# Replication Failure in Azure

If replication stopped working between 2 Microsoft SQL Servers in the Azure cloud suddenly and without reason, what could the reason be? One of the servers is an Azure Managed SQL Server and the other is a SQL server installation on an Azure Virtual Machine. The only evidence that replication ever existed is the "subscription" section on the Azure Managed SQL Server showing a subscription pointing to the Azure Virtual Machine. All of the jobs are missing for snapshots and transactions. In addition to this the distribution database is missing from both servers. Using all the information provided, think as long and hard as possible to come up with potential reasons this could have happened.

Below are a variety of reasons—some straightforward, some more subtle—that could explain why replication “disappeared” between the Azure Managed SQL Server and the SQL Server running on the Azure VM. Since you mentioned all replication jobs are missing, the distribution databases are gone, and only a phantom subscription remains, here are possible scenarios to consider:

1. **Replication was never fully supported or was configured in an unsupported scenario**
    - **Azure SQL Database** (platform as a service) supports replication only in a limited way (e.g., as a subscriber to a SQL Server publisher). It cannot act as a publisher or distributor.
    - **Azure SQL Managed Instance** can support replication both as a publisher and as a subscriber, but only under certain constraints (e.g., replication to an on-premises or Azure VM SQL Server).
    - If you attempted to use Azure SQL Database as distributor or publisher (instead of Azure SQL Managed Instance), the configuration may have never been fully valid and could have been removed automatically or during a version/maintenance upgrade.
2. **Accidental or intentional removal of replication components**
    - An administrator or automated script could have run a command (e.g., `sp_removedbreplication`, `sp_dropdistributiondb`, or the “Disable Publishing and

Distribution” wizard) on one or both servers. This would remove the distribution database and all associated jobs.

- The fact that the only remaining trace is the subscription entry suggests that the process to completely remove replication wasn’t executed cleanly or fully, leaving an orphaned subscription listing.

3. **Distribution database was on a different server or was dropped**
   - Sometimes the distribution database gets configured on a dedicated “distribution” server or a specific instance. If that server or instance was decommissioned or re-imaged, or if an admin dropped the distribution database, replication would vanish.
   - In a scenario where the distribution database was accidentally placed on the Azure SQL Managed Instance (or an Azure SQL Database that doesn’t truly support it), a subsequent change or update might have removed it when it detected an invalid configuration.
4. **Upgrade, patch, or migration changes**
   - If either the Azure VM or the Azure Managed SQL Server was migrated, upgraded, or re-provisioned (for example, switching Azure SQL tiers or VM images), the replication settings may not have carried over.
   - Azure maintenance cycles sometimes involve underlying hardware or OS-level re-deployments. If replication was configured in a non-standard way (or used ephemeral storage), that could explain a silent removal.
5. **Security or networking changes**
   - Firewalls, NSGs (Network Security Groups), or changed endpoints could cause replication jobs to fail or time out repeatedly. If there was an automated process that removed “dead” replication configurations, it might explain the missing objects.
   - Although this would usually leave the distribution database behind, it’s worth noting if an admin or script cleaned things up after repeated failures.
6. **SQL Server Agent jobs were removed or disabled**
   - Replication depends on multiple Agent jobs (Log Reader Agent, Distribution Agent, Snapshot Agent). If something (script, policy, or manual action) removed or disabled SQL Server Agent jobs on the VM, you could lose all replication functionality.
   - Over time, if replication was fully disabled, the distribution database might also have been removed—especially if a DBA noticed “unused” databases and decided to drop it.

7. **Replication was never actually functional beyond a test**
    - It's possible the replication was configured in a test or proof-of-concept mode, the environment was torn down or partially cleaned up, and only the subscription reference remains because it wasn't properly dropped. In other words, it might never have been truly production-ready or fully configured from the start.
8. **Azure policy or compliance script removed the configuration**
    - Some organizations have automation or policies that scan for unsupported or unapproved configurations. If an automated policy flagged the replication setup (e.g., if distribution was set up incorrectly on an Azure SQL Database), the system might have removed it.
9. **Unexpected bug or corruption**
    - Although less common, a corruption in the msdb or distribution database on the VM could remove references to replication jobs. In a worst-case scenario, if the distribution database was corrupt, a DBA might have forced a drop without fully removing the subscription.
    - Upgrades or patch scripts that encountered unexpected states could also remove replication metadata.
10. **Region failover or DR event**

- If you're using geo-replication or automated failover in Azure, a failover event that targeted a region or a different resource might have left replication behind. Perhaps after failover, the environment was never re-synchronized for replication, and a cleanup operation removed leftover replication objects.

---

## Next Steps to Diagnose

1. *Check the MSrepl_ system tables** in the user and master databases on both the VM and Managed Instance (if possible). See if there's any leftover replication metadata in system tables.
2. **Look for the distribution database** on both servers more thoroughly (e.g., in SSMS or via T-SQL). Confirm it truly does not exist anywhere else.

3. **Review SQL Server logs, Agent logs, and Azure logs** for any mention of replication being removed or an error event that might have triggered an automatic cleanup.
4. **Confirm the actual server types** (Azure SQL Database vs. Azure SQL Managed Instance). If it's an Azure SQL Managed Instance, it can act as publisher/distributor under certain conditions. If it's a plain Azure SQL Database, replication as a publisher/distributor is not supported.
5. **Check Azure resource deployments** to see if any new VM, managed instance, or re-deployments took place that might have impacted the environment.

---

## Conclusion

In many cases, the simplest explanation is that the replication was (intentionally or accidentally) dropped or never properly configured to begin with—leading to a scenario where the distribution DB and jobs vanished while a leftover subscription entry lingered. Other plausible causes include unsupported configurations on Azure SQL Database, partial or failed cleanup after an upgrade, or an admin-level action that removed distribution without fully clearing the subscription references.

By reviewing system tables, logs, and verifying the actual environment (Managed Instance vs. Database), you can usually pinpoint which of these scenarios occurred and take steps to reimplement replication—if it's still a requirement—in a fully supported and documented manner.

How can I check sql server logs on both servers for any logs that may exist for replication related jobs that may no longer exist?

Below are some methods you can use to check both SQL Server Agent logs and the main SQL error logs on each server (the Azure SQL VM and the Azure Managed Instance) for any references to replication or replication-related jobs—even if those jobs no longer exist.

---

# 1. On the SQL Server in the Azure VM

### A. Using SQL Server Management Studio (SSMS) to check Agent Job History

1. Open SSMS and connect to the SQL Server instance on your Azure VM.
2. Expand **SQL Server Agent**.
3. Right-click **Jobs** and select **View History** (or expand **Jobs** → right-click a specific job if you still see it).
4. This will show you recent job history.
    - Even if the replication jobs are removed, there might still be leftover history records in **msdb** (unless they were purged).

### B. Querying msdb system tables (in case the job entries still exist in the history)

From a new query window in SSMS:

```sql
USE msdb;
GO

-- Check if there is any leftover replication or distribution job history
SELECT j.name AS JobName,
       jh.run_status,
       jh.run_date,
       jh.run_time,
       jh.message
FROM dbo.sysjobhistory AS jh
INNER JOIN dbo.sysjobs AS j
  ON jh.job_id = j.job_id
WHERE j.name LIKE '%replication%'
   OR j.name LIKE '%distribution%'
   OR j.name LIKE '%snapshot%'
   OR j.name LIKE '%log reader%';
```

- This could turn up any historical entries (if they weren't fully purged).

### C. Checking the SQL Server Agent error log

1. In SSMS, expand **SQL Server Agent** → **Error Logs**.
2. Double-click the current or archived log to open and look for references to "replication," "snapshot agent," "distribution agent," etc.

  - You can also right-click → **View Agent Log** and then use the Find feature.

**D. Checking the main SQL Server error log**

1. In SSMS, expand **Management** → **SQL Server Logs**.
2. Double-click **SQL Server (Current)** or older logs.
3. Scroll or search for key terms (e.g., “Replication”, “Distribution Agent”, “Snapshot Agent”).
   - Alternatively, you can query them via T-SQL:

     ```sql
     EXEC xp_readerrorlog 0, 1, 'Replication';  -- 0 = current log, 1 = SQL Server error log
     EXEC xp_readerrorlog 0, 1, 'Distribution';
     EXEC xp_readerrorlog 0, 1, 'Snapshot';
     ```

   - Adjust the parameters if you want to read older logs. (e.g., `xp_readerrorlog 1, 1, 'Replication'` for the 1st archived log.)

**E. Checking the Windows Event Viewer (on the VM itself)**

- On the VM, open **Event Viewer** → **Windows Logs** → **Application** (and possibly **System**). Filter on:
  - **Source** = “MSSQLSERVER” or “SQLAgent$<InstanceName>”
  - Look for replication or job-related errors that might have been logged before the jobs disappeared.

---

## 2. On the Azure SQL Managed Instance

Azure SQL Managed Instance supports SQL Agent jobs similarly to a traditional SQL instance (unlike Azure SQL Database). However, some restrictions and differences apply. You can check logs and job history here as well:

**A. Check SQL Agent jobs/history within Managed Instance**

1. Connect to your Managed Instance in SSMS.
2. Expand **SQL Server Agent** → **Jobs** and **Job Activity Monitor** (if you have permissions).

3. Check for any references to replication jobs still present (you mentioned they appear missing, but there may be partial references in msdb's metadata).

### B. Query msdb tables on Managed Instance

```sql
USE msdb;
GO

SELECT j.name AS JobName,
       jh.run_status,
       jh.run_date,
       jh.run_time,
       jh.message
FROM dbo.sysjobhistory AS jh
INNER JOIN dbo.sysjobs AS j
  ON jh.job_id = j.job_id
WHERE j.name LIKE '%replication%'
   OR j.name LIKE '%distribution%'
   OR j.name LIKE '%snapshot%';
```

- If the jobs or their history records remain, you'll see them here.

### C. Check the SQL error log via T-SQL

While Managed Instance doesn't allow direct access to Windows event logs, you can still read the SQL error log with extended stored procedures (if you have permissions). For example:

```sql
EXEC xp_readerrorlog 0, 1, 'Replication';
```

- You might see references to replication setup, failures, or job removals.

### D. Azure Portal (Diagnostic settings / Auditing / Extended Events)

You can also configure or check **Diagnostic Settings** for the Managed Instance.

- In the **Azure portal** → your Managed Instance → **Monitoring** → **Diagnostic settings**, see if there is any historical logging or if you have configured "Send to Log Analytics / Event Hub / Storage Account."

- If replication was removed by an automated process, there might be a log or alert captured in these diagnostic channels.

---

## 3. If Jobs and Distribution DB Have Been Fully Removed

- If the replication jobs and distribution database have been completely dropped, **you might not see any direct references in msdb**. Even so, it's worth checking older logs (archived error logs) or any external monitoring solution (e.g., Log Analytics, third-party monitoring) to see if there's a record of who/what dropped the replication objects.
- Also verify if your environment has **automated cleanup scripts** or **policies** that might remove "unused" or "unsupported" replication objects from time to time.

---

### Quick Summary

1. **On the Azure VM**:
    - Check msdb job history tables (sysjobhistory, sysjobs).
    - Check both SQL Server Agent logs and main SQL Server error logs (via SSMS or xp_readerrorlog).
    - If needed, check Windows Event Viewer for MSSQL-related events.
2. **On the Azure Managed Instance**:
    - Check msdb job history tables.
    - Query the SQL error log with xp_readerrorlog (if accessible).
    - Look in Azure Portal → Diagnostic Settings for historical logs or alerts.

By combining these steps, you'll have the best chance of finding any references to replication jobs—successful or failed runs, possible removal notifications, or related errors—even if the jobs themselves are long gone.