

CloudFinOpsKit

How To Run the CloudFinOpsKit Tool — Easy Step-by-Step

This is the short, plain-English guide. If you can open PowerShell and sign in to Azure, you can run this tool. For the full technical detail see [README.md](#).

What this tool does (in one line)

It reads your Azure cost and resource data, then produces an interactive HTML report of where you can save money — always based on **last month's actual bill**, not list prices.

What you need before you start

#	Requirement	How to get it
1	PowerShell 7 or newer	The launcher checks this and can install it for you using <code>winget</code> .
2	Azure CLI	The assessment checks this and can install it for you using <code>winget</code> .
3	Access to Azure (the right roles — see below)	Ask whoever owns your Azure subscription.

You do **not** need to install anything else. The tool auto-installs the small `resource-graph` CLI add-on the first time it runs. If you prefer to install prerequisites yourself, use Microsoft's PowerShell and Azure CLI install guides.

Step 1 — Make sure you have the right access (RBAC roles)

You don't need to be an administrator. The tool works on a **read-only** basis. Here is what each role gives you:

Role you have	What you get in the report	Required?
Reader on the subscription(s)	The full resource inventory + every optimization finding (orphaned resources, rightsizing, hybrid benefit gaps, etc.)	✅ Minimum — the report runs on Reader alone
Cost Management Reader	Real dollar savings figures next to each finding (sourced from your actual bill)	★ Strongly recommended
Billing Reader (on the billing account)	A tenant-wide total that exactly matches the Azure Portal's billing view	Optional — only for the org-wide total
Monitoring Reader	CPU/usage data used for rightsizing accuracy	Optional

If you only have Reader

The report **still runs** and still lists every optimization opportunity. You just won't see the dollar amounts. Run it like this:

```
.\Start-CostAssessment.ps1 -SkipCostQuery
```

...or, if someone can export a cost CSV for you from the Azure Portal (**Cost Management** → **Cost analysis** → **Download**), point the tool at it to get the dollar figures without needing the Cost Management Reader role:

```
.\Start-CostAssessment.ps1 -CostExportCsvPath "C:\path\to\cost-export.csv"
```

How to request roles

If you're missing a role, ask your Azure administrator to assign it on the subscription (or management group). Microsoft's official guide: [Assign Azure roles using the portal](#). For the org-wide total, the **Billing Reader** role is assigned on the **billing account / contract**, not on a subscription — [Manage billing roles](#).

Step 2 — Run the report

Extract the ZIP first — don't run it from inside the ZIP. Right-click `CloudFinOpsKit-Tool.zip`, choose **Extract All...**, and pick any folder (e.g. your Desktop). If you double-click the tool while it's still *inside* the ZIP, Windows only unpacks that one file and the tool can't find its other files — so always extract first. (No admin rights are needed.)

Then open the **extracted** folder and run **the CloudFinOpsKit tool** — just double-click:

```
FinOpsTool.cmd
```

That one file does everything: it checks for PowerShell 7 and Azure CLI and installs them for you if missing, signs you in with `az login` (a browser opens), installs the `resource-graph` add-on if needed, and prompts you for the Azure subscription or tenant to scan. When it finishes, your report opens automatically. You do **not** need to know any PowerShell — just follow the on-screen prompts.

Seeing a “Publisher could not be verified / Unknown Publisher” warning? That’s normal and safe for any script downloaded from the internet — it is not an error. Just click **Run** (or **More info** → **Run anyway**). The launcher then removes the download flag from its own files, so you won’t be prompted again on later runs. To avoid the prompt even the first time, right-click the downloaded `CloudFinOpsKit-Tool.zip` → **Properties** → tick **Unblock** → **OK**, *before* extracting. (Windows doesn’t support digitally signing `.cmd` files, so this one-time prompt is expected — clicking Run is completely safe; the scan only reads your Azure data.)

If you prefer the command line, go to the tool folder and run the script. Pick the line that matches you:

```
# Go to the folder (adjust the path to where you saved it)
cd C:\Tools\AzureCostAssessment

# A) Easiest – it will ask you which subscription to scan
.\Start-CostAssessment.ps1

# B) Scan one specific subscription
.\Start-CostAssessment.ps1 -ScopeType Subscription -ScopeIds "<your-subscription-id>"

# C) Scan everything you can see (whole tenant)
.\Start-CostAssessment.ps1 -ScopeType Tenant
```

When it finishes, the HTML report opens automatically in your browser. A copy is also saved in the `output\` folder.

Step 3 — Read your report (the FinOps Insights band)

Your report opens with a **FinOps Insights** band — nine plain-English panels that sit above the detailed findings and answer the questions leadership actually asks:

Executive Summary — how much you can save, what % of your bill that is, and the top 3 things to do first.

FinOps Maturity Score — a 0–100 score (Crawl / Walk / Run) showing how mature your cost practice is and where to improve.

Commitment & Rate — savings available from Reservations, Savings Plans and Azure Hybrid Benefit, with payback guidance.

Cost Allocation & Showback — whether your tags are good enough to bill costs back to teams.

Top Cost Contributors — the subscriptions, services and resource groups spending the most.

Optimization Waterfall — your spend today vs. an optimized run-rate.

Inventory & Regions — what you're running and where.

Governance Posture — tag coverage, orphaned resources, budgets and anomalies at a glance.

Unit Economics — type in your business numbers (customers, transactions, users) and instantly see your **cloud cost per customer / per transaction**, now vs. after optimization. It calculates in your browser; nothing is sent anywhere.

You don't have to configure anything — it all appears automatically. The Unit Economics panel is the only one you type into, and it's optional.

What the numbers mean

The report always pulls **the previous full calendar month** of cost data. (Run it on 5 June and you get all of May.) This gives you a complete, settled month — not a half-finished one.

Savings figures equal what you **actually paid Microsoft** for each resource, including any EA/MCA discounts and reservations — so they're defensible line-by-line in Cost Analysis.

Step 4 — Run it every month (this is where the real value is)

This tool is built to be a **monthly habit**, not a one-off. Each run automatically saves a small snapshot, so from your **second month onward** the report grows a **Trends & Forecast** band that does the side-by-side comparison *for* you:

Month-over-month change in spend, savings, maturity score and findings — colour-coded so you can see at a glance whether things are improving.

Spend & savings trend charts, with a **forecast** of next month's spend.

Anomaly alerts — it flags any month where your spend jumped or dropped more than 20% versus the month before, so surprises get caught early.

Your first run shows a "baseline established" message; the trends appear from the second month. **The goal is simple: run it monthly, act on the quick wins, and watch the spend line fall and the maturity score climb** — that's your proof of return on investment over time. Use the included **Cost Review Checklist** as the agenda for the monthly review.

The history lives in a single file, `finops-history.json`, in the `output\` folder — nothing leaves your environment. To keep one shared history across teammates or machines, point the tool at a synced folder:

```
.\Start-CostAssessment.ps1 -ScopeType Tenant -HistoryPath "C:\Shared\FinOps\finops-history.json"
```

Easiest ways to run on schedule:

Manually — run it once at the start of each month (the dated HTML files also stack up in `output\`).

Windows Task Scheduler — create a Basic Task that runs monthly with:

Program: `powershell.exe`

Arguments: `-File "C:\Tools\AzureCostAssessment\Start-CostAssessment.ps1" -ScopeType Tenant`

Step 5 — Turn on Anomaly Watch alerts (catch surprises within a day)

Every run already includes the **Cost Anomaly Watch** band: it analyses ~3 months of daily spend per service and flags spikes, sustained increases, slow creep, and brand-new spend sources — including AI token anomalies. Nothing to configure; if the band is green, you're clean.

To turn it into a **daily early-warning system** that messages your team:

1. Create an **incoming webhook** in Teams (Channel → ... → Workflows/Connectors → Incoming Webhook) or Slack (App directory → Incoming WebHooks) and copy the URL.
2. Register the daily scheduled run and pass the webhook:

```
# One-time setup: daily rolling run at 06:00 + monthly run on the 3rd
.\Register-FinOpsSchedule.ps1 -HistoryDir "C:\Shared\FinOps" -ScopeType Tenant

# Or run ad hoc with alerts – posts to the channel only when a HIGH-severity
# anomaly is found (add -NotifyAlways for a daily "all clear" too)
.\Start-CostAssessment.ps1 -ScopeType Tenant `
    -WebhookUrl "https://hooks.slack.com/services/T000/B000/XXXX"
```

Anomalies **never inflate your savings figures** — they're reported separately with a "projected impact if sustained" figure. The full detection math (nothing is a black box) is in [ANOMALY-METHODOLOGY.md](#).

Quick troubleshooting

Problem	Fix
<code>pwsh</code> not recognised	Run the CloudFinOpsKit tool — double-click <code>FinOpsTool.cmd</code> ; it can install PowerShell 7 using <code>winget</code> .
Azure CLI missing	Let the assessment install it when prompted, or install it manually and rerun.
Not signed in	Let the assessment run <code>az login</code> when prompted.
No dollar figures in the report	You're missing Cost Management Reader — use <code>-SkipCostQuery</code> or <code>-CostExportCsvPath</code> (see Step 1).
Org total is lower than the Portal	You don't have access to every subscription. Add <code>-BillingAccountId "<id>"</code> with Billing Reader for the full total. Find the id with <code>az billing account list -query "[].name"</code> .

That's it. Reader access + `FinOpsTool.cmd` = your monthly cost optimization report.

What next — turn the report into a habit

One report finds savings. A **rhythm** banks them. The four rungs:

1. **Measure** — you just did this. Run it monthly.
2. **Optimise** — work the report's **Remediation plan** band, top-down; it's already ordered by payback.

3. **Govern** — take the action items into a review cadence (*Cloud Cost Governance Framework*) and stop the waste returning (*Tagging & Naming Standard*).
4. **Prove ROI** — record what you *actually* realised in the **FinOps Dashboard Pack** workbook. Projected savings are a forecast; realised savings are what protects next year's budget.

Want to ask questions instead of reading a report?

The **FinOps Agent** runs this *same* engine — identical findings — and adds an AI analyst you can ask "*what should I do first?*" or "*why is this disk costing \$120?*". It stays read-only in your own tenant, and it's **grounded**: it explains the numbers the engine measured and refuses to invent one, so every figure stays defensible in Azure Cost Analysis. It also shows Azure and AWS in one dashboard.

Included with Pro and Agency — cloudfinopskit.com/agent.