Useful Kusto Queries

Last updated by | Lisa Liu | Nov 6, 2020 at 10:34 AM PST

Useful Kusto Queries

Tuesday, May 30, 2017
9:50 AM

Active connections

Check the number of active connection, this should not exceed the server limits MySQL limits PostgreSQL limits

```
MonRdmsServerMetrics
| where TIMESTAMP >= ago(7d)
| where LogicalServerName =~ "{ServerName}"
| where metric_name == "active_connections"
| summarize sum(metric_value) by bin(originalEventTimestamp, 1s), metric_name
| render timechart
```

Connection Status

Any connection with Error other than 0 means it ran into issue, list of error codes and states can be found here: Error codes

```
MonLogin
| where (logical_server_name =~ "{ServerName}" )
| where TIMESTAMP >= ago(7d)
| project originalEventTimestamp , database_name , AppName , MachineName , package , event, error , state, is_user_error , peer_address, kill_reason, ac
client_connection_id, os_error
```

Graph failures for dropped connections

```
MonLogin
| where (logical_server_name =~ "{ServerName}" )
| project originalEventTimestamp , database_name , AppName , MachineName , package , event, error , state, is_user_error , peer_address, kill_reason, ac
client_connection_id, os_error, is_success
| summarize count() - sum(is_success)  , sum(is_success) by bin(originalEventTimestamp, 1m)
```

Hung

Check if the server is in Hung state and if the result was "1" please escalate to the PG

```
MonRdmsInstanceAgentInstanceCounter
| where AppTypeName == "Worker.PAL.MySQL" and LogicalServerName == "{LogicalServerName}"
| where TIMESTAMP >= ago(1d)
| project TIMESTAMP, CounterName
| extend HasConnectionCounters = (CounterName == "Connections" or CounterName == "Queries")
| extend HasStorageCounters = (CounterName contains "storage_used")
| summarize sum(HasConnectionCounters), sum(HasStorageCounters) by bin(TIMESTAMP, 5m)
| extend Hang = sum_HasConnectionCounters <= 0 and sum_HasStorageCounters > 0
| project TIMESTAMP, Hang
```

SandBox

Sandbox for mysql is the server logs for MySQL servers, you can find error logs and server backups and all other logs here, it is important to review a
narrow timestamps to get a more granular look at the telemtry

```
MonRdmsMySqlSandbox
| where LogicalServerName == "{ServerName}"
| where TIMESTAMP >= datetime(2019-05-14 10:59:50) and TIMESTAMP <= datetime(2019-05-14 19:05:50)
//| where text contains "your query here"
```

Sandbox PostgreSQL is the server logs for PostgreSQL servers, you can find error logs and server backups and all other logs here, it is important to rev
narrow timestamps to get a more granular look at the telemtry

```
MonRdmsPgSqlSandbox
| where LogicalServerName == "{ServerName}"
| where TIMESTAMP >= datetime(2019-05-06 10:59:50) and TIMESTAMP <= datetime(2019-05-06 11:01:50)
//| where text contains "your query here"
```

Connection Dropped between Gateway and Host

check if the process logging finish on host and for those bring the processes that did not have gateway successful

```
MonLogin
| where TIMESTAMP >= datetime({StartTime}) and TIMESTAMP <= datetime({EndTime})
| where (logical_server_name =~ "{ServerName}")
| where event == "process_login_finish" and AppTypeName == "Host.MySQL"
| project connection_id, event, AppTypeName, originalEventTimestamp
| join kind=leftanti (
MonLogin
| where TIMESTAMP >= datetime({StartTime}) and TIMESTAMP <= datetime({EndTime})
| where (logical_server_name =~ "{ServerName}")
| where event == "process_login_finish" and AppTypeName == "Gateway.MySQL"
| project connection_id, event, AppTypeName, originalEventTimestamp) on connection_id
```

number of connections from a specific IP address

This will list connections from a specific IP address which will help you understand trends for a specific IP if failed all the times or intermittently

```
MonLogin
| where (logical_server_name =~ "{ServerName}" )
| where TIMESTAMP ==
| where AppTypeName == "Host.MySQL"
| where event == "process_login_finish"
| summarize count() by bin (originalEventTimestamp, 1d)
```

Summary for the connection to the server at a specific time range

```
MonLogin
| where TIMESTAMP >= ago(2h)
| where logical_server_name =~ "{ServerName}"
| where database_name =~ "{DatabaseName}"
| where event == "process_login_finish"
| summarize count() by error,  state, package, database_name, peer_address, is_user_error
```

Created with Microsoft OneNote 2016.

How good have you found this content?

🙂 🙁