Connected System Plugin for Appian
# Table of Contents

- **Overview**  
- **Features**  
- **Connected System Configuration**  
- **Create Index**  
- **Delete Index**  
- **Get Index Details**  
- **Get Vector Details**  
- **List Indices**  
- **Upsert Document**  
- **Upsert Text**  
- **Delete Documents**  
- **Query Document**
Overview

Pinecone makes it easy to build high-performance vector search applications. It’s a managed, cloud-native vector database with a simple API and no infrastructure hassles. Unlike traditional search methods that revolve around keywords, vector databases index and search through ML-generated representations of data, called vector embeddings, to find items most similar to the query.

Pinecone indexes and stores vector embeddings for efficient management and fast retrieval. Unlike a standalone vector index, a vector database like Pinecone provides additional capabilities such as index management, data management, metadata storage and filtering, and horizontal scaling.

Features

The Pinecone Connected System includes the following features:

- Create, Get, List and Delete Index in Pinecone
- Upsert, Delete and Query Document/Text Vectors and Get Vector Details.
The Pinecone Connected system requires the following credentials: Environment, PineconeAPI Key and OpenAI API Key.

1. Go to the [Pinecone console](#) and create a new account if you don’t have one already. Go to [API Keys](#).
2. Click on **Create API Key** to generate a new API key and copy both the environment and API key. Paste the environment and API key values in the connected system dialog box.

3. Now, go to the [OpenAI console](#). Make sure that the **API keys** menu is selected.
Click on **Create new secret key** to generate a new API key.

4. Copy the value and save it separately as we won’t be able to access it again. Paste the API key in the connected system dialog box.

5. Once you have entered all the credentials, click on **Test Connection**. Success message will be shown if the provided credentials are correct.
Create Index

Creates a Pinecone Index. A Pinecone index accepts and stores vectors, serves queries over the vectors it contains.

Inputs:

**Index Name** (Text) - Required

**Description:** Provide the name of the index to be created.

**Metric** (Text) - Optional

**Description:** Provide the distance metric to be used for similarity search. Valid values are: 'cosine', 'dotproduct', 'euclidean'. Default: 'cosine'. Refer [https://docs.pinecone.io/docs/indexes#distance-metrics](https://docs.pinecone.io/docs/indexes#distance-metrics) for more details.

**Pods** (Number(Integer)) - Optional

**Description:** Provide the number of pods for the index to use, including replicas. Default: 1

**Replicas** (Number(Integer)) - Optional

**Description:** Provide the number of replicas. Replicas duplicate your index. They provide higher availability and throughput. Default: 1

**Pod Type** (Text) - Optional

**Description:** Provide the type of pod to use. One of s1, p1, or p2 appended with . and one of x1, x2, x4, or x8. Default: p1.x1 Refer [https://docs.pinecone.io/docs/indexes#pods-pod-types-and-pod-sizes](https://docs.pinecone.io/docs/indexes#pods-pod-types-and-pod-sizes) for more details.
Delete Index

Deletes an existing pinecone index.

**Inputs:**

**Index Name** (Text) - Required

**Description:** Provide the name of the index to be deleted.
Get Index Details

This integration returns the pod details and status of the given pinecone index.

**Inputs:**

**Index Name** (Text) - Required

**Description:** Provide the name of the pinecone index.
Get Vector Details

This integration returns the total vectors (chunks) and number of vectors in each namespace.

Inputs:

**Index Name** (Text) - Required

Description: Provide the name of the pinecone index.
List Indices
Lists all the indices present in the environment.
**Upsert Document**

Create Embeddings using OpenAI for the given document and store them as Vectors in the Pinecone database.

**Inputs:**

**Source Document** (Document) - Required

**Description:** Document to be uploaded as vector in pinecone.

**Index name in pinecone** (Text) - Required

**Description:** Name of the pinecone index in which the given document has to be uploaded.

**Chunk Size** (Text) - Optional

**Description:** Provide the size of chunks to be created for the document. Valid values: SMALL, MEDIUM, LARGE. Default: SMALL.

The above chunk sizes represent the number of characters in each chunk(vector) that is upserted to Pinecone. The number of characters for each chunk size values are as follows:

- SMALL = 600 characters
- MEDIUM = 1000 characters
- LARGE = 2000 characters

**Note:** If you are using the queried results from Pinecone to any Large Language Model (LLM) for further processing, consider if there are any size limits for the LLM while choosing the chunk size when upserting into pinecone.

**Metadata** (Text) - Optional

**Description:** Provide the additional metadata to be upserted into pinecone in JSON format using `addToJson()`. By default, chunkNumber, text, documentId, documentName, pageNo will be added.
**Upsert Text**

Create Embeddings using OpenAI for the given raw text or text document and store them as Vectors in the Pinecone database.

**Inputs:**

**Input Type** (Text) - Required

**Description:** Provide the input type for upserting text into Pinecone. Valid values are: DOCUMENT, RAW_TEXT.

**Document** (Document) - Required (when input type is DOCUMENT)

**Description:** Provide the text document to upsert its contents to pinecone.

**Text** (Text) - Required (when input type is RAW_TEXT)

**Description:** Provide the raw text content to upsert.
**Metadata** (Text) - Optional

**Description:** Provide the metadata to upsert into pinecone in JSON format using `a!toJson()`. By default, chunks of the raw/text or text document contents will be added to this metadata with a property name 'text'.

**Index name in pinecone** (Text) - Required

**Description:** Name of the pinecone index in which the given document has to be uploaded.

**Chunk Size** (Text) - Optional

**Description:** Provide the size of chunks to be created for the document. Valid values: SMALL, MEDIUM, LARGE. Default: SMALL.

The above chunk sizes represent the number of characters in each chunk(vector) that is upserted to Pinecone. The number of characters for each chunk size values are as follows:

- SMALL = 600 characters
- MEDIUM = 1000 characters
- LARGE = 2000 characters

**Note:** If you are using the queried results from Pinecone to any Large Language Model (LLM) for further processing, consider if there are any size limits for the LLM while choosing the chunk size when upserting into pinecone.
Delete Documents

Deletes the vectors that correspond to the provided documents. The whole index can be cleared by providing true to the Delete all contents field.

Inputs:

Index name (Text) - Required

Description: Name of the pinecone index in which the given documents have to be deleted.

Delete all contents (Boolean) - Required

Description: Provide true to delete all the vectors(chunks) in the pinecone index.

Documents (List of Documents) - Required when Delete all contents is false

Description: Documents to be deleted from Pinecone
Example: {

todocument(654321),

todocument(765432)

}

Query Document

Get matches and score for the given query using the vector data (upserted as documents) stored in Pinecone.

Inputs:

**Query** (Text) - Required

**Description:** Document to be uploaded as vector in pinecone.

**Index name** (Text) - Required

**Description:** Name of the pinecone index in which the given query has to be executed.

**Namespace** (Text) - Optional

**Description:** Provide the namespace in the index to query.
**topK** (Integer) - Optional

**Description:** Provide the topK value for pinecone. Default: 2.

**Include Vectors** (Boolean) - Optional

**Description:** Provide true to include the vector values in the response. Default: false.

**Exclude Metadata** (Boolean) - Optional

**Description:** Provide true to exclude the metadata in the query results. The metadata will be chunks of queried documents. Default: false.

**Documents** (List of Documents) - Optional

**Description:** Provide the documents for which the query needs to be executed. Provide null to query all the documents in the index.

Example: 

```javascript
{  
  todocument(123456),
  todocument(445678)
}
```