Complete KQL Query Library for Azure

51 Production-Ready Queries for Azure Resource Graph, Log Analytics & Sentinel

By: Azure Noob (azure-noob.com)

Professional KQL queries tested in enterprise Azure environments managing 31,000+ resources across 44 subscriptions.

Table of Contents

Complete KQL Query Library for Azure

51 Production-Ready Queries for Azure Resource Graph, Log Analytics & Sentinel

By: Azure Noob (azure-noob.com)

Professional KQL queries tested in enterprise Azure environments managing 31,000+ resources across 44 subscriptions

Table of Contents

Section 1: Quick Start Queries

Query 1: List All VMs

```
Resources
| where type == "microsoft.compute/virtualmachines"
| project name, location, resourceGroup
```

Use Case: Basic VM inventory across all subscriptions **Output:** VM name, Azure region, resource group

Query 2: Find VMs in Specific Resource Group

```
Resources
| where type == "microsoft.compute/virtualmachines"
| where resourceGroup == "MyResourceGroup"
| project name, location
```

Use Case: Filter VMs to a specific resource group **Tip:** Replace "MyResourceGroup" with your target resource group

Query 3: VM Count by Location

```
Resources
| where type == "microsoft.compute/virtualmachines"
| summarize count() by location
```

Use Case: Understand VM distribution across Azure regions **Output:** Location and VM count (useful for capacity planning)

Section 2: Core Resource Queries

Query 4: List All VMs Across Subscriptions

```
Resources
| where type == "microsoft.compute/virtualmachines"
| project name, type, location, resourceGroup, subscriptionId
```

Use Case: Multi-subscription VM inventory **Output:** Complete VM list with subscription context

Query 5: List All Network Interfaces

```
Resources
| where type == "microsoft.network/networkinterfaces"
| project name, location, resourceGroup
```

Use Case: Network interface inventory **Note:** Use this as a join target for VM-to-NIC correlation

Query 6: List All Managed Disks

```
Resources
| where type == "microsoft.compute/disks"
| project name, location, resourceGroup
```

Use Case: Disk inventory for cost analysis **Tip:** Add disk size for capacity planning (see Query 25)

Query 7: Filter Resources by Tag

```
Resources
| where tags["Environment"] == "Production"
| project name, type, resourceGroup
```

Use Case: Find resources by tag value **Tip:** Replace "Environment" and "Production" with your tag key/value

Query 8: Resources Missing Critical Tags

```
Resources
| where type in ("microsoft.compute/virtualmachines", "microsoft.storage/storageaccounts")
| where isnull(tags) or array_length(bag_keys(tags)) == 0
| project name, type, resourceGroup, location
```

Use Case: Governance audit - find untagged resources **Why:** Untagged resources complicate cost allocation and compliance

Section 3: Join Queries

Query 9: VMs Joined to NICs

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend NetworkInterfaceId = tostring(properties.networkProfile.networkInterfaces[0].id)
| join kind=leftouter (
    Resources
    | where type == "microsoft.network/networkinterfaces"
    | project NetworkInterfaceId = id, PrivateIP = tostring(properties.ipConfigurations[0].propert
) on NetworkInterfaceId
| project VMName = name, PrivateIP, resourceGroup
```

Use Case: Get VM private IP addresses **Output:** VM name with corresponding private IP

Query 10: VMs Joined to OS Disks

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend DiskId = tostring(properties.storageProfile.osDisk.managedDisk.id)
| join kind=leftouter (
    Resources
    | where type == "microsoft.compute/disks"
    | project DiskId = id, DiskSizeGB = toint(properties.diskSizeGB)
) on DiskId
| project VMName = name, DiskSizeGB, resourceGroup
```

Use Case: Audit VM disk sizes **Why:** Identify oversized disks for cost optimization

Query 11: VMs Joined to Subscriptions

```
Resources
| where type == "microsoft.compute/virtualmachines"
| join kind=leftouter (
    ResourceContainers
    | where type == "microsoft.resources/subscriptions"
    | project subscriptionId, SubscriptionName = name
) on subscriptionId
| project VMName = name, SubscriptionName, resourceGroup
```

Use Case: Add subscription names to VM inventory **Why:** Easier to read than subscription GUIDs

Query 12: Complete VM Inventory (Multi-Join)

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend NetworkInterfaceId = tostring(properties.networkProfile.networkInterfaces[0].id),
        DiskId = tostring(properties.storageProfile.osDisk.managedDisk.id),
        OSType = tostring(properties.storageProfile.osDisk.osType),
        Environment = tostring(tags["Environment"])
| join kind=leftouter (
    Resources
    | where type == "microsoft.network/networkinterfaces"
    | project NetworkInterfaceId = id, PrivateIP = tostring(properties.ipConfigurations[0].propert
) on NetworkInterfaceId
| join kind=leftouter (
    Resources
    | where type == "microsoft.compute/disks"
    | project DiskId = id, DiskSizeGB = toint(properties.diskSizeGB)
) on DiskId
| project VMName = name, PrivateIP, OSType, DiskSizeGB, Environment, resourceGroup
```

Use Case: Comprehensive VM inventory for reporting **Output:** All critical VM details in one query

Section 4: JSON Property Extraction

Query 13: Extract VM Computer Name

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend ComputerName = tostring(properties.osProfile.computerName)
| project VMName = name, ComputerName, resourceGroup
```

Use Case: Get internal hostname vs Azure resource name **Why:** Computer name may differ from VM name

Query 14: Extract NIC Private IP

```
Resources
| where type == "microsoft.network/networkinterfaces"
| extend PrivateIP = tostring(properties.ipConfigurations[0].properties.privateIPAddress)
| project NICName = name, PrivateIP, resourceGroup
```

Use Case: Network inventory and IP address management **Note:** Gets first IP configuration only

Query 15: Extract VNet and Subnet from NIC

```
Resources
| where type == "microsoft.network/networkinterfaces"
| extend SubnetId = tostring(properties.ipConfigurations[0].properties.subnet.id),
         VNetName = split(tostring(properties.ipConfigurations[0].properties.subnet.id), "/")[8],
         SubnetName = split(tostring(properties.ipConfigurations[0].properties.subnet.id), "/")[10
| project NICName = name, VNetName, SubnetName, resourceGroup
```

Use Case: Map NICs to network topology **Why:** Understand which VMs are in which VNets/Subnets

Query 16: Extract Disk Size

```
Resources
| where type == "microsoft.compute/disks"
| extend DiskSizeGB = toint(properties.diskSizeGB)
| project DiskName = name, DiskSizeGB, resourceGroup
```

Use Case: Disk capacity planning and cost analysis **Output:** Disk name and size in GB

Query 17: Extract OS Type from VM

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend OSType = tostring(properties.storageProfile.osDisk.osType)
| project VMName = name, OSType, resourceGroup
```

Use Case: Count Windows vs Linux VMs **Output:** VM name with OS type (Windows/Linux)

Query 18: VM with OS and Disk Details

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend OSType = tostring(properties.storageProfile.osDisk.osType),
         DiskSizeGB = toint(properties.storageProfile.osDisk.diskSizeGB)
| project VMName = name, OSType, DiskSizeGB, resourceGroup
```

Use Case: Combined OS and storage audit **Why:** Single query for VM OS and disk information

Section 5: Conditional Logic & Custom Fields

Query 19: Custom OS Categorization

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend OSType = tostring(properties.storageProfile.osDisk.osType),
        DetailedOS = case(
            OSType == "Linux" and properties.storageProfile.imageReference.publisher == "Canonica
            OSType == "Windows" and properties.storageProfile.imageReference.offer contains "Wind
            "Other"
        )
| project VMName = name, DetailedOS, resourceGroup
```

Use Case: Categorize VMs by OS distribution **Why:** More detailed than just Windows/Linux

Query 20: Extract Owner from Tags

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend Owner = tostring(tags["Owner"])
| project VMName = name, Owner, resourceGroup
```

Use Case: Identify VM ownership for accountability **Note:** Returns null if Owner tag is missing

Query 21: Define Update Strategy by OS

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend OSType = tostring(properties.storageProfile.osDisk.osType),
        UpdateMethod = case(
            OSType == "Windows", "Azure Update Manager",
            OSType == "Linux", "Linux Package Manager",
            "Manual"
        )
| project VMName = name, OSType, UpdateMethod, resourceGroup
```

Use Case: Patching strategy planning **Why:** Different update methods for Windows vs Linux

Section 6: Disk Queries

Query 22: List All Managed Disks with Size

```
Resources
| where type == "microsoft.compute/disks"
| extend DiskSizeGB = toint(properties.diskSizeGB)
| project DiskName = name, DiskSizeGB, location, resourceGroup
| order by DiskSizeGB desc
```

Use Case: Disk inventory sorted by size **Why:** Identify large disks for cost optimization

Query 23: Find Disks Larger Than 100GB

```
Resources
| where type == "microsoft.compute/disks"
| extend DiskSizeGB = toint(properties.diskSizeGB)
| where DiskSizeGB > 100
| project DiskName = name, DiskSizeGB, resourceGroup
```

Use Case: Target large disks for rightsizing **Tip:** Adjust threshold (100GB) based on your needs

Query 24: VMs with OS Disk Sizes

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend DiskId = tostring(properties.storageProfile.osDisk.managedDisk.id)
| join kind=leftouter (
    Resources
    | where type == "microsoft.compute/disks"
    | project DiskId = id, DiskSizeGB = toint(properties.diskSizeGB)
) on DiskId
| project VMName = name, DiskSizeGB, resourceGroup
```

Use Case: VM disk sizing audit **Why:** Correlate VM to disk size in single query

Section 7: Production-Ready Templates

Query 25: Complete VM Inventory

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend NetworkInterfaceId = tostring(properties.networkProfile.networkInterfaces[0].id),
        DiskId = tostring(properties.storageProfile.osDisk.managedDisk.id),
        OSType = tostring(properties.storageProfile.osDisk.osType),
        Environment = tostring(tags["Environment"])
| join kind=leftouter (
    Resources
    | where type == "microsoft.network/networkinterfaces"
    | project NetworkInterfaceId = id, PrivateIP = tostring(properties.ipConfigurations[0].propert
) on NetworkInterfaceId
| join kind=leftouter (
    Resources
    | where type == "microsoft.compute/disks"
    | project DiskId = id, DiskSizeGB = toint(properties.diskSizeGB)
) on DiskId
| project VMName = name, PrivateIP, OSType, DiskSizeGB, Environment, resourceGroup
```

Use Case: Enterprise VM inventory with all details **Output:** Export to Excel for reporting

Query 26: Cost Analysis - Resources by Type and Location

```
Resources
| summarize ResourceCount = count() by type, location
| order by ResourceCount desc
```

Use Case: Understand resource distribution for cost planning **Output:** Resource type, location, and count

Query 27: Security Audit - Untagged Resources

```
Resources
| where type in ("microsoft.compute/virtualmachines", "microsoft.storage/storageaccounts", "micros
| where isnull(tags) or array_length(bag_keys(tags)) == 0
| project name, type, resourceGroup, location
| order by type, name
```

Use Case: Governance audit for compliance **Why:** Untagged resources can't be properly allocated

Query 28: VMs by Subscription and VNet

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend NetworkInterfaceId = tostring(properties.networkProfile.networkInterfaces[0].id)
| join kind=leftouter (
    Resources
    | where type == "microsoft.network/networkinterfaces"
    | extend VNetName = split(tostring(properties.ipConfigurations[0].properties.subnet.id), "/")[
    | project NetworkInterfaceId = id, VNetName
) on NetworkInterfaceId
| join kind=leftouter (
    ResourceContainers
    | where type == "microsoft.resources/subscriptions"
    | project subscriptionId, SubscriptionName = name
) on subscriptionId
| project VMName = name, VNetName, SubscriptionName, resourceGroup
```

Use Case: Network topology mapping **Output:** VMs with their VNet and subscription

Query 29: Find Production VMs with Owner Tag

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend Environment = tostring(tags["Environment"]),
        Owner = tostring(tags["Owner"])
| where Environment == "Production"
| project VMName = name, Environment, Owner, resourceGroup
```

Use Case: Production VM accountability **Why:** Ensure production VMs have owners assigned

Query 30: OS Distribution and Update Strategy

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend OSType = tostring(properties.storageProfile.osDisk.osType),
        OSProduct = tostring(properties.storageProfile.imageReference.offer),
        OSVersion = tostring(properties.storageProfile.imageReference.sku),
        DetailedOS = case(
            OSType == "Linux" and properties.storageProfile.imageReference.publisher == "Canonica
            OSType == "Linux" and properties.storageProfile.imageReference.publisher == "RedHat",
            OSType == "Windows" and OSProduct contains "WindowsServer" and OSVersion contains "20
            OSType == "Windows" and OSProduct contains "WindowsServer", "Windows Server - Other",
            "Unknown"
        ),
        UpdateMethod = case(
            OSType == "Windows" and OSProduct contains "WindowsServer", "Azure Update Manager",
            OSType == "Linux", "Linux Package Manager",
            "Manual"
        )
| project VMName = name, DetailedOS, UpdateMethod, resourceGroup
```

Use Case: Patching strategy planning by OS **Output:** VM inventory with detailed OS and recommended patching method

Section 8: Visualization Queries

Query 31: VMs by OS Type (Chart-Ready)

```
Resources
| where type == "microsoft.compute/virtualmachines"
| summarize count() by OSType = tostring(properties.storageProfile.osDisk.osType)
```

Use Case: Visualize Windows vs Linux distribution **Output:** OS type and count (render as pie chart)

Query 32: Disks by Size (Chart-Ready)

```
Resources
| where type == "microsoft.compute/disks"
| summarize count() by DiskSizeGB = toint(properties.diskSizeGB)
```

Use Case: Disk size distribution visualization **Output:** Disk size and count (render as bar chart)

Section 9: Performance Optimization

Example 1: Filter Early (Bad vs Good)

■ BAD (Slow):

```
Resources
| project name, type, location, resourceGroup, tags
| where type == "microsoft.compute/virtualmachines"
| where resourceGroup == "Production-RG"
```

Why slow: Projects all columns from all resources before filtering

■ GOOD (Fast):

```
Resources
| where type == "microsoft.compute/virtualmachines"
| where resourceGroup == "Production-RG"
| project name, location, tags
```

Why fast: Filters first, then projects only needed columns **Performance gain:** 10-20x faster

Example 2: Project Only What You Need

■ BAD:

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend AllProperties = parse_json(properties)
```

■ GOOD:

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend OSType = tostring(properties.storageProfile.osDisk.osType)
```

Why: Parsing entire JSON objects is expensive

Example 3: Use `in` Instead of Multiple `or`

■ BAD:

```
Resources
| where type == "microsoft.compute/virtualmachines"
   or type == "microsoft.storage/storageaccounts"
   or type == "microsoft.network/networksecuritygroups"
   or type == "microsoft.sql/servers"
```

■ GOOD:

```
Resources
| where type in ("microsoft.compute/virtualmachines",
                 "microsoft.storage/storageaccounts",
                 "microsoft.network/networksecuritygroups",
                 "microsoft.sql/servers")
```

Performance gain: 3-5x faster

Example 4: Limit Results During Testing

Testing queries:

```
Resources
| where type == "microsoft.compute/virtualmachines"
| take 10  // Test with 10 VMs first
| extend OSType = tostring(properties.storageProfile.osDisk.osType)
| project name, OSType
```

Why: Testing on 10 rows is instant vs 10,000 rows

Example 5: Join Efficiently (Bad vs Good)

■ BAD (45 seconds):

```
Resources
| join (Resources | where type == "microsoft.network/networkinterfaces") on $left.id == $right.pro
| where type == "microsoft.compute/virtualmachines"
```

■ GOOD (3 seconds):

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend NetworkInterfaceId = tostring(properties.networkProfile.networkInterfaces[0].id)
| join kind=leftouter (
    Resources
    | where type == "microsoft.network/networkinterfaces"
    | project NetworkInterfaceId = id, PrivateIP = tostring(properties.ipConfigurations[0].propert
) on NetworkInterfaceId
| project VMName = name, PrivateIP
```

Performance gain: 15x faster **Why:** Filter both sides before join, use efficient join key

Section 10: Advanced Techniques

Query 33: Dynamic Columns with mv-expand

Extract all NICs from multi-NIC VMs:

```
Resources
| where type == "microsoft.compute/virtualmachines"
| mv-expand NIC = properties.networkProfile.networkInterfaces
| extend NetworkInterfaceId = tostring(NIC.id)
| join kind=leftouter (
    Resources
    | where type == "microsoft.network/networkinterfaces"
    | project NetworkInterfaceId = id, PrivateIP = tostring(properties.ipConfigurations[0].propert
) on NetworkInterfaceId
| summarize NICs = make_list(PrivateIP) by VMName = name
| project VMName, AllPrivateIPs = NICs
```

Use Case: VMs with multiple network interfaces **Why:** `mv-expand` expands arrays into separate rows

Query 34: Find Resources Missing Critical Tags

```
Resources
| where type in ("microsoft.compute/virtualmachines", "microsoft.storage/storageaccounts")
| extend HasEnvironmentTag = isnotnull(tags["Environment"]),
         HasOwnerTag = isnotnull(tags["Owner"]),
         HasCostCenterTag = isnotnull(tags["CostCenter"])
| where HasEnvironmentTag == false or HasOwnerTag == false or HasCostCenterTag == false
| extend MissingTags = strcat(
    iff(HasEnvironmentTag == false, "Environment ", ""),
    iff(HasOwnerTag == false, "Owner ", ""),
    iff(HasCostCenterTag == false, "CostCenter", "")
)
| project name, type, resourceGroup, MissingTags
| order by type, name
```

Use Case: Tag governance audit **Output:** Resources with exactly which tags are missing

Query 35: Complex JSON Parsing - Image Distribution

```
Resources
| where type == "microsoft.compute/virtualmachines"
| extend ImagePublisher = tostring(properties.storageProfile.imageReference.publisher),
         ImageOffer = tostring(properties.storageProfile.imageReference.offer),
         ImageSku = tostring(properties.storageProfile.imageReference.sku),
         ImageVersion = tostring(properties.storageProfile.imageReference.version),
         FullImageString = strcat(ImagePublisher, ":", ImageOffer, ":", ImageSku, ":", ImageVersio
| summarize VMCount = count() by FullImageString
| order by VMCount desc
```

Use Case: Understand image distribution across environment **Why:** Identify non-standard images for security

Query 36: Cross-Subscription Resource Relationships

```
       Resources
       | where type == "microsoft.compute/virtualmachines"
       | extend NetworkInterfaceId = tostring(properties.networkProfile.networkInterfaces[0].id)
       | join kind=leftouter (
           Resources
           | where type == "microsoft.network/networkinterfaces"
           | extend SubnetId = tostring(properties.ipConfigurations[0].properties.subnet.id)
           | extend VNetSubscription = split(SubnetId, "/")[2]
           | project NetworkInterfaceId = id, VNetSubscription, NICSubscription = subscriptionId
       ) on NetworkInterfaceId
       | where VNetSubscription != NICSubscription
       | project VMName = name, VMSubscription = subscriptionId, VNetSubscription, resourceGroup
```

Use Case: Find VMs connected to VNets in different subscriptions **Why:** Cross-subscription networking creates security/cost complexity

Section 11: Microsoft Sentinel Security Queries

Query 37: Failed Sign-In Attempts

```
SigninLogs
| where TimeGenerated > ago(24h)
| where ResultType != "0"  // 0 = success
| summarize FailedAttempts = count(),
            IPAddresses = make_set(IPAddress),
            Locations = make_set(Location)
            by UserPrincipalName
| where FailedAttempts > 5
| order by FailedAttempts desc
```

Use Case: Detect brute force attacks **Output:** Users with >5 failed logins in 24 hours

Query 38: Track Resource Deletions

```
AzureActivity
| where TimeGenerated > ago(7d)
| where OperationNameValue endswith "/delete"
| where ActivityStatusValue == "Success"
| project TimeGenerated, Caller, OperationNameValue, ResourceId, ResourceGroup
| order by TimeGenerated desc
```

Use Case: Security audit trail for deletions **Why:** Know who deleted what and when

Query 39: Monitor VM Deployments

```
AzureActivity
| where TimeGenerated > ago(30d)
| where ResourceProviderValue == "Microsoft.Compute"
| where OperationNameValue has "virtualMachines/write"
| extend VMName = tostring(parse_json(Properties).resource)
| project TimeGenerated, Caller, VMName, ResourceGroup, SubscriptionId
| order by TimeGenerated desc
```

Use Case: Track VM creation for cost control **Output:** Who created VMs and when

Query 40: Detect Logins from New Countries

```
let UserLocations = SigninLogs
    | where TimeGenerated > ago(90d)
    | summarize KnownCountries = make_set(LocationDetails.countryOrRegion) by UserPrincipalName;
SigninLogs
| where TimeGenerated > ago(24h)
| extend Country = tostring(LocationDetails.countryOrRegion)
| join kind=leftouter (UserLocations) on UserPrincipalName
| where Country !in (KnownCountries)
| project TimeGenerated, UserPrincipalName, Country, IPAddress, ResultType
```

Use Case: Detect compromised accounts **Why:** Unusual login locations indicate compromise

Query 41: Track High-Privilege Role Assignments

```
AzureActivity
| where TimeGenerated > ago(30d)
| where OperationNameValue == "Microsoft.Authorization/roleAssignments/write"
| extend RoleDefinition = tostring(parse_json(Properties).roleDefinitionId)
| where RoleDefinition has "Owner" or RoleDefinition has "Contributor"
| project TimeGenerated, Caller, ResourceId, ResourceGroup, SubscriptionId
| order by TimeGenerated desc
```

Use Case: Monitor privilege escalation **Output:** Who assigned Owner/Contributor roles

Query 42: Watchlist Threat Intelligence Integration

```
let ThreatIPs = _GetWatchlist("KnownBadIPs")
    | project IPAddress = SearchKey;
SigninLogs
| where TimeGenerated > ago(24h)
| where IPAddress in (ThreatIPs)
| project TimeGenerated, UserPrincipalName, IPAddress, Location, ResultType
```

Use Case: Correlate Azure activity with threat intelligence **Note:** Requires Sentinel watchlist named "KnownBadIPs"

Appendix: Quick Reference

KQL vs SQL Translation

| Task | SQL | KQL | |------|-----|-----| | Select all | `SELECT * FROM VMs` | `Resources \| where type == "microsoft.compute/virtualmachines"` | | Filter rows | `WHERE location = 'eastus'` | `\| where location == "eastus"` | | Select columns | `SELECT name, location` | `\| project name, location` | | Count rows | `SELECT COUNT(*)` | `\| count` | | Group by | `GROUP BY location` | `\| summarize count() by location` | | Order results | `ORDER BY name` | `\| order by name` | | Limit results | `LIMIT 10` | `\| take 10` |

Common KQL Operators

| Operator | Purpose | Example | |----------|---------|---------| | `where` | Filter rows | `\| where type == "microsoft.compute/virtualmachines"` | | `project` | Select columns | `\| project name, location` | | `extend` | Add calculated columns | `\| extend OSType = tostring(properties.storageProfile.osDisk.osType)` | | `summarize` | Aggregate data | `\| summarize count() by location` | | `join` | Combine tables | `\| join kind=leftouter (Resources) on id` | | `take` | Limit results | `\| take 10` | | `order by` | Sort results | `\| order by name desc` | | `distinct` |

Unique values | `\| distinct location` |

Common Troubleshooting

Query Timeout: - Add `| take 100` to limit results while testing - Filter early with `where` clauses - Use performance optimization techniques (Section 9)

JSON Parsing Errors: - Always use `tostring()` when extracting from `properties` - Check if property exists with `isnotnull()` first - Use `coalesce()` to provide default values

Join Failures: - Verify join keys match exactly (case-sensitive) - Filter both sides of join before joining - Use `kind=leftouter` to include unmatched rows

Empty Results: - Check resource type spelling (all lowercase) - Verify subscriptions are accessible - Use `| count` to check if any rows exist before filtering

About This Library

Created by: Azure Noob (azure-noob.com) **Tested in:** Enterprise environments managing 31,000+ resources across 44 subscriptions **Last Updated:** December 2025

More Resources: - Complete KQL guide: https://azure-noob.com/blog/kql-cheat-sheet-complete/ - Azure governance hub: https://azure-noob.com/hub/governance/ - Azure FinOps hub: https://azure-noob.com/hub/finops/

Questions? Visit azure-noob.com for more Azure tips, real-world solutions, and production-tested queries.