CLOUD AI DEVELOPER GUIDE SERIES

Guide #1

Platform Overview & Your First API Call

AWS Bedrock · Azure OpenAI · GCP Vertex AI

AWS Bedrock	Azure OpenAI	GCP Vertex AI

Beginner ~25 min read Tier 1 of 3

cloudai.dev · 2025 Edition

Table of Contents

01 What Is Cloud AI?

- Foundation models explained
- Why cloud over self-hosted?
- Key terminology

02 Platform Landscape

- AWS Bedrock overview
- Azure OpenAI Service overview
- GCP Vertex AI overview

03 Side-by-Side Comparison

- Models available
- Pricing model
- Auth mechanism
- SDK ecosystem

04 Your First API Call

- AWS Bedrock (Python)
- Azure OpenAI (Python)
- GCP Vertex AI (Python)

05 What To Build Next

- Beginner project ideas
- Recommended learning path

01 What Is Cloud AI?

Foundation Models Explained

A **foundation model** (FM) is a large neural network pre-trained on massive datasets that can be adapted to a wide variety of tasks — text generation, summarisation, code completion, image analysis, and more. These models have billions of parameters and take months and millions of dollars to train from scratch.

Cloud AI platforms give you **API access** to these foundation models without owning any GPU infrastructure. You pay per token (input + output), spin up in minutes, and scale to millions of requests instantly.

Why Cloud vs. Self-Hosted?

Factor	Cloud AI	Self-Hosted
Setup time	Minutes	Days–weeks
GPU cost	Pay-per-token	High upfront CapEx
Model variety	Dozens instantly	Limited by hardware
Compliance / DLP	Platform-level controls	Full control, more work
Scaling	Automatic	Manual provisioning
Model updates	Automatic by provider	Manual upgrade cycle

Key Terminology

Token	The basic unit of text for LLMs (~4 characters or ~0.75 words). Billing is per 1,000 tokens.
Prompt	The input text you send to the model. Divided into a system prompt (instructions) and user message.
Completion / Response	The model's generated output tokens in reply to your prompt.
Context window	Maximum tokens (input + output) the model can process in one call. Ranges from 4K to 1M+ tokens.
Temperature	Controls randomness (0 = deterministic, 1 = creative). Start at 0 for structured tasks.

Foundation Model (FM)	Pre-trained large model you call via API without retraining.
RAG	Retrieval-Augmented Generation — injecting your own data into the prompt for grounded answers.

02 Platform Landscape

AWS Bedrock

Amazon Bedrock is a fully managed service that provides access to high-performing foundation models from leading AI companies including Anthropic, Meta, Mistral, Cohere, and Amazon's own Nova family — all through a single unified API.

- No separate accounts needed — works with your existing AWS IAM
- Guardrails, Knowledge Bases, and Agents built-in
- Data never used to train models by default (important for enterprise)
- Native Step Functions and Lambda integration
- Available in most AWS regions

Popular Models	Claude 3.5 Sonnet, Claude 3 Haiku, Amazon Nova Pro, Nova Lite, Nova Micro, Llama 3.1, Mistral Large
Auth Method	AWS IAM Roles (recommended) or Access Key + Secret Key
Primary SDK	boto3 (Python), AWS SDK for JavaScript, Java, Go

Azure OpenAI Service

Azure OpenAI Service provides enterprise-grade access to OpenAI's models (GPT-4o, GPT-4, o1, DALL-E) with Microsoft's cloud infrastructure, compliance, and regional data residency options.

- Dedicated capacity with Provisioned Throughput Units (PTUs)
- Private deployment — model runs in your Azure tenant
- Deep integration with Azure AI Search for RAG
- SOC 2, HIPAA, FedRAMP compliance certifications
- Content filtering included by default

Popular Models	GPT-4o, GPT-4o mini, GPT-4 Turbo, o1, o1-mini, DALL-E 3, Whisper, text-embedding-3-large
Auth Method	Azure AD / Entra ID (recommended) or API Key
Primary SDK	openai Python SDK (azure provider), Azure SDK for .NET, JavaScript

GCP Vertex AI

Google Cloud's Vertex AI is a unified ML platform offering Gemini models, third-party models via Model Garden, and tools for training, evaluation, and deployment — all integrated with BigQuery and Google's data ecosystem.

- Gemini 1.5 Pro supports a 1 million token context window
- Model Garden: 130+ open and proprietary models
- Native BigQuery ML integration for data-heavy use cases
- Grounding with Google Search for real-time factual answers
- Multi-modal out of the box (text, image, video, audio)

Popular Models	Gemini 1.5 Pro, Gemini 1.5 Flash, Gemini 1.0 Pro, Llama 3 (via Model Garden), Claude 3 (via Model Garden)
Auth Method	Google Service Account (JSON key) or Application Default Credentials (ADC)
Primary SDK	google-cloud-aiplatform Python SDK, Vertex AI REST API

03 Side-by-Side Comparison

At a Glance

Attribute	AWS Bedrock	Azure OpenAI	GCP Vertex AI
Model Families	Anthropic, Amazon, Meta, Mistral, Cohere	OpenAI (GPT-4o, o1, DALL-E)	Google Gemini, Meta Llama, others via Model Garden
Pricing Unit	Per 1K input/output tokens	Per 1K tokens (or PTU for reserved)	Per 1K characters or tokens
Free Tier	On-demand (no free tier)	Limited free credits on signup	$300 credit on new account
Auth	IAM Role or Access Keys	Entra ID or API Key	Service Account or ADC
Latency	Low (us-east-1 baseline)	Low (East US baseline)	Low (us-central1 baseline)
Compliance	SOC2, HIPAA, FedRAMP	SOC2, HIPAA, FedRAMP, ISO 27001	SOC2, HIPAA, ISO 27001
Agent Framework	Bedrock Agents (built-in)	Azure AI Agent Service	Vertex AI Agent Builder
RAG / Search	Knowledge Bases	Azure AI Search	Vertex AI Search
Observability	CloudWatch	Azure Monitor	Cloud Monitoring
Multi-modal	Yes (Nova, Claude)	Yes (GPT-4o Vision, DALL-E)	Yes (Gemini native)

■ Which platform should I start with?

If you already use AWS, start with Bedrock — zero new accounts, IAM just works. If your team is Microsoft-centric (.NET, Azure DevOps), go Azure OpenAI. If you work with large datasets in BigQuery or need massive context windows, start with Vertex AI Gemini.

04 Your First API Call

Each example below sends the same prompt — *"Explain what a vector database is in 2 sentences."* — to each platform using Python. Prerequisites, install commands, and the full response handling are shown.

AWS Bedrock

Prerequisites: AWS account · IAM user with **bedrock:InvokeModel** permission · Python 3.9+

Install:

```
pip install boto3
```

Code:

```
import boto3, json

client = boto3.client("bedrock-runtime", region_name="us-east-1")

response = client.invoke_model(
modelId="anthropic.claude-3-haiku-20240307-v1:0",
body=json.dumps({
"anthropic_version": "bedrock-2023-05-31",
"max_tokens": 256,
"messages": [
{
"role": "user",
"content": "Explain what a vector database is in 2 sentences."
}
]
}),
contentType="application/json",
accept="application/json",
)

result = json.loads(response["body"].read())
print(result["content"][0]["text"])
```

Azure OpenAI Service

Prerequisites: Azure subscription · Azure OpenAI resource created · Deployment named (e.g. *gpt-4o*) · API key from Azure Portal

Install:

```
pip install openai
```

Code:

```
import os
from openai import AzureOpenAI

client = AzureOpenAI(
api_key=os.environ["AZURE_OPENAI_KEY"],
api_version="2024-02-01",
azure_endpoint=os.environ["AZURE_OPENAI_ENDPOINT"],
)

response = client.chat.completions.create(
model="gpt-4o", # your deployment name
messages=[
{
"role": "user",
"content": "Explain what a vector database is in 2 sentences."
}
],
max_tokens=256,
)

print(response.choices[0].message.content)
```

GCP Vertex AI

Prerequisites: GCP project · Vertex AI API enabled · **gcloud auth application-default login** run · Python 3.9+

Install:

```
pip install google-cloud-aiplatform
```

Code:

```
import vertexai
from vertexai.generative_models import GenerativeModel

vertexai.init(project="your-gcp-project-id", location="us-central1")

model = GenerativeModel("gemini-1.5-flash")

response = model.generate_content(
"Explain what a vector database is in 2 sentences."
)

print(response.text)
```

■■ Never hardcode credentials

Always use environment variables, IAM roles, or secret managers. For AWS: prefer IAM Role over access keys. For Azure: prefer Managed Identity over API keys. For GCP: prefer Application Default Credentials (ADC) over JSON key files.

05 What To Build Next

Beginner Project Ideas

CLI Summariser	Read a .txt file and summarise it in 3 bullets using any platform above.
Prompt Tester	Run the same prompt on all 3 platforms and compare outputs side-by-side.
FAQ Bot	Paste 10 FAQs into the system prompt and let users query it from the terminal.
Code Explainer	Accept a Python function and return a plain-English explanation.
Email Drafter	Input bullet points → output a professional email draft.

Recommended Learning Path

Step	Action	Platform
1	Create an AWS account and enable Bedrock model access in the console	AWS
2	Run the Bedrock code sample above locally with Claude Haiku	AWS
3	Create a GCP project, enable Vertex AI, run the Gemini Flash example	GCP
4	Sign up for Azure, create an OpenAI resource, deploy GPT-4o mini	Azure
5	Build a prompt comparison script calling all 3 platforms in parallel	All
6	Read Guide #2: RAG on the Cloud — add your own data	All

■ Up Next: Guide #2 — RAG on the Cloud

Learn how to attach your own documents, databases, and knowledge to a foundation model using AWS Bedrock Knowledge Bases, Azure AI Search, and GCP Vertex AI Search. No ML degree required.