

Enhancing Large Language Model Performance via Fine-tuning and Prompt Engineering

Author: Kong Wang Seng

Date: 4 August 2025

Table of Content

[1.0 Introduction](#)

[2.0 Fine tune a model](#)

[2.1 Steps to implement the fine tune model inside other project.](#)

[2.2 Implement the fine tune model inside the project](#)

[3.0 Implement AI by using API from the Open AI platform](#)

[3.1 Optimizing the prompt](#)

[3.2 Initialization needed before use API key](#)

[3.3 Web Search Model](#)

[3.4 Example to do the AI API request](#)

[4.0 Prompt engineering](#)

[4.1 Role for AI](#)

[4.2 Pros and cons](#)

[4.3 Advantage](#)

[4.4 Limitation](#)

[4.5 Example prompt](#)

[4.6 Practice needed for prompt](#)

[5.0 Reference](#)

1.0 Introduction

Fine-tuning involves retraining a pre-trained model on task-specific data, allowing it to internalize new patterns, vocabulary, or behaviors. This approach is computationally intensive but provides high accuracy and flexibility for downstream tasks. In contrast, Prompt Engineering refers to crafting structured inputs or system-level instructions that guide the model to generate desired outputs without altering its weights. This technique is cost-effective, interpretable, and ideal for zero-shot or few-shot learning.

This report explores both approaches in the context of Large Language Model(LLM). It outlines their underlying principles, implementation workflows, advantages, and limitations. It also discusses scenarios in which one method may be preferred over the other, and how hybrid strategies can be employed to balance cost, control, and performance.

2.0 Fine tune a model

Do fine tune on a large language model so it can perform to full fill certain specific task. For example, a large language model was fine tuned with customer service dataset and it can perform well as a customer service chatbot.

2.1 Steps to implement the fine tune model inside other project.

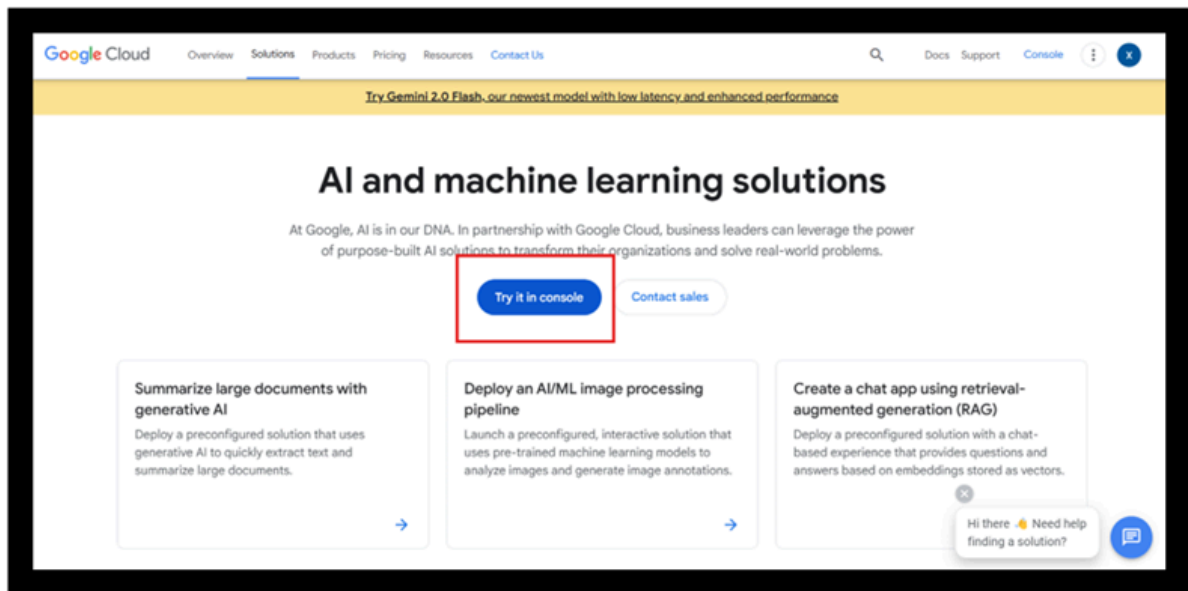


Image 1

Go to [vertex AI](#) click on the button. For the first time user, the billing method needs to be set up first before starting to fine tune. A total of 300 dollars will be provided as free credit with limited time usage.

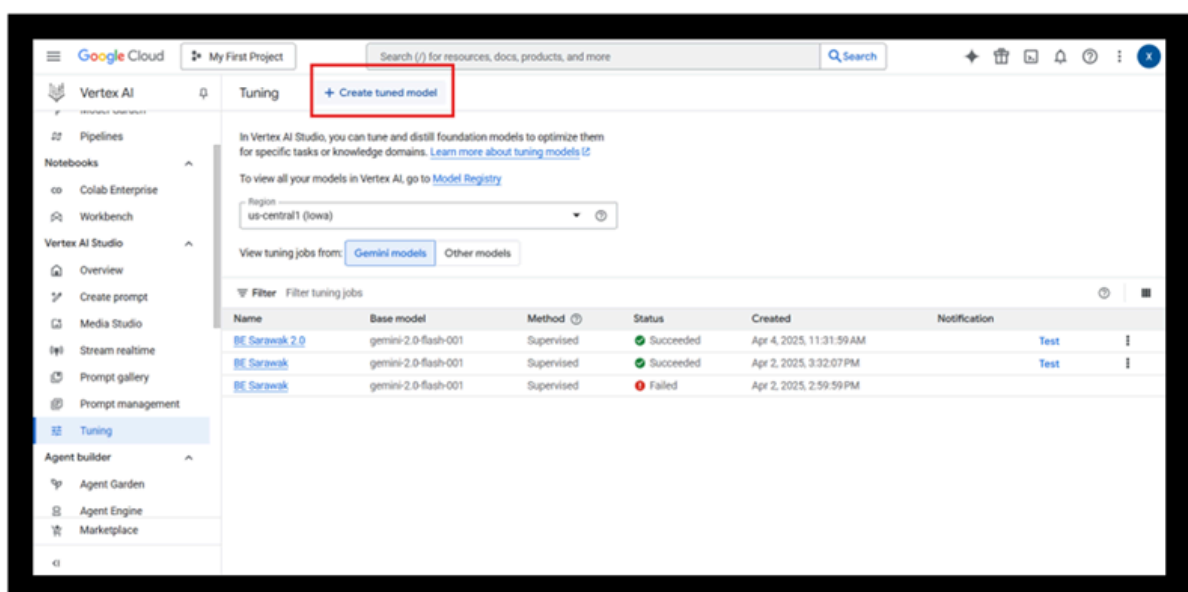


Image 2

Go to the sidebar and find the tuning button. In the tuning page, you can create the new tuned model.

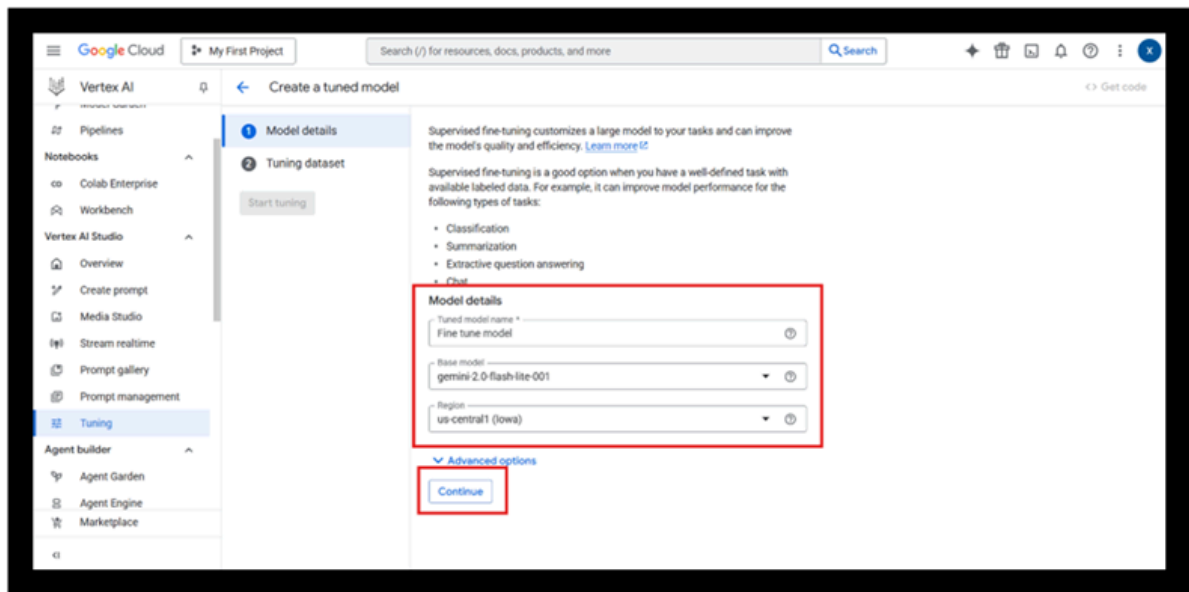


Image 3

Give the model name and you may choose the base model and region. Press continues to proceed.

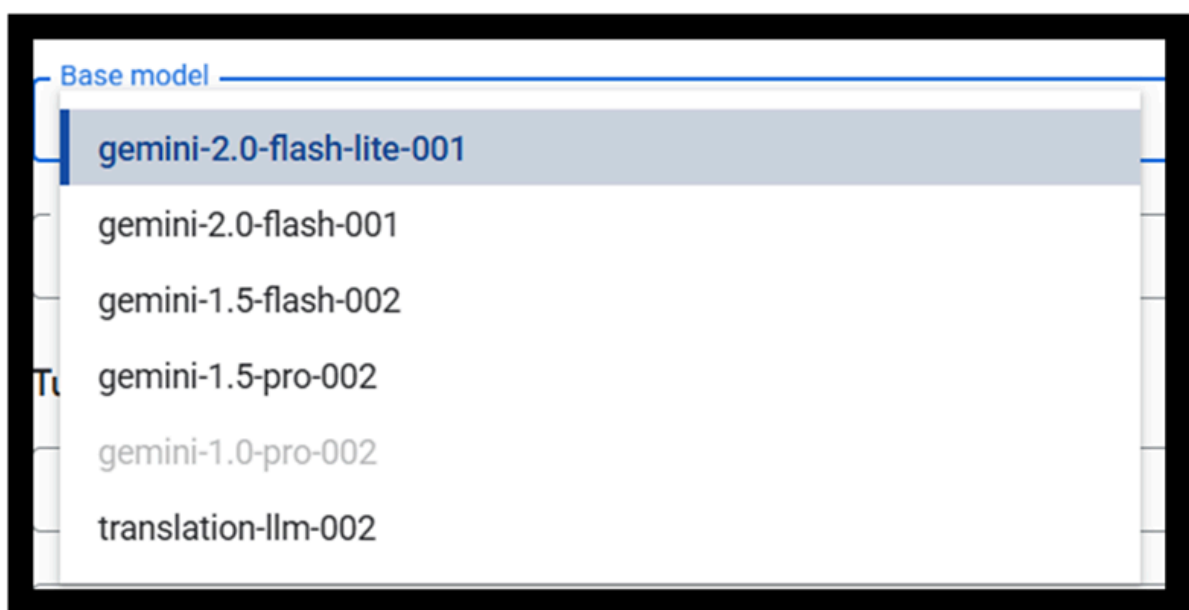
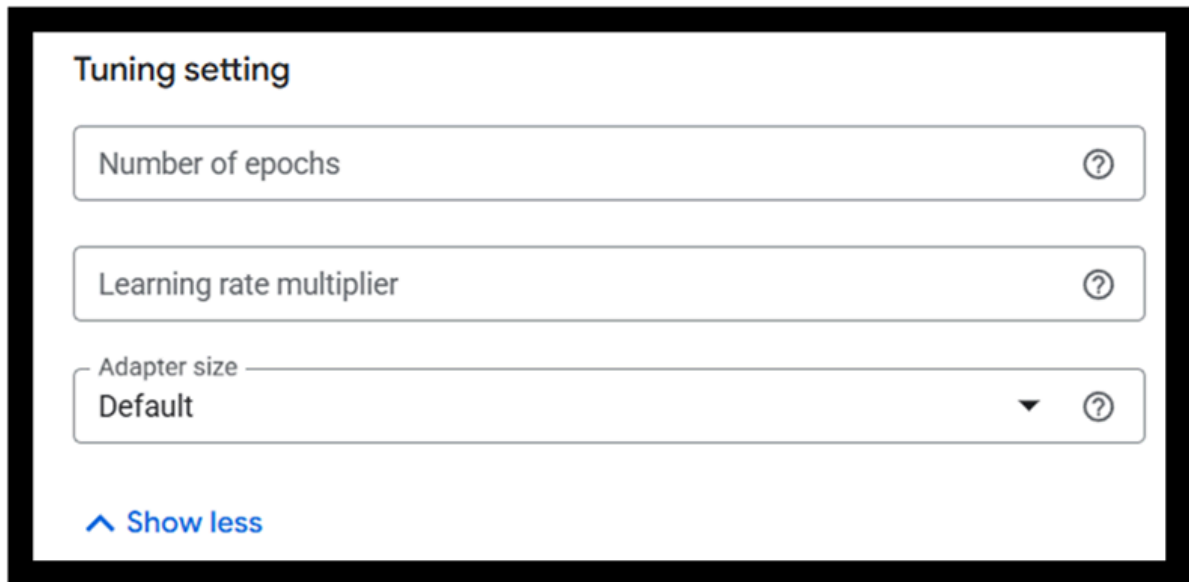


Image 4

You may choose the suitable base model based on your requirement. For the gemini-2.0-flash-001 it is the more powerful and fast response when compare to other.



Tuning setting

Number of epochs

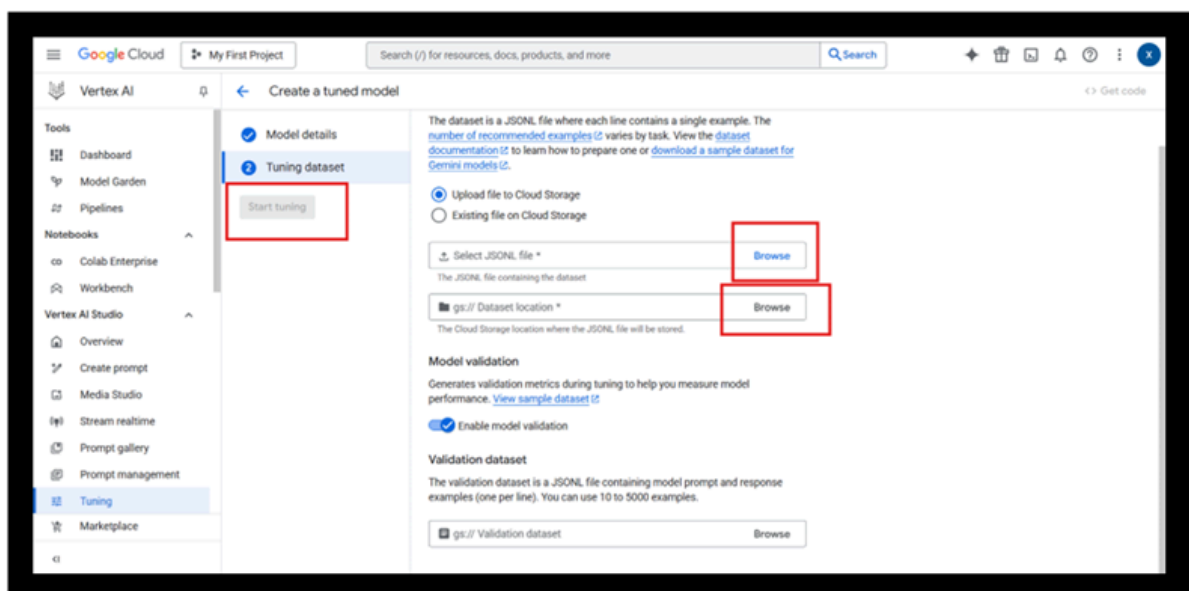
Learning rate multiplier

Adapter size
Default

[Show less](#)

Image 5

You can adjust the tuning setting by clicking on the advance setting button.



Google Cloud | My First Project | Search (/) for resources, docs, products, and more

Vertex AI | Create a tuned model | Get code

Tools: Dashboard, Model Garden, Pipelines, Notebooks, Colab Enterprise, Workbench, Vertex AI Studio: Overview, Create prompt, Media Studio, Stream realtime, Prompt gallery, Prompt management, **Tuning**, Marketplace

Tuning dataset

[Start tuning](#)

The dataset is a JSONL file where each line contains a single example. The number of recommended examples varies by task. View the [dataset documentation](#) to learn how to prepare one or [download a sample dataset for Gemini models](#).

☒ Upload file to Cloud Storage
☐ Existing file on Cloud Storage

Select JSONL file * [Browse](#)

The JSONL file containing the dataset

gs:// Dataset location * [Browse](#)

The Cloud Storage location where the JSONL file will be stored.

Model validation
Generates validation metrics during tuning to help you measure model performance. [View sample dataset](#)

☒ Enable model validation

Validation dataset
The validation dataset is a JSONL file containing model prompt and response examples (one per line). You can use 10 to 5000 examples.

gs:// Validation dataset [Browse](#)

Image 6

```
{"contents": [{"role": "user", "parts": [{"text": ""}]}, {"role": "model", "parts": [{"text": ""}]}]}
```

Upload the dataset and choose the bucket to store the fine tuned data. The data structure needs to be in this format and save in JSONL file for fine tuning.

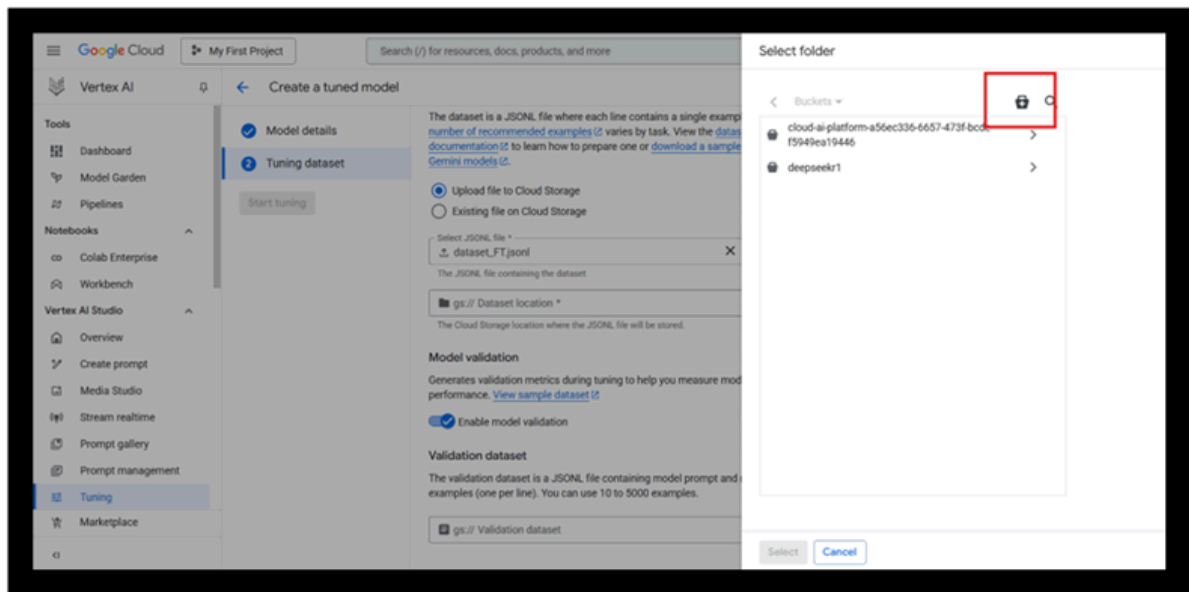


Image 7

To create the data set location, click on the create new bucket button at the right corner.

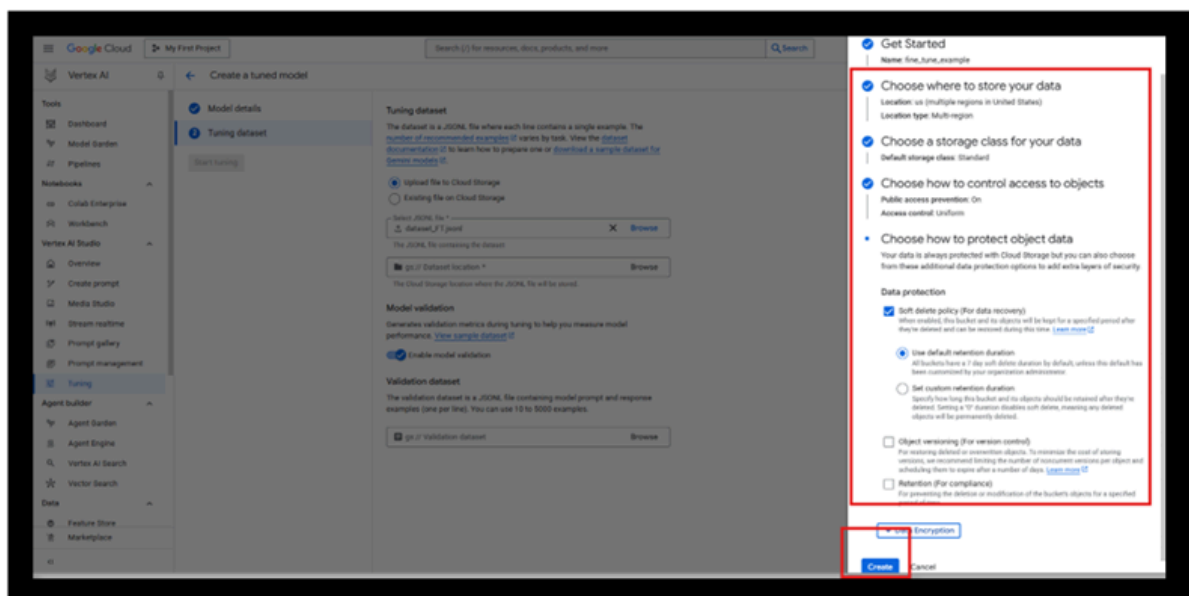


Image 8

You will need to name the bucket to proceed. After choosing the suitable setting, you may create the bucket.

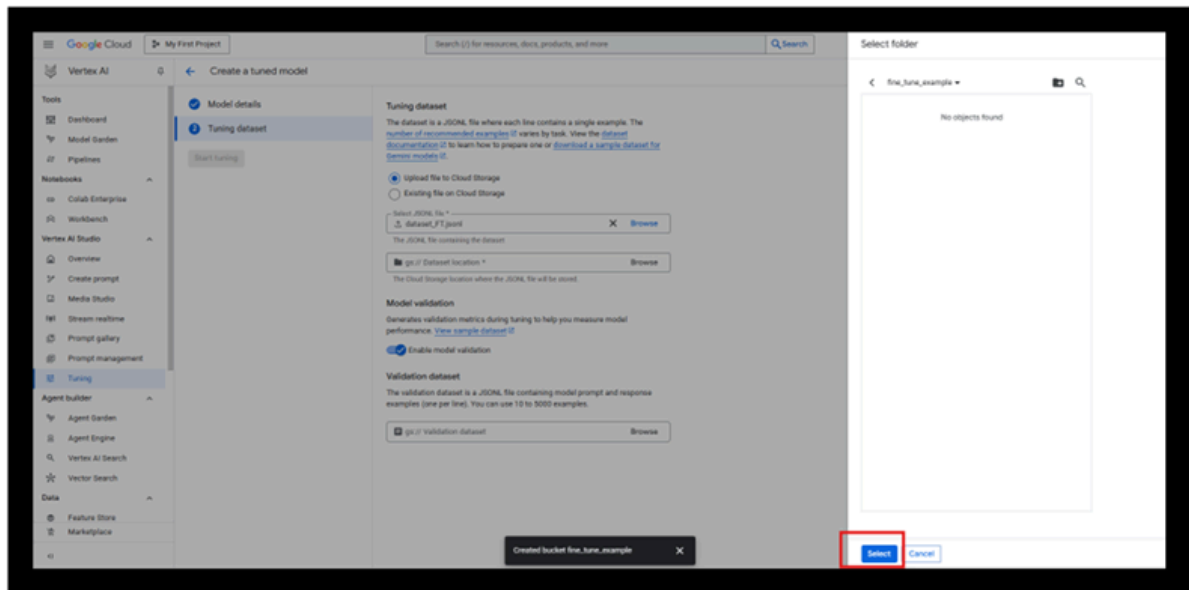


Image 9

Select the created bucket just now.

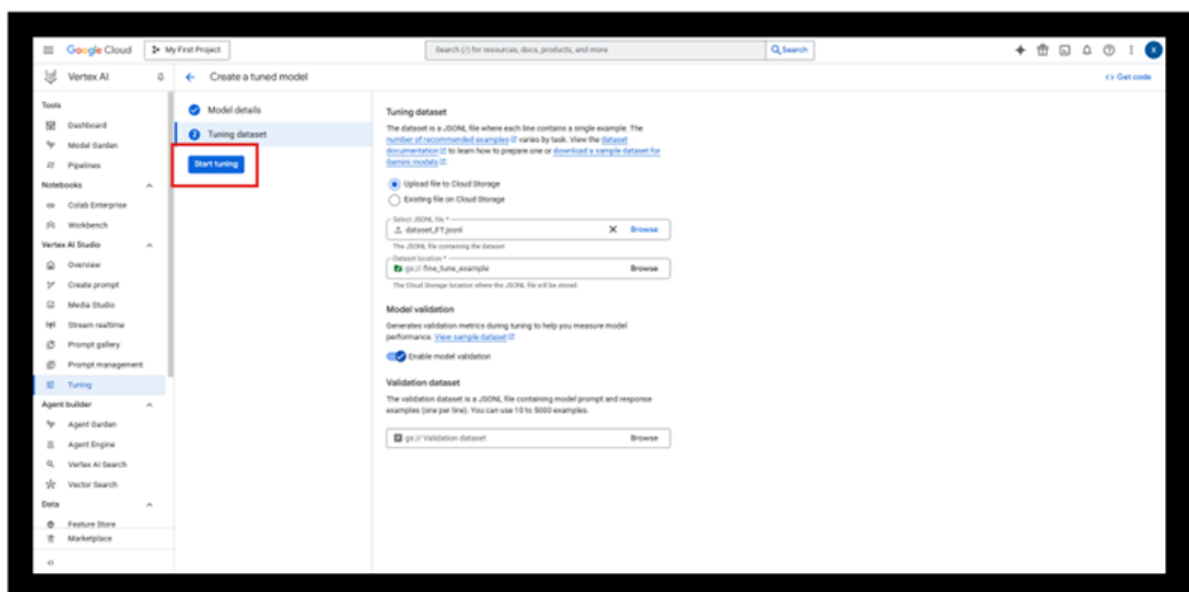


Image 10

Proceed to start fine tuning and wait for the result.

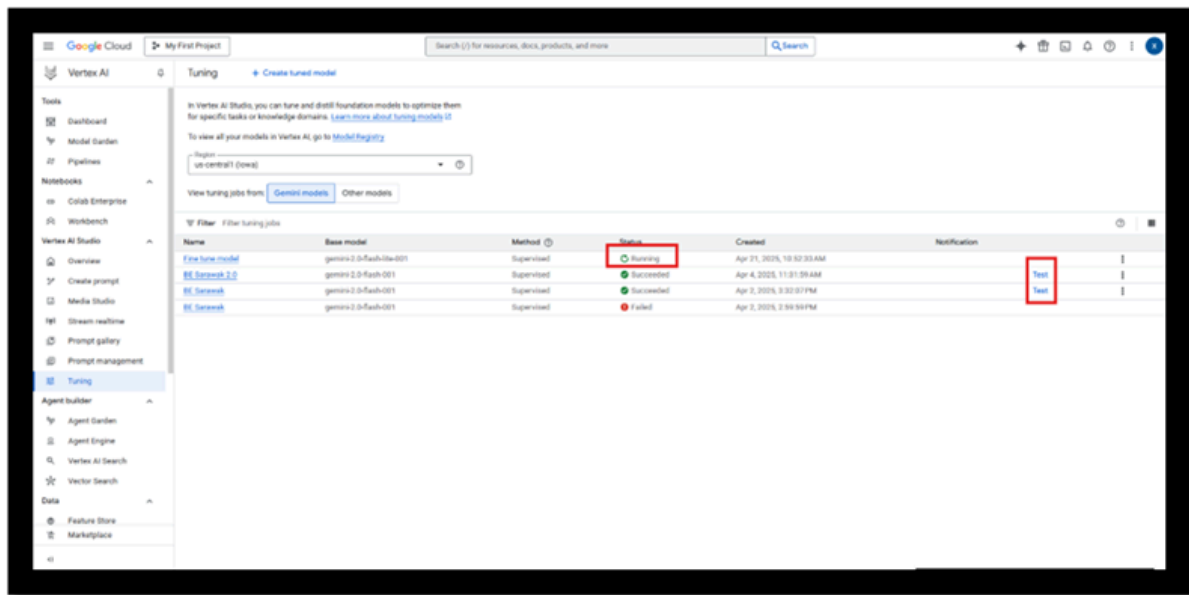


Image 11

After fine tuning, the tuning page will show the current status of the model and you can try the model by clicking on test.

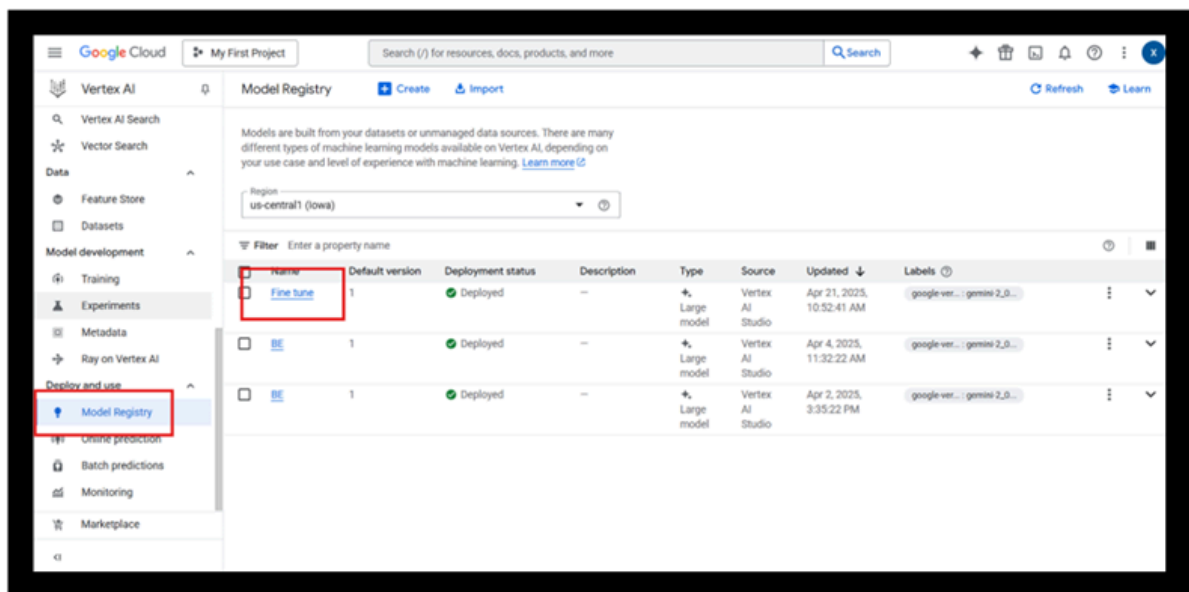


Image 12

Go to the sidebar, find the model registry button and find the model you have fine tuned.

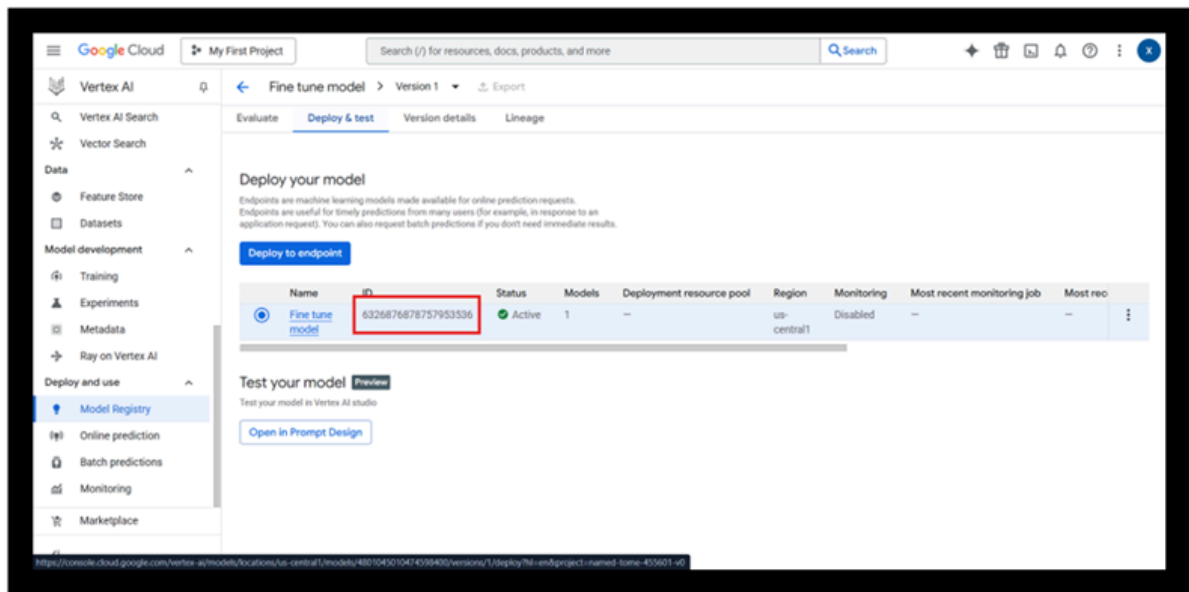


Image 13

By clicking on it, you can get the endpoint ID of the fine tuned model.

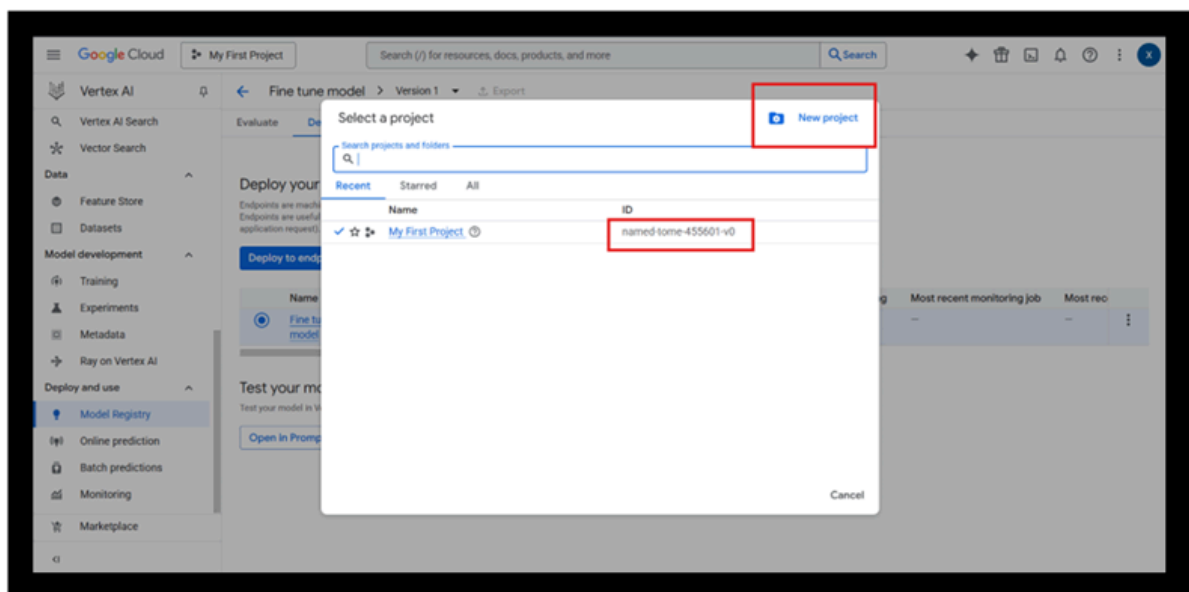


Image 14

At the same time, click on the top left corner “My First Project” and you can get the project ID and create a new project. The deployed fine tune model must be in the same project.

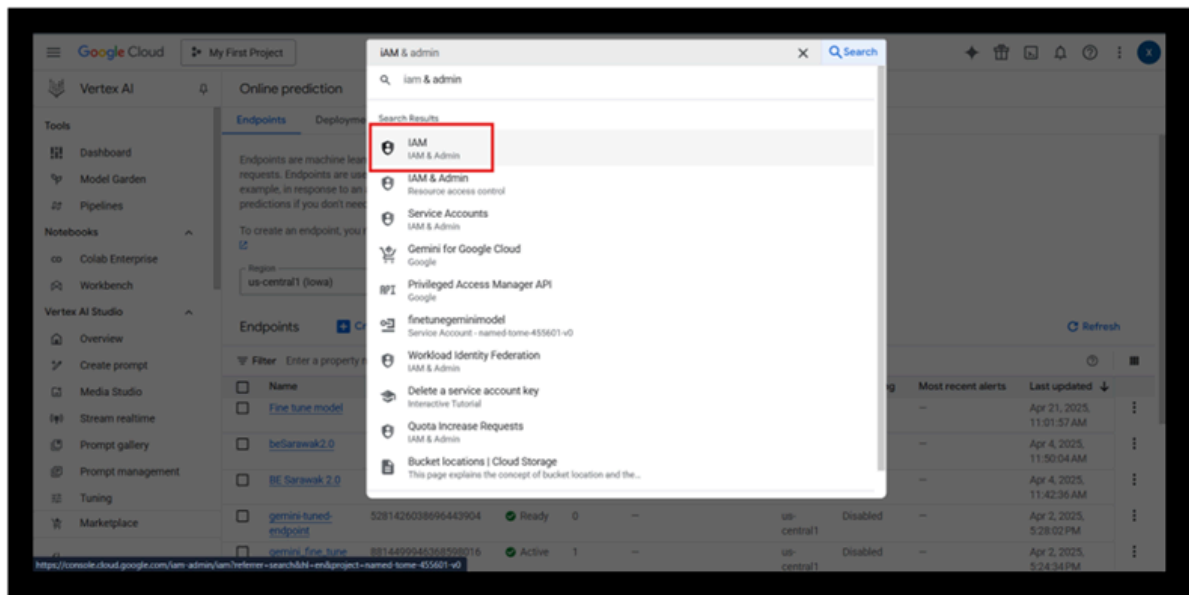


Image 15

Search the IAM from the search bar. Click on it to go to this page.

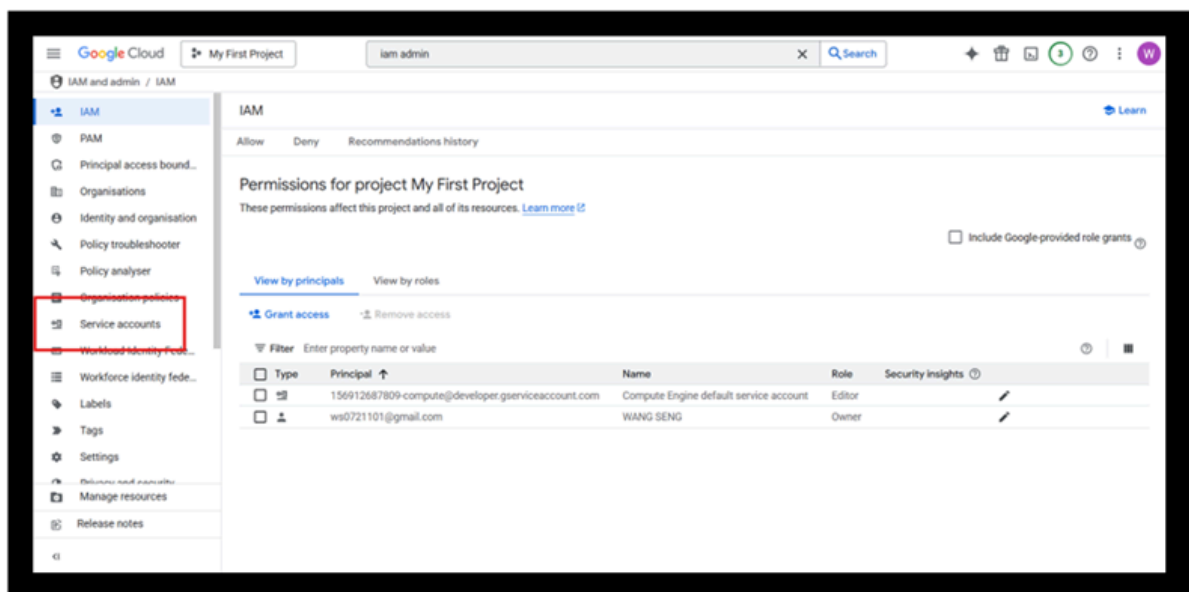


Image 16

From this page, find the service account button at the sidebar. Click on it.

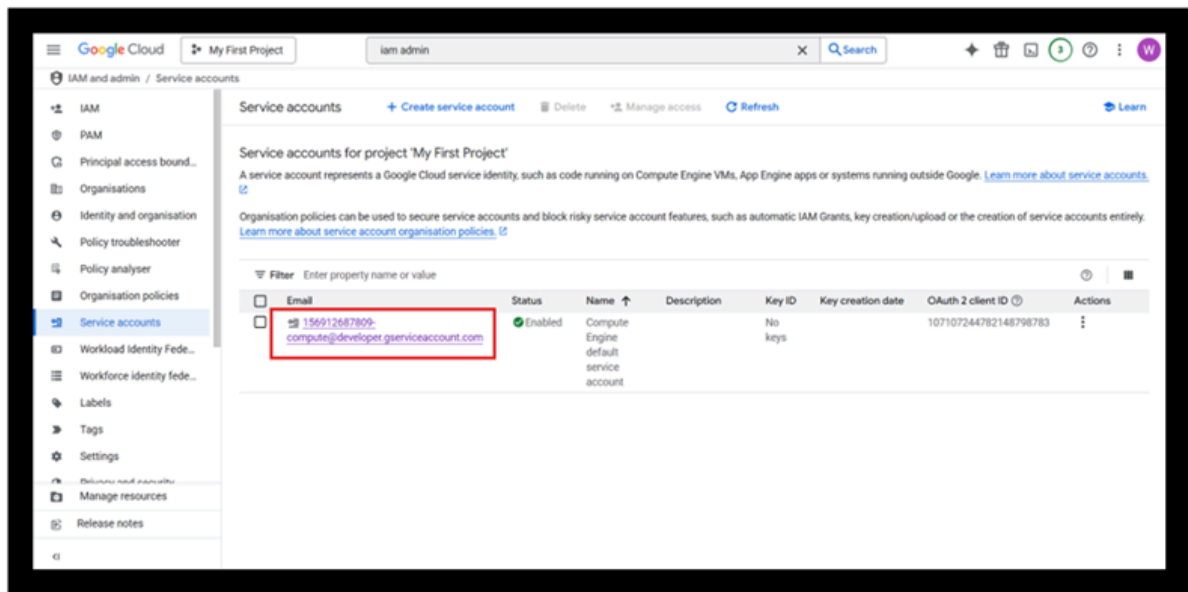


Image 17

You will see the auto generated service account for your project. Click on it.

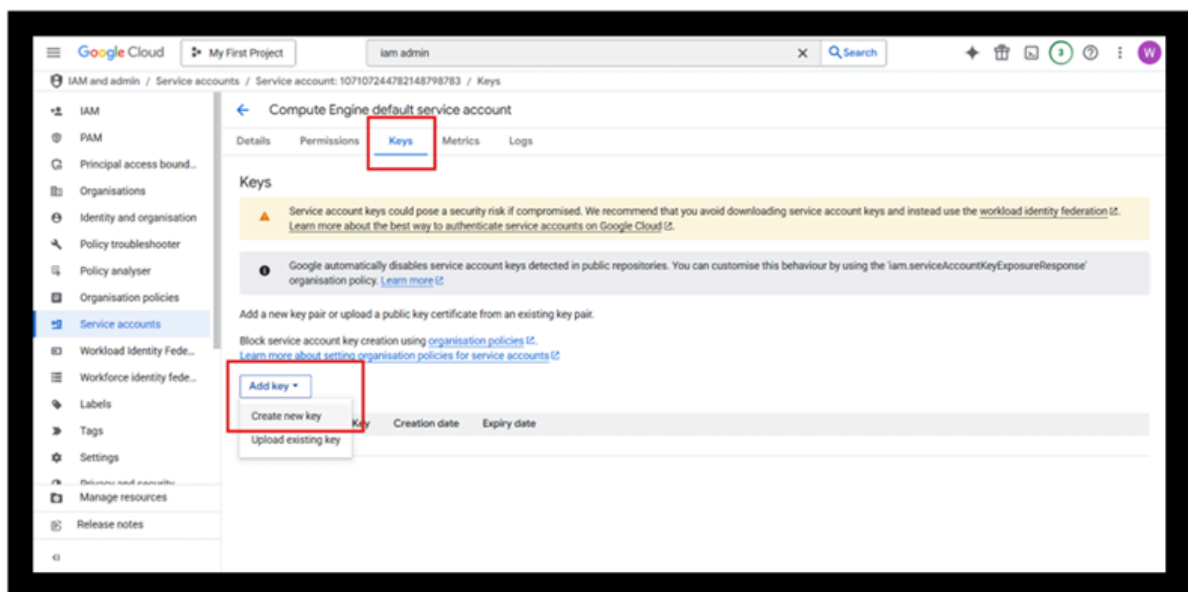


Image 18

Go to the key page and click the Add key button to generate the key file. Click on create new key.

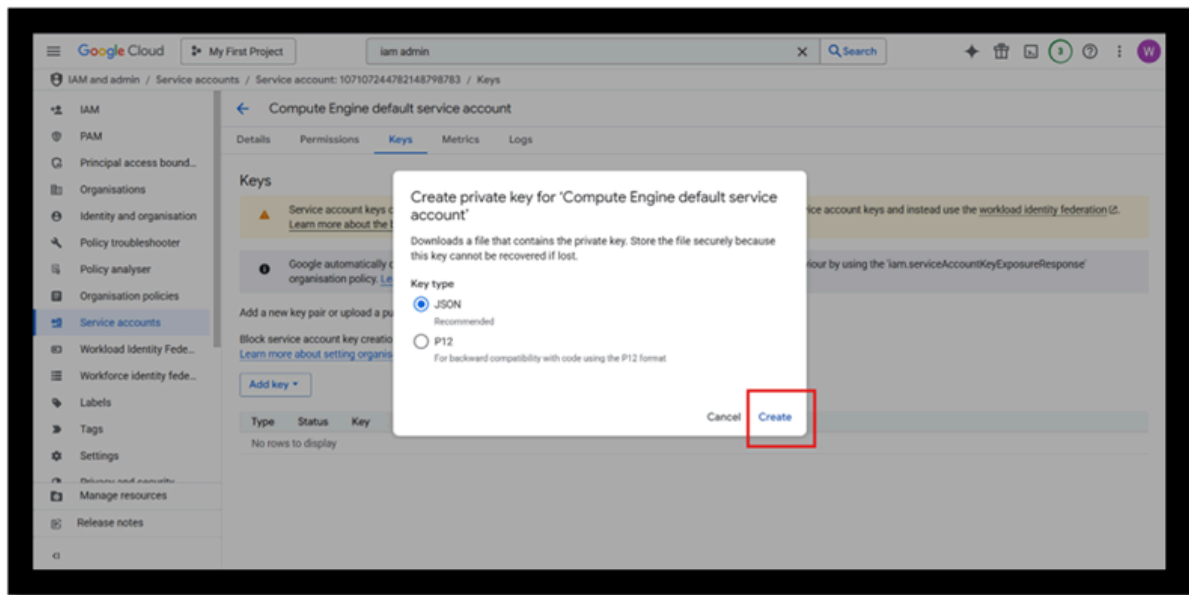


Image 19

Click the create button to download the credential file in JSON format.

2.2 Implement the fine tune model inside the project

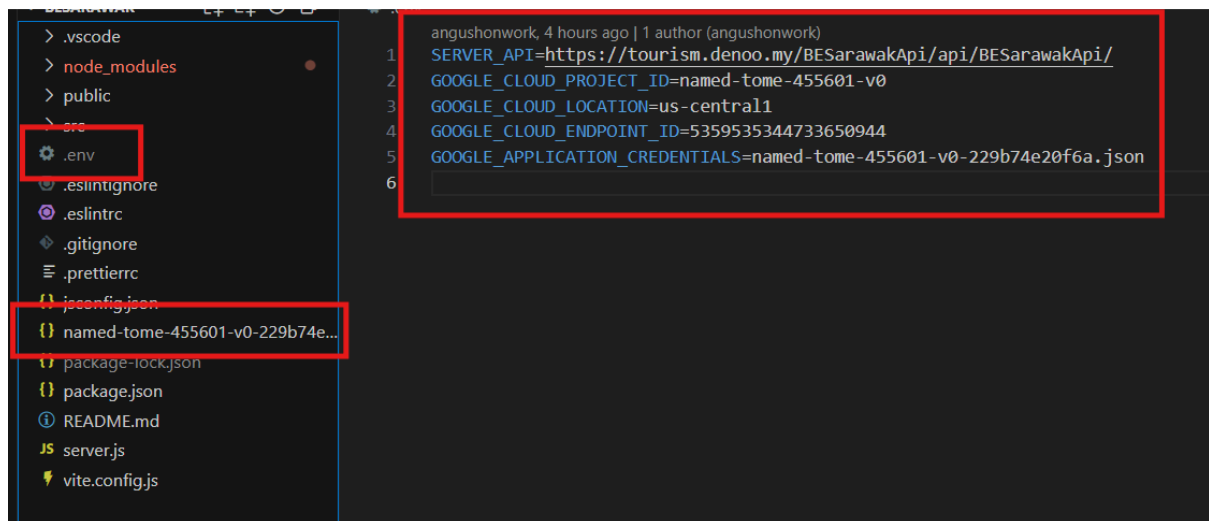


Image 20

Add the .env file and the credential file in the root directory. The environment(endpoint id, project id and location) need to be included in the .env file as a key.

3.0 Implement AI by using API from the Open AI platform

Using the API key to do request

Example code for node.js

```
1  import OpenAI from "openai";
2
3  const openai = new OpenAI();
4
5  async function main() {
6    const completion = await openai.chat.completions.create({
7      messages: [
8        {
9          role: "developer",
10         content: "You are a helpful assistant."
11       }
12     ],
13     model: "gpt-4.1",
14     store: true,
15   });
16
17   console.log(completion.choices[0]);
18 }
19
20 main();
```

Image 21

There are variable models to do requests. Different models come with different pricing and functionality. The request might be different for other AI models such as Gemini, Grok and Claude.

3.1 Optimizing the prompt

Different models come out with different token counters. The input prompt for context will affect the context of output. Different models have their different input and output contexts. Use [tokenizer](#) to do calculation of prompt length.

3.2 Initialization needed before use API key

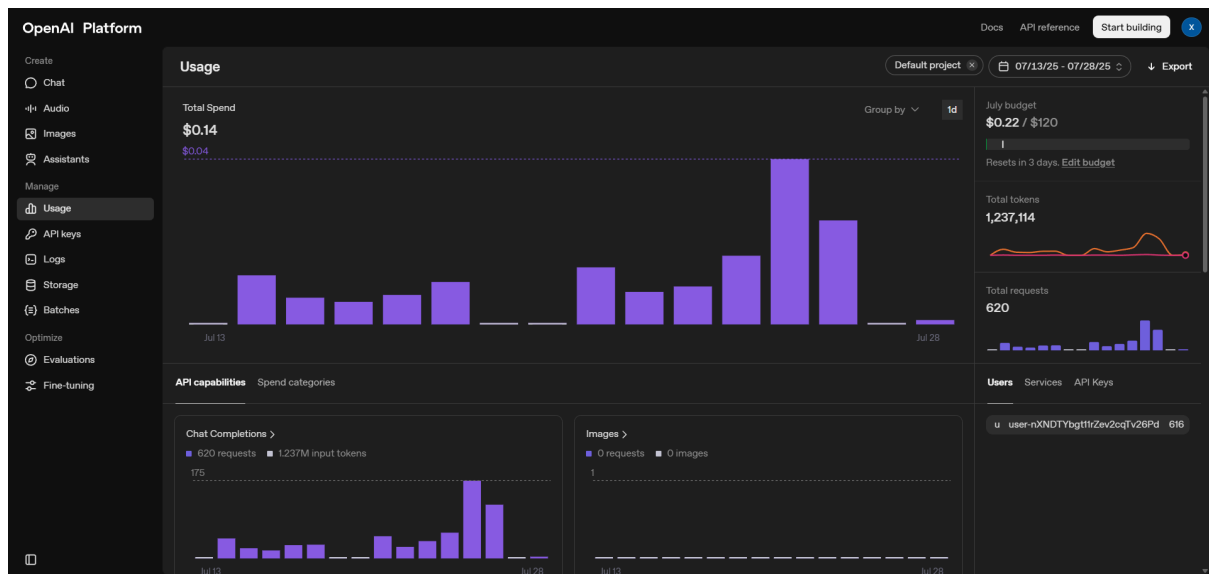


Image 22

Check usage: <https://platform.openai.com/usage>

Based on the usage, you may easily access the request done and total credit used up. It comes with a few graph for easier monitoring on request done by AI.

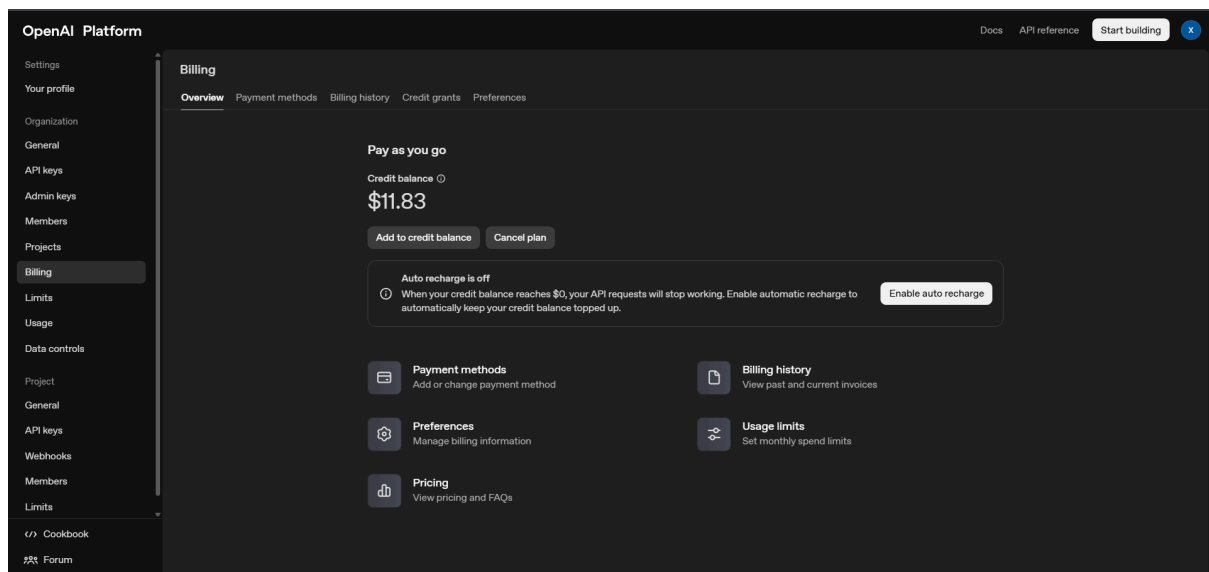
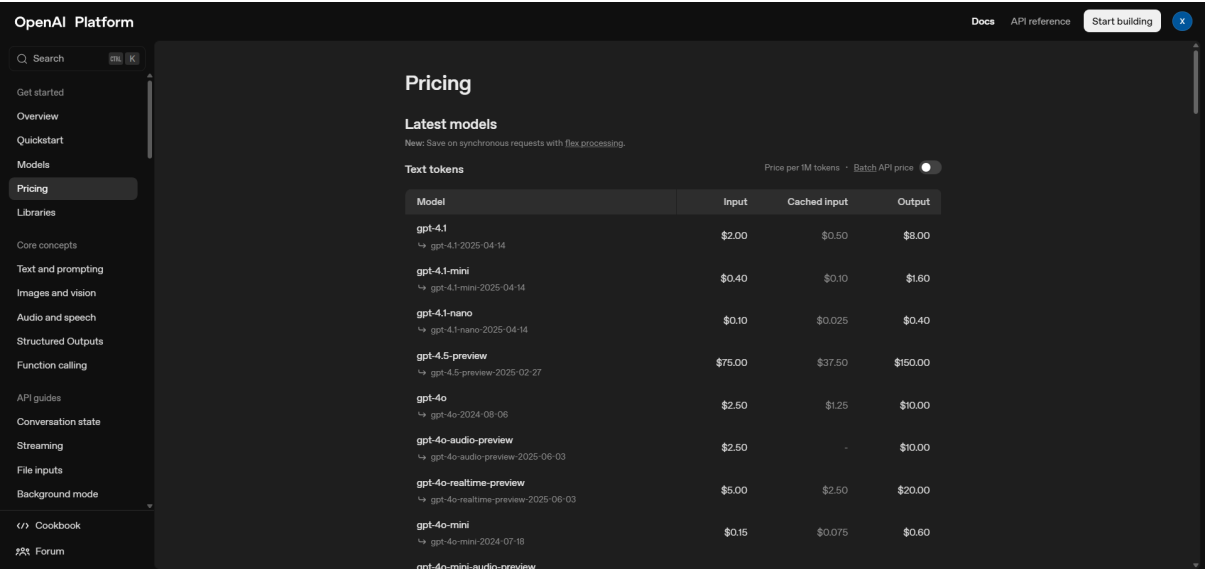


Image 23

Check remaining credit for api key: [Billing Overview](#)

Minimum credit to top up is 5 USD dollars. Can set up auto billing for more convenience use.



The screenshot shows the OpenAI Platform Pricing page. On the left is a sidebar with navigation links: Search, Get started, Overview, Quickstart, Models, Pricing (selected), and Libraries. The main content area is titled 'Pricing' and 'Latest models'. It includes a note: 'New: Save on synchronous requests with flex processing.' Below this is a table for 'Text tokens' with columns for Model, Input, Cached input, and Output. The table lists several models with their respective prices per 1M tokens. A toggle switch at the top right of the table allows switching between 'Price per 1M tokens' and 'Batch API price'.

Model	Input	Cached input	Output
gpt-4.1 <small>↳ gpt-4.1-2025-04-14</small>	\$2.00	\$0.50	\$8.00
gpt-4.1-mini <small>↳ gpt-4.1-mini-2025-04-14</small>	\$0.40	\$0.10	\$1.60
gpt-4.1-nano <small>↳ gpt-4.1-nano-2025-04-14</small>	\$0.10	\$0.025	\$0.40
gpt-4.5-preview <small>↳ gpt-4.5-preview-2025-02-27</small>	\$75.00	\$37.50	\$150.00
gpt-4o <small>↳ gpt-4o-2024-08-06</small>	\$2.50	\$1.25	\$10.00
gpt-4o-audio-preview <small>↳ gpt-4o-audio-preview-2025-06-03</small>	\$2.50	-	\$10.00
gpt-4o-realtime-preview <small>↳ gpt-4o-realtime-preview-2025-06-03</small>	\$5.00	\$2.50	\$20.00
gpt-4o-mini <small>↳ gpt-4o-mini-2024-07-18</small>	\$0.15	\$0.075	\$0.60
gpt-4o-mini-audio-preview			

Image 24

Different model requests come with different pricing. The pricing is more expensive based on the latest model and high end performance models. Choose the suitable model based on the scenario. For my project, the model I used is gpt-4o-mini which comes with lower price for context and it has a more powerful performance.[Pricing for each Open AI models.](#)

3.3 Web Search Model

In the Open AI platform, it provides some AI models that come with web search functionality. The model with web search functionality can help users to do real time searching based on the user questions.



The screenshot shows a table titled 'Web Search' with two rows. The first row lists 'gpt-4o and gpt-4.1 models (including mini models)' with a price of '\$25.00' and '1k calls'. The second row lists 'o3, o4-mini, o3-pro, and deep research models' with a price of '\$10.00' and '1k calls'. Both rows include the note 'Search content tokens free' or 'Search content tokens billed at model rate'.

Web Search	
gpt-4o and gpt-4.1 models (including mini models)	\$25.00 1k calls
Search content tokens free	
Web Search	
o3, o4-mini, o3-pro, and deep research models	\$10.00 1k calls
Search content tokens billed at model rate	

Image 25

3.4 Example to do the AI API request

Use this [endpoint](#) with the post method. In the header section, input the key and value.

Key	Value
Content-Type	application/json
Authorization	Bearer sk-svcacct-i7dr6mw310fh008Xdf_X5wa435nL5QHapTKYraowF6Tvd9sWwyjwXpySNmF...
Key	Value

Image 26

Choose a suitable model and adjust the temperature based on the scenario. The **temperature is the parameter that controls the randomness and creativity of the model's output**. For the lower temperature(0.2), it produces more deterministic and focused outputs which is suitable for tasks where accuracy and consistency are important, such as technical documentation or factual question answering. For the medium temperature(0.5-0.7), the output provides a balance between determinism and randomness, often a good starting point for general-purpose applications. For the high temperature(0.8), the large language model will generate more random and creative outputs which is useful for tasks like brainstorming, creative writing, or generating diverse ideas

```
1 {
2   "model": "gpt-4o-mini",
3   "temperature": 0.6,
4   "messages": [
5     {
6       "role": "system",
7       "content": "You are a professional customer service agent for an e-commerce platform. Always provide concise, polite, and informative answers, and try to help users with their orders, returns, or general inquiries."
8     },
9     {
10      "role": "user",
11      "content": "I ordered a phone case last week but it hasn't arrived yet. Can you check the status?"
12    }
13  ]
14 }
15
```

Image 27

This is the format used to make requests from AI by API. A few different roles can be used for a large language model. For the Open AI model, it has three roles that can be set up in the prompt. Each role provides a different priority for the instruction.

Example output:

```
{
  "id": "chatcmpl-Bx8ms9woImWS081kESBbWKi7QzgTH",
  "object": "chat.completion",
  "created": 1714567890,
  "model": "gpt-4o-mini",
  "usage": {
    "prompt_tokens": 15,
    "completion_tokens": 10,
    "total_tokens": 25
  },
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "I can help you check the status of your phone case. Please provide me with the order number or the email address associated with the purchase, and I'll look it up for you."
      },
      "finish_reason": "stop"
    }
  ]
}
```

```
"created": 1753434830,
"model": "gpt-4o-mini-2024-07-18",
"choices": [
  {
    "index": 0,
    "message": {
      "role": "assistant",
      "content": "I'd be happy to help you with that! Please provide me with your order number, and I
can check the status of your phone case for you.",
      "refusal": null,
      "annotations": []
    },
    "logprobs": null,
    "finish_reason": "stop"
  }
],
"usage": {
  "prompt_tokens": 69,
  "completion_tokens": 31,
  "total_tokens": 100,
  "prompt_tokens_details": {
    "cached_tokens": 0,
    "audio_tokens": 0
  },
  "completion_tokens_details": {
    "reasoning_tokens": 0,
    "audio_tokens": 0,
    "accepted_prediction_tokens": 0,
    "rejected_prediction_tokens": 0
  }
},
"service_tier": "default",
"system_fingerprint": "fp_34a54ae93c"
}
```

Image 28

From the output we can know how many tokens are used for input and output.

OpenAI Platform

Docs API reference Start building

Create

- Chat
- Audio
- Images
- Assistants

Manage

- Usage
- API keys
- Logs**
- Storage
- Batches
- Optimize
- Evaluations
- Fine-tuning

Logs Completions Responses Traces

Quick eval 15s Enter id to view details

Model Date Metadata Tool call Input Search... Output Search... 889 results

Input	Output	Model	Created
Here is the dataset of 2022 monthly event data including PCDS 2030 focus areas, att...	Hello! How can I assist you today? If you have any questions or need assistance wi...	gpt-4o-mini-2024-07-18	Jul 28, 11:16 AM
Here is the dataset of 2022 monthly event data including PCDS 2030 focus areas, att...	Hello! How can I assist you today? If you have any questions or need help with a p...	gpt-4o-mini-2024-07-18	Jul 28, 11:16 AM
Here is the dataset of 2022 monthly event data including PCDS 2030 focus areas, att...	Hello! How can I assist you today? If you have any questions or need help with a sp...	gpt-4o-mini-2024-07-18	Jul 28, 11:15 AM
Here is the dataset of 2022 monthly event data including PCDS 2030 focus areas, att...	Hello! How can I assist you today?	gpt-4o-mini-2024-07-18	Jul 28, 11:14 AM
Here is the dataset of 2022 monthly event data including PCDS 2030 focus areas, att...	Good morning! How can I assist you today?	gpt-4o-mini-2024-07-18	Jul 26, 4:06 PM
Here is the dataset of 2022 monthly event data including PCDS 2030 focus areas, att...	Good morning! How can I assist you today?	gpt-4o-mini-2024-07-18	Jul 26, 4:06 PM
Here is the dataset of 2022 monthly event data including PCDS 2030 focus areas, att...	Hello! How can I assist you today? If you have any questions or need help with a sp...	gpt-4o-mini-2024-07-18	Jul 26, 4:06 PM
Here is the dataset of 2022 monthly event data including PCDS 2030 focus areas, att...	Hello! How can I assist you today? If you have any specific requests or questions, f...	gpt-4o-mini-2024-07-18	Jul 26, 4:06 PM
Here is the dataset of 2022 monthly event data including PCDS 2030 focus areas, att...	Hello! How can I assist you today? If you have any questions or need help with so...	gpt-4o-mini-2024-07-18	Jul 26, 4:06 PM
Here is the dataset of 2022 monthly event data including PCDS 2030 focus areas, att...	Hello! How can I assist you today? If you have any questions or need support with ...	gpt-4o-mini-2024-07-18	Jul 26, 3:54 PM
Here is the dataset of 2022 monthly event data including PCDS 2030 focus areas, att...	Hello! How can I assist you today?	gpt-4o-mini-2024-07-18	Jul 26, 3:54 PM
Here is the dataset of 2022 monthly event data including PCDS 2030 focus areas, att...	Hello! How can I assist you today? If you have any questions or need help with a sp...	gpt-4o-mini-2024-07-18	Jul 26, 3:54 PM
Here is the dataset of 2022 monthly event data including PCDS 2030 focus areas, att...	Hello! How can I assist you today? If you have any questions or need help with a sp...	gpt-4o-mini-2024-07-18	Jul 26, 3:53 PM
Here is the dataset of 2022 monthly event data including PCDS 2030 focus areas, att...	Good morning! How can I assist you today?	gpt-4o-mini-2024-07-18	Jul 26, 3:53 PM

Image 29

In the open AI API platform, we can open logs to do reviews on the conversation history. All the chat completion will be stored inside here.

4.0 Prompt engineering

Prompt engineering is the process of writing effective instructions for a model, such that it consistently generates content that meets your requirements.

The reason to use prompt engineers is because it is low cost and fast change if compared with fine tuning a model. Fine tune a model usually needs high quality data, a more powerful machine and time consumed as it needs a few hours to do fine tuning. The result of fine tuning depends on the quality of data, the data set size and the chosen model. At the same time, prompt engineers to a model can affect the result or the output of the model instantly which is the change that can directly apply to the model. Also, prompt engineers are free and do not take a long time to change. Therefore, prompt engineers are chosen instead of fine tuning a model.

4.1 Role for AI

There are several roles for AI which are user, assistance, system and developer. Different roles might have different priorities for the given prompt. The highest priority is system and developer, then is assistance and least priority is user. AI will first follow the instruction given in the system when processing an output.

For the system role, it provides high-level instructions or a persona for the model to adhere to throughout the conversation. It sets the overall context and behavior for the AI assistant. System messages are prioritized by the model and are not typically displayed to the end-user.

For the assistance role, it represents messages generated by the OpenAI model in response to user or system prompts. These are the AI's outputs, acting as the assistant's contribution to the conversation.

For the user role, it represents messages or prompts provided by the end-user. It contains the queries, instructions, or conversational turns that the user sends to the model.

Therefore, the usage of roles is vital as it can affect the output of the model.

4.2 Pros and cons

Pros	Cons
Low pricing as no hardware and training costs	Cannot change the weight of model
Easy to implement without professional knowledge	Not suitable for long context as it easily lose track
Flexibility to change other models	The knowledge cutoff
Suitable for multitask	Some cannot do real time web search
Can implement other tools to improve performance	The output sometimes is irrelevant
Can make change of model behaviour by modifying prompt.	

Table 1: pros and cons of prompt engineering

4.3 Advantage

There are some advantages to using prompt engineering on LLM. The first advantage is it provides lower costs as it does not charge any fee to do prompt engineering. Also, it provides a fast and more reliable way to make changes on the LLM on the spot. It provides a way that with different prompts, you can complete multiple tasks with only one LLM. This means that it can multitask at the same time just by changing the prompt with a more suitable prompt for the task. It is good for general purpose use and cost effective. At the same time, it provides a more flexible way to change the model once finding a better model, which is better than locking in with only one model. Prompt engineers can work well with other techniques such as retrieval-augmented generation (RAG) and tools used.

4.4 Limitation

However, there are some limitations for the prompt engineer. Prompt engineers cannot change or enhance the AI output deeply such as enhancing the natural language programming (NLP) of the AI and the performance of the AI. Also, when the context is too long, it will lose track of the earlier content, especially when the context hits the input token limit. The model has their knowledge cutoff and outdated understanding. Prompts cannot let the model know what it doesn't know which means that if the model was never trained on recent events or specific info, prompting won't magically provide it the knowledge. Some of the models cannot do web search which means that it cannot do research online of the real time data. This will cause the model's knowledge to be limited, and need to change model in the future to ensure the model stays powerful.

4.5 Example prompt

We can design the prompt based on our use case. There is no better prompt or the best prompt to generate a better output. We need to do trial and error for the prompt by comparing the output of the same model with a different prompt. Refining the prompt is crucial to ensure the output of the LLM can achieve better performance in completing the task. With this comparison of prompts, it can lead us to have a better design for the prompt. Here is the example of a prompt can write as:

Customer service prompt:

```
{
  role: 'system',
  content: `You are a highly professional, empathetic, and knowledgeable customer service assistant.

  Your responsibilities include:
  – Providing clear, helpful, and concise answers to customer inquiries.
  – Maintaining a calm, polite, and respectful tone at all times.
  – Offering solutions or next steps in a friendly and proactive manner.
  – Acknowledging customer emotions and showing genuine willingness to help.
  – Clarifying any confusion without making the customer feel at fault.
  – If you don't have an answer, say so clearly and direct the user to a more appropriate source.

  Always prioritize customer satisfaction while balancing accuracy, transparency, and brand professionalism. Never guess or fabricate information. If an issue requires escalation, recommend contacting support or provide relevant links or contact details.

  Keep your responses structured, solution-oriented, and reassuring. Use plain language unless the customer is clearly using technical terms.`
}
```

4.6 Practice needed for prompt

There are some practices needed to achieve before writing a prompt.

1. Be Clear and Explicit

- Define the role: e.g., "You are a customer service agent..."
- Set goals: "Your job is to help the user resolve any issues..."
- Remove ambiguity—LLMs do not guess well.

2. Give context by add relevant background

- Product info, tone, target audience.
- Include example dialogue or data if needed.

3. Set Constraints and Format

- Specify tone: “Be professional, empathetic.”
- Specify output style: “Reply in bullet points,” or “Use markdown.”

4. Use Structured Instructions

- Use enumerated or paragraph instructions for easier parsing:

5. Encourage Step-by-Step Reasoning

- Within this, the LLMs may do reasoning by Chain of Thought (CoT) before generating reply messages.

6. Avoid Open-Ended Vagueness

- Bad: Summarize this article.
- Better: Summarize the following article in 3 bullet points using plain language for a high-school reader.

Prompt template:

You are a [role] helping with [task].

Your goals:

1. [Goal 1]

2. [Goal 2]

Use a [tone/style], and reply in [format, e.g., markdown/bullets].

Constraints: [word count, language, avoid X, include Y]

5.0 Reference

Harth, M. (2023, November 7). *Understanding role management in OpenAI's API: Two methods compared*. OpenAI Community Forum.
<https://community.openai.com/t/understanding-role-management-in-openais-api-two-methods-compared/253289>

OpenAI. (n.d.). Streaming responses. OpenAI.
<https://platform.openai.com/docs/guides/streaming-responses?api-mode=responses>

OpenAI. (n.d.). *Text generation guide*. OpenAI.
<https://platform.openai.com/docs/guides/text?api-mode=responses&prompt-example=prompt>