

Dropping geo-secondary database or Managed Instance is taking long or stuck in TailLogBackupInProgress

Last updated by | Radhika Shah | Sep 28, 2022 at 7:17 AM PDT

Contents

- [Issue](#)
- [Investigation](#)
- [Mitigation](#)
- [Internal Reference](#)

Issue

Customers attempt to drop a database on their geo-secondary Managed Instance or attempt to drop the entire geo-secondary instance and see that its either long running, or the drop appear to be stuck or face an error message:

```
'statusMessage': '{\\'status\\':\\'Failed\\',\\'error\\':{\\'code\\':\\'ResourceOperationFailure\\',\\'message
```

Investigation

On ASC --> Provisioning, Management operations show that the drop failed with an error message:

The database '<DBName>' was requested to be dropped, and rollback for drop operation is not supported.

On ASC Resource Explorer for the Database properties, you can see the DB state is 'TailLogBackupInProgress'. The state of DB can also be verified from xts or via below Kusto:

```
MonAnalyticsDBSnapshot
| where PreciseTimeStamp >= {Datetime}
| where logical_server_name == {ServerName}
| where logical_database_name == {DatabaseName}
| 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,
           ResourceUri
```

Confirm if the database (to be dropped) is part of Geo-Secondary Managed Instance.

What Happened?

By design, we only take backups on the Primary instance, we do not take backups on geo-secondaries. So, when dropping a DB that is part of geo-secondary, there is an issue in code where we do not check if this DB was secondary in order to skip the TailLogBackup.

Since TailLogBackup is not being skipped for secondaries, and backups are not taken for secondaries, TailLog will first force the Full backup to be taken, in order for log backup to be taken next.

The same applies for secondary Instance drop. When customer drops the Fail Over Group (FOG), the secondary instance becomes standalone MI. This initiates first full backups on the now standalone Instance (since there were no backups taken when instance was geo-secondary). When a drop is initiated at this time, the system will first finish taking the tail log backups before it can drop.

Depending on how big the databases were at the time of drop, the full backup and tail log backup may take longer and eventually the drop will be initiated.

You can check the status of the backups and if any errors were reported using below kusto:

```
let startTime = ago(1h);
let endTime = now();
MonBackup
| where TIMESTAMP > startTime
| where TIMESTAMP < endTime
| where LogicalServerName == {ServerName}
| where event == "database_backup_error"
| project TIMESTAMP, LogicalServerName, event_type, logical_database_id, physical_database_id, backup_type, sq
//| where logical_database_id contains '{database guid}'
```

Mitigation

If you notice that the tail log backups have been taking too long or if there are errors with the backups, raise an ICM with the Backup Restore team to mitigate the issue.

Internal Reference

[ICM 239716218](#) 

[ICM 226333439](#) 

[Repair item](#) 

How good have you found this content?

