

Introducing Generative AI

WHAT IS GENERATIVE AI · FOUNDATION MODELS & LLMs · PROMPT ENGINEERING · AWS GEN AI OFFERINGS ·
AMAZON Q DEVELOPER

STUDY & REVIEW

01 MUST KNOW

if you remember five things, remember these

- 01 New content, on demand:** generative AI is a type of AI that **creates new content and ideas** — conversations, stories, images, videos, music, even code. It is powered by pretrained ML models called **foundation models (FMs)**, and the generated content can be **edited** to fit your needs.
- 02 One model, many tasks:** a **traditional ML model** is siloed to one task and needs its own labeled data, training, and deployment. A **foundation model** is pretrained once on vast data, then **adapted** to many downstream tasks — customizing it to a domain uses only a **small fraction** of the data and compute of training from scratch.
- 03 FMs, LLMs, and FM types:** FMs use deep-learning techniques such as **neural networks**; **LLMs** are the text subset trained on **trillions of words**. FMs come as **text-to-text** (generate/extend text), **text-to-embeddings** (compare and connect text, e.g. search), and **multimodal** (text + images, e.g. Stable Diffusion).
- 04 Prompt → inference → output:** you shape an FM with a **prompt** (instructions · context · input data · output indicator); the model's response is an **inference**, shown as the output. **Prompt engineering** — refining keywords, context, and examples — is the main lever for steering an FM's behavior.
- 05 AWS delivers gen AI in three layers:** **compute** (AWS Trainium trains 100B+ parameter models, AWS Inferentia serves inference), **foundation models as a service** (Amazon Bedrock's API to many FMs; SageMaker JumpStart), and **applications** (Amazon Q Developer). Your data stays encrypted in your **VPC** and never trains the shared FMs.

02 KEY TERMS

TERM	DEFINITION
Generative AI	A type of AI that creates new content and ideas (conversations, stories, images, videos, music, code), powered by pretrained foundation models
Foundation model (FM)	A powerful ML model pretrained on vast data to perform a wide range of downstream tasks; adapted and customized rather than built per task
Large language model (LLM)	A subset of FMs trained on trillions of words that can understand, learn, and generate human-like text across many natural-language tasks
Pretraining	Training an FM once on vast data so a single model can later be adapted to many tasks and customized to a domain
Prompt	The text you input into a model; the model generates a response to it
Inference	The model's act of generating a response to a prompt; the result is shown as the output
Prompt engineering	Designing and refining prompts — keywords, context, examples — so an FM produces a specific type of output
Text-to-text FM	An FM for natural-language processing that takes input text and extends it with new generated text (summaries, answers, auto-completion)
Text-to-embeddings FM	An LLM type that compares text inputs (such as a search query) with indexed data and connects them; powers relevant search results
Multimodal FM	An FM that understands and generates more than one format, such as text and images (e.g. Stable Diffusion)
Amazon Bedrock	Fully managed service offering FMs from AWS and leading AI startups through a single API

TERM	DEFINITION
Amazon Q Developer	Generative AI-powered coding assistant that integrates with your IDE and supports the full software development lifecycle

03 TRADITIONAL ML VS. FOUNDATION MODELS

the core differentiation of this module

TRADITIONAL ML MODELS	FOUNDATION MODELS
<i>siloed — one model per task</i> — Built for a specific task: sentiment analysis, image classification, trend forecasting — Each task needs its own labeled data — Gather data → train a model → deploy, then repeat for the next task — Narrow, single-purpose output	<i>general-purpose — pretrain once, adapt many</i> — Pretrained on vast amounts of data — One model adapted to many downstream tasks (text, Q&A, summaries, images, code) — Customizable to domain-specific functions with a small fraction of data and compute — Large and general-purpose by design

04 THE PROMPT PATH

prompt engineering steers each step

01 Prompt — instructions · context · input · output indicator → 02 FM runs inference on the prompt → 03 Model returns the output → 04 Refine via prompt engineering and repeat
--

05 AWS GENERATIVE AI OFFERINGS AT A GLANCE

know the layer and one-line job of each

OFFERING	LAYER	WHAT IT DOES
Amazon Q Developer	Applications	Gen AI coding companion that integrates with your IDE
Amazon Bedrock	FMs as a service	Fully managed API to FMs from AWS and AI startups; build apps with answers from proprietary knowledge sources
SageMaker JumpStart	FMs / ML	Prebuilt, customizable solutions and reference architectures to start ML quickly (CloudFormation templates)
AWS Trainium	Compute	Second-gen ML accelerator purpose-built to train 100B+ parameter models
AWS Inferentia	Compute	Custom ML chip for high-performance inference; run on EC2 via the AWS Neuron SDK

06 AMAZON Q DEVELOPER ACROSS THE SDLC

one gen AI assistant, five phases

SDLC PHASE	WHAT AMAZON Q DEVELOPER DOES
Plan	Answer questions with contextual guidance; explain code through conversational coding
Create	In-line code recommendations in many languages; build features from comments or prompts; chat in the IDE
Test & secure	Generate unit tests; scan project code for security vulnerabilities with remediation suggestions
Operate	Troubleshoot errors; check network connectivity with VPC Reachability Analyzer
Maintain & modernize	Upgrade language versions with the Amazon Q Developer Agent for code transformation

- × “Generative AI is separate from machine learning.” — It is a **subset of ML**, powered by foundation models that use deep-learning techniques such as neural networks.
- × “A foundation model does one task like a traditional ML model.” — Traditional ML is siloed (one model, its own labeled data, per task); **one pretrained FM adapts to many tasks**.
- × “All foundation models are LLMs.” — **LLMs are the text subset** of FMs. FMs also include text-to-embeddings and multimodal models that handle search and images.
- × “Customizing an FM needs a huge labeled dataset.” — On Amazon Bedrock you can fine-tune with **as few as 20 examples**, using only a small fraction of the data and compute of training from scratch.
- × “Text-to-embeddings models generate long-form content.” — They **compare and connect** text (they power search); **text-to-text** models are the ones that generate and extend text.
- × “AWS Trainium and AWS Inferentia do the same job.” — **Trainium trains** (100B+ parameter models); **Inferentia serves inference**. Both run on EC2 and are invoked through the **AWS Neuron SDK**.
- × “Your data trains the shared FM when you customize on Bedrock.” — **No customer data trains the underlying models**; data is encrypted and stays within your VPC.
- × “Amazon Q Developer only writes code.” — It also **generates unit tests, scans for security vulnerabilities, troubleshoots, and modernizes code** across the entire SDLC.

08 SELF-QUIZ

answers are upside-down below

- Q1** What class of pretrained machine learning model sits at the core of every AI generator?
- Q2** List three kinds of new content that generative AI can create.
- Q3** Which subset of foundation models is trained on trillions of words across natural-language tasks?
- Q4** Which foundation-model type compares an input such as a search query against indexed data to connect the two?
- Q5** What is the term for designing and refining the input to a model so it produces a specific type of output?
- Q6** Name the four parts that make up the structure of a prompt.
- Q7** When a model generates a response to your prompt, what is that act of generating the response called?
- Q8** Which fully managed AWS service makes foundation models from AWS and leading AI startups available through a single API?
- Q9** Which two AWS-designed ML chips handle deep-learning training and high-performance prediction — and which does each?
- Q10** Which generative AI coding assistant integrates with your IDE, supports the full SDLC, and scans code for security vulnerabilities?

ANSWER KEY (rotate the page)

Q1 foundation models (FMs) **Q2** any three of conversations, stories, images, videos, music (and code) **Q3** large language models (LLMs) **Q4** text-to-embeddings **Q5** prompt engineering **Q6** instructions, context, input data, and output indicator **Q7** inference **Q8** Amazon Bedrock **Q9** AWS Trainium (training) and AWS Inferentia (inference) **Q10** Amazon Q Developer