Distribution Agent Unable to connect to onprem Subscriber

Last updated by | Holger Linke | Dec 21, 2022 at 12:40 AM PST

Contents

Issue

The Managed Instance is configured as Publisher and Distributor. The Subscriber is an on-premise SQL Server, created with a Push subscription for which the Distribution Agent is running on the Distributor. This is the supported and recommended configuration.

The Distribution Agent however is consistently unable to connect to the on-premise Subscriber server. Typical error messages might be:

> The process could not connect to Subscriber 'onpremSQLinstance'. (Source: MSSQL_REPL, Error number: MSSQL_REPL0)
> Named Pipes Provider: Could not open a connection to SQL Server [53]. (Source: MSSQLServer, Error number: 53)

> A network-related or instance-specific error has occurred while establishing a connection to SQL Server. Server is not found or not accessible. Check if instance name is correct and if SQL Server is configured to allow remote connections. For more information see SQL Server Books Online. (Source: MSSQLServer, Error number: 53)
> Query timeout expired, Failed Command: (Source: MSSQLServer, Error number: HYT00)

Cause 1: Name resolution failing

The Distribution Agent might be failing because the Managed Instance can't resolve the name of the on-premise SQL Server instance. With Push subscriptions, you need to make sure that the name resolution on MI is able to resolve the FQDN of the on-premise Subscriber.

The on-premise SQL Server is typically connected through ExpressRoute or point-to-site connectivity, thus has an endpoint in an Azure VNet. Replication needs DNS to get the private IP of the server in the virtual network from the server name. See Name resolution for resources in Azure virtual networks for the available DNS options.

Investigation: You can check that the MI can connect to the subscriber by creating a Linked Server on the MI to the Subscriber. If this works, you can also use the public FQDN of the on-premise server as Subscriber name when creating the Push subscription.

Mitigation: To make this work, you need either of the following:

1. Configure a private DNS zone - *the easiest and preferred solution*. The steps for this solution are described in the tutorial at Create a private DNS zone.

2. Configure a custom DNS solution for your VNET - *no longer recommended*. The steps are documented at Configure a custom DNS for Azure SQL Managed Instance.

3. Specify the Subscriber's IP address directly when creating the subscription, instead of the server name. This will work if the IP address is a static public IP address. You may need to open the inbound traffic on the on-premise firewall to allow the traffic from the Managed Instance. Find the IP address through Determine the management endpoint IP address. This IP address may change over time though. See the note below for an example for creating the subscription.

Note regarding #1 and #2: If you had added or changed the DNS configuration after the Managed Instance had been created, then MI might not have picked up that change yet. You would need to flush the DNS cache for MI first before it would be able to work with the new DNS server. The easiest way to flush the DNS cache would be to initiate a failover of the MI, or to scale the MI. Either failover or scaling implies a restart of the Managed Instance and reading fresh settings.

Note regarding #3: This option is not officially supported, but it seems to work in practice. To configure the subscription, you must use replication stored procedures instead of the SSMS UI - the UI attempts and fails on name resolution whereas the stored procedures don't.

Example for creating the Push subscription through SQL commands:

```
EXEC sp_addsubscription @publication = N'<Publication_Name>',
     @subscriber = N'<Subscriber_IP_Address>', -- e.g. '10.0.0.10' or '11.0.0.11,1433'
     @destination_db = N'<Subscribing_DB>',
     @subscription_type = N'Push'

EXEC sp_addpushsubscription_agent @publication = N'<Publication_Name>',
     @subscriber = N'<Subscriber_IP_Address>', -- same value as provided in the command above
     @subscriber_db = N'<Subscriber_Database>',
     @subscriber_security\_mode = 0,
     @subscriber_login = N'<SUBSCRIBER_SQL_USER>', @subscriber_password = N'<SUBSCRIBER_PASSWORD>',
     @job_login = N'<DISTRIBUTION_SQL_USER>', @job_password = N'<DISTRIBUTION_PASSWORD>'
```

Cause 2: Network not configured correctly

Review the Azure VNet configuration to make sure that the prerequisites are met:

- Ensure outgoing port 1433 is open on the MI subnet, and incoming port 1433 on the Subscriber. Currently transactional replication only allows to configure Subscribers if the Subscriber is using the default port 1433. If the customer is using a different port, e.g. a custom port for a Named Instance, then replication will not work.

- Confirm that the correct Subscriber DNS name was used when creating the subscription. Confirm that the correct IP address was used if option #3 from above had been used.
- Verify that the VNet peering is configured correctly (if the Subscriber connection goes over a peered VNet).

Cause 3: Connection blocked by on-premise Subscriber server

- If the on-premise Subscriber server has a firewall, make sure that connections to the port used by the local SQL instance (1433) are open for inbound connections.
- Ensure that the TCP/IP protocol is enabled on the Subscriber SQL Server instance. Confirm this in the SQL Server Configuration Manager (SQL Native Client Configuration -> Client Protocols) on the on-premise server.

More Information

If you configure a Pull subscription instead of Push, the Distribution Agent is executed on the on-premise SQL Server. Usually there is no connectivity from on-premise back to the Azure storage account, and thus the agent usually cannot download the snapshot files to the Subscriber. It would fail with an error similar to the following:

> Command attempted: \storageaccountname.file.core.windows.net \sharename\unc\publisher_publication_name\20190122055620\articlename_5.pre (Transaction sequence number: 0x00000CFF0000019000BA00000000, Command ID: 8)
> Error messages: The process could not read file '\storageaccountname.file.core.windows.net \sharename\unc\publisher_publication_name\20190122055620\articlename_5.pre' due to OS error 1326. (Source: MSSQL_REPL, Error number: MSSQL_REPL20024)
> The user name or password is incorrect. (Source: MSSQL_REPL, Error number: MSSQL_REPL1326)

Therefore, in this scenario, you can only use Push subscriptions, running all replication agents on the Distributor MI.

Public Doc Reference

- Tutorial: Configure transactional replication between Azure SQL Managed Instance and SQL Server

How good have you found this content?