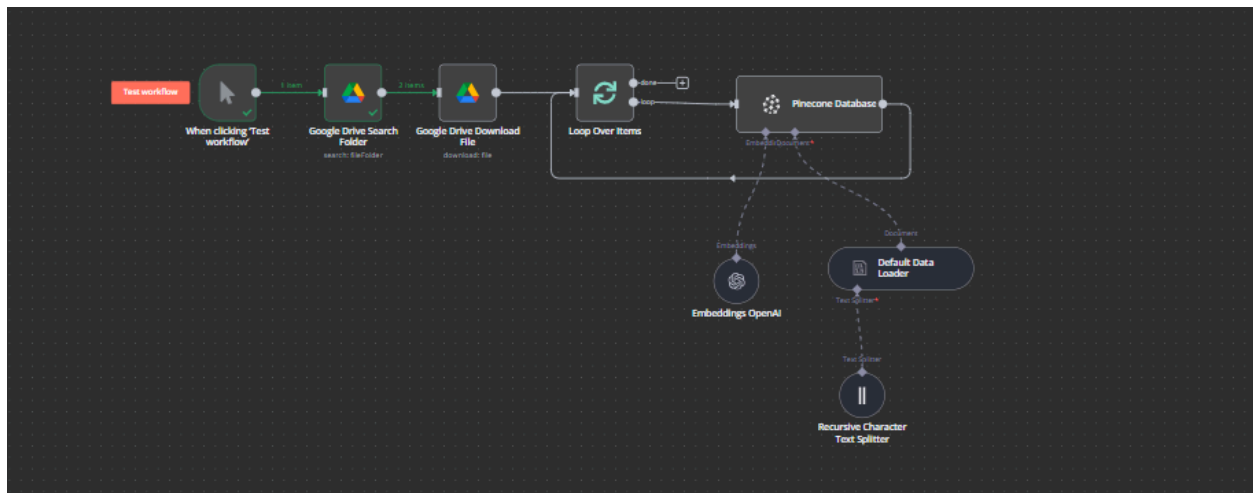


# Voice AI Agent for Support & Low-Ticket Sales on your website

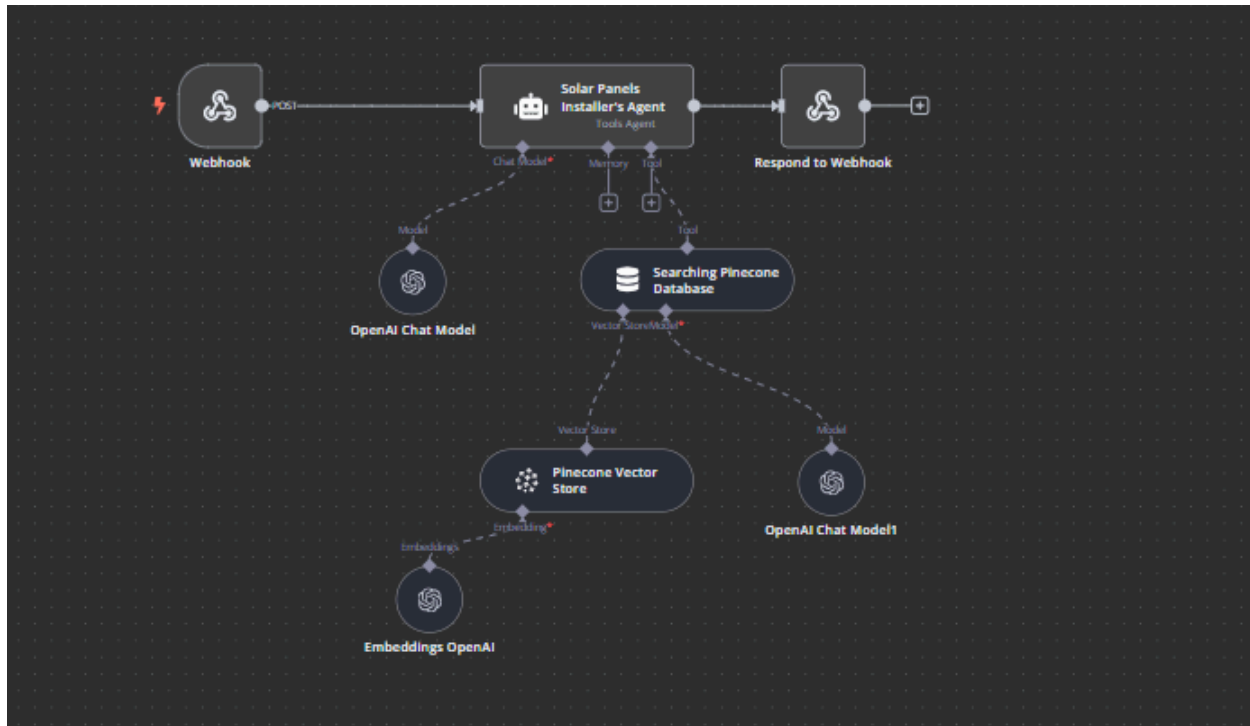
This set up automates customer support and sales inquiries using n8n, Pinecone, and Eleven Labs. In this guide, I'll show you how to build a fully automated AI Voice Agent that can handle customer support and low-ticket sales calls directly from your website – 24/7.

It consists of **two workflows**:

1. **Vectorizing Data for AI Knowledge Base** – Extracts and stores business knowledge into a **Pinecone vector database** for AI retrieval.



1. **AI Voice Agent for Customer Calls** – Uses **Eleven Labs' conversational AI** to handle inbound calls, answer FAQs, and assist with sales.



## Prerequisites

1. **Google Drive Account** – To store and retrieve knowledge base documents.
2. **OpenAI API Key** – Required for embeddings and chat responses.
3. **Pinecone Account** – For storing and searching vectorized knowledge.
4. **Eleven Labs API Key** – To enable AI-generated voice responses.


## How It Works

### Vectorizing Data & Creating a Knowledge Base

This workflow ensures the AI Voice Agent can pull accurate answers from a company's knowledge base.

- **Trigger:** The workflow starts manually or on a schedule to process company documents.
- **Google Drive Search & Retrieval:**
  - The **Google Drive node** searches for relevant documents in a designated folder.

- The **Download Files node** extracts the content from each document.
- **Text Processing & Embedding:**
  - The **Text Splitter** breaks documents into small chunks for efficient AI search.
  - The **OpenAI Embedding node** converts the text into vector format.
- **Storing in Pinecone:**
  - The **Pinecone Vector Store node** saves the processed data, allowing the AI Voice Agent to retrieve relevant information.

 You can scrape or copy and paste information from your or your client's website and add it as a Word document to Google Drive.

## AI Voice Agent for Customer Calls

Once the knowledge base is set, the AI Voice Agent can use it to answer customer queries.

- **Webhook Trigger:**
  - The **Webhook node** listens for incoming call requests from Eleven Labs.

### Retrieving Information from Pinecone:

- The **Pinecone Query node** searches the vector database for relevant answers.

### When the AI Voice Agent receives a question, it follows these steps:

- **Sends a request to n8n Webhook.**
- **n8n searches Pinecone for relevant documents.**
- **Only the relevant text is sent to OpenAI for response generation.**
- **This ensures the AI provides accurate, context-aware responses without unnecessary data.**

### Delivering the Response:

- **The AI Voice Agent speaks the response during the call.**

**Eleven Labs converts text into speech dynamically.**

- Before sending the response to Eleven Labs, the AI ensures that it is structured correctly.
- For example, responses may include a greeting, the requested information, and a closing statement to sound more natural.
- This makes the AI Voice Agent feel more like a real assistant rather than a simple chatbot.

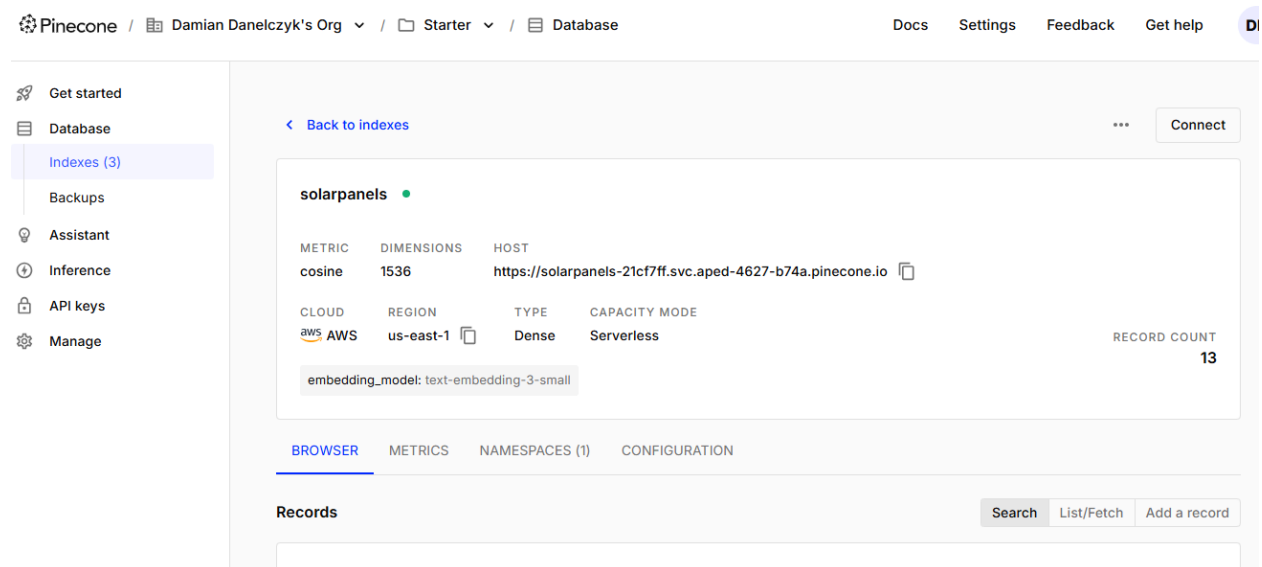
## Step-by-Step Setup

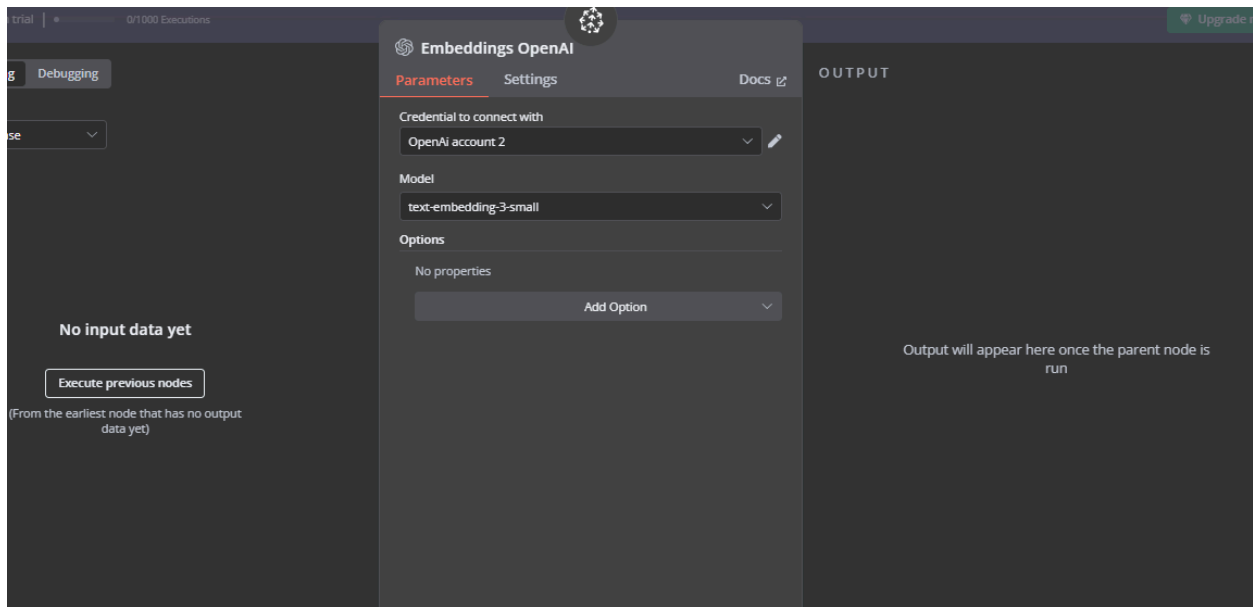
### 1. Configure Google Drive for Document Retrieval

- In **Google Drive**, create a folder with relevant documents.
- In **n8n**, open the **Google Drive Search** node:
- **Authenticate** with Google Drive.
  - **Select the folder** containing the documents you want to vectorized.

### 2. Set Up Pinecone for Vector Storage

- Sign up at [Pinecone.io](https://pinecone.io) and create an index (e.g., "**customer-support**"), ensuring you use **text-embedding-3-small**, as it must match the model in the **OpenAI Embeddings** node.





- In **n8n**, open the **Pinecone Database** node and configure the following:
  - **Credential to connect with:** Select **PineconeApi account**.
  - **Pinecone Index:** Choose your desired index.
  - **Pinecone Namespace:** Assign a name to differentiate this dataset from other documents within the same index.

#### 4. Set Up Eleven Labs for AI Voice Calls

- **Sign up at [Eleven Labs](#) and create an API Key.**
- **Create an AI Agent in Eleven Labs:**
  - Go to **"Agents"** → **"Create New Agent"** → Select **Blank Template**.
  - Choose a **voice model** for the AI agent.
  - Set the **first response message** (e.g., "Hello, how can I assist you today?").
  - In the **system prompt**, instruct the AI to retrieve answers from n8n.  
**(prompt attached below)**

Add additional languages ↕


**First message**  
 The first message the agent will say. If empty, the agent will wait for the user to start the conversation.

Hey, I'm Rob, your solar energy guide! How can I assist you today?

**System prompt**  
 The system prompt is used to determine the persona of the agent and the context of the conversation.

You are a customer support assistant for solar panel and EV charger installations. Provide clear and accurate information about Energy Concerns' services, including options, pricing, and installation.

Use the 'n8n\_solar\_installers' to query the database for details on solar panels, battery storage, EV chargers, and installation procedures. Respond with a concise, friendly answer based on the retrieved data.

Maintain a friendly and approachable tone, using natural fillers like "hmm" to keep the conversation engaging. Avoid repeating the user's question - provide a clear and direct response with the necessary information. 

**Dynamic Variables**  
 Variables like {{user\_name}} in your prompts and first message and dynamic variables in tool parameters will be replaced with actual values when the conversation starts. [Learn more](#)

- **Connect the n8n Webhook to Eleven Labs:**

- In **n8n**, open the **Webhook node** and copy the **Webhook URL**.
- In **Eleven Labs**, under **Functions**, create a **Custom Tool**:
  - **Name:** eg "n8n\_solar\_installers"
  - **Method:** POST
  - **URL:** Paste the **n8n Webhook URL**.
  - **Body Parameter:** Add "question" as the key (this sends customer queries to n8n).

**Edit tool**

**Tool type**

Webhook Client

A webhook tool allows the agent to extract specific information from the call and send it to your server.

**Configuration**  
Describe to the LLM how and when to use the tool.

**Name**  
n8n\_solar\_installers

**Description**  
n8n solar installers agent

**Method** **URL**  
POST <https://n8ndemos.app.n8n.cloud/webhook-test>

**Headers**  
Define headers that will be sent with the request [Add header](#)

Cancel **Save changes**

## Final Step: Add the Voice Widget to Your Website

Once your AI agent is fully set up and tested:

- Go to the **Widget** tab inside your Eleven Labs agent dashboard
- Copy the **embed code** provided under “Embed code”
- Paste it into your website’s HTML — ideally just before the closing `</body>` tag

✔ The voice call widget will now appear on your site, allowing visitors to **click and start a live conversation** with your AI-powered support & sales agent.

The screenshot shows the configuration interface for an AI agent. On the left, under 'AI Agents', there is a search bar and a list containing 'Support agent'. The main panel is titled 'Support agent' and includes a 'Public' status, a 'Test AI agent' button, and a list of tabs: 'Agent', 'Voice', 'Analysis', 'Security', 'Advanced', and 'Widget'. The 'Widget' tab is active, showing three sections: 'Embed code' with a code snippet, 'Feedback collection' set to 'None', and 'Appearance' with a 'Variant' dropdown set to 'Compact'. A preview of the widget shows a 'Need help?' button and a 'Start a call' button.

**AI Agents** Playground +

Search agents by name or ID...

Support agent ...

**Support agent** Public

IMxwkQF6qGdEIZmca5fu

Test AI agent

Agent Voice Analysis Security Advanced **Widget**

**Embed code**  
Add the following snippet to the pages where you want the conversation widget to be.

```
1 <elevenlabs-convai agent-id="IMxwkQF6qGdEIZmca5fu"></elevenlabs-convai><scr
```

**Feedback collection**  
No feedback will be collected. None

**Appearance**  
Customize the colors and shape of the widget to best fit your website.

**Variant**  
Compact

Need help?  
Start a call