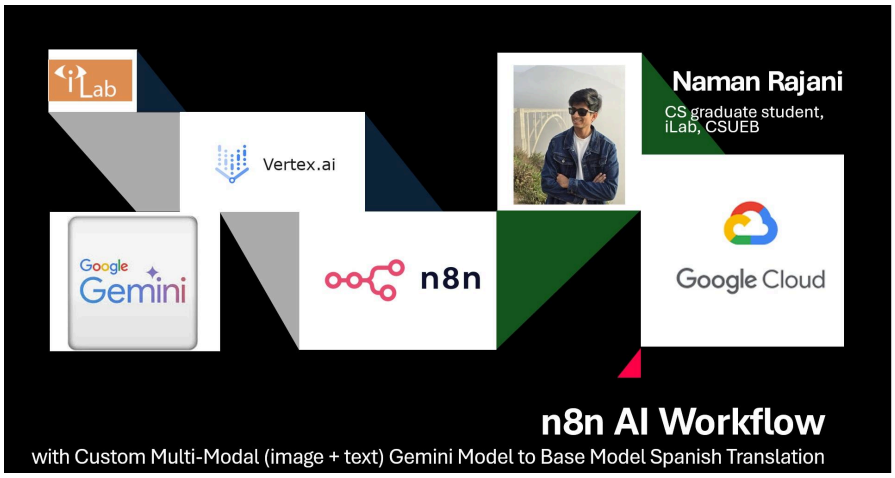
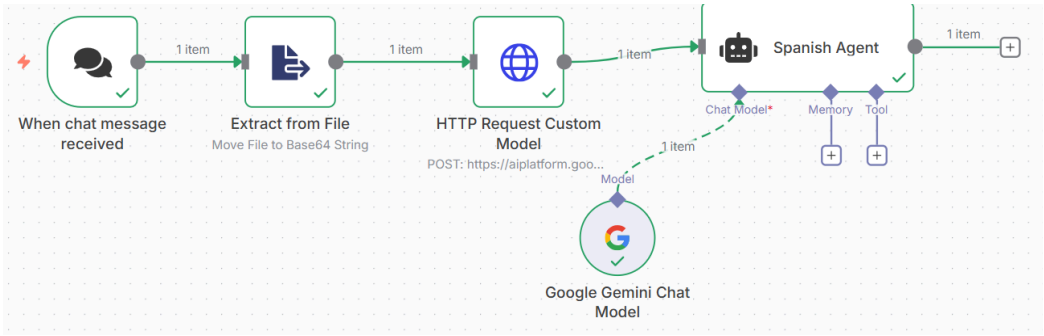


“n8n + Vertex AI Gemini: Using a fine-tuned multi-modal model (endpoint) and Gemini API in one workflow”

Overview



You’ll build an n8n workflow that:

- Accepts chat input (optionally with an image)
- Sends (image + prompt) to your fine-tuned Gemini model deployed on a Vertex AI Endpoint
- Uses an AI Agent for Spanish translation

  via Google Gemini (AI Studio) API via the n8n “Google Gemini (PaLM) API” node

---

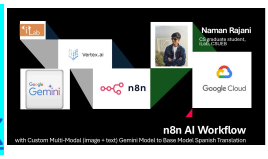
Prerequisites

- Node.js installed locally
- n8n installed and running locally (default: http://localhost:5678)
- Google Cloud project
- Fine-tuned Gemini model deployed to a Vertex AI Endpoint
- Google AI Studio API key (for the Gemini API node)

Keep these handy (you'll paste them later):

- PROJECT_ID
- LOCATION (region)
- ENDPOINT_ID
- OAuth Client ID and Client Secret (from Google Cloud Console)
- Google AI Studio API Key

VIDEO LINK



N8n AI workflow purpose & demo [start-2:00]

Part A — Vertex AI & OAuth Setup

1) Create/confirm the Vertex AI Endpoint [2:00-]

1. Go to Google Cloud Console → Vertex AI → Endpoints.
2. Click Create.
   - Name: your choice
   - Region: same as your model
   - Model: select your fine-tuned Gemini model
3. After creation, open the endpoint and note:
   - Endpoint ID (save as ENDPOINT_ID)
   - Region (save as LOCATION)
4. Also note your Project ID (save as PROJECT_ID).

2) Create OAuth 2.0 client (for n8n to call the endpoint) [2:00-3:50]

1. In Console, go to APIs & Services → Credentials

   IF you see "Remember to Configure the OAuth Consent Screen..." then FOLLOW THE INSTRUCTIONS

2. Click Create Credentials → OAuth client ID. (for web application)

3. Configure:

   - Name: e.g., n8n VertexAI OAuth
   - Authorized JavaScript origins:
     http://localhost:5678
   - Authorized redirect URIs:
     http://localhost:5678/rest/oauth2-credential/callback

4. Create it and download the JSON (you'll use Client ID and Client Secret in n8n).

Part B — n8n Credentials

1) Google OAuth2 API (for calling your custom endpoint) [3:50-5:57]

1. In n8n, click the down arrow next to Create Workflow (top-right) → Credentials → New.

2. Search Google OAuth2 API and select it.

3. From the downloaded JSON:

   Client ID: <your_client_id>

   Client Secret: <your_client_secret>

4. Scope: **https://www.googleapis.com/auth/cloud-platform**

5. Save, then click Connect and complete Google sign-in.

2) Google Gemini (PaLM) API (for calling Gemini via AI Studio key) [3:50-5:57]

1. Click here to create api key : https://aistudio.google.com/apikey

2. In n8n, create another credential → search Google Gemini (PaLM) API.

3. API Key: paste your Google AI Studio key.

4. Save.

Part C — Build the n8n Workflow [5:57-]

1) Chat Input node [5:57-6:30]

- Add Chat Input.
- Enable Allow file uploads (if you want to send an image to your fine-tuned model).

2) Extract from File (Binary → Base64) [6:30-7:25]

- Add Extract from File node.
- Input Binary Field: data0
- Destination Output Field: data
- Options:

    File Encoding: base64

    Keep Source: both

This converts the uploaded image to Base64 in $json.data and keeps the original binary + metadata in $json.files[0].

3) HTTP Request → Call your Vertex AI Endpoint (custom model) [7:25-13:44]

- Add HTTP Request node (name it e.g., Call Fine-Tuned Gemini).
- Method: POST
- URL:

https://aiplatform.googleapis.com/v1/projects/<PROJECT_ID>/locations/<LOCATION>/endpoints/<ENDPOINT_ID>:generateContent

You can hardcode the IDs, or pass them in from Chat Input using fields like $json.projectId, etc.

- Authentication: Predefined Credential Type
- Credential Type: Google OAuth2 API
- Google OAuth2 API: choose the credential you created earlier
- Send Headers: true

    Header: Content-Type: application/json

- Send Body: true

Body Content Type: JSON

JSON Body (copy-paste as is; n8n's expressions are supported inside {{ ... }}):

```
{
 "contents": [
  {
   "role": "user",
   "parts": [
    {
     "text": "{{ $json.chatInput }} what is this image about"
    },
    {
     "inlineData": {
      "mimeType": "{{ $json.files[0].mimeType }}",
      "data": "{{ $json.data }}"
     }
    }
   ]
  }
 ]
}
```

If you don't upload an image, remove the inlineData part or guard it with conditions.

4) AI Agent node (post-processing / orchestration)[13:44-14:48]

- Add AI Agent node.
- Prompt:

  {{ $json.candidates[0].content.parts[0].text }}

  (This pulls the text result from the previous node—adjust the path if your HTTP response structure differs.)
- Options → System Message: e.g.,

  Translate this response to Spanish.
  (or any instruction you want)

5) Google Gemini Chat Model (base model not custom) (AI Studio API) [14:48-16:05]

- Add Google Gemini Chat Model node.
- Credentials: select your Google Gemini (PaLM) API credential.
- Model: gemini-2.5-flash-lite (or pick another).
- Prompt: you can feed the output from the HTTP node, or your own text.

6) Wire it up [all-done as creating]

- Chat Input → Extract from File
- Extract from File → HTTP Request (Endpoint)
- HTTP Request → Google Gemini Chat Model (optional chain, or use either/or)

- Google Gemini Chat Model → AI Agent (or HTTP Request → AI Agent if skipping the Gemini API step)

7) Run workflow [16:05-16:49]

- Use Chat Input to pose question & upload image, hit enter & see results

---

Tips & Notes

- No image?
  Remove the inlineData section or add a conditional so the HTTP body only includes inlineData when a file exists.
- Secrets
  Don't commit your Client Secret or API Key to version control.
- Testing
  Start with a plain text prompt (no image) to confirm your endpoint and OAuth are working, then add image handling.

Quick Checklist

- Vertex AI Endpoint exists and is bound to your fine-tuned model
- Saved PROJECT_ID, LOCATION, ENDPOINT_ID
- OAuth2 credential in n8n (scope: Cloud Platform)
- Google AI Studio API key credential in n8n
- n8n workflow: Chat Input → Extract from File → HTTP Request → AI Agent → Gemini Chat Model

Reference :

https://cloud.google.com/vertex-ai/docs/reference/rest/v1/projects.locations.endpoints/generateContent