Native Restore taking too long - CHECKSUM

Last updated by | Radhika Shah | Aug 22, 2022 at 6:27 PM PDT

Contents

Issue

Customer is complaining about the native restore (restore from Azure Storage) to be taking or have taken too long.

Investigation/Analysis

One common reason for the native restore to take more time than what is expected is due to the backup being used was taken without CHECKSUM.

In Azure SQL Managed Instance, when doing native restore, we check whether the backup was taken with or without CHECKSUM. If the backup was not taken with CHECKSUM, once the DB is restored and recovered, we are switching to single user mode (and hence during this time customers can experience login failures - error 18456) and will execute a DBCC CHECKDB on it. If the backup was taken with CHECKSUM, we are not running CHECKDB validation after restoring, because we know that pages in DB are at least verified with checksum when the backup was taken.

Customers will only have access to the database when the DBCC CHECKDB finishes without any errors. This is to make sure that any database restored to Azure SQL Managed Instance is consistent.

We can see on the error log if the DBCC CHECKDB was executed after the restore.

By running the following Kusto queries you will confirm if the CHECKDB was executed and how long did it take to finish, or if it's still running.

```
MonRestoreEvents
|where TIMESTAMP >=datetime(2022-03-15 11:09:08.8770000)
|where TIMESTAMP <datetime(2022-03-22 13:09:08.8770000)
|where restore_request_id contains "a8c82657-4e0a-4da2-9a52-5b5b6aa99787"
| where LogicalServerName == "<instance name>"
//| where restore_database_progress contains "Executing restore query RESTORE DATABASE" //uncomment this line
| project originalEventTimestamp, LogicalServerName, AppName, event, restore_database_progress, message, resto
```

With the information (AppName and database id) obtained from the previous query we can run the following query

Example on where to find the database id from the output of the previous query: "Executing restore query RESTORE DATABASE [**xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx**] FROM URL = N'..."

```
let databaseid = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"; //database name while restoring, get it from MonResto
MonSQLSystemHealth
| where AppName == "xxxxx"
| where TIMESTAMP >=datetime(2022-03-14 11:09:08.8770000)
| where TIMESTAMP <datetime(2022-03-15 17:09:08.8770000)
| where message contains databaseid or logical_database_guid =~ databaseid
| where message contains "CHECKDB"
| project originalEventTimestamp,  message
```

Customers can also see the CHECKDB being executed by reading the Error Log.

As the Error Log sometimes contains too much information, we can use the sp_mirestoreinfo script to filter and just view the relevant information regarding restores.

Mitigation

There is no mitigation; If the backup was taken without CHECKSUM, the CHECKDB will always be executed. The workaround is to take another backup with CHECKSUM (if possible) and restore using that backup; or customer can do online DB migration with DMS (Database Migration Service). If doing online DB migration, checksum is required on every backup that is being restore on MI, hence no corruptions can occur.

Internal Reference

ICM 243489472

How good have you found this content?