Optimizing Azure Infrastructure: Cloud Monitoring & Cost Reduction with LogicMonitor

Welcome to this comprehensive presentation on cloud infrastructure optimization using Azure and LogicMonitor. We'll explore how this integrated solution enables proactive monitoring, performance analysis, and cost optimization for enterprise cloud environments.

This project demonstrates how proper monitoring and analysis can lead to significant cost reductions while maintaining or improving performance - a critical balance in today's cloud-first enterprise environments.

AG **by Arash Gooran**

Project Overview & Objectives

1 Cloud Infrastructure Monitoring

Configure LogicMonitor and Azure Monitor to track VMs, storage, networking components, and critical KPIs. Implement automated alerting to detect performance issues in real-time before they impact operations.

2 VM Performance Analysis

Analyze CPU utilization, memory consumption, and I/O metrics to identify performance bottlenecks. Determine optimal resource allocation by identifying over-provisioned and under-provisioned workloads.

3 Cost Optimization

Recommend VM right-sizing strategies based on actual utilization data. Create cost analysis reports comparing current and projected expenses. Implement and verify approved changes to demonstrate ROI.

Technology Stack Overview

Azure Monitor

Native Azure monitoring solution providing infrastructure-level metrics and logs. Offers built-in dashboards and visualization tools for tracking cloud resource performance across the entire Azure estate.

LogicMonitor

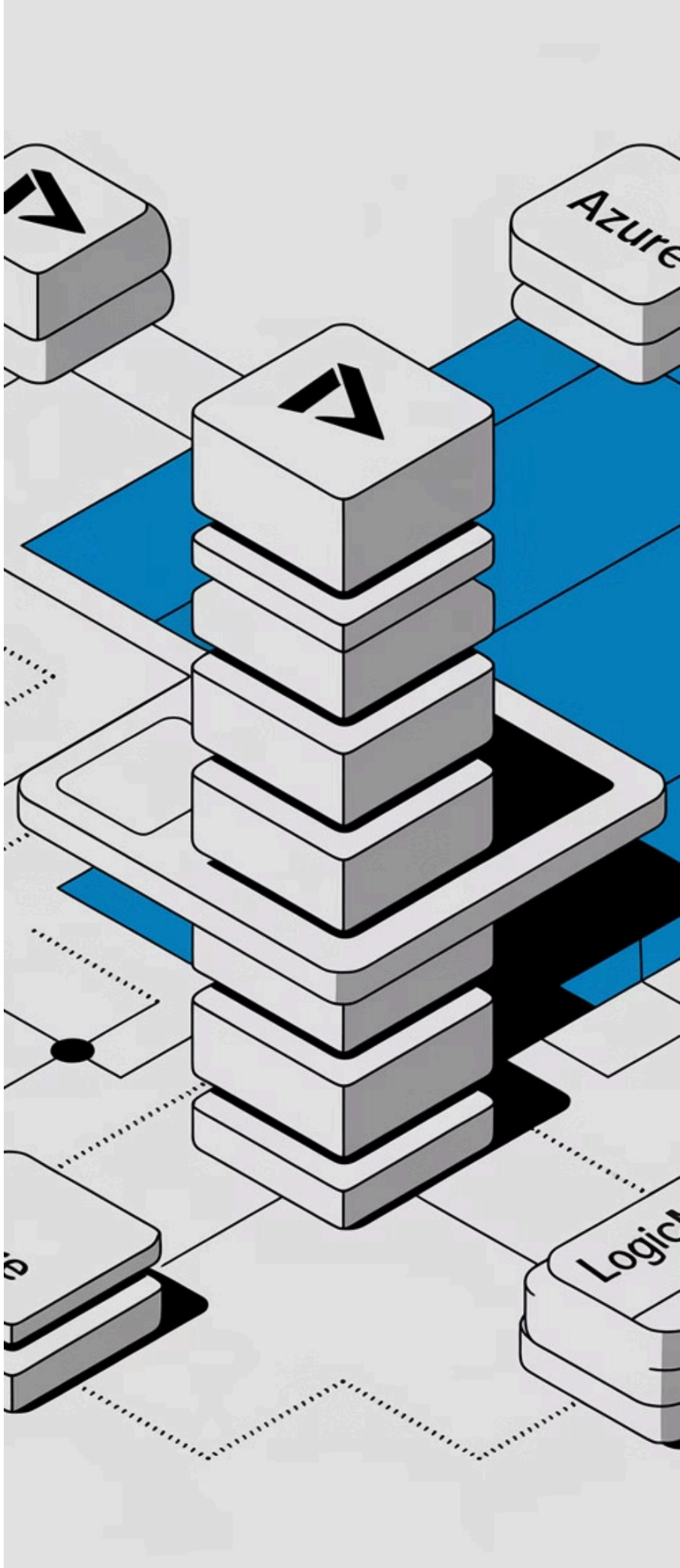
Enterprise-grade monitoring platform offering advanced real-time alerting and monitoring. Provides unified visibility across hybrid cloud environments beyond just Azure resources, with extensive customization options.

Azure Log Analytics

Centralized log management service for collecting, querying, and analyzing telemetry data. Enables custom Kusto queries to extract actionable insights from performance metrics and resource logs.

Azure Cost Management

Built-in cost tracking tool for analyzing historical spend, forecasting future expenses, and implementing budget controls. Integrates with Azure Advisor for specific optimization recommendations.



LogicMonitor Integration Setup

Access Cloud Integrations

Log into the LogicMonitor portal and navigate to the Settings section. Under Integrations, select "Cloud Integrations" to begin connecting with your Azure environment.

Connect Azure Subscription

Add Azure as a monitored environment by providing your Subscription ID and Tenant ID. Configure authentication using an Azure Service Principal with appropriate monitoring permissions.

Configure Resource Monitoring

Enable metrics collection for virtual machines, storage accounts, and networking components. Select the appropriate monitoring templates for each resource type to gather relevant metrics.

Set Up Alert Thresholds

Create custom alert rules based on performance thresholds. Configure notifications for different severity levels and define escalation paths for critical issues requiring immediate attention.

Critical Alert Configuration

Alert Type	Threshold	Notification	Priority
High CPU Utilization	>80% for 15 minutes	Email + SMS	Critical
Low Disk Space	<10GB free space	Email Alert	Warning
Memory Pressure	>90% utilization	Email + Ticket	Critical
VM Unavailable	No response >2 minutes	SMS + Ticket	Critical
Network Latency	>100ms response time	Email Alert	Warning

Azure Monitor Configuration

1

Create Log Analytics Workspace

Navigate to Azure Portal → Monitor → Log Analytics. Create a new workspace to serve as the central repository for collecting and analyzing telemetry data from your cloud resources.

2

Deploy Monitoring Agents

Connect virtual machines to the workspace by installing the Log Analytics agent. For Linux VMs, deploy the OMS Agent; for Windows VMs, use the Microsoft Monitoring Agent to collect detailed performance metrics.

3

Configure Data Collection Rules

Define which performance counters and logs should be collected. Focus on CPU, memory, disk I/O, and network metrics at appropriate collection intervals to balance data granularity with storage costs.

4

Create Custom Dashboards

Build Azure Monitor dashboards that visualize the most important metrics for your environment. Pin relevant charts, graphs, and alerts to provide at-a-glance visibility into system performance.

Key Performance Metrics

CPU Utilization

Tracks processor usage across all virtual cores. The primary indicator of computational load, with sustained high utilization (>80%) suggesting potential bottlenecks requiring VM upscaling or workload redistribution.

Memory Consumption

Monitors available and used RAM. Critical for applications with high memory requirements, with insufficient memory leading to paging, swapping, and significant performance degradation.

Disk I/O Operations

Measures read/write operations and latency. Particularly important for database workloads where disk performance directly impacts application responsiveness and user experience.

Network Throughput

Tracks bandwidth usage and packet flow. Essential for web applications, API services, and other network-intensive workloads that depend on consistent connectivity and low latency.

Kusto Query Language for Performance Analysis

CPU Utilization Query

```
Perf
| where ObjectName == "Processor"
  and CounterName == "% Processor Time"
| summarize AvgCPU = avg(CounterValue) by
Computer
| where AvgCPU > 80
| order by AvgCPU desc
```

This query identifies VMs with sustained high CPU utilization, suggesting potential candidates for upscaling or workload redistribution to prevent performance bottlenecks.

Memory Utilization Query

```
Perf
| where ObjectName == "Memory"
  and CounterName == "Available MBytes"
| summarize AvgMemory = avg(CounterValue) by
Computer
| join (
    Perf
    | where ObjectName == "Memory"
      and CounterName == "Total Memory"
    | summarize TotalMem = max(CounterValue) by
Computer
) on Computer
| project Computer, MemoryUtilizationPercent = (1
- (AvgMemory / TotalMem)) * 100
| order by MemoryUtilizationPercent desc
```

This more complex query calculates memory utilization as a percentage of total memory, helping identify memory-constrained systems requiring attention.

VM Performance Analysis Process

1 Data Collection Period

Gather performance metrics over 2-4 weeks to establish reliable baseline usage patterns. Ensure the collection period includes both peak business hours and typical low-usage periods for a complete utilization profile.

2 Performance Data Analysis

Analyze CPU, memory, and disk metrics to identify usage patterns. Calculate average, peak, and 95th percentile utilization to determine actual resource requirements versus provisioned capacity.

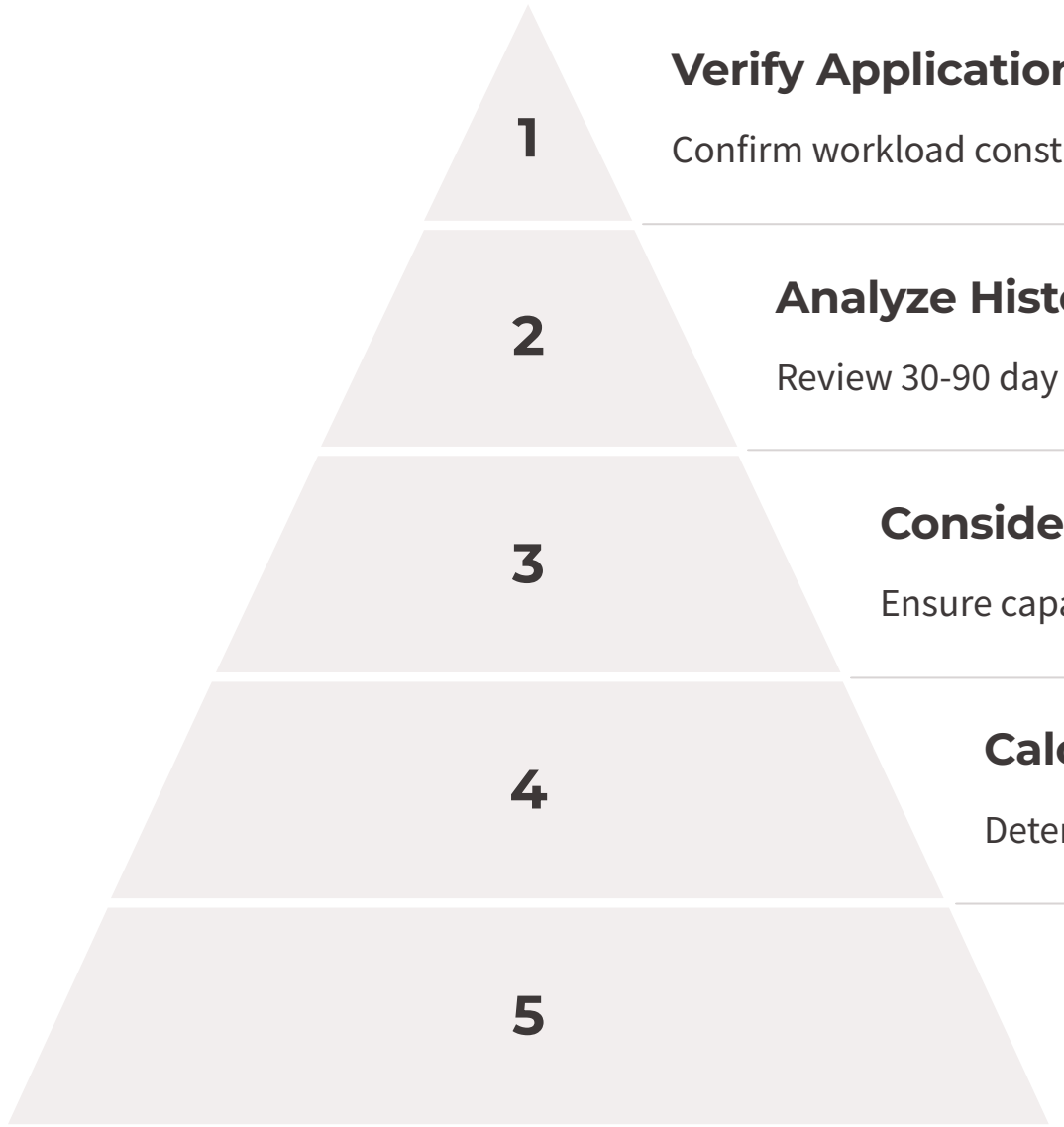
3 Identify Optimization Candidates

Flag VMs with consistent low utilization (under 20% CPU/memory) as downsizing candidates. Mark VMs with regular high utilization (over 80%) for potential upscaling to prevent performance issues.

4 Generate Size Recommendations

Compare current VM sizes against actual resource requirements. Create a recommendation matrix showing potential right-sizing options and associated cost implications for each virtual machine.

VM Right-Sizing Methodology



Verify Application Requirements

Confirm workload constraints and SLAs

Analyze Historical Utilization

Review 30-90 day performance trends

Consider Peak Usage Patterns

Ensure capacity for business-critical periods

Calculate Right-Size Requirements

Determine optimal VM series and size

Implement with Minimal Disruption

Schedule changes during maintenance windows

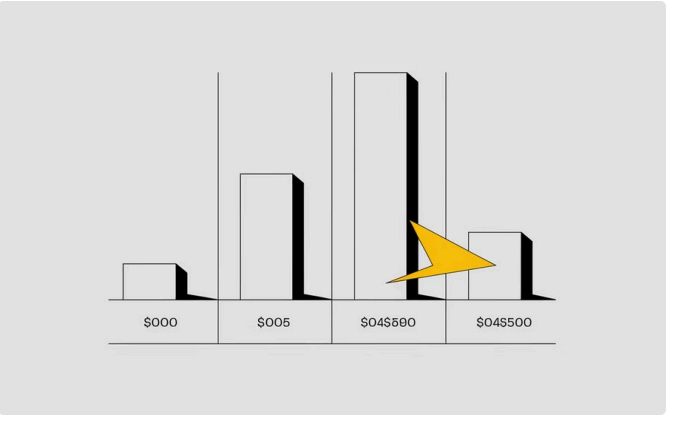
Effective VM right-sizing requires a methodical approach that balances performance needs with cost efficiency. Begin by understanding the true requirements of applications running on each VM, accounting for both typical operations and peak demand periods.

The analysis should cover multiple weeks of performance data to identify patterns, seasonality, and anomalies that might influence sizing decisions. When calculating the optimal size, build in a reasonable buffer (typically 20-30% above average utilization) to accommodate unexpected workload increases.

Cost Analysis Using Azure Pricing Calculator

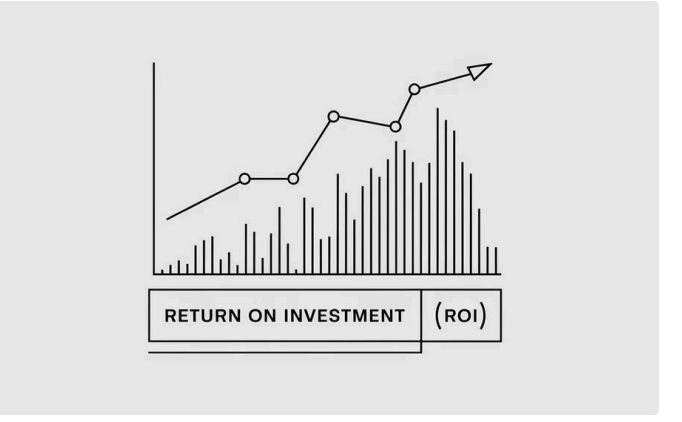
Pricing Calculator Interface

The Azure Pricing Calculator allows you to model different VM configurations and instantly see cost implications. Select your region, VM series, and specific instance size to generate accurate monthly cost estimates.



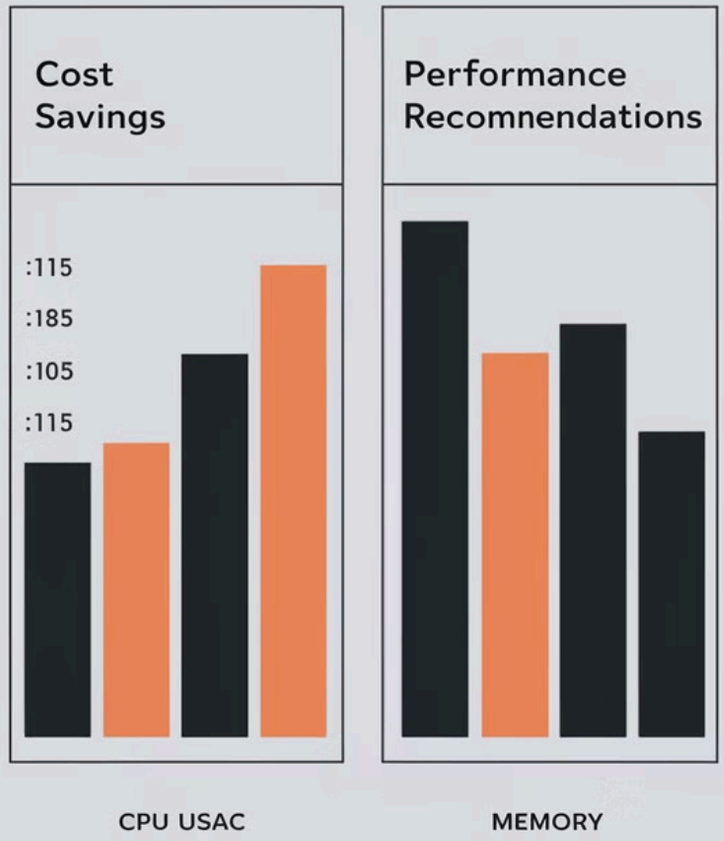
Comparative Cost Analysis

Create side-by-side comparisons between current and proposed VM configurations. Include options for reserved instances (1-year and 3-year commitments) to demonstrate additional potential savings through longer-term commitments.



Total Cost of Ownership Report

Generate comprehensive TCO reports that account for all cost factors including compute, storage, networking, and potential Azure Hybrid Benefit savings. These reports provide executive-friendly visualization of projected cost reductions.



Sample VM Resizing Recommendations

VM Name	Current Size	Utilization	Recommended Size	Monthly Savings
PROD-SQL01	D8s_v3	25% CPU, 40% RAM	D4s_v3	$175.20
PROD-WEB02	D4s_v3	85% CPU, 60% RAM	D8s_v3	-$175.20
DEV-APP01	D4s_v3	10% CPU, 15% RAM	B2ms	$219.00
TEST-SQL01	E8s_v3	5% CPU, 10% RAM	D2s_v3	$444.86
PROD-APP03	F8s_v2	30% CPU, 25% RAM	F4s_v2	$146.00

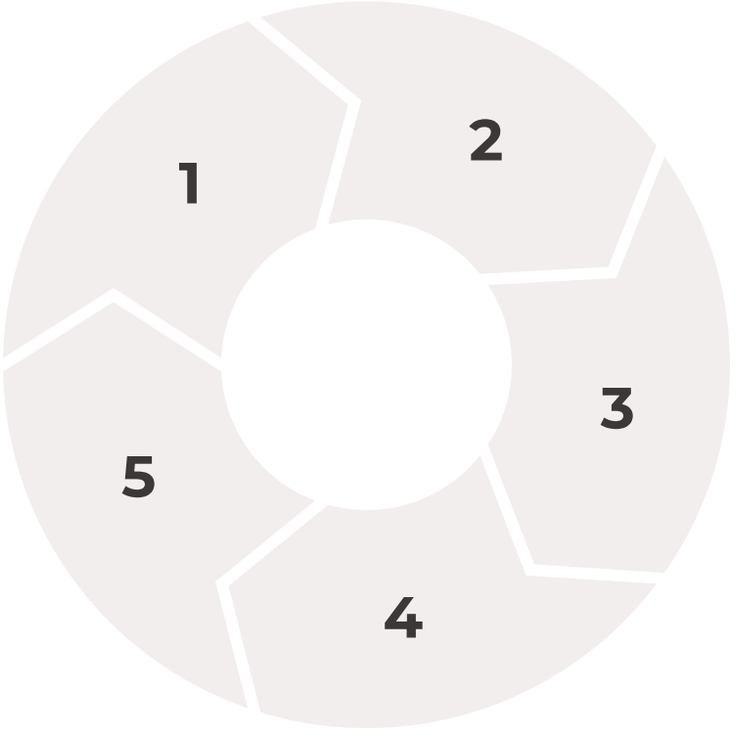
VM Resizing Implementation Process

Stakeholder Approval

Present findings and recommendations to technical stakeholders

Documentation Update

Record configuration changes in system documentation



Maintenance Window

Schedule resizing during approved low-traffic periods

Execute Resizing

Implement changes using PowerShell or Azure Portal

Validation Testing

Verify application performance after resizing

The VM resizing implementation follows a carefully planned circular process to ensure application stability and performance. Begin with thorough planning and securing necessary approvals from application owners and IT management to minimize business impact.

After implementing changes, conduct comprehensive validation to ensure applications continue to function correctly. Monitor performance closely for 24-48 hours following resizing to identify any unexpected behavior or performance issues that might require further adjustment.

PowerShell Automation for VM Resizing

VM Resizing Script Example

```
# Resize VM to a new SKU
$resourceGroup = "MyResourceGroup"
$vmName = "MyVM"
$newSize = "Standard_D2s_v3"

# Stop the VM before resizing
Stop-AzVM -ResourceGroupName $resourceGroup
-Name $vmName -Force

# Change the VM size
Set-AzVMSize -ResourceGroupName
$resourceGroup -VMName $vmName -Size
$newSize

# Start the VM after resizing
Start-AzVM -ResourceGroupName
$resourceGroup -Name $vmName
```

Batch Resizing Implementation

For environments with multiple VMs requiring resizing, develop parameterized scripts that can process a CSV file containing VM details, target sizes, and scheduled implementation times. This approach enables efficient batch processing of changes during maintenance windows.

Incorporate error handling and logging to capture any issues that arise during the resizing process. Include verification steps that check VM status after resizing and send notifications to the operations team about successful completions or any failures requiring intervention.

Consider implementing a rollback mechanism that can restore VMs to their original configuration if post-change validation indicates performance problems or application instability after resizing.

Post-Optimization Monitoring Strategy

Configure Baseline Alerts

Establish new baseline metrics reflecting expected performance after resizing. Adjust alert thresholds based on the new VM specifications to maintain appropriate sensitivity to performance anomalies without generating false alarms.

Implement Enhanced Monitoring

Temporarily increase monitoring frequency for resized VMs to capture more detailed performance data. Focus on application response times and resource utilization patterns during peak business hours to verify sizing adequacy.

Conduct Performance Comparison

Compare pre- and post-optimization performance metrics to quantify improvements. Document changes in resource utilization, application response times, and overall system stability as part of the project deliverables.

Generate Cost Impact Reports

Calculate actual cost savings achieved through resizing efforts. Compare projected savings from the planning phase with actual billing data to validate the financial benefits realized through the optimization initiative.

Azure B-Series VMs for Cost Optimization

Burstable Performance

B-series VMs provide a cost-effective solution for workloads with variable utilization patterns. They accumulate credits during low-usage periods that can be consumed during high-demand intervals, making them ideal for development, testing, and low-traffic web servers.

Credit Banking System

These VMs operate on a credit banking system where CPU credits accumulate when utilization falls below the baseline. These credits can then be "spent" to burst above baseline when needed, providing flexibility for workloads with intermittent processing requirements.

Significant Cost Savings

B-series VMs can deliver up to 70% cost savings compared to equivalent standard series VMs for appropriate workloads. They're particularly valuable for environments with predictable low-utilization periods, such as business applications used primarily during working hours.

Monitoring Requirements

Effective use of B-series VMs requires careful monitoring of credit accumulation and consumption patterns. LogicMonitor and Azure Monitor can be configured to track credit balance and alert when consistent high utilization might exceed available credits.

Azure Reserved Instances for Additional Savings

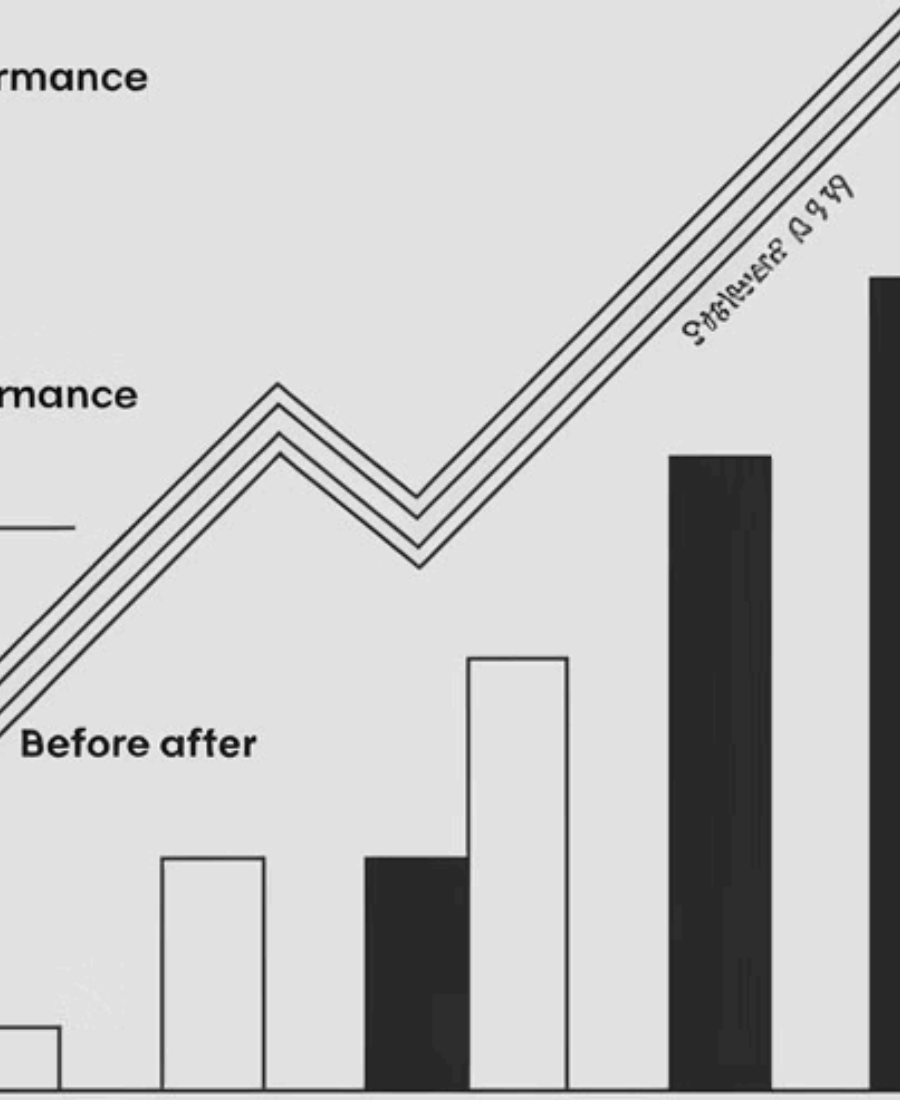
Reserved Instance Benefits

- Up to 72% price savings compared to pay-as-you-go pricing
- 1-year or 3-year commitment options for different budget cycles
- Flexibility to exchange or cancel reservations as needs change
- Available for VMs, databases, storage, and other Azure services

Implementation Strategy

For stable production workloads with predictable resource needs, implementing Reserved Instances (RIs) provides substantial cost advantages. After right-sizing VMs to match actual performance requirements, the next optimization step is converting appropriate instances to RIs.

Begin by identifying VMs that will remain in production for at least one year and run continuously. These make ideal candidates for 1-year reservations. For critical infrastructure with multi-year lifespan projections, 3-year reservations offer the maximum discount level.



Project Results: Performance Improvements

99.99%

VM Uptime

Improved system availability through proactive monitoring and alerting, eliminating surprise outages from resource constraints.

45%

Reduced Response Time

Application responsiveness improved significantly after addressing CPU and memory bottlenecks identified through performance analysis.

60%

Decreased CPU Throttling

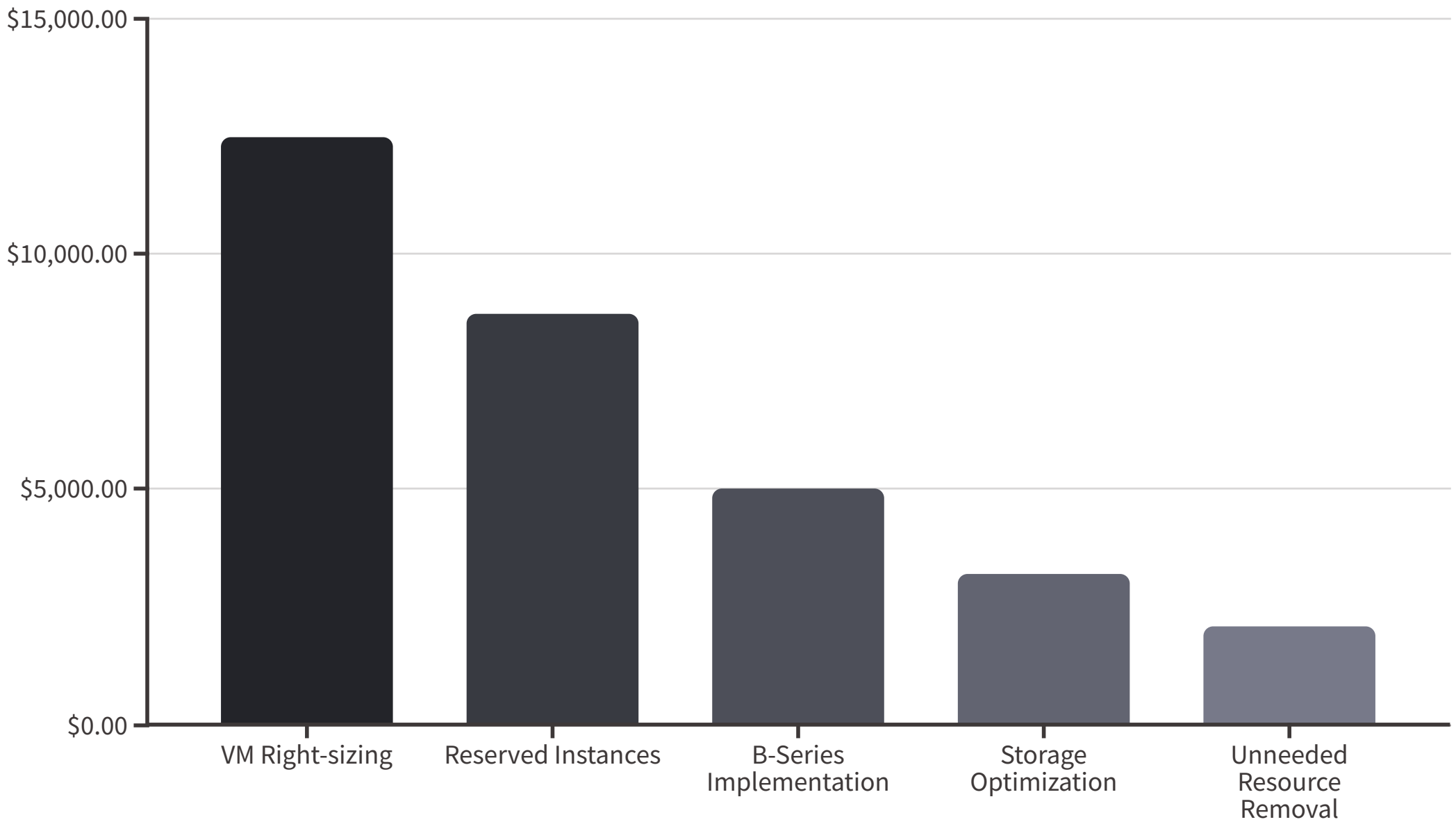
Properly sized VMs experienced fewer CPU throttling events, resulting in more consistent application performance.

3x

Faster Report Generation

Business intelligence reports completed three times faster after optimizing database server resources based on monitoring data.

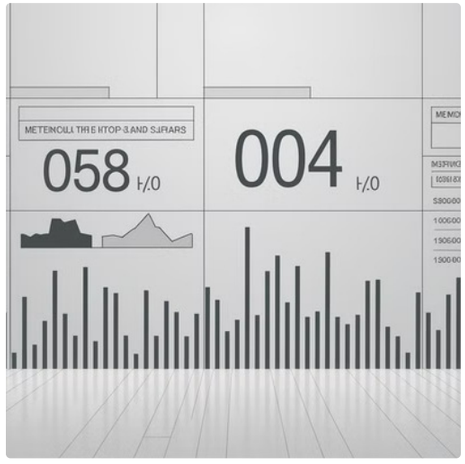
Project Results: Cost Optimization

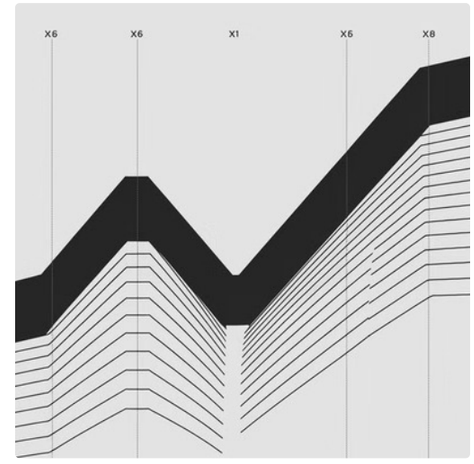


The optimization project delivered substantial cost savings across multiple categories. VM right-sizing produced the largest impact, accounting for nearly 40% of the monthly cost reduction. Implementation of Reserved Instances for stable workloads contributed significantly as well.

Migrating appropriate workloads to B-Series VMs provided additional savings, while storage optimization and cleanup of unused resources rounded out the comprehensive cost reduction strategy. Together, these initiatives reduced monthly Azure spending by approximately 30%.

LogicMonitor Dashboard Examples

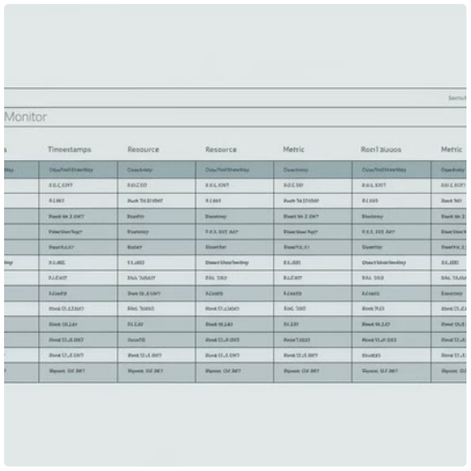


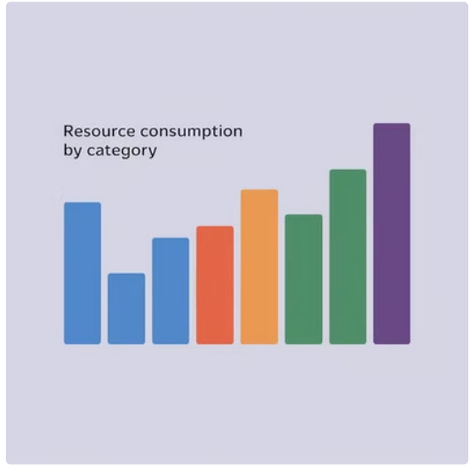


LogicMonitor provides comprehensive visualization capabilities that transform complex monitoring data into actionable insights. These sample dashboards illustrate the platform's ability to present performance metrics, alert configurations, resource health status, and historical trends in an easily digestible format.

Custom dashboards can be tailored to specific stakeholder needs, from technical operations teams requiring detailed metrics to executive leadership needing high-level cost and performance summaries.

Azure Monitor Dashboard Examples





Azure Monitor provides native visualization tools integrated directly into the Azure Portal experience. These dashboards offer immediate visibility into resource health, performance metrics, and cost analysis without requiring third-party tools.

Custom dashboards can combine widgets from multiple Azure services, creating comprehensive monitoring views that align with specific operational requirements and organizational structures.

Reusable Technical Documentation

1

Implementation Guides

Step-by-step configuration documentation

2

Performance Baselines

Documented normal operating parameters

3

Optimization Playbooks

Repeatable processes for ongoing optimization

4

Alerting Documentation

Threshold rationales and response procedures

Comprehensive documentation was a key deliverable from this project, ensuring knowledge transfer and enabling ongoing optimization efforts. The implementation guides provide detailed technical instructions for configuring LogicMonitor and Azure Monitor, including screenshots and step-by-step procedures for future reference.

Performance baseline documentation establishes normal operating parameters for all monitored systems, creating a reference point for detecting abnormal behavior. The optimization playbooks outline methodologies for periodic review and adjustment of cloud resources, ensuring the environment maintains optimal performance and cost efficiency over time.

Automation Script for Performance Reporting

Script Features

- Automated data collection from multiple Azure subscriptions
- Performance trend analysis for all monitored VMs
- Identification of optimization candidates based on configurable thresholds
- Cost impact calculations for recommended changes
- Formatted report generation in Excel and PDF formats
- Scheduled execution and email distribution to stakeholders

Implementation Benefits

This PowerShell-based automation solution transforms what was previously a manual, time-consuming analysis process into a streamlined, consistent reporting workflow. By automating data collection and analysis, the script eliminates human error and ensures all VMs are regularly evaluated against optimization criteria.

The script runs weekly to identify new optimization opportunities as usage patterns evolve. Reports are automatically distributed to IT management and infrastructure teams, maintaining visibility into ongoing optimization potential and helping capture cost savings opportunities as they emerge.

Critical Alert Automation Workflow

1

Performance Threshold Breach

System detects metrics exceeding defined thresholds, such as CPU utilization >90% for 15+ minutes or available memory dropping below 500MB, triggering the automated response workflow.

2

Alert Generation

LogicMonitor or Azure Monitor creates an alert with severity classification based on the impact level. Critical alerts are immediately escalated while warnings are grouped for periodic review.

3

Notification Distribution

Alerts are routed to appropriate channels including email, SMS, and ticketing systems based on severity. On-call rotation schedules ensure notifications reach available staff at all hours.

4

Automated First Response

For specific alert types, automated remediation scripts execute predetermined actions such as service restarts or cache clearing to resolve common issues without human intervention.

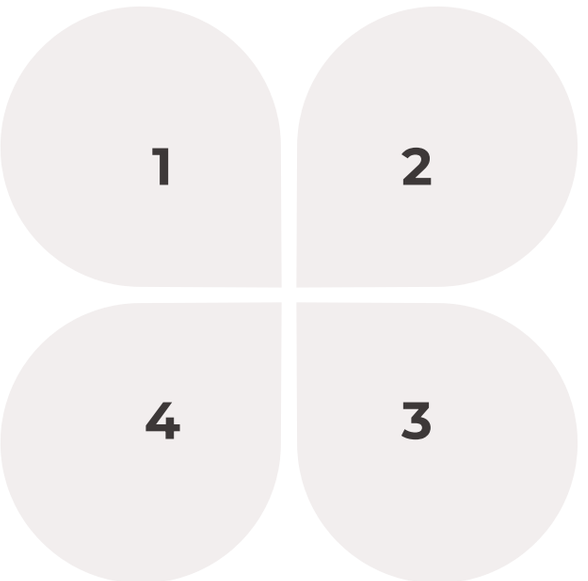
Key Takeaways & Next Steps

Proactive Monitoring

Integrated Azure Monitor and LogicMonitor provide comprehensive visibility into cloud infrastructure, enabling early detection of performance issues before they impact business operations.

Performance Optimization

Data-driven VM sizing ensures optimal resource allocation, eliminating both over-provisioning waste and under-provisioning performance bottlenecks.



Continuous Improvement

Automated reporting and monitoring tools enable ongoing optimization as workload patterns evolve, ensuring sustained efficiency gains over time.

Cost Reduction

Strategic right-sizing combined with reserved instance purchases delivered approximately 30% monthly savings while maintaining or improving application performance.

This project demonstrates how integrated monitoring solutions and systematic analysis can transform cloud resource management. By implementing these optimizations, the organization has not only reduced costs significantly but also improved application performance and reliability.

Moving forward, we recommend implementing automated scaling policies for variable workloads, expanding reserved instance coverage for stable workloads, and establishing quarterly optimization reviews to maintain these efficiency gains as the environment continues to evolve.