Starting April 29, 2025, Gemini 1.5 Pro and Gemini 1.5 Flash models are not available in projects that have no prior usage of these models, including new projects. For details, see Model versions and lifecycle (/vertex-ai/generative-ai/docs/learn/model-versions#legacy-stable).

Use Vertex AI Search as a retrieval backend using Vertex AI RAG Engine

The VPC-SC security control (/vertex-ai/generative-ai/docs/security-controls) is supported by RAG Engine. Data residency, CMEK, and AXT security controls aren't supported.

To see an example of using RAG Engine with Vertex AI Search, run the "RAG Engine with Vertex AI Search" Jupyter notebook in one of the following environments:

Open in Colab (https://colab.research.google.com/github/GoogleCloudPlatform/generative-ai/blob/main/gemini/rag-engine/rag_engine_vertex_ai_search.ipynb) | Open in Colab Enterprise (https://console.cloud.google.com/vertex-ai/colab/import/https%3A%2F%2Fraw.githubusercontent.com%2FGoogleCloudPlatform%2Fgenerative-ai%2Fmain%2Fgemini%2Frag-engine%2Frag_engine_vertex_ai_search.ipynb) | Open in Vertex AI Workbench user-managed notebooks (https://console.cloud.google.com/vertex-ai/workbench/deploy-notebook?download_url=https%3A%2F%2Fraw.githubusercontent.com%2FGoogleCloudPlatform%2Fgenerative-ai%2Fmain%2Fgemini%2Frag-engine%2Frag_engine_vertex_ai_search.ipynb) | View on GitHub (https://github.com/GoogleCloudPlatform/generative-ai/blob/main/gemini/rag-engine/rag_engine_vertex_ai_search.ipynb)

This page introduces Vertex AI Search integration with the Vertex AI RAG Engine.

Vertex AI Search provides a solution for retrieving and managing data within your Vertex AI RAG applications. By using Vertex AI Search as your retrieval backend, you can improve performance, scalability, and ease of integration.

- **Enhanced performance and scalability**: Vertex AI Search is designed to handle large volumes of data with exceptionally low latency. This translates to faster response times and improved

performance for your RAG applications, especially when dealing with complex or extensive knowledge bases.

- **Simplified data management**: Import your data from various sources, such as websites, BigQuery datasets, and Cloud Storage buckets, that can streamline your data ingestion process (/vertex-ai/generative-ai/docs/rag-overview).
- **Seamless integration**: Vertex AI provides built-in integration with Vertex AI Search, which lets you select Vertex AI Search as the corpus backend for your RAG application. This simplifies the integration process and helps to ensure optimal compatibility between components.
- **Improved LLM output quality**: By using the retrieval capabilities of Vertex AI Search, you can help to ensure that your RAG application retrieves the most relevant information from your corpus, which leads to more accurate and informative LLM-generated outputs.

Vertex AI Search

Vertex AI Search (/generative-ai-app-builder/docs/enterprise-search-introduction) brings together deep information retrieval, natural-language processing, and the latest features in large language model (LLM) processing, which helps to understand user intent and to return the most relevant results for the user.

With Vertex AI Search, you can build a Google-quality search application using data that you control.

Configure Vertex AI Search

To set up a Vertex AI Search, do the following:

1. Create a search data store (/generative-ai-app-builder/docs/create-data-store-es).
2. Create a search application (/generative-ai-app-builder/docs/create-engine-es).

Use the Vertex AI Search as a retrieval backend for Vertex AI RAG Engine

Once the Vertex AI Search is set up, follow these steps to set it as the retrieval backend for the RAG application.

Set the Vertex AI Search as the retrieval backend to create a RAG corpus

These code samples show you how to configure Vertex AI Search as the retrieval backend for a RAG corpus.

RESTPython (#python)
(#rest)

To use the command line to create a RAG corpus, do the following:

1. Create a RAG corpus

   Replace the following variables used in the code sample:

   - ***PROJECT_ID***: The ID of your Google Cloud project.
   - ***LOCATION***: The region to process the request.
   - ***DISPLAY_NAME***: The display name of the RAG corpus that you want to create.
   - ***ENGINE_NAME***: The full resource name of the Vertex AI Search engine or Vertex AI Search Datastore.

```
curl -X POST \
-H "Authorization: Bearer $(gcloud auth print-access-token)" \
-H "Content-Type: application/json" \
"https://LOCATION -aiplatform.googleapis.com/v1/projects/PROJECT_ID /lo
-d '{
  "display_name" : "DISPLAY_NAME ",
  "vertex_ai_search_config" : {
    "serving_config": "ENGINE_NAME /servingConfigs/default_search"
  }
}'
```

2. Monitor progress

   Replace the following variables used in the code sample:

   - ***PROJECT_ID***: The ID of your Google Cloud project.
   - ***LOCATION***: The region to process the request.
   - ***OPERATION_ID***: The ID of the RAG corpus create operation.

```
curl -X GET \
-H "Authorization: Bearer $(gcloud auth print-access-token)" \
-H "Content-Type: application/json" \
"https://LOCATION-aiplatform.googleapis.com/v1/projects/PROJECT_ID/lo
```

Retrieve contexts using the RAG API

After the RAG corpus creation, relevant contexts can be retrieved from Vertex AI Search through the `RetrieveContexts` API.

RESTVertex AI SDK for Python...
(#rest)

This code sample demonstrates how to retrieve contexts using REST.

Replace the following variables used in the code sample:

- ***PROJECT_ID***: The ID of your Google Cloud project.
- ***LOCATION***: The region to process the request.
- ***RAG_CORPUS_RESOURCE***: The name of the RAG corpus resource.

  Format: `projects/{project}/locations/{location}/ragCorpora/{rag_corpus}.`
- ***TEXT***: The query text to get relevant contexts.

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $(gcloud auth print-access-token)" \
"https://LOCATION-aiplatform.googleapis.com/v1/projects/PROJECT_ID/locatic
  -d '{
    "vertex_rag_store": {
      "rag_resources": {
        "rag_corpus": "RAG_CORPUS_RESOURCE"
      }
    },
    "query": {
      "text": "TEXT"
    }
```

```
}'
```

Generate content using Vertex AI Gemini API

REST Vertex AI SDK for Python... (#rest)

To generate content using Gemini models, make a call to the Vertex AI `GenerateContent` API. By specifying the `RAG_CORPUS_RESOURCE` in the request, it automatically retrieves data from Vertex AI Search.

Replace the following variables used in the sample code:

- ***PROJECT_ID***: The ID of your Google Cloud project.
- ***LOCATION***: The region to process the request.
- ***MODEL_ID***: LLM model for content generation. For example, `gemini-2.0-flash`.
- ***GENERATION_METHOD***: LLM method for content generation. For example, `generateContent`, `streamGenerateContent`.
- ***INPUT_PROMPT***: The text that is sent to the LLM for content generation. Try to use a prompt relevant to the documents in Vertex AI Search.
- ***RAG_CORPUS_RESOURCE***: The name of the RAG corpus resource. Format: `projects/{project}/locations/{location}/ragCorpora/{rag_corpus}`.
- ***SIMILARITY_TOP_K***: Optional: The number of top contexts to retrieve.

```
curl -X POST \
-H "Authorization: Bearer $(gcloud auth print-access-token)" \
-H "Content-Type: application/json" \
"https://LOCATION-aiplatform.googleapis.com/v1/projects/PROJECT_ID/lo
-d '{
  "contents": {
    "role": "user",
    "parts": {
      "text": "INPUT_PROMPT"
    }
  },
  "tools": {
    "retrieval": {
```

```
        "disable_attribution": false,
        "vertex_rag_store": {
          "rag_resources": {
              "rag_corpus": "RAG_CORPUS_RESOURCE"
            },
          "similarity_top_k": SIMILARITY_TOP_K
        }
      }
    }
  }'
```

What's next

- Retrieval and ranking (/vertex-ai/generative-ai/docs/retrieval-and-ranking)

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the Apache 2.0 License (https://www.apache.org/licenses/LICENSE-2.0). For details, see the Google Developers Site Policies (https://developers.google.com/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2025-06-06 UTC.