

Optimizing cloud infrastructure costs while maintaining performance is a persistent challenge in the cloud-first era. Virtual Machines (VMs) often account for a substantial portion of cloud spend, and incorrectly sized instances can silently incur waste. Azure Advisor offers VM resize recommendations that can help, but effective use requires understanding workload profiles, cloud platform nuances, and careful measurement methods. In this post, I'll share engineering practices that worked for me when leveraging Azure Advisor in tandem with learnings from tools like AWS Compute Optimizer.

Why VM Resizing Matters: The Hidden Cost of Always-On Small Services

Before diving into how to use Azure Advisor, it's critical to recognize the root cause of much cloud waste—"always-on" small services. These could be lightweight internal tooling, maintenance daemons, or monitoring agents running 24x7 across your fleet. Individually, their cost is low, but collectively these background workloads balloon your cloud bill over months and years.



What often goes unnoticed:



- These services have steady CPU usage but often have brief bursts of high utilization.
- VMs hosting such services may be oversized based on peak usage seen over short time frames or misinterpreted averages.
- Outbound networking egress tied to these VMs can also contribute unexpectedly to costs.

The right approach is not to blindly downsize, but to evaluate the *right* observation windows and compute percentile metrics to capture workload variability.

Azure Advisor Versus AWS Compute Optimizer: Different Ecosystems, Similar Goals

Both Azure Advisor and AWS Compute Optimizer provide recommendations to resize VMs or instances based on usage data. But their methodologies and definitions differ, so it's key to understand nuances before making decisions.

Feature	Azure Advisor	AWS Compute Optimizer
Primary data sources	Azure Monitor metrics (CPU, Memory, Disk I/O)	CloudWatch metrics (CPU, Memory, Disk I/O)
Observation windows	7-day to 30-day periods configurable	14-day by default
Use of percentiles	Focus on average and peak usage but lacks transparent percentile-based control	Offers P95 CPU metrics and conservative recommendations
Handling of burstable/shared CPU instances	Requires manual assessment; no automatic spike duration accounting	Explicitly considers burst balances
Outbound networking considerations	Does not explicitly factor egress costs into recommendations	Also does not integrate networking costs but provides usage metrics

While AWS Compute Optimizer is slightly more transparent on percentile options and burst CPU, Azure Advisor remains valuable for its integration with Azure Monitor and recommendations across a broader set of Azure resources.

Shared CPUs Are Not Created Equal: Why vCPU Count Is Misleading Without Context

A common pitfall in VM resizing is treating vCPU counts as strict performance guarantees. Not all vCPUs are equal:

- **Shared or Burstable CPUs:** Instances like Azure B-series or AWS T-series share physical cores and deliver performance credits for bursts. The real-world available CPU fluctuates.
- **Dedicated CPUs:** Offer more predictable performance, especially under sustained loads.

Assuming a VM with 2 shared vCPUs performs the same as one with 2 dedicated vCPUs leads to underperformance or overprovisioning.

Thus, resizing decisions must include:

1. Understanding the SKU's CPU sharing model and burst characteristics.
2. Measuring peak usage with an adequate sample window rather than relying on averages.
3. Considering the duration and frequency of spikes when evaluating if the workload fits within a smaller VM.

Measure Peaks with the Right Observation Window and Percentiles

One reason resizing decisions fail is basing them on average CPU utilization. Average CPU masks spikes that can cause latency, failures, or degraded user experience. Instead, focus on two critical observables:

- **Percentile metrics:** Use P90, P95, or P99 CPU and memory metrics. These metrics tell you the CPU usage level that your workload exceeds only 5-10% of the time.
- **Spike duration:** A 5-minute CPU spike rarely impacts overall performance as much as a sustained 30-minute spike.

Azure Monitor metrics support rich aggregation including percentile-based insights when configured correctly, but many users default to averages.

Here's a practical example of collecting CPU usage percentiles using Azure Monitor:

```
metrics | where ResourceType == "virtualMachines" | summarize percentiles(CPUPercentage, 90, 95) by
bin(TimeGenerated, 1h), ResourceId | order by TimeGenerated desc
```

Implementing this approach allows you to make resizing decisions that preserve headroom for workload spikes without overprovisioning.

Accounting for Outbound Networking in VM Resizing Decisions

It's easy to forget outbound networking egress costs when resizing VMs, but these can represent a non-trivial fraction of your spend. Larger VM sizes might have higher network throughput limits, tempting teams to upsize just for bandwidth. However, egress costs are volume-based, not tied directly to VM size.

Azure Advisor does not explicitly provide recommendations around reducing outbound networking costs, so you'll need to supplement VM resizing decisions with data from Azure Network Watcher or Azure Monitor metrics related to <https://computingforgeeks.com/shared-cpu-cloud-waste-migration-guide/> network egress:

- Identify VMs with high outbound traffic and analyze whether downsizing affects application performance.
- Consider architectural changes like caching or content delivery networks (CDNs) to reduce egress volume.

Combining VM resize recommendations with network egress metrics prevents surprises on the monthly bill.

Step-by-Step: Using Azure Advisor for VM Resize Recommendations with Best Practices

1. Gather Workload Metrics with Adequate Observation Windows

- Configure Azure Monitor to capture VM CPU, memory, disk, and network metrics over 14-30 days.
- Use percentile queries (P90, P95, P99) to understand peak rather than average usage.

2. Review Azure Advisor Recommendations

- In the Azure Portal, open Azure Advisor and filter “Cost” recommendations for your virtual machines.
- Check recommended sizes and their expected savings.

3. Map Recommendations Against Workload Profiles

- Compare recommended VM sizes with observed peak CPU and memory percentiles.
- Factor in whether the VM SKU is shared CPU or dedicated CPU.
- Note the frequency and duration of CPU spikes to assess if a smaller VM can handle workload bursts.

4. Incorporate Network Egress Data

- Use Azure Network Watcher or Monitor to assess outbound egress volume.
- Understand if resizing could impact network throughput or cost.

5. Run Pilot or Canary Resize Trials with Rollback Criteria

- Resize a small set of VMs following agreed criteria (e.g., peak P95 CPU $\leq$ 60% of the new instance capacity).
- Define clear rollback criteria based on latency, error rates, and resource metrics.

6. Iterate and Automate

- Use scripts or tooling to regularly pull recommendations and metrics.
- Set thresholds and automated alerts to identify when workloads grow beyond resized capacity.

Summary and Final Recommendations

Using Azure Advisor for VM resize recommendations is a powerful place to start cost optimization but requires supplementing with careful analysis and engineering judgment:

- Always collect and review percentile-based CPU, memory, and spike duration metrics over at least two weeks.
- Recognize differences in CPU sharing models and avoid equating vCPU counts blindly.
- Don't rely solely on averages or utilization percentages—check P95/P99, spike lengths, and the workload's tolerance to latency.
- Incorporate outbound networking egress analysis to avoid hidden costs that resize alone won't solve.
- Run controlled pilots with rollback plans to validate assumptions before wide rollout.

By combining Azure Advisor with customized percentiles and network metrics, you can confidently reduce VM footprint and cloud spend while maintaining performance and stability.

Additional Resources

- [Azure Advisor Cost Recommendations Documentation](#)
- [Azure Monitor Overview](#)
- [AWS Compute Optimizer](#)

- Azure Network Watcher