

Log Shipping in SQL Server is a disaster recovery and high-availability solution that involves automatically sending transaction log backups from a primary server to one or more secondary servers. While Log Shipping is generally reliable, various issues can arise that impact its functionality. Here are some common issues and their detailed resolutions:

1. Backup Job Failure (Primary Server)

Issue: The log backup job fails on the primary server, preventing transaction log backups from being created.

Causes:

- Insufficient disk space on the backup location.
- Incorrect backup path configuration.
- Permissions issues on the backup folder.
- The database is in the Simple Recovery Model.

Resolution:

- **Disk Space:** Check the available disk space on the backup directory. Clear unnecessary files or move backups to a larger volume.
- **Backup Path:** Verify the backup path in the SQL Server Agent job configuration and ensure it's valid. If needed, correct the path.
- **Permissions:** Ensure the SQL Server service account has write permissions to the backup folder.
- **Recovery Model:** Ensure the database is in **Full** or **Bulk-Logged Recovery Model**. Log Shipping does not work with the **Simple Recovery Model**. Change the recovery model using:

```
ALTER DATABASE [DatabaseName] SET RECOVERY FULL;
```

Verification:

- Manually run the backup job from SQL Server Agent.
- Check the error logs for detailed error messages, including SQL Server Agent and SQL Server error logs.

2. Copy Job Failure (Secondary Server)

Issue: The copy job fails on the secondary server, and transaction log backups are not being copied from the primary server.

Causes:

- Network connectivity issues between the primary and secondary servers.
- Insufficient permissions on the backup folder (secondary server account can't access the backup).
- Incorrectly configured copy job (path mismatch).

Resolution:

- **Network Connectivity:** Test the network connection between the primary and secondary servers using `ping`, `telnet`, or a similar tool. Ensure proper network configuration and firewall settings.
- **Permissions:** Ensure that the account running the SQL Server Agent service on the secondary server has read access to the backup folder on the primary server.
- **Backup Path:** Check the SQL Server Agent job for the copy step and ensure the path to the log backups on the primary server is correct. Adjust if necessary.

Verification:

- Manually trigger the copy job from the SQL Server Agent and verify that the transaction logs are being copied successfully.
- Check the SQL Server Agent job history for any specific error messages.

3. Restore Job Failure (Secondary Server)

Issue: The restore job fails on the secondary server, leading to lag in the restoration of transaction logs.

Causes:

- Incorrect recovery mode for the database on the secondary server (should be in **RESTORING** state).
- Missing transaction logs in the backup chain.
- SQL Server Agent service not running.

Resolution:

- **Database in Restoring State:** Ensure the secondary database is in **RESTORING** mode. If it's in another state (e.g., **Online**), reconfigure it using:

RESTORE DATABASE [DatabaseName] WITH NORECOVERY;

- **Missing Logs:** Check if any transaction log backups are missing. If so, you may need to reinitialize Log Shipping by restoring a new full backup with **NORECOVERY** on the secondary server and restarting the Log Shipping process.
- **SQL Server Agent:** Ensure the SQL Server Agent service is running on both the primary and secondary servers.

Verification:

- Manually run the restore job in SQL Server Agent.
- Check the job history and the secondary server's error logs for detailed error messages.

4. Transaction Log Growth on Primary Server

Issue: Transaction logs on the primary server grow excessively due to delayed log truncation caused by Log Shipping.

Causes:

- Log truncation is delayed because the transaction logs haven't been backed up, copied, or restored on the secondary server.
- Network latency between the primary and secondary servers.

Resolution:

- **Check Backup Job:** Ensure that the backup job on the primary server is running successfully. If it's failing, resolve the issue (see "Backup Job Failure").
- **Check Copy and Restore Jobs:** Ensure that the transaction logs are being copied and restored in a timely manner on the secondary server. Adjust job schedules if necessary to run more frequently.
- **Network Latency:** If there's a delay in copying logs due to network issues, consider increasing the job frequency or using compression for log backups to reduce file sizes.

Verification:

- Monitor the transaction log size on the primary server using the **DBCC SQLPERF** command:

DBCC SQLPERF(LOGSPACE);

5. Out-of-Sync Databases Between Primary and Secondary

Issue: The secondary database lags behind the primary database, causing the data on the secondary to be out-of-sync.

Causes:

- Transaction log backups are not being created frequently enough.
- The restore job on the secondary server is delayed or failing.

Resolution:

- **Backup Frequency:** Adjust the transaction log backup frequency on the primary server to create backups more frequently.
- **Restore Job:** Ensure the restore job is running frequently enough to minimize the lag. If the restore job is failing, troubleshoot it (see "Restore Job Failure").
- **Transaction Log Chain:** Ensure there are no missing transaction log backups that could disrupt the log shipping chain.

Verification:

- Check the Log Shipping status report in SQL Server Management Studio (SSMS) under "Log Shipping Monitor" to review the backup, copy, and restore latencies.
- Monitor the last restored log file on the secondary server.

6. Log Shipping Monitoring Alerts

Issue: Log Shipping alerts are triggered frequently, indicating that the backup, copy, or restore operations are out-of-sync.

Causes:

- Log Shipping thresholds for backup, copy, or restore jobs are too tight.
- There are frequent network issues or delays in log shipping operations.

Resolution:

- **Threshold Settings:** Increase the threshold for alerting on latency in the Log Shipping configuration. This can be done by increasing the alert thresholds in SSMS under the "Log Shipping Monitor" settings.
- **Network Issues:** Review the network configuration to ensure consistent connectivity between the primary and secondary servers. If the network is unreliable, consider scheduling jobs less frequently or using compression for the transaction log backups.

Verification:

- Review the SQL Server Log Shipping status and alert history in SSMS.
- Adjust the alert settings and monitor for any further alerts.

7. Transaction Log Corruption

Issue: A transaction log backup file is corrupted, preventing successful copying or restoring on the secondary server.

Causes:

- Disk corruption or file system errors on the backup storage.
- Network issues causing partial or corrupted log file transfers.

Resolution:

- **Backup Integrity:** Check the integrity of the transaction log backups using the `RESTORE VERIFYONLY` command:

```
RESTORE VERIFYONLY FROM DISK = 'Path\To\Backup.bak';
```

- If the backup is corrupt, discard it and take a new transaction log backup.
- **Network Issues:** If the backup file was corrupted during transfer, retransfer the file or take a new backup. Ensure that the network is stable and reliable.

Verification:

- After resolving corruption, reinitialize the affected log shipping steps and ensure the log backup, copy, and restore processes are functioning correctly.

Best Practices for Log Shipping:

1. **Monitor Regularly:** Set up monitoring and alerting for Log Shipping using SQL Server Management Studio (SSMS) or custom scripts.
2. **Automate Failover Testing:** Periodically test failover to the secondary server to ensure Log Shipping works in a real disaster recovery scenario.
3. **Optimize Backup Compression:** Enable backup compression to reduce transaction log backup sizes and speed up transfer/copy operations.
4. **Use Network File System (NFS):** Store transaction log backups on a shared network location accessible to both the primary and secondary servers to simplify the copy process.
5. **Adjust Job Frequencies:** Tune the frequency of backup, copy, and restore jobs to match your recovery objectives (RPO/RTO).

By following these resolutions and best practices, you can troubleshoot and prevent common Log Shipping issues in SQL Server.