Log Replay Service

Last updated by | Vitor Tomaz | Nov 16, 2022 at 5:41 AM PST

Contents

Self-help content presented in Azure Portal

(This content was shown to the customer during case submission. It's also visible on 'Diagnose and solve problems' blade.)

Log Replay Service (LRS) is a free of charge cloud service enabled for Azure SQL Managed Instance based on SQL Server log-shipping technology. Azure Database Migration Service and LRS use the same underlying migration technology and APIs. LRS further enables complex custom migrations and hybrid architectures between on-premises SQL Server and SQL Managed Instance.

Why use Log Replay Service

When you can't use Azure Database Migration Service for migration, you can use LRS directly with PowerShell, Azure CLI cmdlets, or APIs to manually build and orchestrate database migrations to SQL Managed Instance.

While we recommend automating the migration of databases from SQL Server to SQL Managed Instance by using Database Migration Service, consider using LRS to orchestrate migrations when Database Migration Service doesn't fully support your scenarios.

LRS is the only method to restore differential backups on managed instance. It isn't possible to manually restore differential backups on managed instance, nor to manually set the NORECOVERY mode using T-SQL.

Consider these scenarios of when to use Log Replay Service.

How to migrate using LRS

Migration consists of making database backups on SQL Server with `CHECKSUM` enabled, and copying backup files to Azure Blob Storage. Full, log, and differential backups are supported. LRS cloud service is used to restore backup files from Azure Blob Storage to SQL Managed Instance. Blob Storage serves as an intermediary storage between SQL Server and SQL Managed Instance.

LRS monitors Blob Storage for any new differential or log backups added after the full backup has been restored. LRS then automatically restores these new files. You can use the service to monitor the progress of backup files being restored to SQL Managed Instance, and stop the process if necessary.

LRS doesn't require a specific naming convention for backup files. It scans all files placed on Azure Blob Storage and constructs the backup chain from reading the file headers only. Databases are in a **restoring** state during the migration process. Databases are restored in NORECOVERY mode, so they can't be used for read or write workloads until the migration process completes.

If you're migrating several databases, you need to:

- Place backup files for each database in a separate folder on Azure Blob Storage in a flat-file structure. For example, use separate database folders: `blobcontainer/database1/files` , `blobcontainer/database2/files` , etc.
- Don't use nested folders inside database folders as this structure isn't supported. For example, don't use subfolders: `blobcontainer/database1/subfolder/files` .
- Start LRS separately for each database.
- Specify different URI paths to separate database folders on Azure Blob Storage.

You can start LRS in either autocomplete or continuous mode.

- Autocomplete mode needs to be used only when all backup chain files are available in advance. This mode is recommended for passive workloads for which no data catch-up is required.
- Continuous mode migration needs to be used when you don't have the entire backup chain in advance, and when you plan to add new backup files once the migration is in progress. This mode is recommended for active workloads for which data catch-up is required.

See the migration workflow and the individual steps outlined to migrate database to managed instance using LRS.

Permissions needed

Running LRS through the provided clients requires one of the following Azure roles:

- Subscription Owner role
- SQL Managed Instance Contributor role
- Custom role with the following permission: `Microsoft.Sql/managedInstances/databases/*`

Common errors and issues encountered when migrating using LRS

When initiating LRS, I get error `Resource not found`

Ensure that the database name, resource group name, and managed instance name are spelled correctly. The managed instance name just needs to be the server name, not the full FQDN.

When initiating LRS, I get error `File not found`

To migrate multiple databases using the same Azure Blob Storage container, place all backup files of an individual database into a separate folder inside the container. Use flat-file structure for each database folder, as nested folders aren't supported.

When migrating multiple databases, LRS must be started separately for each database pointing to the full URI path of Azure Blob storage container and the individual database folder: `https://<mystorageaccountname>.blob.core.windows.net/<containername>/<databasefolder>`

Do not list the backup filename in the URI path.

I can't connect to or access the target database

Databases are in a **restoring** state during the migration process. Databases are restored in NORECOVERY mode, so they can't be used for read or write workloads until the migration process completes. The final cutover step makes the database come online and available for read and write access on SQL Managed Instance.

New backup file is not picked up by the service

When started in autocomplete mode, LRS restores all backups until the specified last backup file. All backup files must be uploaded in advance, and it isn't possible to add any new backup files while migration is in progress. This mode is recommended for passive workloads for which no data catch-up is required.

Once LRS has been started in continuous mode, you'll be able to add new log and differential backups to Azure Blob Storage until the manual cutover. Make sure you have the `WITH DIFFERENTIAL` in the backup command when taking differential backup on source. Once manual cutover has been initiated, the migration process is finalized and no additional differential files can be added, nor restored.

Migration is taking longer than expected

Enable backup compression to help the network transfer speeds.

System updates on managed instance will take precedence over database migrations in progress. All pending LRS migrations in case of a system update on Managed Instance will be suspended and resumed once the update has been applied. This system behavior might prolong migration time, especially in cases of large databases. To achieve a predictable time of database migrations, consider configuring maintenance window allowing scheduling of system updates at a specific day/time, and consider running and completing migration jobs outside of the scheduled maintenance window day/time.

Check additional best practices and recommendations for migrating large databases.

Resources

- Migrate databases from SQL Server to Azure SQL Managed Instance using Log Replay Service
- Steps to migrate
- Limitations of LRS
- Troubleshooting

How good have you found this content?

🙂 🙁