



CHECKLIST

Azure Virtualization Cost Optimization Checklist

A practical checklist to help teams reduce Azure virtualization spend without sacrificing performance, resilience, or security.

Azure Virtualization Cost Optimization Checklist

Use this checklist to identify waste, validate savings, and avoid introducing performance, resilience, or security gaps.

1) Define the workload and its purpose

- ☐ Identify the workload owner and business purpose.
- ☐ Confirm whether the workload is production, pre-production, or transient.
- ☐ Document service expectations, including uptime, performance, and recovery needs.
- ☐ Note any compliance, security, or data-handling requirements.

2) Collect enough usage data

- ☐ Review at least 274 weeks of utilization data.
- ☐ Capture CPU, memory, disk IOPS, disk latency, and network throughput.
- ☐ Include backup, snapshot, and retention consumption.
- ☐ Check for daily, weekly, monthly, and quarter-end usage spikes.
- ☐ Avoid making permanent changes based on a quiet period.



3) Identify obvious waste

- ☐ Find idle or nearly idle virtual machines.
- ☐ Flag oversized instances with consistently low utilization.
- ☐ Detect duplicate environments used for convenience rather than need.
- ☐ Review always-on non-production systems.
- ☐ Identify abandoned staging, test, or build resources.
- ☐ Locate unattached or underused disks, snapshots, and images.

4) Rightsize compute safely

- ☐ Compare average utilization and peak behavior, not just one metric.
- ☐ Confirm whether CPU or memory is the real constraint.
- ☐ Check for paging, cache churn, or memory pressure.
- ☐ Keep enough headroom for expected peaks and scheduled jobs.
- ☐ Resize incrementally rather than making large reductions at once.
- ☐ Validate application response time after any change.

5) Schedule non-production workloads

- ☐ Identify dev, test, training, demo, and lab environments that do not need 24/7 uptime.
- ☐ Define business-hour operating windows.
- ☐ Automate shutdown and startup where appropriate.
- ☐ Confirm owners know when the environment will stop and restart.
- ☐ Ensure dependent services are started in the correct order.
- ☐ Verify scheduling does not interfere with patching or testing windows.



6) Review storage choices

- ☐ Match storage performance to actual workload demand.
- ☐ Remove high-performance storage from low-utilization workloads when safe.
- ☐ Check whether disks are larger than the data set requires.
- ☐ Review retention policies for snapshots, backups, and restore points.
- ☐ Delete outdated or unnecessary images and copies.
- ☐ Align storage tier selection with recovery objectives.

7) Validate backup and recovery requirements

- ☐ Confirm the recovery point objective (RPO) and recovery time objective (RTO).
- ☐ Check that backup retention still matches business needs.
- ☐ Test restore procedures after major storage or retention changes.
- ☐ Confirm backup success, restore access, and recovery time after optimization.
- ☐ Avoid reducing protection simply because a resource is underused.

8) Review network-related cost drivers

- ☐ Check for unnecessary cross-region traffic.
- ☐ Look for excessive cross-zone communication.
- ☐ Confirm that workloads are placed close to their dependencies.
- ☐ Review peering, routing, and communication paths for inefficiency.
- ☐ Remove network paths that exist only due to legacy design.
- ☐ Recheck outbound traffic charges after topology changes.

9) Preserve security posture



- ☐ Confirm that cost changes do not weaken access control.
- ☐ Keep required segmentation, monitoring, and hardening in place.
- ☐ Do not remove controls that protect sensitive workloads.
- ☐ Review whether lower-cost settings still meet security standards.
- ☐ Validate any change that affects trust boundaries.

10) Choose the least expensive configuration that still meets requirements

- ☐ Compare the current setup with lower-cost alternatives.
- ☐ Consider consolidation of idle environments.
- ☐ Evaluate smaller instance sizes, fewer disks, or cheaper storage tiers.
- ☐ Keep workload criticality in view when selecting changes.
- ☐ Prefer sustainable savings over short-term reductions that create rework.

11) Test before and after the change

- ☐ Record baseline performance before optimization.
- ☐ Make one change at a time when possible.
- ☐ Test workload behavior under expected load.
- ☐ Check response times, error rates, and saturation after the change.
- ☐ Verify failover, backup, and restore assumptions.
- ☐ Keep a rollback plan ready.

12) Confirm savings are real

- ☐ Compare the post-change bill or consumption report to the baseline.
- ☐ Confirm the workload still meets operational targets.



- ☐ Review the next demand cycle for regressions.
- ☐ Reverse changes that introduce instability or hidden cost.
- ☐ Document what changed, why it changed, and who approved it.

Quick decision rule

- ☐ Optimize only when you can answer all three questions:
- ☐ Does the workload have enough measured headroom?
- ☐ Will the change preserve recovery and security requirements?
- ☐ Can the team verify the result after implementation?

Suggested review cadence

- ☐ Reassess production workloads monthly or after major demand changes.
- ☐ Reassess non-production schedules weekly or biweekly.
- ☐ Revisit storage and backup settings quarterly.
- ☐ Review network paths and dependency placement after architectural changes.
- ☐ Keep optimization as an ongoing control loop, not a one-time project.

Final sign-off

- ☐ Workload purpose documented
- ☐ Baseline data collected
- ☐ Waste identified
- ☐ Change tested
- ☐ Security reviewed
- ☐ Recovery validated
- ☐ Savings confirmed



- ☐ Owner approval recorded