Manual de Instalación

Despliegue en Google Cloud VM (Medgemma)

Tópicos Avanzados en Base de Datos

Jherald Huren Cáceres Apaza



Tabla de contenido

[Introducción 3](#_Toc204951861)

[Objetivo de este Manual 3](#_Toc204951862)

[Documentos relacionados 3](#_Toc204951863)

[Requerimientos del VM 3](#_Toc204951864)

[1. Requisitos de Software 4](#_Toc204951865)

[1.1 Crear un proyecto en Google Cloud Console 4](#_Toc204951866)

[1.2 Habilitar Compute Engine y GPUs 5](#_Toc204951867)

[1.3 Crear instancia VM con GPU 5](#_Toc204951868)

[1.4 Configurar firewall para acceso remoto 6](#_Toc204951869)

[1.5 Conectar vía SSH a la VM 6](#_Toc204951870)

[2. Instalar otro Software 6](#_Toc204951871)

[2.1 Instalar drivers NVIDIA y CUDA 6](#_Toc204951872)

[2.2 Instalar Python y crear entorno virtual 7](#_Toc204951873)

[2.3 Instalar dependencias de MedGemma 8](#_Toc204951874)

[2.4 Configurar Hugging Face CLI e iniciar sesión 8](#_Toc204951875)

[2.5 Descargar y cargar el modelo MedGemma 4B IT desde Hugging Face 8](#_Toc204951876)

[3. Instalar el Producto 8](#_Toc204951877)

[3.1 Configurar y ejecutar el servidor API (FastAPI) 8](#_Toc204951878)

[3.2 Probar inferencia remotamente 9](#_Toc204951879)

[3.3 Abrir puertos para acceso externo 9](#_Toc204951880)

[3.4 Programar encendido y apagado automático de la VM (Schedule) 9](#_Toc204951881)

Manual de Instalación

Introducción

Objetivo de este Manual

Este manual describe el paso a paso para desplegar el sistema de orientación clínica “Consulta Médica IA” en una VM de Google Cloud Platform (GCP) usando FastAPI y el modelo MedGemma 4B, permitiendo aprovechar recursos escalables y la potencia de GPU para cargas grandes.

Documentos relacionados

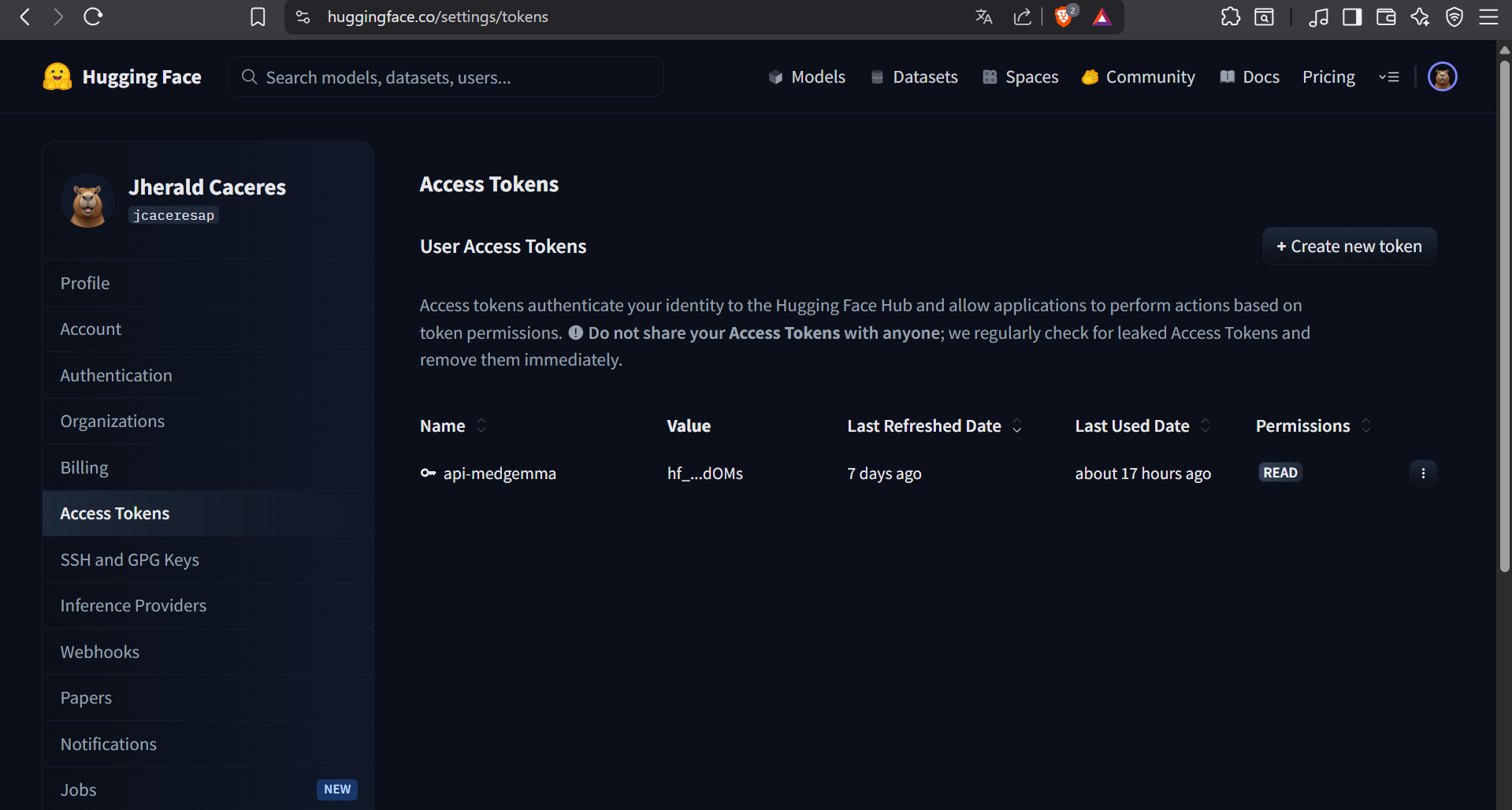
| **Id** | **Nombre** | **Descripción** |
| --- | --- | --- |
| 01 | Repositorio | <https://github.com/JheraldC/VM-Busqueda-Semantica-con-IA-Casos-Clinicos> |

Requerimientos del VM

|  |  |
| --- | --- |
| Nombre | medgemma-vm-ubuntu22 |
| Región / Zona | Escoge la que tenga disponibilidad de GPU (ej: southamerica-east1-c) |
| Procesador/Memoria de Sistema: | n1-standard-8 (8 vCPU, 30GB RAM) |
| Disco de arranque: | mínimo 40GB SSD |
| Sistema Operativo: | Ubuntu 22.04 LTS |
| Tarjeta Gráfica: | GPU NVIDIA T4 (1 unidad) |
| Acceso Red: | Sí, permitir tráfico HTTP y HTTPS |
| Costo Aproximado | 0.70 USD/hora |

# 1. Requisitos de Software

* Tener una cuenta de Google Cloud Platform con facturación activa.
* Saber usar el navegador y la terminal básica de Linux.
* Haber generado un token de Hugging Face.Instalación



## 1.1 Crear un proyecto en Google Cloud Console

* Accede a Google Cloud Console.
* Haz clic en “Seleccionar proyecto” > “Nuevo proyecto”.
* Ingresa el nombre y crea el proyecto.

## 1.2 Habilitar Compute Engine y GPUs

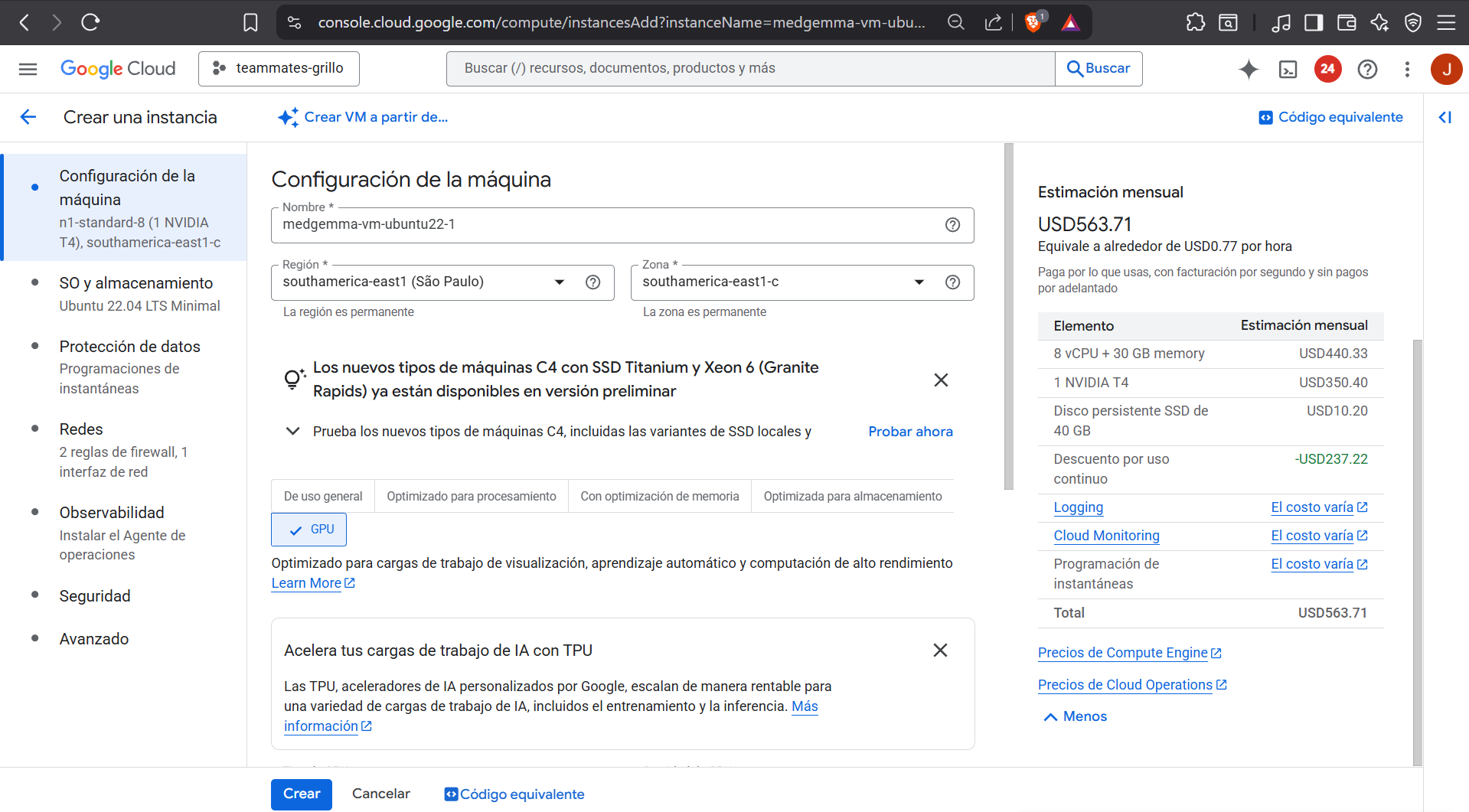
* Ve a **Compute Engine** > Instancias de VM.
* Si es la primera vez, habilita el servicio y espera la inicialización.
* Habilita la cuota de GPU (pide aumento de 0 a 1).

En <https://console.cloud.google.com/iam-admin/quotas> filtra por GPUS\_ALL\_REGIONS. La aprobación dura entre 5 minutos a 2 días hábiles



## 1.3 Crear instancia VM con GPU

* Haz clic en “Crear instancia”.
* **Sigue los requerimientos del VM**
* Asigna una **IP estática** si deseas acceso público permanente.



## 1.4 Configurar firewall para acceso remoto

* Marca “Permitir tráfico HTTP/HTTPS” si la API será pública.
* Abre el puerto necesario (por defecto 8000 para FastAPI) en el firewall de GCP.

## 1.5 Conectar vía SSH a la VM

* Desde la consola de GCP, haz clic en “SSH” para abrir una terminal remota.

# 2. Instalar otro Software

## 2.1 Instalar drivers NVIDIA y CUDA

sudo apt update

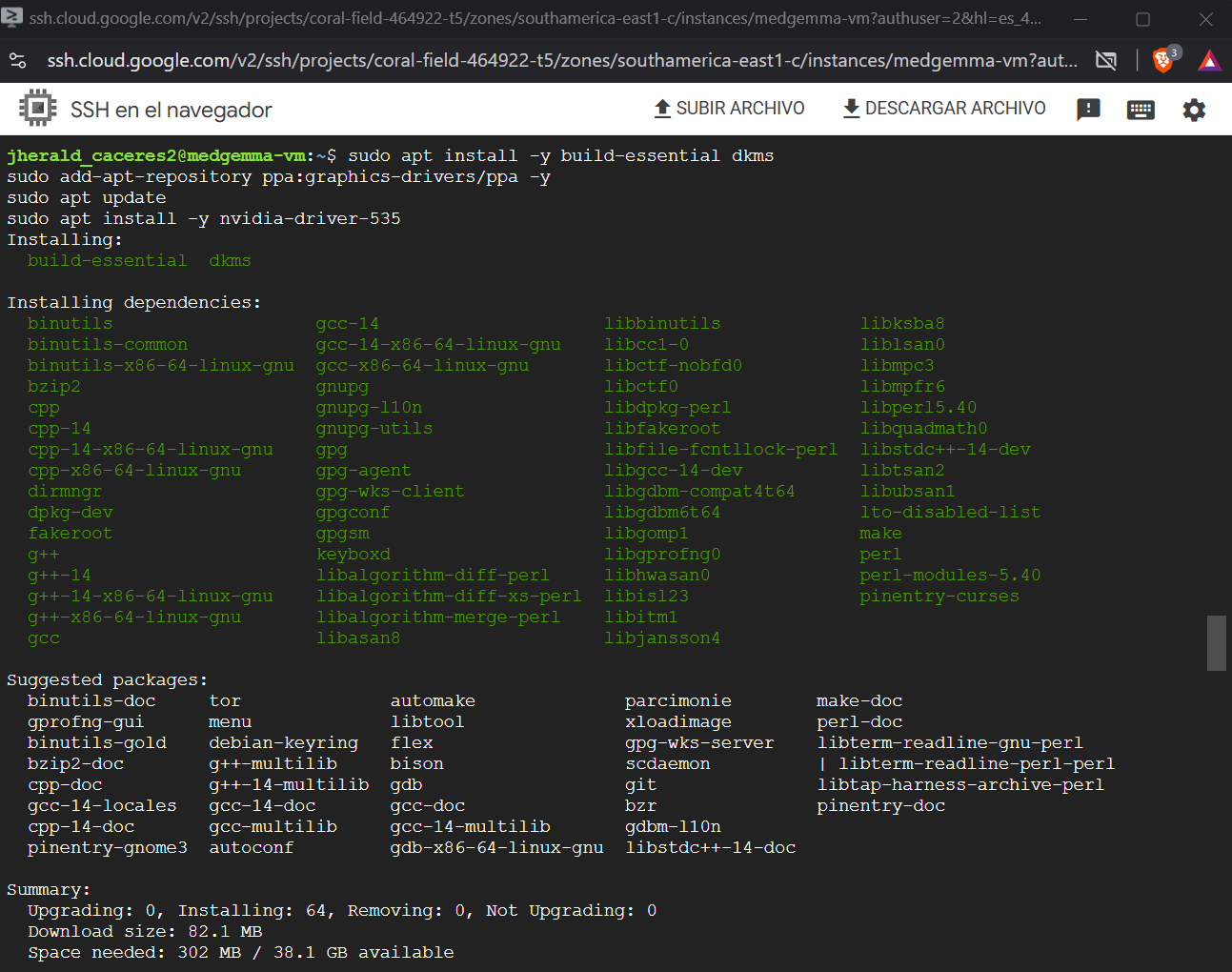
sudo apt install -y build-essential dkms

sudo add-apt-repository ppa:graphics-drivers/ppa -y

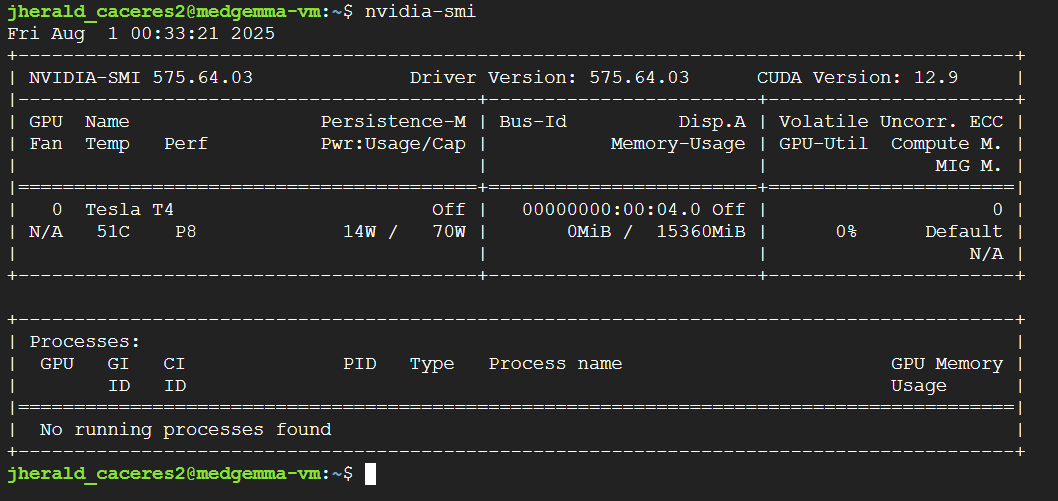
sudo apt update

sudo apt install -y nvidia-driver-575

sudo reboot



**Verifica instalación con:**  
nvidia-smi  
(debe mostrar la GPU y versión de drivers)



## 2.2 Instalar Python y crear entorno virtual

sudo apt install -y python3.12 python3.12-venv python3-pip git

python3.12 -m venv medgemma-env

source medgemma-env/bin/activate

pip install --upgrade pip

## 2.3 Instalar dependencias de MedGemma

Instala requirements del repositorio:

git clone https://github.com/JheraldC/VM-Busqueda-Semantica-con-IA-Casos\_Clinicos.git

cd VM-Busqueda-Semantica-con-IA-Casos\_Clinicos

pip install -r requirements.txt

Instala PyTorch con soporte CUDA:

pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu121

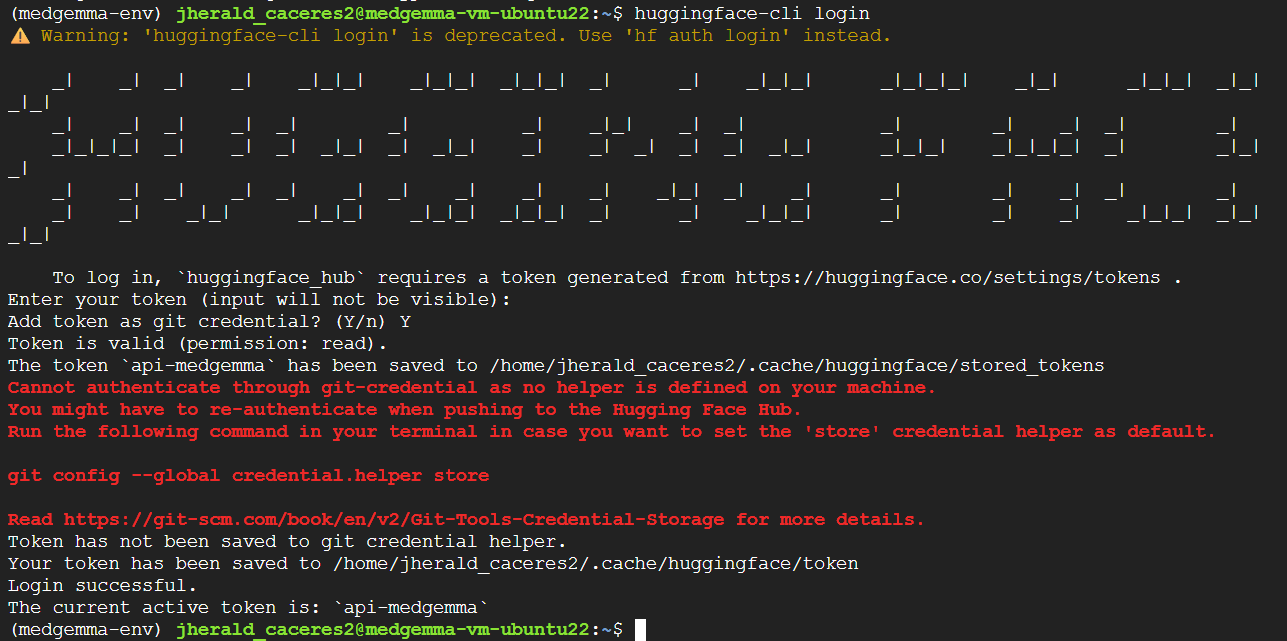
## 2.4 Configurar Hugging Face CLI e iniciar sesión

pip install huggingface\_hub

pip install huggingface-cli

huggingface-cli login

# Pega tu token de Hugging Face (Ctrl+V)



## 2.5 Descargar y cargar el modelo MedGemma 4B IT desde Hugging Face

El modelo se descargará automáticamente la primera vez que ejecutes el código (Siguiente paso).

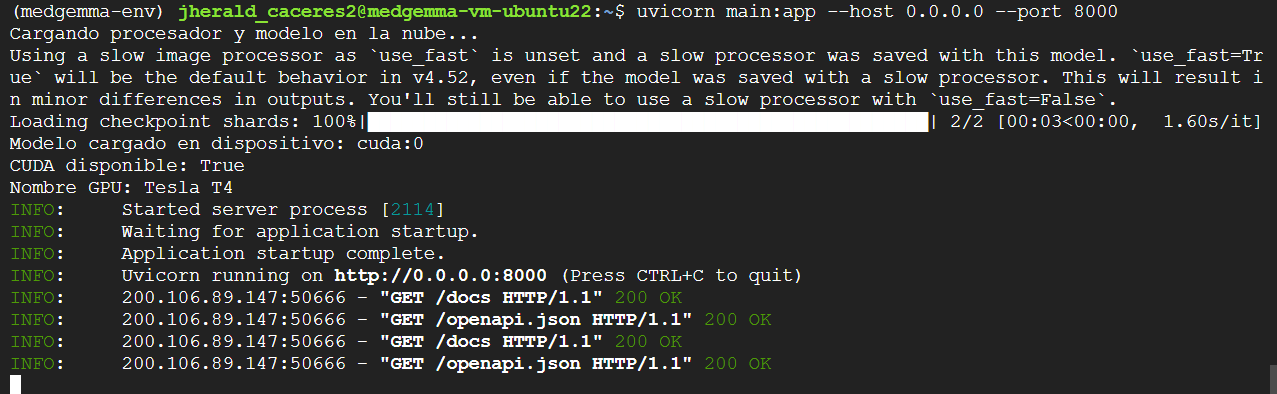
# 3. Instalar el Producto

## 3.1 Configurar y ejecutar el servidor API (FastAPI)

Lanza el servidor:

uvicorn main:app --host 0.0.0.0 --port 8000

La API estará disponible en la IP pública de tu VM, puerto 8000.



## 3.2 Probar inferencia remotamente

Remoto:  
Accede a http://[IP\_PUBLICA\_VM]:8000/docs desde tu navegador.

## 3.3 Abrir puertos para acceso externo

* Abre puertos 8000 en Firewall en GCP para todas las instancias de red
  + IPv4: 0.0.0.0/0
  + TCP: 8000

## 3.4 Programar encendido y apagado automático de la VM (Schedule)

Dejar una máquina virtual con GPU encendida todo el día puede generar costos elevados. Programar horarios automáticos de encendido y apagado permite ahorrar recursos y evitar cobros innecesarios.

**Pasos para crear un schedule en Google Cloud:**

**a) Accede a la sección de Programación de Instancias**

* Ve a **Compute Engine** > **Instancias de VM**.
* En el menú lateral, busca y haz clic en **Programación de instancias** (Instance Schedules).

**b) Crea un nuevo programa**

* Haz clic en **Crear programa** o **Create Schedule**.
* Completa los campos requeridos:
  + **Nombre:** Un identificador (ejemplo: my-schedule).
  + **Región:** Selecciona la misma donde está tu VM (ejemplo: southamerica-east1).
  + **Hora de inicio:** Hora en que se encenderá la VM (ejemplo: 07:00).
  + **Hora de detención:** Hora en que se apagará la VM (ejemplo: 18:00).
  + **Zona horaria:** Elige tu zona local (ejemplo: **hora estándar de Perú (PET)**).
  + **Frecuencia:** Elige si se repetirá todos los días u otro intervalo.

**c) Asocia el programa a tu instancia**

* Una vez creado el schedule, **asócialo a la VM** que deseas automatizar.
* Selecciona la instancia en la lista, haz clic en **Editar**, y busca la opción de "Programar encendido/apagado". Selecciona el programa creado.

**d) Guarda y verifica**

* Haz clic en **Guardar cambios**.
* Puedes ver y editar los schedules activos desde la misma sección.

