Azure KQL Query Library

45+ Production-Ready Queries for Azure Resource Graph, Log Analytics, and Cost Management

Version 2.0 - December 2025
Author: David Swann, Azure Architect
Website: azure-noob.com

About This Library

This query library contains production-tested KQL queries I use daily managing enterprise Azure infrastructure. These aren't textbook examples—they're real queries that solve real operational problems.

What's included:

- Resource inventory and discovery
- Cost analysis and optimization
- Security and compliance auditing
- Performance monitoring
- Network troubleshooting
- Tag governance
- Automation-ready templates

All queries tested on Azure Resource Graph unless noted otherwise.

Table of Contents

---

1. Resource Inventory

Query 1.1: Complete VM Inventory with Power State

```kql
Resources
| where type == "microsoft.compute/virtualmachines"
| extend powerState = properties.extended.instanceView.powerState.code
| join kind=leftouter (
    ResourceContainers
    | where type == 'microsoft.resources/subscriptions'
    | project subscriptionId, subscriptionName = name
) on subscriptionId
| project
    name,
    resourceGroup,
    subscriptionName,
    location,
    vmSize = properties.hardwareProfile.vmSize,
    osType = properties.storageProfile.osDisk.osType,
    powerState,
    tags
```

Use case: Daily VM inventory, capacity planning

Query 1.2: Find VMs Without Backup Configured

```kql
Resources
| where type == "microsoft.compute/virtualmachines"
| where properties.protectedItemId == "" or isempty(properties.protectedItemId)
| project name, resourceGroup, location, tags
```

Use case: Compliance audits, disaster recovery readiness

Query 1.3: Storage Accounts by Size and Cost Tier

```kql
Resources
| where type == "microsoft.storage/storageaccounts"
| extend
    sku = properties.sku.name,
    accessTier = properties.accessTier,
    blobType = properties.primaryEndpoints.blob
| project name, resourceGroup, location, sku, accessTier
```

Use case: Storage cost optimization

Query 1.4: All Resources in a Specific Resource Group

```kql
Resources
| where resourceGroup =~ "YOUR-RESOURCE-GROUP-NAME"
| summarize count() by type
| order by count_ desc
```

Use case: Resource group analysis, decommissioning planning

Query 1.5: Find Resources Created in Last 7 Days

```kql
Resources
| where todatetime(properties.createdTime) >= ago(7d)
| project name, type, resourceGroup, createdTime = properties.createdTime
| order by todatetime(createdTime) desc
```

Use case: Track new resource creation, audit unexpected deployments

---

2. Cost Analysis

Query 2.1: VMs Grouped by Size and Count (Cost Estimation)

```kql
Resources
| where type == "microsoft.compute/virtualmachines"
| extend vmSize = properties.hardwareProfile.vmSize
| summarize count() by vmSize, location
| order by count_ desc
```

Use case: Estimate monthly VM costs by multiplying count × size rate

---

Query 2.2: Unattached Managed Disks (Wasted Money)

```kql
Resources
| where type == "microsoft.compute/disks"
| where properties.diskState == "Unattached"
| extend diskSizeGB = properties.diskSizeGB
| extend sku = properties.sku.name
| project name, resourceGroup, diskSizeGB, sku,
    EstimatedMonthlyCost = case(
        sku contains "Premium", diskSizeGB * 0.12,
        sku contains "Standard", diskSizeGB * 0.05,
        0
    )
| summarize TotalDisks = count(), TotalCost = sum(EstimatedMonthlyCost)
```

Use case: Find quick cost savings, cleanup automation

Query 2.3: Resources Without Cost Center Tags

```kql
```

```
Resources
| where isempty(tags.CostCenter) or tags.CostCenter == ""
| summarize count() by type, resourceGroup
| order by count_ desc
```

Use case: Cost allocation readiness, tag governance

Query 2.4: Storage Accounts with Large Blob Containers

```kql
Resources
| where type == "microsoft.storage/storageaccounts"
| extend
    sku = properties.sku.name,
    usedCapacityGB = properties.primaryEndpoints.blob
| project name, resourceGroup, sku, location
```

Use case: Identify expensive storage, lifecycle policy candidates

---

Query 2.5: Public IPs Costing Money (Static vs Dynamic)

```kql
Resources
| where type == "microsoft.network/publicipaddresses"
| extend
    allocationMethod = properties.publicIPAllocationMethod,
    ipAddress = properties.ipAddress
| where allocationMethod == "Static"
| project name, resourceGroup, allocationMethod, ipAddress
```

Use case: Static IPs cost money even when not attached—find waste

---

3. Security & Compliance

Query 3.1: VMs Without Network Security Groups

```kql
Resources
| where type == "microsoft.compute/virtualmachines"
| extend nicId = tostring(properties.networkProfile.networkInterfaces[0].id)
| join kind=leftouter (
    Resources
    | where type == "microsoft.network/networkinterfaces"
    | extend nsgId = tostring(properties.networkSecurityGroup.id)
    | project nicId = id, nsgId
) on nicId
| where isempty(nsgId)
| project vmName = name, resourceGroup, location
```

Use case: Security compliance, SOC2 audits

Query 3.2: Storage Accounts Allowing Public Access

```kql
Resources
| where type == "microsoft.storage/storageaccounts"
| where properties.allowBlobPublicAccess == true
| project name, resourceGroup, location, tags
```

Use case: Security audits, data exposure risks

Query 3.3: VMs with Old OS Versions (Security Risk)

```kql
Resources
| where type == "microsoft.compute/virtualmachines"
| extend osVersion = properties.storageProfile.imageReference.sku
| where osVersion contains "2012" or osVersion contains "2016"
| project name, resourceGroup, osVersion, location
```

Use case: Identify VMs needing OS upgrades

Query 3.4: Resources Without Owner Tags (Accountability)

```kql
Resources
| where isempty(tags.Owner) or tags.Owner == ""
| summarize count() by type, resourceGroup
| order by count_ desc
```

Use case: Security accountability, incident response contacts

---

Query 3.5: Key Vaults Not Using RBAC (Using Access Policies)

```kql
Resources
| where type == "microsoft.keyvault/vaults"
| where properties.enableRbacAuthorization == false
| project name, resourceGroup, location
```

Use case: Security best practices, migrate to RBAC

---

4. Performance Monitoring

Query 4.1: VMs by Performance Tier (Premium vs Standard Disks)

```kql
Resources
| where type == "microsoft.compute/disks"
| extend diskSku = properties.sku.name
| where diskSku contains "Premium"
| summarize count() by diskSku, location
```

Use case: Performance optimization, cost vs speed tradeoffs

Query 4.2: App Service Plans (Utilization Analysis)

```kql
Resources
| where type == "microsoft.web/serverfarms"
| extend
    sku = properties.sku.name,
    capacity = properties.sku.capacity,
    kind = properties.kind
| project name, resourceGroup, sku, capacity, kind, location
```

Use case: Right-sizing App Service Plans

Query 4.3: Azure SQL Databases by Service Tier

```kql
Resources
| where type == "microsoft.sql/servers/databases"
| extend
    tier = properties.sku.tier,
    capacity = properties.sku.capacity
| where name != "master"
| project name, resourceGroup, tier, capacity, location
```

Use case: Cost optimization, performance tuning

Query 4.4: Find Large VMs (Potential Right-Sizing Candidates)

```kql
Resources
| where type == "microsoft.compute/virtualmachines"
| extend vmSize = properties.hardwareProfile.vmSize
| where vmSize contains "Standard_D16" or vmSize contains "Standard_E32"
| project name, resourceGroup, vmSize, location, tags
```

Use case: Azure Advisor integration, cost optimization

---

5. Network & Connectivity

Query 5.1: Virtual Networks and Address Spaces

```kql
Resources
| where type == "microsoft.network/virtualnetworks"
| extend addressPrefixes = properties.addressSpace.addressPrefixes
| project name, resourceGroup, location, addressPrefixes
```

Use case: IP address management, network planning

Query 5.2: Subnets with Available IP Addresses

```kql
Resources
| where type == "microsoft.network/virtualnetworks"
| mvexpand subnet = properties.subnets
| extend
    subnetName = subnet.name,
    addressPrefix = subnet.properties.addressPrefix,
    usedIPs = array_length(subnet.properties.ipConfigurations)
| project vnetName = name, subnetName, addressPrefix, usedIPs
```

Use case: Capacity planning, subnet exhaustion prevention

Query 5.3: Network Interfaces Without VMs (Orphaned NICs)

```kql
Resources
| where type == "microsoft.network/networkinterfaces"
| where isnull(properties.virtualMachine.id) or isempty(properties.virtualMachine.id)
| project name, resourceGroup, location
```

Use case: Cleanup automation, cost savings

Query 5.4: ExpressRoute Circuits and Bandwidth

```kql
Resources
| where type == "microsoft.network/expressroutecircuits"
| extend
    bandwidth = properties.serviceProviderProperties.bandwidthInMbps,
    provider = properties.serviceProviderProperties.serviceProviderName
| project name, resourceGroup, location, bandwidth, provider
```

Use case: Network capacity planning, cost tracking

Query 5.5: Load Balancers with Backend Pool Count

```kql
Resources
| where type == "microsoft.network/loadbalancers"
| extend backendCount = array_length(properties.backendAddressPools)
| project name, resourceGroup, location, backendCount
```

Use case: Load balancer utilization, rightsizing

---

6. Tag Governance

Query 6.1: Resources Missing Required Tags

```kql
```

```
Resources
| where isempty(tags.Environment)
    or isempty(tags.Owner)
    or isempty(tags.CostCenter)
    or isempty(tags.Application)
| project name, type, resourceGroup, tags
| summarize count() by type
| order by count_ desc
```

Use case: Tag governance enforcement, policy compliance

Query 6.2: Tag Value Variations (Find Inconsistencies)

```
kql

Resources
| where isnotempty(tags.Environment)
| extend env = tostring(tags.Environment)
| summarize count() by env
| order by count_ desc
```

Use case: Find "Production" vs "Prod" vs "production" variations

---

Query 6.3: Resources Tagged for Auto-Shutdown

```
kql

Resources
| where type == "microsoft.compute/virtualmachines"
| where tags.AutoShutdown == "Enabled"
| project name, resourceGroup,
    shutdownTime = tags.ShutdownTime,
    startupTime = tags.StartupTime
```

Use case: Automation validation, cost savings tracking

Query 6.4: Cost Center Tag Coverage by Resource Type

```kql
Resources
| extend hasCostCenter = isnotempty(tags.CostCenter)
| summarize
    Total = count(),
    Tagged = countif(hasCostCenter),
    Untagged = countif(not(hasCostCenter))
    by type
| extend CoveragePercent = round((Tagged * 100.0) / Total, 2)
| order by CoveragePercent asc
```

Use case: Tag governance progress tracking

---

7. Automation Templates

Query 7.1: Export VM Inventory to CSV Format

```kql
Resources
| where type == "microsoft.compute/virtualmachines"
| extend
    vmSize = properties.hardwareProfile.vmSize,
    osType = properties.storageProfile.osDisk.osType,
    powerState = properties.extended.instanceView.powerState.code
| project
    VMName = name,
    ResourceGroup = resourceGroup,
    Location = location,
    Size = vmSize,
    OS = osType,
    PowerState = powerState,
    Owner = tostring(tags.Owner),
    CostCenter = tostring(tags.CostCenter)
```

Use case: Bulk exports to Excel, reporting automation

Query 7.2: Generate Cleanup Script for Orphaned Disks

```kql
Resources
| where type == "microsoft.compute/disks"
| where properties.diskState == "Unattached"
| project
    name,
    resourceGroup,
    AzCommand = strcat("Remove-AzDisk -ResourceGroupName ", resourceGroup, " -DiskName ", name, " -Force")
```

Use case: Copy AzCommand column to PowerShell for bulk cleanup

---

Query 7.3: Find Resources for Tagging Automation

```kql
Resources
| where isempty(tags.Environment)
| project
    id,
    name,
    type,
    resourceGroup,
    TagCommand = strcat("Update-AzTag -ResourceId ", id, " -Tag @{Environment='Production'} -Operation Merge")
```

Use case: Bulk tag application via PowerShell

Query 7.4: Weekly New Resource Report

```kql
Resources
| where todatetime(properties.createdTime) >= ago(7d)
| summarize count() by
    type,
    resourceGroup,
    day = format_datetime(todatetime(properties.createdTime), 'yyyy-MM-dd')
| order by day desc
```

Use case: Weekly email reports, resource sprawl monitoring

Query 7.5: Monthly Cost Estimation Template

```kql
Resources
| where type == "microsoft.compute/virtualmachines"
| extend vmSize = properties.hardwareProfile.vmSize
| extend monthlyCost = case(
    vmSize contains "Standard_B2s", 30.50,
    vmSize contains "Standard_D4s", 175.20,
    vmSize contains "Standard_E8s", 438.00,
    0
)
| summarize
    TotalVMs = count(),
    EstimatedMonthlyCost = sum(monthlyCost)
    by resourceGroup
| order by EstimatedMonthlyCost desc
```

Use case: Budget planning, cost forecasting

Bonus: Advanced Queries

Bonus 1: Cross-Subscription Resource Count

```kql
ResourceContainers
| where type == "microsoft.resources/subscriptions"
| project subscriptionId, subscriptionName = name
| join kind=inner (
    Resources
    | summarize ResourceCount = count() by subscriptionId
) on subscriptionId
| project subscriptionName, ResourceCount
| order by ResourceCount desc
```

Use case: Multi-subscription governance

Bonus 2: Find Resources in Non-Standard Regions

```kql
Resources
| where location !in ("eastus", "eastus2", "westus", "westus2", "centralus")
| summarize count() by location, type
| order by count_ desc
```

Use case: Regional strategy compliance

Bonus 3: VM Extension Inventory (Security Agents)

```kql
Resources
| where type == "microsoft.compute/virtualmachines/extensions"
| extend extensionType = properties.type
| summarize count() by extensionType
| order by count_ desc
```

Use case: Security agent deployment tracking

Query Performance Tips

1. **Filter early:** Put `where` clauses before `extend` and `join`
2. **Use specific types:** `where type == "microsoft.compute/virtualmachines"` before other filters
3. **Limit results:** Add `| take 100` for testing, remove for production
4. **Index friendly:** Use `==` instead of `contains` when possible
5. **Test incrementally:** Build complex queries one operator at a time

---

Next Steps

Want more KQL queries?

- Visit azure-noob.com/blog/kql-cheat-sheet-complete
- Join the newsletter for weekly KQL tips
- Check out the KQL Mastery Hub

Questions or suggestions?

- Email: [contact via website]
- GitHub: Issues and PRs welcome

Version History:

- v2.0 (Dec 2025): Added cost analysis and automation templates
- v1.0 (Jan 2025): Initial release with 30 queries