Backups are missing(slash)Restore failing due to missing backups

Last updated by | Radhika Shah | Mar 17, 2023 at 3:31 PM PDT

Contents

Issue

Customers notice that the backups on their Managed Instance are missing (no restore point available). Or they encounter error when trying to restore a database from a particular Point-in-time.

Error:

```
No backups were found to restore the database to the point in time 2/16/2023 12:30:00 AM (UTC). Please contact
```

Investigation/Analysis

Check for the error on ASC troubleshooter --> Backup/Restore tab --> Restores.

Sample output:

start_time	**2023-03-15 18:07:42**
end_time	2023-03-15 18:12:49
operation_type	CreateManagedRestoreRequest
restore_type	Restore
restore_request_id	5A599F66-B31F-4E22-AD09-AB788400ABD0
state	Failed
sourceSubcriptionId	<sourceSubcriptionId>
SourceServer	<sourceServer>
SourceDatabase	<sourceDBname>
source_edition	
point_in_time	**2023-02-16 00:30:00**
targetSubcriptionId	<targetSubcriptionId>
TargetServer	<targetServer>
TargetDatabase	<targetDBname>
target_edition	
operation_details	No backups were found to restore the database to the point in time 2/16/2023 12:30:00 AM (UTC). Please contact support to restore the database.
error_code	1200003
error_message	No backups were found to restore the database to the point in time 2/16/2023 12:30:00 AM (UTC). Please contact support to restore the database.

Scenario1 - Invalid PiTR

Short term backup retention can be anywhere between 1-35 days. Check the backup retention set on the database. This is ideally found on the properties page of the MI database on ASC. Please note that is the **current** setting of the backup retention on the database. This setting may have been different in the past.

Below kusto can help check the historical backup retention setting on the MI Database and if there were any changes made within the telemetry retention:

```
MonAnalyticsDBSnapshot
| where LogicalServerName =~ '{serverName}' or logical_server_name == '{serverName}'
| where logical_database_name == '{databaseName}'  //this is the source db from where the PiTR is requested
| summarize max(TIMESTAMP) by logical_database_name, physical_database_id, logical_database_id, backup_retenti
```

Sample output:

logical_database_name	physical_database_id	logical_database_id	backup_retention_days	create_t
<databaseName>	00265A78-96A0-4C26-93D6-6F102EE2743A	71A6E2C8-BBE5-4ED5-93C0-BD5E95B39637	**7**	2021-0 11:58:3
<databaseName>	00265A78-96A0-4C26-93D6-6F102EE2743A	71A6E2C8-BBE5-4ED5-93C0-BD5E95B39637	**30**	2021-0 11:58:3

In this example above, we can see that the database backup retention was 7 days and was updated to 30 days on **2023-03-14**. Any updates made to backup retention is applied to **future backups**, the existing backups do not change.

Public doc: https://learn.microsoft.com/en-us/azure/azure-sql/managed-instance/automated-backups-change-settings?view=azuresql&tabs=azure-portal#change-short-term-retention-policy

So, when customer requested the backup retention update, the earliest available backup (to restore) would be 7 days before 2023-03-14 --> 2023-03-07 and after updating the retention, the backups from 2023-03-07 will stay until 30 days after.

The restore attempt made by customer failed because at the time of request, no backups from 2023-02-16 existed and this is by design.

Scenario2 - Source MI part of failover group and is/was secondary Geo-DR at the time of backup point-in-time

Point-in-time Restore (PITR) allows a customer to restore a copy of a database to any point in time within the configured retention period of a database. These restores are fulfilled by using automated backups taken periodically by the service (not customer initiated) and to avoid duplicate data, automated backups are not taken for databases in a secondary role of a geo-relationship. While restore is searching for backups to use, however, we only search through the backups of the source database.

With the current design, this leads to the issue that databases that are currently global primary role allow restore requests for any Point-In-Time but will fail with “No backups found” if the restore requires backups from a time when the database was in a geo-secondary role.

1. Check if the Managed Instance is part of failover group via ASC --> troubleshooter --> 'Geo DR' or via below kusto:

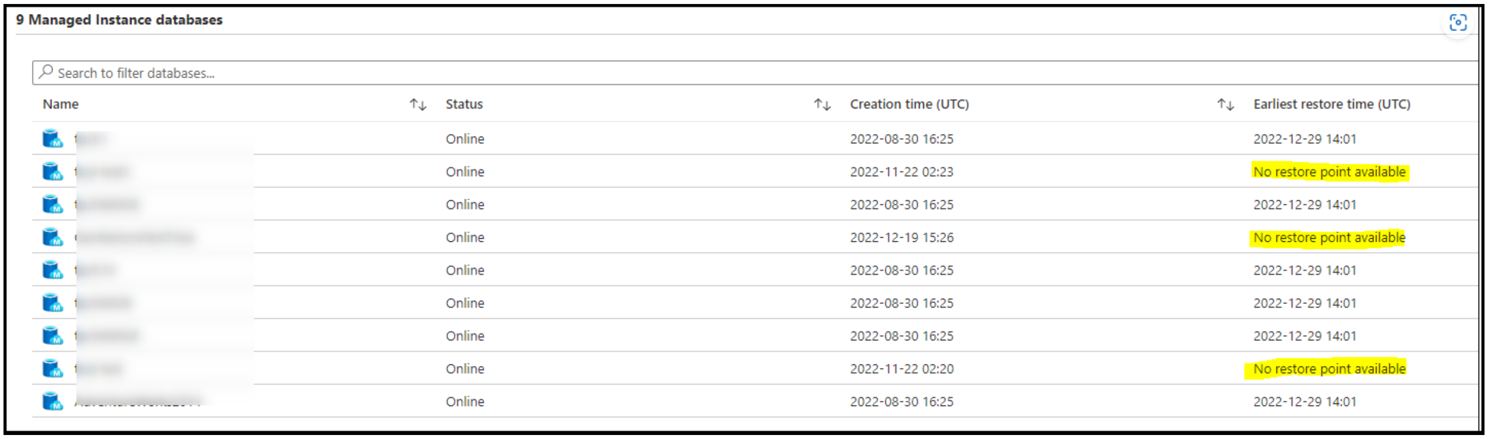
```
MonGeoDRFailoverGroups
| where logical_server_name == {ServerName}
| summarize arg_max(PreciseTimeStamp, *)
| project GroupId = failover_group_id, GroupName = failover_group_name, Policy = failover_policy, Server = log
```

2. If participating in FOG, what role (Primary or Secondary) the Managed Instance was in, at the time of the requested Point-In-Time. Check via ASC --> troubleshooter --> 'Geo DR' or via below kusto:

```
MonGeoDRFailoverGroups
| where TIMESTAMP >= {startTime}
| where TIMESTAMP <= {endTime}
| where logical_server_name == {ServerName}
| project PreciseTimeStamp, start_utc_date, end_utc_date, logical_server_name, role
```

If the Managed Instance is a Geo-Secondary, the backups will not be visible and this is by design. As per system design, Automated backups supporting the PITR functionality are always taken on the **primary** Managed Instance in a failover group (GeoDR) configuration. Upon failover between the primary and secondary, that is in the case when they reverse roles, new backup cycle starts on the primary. As the new backup cycle is started, this means that it is only possible to perform PITR on the current primary, and it is not possible to perform a restore from backups taken on the secondary (former primary). Subsequent failover, reversing the roles between primary and secondary, will allow to access the backup taken on the other (partner) managed instance, as long as it is primary.

It might happen that, for **some DBs** on geo secondary, have earliest restore point, and others do not. This is because of the fact that some DBs were present on that MI **when it was primary**, and others were not.



9 Managed Instance databases

Name	Status	Creation time (UTC)	Earliest restore time (UTC)
[redacted]	Online	2022-08-30 16:25	2022-12-29 14:01
[redacted]	Online	2022-11-22 02:23	No restore point available
[redacted]	Online	2022-08-30 16:25	2022-12-29 14:01
[redacted]	Online	2022-12-19 15:26	No restore point available
[redacted]	Online	2022-08-30 16:25	2022-12-29 14:01
[redacted]	Online	2022-08-30 16:25	2022-12-29 14:01
[redacted]	Online	2022-08-30 16:25	2022-12-29 14:01
[redacted]	Online	2022-11-22 02:20	No restore point available
[redacted]	Online	2022-08-30 16:25	2022-12-29 14:01

In the above example, for some DBs (where Earliest restore time is not available on secondary) indicates that the databases were created on Primary, backups were taken on Primary and ever since the databases were created on Primary (and hence replicated to geo-secondary), the replication roles have not switched (meaning, there was no geo-failover initiated) since the time the databases were created on Primary. Hence no backups/earliest restore point are available on secondary.

Mitigation

Scenario1 - This is by design. Updates to backup retention applies to future backups only, existing backups are not impacted.

Scenario2 - This issue is by design and there is no mitigation for now.

Workaround (for Scenario2):

A possible workaround for customers who need to restore from a specific point-in-time from backup that was taken on previous primary (now secondary), can consider a failover to reverse the roles between primary and secondary. This will allow them to access the backup taken on the other (partner) managed instance, as long as it is primary.

Internal Reference

ICM 214869341

ICM 241073765

ICM 307494409

Blogs

Why no restore point available?

How good have you found this content?

🙂 🙁