

Database in state SetOwnerSid

Contents

- [Issue](#)
- [Investigation/Analysis](#)
 - [What happened](#)
- [Mitigation](#)
- [Internal Reference](#)

Issue

Customers may complain about database disappearing/dropped.

Investigation/Analysis

Verify how/when the database was created - either as create new database or via a restore option. ASC SQM MI Troubleshooter report --> Provisioning --> Operations (if created as new database) or Backup/Restore --> Restores tab should have this information. Otherwise, below kusto can help verify how and when the database was created:

```
let startTime = ago(20d);
let endTime = ago(18d);
let serverName = '>miname<';
let dbName = '>dbName<';
MonManagementOperations
| where originalEventTimestamp >= startTime and originalEventTimestamp <= endTime
| where operation_parameters has serverName and operation_parameters has dbName
| project originalEventTimestamp, request_id, event, operation_type, operation_parameters, error_code, error_m
```

Verify that the operation completed successfully and not rolled back, failed or timed out. In the event that the operation has rolled back or timed out, it is safe to assume that the createdb/restore db operation rolled back and the database was dropped.

To check the database state, on ASC Resource explorer (if database still exists) or on SQL MI Troubleshooter report --> Availability --> Basic --> Database State History, for the timeframe.

Below Kusto can also help verify state from a previous point in time:

```
let startTime = ago(12d);
let endTime = ago(5d);
let serverId = '<managed_server_id>';
let databaseName = '<Databasename>';
MonManagedDatabases
| where PreciseTimeStamp >= startTime and PreciseTimeStamp <= endTime
| where managed_server_id == serverId
| where managed_database_name == databaseName
| summarize arg_max(TIMESTAMP,*) by ['state']
| project managed_database_name, managed_server_id, managed_database_id , ['state'], TIMESTAMP
| join kind = inner (
MonManagedServers
| where managed_server_id == serverId
| distinct managed_server_id, name) on managed_server_id
| project TIMESTAMP, managed_server_id, managed_server_name = name, managed_database_id, managed_database_name
```

Sample Output:

TIMESTAMP	managed_server_id	managed_server_name	managed_database_id	managed_c
2022-10-05 09:19:48.6863405	ManagedServerId	MIServerName	LogicalDatabaseld	DatabaseN

To check the last reported exception, for an operation:

```
let startTime = ago(12d);
let endTime = ago(5d);
let serverId = '<managed_server_id>';
let databaseName = '<Databasename>';
MonManagedDatabases
| where PreciseTimeStamp >= startTime and PreciseTimeStamp <= endTime
| where managed_server_id == serverId
| where managed_database_name == databaseName
| where ['state'] == 'SetOwnerSid'
| project PreciseTimeStamp, ['state'], last_exception
```

Sample Output:

```
<Exception><Message>Could not find a principal name with a matching sid.</Message><Type>System.ServiceModel.Fa
```

Other types or error that can reported:

"The database principal owns a schema in the database, and cannot be dropped."

or

"The database principal owns a schema in the database"

What happened

Create/restore database requests from TSQL get stuck due to mismatching owner_sid between the request issuer and sys.sql_logins table. ManagedDatabaseStateMachine stays stuck in SetOwnerSid state, with last_exception typically being: "<Exception><Message>Could not find a principal name with a matching sid.</Message><Type>..."

This happens when create/restore operation was executed from SqlAgent or by AAD admin user. External connections are allowed before the above state, so from customer side (SSMS) the database is fully usable.

If this issue is not mitigated and database is not in 'Ready' state, the create/restore operation may roll back after a set amount of time and can cause the database to drop.

Mitigation

Once the behavior has been confirmed, check for any LSIs reported for this issue. If the LSI has not been addressed/resolved, create an ICM to the SQL Managed Instance: Database CRUD (not instance CRUD) team.

Internal Reference

[ICM 336360520](#) 

How good have you found this content?



-