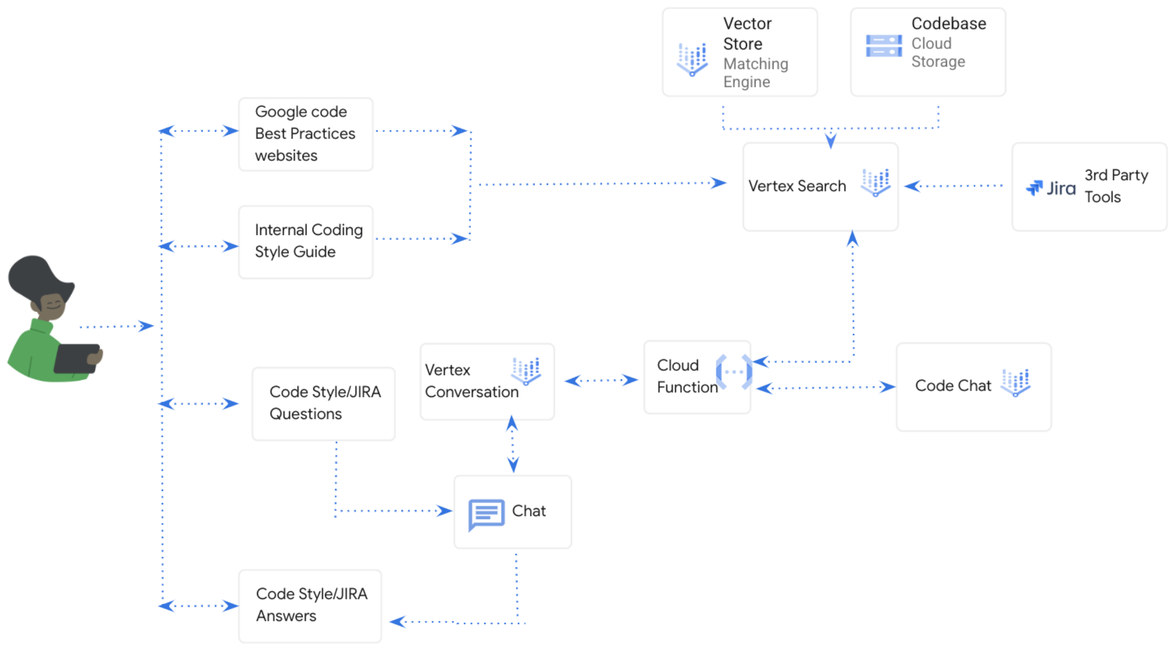
Demo Guide

Lei Pan 11/20

Example: Coding Style Review Doc Search, JIRA Issues Search, and Codebase repo chat

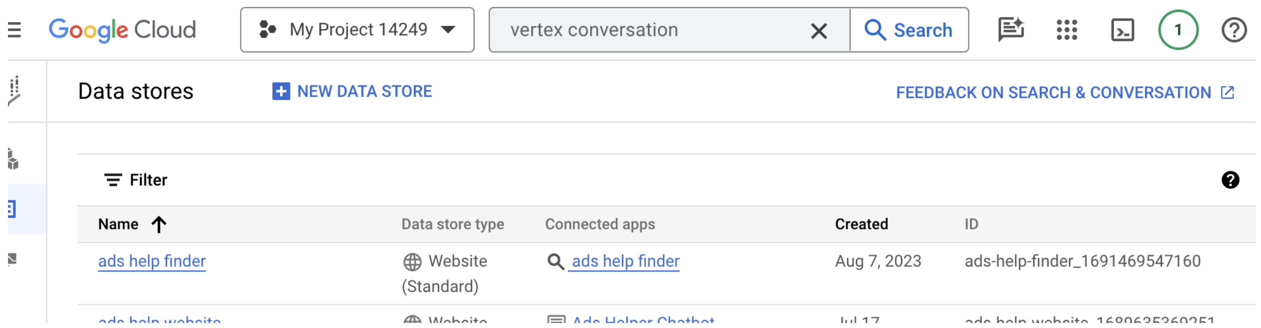


Step 1: Enable Vertex AI Search API and Vertex AI Conversation API

1. Go to the Google Cloud Platform (GCP) Console.
2. In the navigation pane, select APIs & Services > Enable APIs and Services.
3. In the search bar, type Vertex AI Search API and click on the result.
4. Click Enable.
5. Repeat steps 3 and 4 for Vertex AI Conversation API.

Step 2: Build Vertex AI Search Engine with PDFs (Coding Style PDFs)

- Public doc: How to set up unstructured data store in vertex ai search
- For PDFs: use your own pdfs including your coding best practices or use those examples and save them as PDFs and dump them to vertex AI search engine datastore.
- Once you set the datastore up, you will find the search engine id in the UI. Use that in the code.



Name	Data store type	Connected apps	Created	ID
ads help finder	Website (Standard)	ads help finder	Aug 7, 2023	ads-help-finder_1691469547160

Step 3: Build Another Vertex AI Search Engine with JIRA Issue Websites

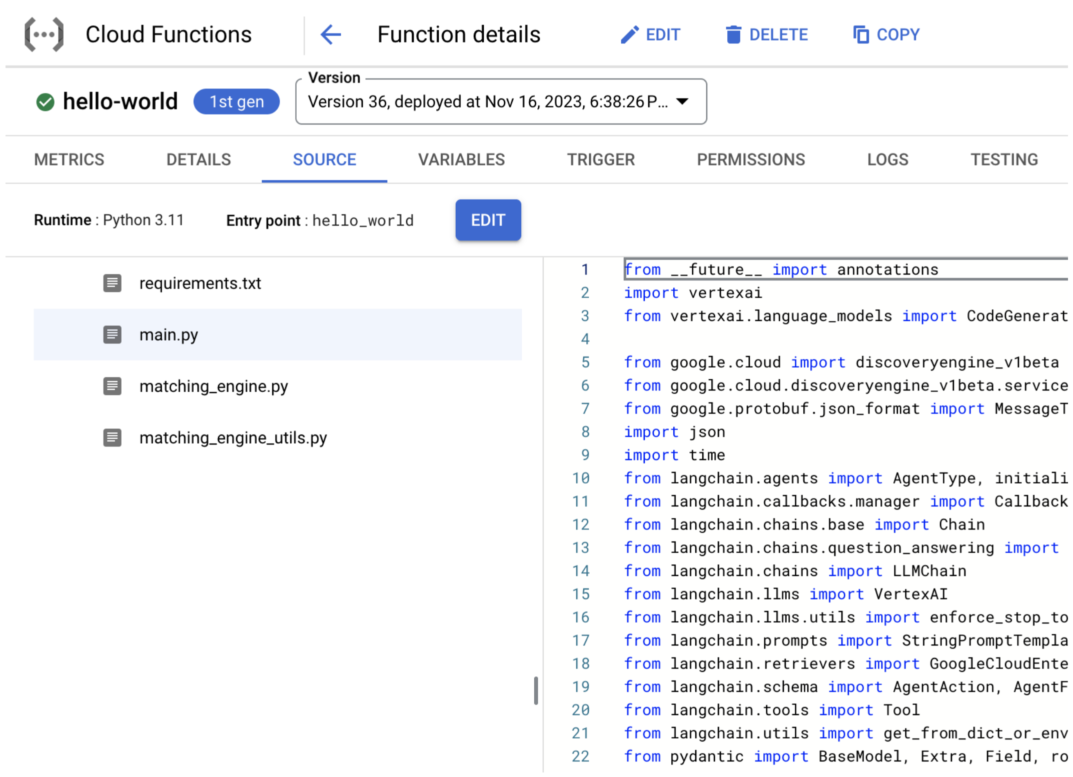
- Public doc: How to set up JIRA in Vertex AI search
- For JIRA links: you can use your own JIRA links or use this public JIRA links to do the demo
- Once you set the datastore up, you will find the search engine id in the UI. Use that in the code. Refer to the screenshot in the step above.

Step 4: Break Down Code Repository (Bank of Anthos) to Chunks and Store Indexes in the Vector Store (Matching Engine)

- Ingest the codebase to GCS bucket. Replace the `GCS_BUCKET_DOCS`with the bucket you use.
- Code reference to do RAG on codebase: github link

Step 5: (Optional - only if you want to demo Google Chat): Deploy Cloud Functions Code which Uses MultiRetrievalQAChain to Retrieve Information (Embedding Spaces + RAG + Codey) from 3 Different Retriever Embedding Spaces

- Public doc: How to deploy cloud function
- Deploy below 3 sources to cloud functions as shown in the screenshot below
    - Webhook cloud function code. Copy the code here to cloud functions as main.py
    - Requirements.text: below code is all you need.
        - Flask==2.2.2
        - Werkzeug==2.3.7
        - google-cloud-aiplatform
        - google-cloud-discoveryengine
        - langchain==0.0.236
    - You can download matching_engine.py and mathcing_engine_utils.py here.

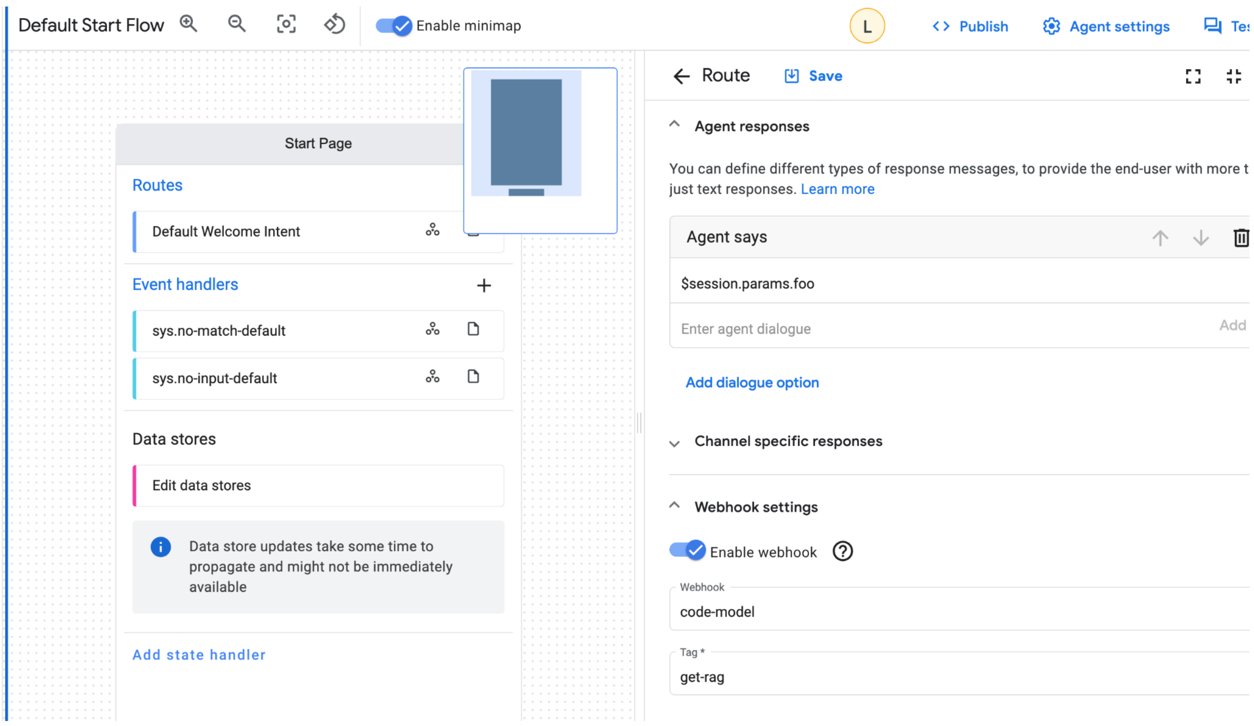


```
from __future__ import annotations
import vertexai
from vertexai.language_models import CodeGenerat

from google.cloud import discoveryengine_v1beta
from google.cloud.discoveryengine_v1beta.service
from google.protobuf.json_format import MessageT
import json
import time
from langchain.agents import AgentType, initiali
from langchain.callbacks.manager import Callback
from langchain.chains.base import Chain
from langchain.chains.question_answering import
from langchain.chains import LLMChain
from langchain.llms import VertexAI
from langchain.llms.utils import enforce_stop_to
from langchain.prompts import StringPromptTempla
from langchain.retrievers import GoogleCloudEnte
from langchain.schema import AgentAction, AgentF
from langchain.tools import Tool
from langchain.utils import get_from_dict_or_env
from pydantic import BaseModel, Extra, Field, ro
```

Step 6 (Optional - only if you want to demo Google Chat): Call Cloud Function in Webhook in a Dialogflow Project

- Public doc: How to set up a dialogflow project
- Public doc: How to set up webhook with cloud function
- Once you set it up, go to default welcome intent, sys.no-match-default, sys.no-input-default and set the agent response to the response from webhook (cloud function that you deployed), please refer to the screenshot below.



Data store updates take some time to propagate and might not be immediately available

You can define different types of response messages, to provide the end-user with more t just text responses. Learn more

Agent says: $session.params.foo

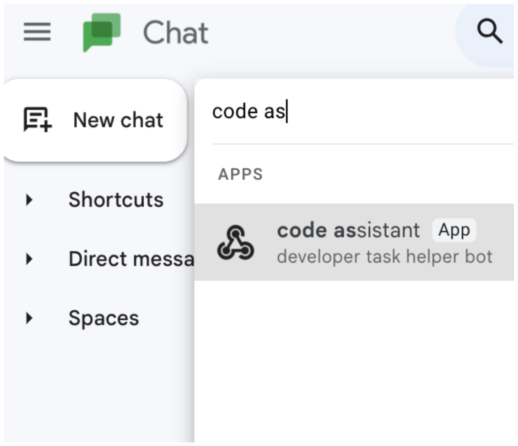
Enter agent dialogue

Enable webhook: on

Webhook: code-model

Tag *: get-rag

-
- Public doc: how to deploy dialogflow project to Google Chat
- After that, you should be able to search the chatbot in your google chat



Step 7: Prompts to Test Different Retrievers

1. To test the PDF search engine, ask questions about specific coding styles or patterns found in the PDFs.

    a. *Prompt: Tell me more about the best java style according to the java coding style guide.*
2. To test the JIRA issue search engine, ask questions about specific JIRA issues or topics related to JIRA issues.
    a. *Prompt: Show me the top 2 Flink issues in JIRA*
3. To test the code search engine, ask questions about specific code snippets or functionalities found in the Bank of Anthos code repository.
    a. *Prompt: how does CI/CD pipeline that powers Bank of Anthos work?*