Connectivity to Azure Storage

Last updated by | Vitor Tomaz | Mar 30, 2023 at 10:11 AM PDT

Contents

Connectivity to Azure Storage

At a high level, Azure SQL Managed Instance communicates with Azure Storage for various purposes, including but not limited to, Backup and Restore, Auditing, Replication, Bulk Insert, Polybase and so on. Lets look at what is needed to be able to set up connectivity between Managed Instance and Azure Storage.

Options for making a network connection to Azure Storage

Using public network access (public endpoint) on Storage, enabled from all networks

Requirements:

- NSG on Managed Instance subnet need to be able to reach Storage public endpoint IP address.

Using public network access (public endpoint) on Storage, enabled from selected virtual networks and IP addresses

Selected virtual networks

Requirements:

- NSG on Managed Instance subnet need to be able to reach Storage public endpoint IP address.

- Need to have Microsoft.Storage added under Service Endpoints on Managed Instance VNet/subnet.
- Need to add SQL MI subnet on the 'Virtual networks' section on Azure Storage firewall.

Selected IP addresses

Requirements:

- NSG on Managed Instance subnet need to be able to reach Storage public endpoint IP address.
- Need to add SQL MI Management Endpoint (Tenant Ring IP) on the firewall IP address range on Azure Storage firewall.

Note: The option 'Allow Azure services on the trusted services list to access this storage account' is not supported for SQL MI.
Only a subset of Azure services are on the trusted services list.

Using Private Endpoint connections on Storage

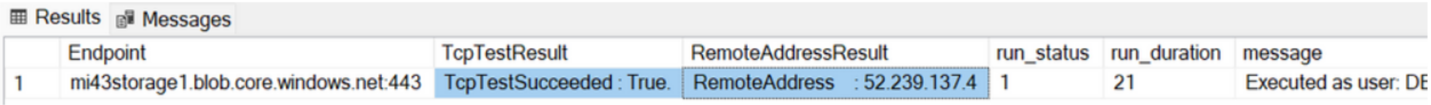
Requirements:

- DNS resolution on SQL MI VNet needs to resolve to Storage **private endpoint** IP address.
  - using Private DNS Zone
  - using Custom DNS
- NSG on Managed Instance subnet need to be able to reach Storage **private endpoint** IP address.

- Networking connectivity must exist between SQL MI subnet and subnet where Storage private endpoint is. For example, in case MI and storage private endpoint sit on different VNets, they need to be peered.

Troubleshoot network connection

We can test if SQL MI can reach the storage account from networking perspective. Before that, it's crucial that you can understand what the results of that test mean:



	Endpoint	TcpTestResult	RemoteAddressResult	run_status	run_duration	message
1	mi43storage1.blob.core.windows.net:443	TcpTestSucceeded : True.	RemoteAddress : 52.239.137.4	1	21	Executed as user: DE

Check if the resolved IP Address is the one you expect it to be (either storage public endpoint or storage private endpoint). Issues with DNS configuration on SQL MI VNet, especially when customer is keen on using private endpoint, are common.

The result for TcpTestSucceeded should be **TcpTestSucceeded : True.** This means that networking connectivity exist, endpoint is reachable from networking perspective. **Note** that *TcpTestSucceeded : True* doesn't necessarily mean you can access the storage account yet, you may still be blocked by the firewall, authentication/authorization, or any other configuration on storage side. You can reach the door, but you may not be allowed in.

If result is *TcpTestSucceeded : False*, it means there's some networking misconfiguration. Customer needs to engage their networking team, help them review all the requirements mentioned above. If you need help, you should reach out to CSS networking (not MI PG).

See How to test TCP connectivity from a SQL Managed Instance for step by step. The article can be shared with the customer.

Options to authenticate/authorize access to Azure Storage

Using SAS

Requirements on Azure Storage:

- **Allow storage account key access** needs to be **Enabled** (under Settings/Configuration).

Tips:

- When using `CREATE CREDENTIAL`, the `WITH IDENTITY` need to have the keyword `SHARED ACCESS SIGNATURE` and the `SECRET` needs to be the SAS token.
- Ensure that `?` is removed from the beginning of SAS token.
- `se` (expiry date) - Ensure that this property is set to some value in the future and it valid at the time of operation performed (note that this is UTC time).
- `st` (start date) - Ensure that this property is set to some time in the past (note that this is UTC time).
- `sp` (permission) - Ensure that this property should allow reading/writing (depending on the operation) the file on the storage account.
- `sip` (ip range) - This parameter specifies an IP address or a range of IP addresses from which to accept requests. When a range is specified, keep in mind that the range is inclusive. Only IPv4 addresses are supported. Remove this parameter if it's present in the SAS token.
- `srt` (allowed resource type) - Ensure that this property contains `c` (Container) to allow access to container.

See what different parameters mean on the SAS token.

- If **Allow Blob public access** is **Disabled**, it disables all public access to an Azure Resource Manager storage account, regardless of the public access setting for an individual container. Check Anonymous public read access to blob data for ARM deployments and [Anonymous public read access to blob data for classic deployments for the consequences for having this switch Disabled.
- SAS needs to be generated at Storage Account level. If SAS is generated at container level, can cause 1 container to work while authorization to others might fail.

Using Managed Identity

Managed identity is a feature of Azure Active Directory (Azure AD) that provides instances of Azure services, such as Azure SQL Managed Instance, with an automatically managed identity in Azure AD, the system-assigned managed identity.

You can use this identity to authorize requests for data access to other Azure resources, including storage accounts. Services such as Azure SQL Managed Instance have a system assigned managed identity, and can also have one or more user-assigned managed identities.

Depending on the feature (backup/Restore, Bulk Insert, Polybase, etc.), check to see if Managed Identity is a supported scenario to authorize the requests.

To be able to connect to storage with managed identity, in the Azure portal, on the Access Control (IAM) pane of a storage account, select Add role assignment, select the Storage Blob Data Contributor built-in Azure role-based access control (RBAC) role. This provides read/write access to the managed identity for the necessary Azure Blob Storage containers. On the next page, for Assign access to, select Managed identity. Choose Select members and then, in the Managed identity dropdown list, select the appropriate managed identity.

Requirements on SQL MI:

- When using `CREATE CREDENTIAL`, the `WITH IDENTITY` need to have the keyword `Managed Identity` (no need to specify the `Secret` here).

Requirements on Azure Storage:

- Add **Storage Bob Data Contributor** role on Storage to the MI identity (System or User).

Tips:

- If **Allow Blob public access** is **Disabled**, it disables all public access to an Azure Resource Manager storage account, regardless of the public access setting for an individual container. Check Anonymous public read access to blob data for ARM deployments and [Anonymous public read access to blob data for classic deployments for the consequences for having this switch Disabled.

Where are the settings?

Settings on Storage (Azure Portal)

On Azure Portal --> Storage --> Access Control (IAM)

- Add RBAC for Storage Blob Data Contributor role SQL MI identity (System or User).

On Azure Portal --> Storage --> Data storage --> Containers

- Check `Public access level` of container.

On Azure Portal --> Storage --> Security --> Networking

- Setting for Public network access under `Firewalls and virtual networks` (Enabled from all networks, Enabled from selected virtual networks and IP addresses, Disabled)
- Setting for `Virtual networks` to add MI VNet/Subnet.
- Setting for `Firewall` to add IP range.
- Setting for `Private endpoint connections` on Azure Storage.

On Azure Portal --> Storage --> Security --> Shared access signature

- Generate SAS key

On Azure Portal --> Storage --> Settings --> Configuration

- Allow Blob public access
- Allow storage account key access

Settings on Storage (ASC)

On ASC --> Storage --> Summary --> Properties

- Account kind (for example, Standard, General Purpose v1, General Purpose v2, etc)
- Create date, location (azure region)
- Allow Blob Public Access (Enabled vs Disabled)
- Allow Shared Key Access (Enabled vs Disabled)

On ASC --> Storage --> Summary --> Firewalls and Virtual Networks

- Allow public network access (enabled vs disabled)
- Allow access from (All networks vs selected networks)
- Virtual networks (list of VNets/Subnets added to the virtual networks on Storage)

On ASC --> Storage --> Summary --> Firewall

- List of IP range that has been allowed access

On ASC --> Storage --> Summary --> Private Endpoints

- Lists if Private endpoints exists on storage and details around it.

On ASC --> Storage --> Blob --> Search Blob/Container

- Select 'container' and search by container name to fetch container level properties such as `Public Access Type` (at container level), `Access Policies` and its expiration, etc.

On ASC --> Storage --> RBAC Roles (Storage)

- To check if `Storage Blob Data Contributor` RBAC has been added to Azure Storage.

Public Doc Reference

- How to test TCP connectivity from a SQL Managed Instance
- Use private endpoints for Azure Storage
- Different parameters on the SAS token.