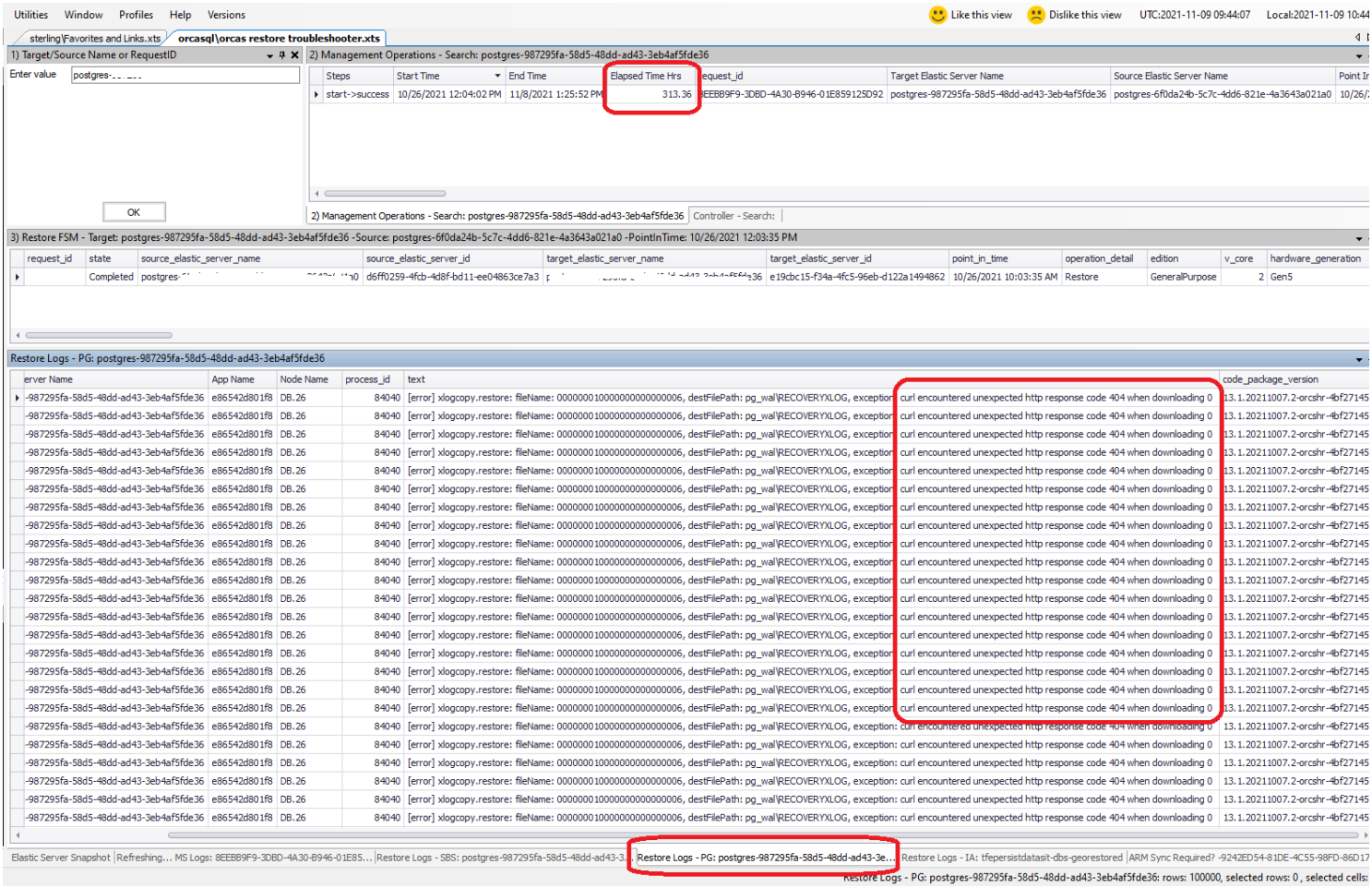
Long running PITR operations troubleshooting - after K61

Last updated by | Hamza Aqel | Jan 12, 2022 at 1:14 AM PST

We are receiving recently some cases where the customers are complaining a long restore operation , especially for newly create servers , and we noticed that after K61 deployment , to troubleshoot and make sure you are facing this issue , please follow the below instructions :

Check the restore process from our telemetry data in XTS/Kusto :

XTS (Orcas restore troubleshooter.xts):



Kusto:

```
MonRdmsPgSqlSandbox
| where LogicalServerName == "pg-server-name"
| where TIMESTAMP >= datetime(2021-11-08 12:00:04.0970232) and TIMESTAMP <= datetime(2021-11-08
14:50:04.0970232)
| project TIMESTAMP,text
```

2021-11-08 13:06:04.1062600	2021-11-08 13:06:00 UTC-6177ef7b.2c-LOG: switched WAL source from stream to archive after failure
2021-11-08 13:06:04.1062916	[info] xlogcopy.downloadXlogFile: start to download 0000000100000003000000AF
2021-11-08 13:06:04.1062976	[warning] xlogcopy.PgArchiveBlobContainer.download: curl encountered unexpected http response code 404 when downloading 0000000100000003000000AF.lz4 (2 retries left)
2021-11-08 13:06:04.1063030	[warning] xlogcopy.PgArchiveBlobContainer.download: curl encountered unexpected http response code 404 when downloading 0000000100000003000000AF.lz4 (1 retries left)
2021-11-08 13:06:04.1063080	[warning] xlogcopy.PgArchiveBlobContainer.download: curl encountered unexpected http response code 404 when downloading 0000000100000003000000AF.lz4 (0 retries left)
2021-11-08 13:06:04.1063135	[info] xlogcopy.downloadXlogFile: 0000000100000003000000AF.lz4 does not exist, trying uncompressed version
2021-11-08 13:06:04.1063183	[warning] xlogcopy.PgArchiveBlobContainer.download: curl encountered unexpected http response code 404 when downloading 0000000100000003000000AF (2 retries left)
2021-11-08 13:06:04.1063231	[warning] xlogcopy.PgArchiveBlobContainer.download: curl encountered unexpected http response code 404 when downloading 0000000100000003000000AF (1 retries left)
2021-11-08 13:06:04.1063278	[warning] xlogcopy.PgArchiveBlobContainer.download: curl encountered unexpected http response code 404 when downloading 0000000100000003000000AF (0 retries left)

If you noticed such error “curl encountered unexpected http response code 404 when downloading” this is indicating that the restore process is trying to download none existed WAL file .

Check IA logs:

```
MonRdmsInstanceAgent
| where originalEventTimestamp >= ago(15d)
| where LogicalServerName == "pg-server-name"
| where message_systemmetadata contains "txid_current"
| project originalEventTimestamp, LogicalServerName, process_id, NodeName,  message_systemmetadata
| order by originalEventTimestamp desc
```

2021-11-08 13:42:49.8053450	postgres-6f0da24b-5c7c-4dd6-821e-4a3643a021a0	192472	DB.0	[PostgresStatCollectorActor].ProcessQuery: Query='SELECT txid_current();' Exception='System.InvalidOperationException: Connection is not open at Npgsql.NpgsqlConnection.CheckReadyAndGetConnector() at Npgsql.NpgsqlCommand.<ExecuteDbDataReader>d__100.MoveNext() --- End of stack trace from previous location where exception was thrown --- at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw() at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task) at Npgsql.NpgsqlCommand.ExecuteDbDataReader(CommandBehavior behavior) at Npgsql.NpgsqlCommand.ExecuteReader()
2021-11-08 13:42:49.8053756	postgres-6f0da24b-5c7c-4dd6-821e-4a3643a021a0	192472	DB.0	[PostgresStatCollectorActor].ExecuteQueries: Error in writing to TSF file for command: 'SELECT txid_current();' and view name: 'MonDmPgSqlDummyTran'
2021-11-08 13:47:54.5242618	postgres-6f0da24b-5c7c-4dd6-821e-4a3643a021a0	192472	DB.0	[PostgresStatCollectorActor].ExecuteQueries: Executing query (SELECT txid_current();) at Server level for database
2021-11-08 13:47:54.5246353	postgres-6f0da24b-5c7c-4dd6-821e-4a3643a021a0	192472	DB.0	[PostgresStatCollectorActor].ProcessQuery: Query='SELECT txid_current();' Exception='System.InvalidOperationException: Connection is not open at Npgsql.NpgsqlConnection.CheckReadyAndGetConnector() at Npgsql.NpgsqlCommand.<ExecuteDbDataReader>d__100.MoveNext() --- End of stack trace from previous location where exception was thrown --- at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw() at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task) at Npgsql.NpgsqlCommand.ExecuteDbDataReader(CommandBehavior behavior) at Npgsql.NpgsqlCommand.ExecuteReader()

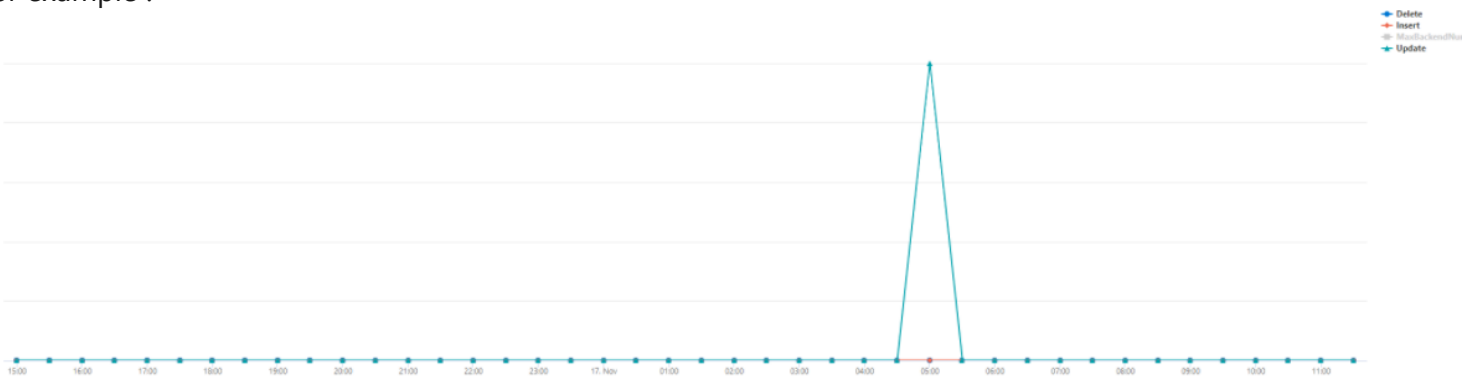
Check write activity :

Check if there is any write activity on the customer source Database after PITR as we recently had a bug in PITR , where it will not stop before hitting a write transaction after PITR time :

```
let TimeCheckStart = datetime('xxxx-xx-xx xx:xx:xx');
            let TimeCheckEnd = datetime('xxxx-xx-xx xx:xx:xx');
            MonDmPgSqlTransactionStats
            | where PreciseTimeStamp > TimeCheckStart and PreciseTimeStamp < TimeCheckEnd and LogicalServerName =~ 'pg-source-servername'
            | where database_name !in ('template1','template0', 'azure_sys','azure_maintenance')
              | summarize sum(tolong(numbackends)),sum(tolong(tup_inserted)),sum(tolong(tup_updated)),sum(tolong(tup_deleted)) by bin(PreciseTimeStamp,1m)
              | order by PreciseTimeStamp asc
              | serialize
              | extend tup_inserted_lastrecord=prev(sum_tup_inserted,1)
              | extend tup_updated_lastrecord=prev(sum_tup_updated,1)
              | extend tup_deleted_lastrecord=prev(sum_tup_deleted,1)
              | project PreciseTimeStamp,
                backendprocesses=sum_numbackends,
                diff_tup_inserted=iff( (sum_tup_inserted-tup_inserted_lastrecord)<0,0, (sum_tup_inserted-tup_inserted_lastrecord) ),
```

```
                diff_tup_updated=iff( (sum_tup_updated-tup_updated_lastrecord)<0, 0,(sum_tup_updated-
tup_updated_lastrecord) ),
                diff_tup_deleted=iff( (sum_tup_deleted-tup_deleted_lastrecord)<0,0, (sum_tup_deleted-
tup_deleted_lastrecord) )
            | summarize MaxBackendNum = max(backendprocesses), Insert = sum(diff_tup_inserted), Update = sum(diff_tup_
updated), Delete = sum(diff_tup_deleted) by bin(PreciseTimeStamp,30m)
            | order by PreciseTimeStamp asc
            | render timechart
```

for example :



in the above screen shot , we had only one write transaction happened at 5:00 , for any PITR after that date , for example if the PITR was at 11:00 , it will not finish till the customer perform a write transaction.

Customer RCA

Essentially Azure database for PostgreSQL Single server managed service attempts to complete a periodic synthetic transaction to generate a transaction record every minute.

This transaction record is what facilitates restores to customer chosen target time even if there has been no write workload activity around the time to generate transaction records naturally.

However due to recently discovered bug, our service's connection to PostgreSQL sometimes gets stalled intermittently without failing/completing. Subsequent statements executed on this connection fail resultingly, including the one to create synthetic transaction.

We are working on a fix for this and plan to roll it out in the next deployment which will be by end of Jan 22 (will share more details later) . In the meantime, please issue some write activity before performing restore and then issue the restore request, preferably to a time post writes.