Azure Instance Metadata Service for virtual machines - Azure Virtual Machines

By KumariSupriya

Archived: 2026-04-06 01:51:56 UTC

Applies to: ✔️ Linux VMs ✔️ Windows VMs ✔️ Flexible scale sets

The Azure Instance Metadata Service (IMDS) provides information about currently running virtual machine instances. You can use it to manage and configure your virtual machines. This information includes the SKU, storage, network configurations, and upcoming maintenance events. For a complete list of the data available, see the Endpoint Categories Summary.

IMDS is available for running instances of virtual machines (VMs) and scale set instances. All endpoints support VMs created and managed by using Azure Resource Manager. Only the Attested category and Network portion of the Instance category support VMs created by using the classic deployment model. The Attested endpoint does so only to a limited extent.

IMDS is a REST API that's available at a well-known, non-routable IP address (`169.254.169.254`). You can only access it from within the VM. Communication between the VM and IMDS never leaves the host. Have your HTTP clients bypass web proxies within the VM when querying IMDS.

To access IMDS, create a VM from Azure Resource Manager or the Azure portal, and use the following samples. For more examples, see Azure Instance Metadata Samples.

Here's sample code to retrieve all metadata for an instance. To access a specific data source, see Endpoint Categories for an overview of all available features.

Request

Important

This example bypasses proxies. You **must** bypass proxies when querying IMDS. See Proxies for additional information.

Note

IMDS requests must be sent using the VM's primary NIC and primary IP, and DHCP must be enabled.

- Windows
- Linux

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri "http://169.254.169.254/metadata/inst
```

`-NoProxy` requires PowerShell V6 or greater. See our samples repository for examples with older PowerShell versions.

Response

Note

The response is a JSON string. The following example response is formatted for readability.

- Windows
- Linux


```
{
   "compute": {
       "azEnvironment": "AZUREPUBLICCLOUD",
       "additionalCapabilities": {
           "hibernationEnabled": "true"
       },
       "hostGroup": {
         "id": "testHostGroupId"
       },
       "extendedLocation": {
           "type": "edgeZone",
           "name": "microsoftlosangeles"
       },
       "evictionPolicy": "",
       "isHostCompatibilityLayerVm": "true",
       "licenseType":  "Windows_Client",
       "location": "westus",
       "name": "examplevmname",
       "offer": "WindowsServer",
       "osProfile": {
           "adminUsername": "admin",
           "computerName": "examplevmname",
           "disablePasswordAuthentication": "true"
       },
       "osType": "Windows",
       "placementGroupId": "f67c14ab-e92c-408c-ae2d-da15866ec79a",
       "plan": {
           "name": "planName",
           "product": "planProduct",
           "publisher": "planPublisher"
       },
       "platformFaultDomain": "36",
       "platformSubFaultDomain": "",
       "platformUpdateDomain": "42",
       "priority": "Regular",
```

```
        "publicKeys": [{
                "keyData": "ssh-rsa 0",
                "path": "/home/user/.ssh/authorized_keys0"
            },
            {
                "keyData": "ssh-rsa 1",
                "path": "/home/user/.ssh/authorized_keys1"
            }
        ],
        "publisher": "RDFE-Test-Microsoft-Windows-Server-Group",
        "resourceGroupName": "macikgo-test-may-23",
        "resourceId": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/resourceGroups/macikgo-test-may-23/provi
        "securityProfile": {
            "secureBootEnabled": "true",
            "virtualTpmEnabled": "false",
            "encryptionAtHost": "true",
            "securityType": "TrustedLaunch"
        },
        "sku": "2019-Datacenter",
        "storageProfile": {
            "dataDisks": [{
                "bytesPerSecondThrottle": "979202048",
                "caching": "None",
                "createOption": "Empty",
                "diskCapacityBytes": "274877906944",
                "diskSizeGB": "1024",
                "image": {
                  "uri": ""
                },
                "isSharedDisk": "false",
                "isUltraDisk": "true",
                "lun": "0",
                "managedDisk": {
                  "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/resourceGroups/macikgo-test-may-23/prov
                  "storageAccountType": "StandardSSD_LRS"
                },
                "name": "exampledatadiskname",
                "opsPerSecondThrottle": "65280",
                "vhd": {
                  "uri": ""
                },
                "writeAcceleratorEnabled": "false"
            }],
            "imageReference": {
                "id": "",
                "offer": "WindowsServer",
                "publisher": "MicrosoftWindowsServer",
```

```
            "sku": "2019-Datacenter",
            "version": "latest",
            "communityGalleryImageId": "/CommunityGalleries/testgallery/Images/1804Gen2/Versions/latest",
            "sharedGalleryImageId": "/SharedGalleries/1P/Images/gen2/Versions/latest",
            "exactVersion": "1.1686127202.30113"
        },
        "osDisk": {
            "caching": "ReadWrite",
            "createOption": "FromImage",
            "diskSizeGB": "30",
            "diffDiskSettings": {
                "option": "Local"
            },
            "encryptionSettings": {
              "enabled": "false",
              "diskEncryptionKey": {
                "sourceVault": {
                  "id": "/subscriptions/test-source-guid/resourceGroups/testrg/providers/Microsoft.KeyVault/
                },
                "secretUrl": "https://test-disk.vault.azure.net/secrets/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/xx
              },
              "keyEncryptionKey": {
                "sourceVault": {
                  "id": "/subscriptions/test-key-guid/resourceGroups/testrg/providers/Microsoft.KeyVault/vau
                },
                "keyUrl": "https://test-key.vault.azure.net/secrets/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/xxxxx-x
              }
            },
            "image": {
                "uri": ""
            },
            "managedDisk": {
                "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/resourceGroups/macikgo-test-may-23/pr
                "storageAccountType": "StandardSSD_LRS"
            },
            "name": "exampleosdiskname",
            "osType": "Windows",
            "vhd": {
                "uri": ""
            },
            "writeAcceleratorEnabled": "false"
        },
        "resourceDisk": {
            "size": "4096"
        }
    },
    "subscriptionId": "xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx",
```

```
        "tags": "baz:bash;foo:bar",
        "userData": "Zm9vYmFy",
        "version": "15.05.22",
        "virtualMachineScaleSet": {
            "id": "/subscriptions/xxxxxxxx-xxxxx-xxx-xxx-xxxx/resourceGroups/resource-group-name/providers/Micr
        },
        "vmId": "02aab8a4-74ef-476e-8182-f6d2ba4166a6",
        "vmScaleSetName": "crpteste9vflji9",
        "vmSize": "Standard_A3",
        "zone": ""
    },
    "network": {
        "interface": [{
            "ipv4": {
               "ipAddress": [{
                    "privateIpAddress": "10.144.133.132",
                    "publicIpAddress": ""
                }],
                "subnet": [{
                    "address": "10.144.133.128",
                    "prefix": "26"
                }]
            },
            "ipv6": {
                "ipAddress": [
                ]
            },
            "macAddress": "0011AAFFBB22"
        }]
    }
}
```


The Instance Metadata Service is only accessible from within a running virtual machine instance on a non-routable IP address. VMs can only interact with their own metadata/functionality. The API is HTTP only and never leaves the host.

In order to ensure that requests are directly intended for IMDS and prevent unintended or unwanted redirection of requests, requests:

- **Must** contain the header `Metadata: true`
- Must **not** contain an `X-Forwarded-For` header

Any request that doesn't meet **both** of these requirements are rejected by the service.

Important

IMDS is **not** a channel for sensitive data. The API is unauthenticated and open to all processes on the VM. Information exposed through this service should be considered as shared information to all applications running inside the VM.

If it isn't necessary for every process on the VM to access IMDS endpoint, you can set local firewall rules to limit the access. For example, if only a known system service needs to access instance metadata service, you can set a firewall rule on IMDS endpoint, only allowing the specific process(es) to access, or denying access for the rest of the processes.

IMDS is **not** intended to be used behind a proxy and doing so is unsupported. Most HTTP clients provide an option for you to disable proxies on your requests, and this functionality must be utilized when communicating with IMDS. Consult your client's documentation for details.

Important

Even if you don't know of any proxy configuration in your environment, **you still must override any default client proxy settings**. Proxy configurations can be automatically discovered, and failing to bypass such configurations exposes you to outage risks should the machine's configuration be changed in the future.

In general, requests to IMDS are limited to 5 requests per second (on a per VM basis). Requests exceeding this threshold will be rejected with 429 responses. Requests to the Managed Identity category are limited to 20 requests per second and 5 concurrent requests (on a per VM basis).

The following HTTP verbs are currently supported:

Verb	Description
`GET`	Retrieve the requested resource

Endpoints may support required and/or optional parameters. See Schema and the documentation for the specific endpoint in question for details.

IMDS endpoints support HTTP query string parameters. For example:

```
http://169.254.169.254/metadata/instance/compute?api-version=2025-04-07&format=json
```

Specifies the parameters:

Name	Value
`api-version`	`2025-04-07`
`format`	`json`

Requests with duplicate query parameter names will be rejected.

For some endpoints that return larger json blobs, we support appending route parameters to the request endpoint to filter down to a subset of the response:

```
http://169.254.169.254/metadata/<endpoint>/[<filter parameter>/...]?<query parameters>
```

The parameters correspond to the indexes/keys that would be used to walk down the json object were you interacting with a parsed representation.

For example, `/metadata/instance` returns the json object:

```
{
   "compute": { ... },
   "network": {
       "interface": [
           {
               "ipv4": {
                  "ipAddress": [{
                       "privateIpAddress": "10.144.133.132",
                       "publicIpAddress": ""
                   }],
                   "subnet": [{
                       "address": "10.144.133.128",
                       "prefix": "26"
                   }]
               },
               "ipv6": {
                   "ipAddress": [{
                       "privateIpAddress": "b4bc:8fce:f33b:4990:cced:d94e:ab4f:6ea0"
                   }]
               },
               "macAddress": "0011AAFFBB22"
           },
           ...
       ]
   }
}
```

If we want to filter the response down to just the compute property, we would send the request:

```
http://169.254.169.254/metadata/instance/compute?api-version=<version>
```

Similarly, if we want to filter to a nested property or specific array element we keep appending keys:

```
http://169.254.169.254/metadata/instance/network/interface/0?api-version=<version>
```

would filter to the first element from the `Network.interface` property and return:

```
{
   "ipv4": {
      "ipAddress": [{
          "privateIpAddress": "10.144.133.132",
          "publicIpAddress": ""
       }],
       "subnet": [{
          "address": "10.144.133.128",
          "prefix": "26"
       }]
   },
   "ipv6": {
       "ipAddress": [{
          "privateIpAddress": "b4bc:8fce:f33b:4990:cced:d94e:ab4f:6ea0"
       }]
   },
   "macAddress": "0011AAFFBB22"
}
```

Note

When filtering to a leaf node, `format=json` doesn't work. For these queries `format=text` needs to be explicitly specified since the default format is json.

By default, IMDS returns data in JSON format (`Content-Type: application/json`). However, endpoints that support response filtering (see Route Parameters) also support the format `text` .

To access a non-default response format, specify the requested format as a query string parameter in the request. For example:

- Windows
- Linux

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri "http://169.254.169.254/metadata/ins
```

In json responses, all primitives will be of type `string` , and missing or inapplicable values are always included but will be set to an empty string.

IMDS is versioned and specifying the API version in the HTTP request is mandatory. The only exception to this requirement is the versions endpoint, which can be used to dynamically retrieve the available API versions.

As newer versions are added, older versions can still be accessed for compatibility if your scripts have dependencies on specific data formats.

When you don't specify a version, you get an error with a list of the newest supported versions:


```
 {
    "error": "Bad request. api-version is invalid or was not specified in the request. For more information refe
    "newest-versions": [
        "2025-04-07",
        "2024-07-17",
        "2024-03-15"
    ]
}
```


- 2025-04-07
- 2024-07-17
- 2024-03-15
- 2023-11-15
- 2023-07-01
- 2021-12-13
- 2021-11-15
- 2021-11-01
- 2021-10-01
- 2021-08-01
- 2021-05-01
- 2021-03-01
- 2021-02-01
- 2021-01-01
- 2020-12-01
- 2020-10-01
- 2020-09-01
- 2020-07-15
- 2020-06-01
- 2019-11-01
- 2019-08-15
- 2019-08-01
- 2019-06-04
- 2019-06-01
- 2019-04-30
- 2019-03-11
- 2019-02-01
- 2018-10-01
- 2018-04-02
- 2018-02-01
- 2017-12-01
- 2017-10-01

- 2017-08-01
- 2017-04-02
- 2017-03-01

A full Swagger definition for IMDS is available at: https://github.com/Azure/azure-rest-api-specs/blob/main/specification/imds/data-plane/InstanceMetadataService/readme.md

The service is **generally available** in all Azure clouds.

The root endpoint is `http://169.254.169.254/metadata`.

The IMDS API contains multiple endpoint categories representing different data sources, each of which contains one or more endpoints. See each category for details.

Category root	Description	Version introduced
`/metadata/attested`	See Attested Data	2018-10-01
`/metadata/identity`	See Managed Identity via IMDS	2018-02-01
`/metadata/instance`	See Instance Metadata	2017-04-02
`/metadata/loadbalancer`	See Retrieve Load Balancer metadata via IMDS	2020-10-01
`/metadata/scheduledevents`	See Scheduled Events via IMDS	2017-08-01
`/metadata/versions`	See Versions	N/A

Returns the set of supported API versions.

```
GET /metadata/versions
```

None (this endpoint is unversioned).

```
{
 "apiVersions": [
   "2017-03-01",
   "2017-04-02",
   ...
 ]
}
```

Exposes the important metadata for the VM instance, including compute, network, and storage.

```
GET /metadata/instance
```

Name	Required/Optional	Description
`api-version`	Required	The version used to service the request.
`format`	Optional*	The format (`json` or `text`) of the response. *Note: May be required when using request parameters

This endpoint supports response filtering via route parameters.

- Windows
- Linux


```
{
   "compute": {
       "azEnvironment": "AZUREPUBLICCLOUD",
       "additionalCapabilities": {
           "hibernationEnabled": "true"
       },
       "hostGroup": {
         "id": "testHostGroupId"
       },
       "extendedLocation": {
           "type": "edgeZone",
           "name": "microsoftlosangeles"
       },
       "evictionPolicy": "",
       "isHostCompatibilityLayerVm": "true",
       "licenseType":  "Windows_Client",
       "location": "westus",
       "name": "examplevmname",
       "offer": "WindowsServer",
       "osProfile": {
           "adminUsername": "admin",
           "computerName": "examplevmname",
           "disablePasswordAuthentication": "true"
       },
       "osType": "Windows",
       "placementGroupId": "f67c14ab-e92c-408c-ae2d-da15866ec79a",
       "plan": {
           "name": "planName",
           "product": "planProduct",
           "publisher": "planPublisher"
       },
       "platformFaultDomain": "36",
       "platformSubFaultDomain": "",
```

```
        "platformUpdateDomain": "42",
        "priority": "Regular",
        "publicKeys": [{
                "keyData": "ssh-rsa 0",
                "path": "/home/user/.ssh/authorized_keys0"
            },
            {
                "keyData": "ssh-rsa 1",
                "path": "/home/user/.ssh/authorized_keys1"
            }
        ],
        "publisher": "RDFE-Test-Microsoft-Windows-Server-Group",
        "resourceGroupName": "macikgo-test-may-23",
        "resourceId": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/resourceGroups/macikgo-test-may-23/provi
        "securityProfile": {
            "secureBootEnabled": "true",
            "virtualTpmEnabled": "false",
            "encryptionAtHost": "true",
            "securityType": "TrustedLaunch"
        },
        "sku": "2019-Datacenter",
        "storageProfile": {
            "dataDisks": [{
                "bytesPerSecondThrottle": "979202048",
                "caching": "None",
                "createOption": "Empty",
                "diskCapacityBytes": "274877906944",
                "diskSizeGB": "1024",
                "image": {
                  "uri": ""
                },
                "isSharedDisk": "false",
                "isUltraDisk": "true",
                "lun": "0",
                "managedDisk": {
                  "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/resourceGroups/macikgo-test-may-23/prov
                  "storageAccountType": "StandardSSD_LRS"
                },
                "name": "exampledatadiskname",
                "opsPerSecondThrottle": "65280",
                "vhd": {
                  "uri": ""
                },
                "writeAcceleratorEnabled": "false"
            }],
            "imageReference": {
                "id": "",
```

```
            "offer": "WindowsServer",
            "publisher": "MicrosoftWindowsServer",
            "sku": "2019-Datacenter",
            "version": "latest",
            "communityGalleryImageId": "/CommunityGalleries/testgallery/Images/1804Gen2/Versions/latest",
            "sharedGalleryImageId": "/SharedGalleries/1P/Images/gen2/Versions/latest",
            "exactVersion": "1.1686127202.30113"
        },
        "osDisk": {
            "caching": "ReadWrite",
            "createOption": "FromImage",
            "diskSizeGB": "30",
            "diffDiskSettings": {
                "option": "Local"
            },
            "encryptionSettings": {
              "enabled": "false",
              "diskEncryptionKey": {
                "sourceVault": {
                  "id": "/subscriptions/test-source-guid/resourceGroups/testrg/providers/Microsoft.KeyVault/
                },
                "secretUrl": "https://test-disk.vault.azure.net/secrets/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/xx
              },
              "keyEncryptionKey": {
                "sourceVault": {
                  "id": "/subscriptions/test-key-guid/resourceGroups/testrg/providers/Microsoft.KeyVault/vau
                },
                "keyUrl": "https://test-key.vault.azure.net/secrets/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/xxxxx-x
              }
            },
            "image": {
                "uri": ""
            },
            "managedDisk": {
                "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/resourceGroups/macikgo-test-may-23/pr
                "storageAccountType": "StandardSSD_LRS"
            },
            "name": "exampleosdiskname",
            "osType": "Windows",
            "vhd": {
                "uri": ""
            },
            "writeAcceleratorEnabled": "false"
        },
        "resourceDisk": {
            "size": "4096"
        }
```

```
        },
        "subscriptionId": "xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx",
        "tags": "baz:bash;foo:bar",
        "userData": "Zm9vYmFy",
        "version": "15.05.22",
        "virtualMachineScaleSet": {
            "id": "/subscriptions/xxxxxxxx-xxxxx-xxx-xxx-xxxx/resourceGroups/resource-group-name/providers/Micro
        },
        "vmId": "02aab8a4-74ef-476e-8182-f6d2ba4166a6",
        "vmScaleSetName": "crpteste9vflji9",
        "vmSize": "Standard_A3",
        "zone": ""
    },
    "network": {
        "interface": [{
            "ipv4": {
               "ipAddress": [{
                    "privateIpAddress": "10.144.133.132",
                    "publicIpAddress": ""
                }],
                "subnet": [{
                    "address": "10.144.133.128",
                    "prefix": "26"
                }]
            },
            "ipv6": {
                "ipAddress": [
                 ]
            },
            "macAddress": "0011AAFFBB22"
        }]
    }
}
```


Schema breakdown:

Compute

Data	Description	Version introduced
`azEnvironment`	Azure Environment where the VM is running in	2018-10-01
`additionalCapabilities.hibernationEnabled`	Identifies if hibernation is enabled on the VM	2021-11-01

Data	Description	Version introduced
`customData`	This feature is deprecated and disabled in IMDS. It has been superseded by `userData`	2019-02-01
`evictionPolicy`	Sets how a Spot VM will be evicted.	2020-12-01
`extendedLocation.type`	Type of the extended location of the VM.	2021-03-01
`extendedLocation.name`	Name of the extended location of the VM	2021-03-01
`host.id`	Name of the host of the VM. Note that a VM will either have a host or a hostGroup but not both.	2021-11-15
`hostGroup.id`	Name of the hostGroup of the VM. Note that a VM will either have a host or a hostGroup but not both.	2021-11-15
`isHostCompatibilityLayerVm`	Identifies if the VM runs on the Host Compatibility Layer	2020-06-01
`licenseType`	Type of license for Azure Hybrid Benefit. This is only present for AHB-enabled VMs	2020-09-01
`location`	Azure Region the VM is running in	2017-04-02
`name`	Name of the VM	2017-04-02
`offer`	Offer information for the VM image and is only present for images deployed from Azure image gallery	2017-04-02
`osProfile.adminUsername`	Specifies the name of the admin account	2020-07-15
`osProfile.computerName`	Specifies the name of the computer	2020-07-15
`osProfile.disablePasswordAuthentication`	Specifies if password authentication is disabled. This is only present for Linux VMs	2020-10-01
`osType`	Linux or Windows	2017-04-02
`physicalZone`	Physical zone of the VM	2023-11-15
`placementGroupId`	Placement Group of your scale set	2017-08-01

Data	Description	Version introduced
`plan`	Plan containing name, product, and publisher for a VM if it's an Azure Marketplace Image	2018-04-02
`platformUpdateDomain`	Update domain the VM is running in	2017-04-02
`platformFaultDomain`	Fault domain the VM is running in	2017-04-02
`platformSubFaultDomain`	Sub fault domain the VM is running in, if applicable.	2021-10-01
`priority`	Priority of the VM. Refer to Spot VMs for more information	2020-12-01
`provider`	Provider of the VM	2018-10-01
`publicKeys`	Collection of Public Keys assigned to the VM and paths	2018-04-02
`publisher`	Publisher of the VM image	2017-04-02
`resourceGroupName`	Resource group for your Virtual Machine	2017-08-01
`resourceId`	The fully qualified ID of the resource	2019-03-11
`sku`	Specific SKU for the VM image	2017-04-02
`securityProfile.secureBootEnabled`	Identifies if UEFI secure boot is enabled on the VM	2020-06-01
`securityProfile.virtualTpmEnabled`	Identifies if the virtual Trusted Platform Module (TPM) is enabled on the VM	2020-06-01
`securityProfile.encryptionAtHost`	Identifies if Encryption at Host is enabled on the VM	2021-11-01
`securityProfile.securityType`	Identifies if the VM is a Trusted VM or a Confidential VM	2021-12-13
`storageProfile`	See Storage Profile below	2019-06-01
`subscriptionId`	Azure subscription for the Virtual Machine	2017-08-01
`tags`	Tags for your Virtual Machine	2017-08-01

Data	Description	Version introduced
`tagsList`	Tags formatted as a JSON array for easier programmatic parsing	2019-06-04
`userData`	The set of data specified when the VM was created for use during or after provisioning (Base64 encoded)	2021-01-01
`version`	Version of the VM image	2017-04-02
`virtualMachineScaleSet.id`	ID of the Virtual Machine Scale Set created with flexible orchestration the Virtual Machine is part of. This field isn't available for Virtual Machine Scale Sets created with uniform orchestration.	2021-03-01
`vmId`	Unique identifier for the VM. The blog referenced only suits for VMs that have SMBIOS < 2.6. For VMs that have SMBIOS >= 2.6, the UUID from DMI is displayed in little-endian format, thus, there's no requirement to switch bytes.	2017-04-02
`vmScaleSetName`	Virtual Machine Scale Set Name of your scale set	2017-12-01
`vmSize`	VM size	2017-04-02
`zone`	Availability Zone of your virtual machine	2017-12-01

† This version isn't fully available yet and may not be supported in all regions.

Storage profile

The storage profile of a VM is divided into three categories: image reference, OS disk, and data disks, plus an additional object for the local temporary disk.

The image reference object contains the following information about the OS image, please note that an image could come either from the platform, marketplace, community gallery, or direct shared gallery but not both:

Data	Description	Version introduced
`id`	Resource ID	2019-06-01

Data	Description	Version introduced
`offer`	Offer of the platform or marketplace image	2019-06-01
`publisher`	Publisher of the platform or marketplace image	2019-06-01
`sku`	Sku of the platform or marketplace image	2019-06-01
`version`	Version of the image	2019-06-01
`communityGalleryImageId`	Resource ID of the community image, empty otherwise	2023-07-01
`sharedGalleryImageId`	Resource ID o direct shared image, empty otherwise	2023-07-01
`exactVersion`	Version of the community or direct shared image	2023-07-01

The OS disk object contains the following information about the OS disk used by the VM:

Data	Description
`caching`	Caching requirements
`createOption`	Information about how the VM was created
`diffDiskSettings`	Ephemeral disk settings
`diskSizeGB`	Size of the disk in GB
`image`	Source user image virtual hard disk
`managedDisk`	Managed disk parameters
`name`	Disk name
`vhd`	Virtual hard disk
`writeAcceleratorEnabled`	Whether or not writeAccelerator is enabled on the disk

The data disks array contains a list of data disks attached to the VM. Each data disk object contains the following information:

Data	Description	Version introduced
`bytesPerSecondThrottle` *	Disk read/write quota in bytes	2021-05-01
`caching`	Caching requirements	2019-06-01
`createOption`	Information about how the VM was created	2019-06-01

Data	Description	Version introduced
`diffDiskSettings`	Ephemeral disk settings	2019-06-01
`diskCapacityBytes` *	Size of disk in bytes	2021-05-01
`diskSizeGB`	Size of the disk in GB	2019-06-01
`encryptionSettings`	Encryption settings for the disk	2019-06-01
`image`	Source user image virtual hard disk	2019-06-01
`isSharedDisk` *	Identifies if the disk is shared between resources	2021-05-01
`isUltraDisk`	Identifies if the data disk is an Ultra Disk	2021-05-01
`lun`	Logical unit number of the disk	2019-06-01
`managedDisk`	Managed disk parameters	2019-06-01
`name`	Disk name	2019-06-01
`opsPerSecondThrottle` *	Disk read/write quota in IOPS	2021-05-01
`osType`	Type of OS included in the disk	2019-06-01
`vhd`	Virtual hard disk	2019-06-01
`writeAcceleratorEnabled`	Whether or not writeAccelerator is enabled on the disk	2019-06-01

*These fields are only populated for Ultra Disks; they are empty strings from non-Ultra Disks.

The encryption settings blob contains data about how the disk is encrypted (if it's encrypted):

Data	Description	Version introduced
`diskEncryptionKey.sourceVault.id`	The location of the disk encryption key	2021-11-01
`diskEncryptionKey.secretUrl`	The location of the secret	2021-11-01
`keyEncryptionKey.sourceVault.id`	The location of the key encryption key	2021-11-01
`keyEncryptionKey.keyUrl`	The location of the key	2021-11-01

The resource disk object contains the size of the Local Temp Disk attached to the VM, if it has one, in kilobytes. If there's no local temp disk for the VM, this value is 0.

Data	Description	Version introduced
`resourceDisk.size`	Size of the local temp disk for the VM (in kB)	2021-02-01

Network

Data	Description	Version introduced
`ipv4.ipAddress.privateIpAddress`	Local IPv4 address of the VM	2017-04-02
`ipv4.ipAddress.publicIpAddress`	Public IPv4 address of the VM	2017-04-02
`ipv4.subnet.address`	Subnet address of the VM	2017-04-02
`ipv4.subnet.prefix`	Subnet prefix, example 24	2017-04-02
`ipv6.ipAddress.privateIpAddress`	Local IPv6 address of the VM	2017-04-02
`macAddress`	VM mac address	2017-04-02

Note

The nics returned by the network call are not guaranteed to be in order.

When creating a new VM, you can specify a set of data to be used during or after the VM provision, and retrieve it through IMDS. Check the end to end user data experience here.

To set up user data, utilize the quickstart template here. The sample below shows how to retrieve this data through IMDS. This feature is released with version `2021-01-01` and above.

Note

Security notice: IMDS is open to all applications on the VM, sensitive data should not be placed in the user data.

- Windows
- Linux

```
 $userData = Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri "http://169.254.169.254/m
[System.Text.Encoding]::UTF8.GetString([Convert]::FromBase64String($userData))
```

As a service provider, you may require to track the number of VMs running your software or have agents that need to track uniqueness of the VM. To be able to get a unique ID for a VM, use the `vmId` field from Instance Metadata Service.

Request

- Windows
- Linux

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri "http://169.254.169.254/metadata/inst
```

Response

```
5c08b38e-4d57-4c23-ac45-aca61037f084
```

For certain scenarios, placement of different data replicas is of prime importance. For example, HDFS replica placement or container placement via an orchestrator might require you to know the `platformFaultDomain` and `platformUpdateDomain` the VM is running on. You can also use Availability Zones for the instances to make these decisions. You can query this data directly via IMDS.

Request

- Windows
- Linux

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri "http://169.254.169.254/metadata/ins
```

Response

```
0
```

VM tags are included the instance API under instance/compute/tags endpoint. Tags may have been applied to your Azure VM to logically organize them into a taxonomy. The tags assigned to a VM can be retrieved by using the request below.

Request

- Windows
- Linux

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri "http://169.254.169.254/metadata/ins
```

Response

```
Department:IT;ReferenceNumber:123456;TestStatus:Pending
```

The `tags` field is a string with the tags delimited by semicolons. This output can be a problem if semicolons are used in the tags themselves. If a parser is written to programmatically extract the tags, you should rely on the `tagsList` field. The `tagsList` field is a JSON array with no delimiters, and consequently, easier to parse. The tagsList assigned to a VM can be retrieved by using the request below.

Request

- Windows
- Linux

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri "http://169.254.169.254/metadata/inst
```

Response

- Windows
- Linux

```
{
   "value":  [
                {
                    "name":  "Department",
                    "value":  "IT"
                },
                {
                    "name":  "ReferenceNumber",
                    "value":  "123456"
                },
                {
                    "name":  "TestStatus",
                    "value":  "Pending"
                }
            ],
   "Count":  3
}
```

As a service provider, you may get a support call where you would like to know more information about the VM. Asking the customer to share the compute metadata can provide basic information for the support professional to know about the kind of VM on Azure.

Request

- Windows
- Linux

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri "http://169.254.169.254/metadata/inst
```

Response

Note

The response is a JSON string. The following example response is pretty-printed for readability.

- Windows
- Linux

```
{
   "azEnvironment": "AZUREPUBLICCLOUD",
   "extendedLocation": {
     "type": "edgeZone",
     "name": "microsoftlosangeles"
   },
   "evictionPolicy": "",
   "additionalCapabilities": {
       "hibernationEnabled": "false"
   },
   "hostGroup": {
     "id": "testHostGroupId"
   },
   "isHostCompatibilityLayerVm": "true",
   "licenseType":  "Windows_Client",
   "location": "westus",
   "name": "examplevmname",
   "offer": "WindowsServer",
   "osProfile": {
       "adminUsername": "admin",
       "computerName": "examplevmname",
       "disablePasswordAuthentication": "true"
   },
   "osType": "Windows",
   "placementGroupId": "f67c14ab-e92c-408c-ae2d-da15866ec79a",
   "plan": {
       "name": "planName",
       "product": "planProduct",
       "publisher": "planPublisher"
   },
   "platformFaultDomain": "36",
   "platformUpdateDomain": "42",
   "priority": "Regular",
   "publicKeys": [{
           "keyData": "ssh-rsa 0",
           "path": "/home/user/.ssh/authorized_keys0"
       },
       {
           "keyData": "ssh-rsa 1",
           "path": "/home/user/.ssh/authorized_keys1"
       }
   ],
   "publisher": "RDFE-Test-Microsoft-Windows-Server-Group",
   "physicalZone": "useast-AZ01",
   "resourceGroupName": "macikgo-test-may-23",
   "resourceId": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/resourceGroups/macikgo-test-may-23/providers/
```

```
    "securityProfile": {
        "secureBootEnabled": "true",
        "virtualTpmEnabled": "false",
        "encryptionAtHost": "true",
        "securityType": "TrustedLaunch"
    },
    "sku": "2019-Datacenter",
    "storageProfile": {
        "dataDisks": [{
            "bytesPerSecondThrottle": "979202048",
            "caching": "None",
            "createOption": "Empty",
            "diskCapacityBytes": "274877906944",
            "diskSizeGB": "1024",
            "image": {
              "uri": ""
            },
            "isSharedDisk": "false",
            "isUltraDisk": "true",
            "lun": "0",
            "managedDisk": {
              "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/resourceGroups/macikgo-test-may-23/provider
              "storageAccountType": "StandardSSD_LRS"
            },
            "name": "exampledatadiskname",
            "opsPerSecondThrottle": "65280",
            "vhd": {
              "uri": ""
            },
            "writeAcceleratorEnabled": "false"
        }],
        "imageReference": {
            "id": "",
            "offer": "WindowsServer",
            "publisher": "MicrosoftWindowsServer",
            "sku": "2019-Datacenter",
            "version": "latest",
            "communityGalleryImageId": "/CommunityGalleries/testgallery/Images/1804Gen2/Versions/latest",
            "sharedGalleryImageId": "/SharedGalleries/1P/Images/gen2/Versions/latest",
            "exactVersion": "1.1686127202.30113"
        },
        "osDisk": {
            "caching": "ReadWrite",
            "createOption": "FromImage",
            "diskSizeGB": "30",
            "diffDiskSettings": {
                "option": "Local"
```

```
            },
            "encryptionSettings": {
              "enabled": "false",
              "diskEncryptionKey": {
                "sourceVault": {
                  "id": "/subscriptions/test-source-guid/resourceGroups/testrg/providers/Microsoft.KeyVault/vau
                },
                "secretUrl": "https://test-disk.vault.azure.net/secrets/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/xxxxx-
              },
              "keyEncryptionKey": {
                "sourceVault": {
                  "id": "/subscriptions/test-key-guid/resourceGroups/testrg/providers/Microsoft.KeyVault/vaults/
                },
                "keyUrl": "https://test-key.vault.azure.net/secrets/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/xxxxx-xxxx-
              }
            },
            "image": {
                "uri": ""
            },
            "managedDisk": {
                "id": "/subscriptions/xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/resourceGroups/macikgo-test-may-23/provi
                "storageAccountType": "StandardSSD_LRS"
            },
            "name": "exampleosdiskname",
            "osType": "Windows",
            "vhd": {
                "uri": ""
            },
            "writeAcceleratorEnabled": "false"
        },
        "resourceDisk": {
            "size": "4096"
        }
    },
    "subscriptionId": "xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx",
    "tags": "baz:bash;foo:bar",
    "version": "15.05.22",
    "virtualMachineScaleSet": {
      "id": "/subscriptions/xxxxxxxx-xxxxx-xxx-xxx-xxxx/resourceGroups/resource-group-name/providers/Microsoft.C
    },
    "vmId": "02aab8a4-74ef-476e-8182-f6d2ba4166a6",
    "vmScaleSetName": "crpteste9vflji9",
    "vmSize": "Standard_A3",
    "zone": "3"
}
```

Azure has various sovereign clouds like Azure Government. Sometimes you need the Azure Environment to make some runtime decisions. The following sample shows you how you can achieve this behavior.

Request

- Windows
- Linux

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri "http://169.254.169.254/metadata/inst
```

Response

```
AzurePublicCloud
```

The cloud and the values of the Azure environment are listed here.

Cloud	Azure environment
All generally available global Azure regions	AzurePublicCloud
Azure Government	AzureUSGovernmentCloud
Microsoft Azure operated by 21Vianet	AzureChinaCloud
Azure Germany	AzureGermanCloud

Request

- Windows
- Linux

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri "http://169.254.169.254/metadata/inst
```

Response

```
{
 "interface": [
   {
     "ipv4": {
       "ipAddress": [
         {
           "privateIpAddress": "10.1.0.4",
           "publicIpAddress": "X.X.X.X"
         }
       ],
```

```
        "subnet": [
          {
            "address": "10.1.0.0",
            "prefix": "24"
          }
        ]
      },
      "ipv6": {
        "ipAddress": [{
          "privateIpAddress": "b4bc:8fce:f33b:4990:cced:d94e:ab4f:6ea0"
        }]
      },
      "macAddress": "000D3AF806EC"
    }
  ]
}
```


- Windows
- Linux

```
Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri "http://169.254.169.254/metadata/inst
```

Note

- If you're looking to retrieve IMDS information for **Standard** SKU Public IP address, review Load Balancer Metadata API for more infomration.

IMDS helps to provide guarantees that the data provided is coming from Azure. Microsoft signs part of this information, so you can confirm that an image in Azure Marketplace is the one you're running on Azure.

```
GET /metadata/attested/document
```

Name	Required/Optional	Description
`api-version`	Required	The version used to service the request.
`nonce`	Optional	A 10-digit string that serves as a cryptographic nonce. If no value is provided, IMDS uses the current UTC timestamp.


```
 {
    "encoding":"pkcs7",
    "signature":"MIIEEgYJKoZIhvcNAQcCoIIEAzCCA/8CAQExDzANBgkqhkiG9w0BAQsFADCBugYJKoZIhvcNAQcBoIGsBIGpeyJub25jZSI
}
```

The signature blob is a pkcs7-signed version of document. It contains the certificate used for signing along with certain VM-specific details.

For VMs created by using Azure Resource Manager, the document includes `vmId`, `sku`, `nonce`, `subscriptionId`, `timeStamp` for creation and expiry of the document, and the plan information about the image. The plan information is only populated for Azure Marketplace images.

For VMs created by using the classic deployment model, only the `vmId` and `subscriptionId` are guaranteed to be populated. You can extract the certificate from the response, and use it to confirm that the response is valid and is coming from Azure.

The decoded document contains the following fields:

Data	Description	Version introduced
`licenseType`	Type of license for Azure Hybrid Benefit. This is only present for AHB-enabled VMs.	2020-09-01
`nonce`	A string that can be optionally provided with the request. If no `nonce` was supplied, the current Coordinated Universal Time timestamp is used.	2018-10-01
`plan`	The Azure Marketplace Image plan. Contains the plan ID (name), product image or offer (product), and publisher ID (publisher).	2018-10-01
`timestamp.createdOn`	The UTC timestamp for when the signed document was created	2018-20-01
`timestamp.expiresOn`	The UTC timestamp for when the signed document expires	2018-10-01
`vmId`	Unique identifier for the VM	2018-10-01
`subscriptionId`	Azure subscription for the Virtual Machine	2019-04-30
`sku`	Specific SKU for the VM image (correlates to `compute/sku` property from the Instance Metadata endpoint [`/metadata/instance`])	2019-11-01

Note

For Classic (non-Azure Resource Manager) VMs, only the vmId is guaranteed to be populated.

Example document:

```
{
  "nonce":"20201130-211924",
  "plan":{
```

```
        "name":"planName",
        "product":"planProduct",
        "publisher":"planPublisher"
    },
    "sku":"Windows-Server-2012-R2-Datacenter",
    "subscriptionId":"aaaa0a0a-bb1b-cc2c-dd3d-eeeeee4e4e4e",
    "timeStamp":{
        "createdOn":"11/30/20 21:19:19 -0000",
        "expiresOn":"11/30/20 21:19:24 -0000"
    },
    "vmId":"02aab8a4-74ef-476e-8182-f6d2ba4166a6"
}
```


When validating the signature, you should confirm that the signature was created with a certificate from Azure. This is done by validating the certificate Subject Alternative Name (SAN).

Example SAN `DNS Name=eastus.metadata.azure.com, DNS Name=metadata.azure.com`

Note

The domain for the public cloud and each sovereign cloud will be different.

Cloud	Domain in SAN
All generally available global Azure regions	*.metadata.azure.com
Azure Government	*.metadata.azure.us
Azure operated by 21Vianet	*.metadata.azure.cn
Azure Germany	*.metadata.microsoftazure.de

Note

The certificates might not have an exact match for the domain. For this reason, the certification validation should accept any subdomain (for example, in public cloud general availability regions accept `*.metadata.azure.com`).

We don't recommend certificate pinning for intermediate certs. For further guidance, see Certificate pinning - Certificate pinning and Azure services. Please note that the Azure Instance Metadata Service will NOT offer notifications for future Certificate Authority changes. Instead, you must follow the centralized Azure Certificate Authority details article for all future updates.

Vendors in Azure Marketplace want to ensure that their software is licensed to run only in Azure. If someone copies the VHD to an on-premises environment, the vendor needs to be able to detect that. Through IMDS, these vendors can get signed data that guarantees response only from Azure.

Note

This sample requires the jq utility to be installed.

Validation

- Windows
- Linux

```
 # Get the signature
$attestedDoc = Invoke-RestMethod -Headers @{"Metadata"="true"} -Method GET -NoProxy -Uri http://169.254.169.254/
# Decode the signature
$signature = [System.Convert]::FromBase64String($attestedDoc.signature)
```

Verify that the signature is from Microsoft Azure and checks the certificate chain for errors.

```
 # Get certificate chain
$cert = [System.Security.Cryptography.X509Certificates.X509Certificate2]($signature)
$chain = New-Object -TypeName System.Security.Cryptography.X509Certificates.X509Chain
$chain.Build($cert)
# Print the Subject of each certificate in the chain
foreach($element in $chain.ChainElements)
{
    Write-Host $element.Certificate.Subject
}

# Get the content of the signed document
Add-Type -AssemblyName System.Security
$signedCms = New-Object -TypeName System.Security.Cryptography.Pkcs.SignedCms
$signedCms.Decode($signature);
$content = [System.Text.Encoding]::UTF8.GetString($signedCms.ContentInfo.Content)
Write-Host "Attested data: " $content
$json = $content | ConvertFrom-Json
# Do additional validation here
```

The `nonce` in the signed document can be compared if you provided a `nonce` parameter in the initial request.

A managed identity, assigned by the system, can be enabled on the VM. You can also assign one or more user-assigned managed identities to the VM. You can then request tokens for managed identities from IMDS. Use these tokens to authenticate with other Azure services, such as Azure Key Vault.

For detailed steps to enable this feature, see Acquire an access token.

When you place virtual machine or virtual machine set instances behind an Azure Standard Load Balancer, you can use IMDS to retrieve metadata related to the load balancer and the instances. For more information, see Retrieve load balancer information.

You can obtain the status of the scheduled events by using IMDS. Then the user can specify a set of actions to run upon these events. For more information, see Scheduled events for Linux or Scheduled events for Windows.

The following table lists samples of calling IMDS by using different languages inside the VM:

Language	Example
Bash	https://github.com/Microsoft/azureimds/blob/master/IMDSSample.sh
C#	https://github.com/Microsoft/azureimds/blob/master/IMDSSample.cs
Go	https://github.com/Microsoft/azureimds/blob/master/imdssample.go
Java	https://github.com/Microsoft/azureimds/blob/master/imdssample.java
NodeJS	https://github.com/Microsoft/azureimds/blob/master/IMDSSample.js
Perl	https://github.com/Microsoft/azureimds/blob/master/IMDSSample.pl
PowerShell	https://github.com/Microsoft/azureimds/blob/master/IMDSSample.ps1
Puppet	https://github.com/keirans/azuremetadata
Python	https://github.com/Microsoft/azureimds/blob/master/IMDSSample.py
Ruby	https://github.com/Microsoft/azureimds/blob/master/IMDSSample.rb

If there's a data element not found or a malformed request, the Instance Metadata Service returns standard HTTP errors. For example:

HTTP status code	Reason
`200 OK`	The request was successful.
`400 Bad Request`	Missing `Metadata: true` header or missing parameter `format=json` when querying a leaf node
`404 Not Found`	The requested element doesn't exist
`405 Method Not Allowed`	The HTTP method (verb) isn't supported on the endpoint.
`410 Gone`	Retry after some time for a max of 70 seconds
`429 Too Many Requests`	API Rate Limits have been exceeded
`500 Service Error`	Retry after some time

- **I'm getting the error `400 Bad Request, Required metadata header not specified`. What does this mean?**
    - IMDS requires the header `Metadata: true` to be passed in the request. Passing this header in the REST call allows access to IMDS.
- **Why am I not getting compute information for my VM?**
    - Currently, IMDS only supports instances created with Azure Resource Manager.
- **I created my VM through Azure Resource Manager some time ago. Why am I not seeing compute metadata information?**
    - If you created your VM after September 2016, add a tag to start seeing compute metadata. If you created your VM before September 2016, add or remove extensions or data disks to the VM instance to refresh metadata.
- **Is user data the same as custom data?**
    - User data offers the similar functionality to custom data, allowing you to pass your own metadata to the VM instance. The difference is, user data is retrieved through IMDS, and is persistent throughout the lifetime of the VM instance. Existing custom data feature will continue to work as described in this article. However you can only get custom data through local system folder, not through IMDS.
- **Why am I not seeing all data populated for a new version?**
    - If you created your VM after September 2016, add a tag to start seeing compute metadata. If you created your VM before September 2016, add or remove extensions or data disks to the VM instance to refresh metadata.
- **Why am I getting the error `500 Internal Server Error` or `410 Resource Gone`?**
    - Retry your request. For more information, see Transient fault handling. If the problem persists, create a support issue in the Azure portal for the VM.
- **Would this work for scale set instances?**
    - Yes, IMDS is available for scale set instances.
- **I updated my tags in my scale sets, but they don't appear in the instances (unlike single instance VMs). Am I doing something wrong?**
    - Currently tags for scale sets only show to the VM on a reboot, reimage, or disk change to the instance.
- **Why am I'm not seeing the SKU information for my VM in `instance/compute` details?**

- For custom images created from Azure Marketplace, Azure platform doesn't retain the SKU information for the custom image and the details for any VMs created from the custom image. This is by design and hence not surfaced in the VM `instance/compute` details.

- **Why is my request timed out (or failed to connect) for my call to the service?**

  - Metadata calls must be made from the primary IP address assigned to the primary network card of the VM. Additionally, if you've changed your routes, there must be a route for the 169.254.169.254/32 address in your VM's local routing table.

    - Windows
    - Linux

    1. Dump your local routing table and look for the IMDS entry. For example:

       ```
       route print
       ```

       ```
       IPv4 Route Table
       ===========================================================================
       Active Routes:
       Network Destination        Netmask          Gateway       Interface  Metric
               0.0.0.0          0.0.0.0      172.16.69.1      172.16.69.7     10
               127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
               127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
       127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
           168.63.129.16  255.255.255.255      172.16.69.1      172.16.69.7     11
       169.254.169.254  255.255.255.255      172.16.69.1      172.16.69.7     11
       ... (continues) ...
       ```

    2. Verify that a route exists for `169.254.169.254`, and note the corresponding network interface (for example, `172.16.69.7`).

    3. Dump the interface configuration and find the interface that corresponds to the one referenced in the routing table, noting the MAC (physical) address.

       ```
       ipconfig /all
       ```

       ```
       ... (continues) ...
       Ethernet adapter Ethernet:

       Connection-specific DNS Suffix  . : xic3mnxjiefupcwr1mcs1rjiqa.cx.internal.cloudapp.net
       Description . . . . . . . . . . . : Microsoft Hyper-V Network Adapter
       Physical Address. . . . . . . . . : 00-0D-3A-E5-1C-C0
       DHCP Enabled. . . . . . . . . . . : Yes
       ```

```
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::3166:ce5a:2bd5:a6d1%3(Preferred)
IPv4 Address. . . . . . . . . . . : 172.16.69.7(Preferred)
Subnet Mask . . . . . . . . . . . : 255.255.255.0
... (continues) ...
```

4. Confirm that the interface corresponds to the VM's primary NIC and primary IP. You can find the primary NIC and IP by looking at the network configuration in the Azure portal, or by looking it up with the Azure CLI. Note the private IPs (and the MAC address if you're using the CLI). Here's a PowerShell CLI example:

```
 $ResourceGroup = '<Resource_Group>'
$VmName = '<VM_Name>'
$NicNames = az vm nic list --resource-group $ResourceGroup --vm-name $VmName | ConvertFrom
foreach($NicName in $NicNames)
{
    $Nic = az vm nic show --resource-group $ResourceGroup --vm-name $VmName --nic $NicName
    Write-Host $NicName, $Nic.primary, $Nic.macAddress
}
```

```
 wintest767 True 00-0D-3A-E5-1C-C0
```

5. If they don't match, update the routing table so that the primary NIC and IP are targeted.

- **Fail over clustering in Windows Server**
  - When you're querying IMDS with failover clustering, it's sometimes necessary to add a route to the routing table. Here's how:
    1. Open a command prompt with administrator privileges.
    2. Run the following command, and note the address of the Interface for Network Destination (`0.0.0.0`) in the IPv4 Route Table.

```
route print
```

Note

The following example output is from a Windows Server VM with failover cluster enabled. For simplicity, the output contains only the IPv4 Route Table.

```
 IPv4 Route Table
===========================================================================
Active Routes:
```

```
Network Destination        Netmask          Gateway       Interface  Metric
        0.0.0.0          0.0.0.0        10.0.1.1        10.0.1.10    266
        10.0.1.0  255.255.255.192         On-link         10.0.1.10    266
        10.0.1.10  255.255.255.255         On-link         10.0.1.10    266
        10.0.1.15  255.255.255.255         On-link         10.0.1.10    266
        10.0.1.63  255.255.255.255         On-link         10.0.1.10    266
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
    169.254.0.0      255.255.0.0         On-link     169.254.1.156    271
    169.254.1.156  255.255.255.255         On-link     169.254.1.156    271
169.254.255.255  255.255.255.255         On-link     169.254.1.156    271
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link     169.254.1.156    271
255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
255.255.255.255  255.255.255.255         On-link     169.254.1.156    271
255.255.255.255  255.255.255.255         On-link         10.0.1.10    266
```

Run the following command and use the address of the Interface for Network Destination (`0.0.0.0`), which is (`10.0.1.10`) in this example.

```
route add 169.254.169.254/32 10.0.1.10 metric 1 -p
```

If you aren't able to get a metadata response after multiple attempts, you can create a support issue in the Azure portal.

You can provide product feedback and ideas to our user feedback channel under Virtual Machines > Instance Metadata Service here

- Acquire an access token for the VM
- Scheduled events for Linux
- Scheduled events for Windows

Source: https://docs.microsoft.com/en-us/azure/virtual-machines/windows/instance-metadata-service?tabs=windows