

Running Gemini on Your Google Cloud \$300 Credit (Vertex AI Setup — Windows)

A short, repeatable guide to wiring a Windows machine up to Vertex AI so your Gemini calls bill against the Google Cloud Free Trial (\$300) credit.

Replace `YOUR_PROJECT_ID` with your actual Google Cloud project ID (found in Cloud Console, top project selector, or via `gcloud projects list`).

Commands below assume **PowerShell** (the default terminal in Windows Terminal / VS Code on Windows). Notes are included where Command Prompt (`cmd`) differs.

Prerequisites

- A Google Cloud account with the Free Trial (\$300) credit active
- A project created in that account, **linked to the trial billing account**
- Windows 10/11 with PowerShell

1. Install the Google Cloud CLI

Option A — winget (built into modern Windows):

```
powershell
winget install Google.CloudSDK
```

Option B — official installer: Download and run the installer from <https://dl.google.com/dl/cloudsdk/channels/rapid/GoogleCloudSDKInstaller.exe> (leave "Start Cloud SDK Shell" and "Run gcloud init" checked at the end).

Close and reopen your terminal, then confirm:

```
powershell
gcloud --version
```

If `gcloud` isn't recognized after install, fully close all terminals (and VS Code) and reopen so the `PATH` updates.

2. Authenticate

Log in with the Google account that holds the \$300 credit:

```
powershell
gcloud auth login
```

Set up Application Default Credentials (the identity Vertex AI uses — Vertex does **not** accept API keys):

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

3. Point gcloud at your project

```
powershell
gcloud config set project YOUR_PROJECT_ID
```

Set the quota/billing project for ADC (clears the "Cannot find a quota project" warning):

```
powershell
gcloud auth application-default set-quota-project YOUR_PROJECT_ID
```

4. Enable the Vertex AI API

```
powershell
gcloud services enable aiplatform.googleapis.com
```

5. Confirm billing is linked (optional check)

```
powershell
gcloud billing projects describe YOUR_PROJECT_ID
```

You want to see:

```
billingAccountName: billingAccounts/XXXXXX-XXXXXX-XXXXXX
billingEnabled: true
```

If `billingEnabled: false`, link the project to your trial billing account in Cloud Console
→ Billing before continuing.

6. Set environment variables

For your app: add these to your project's `.env` file:

```
bash
GOOGLE_CLOUD_PROJECT=YOUR_PROJECT_ID
GOOGLE_CLOUD_LOCATION=global
GOOGLE_GENAI_USE_VERTEXAI=True
```

For the current PowerShell session (if a tool reads them from the shell rather than `.env`):

```
powershell
$env:GOOGLE_CLOUD_PROJECT = "YOUR_PROJECT_ID"
$env:GOOGLE_CLOUD_LOCATION = "global"
$env:GOOGLE_GENAI_USE_VERTEXAI = "True"
```

To set them permanently (persists across new terminals):

```
powershell
setx GOOGLE_CLOUD_PROJECT "YOUR_PROJECT_ID"
setx GOOGLE_CLOUD_LOCATION "global"
setx GOOGLE_GENAI_USE_VERTEXAI "True"
```

`setx` only affects **new** terminal windows — reopen your terminal after running it.
In Command Prompt (`cmd`) instead of PowerShell, use `set NAME=value` for the session.

Notes:

- `GOOGLE_GENAI_USE_VERTEXAI=True` forces calls through Vertex AI (`aiplatform.googleapis.com`) instead of the AI Studio Developer API — **this is what makes the \$300 credit apply**.
- Use `GOOGLE_CLOUD_LOCATION=global` for the newest Gemini 3.x **preview** models (they're only on the global endpoint). For GA models, a region like `us-central1` also works.

7. Run it

Launch your coding CLI (e.g. Gemini CLI):

```
powershell
gemini
```

It should report `Authenticated with vertex-ai`. Pick a model and go.

Quick raw test (PowerShell), confirms the whole chain works:

```
powershell
```

```
$token = gcloud auth print-access-token
```

```
curl.exe -X POST `
```

```
-H "Authorization: Bearer $token" `
```

```
-H "Content-Type: application/json" `
```

```
"https://aiplatform.googleapis.com/v1/projects/YOUR_PROJECT_ID/locations/global/publishers/google/models/gemini-2.5-flash:generateContent" `
```

```
-d '{"contents":[{"role":"user","parts":[{"text":"say hello"}]}}'
```

Use `curl.exe` (not `curl`) in PowerShell — bare `curl` is an alias for `Invoke-WebRequest` and won't accept these flags. The backtick ``` is PowerShell's line-continuation character.

If it returns text, you're done — calls are landing on Vertex and drawing down the credit.

Where to watch the credit

- **Spend:** Cloud Console → Billing → Reports → filter Service = "Vertex AI" → enable the Credits breakdown. Posts with a few hours' lag.
 - **Credit balance:** Cloud Console → Billing → Credits.
-

Things worth knowing

- The \$300 credit works on **Vertex AI** (`aiplatform.googleapis.com`), **not** AI Studio / the Gemini Developer API (`generativelanguage.googleapis.com`). Same model names, two different billing paths.
- The credit expires **90 days** from signup, whatever's left.
- Once a project is billing-enabled, the trial's auto-stop is gone — overage hits your card. Set a budget alert (Billing → Budgets & alerts) at, e.g., 80% and 95% of \$300.
- Preview models (e.g. `gemini-3.1-pro-preview`) are the most expensive and drain the credit fastest. For routine work use a Flash model; reserve Pro for hard passes.
- Failed calls (4xx/5xx) aren't billed — only successful (200) responses draw the credit.