# Maintenance event outside Maintenance Window

Last updated by | Vitor Tomaz | Jun 8, 2022 at 5:33 AM PDT

#### **Contents**

- Maintenance Event outside Maintenance Window
  - Probable reasons
- Recommended Steps
- Kusto Telemetry

## Maintenance Event outside Maintenance Window

### Probable reasons

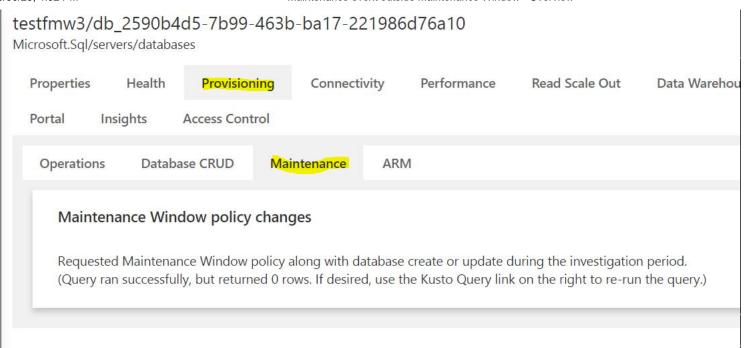
If noticed connectivity glitch or unavailability signal outside of the maintenance window? It may happen due to:

- Hardware failure (unplanned failover)
- Instance management operation ☑
- <u>User-initiated failover</u> ☑
- In very rare circumstances, due to urgent security patching that couldn't be postponed until the next maintenance window.

# **Recommended Steps**

- Check the <u>Resource Health history</u> ☑ of your database for more details
- Check ASC to check the maintenance window that the customer has set

You will receive information on the Maintenance Window set by the customer during the database create or update.

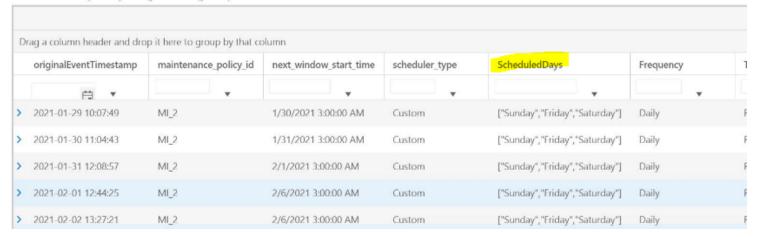


#### Maintenance Policy Operations

Maintenance Policy updates during instance create or update during the investigation period (Query ran successfully, but returned 0 rows. If desired, use the Kusto Query link on the right to re-run the query.)

#### Maintenance Policy History

Maintenance Policy history during the investigation period



• Check for any errors during the FMW setup

#### Kusto Query 🙂 🔅 Maintenance Window policy changes Requested Maintenance Window policy along with database create or update during the investigation period. MaintenanceWindow st\_id operation\_type Result error\_code 36A6-4884--8A7C-UpdateLogicalDatabase DB 2 success D0B54AD5 3FE9-E8A4-984B-UpdateLogicalDatabase DB\_1 success 206F8D2 D753-EE2A-B63C-UpdateLogicalDatabase System Default success EB1C4DA

# **Kusto Telemetry**

(Note: Please use ASC as the primary source for troubleshooting)

1 - 3 of 3 items

For SQL DB:

```
let starttime = datetime('{StartTime}');
            let endtime = datetime('{EndTime}');
            let subscription = '{SubscriptionId}';
            let server = '>{ServerName}<';</pre>
            let database = '>{DatabaseName}<';</pre>
            MonManagementOperations
              where originalEventTimestamp >= starttime and originalEventTimestamp <= endtime
              where subscription id =~ subscription or SubscriptionId =~ subscription
              where event == 'management operation start'
              where operation type == 'UpdateLogicalDatabase' or operation type == 'CreateLogicalDatabase'
              where operation parameters has '<RequestedMaintConfigInternalId>'
              where operation parameters contains server and operation parameters contains database
              extend MaintenanceWindow = parse xml(operation parameters)['InputParameters']['RequestedMaintCon
              project PreciseTimeStamp, request id, operation type, MaintenanceWindow
            | join kind = leftouter
            MonManagementOperations
              where originalEventTimestamp >= starttime
              where subscription id =~ subscription or SubscriptionId =~ subscription
              where event != 'management operation start' and event != 'management operation released'
              where operation type == 'UpdateLogicalDatabase' or operation type == 'CreateLogicalDatabase'
              where operation parameters contains server and operation parameters contains database
            project PreciseTimeStamp, request id, event, error code, error message, operation result
            ) on request id
            | extend Result = replace(@'management_operation_(.*)', @'\1', event)
            | project StartTime = PreciseTimeStamp, EndTime = PreciseTimeStamp1, request_id, operation_type, M
            order by StartTime asc
```

For Elastic Pool

```
let starttime = datetime('{StartTime}');
let endtime = datetime('{EndTime}');
 let subscription = '>{SubscriptionId}<';</pre>
 let server = '>{ServerName}<';</pre>
 let pool = '>{ElasticPoolName}<';</pre>
 MonManagementOperations
 | where originalEventTimestamp >= starttime and originalEventTimestamp <= endtime
 | where operation_type == 'UpdateLogicalElasticPool' or operation_type == 'CreateLogicalElasticPool'
 | where event == 'management_operation_start'
 where operation_parameters contains subscription
 | where operation parameters contains server and operation parameters contains pool
 | where operation parameters has '<RequestedMaintConfigInternalId>' or operation parameters has '<MaintConfig
 extend InputArgumentsParsed = parse_xml(operation_parameters)['InputParameters']
 | extend MaintenanceWindow = coalesce(InputArgumentsParsed['RequestedMaintConfigInternalId'], InputArgumentsP
 | project PreciseTimeStamp, request_id, operation_type, operation_parameters, MaintenanceWindow
 | join kind = leftouter
 MonManagementOperations
 | where originalEventTimestamp >= starttime
 | where operation_type == 'UpdateLogicalElasticPool' or operation_type == 'CreateLogicalElasticPool'
 | where event != 'management_operation_start' and event != 'management_operation_released'
 | where operation_parameters contains subscription
 | where operation parameters contains server and operation parameters contains pool
 | project PreciseTimeStamp, request_id, event,error_code, error_message, operation_result
 ) on request_id
 extend Result = replace(@'management operation (.*)', @'\1', event)
 | project PreciseTimeStamp, PreciseTimeStamp1, request_id, operation_type, MaintenanceWindow, Result, error_c
 | project-rename StartTime = PreciseTimeStamp, EndTime = PreciseTimeStamp1
 order by StartTime asc
```

For SQL MI

```
let starttime = datetime('{StartTime}');
            let endtime = datetime('{EndTime}');
            let subscription = '{SubscriptionId}';
            let MI = '>{ServerName}<';</pre>
            MonManagementOperations
              where originalEventTimestamp >= starttime and originalEventTimestamp <= endtime</pre>
              where subscription id =~ subscription or SubscriptionId =~ subscription
              where event == 'management operation start'
              where operation parameters has '<MaintenancePolicyId>'
              where operation parameters contains MI
              project originalEventTimestamp, request id, operation type, operation parameters
              join kind = leftouter
            MonManagementOperations
              where originalEventTimestamp >= starttime
              where subscription id =~ subscription or SubscriptionId =~ subscription
              where event != 'management operation start'
              where operation parameters has '<MaintenancePolicyId>'
              where operation parameters contains MI
              project originalEventTimestamp, request id, event, error code, error message, operation result
            ) on request id
              extend Result = replace(@'management operation (.*)', @'\1', event)
              extend Input = tostring(parse xml(operation parameters))
              extend InstanceOperationType = parse json(Input).InputParameters.OperationType
              extend MaintenancePolicyId = tostring(parse_json(Input).InputParameters.MaintenancePolicyId)
              project originalEventTimestamp, originalEventTimestamp1, request_id, InstanceOperationType, Main
              project-rename StartTime = originalEventTimestamp, EndTime = originalEventTimestamp1
              order by StartTime asc
             join kind=leftouter (
            MonManagement
              where event == 'maintenance window info'
              where isnotempty(maintenance policy id)
              summarize arg max(TIMESTAMP, scheduler pattern) by maintenance policy id
              extend SchedulerPattern = parse xml(scheduler pattern)
              extend ScheduledDays = tostring(SchedulerPattern.WindowScheduler.ScheduledDays.DayOfWeek)
              extend Frequency = SchedulerPattern.WindowScheduler.Frequency
              extend TimeZone = SchedulerPattern.WindowScheduler.TimeZone
              extend Time = SchedulerPattern.WindowScheduler.Time
              extend Duration = SchedulerPattern.WindowScheduler.Duration
              project maintenance_policy_id, ScheduledDays, Frequency, Time, Duration
              on $left.MaintenancePolicyId == $right.maintenance_policy_id
              project StartTime, EndTime, request_id, InstanceOperationType, Result, MaintenancePolicyId, Sche
              order by StartTime asc
```

## How good have you found this content?

