



MICROSOFT FABRIC FINOPS

The Four-Step Fabric FinOps Playbook

From baseline to accountability: what to do once you can see the problem

Microsoft Fabric's price is fixed. What runs on it is not. Four steps that turn visibility into control.

THE PLAYBOOK AT A GLANCE

- 1 Baseline **Know normal**
- 2 Organize **Decide what shares**
- 3 Control **Cheapest fix first**
- 4 Accountability **Name owners**

WHY A PLAYBOOK

Fabric's price is fixed. What runs on it is not.

Seeing which job caused the slowdown, and whose it is, is real progress. But visibility on its own doesn't fix anything. Without a plan, you watch the same spike happen every Monday, with better charts.

WHERE MOST TEAMS GET STUCK

"Fabric is slow."

WHERE THE PLAYBOOK GETS YOU

"Our refresh schedule is colliding with finance's reporting window, and here's what we're changing."

THE FOUR STEPS

Work through them in order. Each builds on the last.

- | | | |
|---|--|----------------|
| 1 | Establish a capacity baseline
Know what “normal” looks like before you change anything. | Baseline |
| 2 | Organize workloads intentionally
Decide what shares capacity, or Fabric decides for you. | Organize |
| 3 | Apply the right controls
Try the cheapest fix first. Resizing is the last move. | Control |
| 4 | Create accountability
Once people can see their own number, behavior changes. | Accountability |

STEP 1 OF 4 · BASELINE

Establish a capacity baseline

Know what “normal” looks like before you change anything. Without a baseline, every spike feels like an emergency, and every fix is a guess.

KEEP IN MIND

The Capacity Metrics App keeps only a short window of history. Export snapshots so you can compare over time.



A useful baseline answers

- **Who consumes the most?** Workspaces and items behind most of your use
- **What’s recurring, and what’s a one-off?** A weekly pattern is planning; one spike is an incident
- **When do users actually feel it?** Complaints matter more than one metric moving



Watch for

- Skipping the baseline in the middle of an incident
- Spikes that feel random because nobody mapped them to the calendar



This week

- ☐ Pull at least two weeks of consumption from the Capacity Metrics App, by workspace and item
- ☐ Flag your top five consumers, and mark each spike as recurring or one-time
- ☐ Overlay your business calendar: month-end close, launches, refresh waves

STEP 2 OF 4 · ORGANIZE

Organize workloads intentionally

Decide what shares capacity, or Fabric decides for you: everything shares. That's how critical dashboards end up competing with an experimental pilot.

PROTECT FIRST

High-value interactive work: the reports leadership opens every morning. Keep heavy background jobs out of their way.



Four common ways to split capacity

Dev/test vs. production

By business domain

By workload type

AI vs. established reporting



Watch for

- "Everything's on one capacity. That's where it landed."
- Or the opposite: so many small capacities that compute sits unused



This week

- ☐ List the dashboards and reports leadership relies on every day
- ☐ Check which background jobs share their capacity and run during their hours
- ☐ Decide where one intentional boundary is needed, and resist creating ten

Apply the right controls, cheapest fix first

The order matters most here. Resizing belongs at the end of the list, not the top.

1

Optimize

the workload, if it consumes far more than it should even when it runs alone

Costs: time

2

Reschedule

background work, if it collides with busy interactive hours

Costs: time

3

Isolate

critical workloads, if they keep absorbing the impact of other teams' jobs

Costs: a boundary

4

Resize

the capacity, only if utilization stays high around the clock after 1–3 are ruled out

Costs: your bill, every month

THIS WEEK ☐ Take your biggest recurring issue and work through this order before touching the SKU.

Precision tools for specific problems

Each is useful in the right situation. None of them is a strategy on its own.



Autoscale Billing for Spark

REACH FOR IT WHEN

Spark jobs compete with BI on shared capacity. It moves them to separate pay-as-you-go compute.

KEEP IN MIND

Spark is billed pay-as-you-go, outside smoothing, and you still need base capacity for everything else.



Surge protection

REACH FOR IT WHEN

A background job keeps running hot. It caps background consumption so one job can't drag down interactive work.

KEEP IN MIND

It limits background use, but doesn't guarantee interactive users never see a delay.



Capacity overage

REACH FOR IT WHEN

You need short-term breathing room. Pay a metered rate instead of being throttled, within a limit you set.

KEEP IN MIND

It's a premium rate: a tool for a genuine crunch, not a substitute for right-sizing.

STEP 4 OF 4 · ACCOUNTABILITY

Create accountability

Once people can see their own number, **behavior changes**. When the next backlog hits, “whose job caused this?” should have an immediate answer.

KEEP IN MIND

Steps 1–3 fix today’s problem. Step 4 is what keeps it from coming back.



What accountability looks like

- **Two owners:** business for value, technical for consumption
- **Showback:** usage visible to the team that creates it
- **A rhythm:** regular reviews, not only after something breaks



Watch for

- The platform team listed as owner of everything
- Capacity reviewed only after throttling or a surprise bill



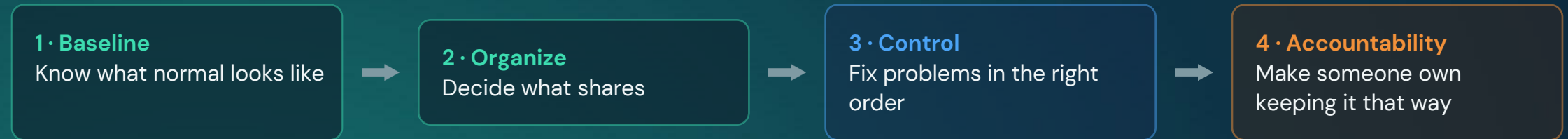
This week

- ☐ Name a business owner and a technical owner for each major workspace
- ☐ Share a showback view with those owners. Chargeback can wait.
- ☐ Put a recurring capacity review on the calendar [cadence, e.g. monthly]

THE GOAL

The goal isn't the smallest bill.

It's matching spend to value while protecting the workloads your business depends on.



Fabric Capacity Health Check

Not sure your team could answer all five? That's exactly what the Lantern Fabric Capacity Health Check is for.

The first move probably isn't buying more capacity. It's a clear baseline. We work with your team to find what's driving your consumption, where performance or cost risk is emerging, and which actions to prioritize. It's built on the same framework as this checklist: sizing, organization, controls, and accountability.



What's actually driving your consumption



Where the risk is building



What to fix first



Book your Health Check: lanternstudios.com/fabric-health-check



KEEP GOING

Go deeper on every step



Watch the webinar on demand

Control Microsoft Fabric Costs: A FinOps Playbook for Capacity and Throttling



Read the Fabric FinOps series

Sizing decisions, smoothing and throttling, monitoring, the four-step playbook, and real-world scenarios.



Run the Five-Question Checklist

Test whether your team can answer five capacity questions with data, or only with a best guess.

