

# Strategies to reduce the backup storage

Last updated by | Vitor Tomaz | Jun 8, 2022 at 5:35 AM PDT

## Contents

- [Issue](#)
- [Investigation/Analysis](#)
- [Mitigation](#)
- [Public Doc Reference \(optional\)](#)
- [Internal Reference \(optional\)](#)

## Issue

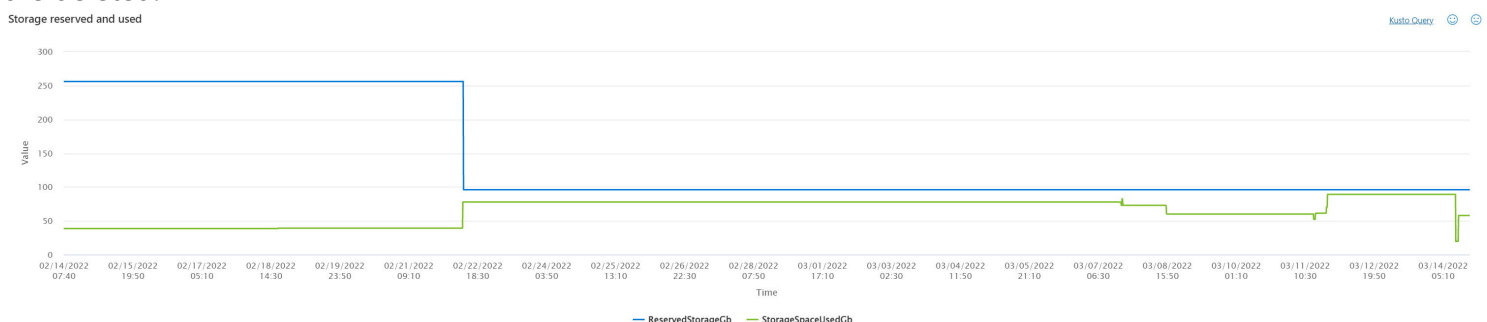
When customer reach us about having too much backup storage being used and they don't understand why there are some justifications we can provide and some strategies we can share for the customer to use less backup storage.

## Investigation/Analysis

The best source of information about backups is the ASC.

Using the Resource Explorer or by running a troubleshooter go to the Backup/Restore section and then select the "Short-Term Backup storage consumption" tab. There you will see information about 'SQL Database Managed Instance PITR Backup Storage' billing meter and Storage reserved and used.

In this example we can see that the reserved space has drop because the Azure MI reserved storage was reduced but the backup used storage change according to the backups that are being taken and the ones that are deleted.




Because customer can change the Azure MI reserved storage they can become from not paying for the backup storage to start paying for it because the backup storage used is only released when the backups are outside of the retention window plus the period needed to have a full restore chain since the previous full backup of the oldest restore point of the retention window.

See the wiki article on the internal reference section for a more detailed explanation.

## Mitigation

Strategies to reduce the backup storage used:

1. Review the backup retention policy for all the databases on the Azure Managed Instance to confirm that the retention window is in fact what is needed, maybe there are some databases that can have a shorter retention window.
2. Before dropping a database and if we know that we will not be needing the backup to restore it later, reduce the backup retention window to the lowest possible value we are comfortable with.  
This is specially usefull and can save some money for customer that have a pattern of creating from a backup (restore) and dropping databases very frequently. By default any database created will have 7 days of backup retention window and will have a full backup being taken after the restore. As you can see even if the database is only used for a couple of hours the backups will remain available and consuming storage for the retention period.  
**ALERT**, after reducing the backup retention window there is a chance of the customer not be able to restore to an older point in time even if he increase again the backup retention window.
3. Avoid doing unnecessary index maintenance. Index maintenance cause a high usage of transaction log that will cause transaction log backups to be bigger. Advise the customer to just rebuild\reorg the indexes that really need to be rebuild\reorg.  
There are some scripts that do that automatically but we can only advise to use one of the followings:
  - [AdaptiveIndexDefrag](#) 
4. Use TempDB instead of permanent tables in your application logic for storing temporary results and/or transient data
5. Disable TDE on databases that don't require being encrypted. Backups are compressed by the system, and encrypted databases can't be compressed as much as non-encrypted databases, causing more IO and taking more time.

## Public Doc Reference (optional)

[Backup storage consumption](#) 

## Internal Reference (optional)

[Point-In-Time Process explained](#)

## How good have you found this content?



-