Database restore keeps ongoing with multiple Dumps

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:29 AM PST

Contents

Issue

Customer might report that a restore from a backup that has been taken on on-prem is slow/stuck or that has been cancelled after long hours.

The Managed Instance where the backup is being restored is on General Purpose.

Investigation/Analysis

On ASC multiple dumps ocuring:

The restore could show that has been canceled:

```
MonRestoreRequests
        | where start_time >= datetime(2021-10-29 00:00:00) and start_time <= datetime(2021-11-05 13:27:00
        | extend SourceServer = iif(archived_backup_resource_id!= '',              extract('/longTermRet
           source_managed_server_name)
        | extend SourceDatabase = iif(archived_backup_resource_id!= '',           extract('/longTermRetentio
           source_managed_database_name)
        | extend TargetServer = target_managed_server_name, TargetDatabase = target_managed_database_name
        | where (SourceServer =~ 'server'
           and (isempty('') or SourceDatabase =~ '')
        ) or
        (TargetServer =~ 'server'
           and (isempty('') or TargetDatabase =~ '')
        )
        | summarize arg_max(TIMESTAMP, *) by restore_request_id
        | project restore_request_id = tolower(restore_request_id), TargetServer, TargetDatabase
        | join kind=inner (
        MonRestoreEvents
        | project originalEventTimestamp, restore_request_id = tolower(restore_request_id), restore_databa
        ) on restore_request_id
        | sort by originalEventTimestamp desc nulls last
        | project originalEventTimestamp, restore_request_id = tolower(restore_request_id), TargetServer,
        | limit 100
```


But the database is still in creation mode:

```
MonAnalyticsDBSnapshot
        | where PreciseTimeStamp >= ago(7d)
        | where logical_server_name == 'server_name'
        |where logical_database_name like 'database_name'
        | summarize by logical_database_name, customer_subscription_id, resource_group, logical_server
        //| extend ResourceUri = strcat('/subscriptions/', customer_subscription_id ,'/resourcegroups/
        | project Database = logical_database_name,
     SLO = service_level_objective,
     FsmState = state,
      TIMESTAMP
        |order by TIMESTAMP desc
        |take 100
```

Database	SLO	FsmState	TIMESTAMP
MiscData	MIGP16G5	WaitingForPhysicalDatabaseCreation	2021-11-04 17:36:00.9982389
MiscData	MIGP16G5	WaitingForPhysicalDatabaseCreation	2021-11-04 17:06:09.9176535
MiscData	MIGP16G5	WaitingForPhysicalDatabaseCreation	2021-11-04 16:35:31.9303832
MiscData	MIGP16G5	WaitingForPhysicalDatabaseCreation	2021-11-04 16:05:54.9897063
MiscData	MIGP16G5	WaitingForPhysicalDatabaseCreation	2021-11-04 15:36:00.0181900

The database had memory-optimized objects while onprem.

Mitigation

The RCA is that the customer DB is already corrupted on onprem.

Suggested mitigation is to:

- create a new database
- verify the Hekaton fields in the bootpage are zeroed out use query

```
DBCC DBINFO('<db_name>') WITH TABLERESULTS
```

and these fields value should be zero: dbi_hkRecoveryLSN, dbi_hkLogTruncationLSN, dbi_hkRootFile, dbi_hkRootFileWatermark, dbi_hkTrimLSN

- migrate data to new database -
- verify the Hekaton fields in the bootpage are zeroed out

use query

```
DBCC DBINFO('<db_name>') WITH TABLERESULTS
```

and these fields value should be zero: dbi_hkRecoveryLSN, dbi_hkLogTruncationLSN, dbi_hkRootFile, dbi_hkRootFileWatermark, dbi_hkTrimLSN

- take a full backup
- restore on Managed Instance

The PG is already working on a fix where the customer is prompted about the issue.

Internal reference

Work item: 1852942 ↗

ICM:

356531097 ↗

352490213 ⤢

309687205 ⤢

270529546 ⤢

267412886 ⤢

How good have you found this content?