

การยืนยันตัวตน (Authentication) เพื่อใช้งาน **Vertex AI API** ครับ เนื่องจาก **Vertex AI** โดยทั่วไปแล้วจะไม่ใช่ **API Key** (ที่เป็นสตริงง่ายๆ) เหมือนกับ Google AI Studio แต่จะใช้ระบบที่ปลอดภัยกว่าที่เรียกว่า **บัญชีบริการ (Service Accounts)** หรือ **Application Default Credentials (ADC)**

นี่คือขั้นตอนการตั้งค่าที่คุณต้องทำเพื่อเรียกใช้ Vertex AI API ด้วย Python ในสภาพแวดล้อม Production:

## ขั้นตอนที่ 1: การตั้งค่า Google Cloud Project

1. ไปที่ **Google Cloud Console**:  
เข้าสู่ระบบ [Google Cloud Console](#)
2. เลือกหรือสร้าง **Project**:  
เลือก Project ที่คุณต้องการใช้งาน หรือสร้าง Project ใหม่ขึ้นมา
3. เปิดใช้งาน **Billing**:  
ตรวจสอบให้แน่ใจว่า **Billing (การเรียกเก็บเงิน)** ได้ถูกเปิดใช้งานแล้วใน Project ของคุณ (จำเป็นสำหรับการใช้งานจริง)
4. เปิดใช้งาน **Vertex AI API**:  
ไปที่ **APIs & Services > Library** ค้นหา "**Vertex AI API**" แล้วคลิก **Enable** (เปิดใช้งาน)

## ขั้นตอนที่ 2: การตั้งค่าบัญชีบริการ (Service Account) และสิทธิ์

สำหรับการใช้งานที่ปลอดภัยและสม่ำเสมอใน Production คุณควรใช้ Service Account:

1. สร้าง **Service Account**:
  - ไปที่ **IAM & Admin > Service Accounts** ใน Cloud Console
  - คลิก "+ **CREATE SERVICE ACCOUNT**" (สร้างบัญชีบริการ)
  - ตั้งชื่อ (เช่น `vertex-ai-translator-sa`) แล้วคลิก **DONE**
2. กำหนดสิทธิ์ (Role):
  - ไปที่ **IAM & Admin > IAM** (Identity and Access Management)
  - คลิก "+ **GRANT ACCESS**"
  - ในช่อง **New principals** ให้ค้นหาและเลือกชื่อ Service Account ที่คุณสร้างไว้
  - ในช่อง **Role** ให้ค้นหาและเลือกสิทธิ์ที่จำเป็นสำหรับการใช้งาน Gemini เช่น:
    - **Vertex AI User**
    - **Service Usage Consumer**
  - คลิก **SAVE**

## ขั้นตอนที่ 3: การยืนยันตัวตนสำหรับการพัฒนา (Local Authentication - ADC)

สำหรับนักพัฒนาที่ต้องการทดสอบโค้ดบนเครื่องตัวเองโดยใช้สิทธิ์ของ Service Account (วิธีที่แนะนำและปลอดภัยที่สุด):

1. ติดตั้ง **Google Cloud CLI**:  
ติดตั้งและตั้งค่า **gcloud CLI** บนเครื่องของคุณ (เป็นเครื่องมือ Command Line ของ Google Cloud)

## 2. ล็อกอินและตั้งค่า ADC:

เปิด Terminal/Command Prompt และรันคำสั่งนี้เพื่อยืนยันตัวตนและสร้าง **Application Default Credentials (ADC)** ซึ่งจะช่วยให้โค้ด Python ของคุณสามารถดึงสิทธิ์ไปใช้ได้โดยอัตโนมัติ:

Bash

```
gcloud auth application-default login
```

- คำสั่งนี้จะเปิดหน้าเว็บให้คุณล็อกอินด้วยบัญชี Google ที่มีสิทธิ์ใน Project นั้น

## 3. ตั้งค่า Project ID (ไม่บังคับ แต่แนะนำ):

ตั้งค่า Project ID ใน Terminal เพื่อให้ Client Library ทราบว่าจะเรียกใช้ Project ไດ:

Bash

```
gcloud config set project YOUR_PROJECT_ID
```

---

## ขั้นตอนที่ 4: การเรียกใช้ API ด้วย Python (Vertex AI SDK)

เมื่อคุณตั้งค่า ADC แล้ว โค้ด Python ของคุณก็พร้อมที่จะเรียกใช้โมเดล Gemini บน Vertex AI ได้เลย

### 1. ติดตั้ง Vertex AI SDK:

Bash

```
pip install google-cloud-aiplatform
```

### 2. ตัวอย่างโค้ด Python:

Client Library ของ Vertex AI จะดึงสิทธิ์การยืนยันตัวตนจาก ADC ที่คุณตั้งค่าไว้โดยอัตโนมัติ:

Python

```
from google.cloud import aiplatform
```

```
# กำหนดค่า Project ID และ Region
```

```
PROJECT_ID = "YOUR_PROJECT_ID" # แทนที่ด้วย Project ID ของคุณ
```

```
REGION = "asia-southeast1" # เลือก Region ที่รองรับ Vertex AI และ Gemini
```

```
# 1. Initialize the Client
```

```
# Client จะใช้ Application Default Credentials (ADC) โดยอัตโนมัติ
```

```
aiplatform.init(project=PROJECT_ID, location=REGION)
```

```
# 2. เตรียมข้อมูลสำหรับ Gemini
```

```
# โมเดล Gemini Pro บน Vertex AI
```

```
model_name = "gemini-2.5-flash"
```

```
prompt = "Translate this phrase into Thai: 'The silent watcher of the night sky.'"
```

```
# 3. เรียกใช้ API
```

```
try:
```

```
    response = aiplatform.predict_text_model(  
        endpoint=f"projects/{PROJECT_ID}/locations/{REGION}/models/{model_name}",  
        instances=[{"content": prompt}],  
    )
```

```
# 4. แสดงผลลัพธ์
```

```
print("--- คำแปลโดย Vertex AI (Gemini) ---")  
print(response.predictions[0].get('content', 'Translation failed'))
```

```
except Exception as e:
```

```
    print(f"An error occurred: {e}")
```

**สรุป:** ใน Vertex AI คุณไม่ได้สร้าง "API Key" แต่คุณสร้าง "**Service Account**" และใช้ `gcloud auth application-default login` เพื่อให้โค้ดของคุณสามารถยืนยันตัวตนได้อย่างปลอดภัยและเป็นไปตามมาตรฐาน Production ของ Google Cloud ครับ