

Dominik Heinrich, multi-cloud DevOps & AI solution architect

Cut your cloud bill without breaking anything

The FinOps checklist I use on Azure, AWS and Kubernetes platforms. Ten steps, in the order that saves the most with the least risk.

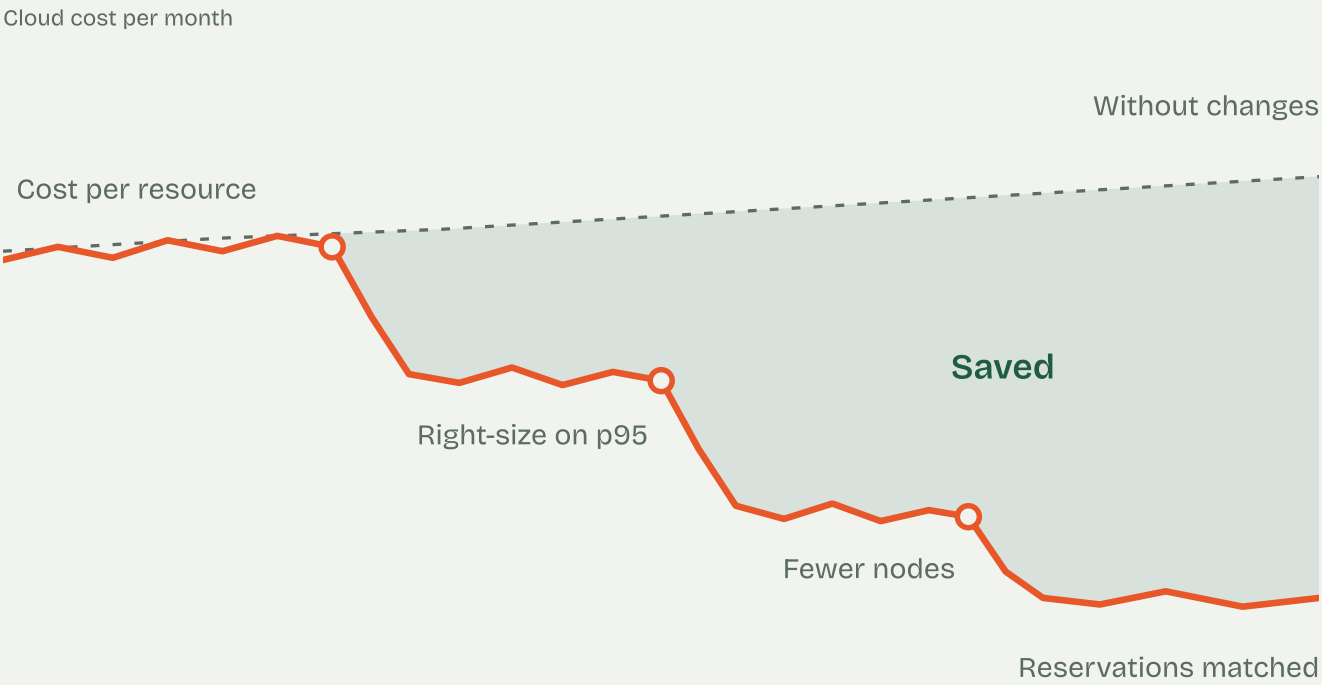


Illustration: how a cloud bill typically comes down in a FinOps engagement

From a recent FinOps engagement, anonymized

−48%	**7 → 4**	**~40×**	**~35**
Azure run-rate in five weeks	Kubernetes nodes	less Cosmos DB load	changes in one month, one ticket each

The ten steps

Work through them in order and tick them off. Each one names what to check before anything changes.

01 See where the money goes

- [ ] Pull amortized cost per resource and meter from the billing API, not just the monthly invoice.
- [ ] Build a forward run-rate so you see the effect of a change within days, not at month end.
- [ ] Tag every resource with product, team and stage. Untagged spend gets an owner within a week.
- [ ] Send a short daily cost report to the team chat. What gets seen gets fixed.

02 Check before you cut

- [ ] **For every candidate:** what runs on it, who uses it, and the peak load (p95), never the average.
- [ ] Anything a customer could notice is out of scope until proven safe.
- [ ] One change per ticket, each with impact, risk and the way back.

03 Right-size compute

- [ ] Set Kubernetes requests and limits from real p95 usage, then let the node count follow.
- [ ] Autoscale (cluster autoscaler or Karpenter) and use spot capacity for jobs that can be interrupted.
- [ ] **In one engagement:** a production cluster went from 7 to 4 nodes with no customer impact.

04 Downsize and delete

- [ ] Idle VMs, orphaned disks and volumes, unattached IPs, old snapshots and forgotten test resources.
- [ ] Database standbys and high-availability tiers only where the business really needs them.

05 Write a data retention policy

- [ ] **Logs:** debug off in production, retention per log class, cheap storage tiers for audit logs.
- [ ] **Backups:** retention matched to your recovery goals and legal duties, not to defaults.
- [ ] **Object storage:** lifecycle rules that move old data to cold tiers and delete what nobody needs.

06 Fix the expensive queries

- [ ] Find the top queries by cost, not by count. One fixed query cut the load on a Cosmos DB database about 40×.
- [ ] Move small tenant databases into the cluster (for example with CloudNativePG) when it pays off, with row counts and checksums checked.

07 Commit last, not first

- [ ] Buy reservations or savings plans after right-sizing, so you don't commit to machines you are about to remove.
- [ ] Treat commitments as hard to undo: Azure savings plans can't be cancelled, AWS Savings Plans can only be returned within seven days, and Azure reservation refunds are capped. When the footprint changes, use the exchange options that exist.

08 Turn environments off

- [ ] Dev and test off at night and on weekends by default, rebuilt automatically in the morning.
- [ ] Preview environments with a time limit, and nightly sweeps for orphaned resources.

09 Watch the network

- [ ] Alert on cross-zone and egress traffic, and review NAT gateways and load balancers nobody owns.

10 Keep it down

- [ ] Budgets with forecast alerts per team, and a weekly one-page report with a two-month forecast for management.
- [ ] A named owner per cost center, and the checklist rerun every quarter.

Mistakes I see most often

- Buying reservations first and right-sizing later, then paying for machines you no longer run.
- Sizing on averages: the cluster looks idle until the Monday peak.
- Deleting before checking who uses it. One forgotten consumer costs more than the saving.
- Turning logs off instead of tiering them, and losing the audit trail.
- One big clean-up weekend instead of one ticket per change, so nobody knows what broke what.

Ways to work together

The same quality for a ten-person startup and a global group. The format adapts to you.

Startups

Fractional DevOps & AI lead

A few days a month: architecture, the release pipeline and AI features, so the founders can focus on customers. From prototype to production without a rewrite.

Mid-sized companies

A project with a clear outcome

A defined goal like a lower cloud bill, a migration or the first AI agents in production, with a fixed scope, weekly progress and a handover your team can run.

Enterprises

An architect inside your teams

Embedded with your platform, security and product teams: architecture decisions, governance for AI and regulated environments. For larger scopes I bring a trusted senior team.

How I work

Decisions written down: architecture decision records with the trade-offs, so your team knows why, not just what.

Secure by default: OIDC and managed identities, least privilege, secrets in a vault, an audit trail.

Operated, not just delivered: SLOs, alerts on business outcomes and an incident routine.

Measured: cost, lead time, uptime and hours saved, reported every week.

Knowledge stays with you: runbooks, docs and workshops for your team.

One point of contact who talks to engineering and to management, in German, English and Spanish.



Want me to run this on your bill?

Book a 30-minute call

E-mail mail@dominikheinrich.ch
Web dominikheinrich.ch

Available worldwide, remote. On-site on request, depending on the country.