

Guide

Overspending on Microsoft Fabric?

Key warning signs,
practical fixes, and a
30-day action plan

Table of Contents

Getting Oriented

Why Fabric Spend Gets Away From You	3
Capacity Options: Pay-as-You-Go vs. Reserved	4
Signs You Have a Fabric Spending Problem	5

Key Warning Signs

Persistent Throttling	6
Excessive Unused Capacity	
No Cost Visibility	
Uncontrolled Data Movement	
Capacity Reserved but Paused	
Fragmented, Underutilized Workspaces	
No Real-Time Usage Visibility	

How to Fix It

Immediate Actions to Fix Fabric Overspend	13
Sizing and Licensing Fixes	14
Workload and Storage Cleanup	15

Your Immediate Action Plan

What to Change in the Next 30 Days	16
30-Day Action Plan	17

Overview

In this guide to Microsoft Fabric

It's not uncommon for companies to find significant waste in their Fabric deployment, potentially losing 20-30% of their investment to poor setup. In this guide, we show you how not to.

- Spot **warning signs** like chronic throttling, capacity utilization below 50%, and fragmented, underused capacities.
- Use **Fabric monitoring tools** to see which workspaces and workloads drive costs and where 20–30% waste is hiding.
- **Right size capacity**, choose the right mix of Pay-as-You-Go and Fabric capacity SKUs (up to about 40% cheaper when used consistently), and automate off hours scaling.
- **Clean up inactive workspaces**, datasets, and OneLake storage to eliminate ongoing costs without sacrificing performance.

Signs You May Have a Fabric Spending Problem

Getting your Fabric investment back in shape

Every January, eager New-Years-Resolution setters flock to the gyms to buy a membership that most won't use.

Think of Fabric capacity like a gym membership: everyone swears they will use it every day, but a quiet 20-30% of the spend just sits there gathering dust – in this case, through inefficiencies and underused capacity.

Your total Fabric cost is driven by **provisioned capacity**, measured in Capacity Units (CUs), so the central question is whether your capacity profile reflects **actual usage patterns**. The closer the match, the lower your effective unit cost.

Understanding Fabric Spend

Capacity options

Capacity in Microsoft Fabric comes in two flavors: one that bends with your workload, and one that rewards you for running flat-out, all day, every day.

Option	Best for	Key benefit	Watch out
Pay-as-You-Go (PayG)	Sporadic or highly variable workloads	Scale CUs up or down and pause capacity when idle	Costs can spike if you forget to pause during off hours
Reserved capacity	Consistent, predictable workloads with steady utilization	Up to 40% discount compared with PayG	Billed continuously; cannot pause to save costs

1

Persistent throttling

2

Unused capacity

3

No cost visibility

4

Costly data movement

5

Paused reserved cap.

6

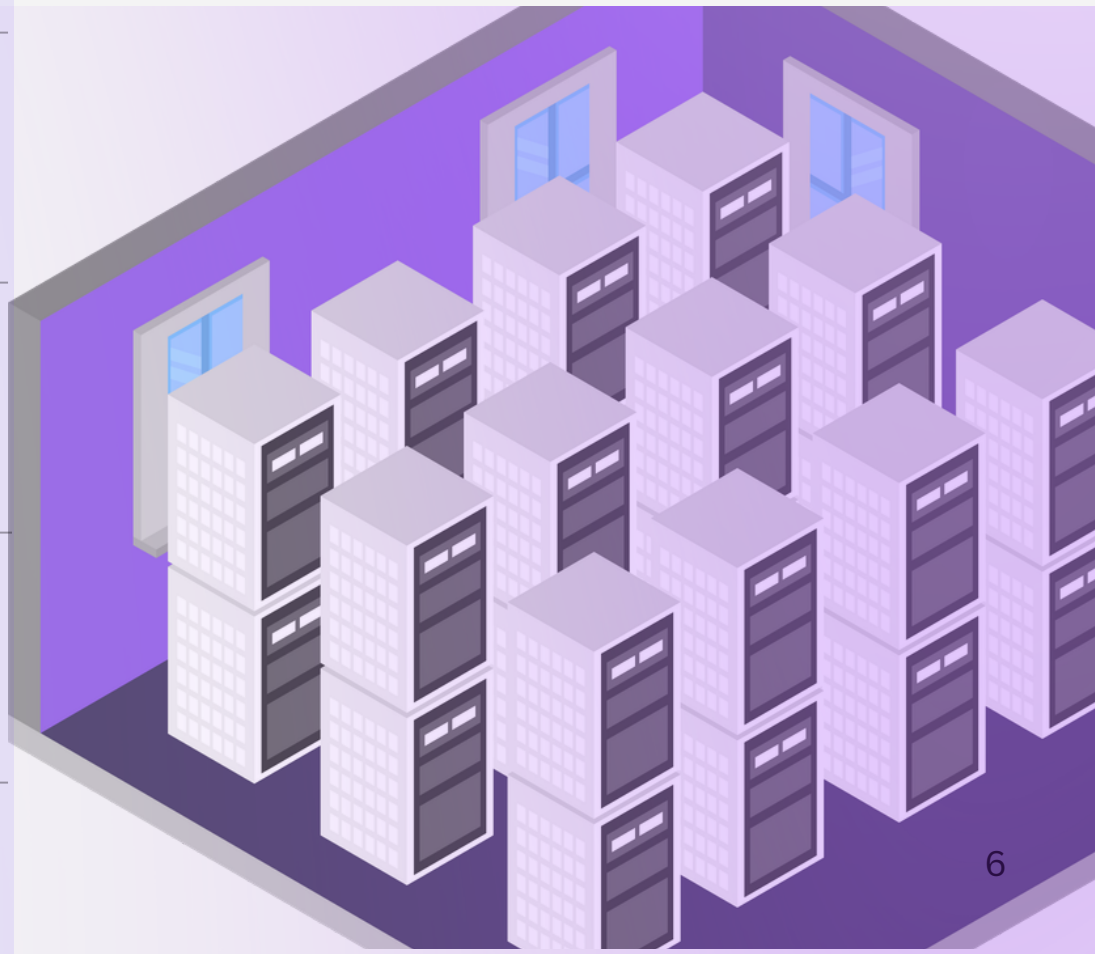
Underutilized workspace

7

No real time visibility

Key Warning Signs and Indicators

A handful of common patterns most often signal a Microsoft Fabric spending problem long before the finance team raises a red flag.



1

Persistent throttling

2

Unused capacity

3

No cost visibility

4

Costly data movement

5

Paused reserved cap.

6

Underutilized workspace

7

No real time visibility

Persistent throttling and performance degradation

When pipelines drag and jobs stall, Fabric is telling you your capacity design is wrong.

Throttling is what happens when CUs run out faster than your SKU can deliver, and short bursts aside, persistent throttling almost always means you are paying more for a misconfigured platform than for real performance.

1

Persistent throttling

2

Unused capacity

3

No cost visibility

4

Costly data movement

5

Paused reserved cap.

6

Underutilized workspace

7

No real time visibility

Excessive unused capacity

Nothing says “cloud waste” quite like an F64 that idles at 10%. In Fabric there is no rainy-day jar for unused CUs; every hour you do not use them, they simply drain from the balance sheet.

Fragmented, over-sized capacities across regions and departments turn into a steady leak.

1

Persistent throttling

2

Unused capacity

3

No cost visibility

4

Costly data movement

5

Paused reserved cap.

6

Underutilized workspace

7

No real time visibility

Lack of cost visibility and monitoring

If you cannot point to the three workspaces and five pipelines that dominate your Fabric bill, you are flying the platform blind.

Absence of proper telemetry does not just hide waste; it removes any rational basis for prioritizing fixes or defending spend in front of finance.

1

Persistent throttling

2

Unused capacity

3

No cost visibility

4

Costly data movement

5

Paused reserved cap.

6

Underutilized workspace

7

No real time visibility

Uncontrolled data movement costs

Treating data movement as “just plumbing” is an expensive habit.

Moving terabytes back and forth through poorly designed pipelines burns CUs quickly, and the wrong pattern can make Fabric look pricier than tools it should easily beat, purely because the traffic plan was never drawn.

1

Persistent throttling

2

Unused capacity

3

No cost visibility

4

Costly data movement

5

Paused reserved cap.

6

Underutilized workspace

7

No real time visibility

Capacity reserved but paused

A paused reserved capacity is the cloud equivalent of heating an empty office.

You keep paying the reservation every month, regardless of whether anything runs on it, which is usually a sign that assumptions have changed but the commercial model has not caught up.

1

Persistent
throttling

2

Unused
capacity

3

No cost
visibility

4

Costly data
movement

5

Paused
reserved cap.

6

Underutilized
workspace

7

No real time
visibility

Multiple underutilized workspaces and capacities

Carving capacity into small silos for every region or team feels orderly and fair, right up until you compare utilization.

One island runs at 90%, another at 30%, and in aggregate you are buying far more CUs than a shared, well-governed pool would need.

1

Persistent throttling

2

Unused capacity

3

No cost visibility

4

Costly data movement

5

Paused reserved cap.

6

Underutilized workspace

7

No real time visibility

Limited real time usage visibility

If nobody on the team can sketch your peak hours, quiet periods, and seasonal spikes, capacity decisions become superstition rather than engineering.

Without a clear view of demand, you either oversize to sleep at night or accept chronic throttling, instead of using automation to scale with intent.

Immediate Actions to Fix Fabric Overspend

Five moves you can make this quarter

1

Deploy monitoring and visibility tools

Implement the Fabric Capacity Metrics app and FUAM so you can see which workspaces, datasets, and pipelines actually consume capacity. Once you have hourly utilization and throttling in one place, optimization stops being guesswork.

2

Audit and right size your capacity

Analyze at least 30 days' usage to separate true undersizing from short spikes. Size your SKU for baseline demand plus a 15–20% buffer, and consider reserved capacity when utilization stays above 60% – when the up-to 40% discount is realistic.

3

Consolidate underutilized capacity

Find capacities running below 50% utilization and merge low value or overlapping workspaces into shared pools. Consolidation removes stranded resources and smooths load across your Fabric estate.



4

Eliminate inactive datasets, reports, and workspaces

Retire datasets that haven't refreshed in 90 days; reports with no access in 60 days; and workspaces without a clear owner. This cuts needless refresh and admin overhead and reduces OneLake storage that would otherwise continue to accrue cost even when capacity is paused.

5

Implement automated scaling and off hours management

For PayG capacity, use Logic Apps or Azure Functions to pause outside business hours and resume before the day starts. Automate scale up and down around predictable peaks, and remember that pausing reserved capacity does not save money.

Immediate Actions: What to Change in the Next 30 days

You don't need a six-month program to stop immediate Fabric challenges.

In the first 30 days, focus on a few configuration changes, and one review rhythm.

1

Observe

Turn on the Fabric Capacity Metrics app, baseline usage by day and hour, and flag chronic over- and under-utilization.

2

Decide

Agree which capacities to right-size, which workloads to reschedule, and which assets to clean up or retire.

3

Act

Implement a few changes (such as removing stale, unused data sets and inactive users), then compare CU usage and performance week-over-week.

TL;DR:

Sizing and licensing fixes

Use these levers when your warning signs point to a **sizing or licensing issue**.

Capacity management strategy	What to do
Right-size your capacity	Avoid paying for more CUs than you need. Use the Fabric Capacity Metrics app to look at real usage and move your F-SKU up or down instead of guessing.
Automate pausing and scaling	For PayG capacity, use Logic Apps or Azure Functions to pause at night and on weekends, and to scale up only for predictable peaks.
Use reserved capacity wisely	Reserve capacity for workloads that run steadily and keep utilization high, and handle the odd spike with Pay-as-you-go instead of buying a bigger SKU.
Optimize licensing	For F64 and above, most report viewers do not need a separate Power BI Pro license. Audit your estate so you stop paying author prices for read-only consumers.

TL;DR:

Inefficient and low-value cleanup

Use these patterns when the problem is **inefficient workloads** or **too much low-value data**.

Optimization strategy	What to do
Reduce unnecessary data movement	In Fabric's OneLake world, process data where it already lives instead of shuttling it between engines or regions and burning CUs on plumbing.
Optimize storage	Delete unused assets, compact many small Delta files into larger ones, and clean up orphaned data so you are not paying to store noise month after month.
Improve workload efficiency	Tune Spark notebooks and SQL queries, use pushdown filters and caching, and run heavy jobs away from interactive peaks so capacity is doing useful work.
Optimize refreshes	For large semantic models and dataflows, switch from full refreshes to incremental refresh so you scan less data, finish sooner, and still keep reports current.

Want to learn more?

More resources to help you boost your bottom line with Microsoft Fabric.

What Managed Self-Service Should Look Like in Microsoft Fabric: A Step-by-Step Guide



From Chaos to Clarity: How to Clean up Your Data Model



Bringing Scalable Standards and Governance to an Ungoverned Environment: Microsoft Fabric



AI Without a Strong Data Culture is a Castle Built on Sand





Let's get to
work today,
together.

To discover how we
can help you do more
with data, book a
complimentary 1:1 with
one of our specialists.

Start with a free 1:1



About DI Squared

DI Squared empowers organizations to extract the full value of their data through enhanced data platforms and processes.

We engineer flexible, reliable data infrastructure – from data lakes and pipelines to dashboards and augmented analytics – ensuring long-term scalability.

To learn more, visit

www.disqr.com

About DI Squared

DI Squared empowers organizations to extract the full value of their data by engineering data platforms and processes. We engineer flexible, reliable data analytics infrastructure – from data lakes and pipelines to dashboards and augmented analytics – ensuring long-term scalability. To learn more, visit www.disqr.com