

# PGConf NYC

Building GenAI Applications  
With Postgres and Gemini

Google Cloud



## Speakers



**Cody Fincher**

Staff Solutions Consultant  
Database Black Belts @ Google  
Cloud

Where you can find me:

- LinkedIn: <https://www.linkedin.com/in/cofin>
- Github
  - <https://github.com/cofin>
  - <https://github.com/litestar-org>





## Agenda

- Provide a **introductory guide** on using Gemini with Postgres
- How you can use agents today
  - Agents that enhance Postgres (Gemini CLI & MCP Toolbox)
  - Building agents from scratch with Postgres

---

The magic of **GenAI**  
captured the world's  
imagination

Today organizations are  
looking to turn that  
excitement into  
**business results**



---

Your **AI** needs to be  
fueled by your  
business data

Your **business data**  
needs to be  
unlocked by AI



**85%**

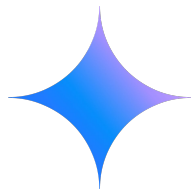
companies experimenting or in  
limited production with GenAI

**5%**

companies that have GenAI  
applications implemented at scale

However,  
**existing data  
platforms** don't  
meet the needs  
of the **AI era**

Source: Wavestone, [2024 Data and AI Leadership Executive Survey](#)



# Gemini 2.5 Ecosystem

|                   | 2.5 Flash-Lite<br><small>THINKING OFF</small>           | 2.5 Flash<br><small>THINKING</small>                      | 2.5 Pro<br><small>THINKING</small>                        |
|-------------------|---|---|---|
| Best for          | High volume cost-efficient tasks                        | Fast performance on everyday tasks                        | Coding and highly complex tasks                           |
| Thinking controls | ✓   | ✓   | ✓   |
| Speed             |   |   |   |
| Performance       |   |   |   |
| Cost              | Input price<br><small>\$/1M tokens (no caching)</small> | \$0.10<br><small>Audio input: \$0.50</small>              | \$0.30<br><small>Audio input: \$1.00</small>              |
|                   | Output price<br><small>\$/1M tokens</small>             | \$0.40  | \$1.25<br><small>\$2.50 &gt; 200k tokens</small>          |
| Availability      | Preview   | Generally available                                       | Generally available                                       |
|                   | Google AI Studio<br>Vertex AI<br>Gemini API             | Google AI Studio<br>Vertex AI<br>Gemini API<br>Gemini app | Google AI Studio<br>Vertex AI<br>Gemini API<br>Gemini app |

# What is Vertex AI?



**Vertex AI**

[cloud.google.com/vertex-ai](https://cloud.google.com/vertex-ai)

- Managed, End-to-End **AI & ML Platform on Google Cloud**
- **Model Garden** - Generative & Predictive AI Models from Google, Partners and Open Source
- **Vertex AI Studio** - Experiment with Models
- **Custom Models** - Training/Prediction Pipelines
- **Vector Search** for Embeddings
- **Colab Enterprise** for Jupyter Notebooks
- **Enterprise-Grade** Security/Reliability



**Gemini CLI is a  
Powerful AI agent  
that wraps your  
terminal**

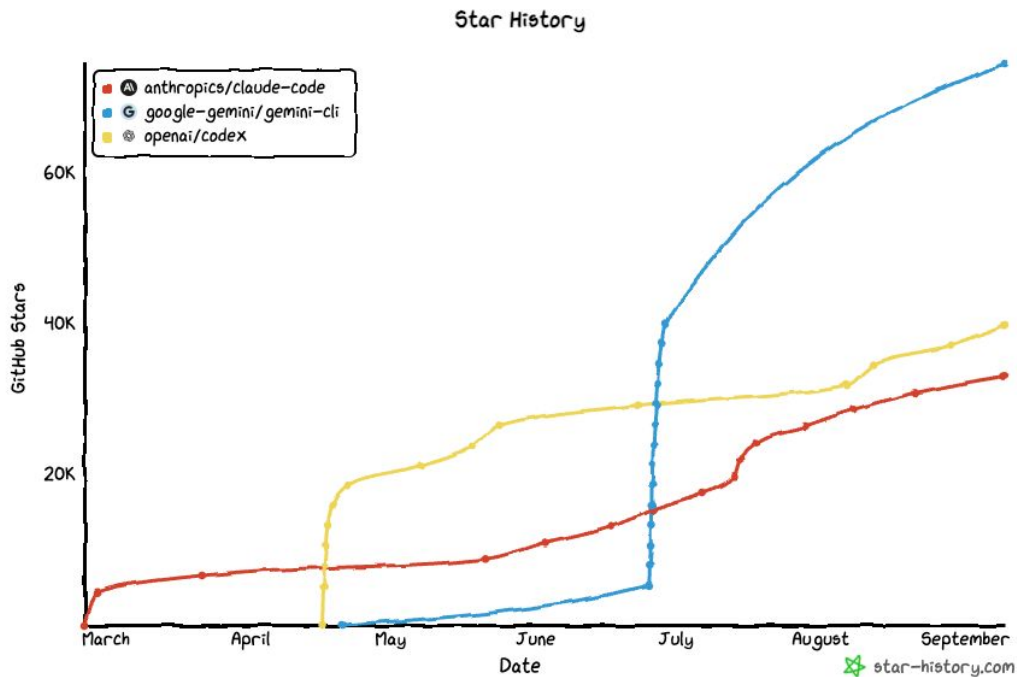


# What can I use it for?



- Coding
- Deep Research
- Task Automation
- Debugging
- Database Analysis
- Custom Workflows
- Extendable with slash commands and MCP integrations

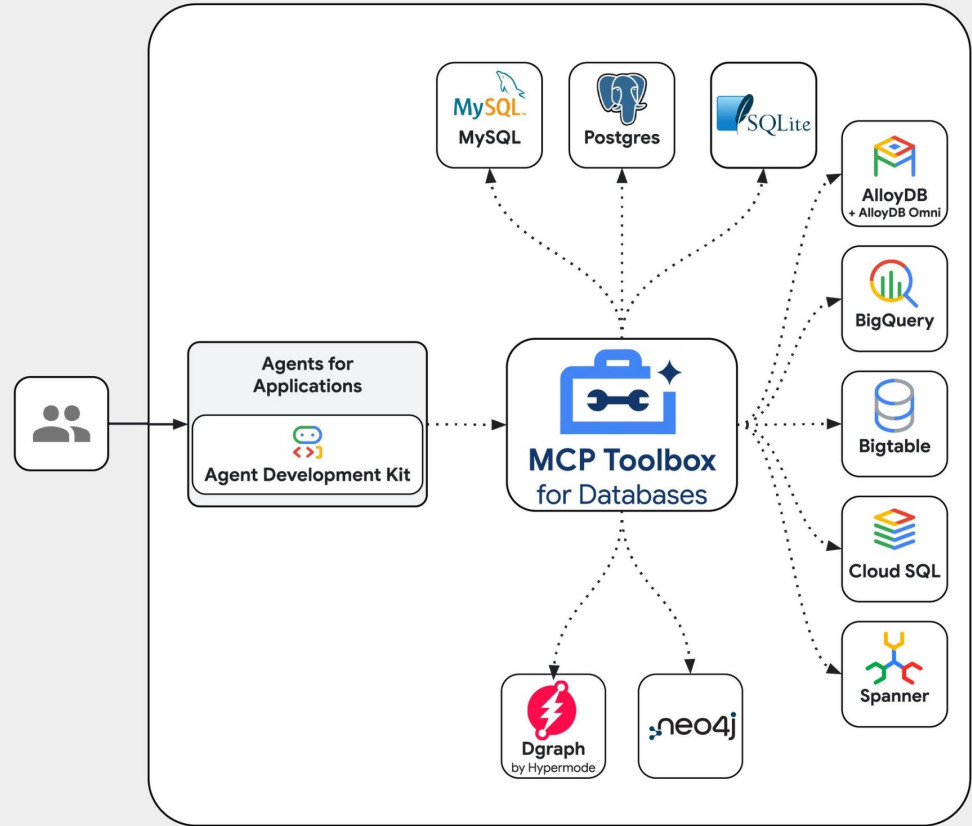
# Gemini CLI is popular!



- Open source AI Agent for
  - Accelerated coding and debugging
  - Streamlined development workflows
  - Effortless data exploration and manipulation
  - Content creation & summarization
- Gemini CLI is easily extensible!!

# MCP Toolbox for Databases

- Standalone tooling to enable MCP integration for agentic services
- Supports multiple databases with a single tool
- Natively integrates as a Gemini CLI extension



[MCP Quickstart](#)

# Gemini CLI + PostgreSQL

**Goal:** *Meet users where they are* and provide an **easy & elegant agentic** experience for **PostgreSQL** via **Gemini CLI**

## Postgres Extensions

- [postgres](#)
- [cloud-sql-postgresql](#)
- [cloud-sql-postgresql-observability](#)
- [alloydb](#)
- [alloydb-observability](#)

# Growing First Party Ecosystem

- [bigquery-conversational-analytics](#)
- [bigquery-data-analytics](#)
- [cloud-sql-mysql](#)
- [cloud-sql-mysql-observability](#)
- [cloud-sql-sqlserver](#)
- [cloud-sql-sqlserver-observability](#)
- [dataplex](#)
- [firestore-native](#)
- [looker](#)
- [Mysql](#)
- [spanner](#)
- [sql-server](#)
- [mcp-toolbox](#)

# Third Party Utilities

- [Zen MCP Server](#)
- [Context7](#)



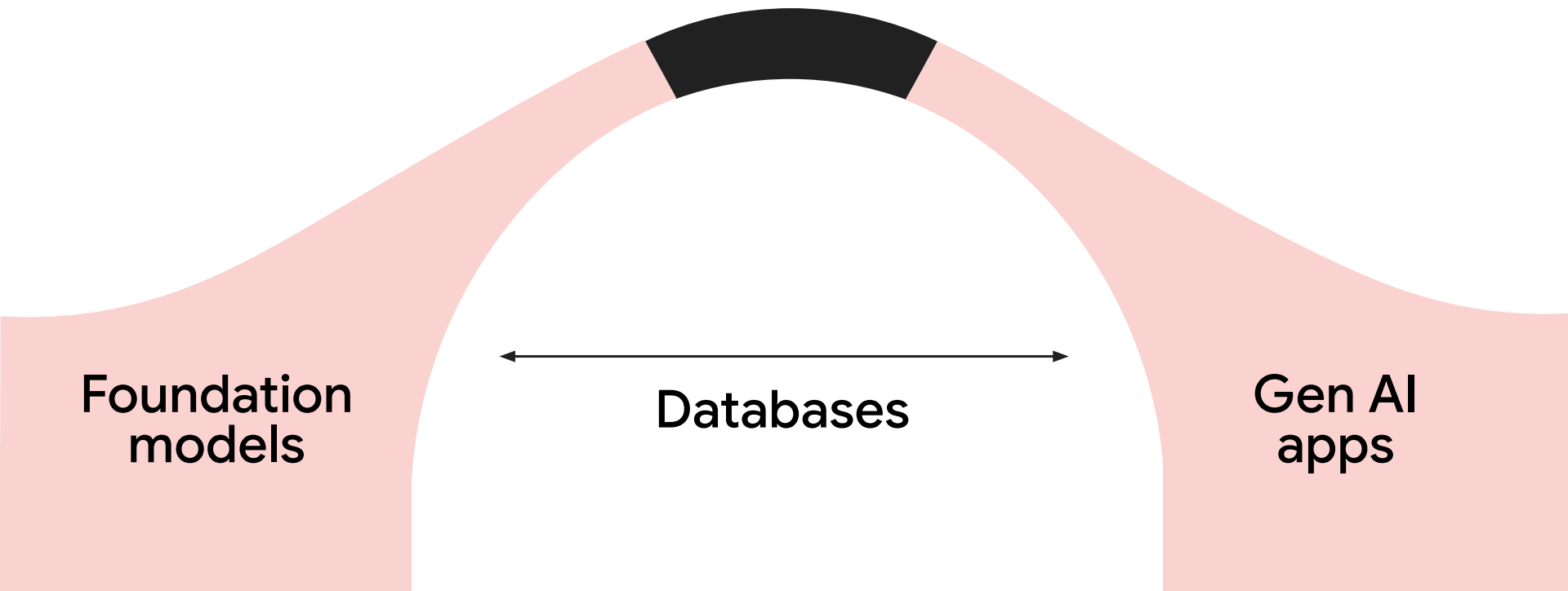
Let's see  
Gemini CLI  
in action

# Building an Application





# Operational data is key to exploiting the power of gen AI



# Google Gen AI SDK

```
pip install google-genai
```

```
from google import genai

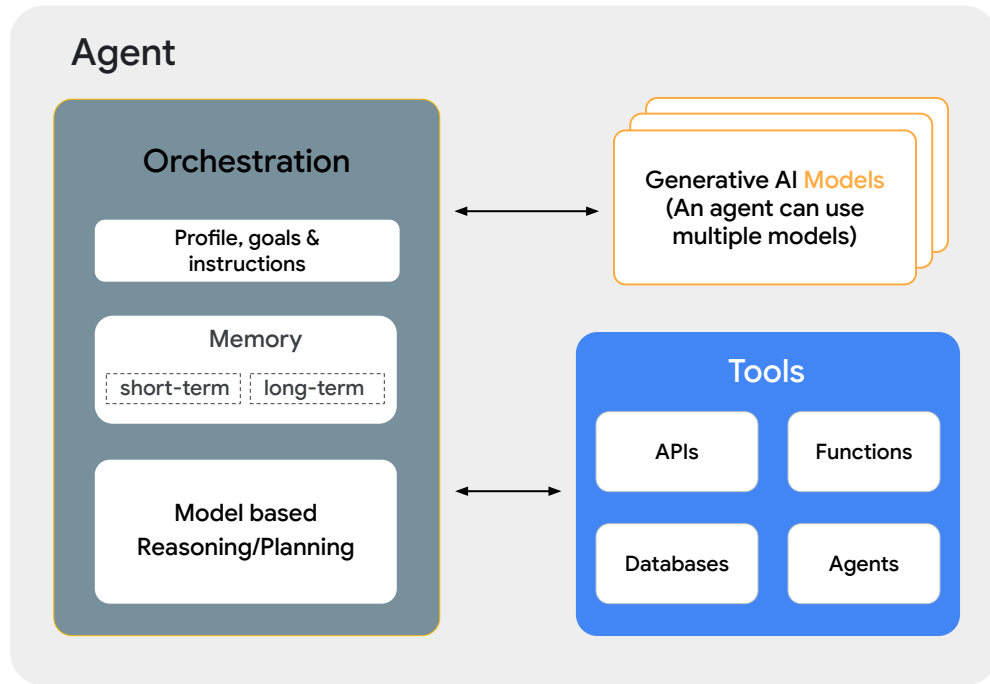
# Gemini Developer API
client = genai.Client(api_key="YOUR_API_KEY")

# Vertex AI API
client = genai.Client(vertexai=True, project="your-project-id", location="us-central1")

response = client.models.generate_content(
    model="gemini-2.5-flash",
    prompt="How does AI work?"
)
print(response.text)
```

[goo.gl/genai-sdk](https://goo.gl/genai-sdk)

# General agents architecture

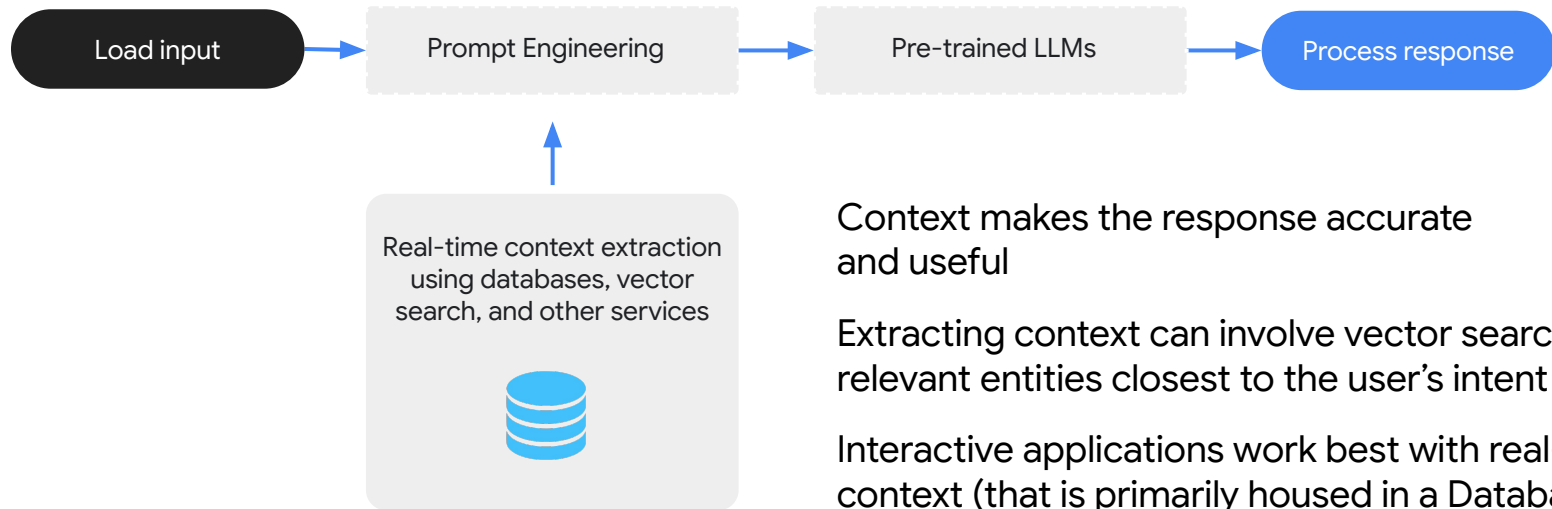


## Key components

- **Model:** Used to reason over goals, determine the plan and generate a response
- **Tools:** Fetch data, perform actions or transactions by calling other APIs or services
- **Orchestration:** Maintain memory and state (including the approach used to plan), tools, data provided/fetched, etc

# Grounding an LLM with RAG

Retrieval Augmented Generation



# What are Embeddings?

Numeric representations (vectors) of words and phrases that help define the way that the phrases relates to various concepts and other words.

## **Traditional Search:**

"espresso" matches only "espresso"

## **Embedding Search:**

"espresso" also finds:

- "strong coffee" (close in meaning)
- "concentrated brew" (similar concept)
- "intense flavor" (related characteristic)

# Populating Embeddings

```
1  async def create_embedding(self, text: str) -> list[float]:
2      """Create embeddings using Vertex AI."""
3      try:
4          # Use the native Vertex AI embedding model
5
6          model = TextEmbeddingModel.from_pretrained(self.embedding_model)
7          embeddings = await model.get_embeddings_async([text])
8
9          if embeddings and len(embeddings) > 0:
10             return cast("list[float]", embeddings[0].values)
11             # Fallback to mock embedding for development
12
13     except Exception:
14         # Log the error and fallback to mock embedding
15         logger.exception("Embedding generation failed, using fallback")
16         return [0.0] * 768 # Standard embedding dimension
17     else:
18         return [0.0] * 768
```

```
- Yummy Waffles: [37.7749, -122.4194, ...]
- Pancakes: [0.2, -0.5, 0.8, ...]
```

```
1  UPDATE product
2  SET embedding = :embedding,
3      embedding_generated_on = SYSTIMESTAMP
4  WHERE id = :id
```

# Vector Similarity

Try searching for coffee products using natural language:

Search Products

407ms Total Time   375.5ms Embedding Generation   32.0ms Vector Search

Found 5 matching products:

## Hazelnut Horizon

A sweet and nutty latte with a hint of hazelnut.

35.5%

## Java Journey

Java Journey - Medium

35.5%

## Hot Chocolate

Rich chocolate drink made with premium cocoa and steamed milk

34.8%

## Midnight Espresso

A shot of espresso with steamed milk, topped with a layer of foam.

34.4%

## Honey Bee Latte


A sweet and creamy latte with a hint of honey.

34.3%



"I need something bold"  
"Caffeine please"  
"How's it going?"

# Intent Exemplars

 **Intent Detection** ×

**DETECTION RESULTS**

|            |                |
|------------|----------------|
| Intent     | PRODUCT_SEARCH |
| Confidence | 79.3%          |

**POSTGRESQL QUERY**

```
WITH query_embedding AS (  
  SELECT intent, phrase,  
         1 - (embedding <=> $1) AS similarity,  
         confidence_threshold,  
         usage_count  
  FROM intent_exemplar)  
SELECT intent, phrase, similarity, confidence_threshold, usage_count  
FROM query_embedding  
WHERE similarity > $2  
ORDER BY similarity DESC  
LIMIT $3
```







Let's  
Demo!

# Gemini CLI Demonstration



```
> GEMINI

Tips for getting started:
1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. ./help for more information.

Using: 2 GEMINI.md files | 4 MCP servers (ctrl+t to view)

> █ Type your message or @path/to/file

~/code/g/postgres-vertexai-demo (main*) no sandbox (see /docs) gemini-2.5-pro (100% context left)
```

# Vector Search Demonstration



Powered by PostgreSQL + pgvector + Google Vertex AI

## PERFORMANCE DASHBOARD

[← Back to Chat](#)

### Key Performance Metrics ?

TOTAL SEARCHES ?

6

→ No change

AVG RESPONSE TIME ?

1562ms

→ Stable

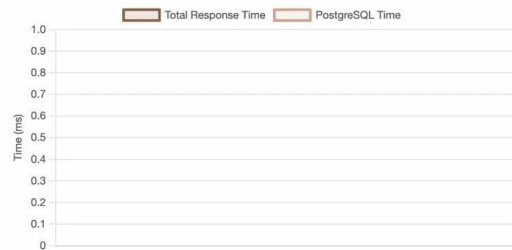
POSTGRESQL VECTOR TIME ?

123ms

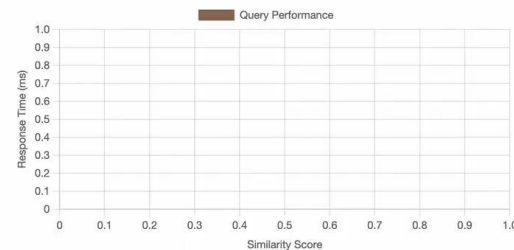
7.9% of total time

### Performance Analytics ?

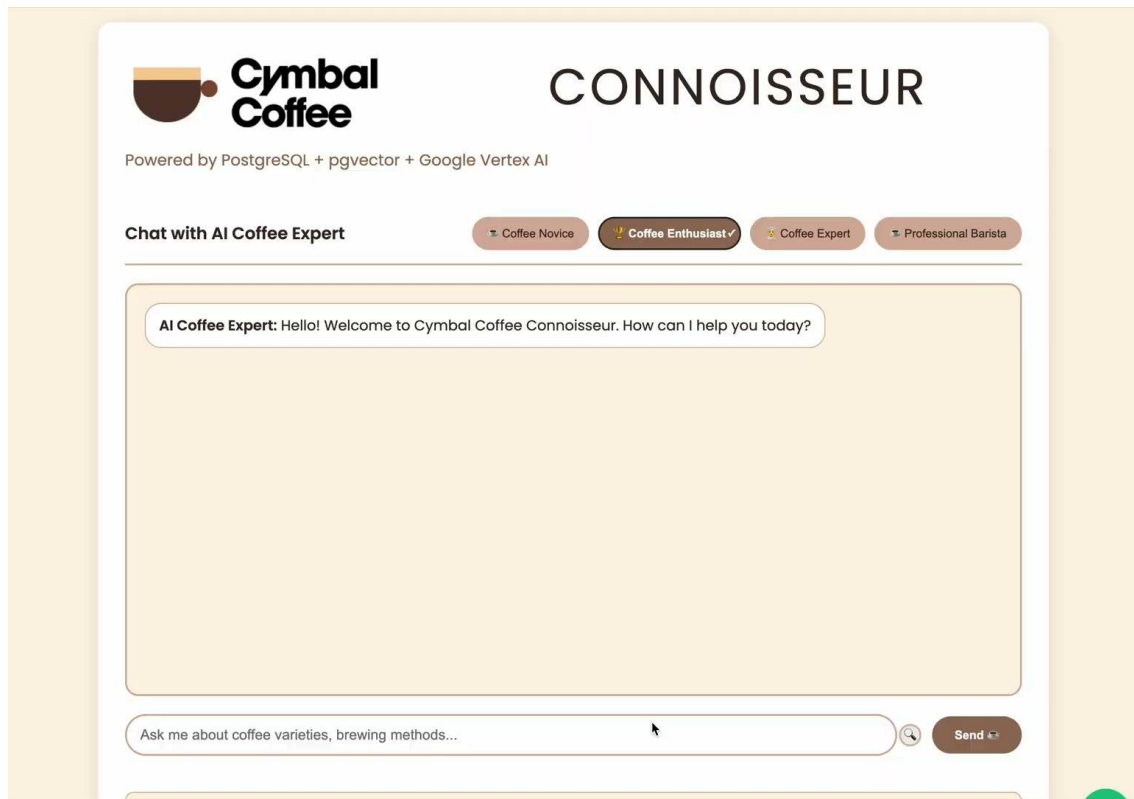
#### Response Time Trends (Last 24 Hours) ?



#### Vector Search Performance ?



# Coffee Chat Demonstration



# AI Cookbook

## Gemini Quickstarts



### Intro to Gemini 2.0 Flash

Get started with Gemini 2.0 Flash in Vertex AI with the Gen AI Python SDK.

[FUNCTION CALLING](#) [GEMINI](#)

[GROUNDING](#) [MULTIMODAL](#)

[PROMPTING](#)

[View on GitHub](#) →



### Intro to Gemini 2.0 Pro

Get started with Gemini 2.0 Pro in Vertex AI with the Gen AI Python SDK.

[FUNCTION CALLING](#) [GEMINI](#)

[GROUNDING](#) [MULTIMODAL](#)

[PROMPTING](#)

[View on GitHub](#) →



### Intro to Gemini 2.0 Flash-Lite

Get started with Gemini 2.0 Flash-Lite in Vertex AI with the Gen AI Python SDK.

[FUNCTION CALLING](#) [GEMINI](#)

[MULTIMODAL](#) [PROMPTING](#)

[View on GitHub](#) →



### Get Started with the Multimodal Live API

Get started with Gemini 2.0 Multimodal Live API in Vertex AI using the Gen AI Python SDK

[GEMINI](#) [LIVE API](#) [MULTIMODAL](#)

[View on GitHub](#) →



### Get Started with Gemini 2.0 Flash Thinking

Get started with Gemini 2.0 Flash Thinking in Vertex AI using the Gen AI Python SDK to get more detailed reasoning and thinking steps.

[GEMINI](#) [MULTIMODAL](#)

[View on GitHub](#) →



### Intro to Prompt Engineering

Learn the essentials and best practices of prompt engineering.

[GEMINI](#) [PROMPTING](#)

[View on GitHub](#) →

## All Tutorials

Filter by: [Categories \(1\)](#) ▾

Search for...

[GEMINI](#)

### Intro to Gemini 2.0 Flash

Get started with Gemini 2.0 in Vertex AI with the Gen AI Python SDK.

[GEMINI](#)

### Intro to Prompt Engineering

[PROMPTING](#)

Learn the essentials and best practices of prompt engineering.

[FUNCTION CALLING](#)

### Function Calling with Gemini

[GEMINI](#)

Connect Gemini to external tools using function calling.

[GEMINI](#)

### Grounding with Gemini

[GROUNDING](#)

Connect Gemini to real-world data from Google Search or Vertex AI Search to improve response quality.

[RAG](#)

[SEARCH](#)

[BATCH PREDICTION](#)

### Gemini Batch prediction

[GEMINI](#)

Use Batch Prediction to run inference on a large number of examples.

[GEMINI](#)

### Long Context Window

Use the Long Context Window to process large amounts of multimodal data.

[GEMINI](#)

### Intro to Context Caching

Use context caching to store frequently used data.

[GEMINI](#)








### Intro to Controlled Generation with the Gemini API

Learn to control Gemini API output formats for easier data processing.



[goo.gle/vertex-ai-cookbook](https://goo.gle/vertex-ai-cookbook)

# Google Cloud Generative AI GitHub

|   | Description   |
|---|---|
| <br><a href="#">gemini/</a>        | Discover Gemini through starter notebooks, use cases, function calling, sample apps, and more.  |
| <br><a href="#">search/</a>        | Use this folder if you're interested in using <a href="#">Vertex AI Search</a> , a Google-managed solution to help you rapidly build search engines for websites and across enterprise data. (Formerly known as Enterprise Search on Generative AI App Builder)   |
| <br><a href="#">rag-grounding/</a> | Use this folder for information on Retrieval Augmented Generation (RAG) and Grounding with Vertex AI. This is an index of notebooks and samples across other directories focused on this topic.   |
| <br><a href="#">conversation/</a>  | Use this folder if you're interested in using <a href="#">Vertex AI Conversation</a> , a Google-managed solution to help you rapidly build chat bots for websites and across enterprise data. (Formerly known as Chat Apps on Generative AI App Builder)  |
| <br><a href="#">language/</a>      | Use this folder if you're interested in building your own solutions from scratch using Google's language foundation models (Vertex AI PaLM API).  |
| <br><a href="#">vision/</a>        | Use this folder if you're interested in building your own solutions from scratch using features from Imagen on Vertex AI (Vertex AI Imagen API). These are the features that Imagen on Vertex AI offers: <ul style="list-style-type: none"><li>• Image generation</li><li>• Image editing</li><li>• Visual captioning</li><li>• Visual question answering</li></ul> |
| <br><a href="#">audio/</a>         | Use this folder if you're interested in building your own solutions from scratch using features from Chirp, a version of Google's Universal Speech Model (USM) on Vertex AI (Vertex AI Chirp API).  |



[goo.gle/gen-ai-github](https://goo.gle/gen-ai-github)

# Thank you



[github.com/google-gemini/gemini-cli](https://github.com/google-gemini/gemini-cli)



[github.com/cofin/postgres-vertexai-demo](https://github.com/cofin/postgres-vertexai-demo)

**Google** Cloud

