CloudFinOpsKit

GCP Cost Assessment — Quick Start

Get your first Google Cloud cost report in about 10 minutes. Plain-English, step-by-step. For a full end-to-end validation (including the trend/anomaly features that need billing-export history), see `VALIDATION-RUNBOOK.md`.

What you need

A **Google Cloud project** you can read (you're Owner/Editor, or have the read-only roles below).

That's it. **There is nothing to install by hand** — the launcher installs every prerequisite for you.

The easy way — one double-click

Windows: extract the ZIP, open the extracted folder, and double-click `GcpFinOpsTool.cmd`.

macOS / Linux: extract, open a terminal in the folder, and run `./gcp-finops-tool.sh`.

In one go, it will:

Step	What happens	Who does it
1	**Python 3** installed if missing	The launcher (winget / Homebrew / apt / dnf)
2	**Google Cloud CLI** installed if missing	The launcher, same way
3	A **private `.venv` folder** created here	The launcher
4	The tool's **Python packages** installed into it	The launcher
5	A browser opens for you to **sign in**	You — credentials are yours
6	Your project(s) scanned, **read-only**	The tool
7	Report generated and opened	The tool

The report and a `findings.v1.json` export land in the `output/` folder.

> **Your system Python is never modified.** Everything installs into a `.venv` folder inside the tool's own directory. Delete that folder and the tool leaves no trace on your machine.
>
> **Signing in is the one thing left to you.** The launcher only triggers Google's own browser sign-in when you have no credentials yet; it never sees, handles or stores them. If you're already signed in, it skips straight past.
>
> **If the package manager isn't available** (no `winget` on a locked-down Windows build, no Homebrew on a Mac), the launcher says so and gives you the direct download link rather than failing silently.

The manual way — if you'd rather drive it yourself

1. Turn on the APIs (once per project)

```
gcloud services enable \
  recommender.googleapis.com compute.googleapis.com \
  cloudbilling.googleapis.com billingbudgets.googleapis.com \
  bigquery.googleapis.com storage.googleapis.com \
  monitoring.googleapis.com aiplatform.googleapis.com
```

2. Sign in (Application Default Credentials)

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

3. Install and run

```
cd products/tools/GcpCostAssessment
python -m venv .venv
source .venv/bin/activate            # Windows: .venv\Scripts\activate
pip install -r requirements.txt

python start_cost_assessment.py --projects YOUR_PROJECT_ID
```

Open the report it prints: `output/GcpCostReport_<timestamp>.html` .

Prefer no install at all? Use **Cloud Shell** (the `>_` icon in the Cloud Console). Python and `gcloud` are pre-installed and already signed in, so steps 1–2 are done for you.

Useful options

Flag	What it does
`--projects a b c`	Scan several projects at once.
`--regions us-central1 europe-west1`	Limit the region set (default: a common multi-continent set).
`--label environment=production`	Only resources with that label (repeat for AND-combined keys).
`--billing-export project.dataset.table`	Point at your Cloud Billing BigQuery export (else it auto-discovers). Unlocks trend, forecast and anomaly.
`--webhook-url <Slack/Teams URL>`	Send Cost Anomaly Watch alerts (`--notify-min-severity High` by default).
`--output path.html`	Custom report path.

What you get

An interactive HTML report: a **FinOps maturity score**, **spend trend & forecast**, **Cost Anomaly Watch**, a **CUD optimization rate**, and a priced, filterable findings table — each priced from your **real** bill via Google's Active Assist recommenders.

Coverage mirrors the recommender set behind Google's own FinOps hub: idle VMs / disks / IPs / images, machine-type and MIG right-sizing, **GKE** cluster and workload sizing, **Cloud SQL** idle and over/under-provisioned, **Cloud Run** CPU allocation, resource-based **and** spend-based **CUDs**, idle and underutilised **reservations**, and **unattended projects** — plus storage tiering, network tier and Vertex AI token waste on top.

`output/findings_<ts>.json` (all findings) and, when anomalies fire, `output/anomalies_<ts>.json`.

`findings.v1.json` + `insights.json` — the canonical contract the **FinOps Agent** reads, so the same scan can drive the AI analyst.

Minimum read-only roles (if you're not Owner)

Grant on each project (and `roles/billing.viewer` on the billing account): `roles/recommender.viewer`, `roles/compute.viewer`, `roles/billing.viewer`, `roles/bigquery.dataViewer` + `roles/bigquery.jobUser`, `roles/storage.objectViewer` (+ bucket list), `roles/monitoring.viewer`.

Notes

Read-only. The tool never creates or changes anything in your account.

Trend / forecast / anomaly need history. A brand-new project has none; enable the BigQuery billing export and re-run once data accumulates (the tool still runs meanwhile and tells you the export is missing).

Costs pennies. BigQuery export storage + the tool's queries are a few cents at this scale.

Something skipped with a `!` warning usually means an API isn't enabled (step 1) or a role is missing — the tool degrades gracefully and never crashes on it.