

LAB10: Query and inventory your Azure Arc-enabled servers using Azure Resource Graph

Student Lab Manual

Table of Contents

Exercise 1 - Use Resource Graph Queries in the Azure Portal and in Powershell to examine your Arc-enabled servers

Task 1 - Apply resource tags to Azure Arc-enabled servers

Task 2 - The Azure Resource Graph Explorer

Task 3 - Run a query to show all Azure Arc-enabled servers in your subscription

Task 4 - Query your server inventory using the available metadata

Task 5 - Use the resource tags in your Graph Query

Task 6 - List the extensions installed on the Azure Arc-enabled servers

Task 7 - Query other properties

====

Exercise 1 - Use Resource Graph Queries in the Azure Portal and in Powershell to examine your Arc-enabled servers

Objective

In this exercise, you will learn how to use the Azure Resource queries both in the Azure Graph Explorer and Powershell to demonstrate inventory management of your Azure Arc connected servers. Note that the results you get by running the graph queries in this module might be different from the sample screenshots as your environment might be different e.g. as a result of working with the other labs.

Estimated Time to Complete This Lab

30 minutes

Explanation

Azure Resource Graph is an extremely powerful extension to Azure Resource Management that provides efficient resource exploration at scale. It also provides the ability to do complex filtering and grouping. It can do this because it uses a subset of the Kusto Query Language (KQL).

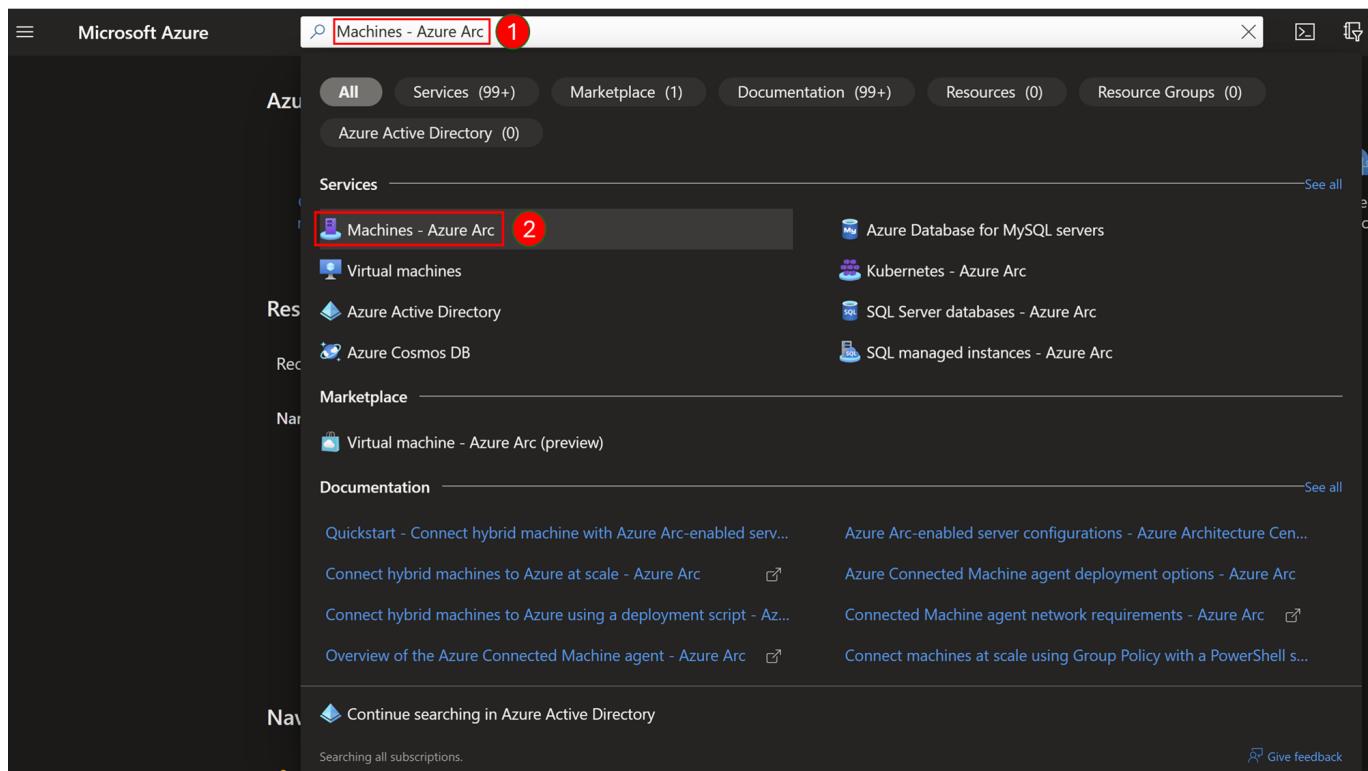
There are a number of tables you can query in Azure Resource Graph. The most common table is the "resources" table. This is where all resources in your Azure subscriptions will live. With few exceptions everything in Azure is a resource. Queries can be run against the Azure Resource Graph API, with PowerShell, or in the Azure portal.

====

Task 1: Apply resource tags to Azure Arc-enabled servers

In this first step, you will assign Azure resource tags to some of your Azure Arc-enabled servers. This gives you the ability to easily organize and manage server inventory.

1. [] Enter "Machines - Azure Arc" in the top search bar in the Azure portal and select it from the displayed services.



2. [] Click on any of your Azure Arc-enabled servers.

The screenshot shows the Azure Arc | Machines blade. On the left, there's a navigation menu with items like Overview, All Azure Arc resources, Azure Arc resources (Machines, Kubernetes clusters), Host environments, Data services, Internet of Things (IoT), Application services, Licenses, Management, and DevOps. The 'Machines' item under 'Azure Arc resources' is selected and highlighted. The main area displays a table with two records:

Name	Kind	Arc agent status	Resource group	Subscription
Arcbox-Ubuntu-01		Connected	ArcBox	MIPDG
ArcBox-Win2K25		Connected	ArcBox	MIPDG

3. [] Click on "Tags". Add a new tag with Name="Scenario" and Value="azure_arc_servers_inventory". Click Apply when ready.

The screenshot shows the Tags page for the machine 'ArcBox-Win2K25'. The left sidebar has a 'Tags' item selected and highlighted with a red box. The main area shows a table with one row:

Name	Value
Project	: jumpstart_arcbox
Scenario	: azure_arc_servers_inventory

At the bottom, there are 'Apply' and 'Discard changes' buttons, with 'Apply' also highlighted with a red box.

4. [] Repeat the same process in other Azure Arc-enabled servers if you wish. This new tag will be used later when working with Resource Graph Explorer queries.

Task 1 has been completed

Click **Next** for the next task or **Go back to the main table of content**

====

Task 2: The Azure Resource Graph Explorer

- Now we will explore our hybrid server inventory using a number of Azure Graph Queries. Enter "Resource Graph Explorer" in the top search bar in the Azure portal and select it.

The screenshot shows the Microsoft Azure portal's search interface. The search bar at the top contains the text "Resource Graph Explorer". Below the search bar, there are several categories and links. The "Services" category is expanded, showing "Resource Graph Explorer" highlighted with a red box and a red circle labeled "2". Other services listed include "Resource groups", "Resource Guards", "All resources", "Bing Resources", and "Resource bridges". The "Documentation" section below lists various Azure Resource Graph tutorials and quickstarts. At the bottom of the search results, there is a "Give feedback" link.

- The scope of the Resource Graph Explorer can be set as seen below

The screenshot shows the Azure Resource Graph Explorer interface. On the left, a sidebar menu is open under the "Scope" heading, showing options like "Subscription", "Directory", and "Management group". A red box highlights the "Subscription" option, which is currently selected. The main area features a query editor with a title "Query 1". Below the editor, there are tabs for "Results", "Charts", and "Messages". The "Results" tab is active. In the results pane, there are three examples of Azure Resource Graph queries: "Count Azure resources", "Count key vault resources", and "List resources sorted by name". Each example includes a brief description and a "Open query" button.

Task 2 has been completed

Click **Next** for the next task or **Go back to the main table of content**

====

Task 3: Run a query to show all Azure Arc-enabled servers in your subscription

1. [] In the query window, enter and run the following query and examine the results which should show your Arc-enabled servers. Note the use of the KQL equals operator (`=~`) which is case insensitive [KQL =~ \(equals\) operator](#).

```
Resources
| where type =~ 'Microsoft.HybridCompute/machines'
```

The screenshot shows the Azure Resource Graph Explorer interface. On the left, there's a sidebar with categories like General, AI + machine learning, Analytics, Compute, Containers, Databases, DevOps, Hybrid + multicloud, Identity, Integration, Internet of Things, Management and governance, and Migration. The main area has a 'Query 1' tab with the following KQL query:

```
1 Resources
2 | where type =~ 'Microsoft.HybridCompute/machines'
```

Below the query, there are tabs for 'Get started', 'Results' (which is selected), 'Charts', and 'Messages'. A red box highlights the 'Results' tab. At the bottom, there's a table with the following data:

Name	Type	tenantId ↑	kind ↑	Location	Resource group	Subscription	managedBy ↑	sku
Arcbox-Ubuntu-01	Machine - Azure Arc	4fce372a-37a4-44f8-9...		UK South	arcbox	MPDG-lod5023453		null
ArcBox-Win2K25	Machine - Azure Arc	4fce372a-37a4-44f8-9...		UK South	arcbox	MPDG-lod5023453		null

A red box highlights the entire table. On the right side of the interface, there are buttons for 'Download formatted results as CSV' and 'Pin to dashboard'. A 'Formatted results' toggle switch is turned on, indicated by a red box.

2. [] Scroll to the right on the results pane and click "See Details" to see all the Azure Arc-enabled server metadata. Note for example the list of detected properties, we will be using these in the next task.
3. [] You can also run the same query using PowerShell (e.g. using Azure Cloud Shell) providing that you have added the required module "Az.ResourceGraph" as explained in [Run your first Resource Graph query using Azure PowerShell](#).

To install the PowerShell module, run the following command

```
Install-Module -Name Az.ResourceGraph
```

Then run the query in PowerShell

```
Search-AzGraph -Query "Resources | where type =~ 'Microsoft.HybridCompute/machines'"
```

Task 3 has been completed

Click **Next** for the next task or [Go back to the main table of content](#)

====

Task 4: Query your server inventory using the available metadata

1. [] Use PowerShell and/or the Resource Graph Explorer to summarize the server count by "logical cores" which is one of the detected properties referred to in the previous task.

Using the Portal:

```
Resources
| where type =~ 'Microsoft.HybridCompute/machines'
| extend logicalCores = tostring(properties.detectedProperties.logicalCoreCount)
| summarize serversCount = count() by logicalCores
```

Using PowerShell:

```
Search-AzGraph -Query "Resources
| where type =~ 'Microsoft.HybridCompute/machines'
| extend logicalCores = tostring(properties.detectedProperties.logicalCoreCount)
| summarize serversCount = count() by logicalCores"
```

2. [] The Graph Explorer allows you to get a graphical view of your results by selecting the "charts" option.

The screenshot shows the Azure Resource Graph Explorer interface. On the left, there's a sidebar with categories like General, AI + machine learn, Analytics, Compute, Containers, Databases, DevOps, Hybrid + multicloud, Identity, Integration, Internet of Things, Management and governance, Migration, Mixed reality, and Monitor. The 'Charts' tab is selected in the top navigation bar. A red box highlights the 'Query 1' section, which contains the PowerShell-like query:

```
Resources
| where type =~ 'Microsoft.HybridCompute/machines'
| extend logicalCores = tostring(properties.detectedProperties.logicalCoreCount)
| summarize serversCount = count() by logicalCores
```

. Below the query, there's a dropdown menu set to 'Donut chart'. A red box also highlights this dropdown. To the right, there's a large blue donut chart with the number '2' in the center, indicating two resources. In the bottom right corner of the chart area, there's a small red box with the number '1'.

Task 4 has been completed

Click **Next** for the next task or [Go back to the main table of content](#)

====

Task 5: Use the resource tags in your Graph Query

1. [] Let's now build a query that uses the tag we assigned earlier to some of our Azure Arc-enabled servers. Use the following query that includes a check for resources that have a value for the "Scenario" tag. Feel free to use the portal or PowerShell. Check that the results match the servers that you set tags for earlier.

Using the Portal:

```
Resources
| where type =~ 'Microsoft.HybridCompute/machines' and
isnotempty(tags['Scenario'])
| extend Scenario = tags['Scenario']
| project name, tags
```

Using PowerShell:

```
Search-AzGraph -Query "Resources
| where type =~ 'Microsoft.HybridCompute/machines' and
isnotempty(tags['Scenario'])
| extend Scenario = tags['Scenario']
| project name, tags"
```

Task 5 has been completed

Click **Next** for the next task or [Go back to the main table of content](#)

====

Task 6: List the extensions installed on the Azure Arc-enabled servers

1. [] Run the following advanced query which allows you to see what extensions are installed on the Arc-enabled servers. Notice that running the query in PowerShell requires us to escape the \$ character as explained in [Escape Characters](#).

Using the Portal:

```
Resources
| where type == 'microsoft.hybridcompute/machines'
| project id, JoinID = toupper(id), ComputerName =
tostring(properties.osProfile.computerName), OSName = tostring(properties.osName)
```

```

| join kind=leftouter(
  Resources
  | where type == 'microsoft.hybridcompute/machines/extensions'
  | project MachineId = toupper(substring(id, 0, indexof(id, '/extensions'))),
ExtensionName = name
) on $left.JoinID == $right.MachineId
| summarize Extensions = make_list(ExtensionName) by id, ComputerName, OSName
| order by tolower(OSName) desc

```

Using PowerShell:

```

Search-AzGraph -Query "Resources
| where type == 'microsoft.hybridcompute/machines'
| project id, JoinID = toupper(id), ComputerName =
  tostring(properties.osProfile.computerName), OSName = tostring(properties.osName)
| join kind=leftouter(
  Resources
  | where type == 'microsoft.hybridcompute/machines/extensions'
  | project MachineId = toupper(substring(id, 0, indexof(id, '/extensions'))),
ExtensionName = name
) on `$left.JoinID == `$right.MachineId
| summarize Extensions = make_list(ExtensionName) by id, ComputerName, OSName
| order by tolower(OSName) desc"

```

- If you have used the portal to run the query then you should see results similar to those shown in the screenshot below.

Name	ComputerName	OSName	Extensions
ArcBox-Win2K25	ArcBox-Win2K25	windows	["ChangeTracking-Windows", "WindowsOsUpdateExtension", "AzureMonito... See details
Arcbox-Ubuntu-01	arcbox-ubuntu-01	linux	["ChangeTracking-Linux", "AzureMonitorLinuxAgent", "LinuxOsUpdateExtens... See details

Task 6 has been completed

Click **Next** for the next task or [Go back to the main table of content](#)

====

Task 7: Query other properties

1. [] Azure Arc provides additional properties on the Azure Arc-enabled server resource that we can query with Resource Graph Explorer. In the following example, we list some of these key properties, like the Azure Arc Agent version installed on your Azure Arc-enabled servers

Using the Portal

```
Resources
| where type == 'Microsoft.HybridCompute/machines'
| extend arcAgentVersion = tostring(properties['agentVersion']), osName =
  tostring(properties['osName']), osVersion = tostring(properties['osVersion']),
  osSku = tostring(properties['osSku']),
  lastStatusChange = tostring(properties['lastStatusChange'])
| project name, arcAgentVersion, osName, osVersion, osSku, lastStatusChange
```

Using PowerShell

```
Search-AzGraph -Query "Resources
| where type == 'Microsoft.HybridCompute/machines'
| extend arcAgentVersion = tostring(properties['agentVersion']), osName =
  tostring(properties['osName']), osVersion = tostring(properties['osVersion']),
  osSku = tostring(properties['osSku']),
  lastStatusChange = tostring(properties['lastStatusChange'])
| project name, arcAgentVersion, osName, osVersion, osSku, lastStatusChange"
```

2. [] Running the query in the portal should result in something like the following

The screenshot shows the Azure Resource Graph Explorer interface. On the left, there's a sidebar with 'Categories' and a search bar. The main area has a 'Query 1' section containing the previously copied PowerShell command. Below it is a results table with columns: name, arcAgentVersion, osName, osVersion, osSku, and lastStatusChange. Two rows of data are visible: 'ArcBox-Win2K25' and 'Arcbox-Ubuntu-01'. A red box highlights the entire results table. At the bottom right of the results table, there's a 'Formatted results' toggle switch which is turned on.

name ↑	arcAgentVersion ↑	osName ↑	osVersion ↑	osSku ↑	lastStatusChange ↑
ArcBox-Win2K25	149.02952.2030	windows	10.0.26100.2605	Windows Server 2025 Datacenter	2025-02-26T12:32:12.188Z
Arcbox-Ubuntu-01	149.02952.182	linux	5.15.0-113-generic	Ubuntu 22.04.4 LTS	2025-02-26T10:03:23.614Z

Task 7 has been completed

Congratulations, you have completed all tasks in this lab

Click **Next** for the next lab or [Go back to the main table of content](#)