



Vertex AI Studio

In this module, you learn to ...

01

Use Vertex AI Studio to implement generative AI use cases

02

Explore the available models and sample use cases in Vertex AI Studio

03

Write and test prompts using the Google Cloud Console

04

Protect your data while using Vertex AI Studio



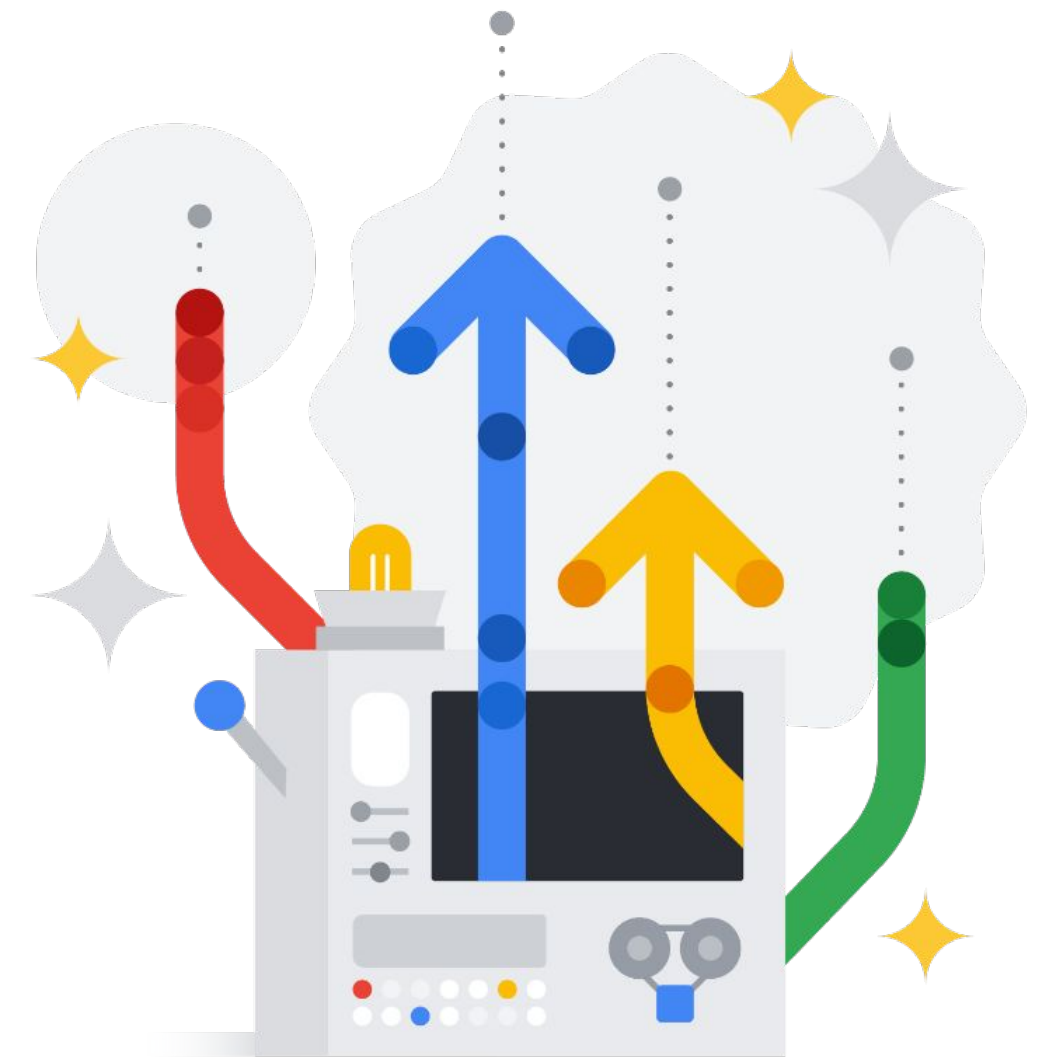
Topics

01	Vertex AI Studio
02	Designing and testing prompts
03	Data governance in Vertex AI Studio



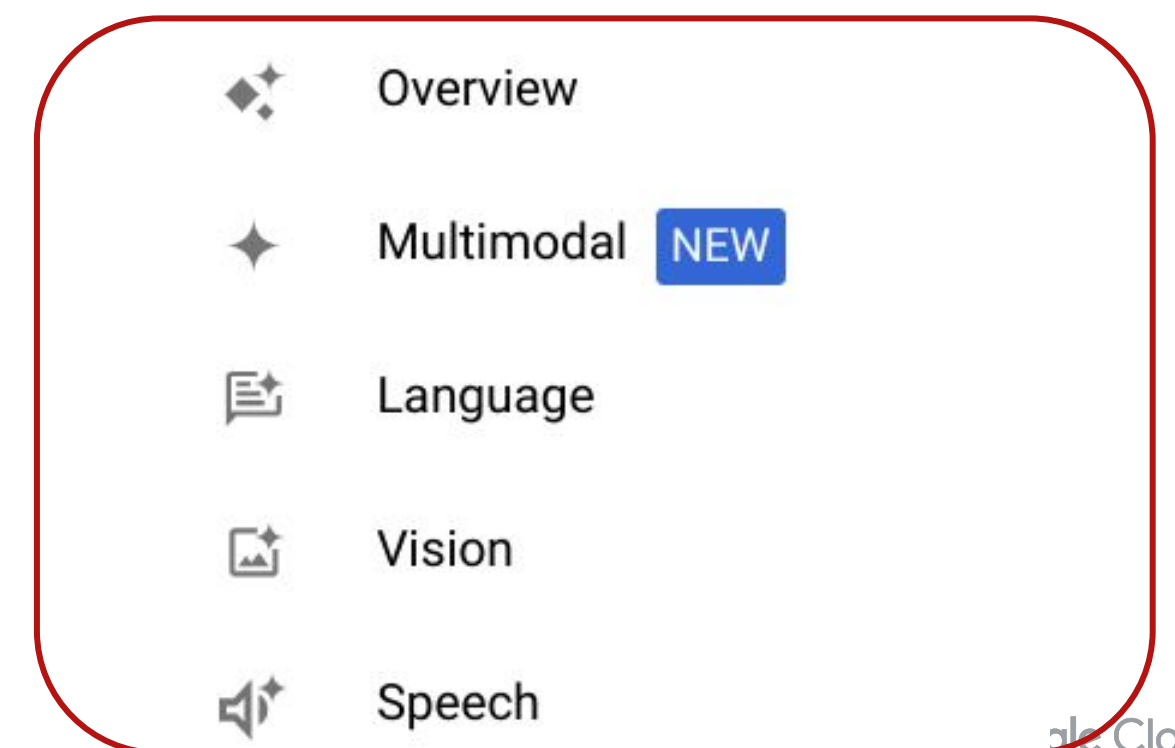
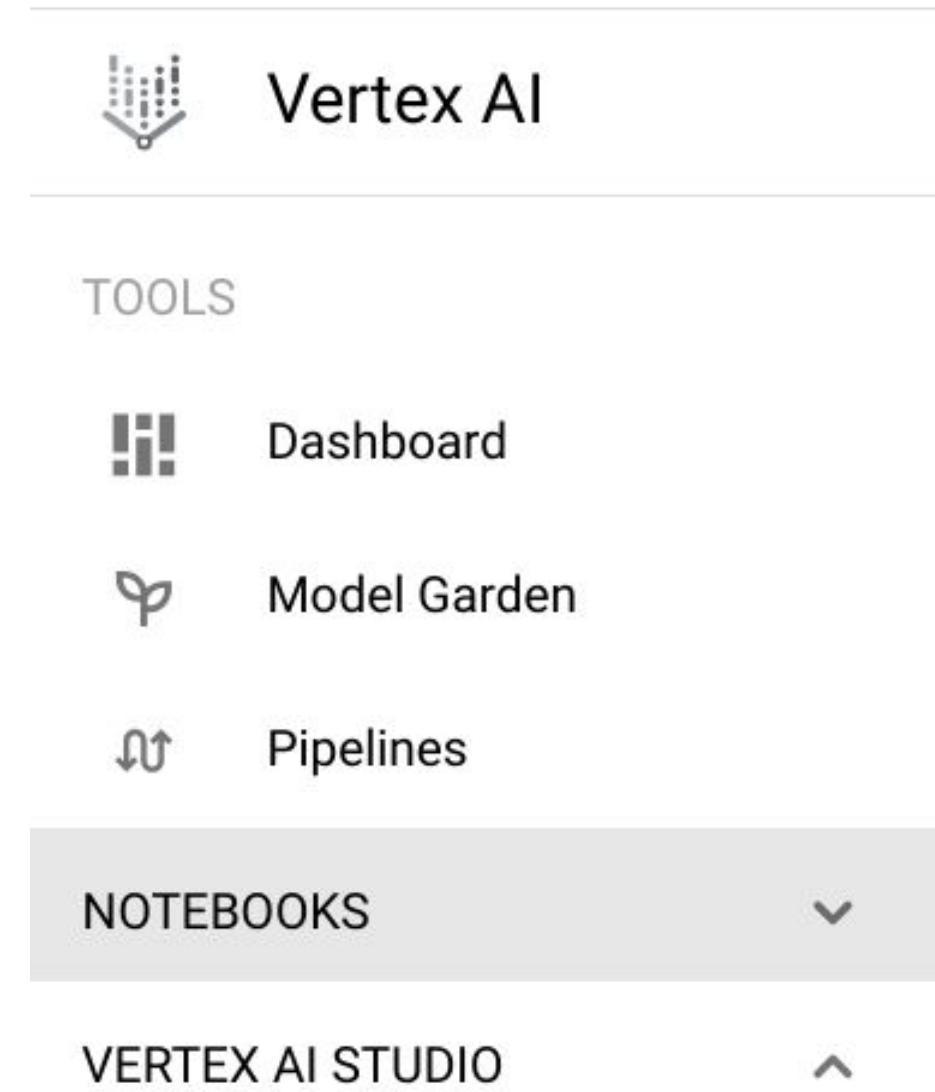
Vertex AI Studio provides an intuitive interface to prompt, tune, and deploy Google foundation models

- Supports Google foundational models
 - Gemini for multimodal
 - PaLM 2 for text and chat
 - Codey for code generation and chat
 - Chirp for speech to text
 - Imagen for text to image generation
- Allows users to easily experiment with prompts
 - Simple, intuitive design
 - Easily experiment with parameters
 - Add context and examples

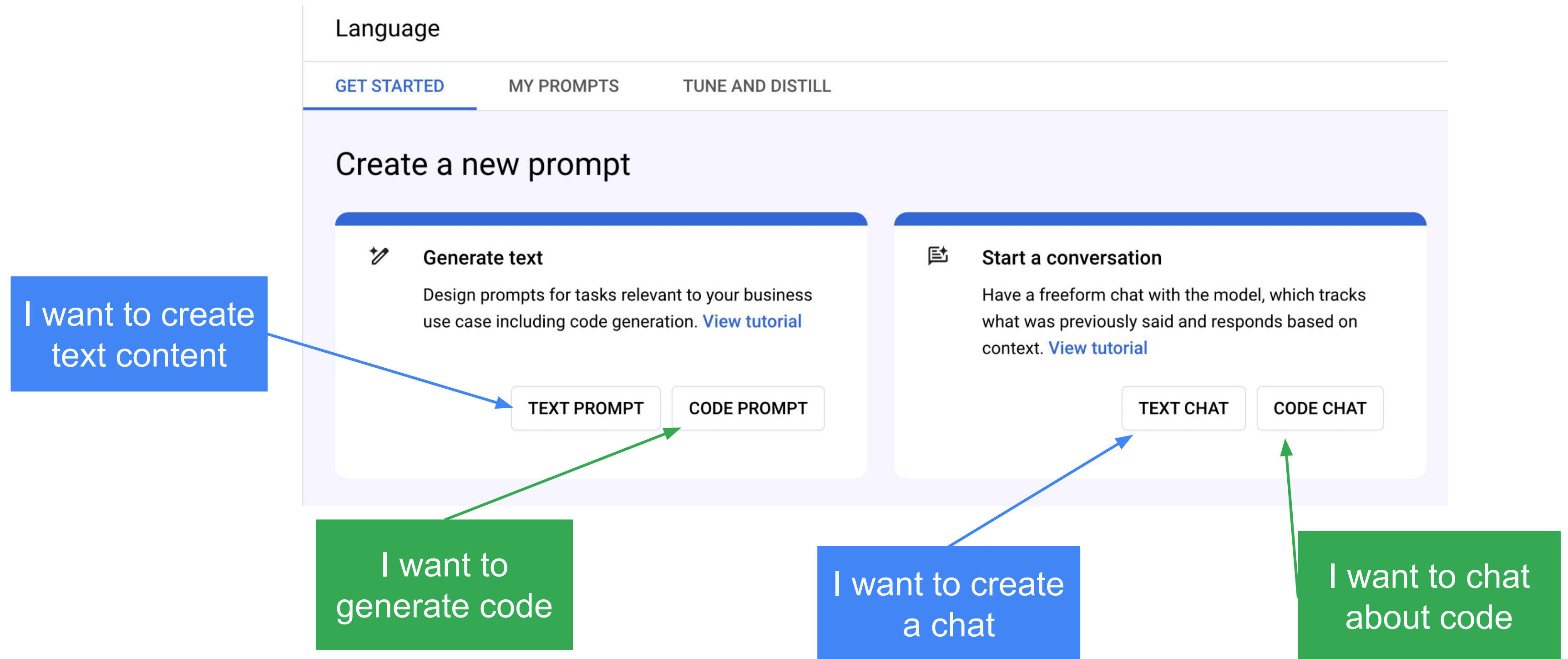


Vertex AI Studio is available as a feature of Google Cloud Vertex AI

- Choose from Multimodal, Language, Vision or Speech models



To get started, choose a task based on your goal



Vertex AI Studio UI

Save the model

Enter a Prompt

Select the foundational model

Adjust the parameters

The response

Submit the prompt

← grace-hopper-prompt ✎

FREEFORMSTRUCTURED

SAVEVIEW CODEDELETE

Prompt

Who was Grace Hopper?

Response

Grace Hopper, known as "Amazing Grace", was an American computer scientist and United States Navy rear admiral. She was a pioneer in the development of the COBOL programming language and was also one of the first programmers of the Harvard Mark I computer. She is widely considered to be one of the most important figures in the history of computing.

Model may display inaccurate or offensive information that doesn't represent Google's view. Not all languages are supported. [Learn more.](#)

Region

us-central1 (Iowa)

Model

text-bison (latest)

Temperature

0.2

Advanced

Token limit

1024

Top-K

40

Top-P

0.8

Max responses

1

Add stop sequence

Press Enter after each sequence

Streaming responses

Print responses as they're generated

Safety filter threshold

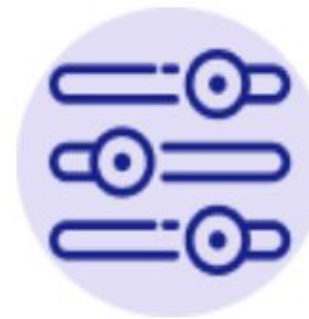
Block few

SUBMIT

RESET PARAMETERS

You can also use Vertex AI Studio to fine-tune foundational models

- You supply your own training data in JSON format
- The foundational model is customized for your particular use case
 - A new model is created based on PaLM 2
 - Uses a process called “Transfer Learning”
- The training job runs on Google infrastructure
- Your model and data are stored separately in your own project



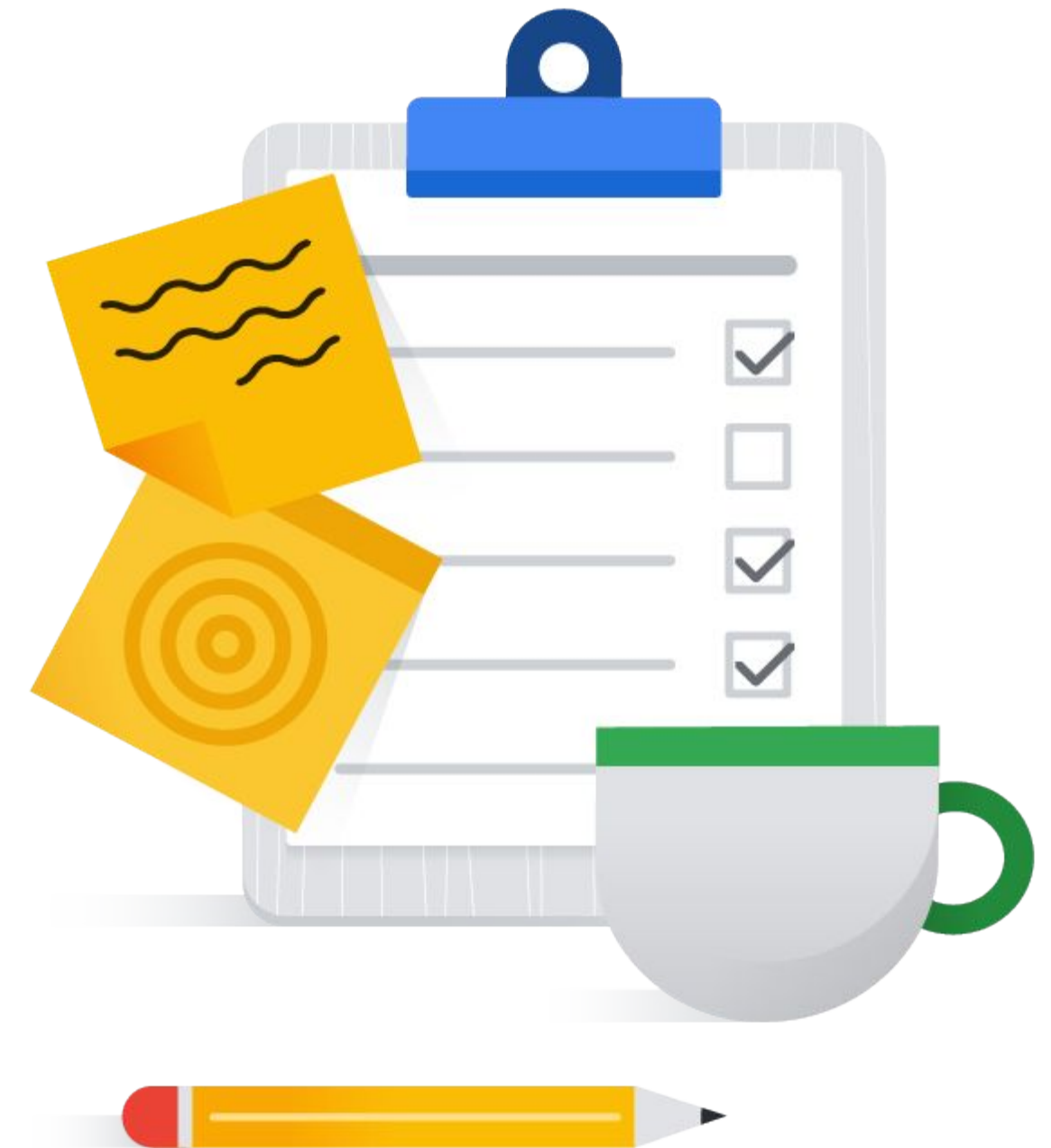
Tune a model

Tune a model so it's better equipped for your use case, then deploy to an endpoint to get predictions or test it in prompt design. [View tutorial](#)


NEW TUNED MODEL

Topics

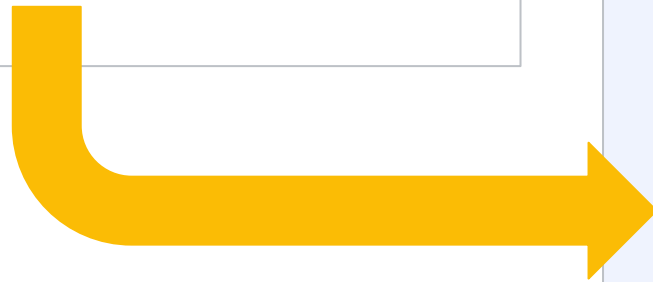
01	Vertex AI Studio
02	Designing and testing prompts
03	Data governance in Vertex AI Studio





Start with a simple text prompt and click the Submit button to see the results

 Prompt

Tell me about Grace Hopper



 Response

 Markdown

Grace Hopper was an American computer scientist and United States Navy rear admiral. She was one of the pioneers of computer programming, and is credited with coining the term "debugging". She also helped develop the first compiler, which translates human-readable code into machine code that computers can understand.

Hopper was born in New York City in 1906. She attended Vassar College, where she studied mathematics and physics. After graduating, she taught mathematics at Vassar for a year before joining the United States Navy in 1943.

During World War II, Hopper worked on the Mark I computer, one of the first electronic computers. She was responsible for developing the first compiler, which translated human-readable code into machine code that computers could understand. This work made it much easier for programmers to write code for computers, and helped to make computers more accessible to a wider range of people.

After the war, Hopper continued her work in computer science. She helped develop the COBOL programming language, which is still widely used today. She also served as the director of the Navy's computer science research center.

In 1986, Hopper was awarded the National Medal of Technology for her contributions to computer science

How about a code prompt?

▼ Prompt

Write the Terraform code to create a Google Cloud VPC with Subnets in us-central1 and us-east4. Create variables for the IP address ranges.

▼ Response



```
provider "google" {
  project = var.project_id
  region  = var.region
}

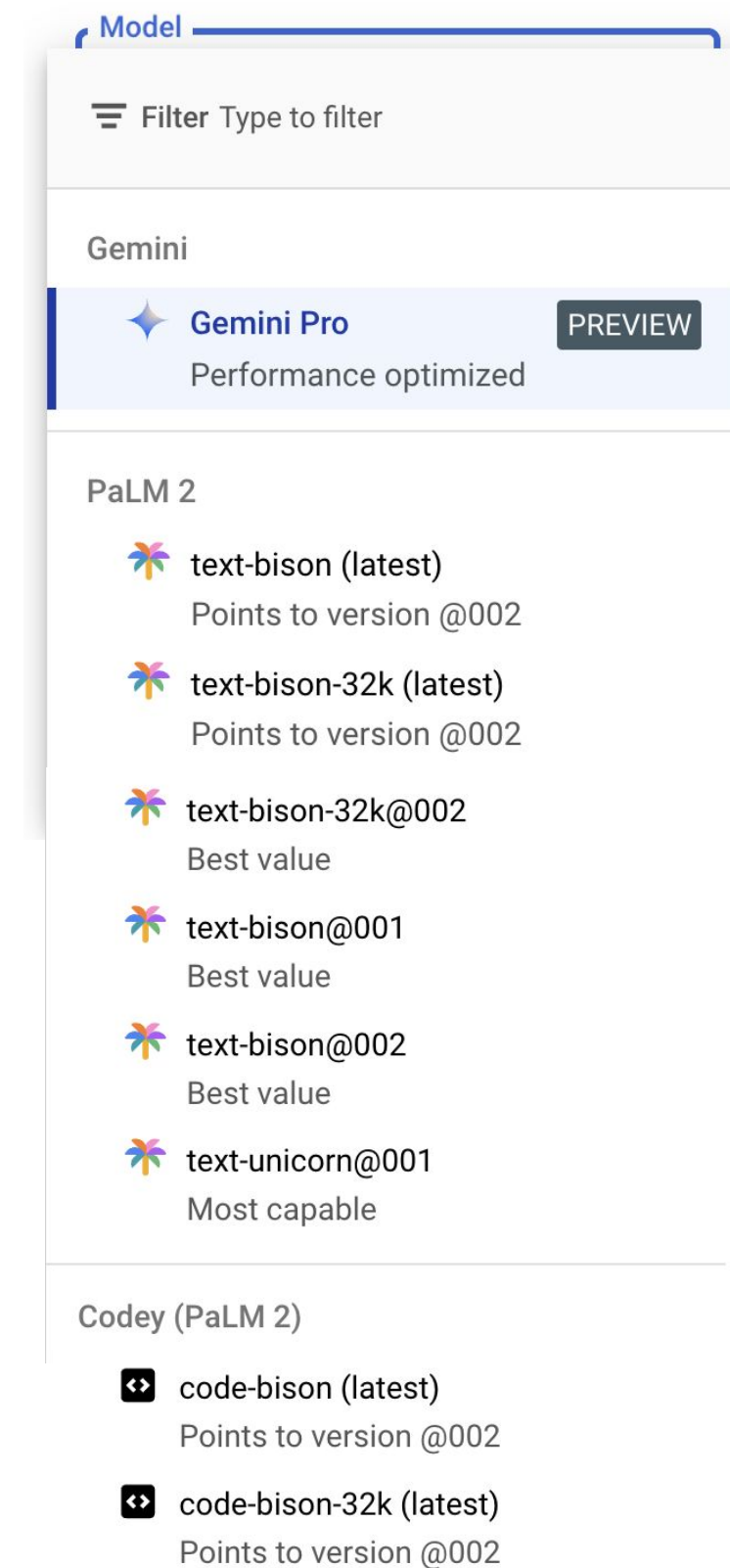
resource "google_compute_network" "default" {
  name = var.network_name
}

resource "google_compute_subnetwork" "us-central1" {
  name          = var.subnetwork_name
  network       = google_compute_network.default.name
  region        = var.region
  ip_range      = var.ip_range_us_central1
}

resource "google_compute_subnetwork" "us-east4" {
  name          = var.subnetwork_name
  network       = google_compute_network.default.name
  region        = var.region
  ip_range      = var.ip_range_us_east4
}
```


In Vertex AI Studio, choose the right model for text or code

- Use different model versions depending on the use case
 - The latest versions are more capable, but also more expensive
- PaLM 2 is for text
 - text-bison (latest) is the best, but most expensive
 - text-bison@### is not the latest, but cheaper
- Codey is for code
 - code-bison@### is the best
 - code-gecko@### is less expensive
- Gemini is for multimodal
- There are smaller models that will run entirely on mobile devices when building apps



Select a chat to have a conversation, use Codey to talk about code

USER

Write me the code to create a Google Cloud Storage bucket

AI

To create a Google Cloud Storage bucket, you can use the following code:

```
import google.cloud.storage as gcs

# Create a client
client = gcs.Client()

# Create a bucket
bucket = client.create_bucket('my_bucket')
```

This code will create a bucket named my_bucket in your Google Cloud Storage account. You can then use this bucket to store data.

USER

How about using a shell script

AI

Sure, here is the code for creating a Google Cloud Storage bucket using a shell script:

```
#!/bin/bash

# Create the bucket
gsutil mb gs://my_bucket
```

This code will create a bucket named my_bucket in your Google Cloud Storage account. You can then use this bucket to store data.

PaLM 2 is good at summarizing articles and documents

✓ Prompt

Summarize the following Wikipedia article about Grace Hopper:

Grace Brewster Hopper (née Murray; December 9, 1906 – American computer scientist, mathematician, and United States Navy rear admiral. [1] One of the first programmers of the Harvard Mark I computer, Hopper was one of the first programmers of computer programming who invented one of the first languages of computer programming. She was the first to devise the theory of machine-independent programming. Hopper created the first programming language, FLOW-MATIC, which she created using the theory of machine-independent programming. She extended it to create COBOL, an early high-level programming language that is still used today.

Prior to joining the Navy, Hopper earned a Ph.D. in both mathematics and mathematical physics from Yale University and was a professor of mathematics at Vassar College. Hopper attempted to enlist in the Navy during World War II.

✓ Response

☒ Markdown

Grace Hopper was an American computer scientist and United States Navy rear admiral. She was one of the first programmers of the Harvard Mark I computer, and she was a pioneer in the development of computer programming languages. She is credited with coining the term "debugging".

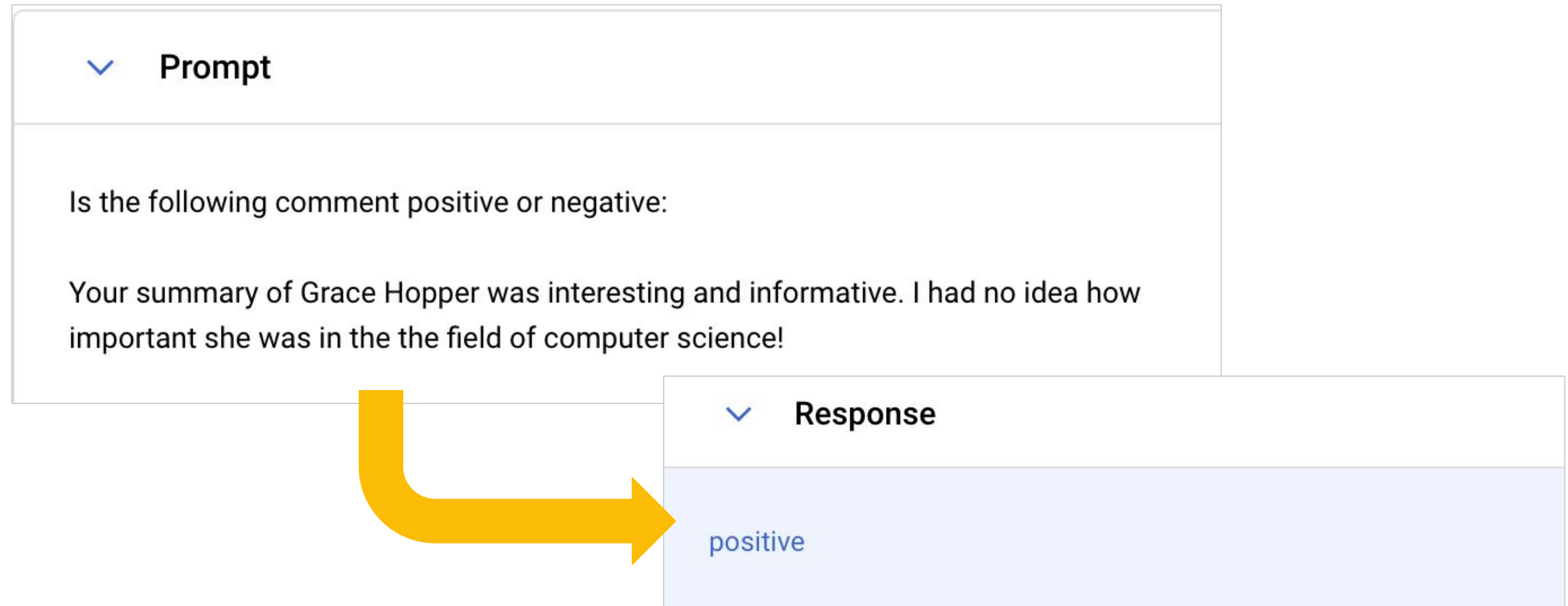
Hopper was born in New York City in 1906. She earned a Ph.D. in mathematics from Yale University in 1934. She began her career as a professor of mathematics at Vassar College, but she left in 1943 to join the United States Navy Reserve.

Hopper served on the Mark I computer programming staff headed by Howard H. Aiken. She worked on the development of the UNIVAC I computer, and she was a key figure in the development of the COBOL programming language.

Hopper retired from the Navy in 1986, but she continued to work as a consultant for Digital Equipment Corporation. She died in 1992 at the age of 85.

Hopper was a pioneer in the field of computer science, and she made significant contributions to the

How about sentiment analysis...



Ideation...

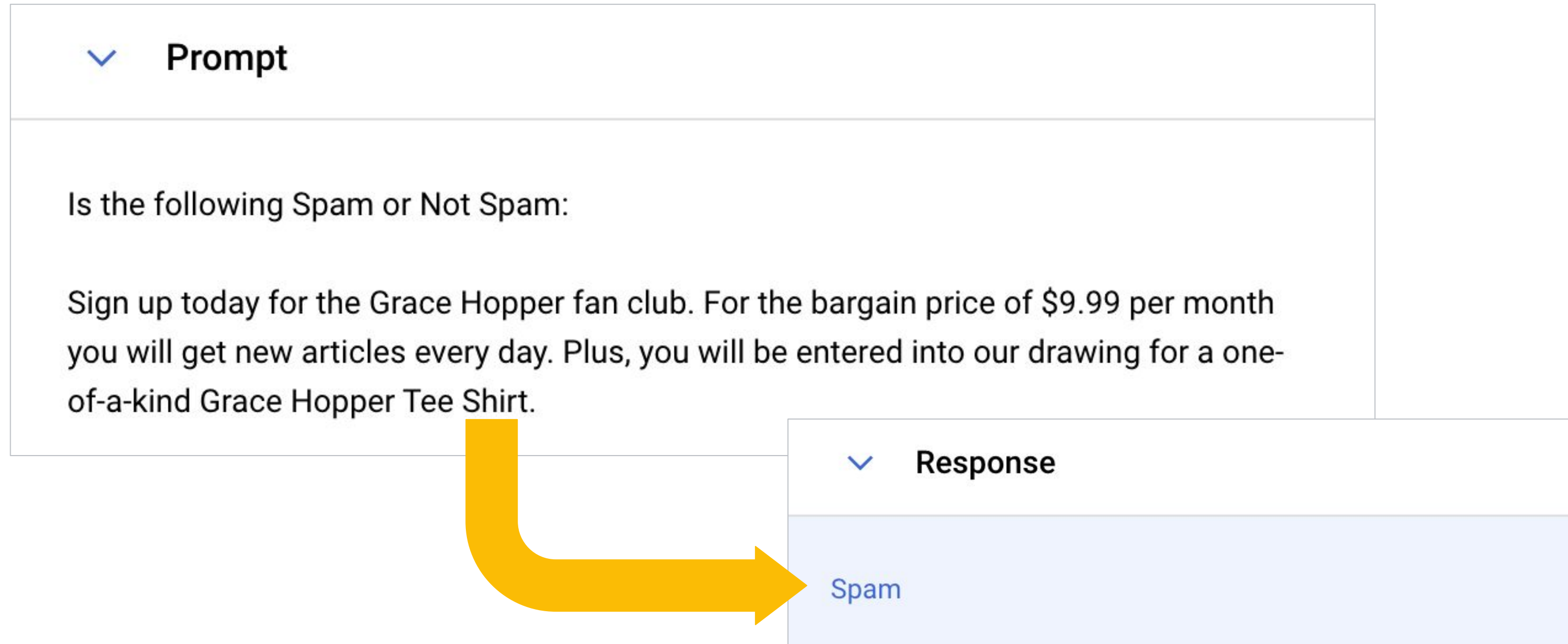
✓ Prompt

I wrote an article about Grace Hopper. What are some good titles?

✓ Response

- Grace Hopper: A Pioneer in Computer Science
- Grace Hopper: The Woman Who Invented COBOL
- Grace Hopper: A Life of Ingenuity and Inspiration
- Grace Hopper: A True Renaissance Woman
- Grace Hopper: A Legacy of Leadership and Innovation

Classification...



Extraction...

▼ Prompt

Extract the important dates from the following Wikipedia article about Grace Hopper

Grace Brewster Hopper (née Murray; December 9, 1906 – American computer scientist, mathematician, and United States Navy rear admiral. [1] One of the first programmers of the Harvard Mark I computer, Hopper was one of the first women of computer programming who invented one of the first languages for computers, COBOL. She was first to devise the theory of machine-independent programming, and she was the first to create the FLOW-MATIC programming language she created using the theory. She later extended to create COBOL, an early high-level programming language that is still used today.

Prior to joining the Navy, Hopper earned a Ph.D. in both mathematics and mathematical physics from Yale University and was a professor at Yale.

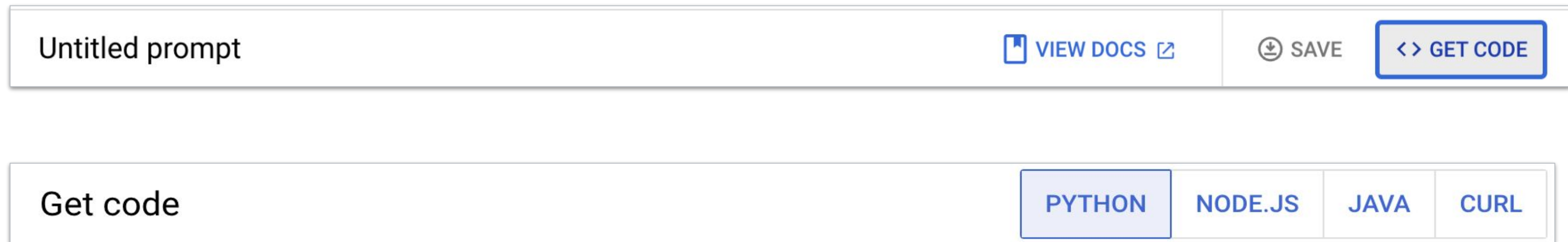
▼ Response

Important dates in Grace Hopper's life:

- * December 9, 1906: Grace Brewster Murray is born in New York City.
- * 1928: Graduates from Vassar College with a bachelor's degree in mathematics and physics.
- * 1930: Earns a master's degree from Yale University.
- * 1934: Earns a Ph.D. in mathematics from Yale University.
- * 1943: Joins the United States Navy Reserve.
- * 1944: Serves on the Mark I computer programming staff headed by Howard H. Aiken.
- * 1949: Joins the Eckert–Mauchly Computer Corporation as a senior mathematician.
- * 1952: Develops the first compiler, which is written for the A-0 System.
- * 1954: Named the company's first director of automatic programming.
- * 1966: Retires from the Naval Reserve with the rank of commander.
- * 1967: Recalled to active duty for a six-month period that turns into an indefinite assignment.

Ok, but how do I use this in my app?

- In Vertex AI Studio, click the **View Code** button
 - Returns the code in Python, Node.js, Java or CURL



Python code example

View code

PYTHON

Use this script to request a model response in your application.

1. Set up the [Vertex AI SDK for Python](#)
2. Use the following code in your application to request a model response

```
import vertexai
from vertexai.language_models import TextGenerationModel

vertexai.init(project="vertex-ai-dar", location="us-central1")
parameters = {
    "temperature": 0.2,
    "max_output_tokens": 256,
    "top_p": 0.8,
    "top_k": 40
}
model = TextGenerationModel.from_pretrained("text-bison@001")
response = model.predict(
    """Is the following Spam or Not Spam:

Hi Doug, I read this article from Wikipedia on Grace Hopper. Thought you might
    **parameters
)
print(f"Response from Model: {response.text}")
```


cURL code example

View code

PYTHON PYTHON

Use the command-line interface (CLI) to request a model response

1. Install Google Cloud SDK [Google Cloud SDK](#) if you haven't already or open Cloud Shell and skip to step 3.
2. Run the following command to authenticate using your Google account.

```
$ gcloud auth application-default login
```

3. Enter the following to request a model response

```
API_ENDPOINT="us-central1-aiplatform.googleapis.com"
PROJECT_ID="vertex-ai-dar"
MODEL_ID="text-bison@001"

curl \
-X POST \
-H "Authorization: Bearer $(gcloud auth print-access-token)" \
-H "Content-Type: application/json" \
"https://${API_ENDPOINT}/v1/projects/${PROJECT_ID}/locations/us-central1/publishers/google/models/${MODEL_ID}:predict" \
$' {
  "instances": [
    {
      "content": "Is the following Spam or Not Spam:

Hi Doug, I read this article from Wikipedia on Grace Hopper. Thought you might be interested."
    }
  ],
  "parameters": {
    "temperature": 0.2,
    "maxOutputTokens": 256,
    "topP": 0.8,
    "topK": 40
  }
}
```

But I want a different language... Ask Codey!

▼ Prompt

Can you convert the following Python code into Go?

import vertexai
from vertexai.language_models import TextGenerationModel

vertexai.init(project="vertext-ai-dar", location="us-central1")
parameters = {
 "temperature": 0.2,
 "max_output_tokens": 256,
 "top_p": 0.8


▼ Response Markdown

```
package main

import (
    "fmt"

    "google.golang.org/api/vertexai/v1alpha1"
)

func main() {
    // Initialize the client.
    client, err := vertexai.NewClient(vertexai.ClientConfig{
        ProjectID: "vertext-ai-dar",
        Location:  "us-central1",
    })
    if err != nil {
        panic(err)
    }
}
```



You can explore sample use cases and prompts from Vertex AI Vertex AI Studio

Prompt examples

Try a sample prompt to test a foundation model with a practical use case

Just getting started? See [Introduction to prompt design](#) to learn more.

Summarization

Freeform

Financial table summariz...

Summarize key insights from a financial report table with heavy numerical info.

OPEN

Freeform

Support chat summary

Summarize a customer support call.

OPEN

Freeform

Support chat summary - ...

Summarize a customer support chat in French.

OPEN

Freeform

Article summary

Summarize a news article.

OPEN

Freeform

Support call summary

Summarize a customer support call from the agent perspective.

OPEN

Freeform

Support call next steps

Summarize actions taken during a custom support call and agent next steps.

OPEN

Structured

Transcript summary

Summarize a transcript using examples.

OPEN

Structured

Dialog summary

Summarize a conversation using examples.

OPEN

Structured

Hashtag tokenization

Create and tokenize hashtags.

OPEN

Structured

Title generation

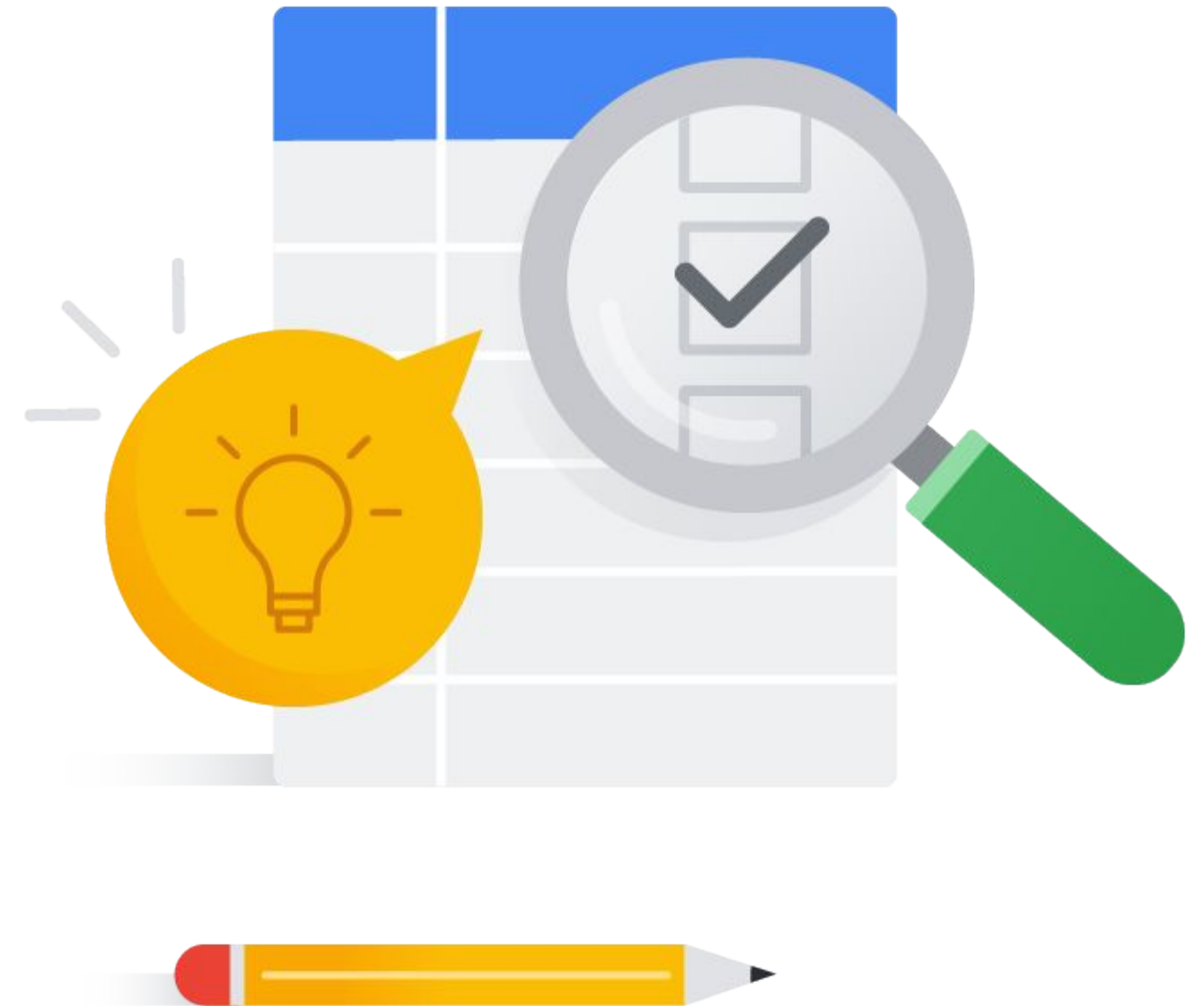
Create a title for an article using examples.

OPEN

Lab

🕒 30 min ⚙️

Getting started with Vertex AI Studio's User Interface



Topics

01	Vertex AI Studio
02	Designing and testing prompts
03	Data governance in Vertex AI Studio



AI Data Governance

Google Cloud's approach to governance of customer data for Cloud AI Large Models and Generative AI



We have no access to customer data, in accordance with [GCP Terms](#) and [Cloud Data Processing Addendum](#).



We're committed to transparency, compliance with regulations like the EU General Data Protection Regulation (GDPR), and privacy best practices.



We will not use customer data to train our models



Security Gen AI Risk Pillars

01

Privacy & Safety

Transparency on how Google LLMs operate is core to the Google's mission for Responsible AI

02

Prompt Hacking

Large Language Models can be manipulated to output responses that are not aligned with its objectives

03

Sec Operations

Gen AI can be leveraged to monitor for malicious activities

04

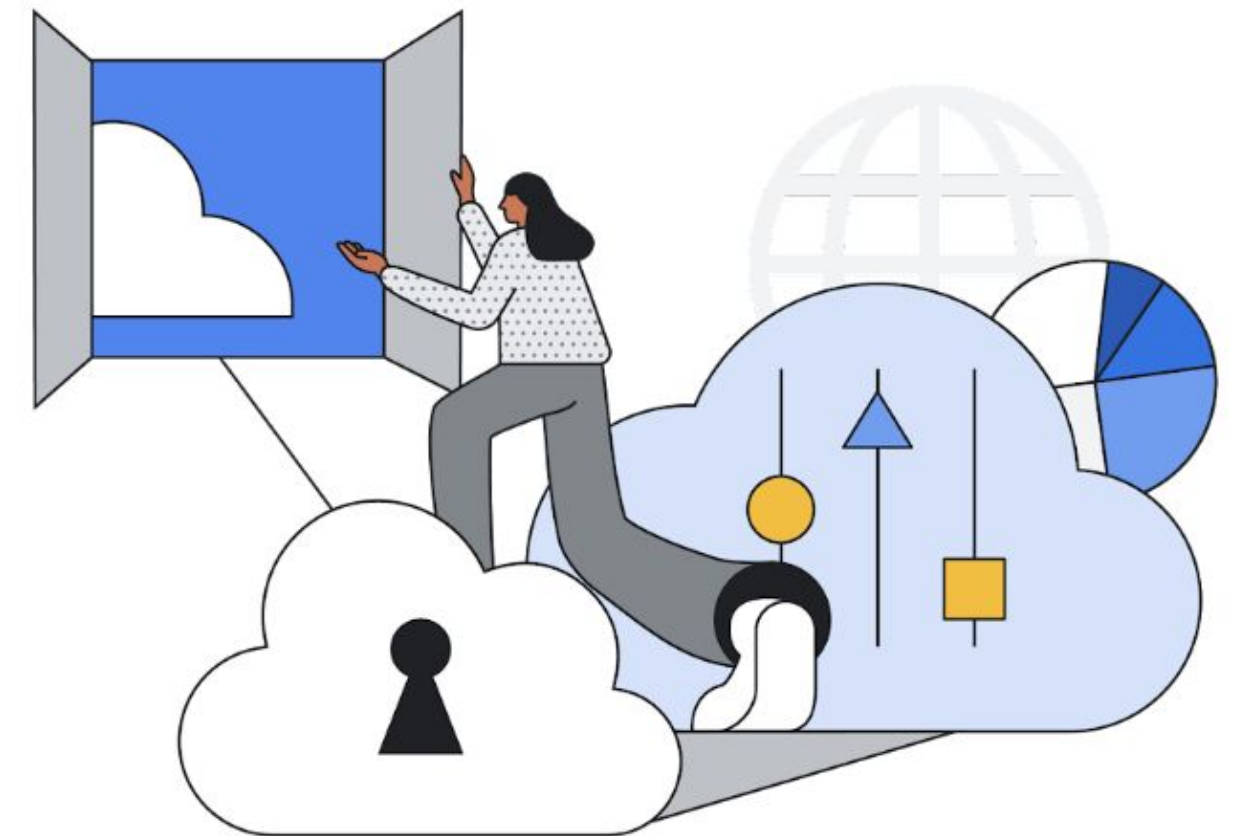
Cloud Resources

Gen AI workloads require the same security controls as traditional workloads



Customer data is kept private during prompt design

- Customer Data is encrypted in-transit when a request is submitted to a foundational model
 - Uses industry-standard cryptographic protocols to protect the confidentiality of customer data
- The foundation model processes customer data to generate a response
 - Does not retain the data for any other purpose
- Google does not use customer data to train its models without the express consent of its customers



Model tuning data is not retained by Google

Fine tuning does not rebuild foundational models

- Creates additional parameters, called “adapter weights”
- Passed to foundational model during inference

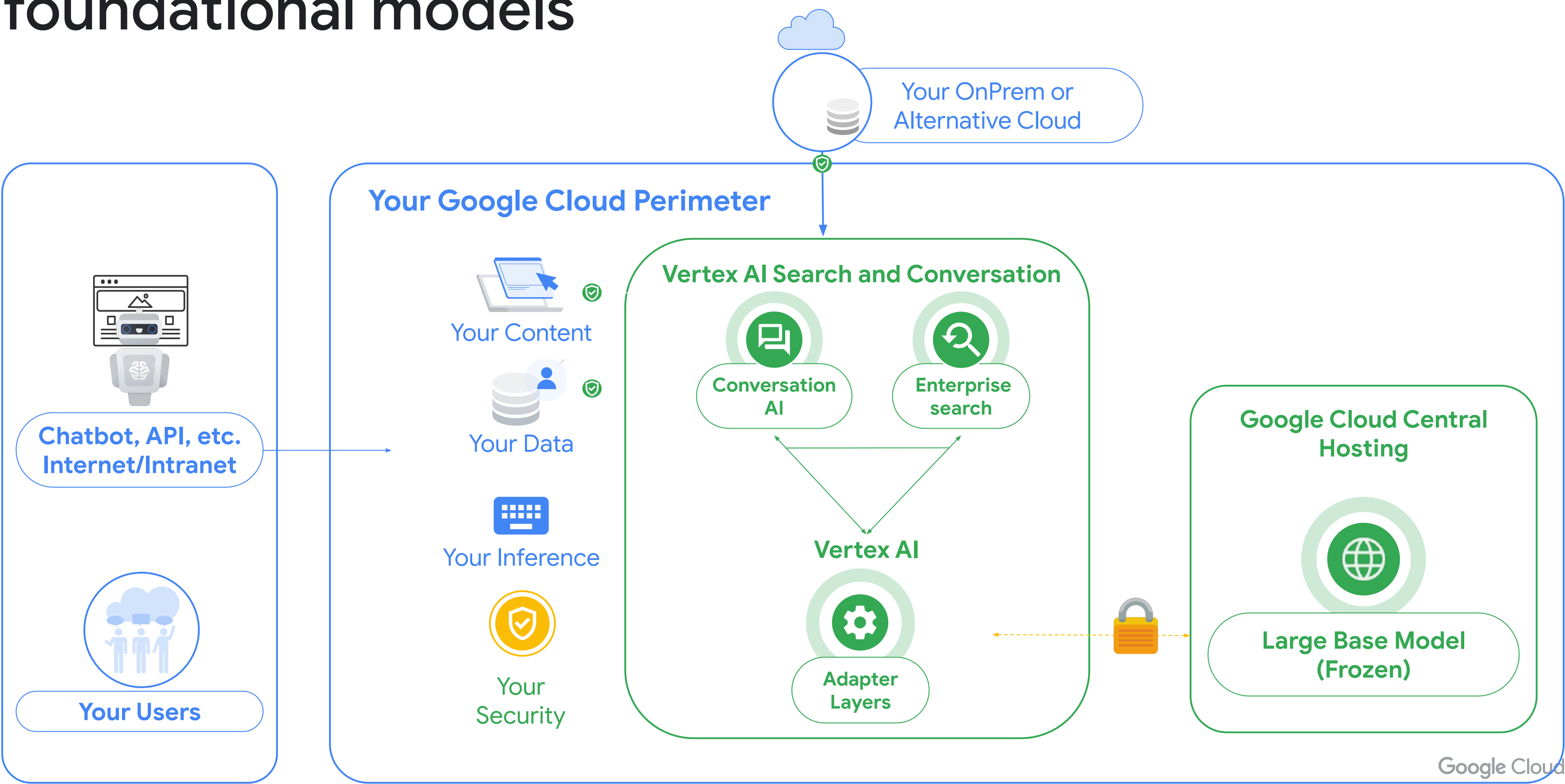
Customers have sole access to tuned models

- Can control encryption keys
- Can delete the model

Customer data is secure

- All data is encrypted at rest and in-transit
- Prompts and training data aren’t logged

Your fine-tuned models are separated from the foundational models



Content processing in the PaLM API is assessed against a list of safety attributes

- Scores are values from 0.0 to 1.0
 - Rounded to one decimal
- The scores are ML predictions
 - Thus, cannot be relied on for 100% accuracy
- If a response exceeds the safety threshold it is blocked
- If content is blocked, the model will return a canned response
 - e.g. "I'm not able to help with that, as I'm only a language model"

[illegible]

To adjust the likelihood of content being blocked, set the Safety filter threshold attribute in Vertex AI Studio

PaLM

Safety filter threshold

Block few

Safety filter threshold

Block most

Block some

Block few

Gemini

Model

Gemini Pro Vision

Region

us-central1 (Iowa)

Temperature

0

1

0.4

Token limit

1

2048

2048

Add stop sequence

Press Enter after each sequence

SAFETY SETTINGS

> Advanced

Safety settings

You can adjust the likelihood of receiving a model response that could contain harmful content. Content is blocked based on the probability that it's harmful. [Learn more](#)

Hate speech

Block some

Dangerous content

Block some

Sexually explicit content

Block some

Harassment content

Block some

RESET DEFAULTS

SAVE

CLOSE

To use Gen AI responsibly follow these best practices

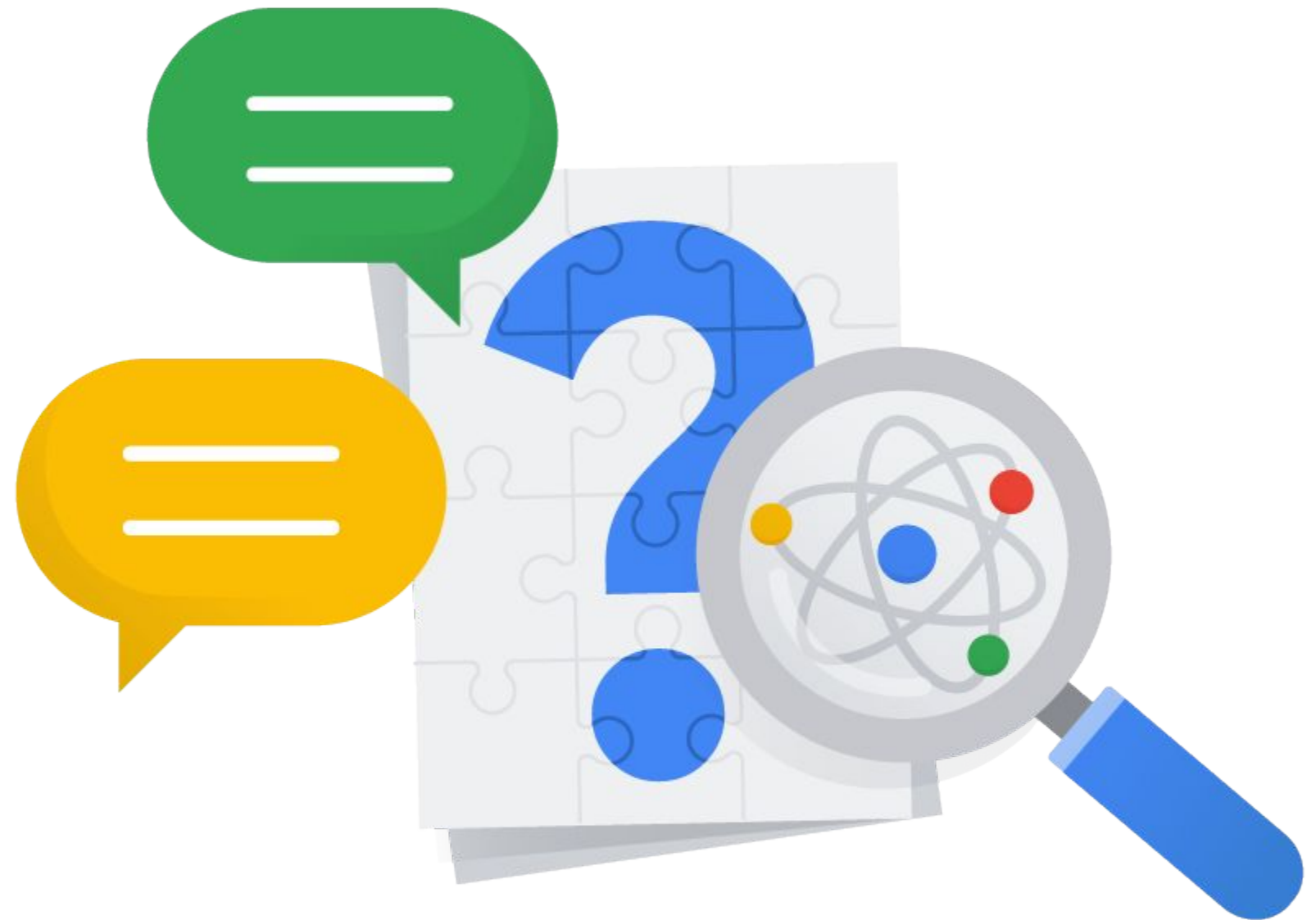
- Assess your application's security risks
- Consider adjustments to mitigate safety risks
- Perform safety testing appropriate to your use case
- Solicit user feedback and monitor content

In this module, you learned to ...

- 01 Use Vertex AI Studio to implement generative AI use cases
- 02 Explore the available models and sample use cases in Vertex AI Studio
- 03 Write and test prompts using the Google Cloud Console
- 04 Protect your data while using Vertex AI Studio



Questions and answers



Quiz question

Match the model to the task

PaLM 2

Codey

Chirp

Imagen

Gemini

Generate text from speech

Generate code

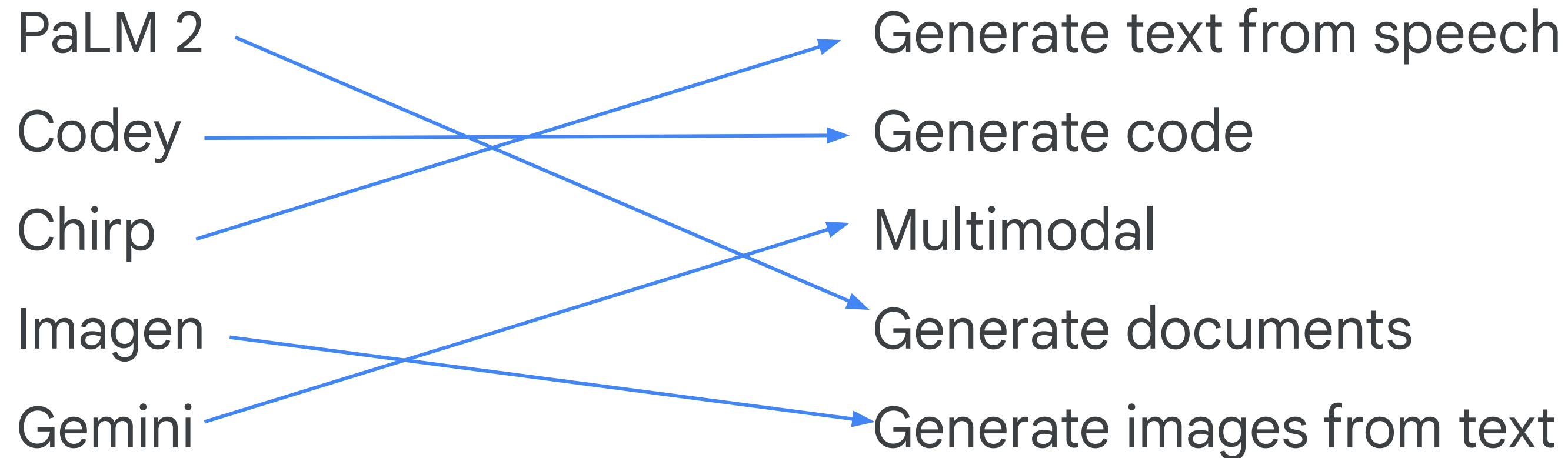
Multimodal

Generate documents

Generate images from text

Quiz question

Match the model to the task



Quiz question

Vertex AI Studio will generate code in which languages? (choose all that apply)

A: Python

B: IPython (Jupyter)

C: Go

D: Node.js

E: cURL

F: Java

Quiz question

Vertex AI Studio will generate code in which languages? (choose all that apply)

A: Python

B: IPython (Jupyter)

C: Go

D: Node.js

E: cURL

F: Java

Quiz question

List some use cases for PaLM 2 language models:

Quiz question

List some use cases for PaLM 2 language models:

Writing

Sentiment analysis

Classification

Extraction

Summarization

Ideation

Chat

