defects or edge cases.

PART 3

Building with Generative AI at AWS

Customers have asked us how they can take advantage of what is out there today (and what is likely coming tomorrow) and quickly begin using FMs and generative AI within their business or organization to drive new levels of productivity and transform their offerings.

Customers can use generative AI across all lines of business including engineering, marketing, customer service, finance, and sales. Developers can get code suggestions to improve productivity and develop applications faster. Marketing teams can create ad copy and stunning visuals in minutes using natural language instructions. Customer service departments can transform call centers using generative AI-powered chatbots to answer customer questions with conversational interfaces. Finance teams can produce reports to summarize financial results tailored to different business units. Today, many AWS customers are seeing an impact from generative AI. For example, M5, a group within Amazon Search that helps teams across Amazon bring large models to their applications, trained large models to improve search results on Amazon.com.

Why customers choose AWS to build generative AI applications

First, Amazon AI/ML heritage.

AI and ML have been a focus for Amazon for over 20 years, and many of the capabilities customers use with Amazon are driven by ML. Our e-commerce recommendations engine is driven by ML; the paths that optimize robotic picking routes in our fulfillment centers are driven by ML; and our supply chain, forecasting, and capacity planning are informed by ML. Prime Air (our drones) and the computer vision technology in Amazon Go (our physical retail experience that lets consumers select items off a shelf and leave the store without having to formally check out) use deep learning. Alexa, powered by more than 30 different ML systems, helps customers billions of times each week to manage smart homes, shop, get information and entertainment, and more. We have thousands of engineers at Amazon committed to ML, and it's a big part of our heritage, current ethos, and future.

At AWS, we have played a key role in democratizing ML and making it accessible to anyone who wants to use it, including more than 100,000 customers of all sizes and industries. AWS has the broadest and deepest portfolio of AI and ML services at all three layers of the stack. We've invested and innovated to offer the most performant, scalable infrastructure for cost-effective ML training and inference;

developed Amazon SageMaker, which is the easiest way for all developers to build, train, and deploy models; and launched a wide range of services that allow customers to add AI capabilities like image recognition, forecasting, and intelligent search to applications with a simple API call. This is why enterprises, startups, and government agencies around the world are transforming themselves, their industries, and their missions with ML. We take the same democratizing approach to generative AI: we work to take these technologies out of the realm of research and experiments and extend their availability far beyond a handful of startups and large, well-funded tech companies.

Second, AWS is the easiest place to build with FMs.

1. **ML democratization through model choice**
   We've always been focused on ML democratization—just take a look at all the models accessible today in Amazon SageMaker—but we're taking this a step further with **Amazon Bedrock**, a new service that makes FMs from AI21 Labs, Anthropic, Stability AI, and Amazon accessible via an API. Bedrock is the easiest way for customers to build and scale generative AI-based applications using FMs, democratizing access for all builders. Bedrock will offer the ability to access a range of powerful FMs for text and images—including **Amazon's Titan FMs**, **which consist of two new LLMs**—through a scalable, reliable, and secure AWS managed service. With Bedrock's serverless experience, customers can easily find the right model for what they're trying to get done, get started quickly, privately customize FMs with their own data, and easily integrate and deploy them into their applications using the AWS tools and capabilities they are familiar with, without having to manage any infrastructure (including integrations with Amazon SageMaker ML features like Experiments to test different models and Pipelines to manage their FMs at scale).

   Amazon Bedrock
   A new service that makes FMs from AI21 Labs, Anthropic, Stability AI, and Amazon accessible via an API.

   Bedrock customers can choose from some of the most cutting-edge FMs available today. This includes the Jurassic-2 family of multilingual LLMs from AI21 Labs, which follow natural language instructions to generate text in Spanish, French, German, Portuguese, Italian, and Dutch. Claude, Anthropic's LLM, can perform a wide variety of conversational and text processing tasks and is based on Anthropic's extensive research into training honest and responsible AI systems. Bedrock also makes it easy to access Stability AI's suite of text-to-image foundation models, including Stable Diffusion (the most popular of its kind), which is capable of generating unique, realistic, high-quality images, art, logos, and designs.

   We have been previewing Amazon's new Titan FMs with a few customers before we make them available more broadly. We'll initially have two Titan models. The first is a generative LLM for tasks such as summarization, text generation (for example, creating a blog post), classification, open-ended Q&A, and information extraction. The second is an embeddings LLM that translates text inputs (words, phrases or possibly large units of text) into numerical representations (known as embeddings) that contain the semantic meaning of the text. While this LLM will not generate text, it is useful for applications like personalization and search because by comparing embeddings the model will produce more relevant and contextual responses than word matching. To continue supporting best practices in the responsible use of AI, Titan FMs are built to detect and remove harmful content in the data, reject inappropriate content in the user input, and filter the models' outputs that contain inappropriate content (such as hate speech, profanity, and violence).

2. **Secure customization to drive differentiation**
   One of the most important capabilities of Bedrock is how easy it is to customize a model. Customers simply point Bedrock at a few labeled examples in Amazon S3, and the service can fine-tune the model for a particular task without having to annotate large volumes of data (as few as 20 examples is enough). Imagine a content marketing manager who works at a leading fashion retailer and needs to develop fresh, targeted ad and campaign copy for an upcoming new line of handbags. To do this, they provide Bedrock a few labeled examples of their best performing taglines from past campaigns, along with the associated product descriptions. Bedrock makes a separate copy of the base foundational model that is accessible only to the customer and trains this private copy of the model. After training, Bedrock will automatically start generating effective social media, display ad, and web copy for the new handbags. None of the customer's data is used to train the original base models, and since all data is encrypted and does not leave a customer's Virtual Private Cloud (VPC), customers can trust that their data will remain private and confidential.

3. **Easy to start building**
   Given the huge potential of improving productivity and customer experiences with generative AI, customers are eager to get started quickly without having to spend months preparing and setting up their environments. With Amazon Bedrock, customers can readily integrate and deploy FMs into their applications and workloads running on AWS using familiar controls and integrations with AWS's depth and breadth of capabilities and services like Amazon SageMaker and Amazon S3. Bedrock is now in limited preview, and customers like **Coda** are excited about how fast their development teams have gotten up and running. Shishir Mehrotra, Co-founder and CEO of Coda, says, "As a longtime happy AWS customer, we're excited about how Amazon Bedrock can bring quality, scalability, and performance to Coda AI. Since all our data is already on AWS, we are able to quickly incorporate generative AI using Bedrock, with all the security and privacy we need to protect our data built-in. With over tens of thousands of teams running on Coda, including large teams like Uber, the New York Times, and Square, reliability and scalability are really important."

Third, AWS offers the most price-performant infrastructure for machine learning which is critical for customer's long-term ability to scale with generative AI.

Whatever customers are trying to do with FMs—running them, building them, customizing them—they need the most performant, cost-effective infrastructure that is purpose-built for ML. Over the last five years, AWS has been investing in our own silicon to push the envelope on performance and price performance for demanding workloads like ML training and Inference, and our AWS Trainium and AWS Inferentia chips offer the lowest cost for training models and running inference in the cloud. This ability to maximize performance and control costs by choosing the optimal ML infrastructure is why leading AI startups, like AI21 Labs, Anthropic, Cohere, Grammarly, Hugging Face, Runway, and Stability AI run on AWS.

As discussed earlier, in the future when FMs are deployed at scale, most costs will be associated with running the models and doing inference—constantly generating predictions. Inf2 instances deliver up to 4x higher throughput and up to 10x lower latency compared to the prior generation Inferentia-based instances. They also have ultra-high-speed connectivity between accelerators to support large-scale distributed inference. These capabilities drive up to 40% better inference price performance than other comparable Amazon EC2 instances and the lowest cost for inference in the cloud. Customers like Runway are seeing up to 2x higher throughput with Inf2 than comparable Amazon EC2 instances for some of their models. This high-performance, low-cost inference will enable Runway to introduce more features, deploy more complex models, and ultimately deliver a better experience for the millions of creators using Runway.

These capabilities drive up to 40% better inference price performance than other comparable Amazon EC2 instances and the lowest cost for inference in the cloud.

Fourth, customers can get started right away with some game-changing generative AI applications like Amazon CodeWhisperer.

One area where we see the use of generative AI growing rapidly is in coding. Software developers today spend a significant amount of their time writing code that is pretty straightforward and undifferentiated. They also spend a lot of time trying to keep up with a complex and ever-changing tool and technology landscape. All of this leaves developers less time to develop new, innovative capabilities and services. Developers try to overcome this by copying and modifying code snippets from the web, which can result in inadvertently copying code that doesn't work, contains security vulnerabilities, or doesn't track usage of open source software.

Generative AI can take this heavy lifting out of the equation by "writing" much of the undifferentiated code, allowing developers to build faster while freeing them up to focus on the more creative aspects of coding. This is why, last year, we announced the preview of **Amazon CodeWhisperer**, an AI coding companion that uses a FM under the hood to radically improve developer productivity by generating code suggestions in real-time based on developers' comments in natural language and prior code in their Integrated Development Environment (IDE). Developers can simply tell CodeWhisperer to do a task, such as "parse a CSV string of songs" and ask it to return a structured list based on values such as artist, title, and highest chart rank. CodeWhisperer provides a productivity boost by generating an entire function that parses the string and returns the list as specified. During the preview, we ran a productivity challenge, and participants who used CodeWhisperer completed tasks 57% faster, on average, and were 27% more likely to complete them successfully than those who didn't use CodeWhisperer. This is a giant leap forward in developer productivity, and we believe this is only the beginning.

On April 13, 2023 we announced the **general availability of Amazon CodeWhisperer** for Python, Java, JavaScript, TypeScript, and C#—plus ten new languages, including Go, Kotlin, Rust, PHP, and SQL. CodeWhisperer can be accessed from IDEs such as VS Code, IntelliJ IDEA, AWS Cloud9, and many more via the AWS Toolkit IDE extensions. CodeWhisperer is also available in the AWS Lambda console. In addition to learning from the billions of lines of publicly available code, CodeWhisperer has been trained on Amazon code. We believe CodeWhisperer is now the most accurate, fastest, and most secure way to generate code for AWS services, including Amazon EC2, AWS Lambda, and Amazon S3.

Participants who used CodeWhisperer completed tasks 57% faster, on average, and were 27% more likely to complete them successfully than those who didn't use CodeWhisperer.

Developers aren't truly going to be more productive if code suggested by their generative AI tool contains hidden security vulnerabilities or fails to handle open source responsibly. CodeWhisperer is the only AI coding companion with built-in security scanning (powered by automated reasoning) for finding and suggesting remediations for hard-to-detect vulnerabilities, such as those in the top ten Open Worldwide Application Security Project (OWASP), those that don't meet crypto library best practices, and others. To help developers code responsibly, CodeWhisperer filters out code suggestions that might be considered biased or unfair, and CodeWhisperer is the only coding companion that can filter and flag code suggestions that resemble open source code that customers may want to reference or license for use.

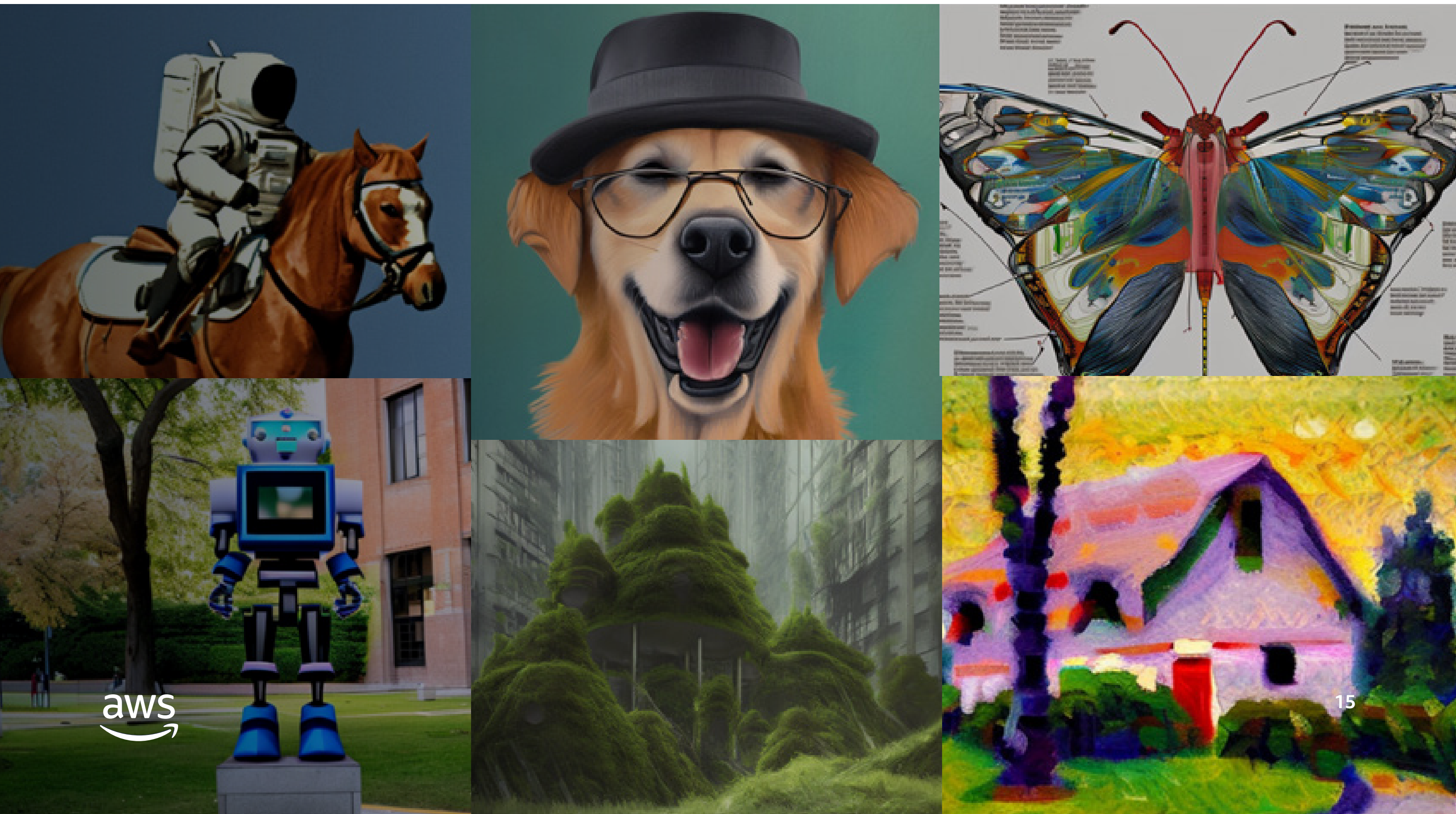
We know generative AI is going to change the game for developers, and we want it to be useful to as many as possible. This is why CodeWhisperer is free for all individual users with no qualifications or time limits for generating code! Anyone can sign up for CodeWhisperer with just an email account and become more productive within minutes. You don't even have to have an AWS account. For business users, we're offering a CodeWhisperer Professional Tier that includes administration features like single sign-on (SSO) with AWS Identity and Access Management (IAM) integration, as well as higher limits on security scanning.

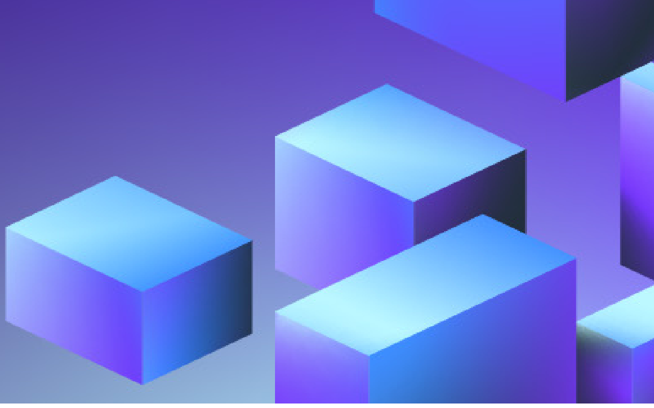
Many customers are already using CodeWhisperer to gain quick wins and improve their developer productivity. For example, Accenture uses CodeWhisperer to accelerate coding as part of its software engineering best practices initiative in its Velocity platform. Using CodeWhisperer, the company reduced development efforts by 30%.

At AWS we also focus on a few more topics that are critical for deploying generative AI at scale:

A focus on responsible AI: AWS builds foundation models with responsible AI in mind at each stage of its comprehensive development process. Throughout design, development, deployment, and operations we consider a range of factors including accuracy, fairness, intellectual property and copyright considerations, appropriate usage, toxicity, and privacy. We build solutions to address these issues into our processes for acquiring training data, into the FMs themselves, and into the technology that we use to pre-process user prompts and post-process outputs. For all our FMs, we invest actively to improve our features, and to learn from customers as they experiment with new use cases. At AWS, we know that generative AI technology and how it is used will continue to evolve, posing new challenges that will require additional attention and mitigation. Together with academic, industry and government partners, we are committed to the continued development of generative AI in a responsible way.

Partner community: To accelerate generative AI initiatives, customers can work with the AWS global partner community. Many global systems integrators such as Accenture, Deloitte, Infosys, and Slalom, and are building practices to help enterprises go faster with generative AI. And, ISVs like C3AI and Pega are excited to leverage Amazon Bedrock for easy access to its great selection of FMs with all of the security, privacy, and reliability they expect from AWS.





PART 4

Summing up

Our approach is to democratize generative AI. We work to take these technologies out of the realm of research and experiments and extend their availability far beyond a handful of startups and large, well-funded tech companies.

To recap some of the top reasons to build with generative AI on AWS consider:

- **Amazon's AI/ML heritage.** AI and ML have been a focus for Amazon for over 20 years, and many of the capabilities customers use with Amazon are driven by ML from our e-commerce recommendation engine to Amazon Go. At AWS, we have played a key role in democratizing ML and making it accessible to anyone who wants to use it, including more than 100,000 customers of all sizes and industries.
- **The easiest place to build with FMs.** AWS offers the broadest choice of models and secure customization – which makes it easy to start building. Customers have the ability to select the right model for their use case with a wide selection of FMs from lead AI startups include AI21 Labs, Anthropic, Stability AI, and Amazon. It is easy to customize a model with Amazon Bedrock with just a few labeled examples, and since all data is encrypted and does not leave a customer's VPC, customers can trust that their data will remain private and confidential.
- **The most price-performant infrastructure.** Whatever customers are trying to do with FMs—running them, building them, customizing them—they need the most performant, cost-effective infrastructure that is purpose-built for ML. Over the last five years, AWS has been investing in our own silicon to push the envelope on performance and price performance. Customers get the best price performance for generative AI.
- **Game-changing generative AI applications.** Get started right away with Amazon CodeWhisperer, an AI coding companion, that uses a FM under the hood to radically improve developer productivity by generating code suggestions in real-time. CodeWhisperer is the only AI coding companion with built-in security scanning for finding and suggesting remediations for hard-to-detect vulnerabilities – and it is available to all individual users for free.
- **Responsible AI.** AWS builds foundation models with responsible AI in mind at each stage of its comprehensive development process.

Start today with our generative AI-powered solutions ›

With generative AI built-in, services such as Amazon CodeWhisperer can help you improve productivity. In addition, you can deploy common generative AI use cases like call summarization and question answering using AWS sample solutions that combine AWS AI services with leading FMs.