

# 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](#) 
- [User-initiated failover](#) 
- In very rare circumstances, due to urgent security patching that couldn't be postponed until the next maintenance window.

### Recommended Steps

- Check the [Resource Health history](#)  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.

3/30/23, 4:02 PM

Maintenance event outside Maintenance Window - Overview

testfmw3/db\_2590b4d5-7b99-463b-ba17-221986d76a10

Microsoft.Sql/servers/databases

PropertiesHealthProvisioningConnectivityPerformanceRead Scale OutData Warehouse

PortalInsightsAccess Control

OperationsDatabase CRUDMaintenanceARM

Maintenance Window policy changes

Requested Maintenance Window policy along with database 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 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

Drag a column header and drop it here to group by that column						
originalEventTimestamp	maintenance_policy_id	next_window_start_time	scheduler_type	ScheduledDays	Frequency	
<div>2021-01-29 10:07:49</div>	<div>ML_2</div>	<div>1/30/2021 3:00:00 AM</div>	<div>Custom</div>	<div>["Sunday","Friday","Saturday"]</div>	<div>Daily</div>	
<div>2021-01-30 11:04:43</div>	<div>ML_2</div>	<div>1/31/2021 3:00:00 AM</div>	<div>Custom</div>	<div>["Sunday","Friday","Saturday"]</div>	<div>Daily</div>	
<div>2021-01-31 12:08:57</div>	<div>ML_2</div>	<div>2/1/2021 3:00:00 AM</div>	<div>Custom</div>	<div>["Sunday","Friday","Saturday"]</div>	<div>Daily</div>	
<div>2021-02-01 12:44:25</div>	<div>ML_2</div>	<div>2/6/2021 3:00:00 AM</div>	<div>Custom</div>	<div>["Sunday","Friday","Saturday"]</div>	<div>Daily</div>	
<div>2021-02-02 13:27:21</div>	<div>ML_2</div>	<div>2/6/2021 3:00:00 AM</div>	<div>Custom</div>	<div>["Sunday","Friday","Saturday"]</div>	<div>Daily</div>	

- Check for any errors during the FMW setup

Maintenance Window policy changes

[Kusto Query](#) 😊 ☹️

Requested Maintenance Window policy along with database create or update during the investigation period.

📄

🔍

☰

⚙️

st_id	operation_type	MaintenanceWindow	Result	error_code
<div><div></div><div>▼</div></div>	<div><div></div><div>▼</div></div>	<div><div></div><div>▼</div></div>	<div><div></div><div>▼</div></div>	<div><div></div></div>
36A6-4884-8A7C-00B54AD53FE9-E8A4-984B-2206F8D20753-EE2A-B63C-EB1C4DA	UpdateLogicalDatabase	DB_2	success	
	UpdateLogicalDatabase	DB_1	success	
	UpdateLogicalDatabase	SystemDefault	success	

1 - 3 of 3 items

## Kusto Telemetry

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

For SQL DB:

```

let starttime = datetime('{StartTime}');
let endtime = datetime('{EndTime}');
let subscription = '{SubscriptionId}';
let server = '>{ServerName}<';
let database = '>{DatabaseName}<';
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}<';
let server = '>{ServerName}<';
let pool = '>{ElasticPoolName}<';
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}<';
MonManagementOperations
| where originalEventTimestamp >= starttime and originalEventTimestamp <= endtime
| 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?**

