GeoDR - ASC, XTS and Kusto

Last updated by | Balaji Barmavat | Nov 30, 2020 at 5:38 PM PST

Contents

ASC:

The ASC report was filed only on one of the geo-databases. For geo-replicated databases, especially ones involving geo-failover, It would be good to run the ASC for both the source of geo-failover and target of geo-failover, so that the combined insights can answer the questions

XTS:

For a quick overview: open **Database Replicas.xts** for the primary database and the check HADRON DMW subquery.

Details to look at:

- **last_hardened_lsn** should ideally be the same on all replicas, or very close to each other
- **last_redone_lsn** should be close to the **last_harded_lsn**, and **redo_queue_size** should be a low value < 1000000.
- **catchup_progress** tells you about the transfer of log records from primary to secondary database; the difference between **last_redone_lsn** and **last_harded_lsn** tells you how far the transferred log records have been applied to the secondary.

If **catchup_progress** is < 100%, there is a delay between primary and secondary;
if the **redo_queue_size** is large, then the latency is caused locally at the secondary database (e.g. undersized SLO, blocking)

Kusto:

Troubleshooting errors:

```
HttpIncomingRequests
| where TIMESTAMP >= datetime(2017-02-09 13:00:00Z)
| where TIMESTAMP <= datetime(2017-02-10 10:00:00Z)
| where subscriptionId =~ "Input SubscriptionId here"
| project TIMESTAMP, operationName, httpMethod, PreciseTimeStamp, commandName, parameterSetName,
targetUri, userAgent, durationInMilliseconds, httpStatusCode, exceptionMessage, errorCode, failureCause,
errorMessage, SourceNamespace, ActivityId, correlationId, clientRequestId, clientApplicationId,
serviceRequestId, Deployment, RoleInstance, TaskName, authorizationAction
//| where clientRequestId =~ "9073d335-71d3-46d6-a390-221a2721b7ef"
| where (clientRequestId =~ "9073d335-71d3-46d6-a390-221a2721b7ef" or correlationId == "4ee75fc4-ca49-
40fb-ac3b-95d2f4901a30")
| limit 1000

HttpIncomingRequests
| where TIMESTAMP >= datetime(2017-05-18 04:00:00Z)
| where TIMESTAMP <= datetime(2017-05-30 12:00:00Z)
| where subscriptionId =~ "Input SubscriptionId here"
| where targetUri contains "unily3devsql" //server name
| where targetUri contains "Analytics_Aetna" // database name
//| where operationName !contains "METRICS"
//| where httpMethod != "GET"
//| where httpMethod =~ "DELETE"
| project TIMESTAMP, operationName, httpMethod, PreciseTimeStamp, commandName, parameterSetName,
targetUri, userAgent, durationInMilliseconds, httpStatusCode, exceptionMessage, errorCode, failureCause,
errorMessage, SourceNamespace, ActivityId, correlationId, clientRequestId, clientApplicationId,
serviceRequestId, Deployment, RoleInstance, TaskName, authorizationAction
//| where clientRequestId =~ "9b98e4f4-d2c4-46f3-8eaf-02997d33603c"
//| where (clientRequestId =~ "9073d335-71d3-46d6-a390-221a2721b7ef" or correlationId =~ "4ee75fc4-ca49-
40fb-ac3b-95d2f4901a30")
| limit 1000

JobTraces
| where TIMESTAMP >= datetime(2017-02-09 13:00:00Z)
| where TIMESTAMP <= datetime(2017-02-10 10:00:00Z)
| where subscriptionId =~ "Input SubscriptionId here"
| where correlationId =~ "4ee75fc4-ca49-40fb-ac3b-95d2f4901a30"
```

Monitoring delays:

```
MonDmDbHadrReplicaStates
| where TIMESTAMP >= datetime(2018-03-06 06:00:00Z)
| where TIMESTAMP <= datetime(2018-03-07 15:00:00Z)
| where AppName =~ "e11dfd6f184f"
| where redo_queue_size > 0
//| where group_database_id =~ "85EEA3FB-D45B-486D-8496-EE0BEB64C0E3"
//| where redo_queue_size > 10000000
//| where is_local == 1 and is_primary_replica == 0 // query secondary ony
| summarize min(redo_queue_size) , max(last_redone_time), count(TIMESTAMP), max(last_hardened_time) ,
max(last_redone_time) by AppName, NodeName, ClusterName, group_database_id, replica_id, last_redone_lsn,
TIMESTAMP
| extend physical_database_id = group_database_id
| extend duration = max_last_hardened_time - max_last_redone_time
| extend startTime = max_last_redone_time
| project TIMESTAMP, AppName, NodeName, physical_database_id, min_redo_queue_size, last_redone_lsn,
max_last_redone_time, max_last_hardened_time, duration, startTime
| order by TIMESTAMP desc
```

```
MonDmDbHadrReplicaStates
| where TIMESTAMP >= datetime(2018-03-07 10:00:00Z)
| where TIMESTAMP <= datetime(2018-03-08 18:00:00Z)
| where AppName =~ "e11dfd6f184f"
| where redo_queue_size > 0
| where group_database_id =~ "85EEA3FB-D45B-486D-8496-EE0BEB64C0E3"
| extend physical_database_id = group_database_id
| extend duration = last_hardened_time - last_redone_time
| extend secondary_lag_minutes = secondary_lag_seconds / 60.0
| project TIMESTAMP, AppName, NodeName, replica_id, physical_database_id, is_local, is_primary_replica,
is_forwarder, redo_queue_size, last_hardened_lsn, last_commit_lsn, last_redone_lsn, last_redone_time,
last_hardened_time, duration, secondary_lag_seconds, secondary_lag_minutes
| order by TIMESTAMP desc
| limit 1000
```

How good have you found this content?