

# Hyperscale Reverse Migration

Last updated by | Subbu Kandhaswamy | Feb 9, 2022 at 4:00 PM PST

## Contents

- [What Hyperscale Reverse migration feature do?](#)
- [Mechanism \(Internal\)](#)
- [Business Scenarios / Reasons to reverse migrate](#)
- [Current Limitations](#)
  - [Prerequisites](#)
  - [Reason for the limitations](#)
- [Monitoring Migration](#)
- [Migrating using APIs/Tools](#)
  - [Migrating out of Hyperscale using T-SQL](#)
  - [Migrating out of Hyperscale using Azure CLI](#)
  - [Migrating out of Hyperscale using PowerShell](#)
- [Handling Customer Questions](#)
- [Architecture/benefits/limitations walk thru by PG](#)

## What Hyperscale Reverse migration feature do?

Currently, a database that was created in or moved to Hyperscale tier, cannot change its tier/SLO to any other non-Hyperscale Azure SQL DB service tier (GP, Business critical, Premium etc.,) by changing its tier through portal/SLO through an API. This is because of the fundamentally different file structure and storage architecture used by Hyperscale compared to other SQL Database tiers. Hyperscale reverse migration feature allows customers the opportunity to move a Hyperscale database to a General-Purpose tier for any reason (*with some limitations/restrictions*).

## Mechanism (Internal)

To move a Hyperscale database to a GP tier, we consolidate the Hyperscale database files into a single backup that can then be restored in the GP tier. When the user issues a command that would trigger this move out of Hyperscale, we begin taking a full backup of the database. When this is complete, we restore the backup in the target tier and then catch the new restore up to the latest (since backup started) from the Hyperscale database. At this time, we terminate all connections to the source HS database and allow the application to re-establish connections on the target (General-Purpose) database.

## Business Scenarios / Reasons to reverse migrate

Hyperscale is positioned as the default SQL Database tier for all kinds of SQL database data. Reverse Migrating back from Hyperscale may include both performance and functionality reasons that are yet to be fully resolved in Hyperscale tier. Some of the identified reasons includes missing functionalities and optimization of cost for businesses.

- **Missing functionality –**
  - In-Memory OLTP (Hekaton) - supported only in BC tier
  - Shrink database, GeoDR, .. etc. – (Planned)
- **Cost optimization-** GP tier is less expensive and may serve better for dev/test and other scenarios.
  - One known example is where the customer wants to make a copy of the production Hyperscale database and move it to the General-Purpose tier for testing/development.
  - This scenario is P2 and should be addressed in long term pricing/tier positioning of SQL DB.

## Current Limitations

This feature has always been thought of in terms of providing insurance policy for customers and never as regular service level objective change operation. These are the limitations currently in place on the reverse migration feature.

- Migration is supported from HS to GP tier only(provisioned/Serverless). Direct migrations to other SQL DB tiers are not supported. Such further migrations can be done from GP tier.
- Databases originally created as Hyperscale (non-migrated ones) are NOT eligible for reverse migration.
- Databases on Hyperscale tier with uninterrupted N(N=45) days of tenure (not migrated out within N days after originally migrated to HS) are ineligible for Reverse Migration.
- An Example of a valid scenario
  - BC/GP => HS (for less than N days) => GP.
- Examples of invalid scenarios
  - HS => HS\_copy (brand-new HS db) => GP.
  - HS => HS\_restore (brand-new HS db) => GP.
  - HS (brand-new HS db) => GP.
  - BC/GP => HS (for N days or more) => GP.
- Migration back to GP / HS within former incarnations' short term backup retention period will result in dropping former incarnations' backups / snapshots.
- This means that if you have moved from GP to Hyperscale and back to GP, the only backups available are the ones from the current GP incarnation and the immediate previous Hyperscale incarnation. Any previous tiers tried will not have backups available.

For example, if you migrate 5 times this way:

1. GP
2. => Hyperscale
3. => GP
4. => Hyperscale
5. => GP

- The only backups available will be from steps 4 and 5 of the timelines. Any backups from previous steps will be unavailable.

## Prerequisites

The following checks are done, and Reverse migration fails if not met and limitations apply

- The HS database has a size and/or compute (Cores) larger than the supported max size and/or Cores of the target GP SLO.
- The HS database is larger than the max size supported by the target SLO(GP).
- Applications on HS databases using functionality not supported in General Purpose tier may not work after this migration.
- For e.g. The HS database using Memory Optimized Table Types not supported by GP.
- After Reverse Migration from Hyperscale to GP, any existing Change Data Capture (CDC) files are deleted.
- After Reverse Migration from Hyperscale to GP, any existing Synapse Link Synapse metadata and objects are deleted
- You cannot migrate out of the Hyperscale HS\_Gen4\_1 (1 CPU) tier. You will need to scale up your HS database to more than 1 core (Gen4\_xx or Gen5\_xx where xx > 1) before performing the migration.
- To migrate out of Hyperscale, the only valid target tier is the GP tier (Provisioned/Serverless). Once in GP tier, further tier changes to BC/Premium are possible.
- Due to the differences in the Maximum log size limits between Hyperscale(unlimited) and GP (resource limits) Tiers, Reverse-migrated database may temporarily exceed the documented log size limits of target tier. SQL database system will eventually reconcile these differences and enforce the documented GP tier resource limits. Customers who are further changing tier of the Reverse migrated database before this reconciliation may encounter errors.

## Reason for the limitations

- Hyperscale has a different, newer underlying architecture from other non-Hyperscale SQL database tiers and provides highly scalable storage and compute performance. It is also positioned as the basis of all newer generations of SQL Database tiers.
- We expect Hyperscale to provide a superset of all the features and functionality of all the other tiers in the near future.
- Given this roadmap, we are looking at the Reverse Migration feature as a re-assurance feature for customers moving from other SQL database tiers in case they wanted to move back for whatever reason. Hence its scope is limited at this time.
- We also want to assure customers that we will make sure that their Hyperscale deployments are successful, and it will not be necessary for them to move back to other tiers. So, it is a technical & business decision to have limited scope for this feature.

## Monitoring Migration

Migration progress can be tracked by querying a DMV in the master DB. 'Percent complete' column shows the progress.

```
SELECT * FROM sys.dm_operation_status
WHERE major_resource_id = 'DB Name'
ORDER BY start_time DESC;
```

## Migrating using APIs/Tools

Experience is similar to migrating between vCore tiers, for e.g.: BusinessCritical to GeneralPurpose

## Migrating out of Hyperscale using T-SQL

```
ALTER DATABASE <database_name>

MODIFY (SERVICE_OBJECTIVE = 'GP_Gen5_4', MAXSIZE = 1024 GB);
```

When the maxsize argument is passed, we will honor it. If the max-size passed exceeds a valid max size for the target sku, we will produce an error. If the argument is not passed, we will default to the max\_size available for the given SKU.

## Migrating out of Hyperscale using Azure CLI

```
az sql db update -g mygroup -s myserver -n mydb --edition GeneralPurpose --service-objective GP_Gen5_4 [--max-size <Size>]
```

When the max-size argument is passed, we will honor it if possible. If the max-size passed exceeds a valid max size for the target SKU, we will produce an error. If the argument is not passed, we will default to the max\_size available for the target service-objective.

## Migrating out of Hyperscale using PowerShell

```
Set-AzSqlDatabase -ResourceGroupName $resourceGroupName -ServerName $servername -
DatabaseName $databasename -Edition "GeneralPurpose" -RequestedServiceObjectiveName "GP_Gen5_4"
```

As with the Azure CLI, if the argument MaxSizeBytes is passed, we will honor it. If the argument provided is larger than the available size of the target SKU, we will throw an error. If the size is not specified, we will default to the maximum available for the target service-objective.

## Handling Customer Questions

- **Will my database experience down time when I perform this operation?**
  - In general, this process is a size-of-data operation and could take several hours. This duration depends mainly on the size of the database and concurrent write activities happening during migration. However, you will only experience a short time (few minutes) of downtime during the final cutover to the new target GP tier.
- **What happens to backups when I move out of Hyperscale?**

- Only backups from the current and once-previous tier of your database will be available for restore. This means that if you have moved from GP to Hyperscale and back to GP, the only backups available are the ones from the current GP incarnation and the immediate previous Hyperscale incarnation. Any previous tiers tried will not have backups available.

For example, if you migrate 5 times this way:

1. GP
2. Hyperscale
3. GP
4. Hyperscale
5. GP

The only backups available will be from steps 4 and 5 of the timelines. Any backups from previous steps will be unavailable.

- **How long does the migration take to complete?**

The first phase of the migration work is the backup process, which is the size of data operation and can take hours to complete. After this, the restore into the new SLO takes an amount of time dependent on the size of the data, and any ongoing workload.

- **What will be the impact to Billing?**

- Regular Hyperscale and GP billing rates apply to this feature, even during Private/Public preview. If any backups exist, you will be billed for those as the existing guidelines.
- Please note that databases that leave behind backups in Hyperscale will be billed for the backup snapshots and for a size-of-data blob that must be retained to be able to restore the backup – although the database is moved to a different tier, hyperscale backups will treat the database as deleted for backup billing and retention purposes.

- **How to access this feature from Portal**

- Currently Portal does not support tier change from Hyperscale to any other SQL database tier. Available target tiers are grayed out.
- This experience will be changed to provide Portal experience identical to migrating a Business-Critical database to a compatible General-Purpose database. i.e., the drop-down list of target tiers should show General Purpose tier (both Provisioned and Serverless), in addition to Hyperscale tier. Other tiers should remain grayed out for this tier change.

## Architecture/benefits/limitations walk thru by PG

[Link to video](#) 

### How good have you found this content?

