Transaction Log full due to backups

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

Contents

Issue

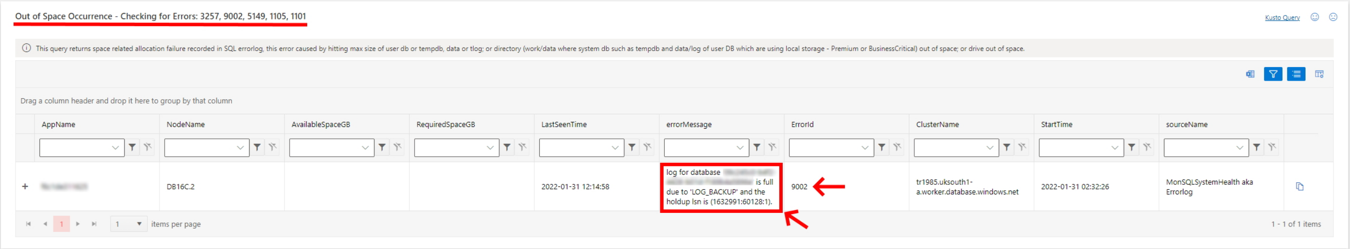
The database transaction log file is full due to the lack of backups being taken.
The errors customer will receive are the 9002 or 40552 when the transaction log is full and cannot accept new transactions.

Investigation/Analysis

CSS side

1. Check if there are any ASC insights about transaction log full or issues with backups
    - The insights will provide more information on what is happening and an action plan.

2. Check or generate a ASC Managed Instance troubleshooter report
    - Go to the *Performance* tab and select the *Space issues*
      Check if there are any occurrence and what is the error message of it.



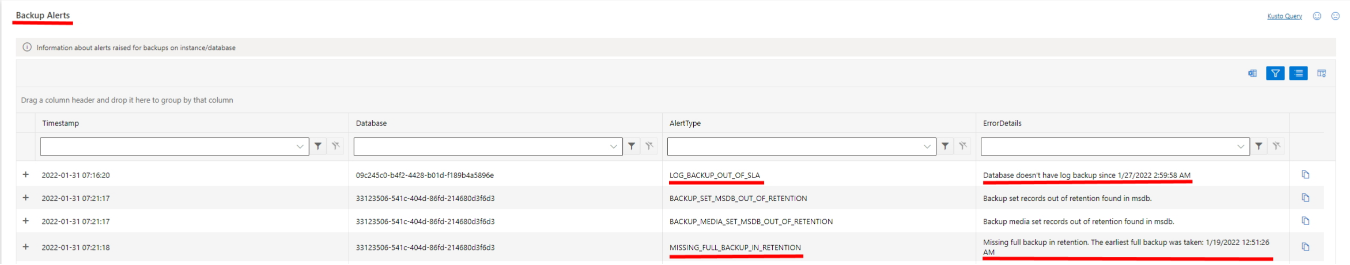
Out of Space Occurrence - Checking for Errors: 3257, 9002, 5149, 1105, 1101

This query returns space related allocation failure recorded in SQL errorlog, this error caused by hitting max size of user db or tempdb, data or tlog; or directory (work/data where system db such as tempdb and data/log of user DB which are using local storage - Premium or BusinessCritical) out of space; or drive out of space.

Drag a column header and drop it here to group by that column

AppName	NodeName	AvailableSpaceGB	RequiredSpaceGB	LastSeenTime	errorMessage	ErrorId	ClusterName	StartTime	sourceName
[redacted]	DB16C.2			2022-01-31 12:14:58	log for database [redacted] [redacted] is full due to 'LOG_BACKUP' and the holdup lsn is (1632991:60128:1).	9002	tr1985.uksouth1-a.worker.database.windows.net	2022-01-31 02:32:26	MonSQLSystemHealth aka Errorlog

    - Go to the *Backup/Restore* and select the *Short-Term Retention Backups* Confirm if there are any alerts regarding backups not being taken



Backup Alerts

Information about alerts raised for backups on instance/database

Drag a column header and drop it here to group by that column

Timestamp	Database	AlertType	ErrorDetails
2022-01-31 07:16:20	09c245c0-b4f2-4428-b01d-f189b4a5896e	LOG_BACKUP_OUT_OF_SLA	Database doesn't have log backup since 1/27/2022 2:59:58 AM
2022-01-31 07:21:17	33123506-541c-404d-86fd-214680d3f6d3	BACKUP_SET_MSDB_OUT_OF_RETENTION	Backup set records out of retention found in msdb.
2022-01-31 07:21:17	33123506-541c-404d-86fd-214680d3f6d3	BACKUP_MEDIA_SET_MSDB_OUT_OF_RETENTION	Backup media set records out of retention found in msdb.
2022-01-31 07:21:18	33123506-541c-404d-86fd-214680d3f6d3	MISSING_FULL_BACKUP_IN_RETENTION	Missing full backup in retention. The earliest full backup was taken: 1/19/2022 12:51:26 AM

3. Confirm when were the last backups taken
   Take into account the backup frequency, full backups are taken every 7 days, diff backups every 12-24 hours and log backups every 5 to 10 minutes.

```
MonBackup
| where logical_server_name =~ '<instance name>'
| where event_type == 'BACKUP_METADATA_DETAILS'
| summarize max(PreciseTimeStamp) by logical_database_name, backup_type
| order by logical_database_name
```

Something more rare, but can happen, is corruption. If corruption is detected in the transaction log the backups are not taken causing the log file to grow until it's fixed.
To confirm if corruption was detected we need to run the following Kusto queries:

```
MonSQLSystemHealth
| where TIMESTAMP > datetime(2022-03-14 12:49:31.4520426) //place the same timestamp as the last backup
| where SubscriptionId =~ "<subscription id>"
| where LogicalServerName =~ "<instance name>"
| where message contains "corruption in database"
| project originalEventTimestamp, LogicalServerName, AppName, SubscriptionId, message
```

Customer side To discover what is preventing log truncation in a given case, refer to log_reuse_wait_desc in sys.databases. The log reuse wait informs you to what conditions or causes are preventing the transaction log from being truncated by a regular log backup.

```
SELECT [name], log_reuse_wait_desc FROM sys.databases;
```

The following values of log_reuse_wait_desc in sys.databases may indicate the reason why the database's transaction log truncation is being prevented:

log_reuse_wait_desc	Diagnosis	Response required
NOTHING	Typical state. There is nothing blocking the log from truncating.	No.
CHECKPOINT	A checkpoint is needed for log truncation. Rare.	No response required unless sustained. If sustained, create a CRI.
LOG BACKUP	A log backup is in progress.	No response required unless sustained. If sustained, create a CRI.
ACTIVE BACKUP OR RESTORE	A database backup is in progress.	No response required unless sustained. If sustained, create a CRI.
ACTIVE TRANSACTION	An ongoing transaction is preventing log truncation.	The log file cannot be truncated due to active and/or uncommitted transactions. See mitigation section.
REPLICATION	In Azure SQL Managed Instance, due to replication or Change Data Capture (CDC).	In Azure SQL Managed Instance, if sustained, investigate agents involved with CDC or replication. For troubleshooting CDC, query jobs in msdb.dbo.cdc_jobs. If not present, add via sys.sp_cdc_add_job. For replication, consider Troubleshooting transactional replication. If unresolvable, create a CRI.
AVAILABILITY_REPLICA	Synchronization to the secondary replica is in progress.	No response required unless sustained. If sustained, create a CRI.

Another point to look at is the file autogrowth settings for the transaction log. Look if the autogrowth is enabled and if the autogrowth value is big enough to handle the current workload.

Mitigation

If the transaction log file is full or continuously growing due to backup issues then open a CRI.

To avoid or to temporarely mitigate the issue customer can increase the size of the transaction log file while the PG is working on the backup issues.
To get the current transaction log details run the following T-SQL.

```
USE [<database name>]
GO

SELECT
        [name] LogFileName,
        size*8 FileSizeInKB,
        size/128.0 FileSizeInMB,
        size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS int)/128.0 AS EmptySpaceInMB,
        max_size/128.0 MaxSzeInMB,
        growth/128.0 GrowthInMB,
        is_percent_growth,
        is_read_only
FROM sys.database_files
WHERE type_desc = 'LOG'
```

To increase the transaction log file use the following T-SQL.

```
USE [master]
GO

ALTER DATABASE [<database name>] MODIFY FILE ( NAME = N'<transaction log file name without extenction>', SIZE
GO
```

Public Doc Reference (optional)

[Troubleshooting transaction log errors with Azure SQL Database and Azure SQL Managed Instance]

Root Cause Classification

Cases resolved by this TSG should be coded to the following root cause:
Azure SQL v3\Backup/Restore\Automated Backups

How good have you found this content?