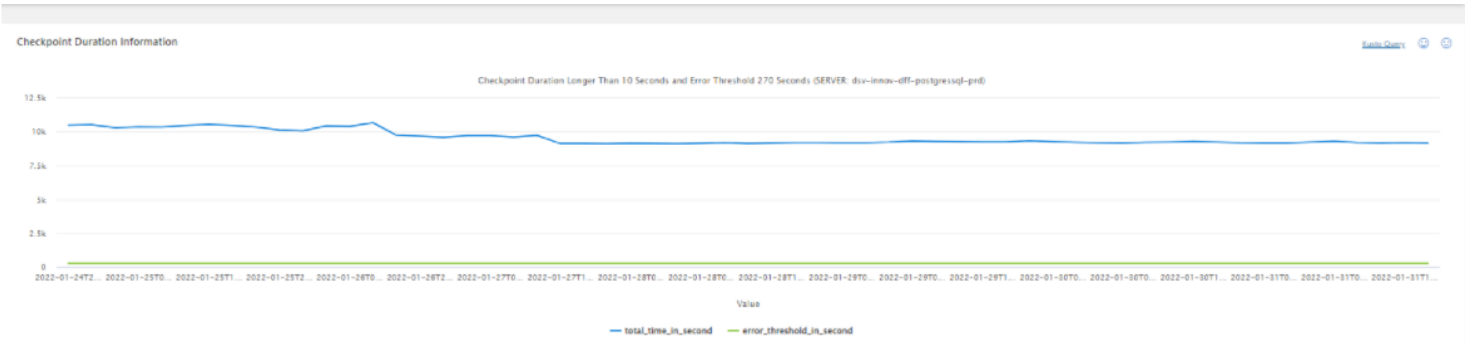
Create or drop database take too long and never complete - PGSS

Last updated by | Eduardo Santana | Dec 30, 2022 at 1:39 AM PST

This TSG is part of GT for any change please contact haaqel@microsoft.com

Some customers reporting that they are not able to create new databases , because the create command is taking too long and never ends.

Create or Drop Database internally trigger a checkpoint. so first check our telemetry to see if the checkpoints are taking really long which correspondingly causes DB create/drop to take long as well. You can check that from ASC (Perf tab) :



Checkpoints are known to take long time when there are large number of snapshot change files which are produced when logical replication is set but logical replication consumer does not consume the changes, resulting in such files' accumulation on primary server side.

Please check below TSG about unexpected increase in the storage consumption:

https://dev.azure.com/Supportability/AzureDBPostgreSQL/_wiki/wikis/AzureDBPostgreSQL/587393/Unexpected-increase-in-the-storage-consumption

These files are visited for every checkpoint, causing checkpoint to slow down and impacting create/drop database performance , so check if the customer has any inactive replication slot :

```
MonDmPgSqlReplicationStatsPrimary

| where LogicalServerName == "pgservername"

| order by LogicalServerName asc, slot_name asc, PreciseTimeStamp asc | serialize

| extend next_LogicalServerName = next(LogicalServerName, 1), prev_LogicalServerName =
prev(LogicalServerName, 1),

  next_AppName = next(AppName, 1),                          prev_AppName = prev(AppName, 1),

  next_slot_name = next(slot_name, 1),                      prev_slot_name = prev(slot_name, 1),

  next_active = next(active, 1),                            prev_active = prev(active, 1),

  slot_type
```

```
| where LogicalServerName != next_LogicalServerName        or LogicalServerName !=
prev_LogicalServerName

  or AppName != next_AppName                              or AppName != prev_AppName

  or slot_name != next_slot_name                          or slot_name != prev_slot_name

  or active != next_active                                or active != prev_active

| project PreciseTimeStamp, LogicalServerName, AppName, slot_name,slot_type, active,
wal_sender_state, application_name, total_lag_in_bytes, now = now()
```

PreciseTimeStamp	LogicalServerName	AppName	slot_name	slot_type	ac...	wal_sender_state	application_name	total_lag_in_bytes	now
2022-01-28 18:19:12.1460196	[illegible]	ef0b5277d2aa	cdctest	logical	False				2022-02-25 11:40:10.4310135
2022-01-29 18:10:38.1540267	[illegible]	ef0b5277d2aa	cdctest	logical	False				2022-02-25 11:40:10.4310135

Please share this link with customer - they are strongly advised to monitor replication slots and drop unused/inactive ones to avoid severe problems, including unavailability in case of a restart/failover/deployment.

once they dropped the inactive slots , they should be able to create the database and the checkpoint time should be under normal rang.

https://docs.microsoft.com/en-us/azure/postgresql/concepts-logical#monitoring-slots