



FINOPS & COSTS

FinOps on Azure optimizing costs and reservations

An operational guide to regaining control of your Azure bill: real cost visibility, trade-offs between reservations, savings plans and Spot, rightsizing and lasting governance.

11 pages · Commitment decision matrix, tagging model, audit grid and roadmap included

72%

maximum saving announced by Microsoft for a 3-year Azure reservation

65%

maximum saving for a 3-year Azure Savings Plan for Compute

90%

potential saving on Spot VMs, for interruptible workloads

CONTENTS

What you will find in this document

This document starts from the economics of the Azure cloud, details the technical mechanisms of optimization and commitment, then offers a roadmap and a checklist you can apply directly.

01	Executive summary — the essentials on one page	p.3
02	Why Azure FinOps Is Unavoidable	p.4
03	Visibility, Tagging and Allocation	p.5
04	Reservations, Savings Plans and Spot	p.6
05	Rightsizing, Elasticity and PaaS	p.7
06	FinOps Governance and Culture	p.8
07	Self-assessment grid	p.9
08	A 5-step roadmap	p.10
09	Conclusion & about Move2Cloud	p.11

Who is this document for? CTOs, CIOs, platform and FinOps leads, cloud architects and finance/IT procurement teams managing a significant Azure footprint (from a few dozen to several thousand resources) and looking to cut the bill sustainably without slowing product teams down.

EXECUTIVE SUMMARY

The essentials on one page

Regaining control of an Azure bill is not a matter of commercial negotiation but of engineering discipline: measure, arbitrate, commit, then start again.

Cloud spend has moved from the status of a variable budget line to that of a major structural cost. For several years now, Flexera's annual surveys have ranked cloud cost optimisation as the **number-one priority** of the organisations surveyed, which estimate they waste **around 30%** of their spend. On the Azure side, Microsoft advertises discounts of up to **72%** on a 3-year reservation versus pay-as-you-go, and up to **90%** on Spot virtual machines: the gap between a managed footprint and an unmanaged one runs into millions of euros over three years for a mid-sized IT department.

This tension is worsening for two reasons. First, the Azure pricing grid has become combinatorial: reservations, compute savings plans, Spot, Azure Hybrid Benefit, Dev/Test rates and MCA/CSP contractual discounts partially overlap, and a poorly targeted commitment can remain under-used for thirty-six months. Second, the rise of data and AI workloads — GPUs, hot storage, inter-region traffic — shifts the cost base towards expensive, barely elastic resources that are poorly covered by the historical FinOps reflexes centred on the general-purpose VM.

The good news is that these levers can largely be mechanised. Reliable cost allocation (tags, management groups, billing scopes), rightsizing driven by real metrics and a commitment coverage built on the stable baseline — not on the peak — deliver quick wins without slowing product teams down. This document provides the end-to-end method: visibility model, rules for arbitrating between reservations, savings plans and Spot, quantified decision thresholds and governance rituals so that the savings achieved do not erode by the next quarter.

72%

maximum discount advertised by Microsoft for a 3-year reservation versus pay-as-you-go

≈ 30%

share of cloud spend considered wasted by respondents to the Flexera survey

5

Azure pricing levers to arbitrate between: RI, savings plan, Spot, Hybrid Benefit, Dev/Test

The question this document addresses is therefore not "how do we pay less for Azure this quarter", but "how do we build a cost structure that stays optimal while the infrastructure itself changes constantly". The sections that follow set out an eight-step method — from the visibility model through to the governance of commitments — together with an operational checklist that can be applied directly to your tenant.

SECTION 02

Why Azure FinOps Is Unavoidable

Azure FinOps is not a governance fad: it is the answer to three simultaneous dynamics — spend that has become structural, increasingly combinatorial pricing, and cost decisions taken by engineering teams. Here is why the subject can no longer wait for the next budget cycle.

1. 1. The cloud has become a structural cost line

Cloud spend is no longer a project budget but a structural cost line, scrutinised in the same way as payroll. In Flexera's annual surveys, optimising cloud costs remains respondents' number-one priority, and they estimate they waste close to a third of their spend.

2. 2. A pricing grid that has become multi-dimensional

Azure now layers at least five pricing levers on top of one another: 1- or 3-year reservations, compute savings plans, Spot, Azure Hybrid Benefit and Dev/Test rates. With advertised discounts of up to 72% (3-year RI) and 90% (Spot), the trade-off is anything but intuitive.

3. 3. Cost is decided in the code, not in the spreadsheet

Most of the bill is decided upstream, in the code and the IaC templates: SKU family, redundancy level, storage tier, inter-region traffic. Without a feedback loop back to the product teams, FinOps remains an after-the-fact accounting exercise.

“ On Azure, the discount is not negotiated once a year: it is designed, measured and steered every month. ”

The following sections detail the mechanics: real cost visibility, trade-offs between reservations, savings plans and Spot, rightsizing, then durable governance.

SECTION 03

Visibility, Tagging and Allocation

On an ungoverned Azure bill, it is common for 20 to 40 % of spend to be attributable to no identified application: until that figure is close to zero, any optimisation remains a matter of opinion.

PRACTICE	WHAT IT DELIVERS
Management group hierarchy	Structuring subscriptions by entity, environment and criticality provides a first native allocation axis, independent of tagging quality, and the natural enforcement point for policies.
Azure Policy in Modify mode + remediation	Rather than auditing, you enforce: automatic inheritance of tags from the resource group down to resources, and denial of creation when mandatory tags are missing, with remediation tasks on the existing estate.
Daily export of cost data	A scheduled export of amortised costs to a storage account, then its exploitation in Power BI or a warehouse, lifts the limits of the Cost Analysis interface and makes it possible to cross-reference costs, tags and business metrics.

Key takeaway: Freeze a short taxonomy — 5 to 6 tags maximum (owner, application, environment, cost centre, criticality) — make it mandatory through Azure Policy at resource group level, and base all your reporting on amortised cost, never on raw billed cost.

The case of shared costs: AKS, networking and observability

Shared costs are the sticking point in 90 % of allocation initiatives: a pooled AKS cluster, an ExpressRoute gateway, a Log Analytics workspace or an Application Gateway carry real costs that no resource tag can break down. Two strategies coexist. The first, pragmatic one, consists in isolating physically: one resource group — or even one subscription — per product, at the cost of losing some pooling. The second consists in splitting costs pro rata to a measured usage: on AKS, requests/limits metrics per namespace (via OpenCost or Kubernetes cost allocation solutions) provide a defensible allocation key; on Log Analytics, the `Usage` table gives the ingested volume by data type and by source. What matters is to document the key, to have it collectively accepted and to stop renegotiating it every month: an imperfect but stable key is worth more than an exact but contested one.

SECTION 04

Reservations, Savings Plans and Spot

On a stable compute estate, committing to nothing structurally costs 30 to 60 % more than the price available: on Azure, on-demand is the most expensive rate in the catalogue.

MECHANISM	INDICATIVE DISCOUNT	HOW IT WORKS	BEST SUITED TO
Reserved Instances	Up to ~72 % over 3 years	Commitment on a VM family in a given region, 1 or 3 years; instance flexibility within the series, exchange possible under certain conditions.	Stable, architecturally frozen base: databases, production application VMs, legacy workloads unlikely to change family.
Savings Plan for compute	Up to ~65 % over 3 years	Commitment on an hourly dollar amount, applied automatically to eligible compute resources, across all families and regions (VM, App Service, Container Instances, Functions Premium).	Compute estates in motion: modernisation under way, multi-region, IaaS/PaaS mix, frequent family trade-offs.
Spot virtual machines	Up to ~90 %	Azure's surplus capacity, reclaimable by the platform with 30 seconds' notice, with eviction policy by price or by capacity.	Batch, CI/CD, rendering, model training, interruptible AKS node pools — anything stateless, resumable and without an availability SLA.

Common pitfall: Buying reservations before doing the rightsizing amounts to locking in your waste for three years. Rightsize first, stabilise for two to three months, commit afterwards — and never commit on the basis of a seasonal peak or a temporary pre-production environment.

Sizing the commitment: think in coverage, not in volume

The right question is not "how many reservations should I buy" but "what percentage of my hourly consumption is genuinely incompressible". Plot the hourly curve of your compute consumption over 90 days: the flat base at the bottom — the one you consume 24/7, 365 days a year — is the committable perimeter. Common practice is to cover 70 to 85 % of that base, deliberately leaving an uncommitted margin to absorb migrations, decommissionings and architecture changes. Then set the commitments to "shared" scope so that they apply automatically across all subscriptions of the billing agreement, and track two distinct monthly KPIs: the coverage rate (share of consumption benefiting from a discount) and the commitment utilisation rate (share of the commitment actually consumed). A utilisation rate below 98 % signals a badly sized commitment, or a resource that was shut down without the commitment being reallocated.

SECTION 05

Rightsizing, Elasticity and PaaS

By default, Azure Advisor surfaces no memory-based recommendation: without a collection agent configured, rightsizing relies on CPU alone and misses the essentials.

Rightsizing driven by metrics, not by intuition

Reason on the P95 over 30 days, not on the average: a VM whose P95 CPU peaks at 15 % and memory at 40 % is oversized by two notches. Enable memory collection through Azure Monitor Agent before any campaign.

Change family before changing size

Recent generations (Dv5/Ev5, or Dpsv5/Epsv5 on Arm Ampere) offer a better performance/price ratio than the Dv3/Ev3 still massively deployed. Changing family is often more profitable than dropping one size notch.

Elasticity: the cheapest resource is the one that is switched off

A dev/test environment shut down at night and at weekends only runs 40 to 50 hours out of 168. Automate shutdown (Start/Stop VMs, Automation, schedule tags) and enable AKS autoscale with a Spot node pool for interruptible workloads.

LEVER	EXPECTED EFFECT
Shutdown outside business hours (dev/test)	Up to ~70 % fewer compute hours on non-production environments, with no impact whatsoever on production.
Blob storage lifecycle management	Automatic tiering Hot → Cool → Cold → Archive based on age: the cost per GB is divided by a significant factor on cold data.

SECTION 06

FinOps Governance and Culture

The FinOps Framework structures the approach into three iterative phases — Inform, Optimize, Operate — and the most frequent pitfall consists in trying to optimise before having informed.

The monthly ritual and the accountability chain

A 45-minute monthly review bringing together product owners, the platform team and management control is enough, provided it relies on a stable KPI set: amortised cost per product, unit cost (per transaction, per active customer, per environment), coverage rate and commitment utilisation rate, and the volume of identified waste left untreated. Showback — showing each team what it consumes — always precedes chargeback: recharging before trust in the figures has been built guarantees months of dispute over the allocation method rather than optimisation actions. Every significant variance must translate into a named ticket in a backlog, with an owner and a deadline, otherwise the review becomes an exercise in observation.

Automated guardrails and cost in the pull request

Durable governance rests on automated guardrails rather than on human vigilance. Three mechanisms cover the essentials: Azure budgets with alerts at 80 % and 100 % of forecast, routed to the owning team and not to a generic mailbox; Azure Policies restricting the SKUs, regions and VM sizes authorised per environment, which structurally prevent the deployment of a GPU in a staging environment; and Cost Management anomaly detection, which flags trend breaks before the end of the month. To this is added shifting left: estimating the cost of an infrastructure change directly in the pull request, via an estimation tool on the Terraform code, turns the financial trade-off into an engineering decision taken at the right moment — before deployment, not thirty days after the invoice.

Key takeaway: Appoint a cost owner per product, not per infrastructure team: as long as cost does not appear in the backlog review of the product that generates it, it will remain someone else's problem.

A FinOps initiative does not produce a one-off gain but a trajectory: the first months release the obvious savings — orphaned resources, forgotten environments, gross oversizing —, the following ones put the commitments in place, and it is only from the third or fourth cycle onwards that unit cost becomes an indicator steered in the same way as latency or error rate. What distinguishes the organisations that hold the trajectory from those that fall back after six months rarely comes down to tooling: it is the regularity of the ritual and the clarity of accountability. Measure, arbitrate, commit, then start again.

SECTION 07

Self-assessment grid

Review these six questions with your platform and finance teams before committing a single euro to reservations or savings plans.

- ☐ **Do you have a formalised tag taxonomy (5 to 6 keys maximum) enforced by Azure Policy at resource group level?** Without reliable tags enforced at creation time, no cost allocation by application or by team is usable.
- ☐ **Do your cost reports rely on amortised cost rather than raw billed cost?** Billed cost concentrates reservation purchases in a single month and distorts any trend comparison.
- ☐ **Have you configured memory metric collection to feed rightsizing recommendations?** Without a collection agent, Azure Advisor only reasons on CPU and misses the majority of real oversizing.
- ☐ **Have you stabilised your sizing for two to three months before any reservation or savings plan purchase?** Committing on an unoptimised estate means contractually locking in your waste for one to three years.
- ☐ **Do you track your coverage rate and the utilisation rate of existing commitments on a monthly basis?** An under-used reservation costs more than the pay-as-you-go it was supposed to replace.
- ☐ **Is there a recurring ritual (monthly or per sprint) where costs are arbitrated with a named owner per application?** The regularity of the ritual and the clarity of accountability are what separate initiatives that last from those that run out of steam.

How to read your answers: Three "no" answers or more indicate that you need to consolidate visibility and rightsizing before any commitment strategy.

SECTION 08

A 5-step roadmap

This trajectory unfolds without interrupting production: each step delivers a measurable gain and creates the conditions for the next, rather than imposing a wholesale transformation.

- 1 Establish visibility**
Deploy the tag taxonomy through Azure Policy, rebuild your reporting on amortised cost and establish a clear split between directly attributable costs and shared costs. Objective: no significant line of the bill should remain orphaned.
- 2 Eliminate obvious waste**
Delete unattached disks, reserved public IP addresses with no usage, old snapshots and forgotten test environments. These immediate gains fund the rest of the initiative and build credibility with the teams.
- 3 Rightsize resources**
Enable memory metric collection, cross-reference it with CPU and IOPS, then resize progressively starting with non-production environments. Introduce scheduled shutdown and autoscaling wherever the workload allows.
- 4 Commit in increments**
After two to three months of stabilised consumption, cover the incompressible baseline with reservations, the variable share with savings plans, and shift interruption-tolerant workloads to Spot. Commit in increments rather than all at once.
- 5 Embed governance**
Set up a monthly arbitration ritual with a named owner per application, budget alerts per environment and tracking of unit cost relative to business usage. Reassess coverage and utilisation at every cycle.

CONCLUSION

Unit cost, an engineering metric like any other

Organisations that treat cloud cost as an engineering metric — measured, allocated, arbitrated continuously — gain far more than a one-off reduction in their bill: they gain the ability to correlate an architecture decision with its economic effect, and therefore to make informed trade-offs between performance, resilience and spend.

The difficulty of FinOps on Azure is not technical: visibility tooling, commitment levers and elasticity mechanisms exist and are documented. It is organisational. As long as the bill remains an end-of-month topic handled by the finance department without the teams that provision the resources, the same drifts will reappear. Moving the spending decision as close as possible to the design decision, with reliable data and a short ritual, is the investment that pays off most durably.

Move2Cloud is a consulting firm specialised in infrastructure, cloud and IT transformation. We support organisations in the audit, financial optimisation and governance of their cloud environment.

Contact Move2Cloud — Paris · move2cloud.fr

Next step A FinOps audit of a few days — review of the tag taxonomy, analysis of amortised cost over twelve months, inventory of orphaned resources and commitment coverage diagnostic — is generally enough to turn the "no" answers in the checklist above into a costed action plan.

Sources

FinOps Foundation — FinOps Framework and State of FinOps (annual edition) — Microsoft — Azure Well-Architected Framework, Cost Optimization pillar — Microsoft Learn — Microsoft Cost Management documentation, Azure reservations and savings plans — Microsoft Learn — Cloud Adoption Framework for Azure, governance and naming and tagging conventions — Flexera — State of the Cloud Report (annual edition)

Document written by Move2Cloud as an example and illustration, for educational purposes. Azure pricing mechanisms, machine families and commitment terms change regularly: check the terms in force before any purchase.