Maintenance Windows - Telemetry & Resources

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

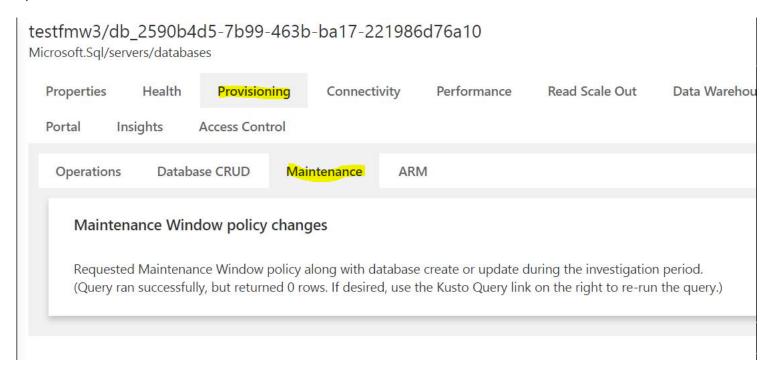
Contents

- ASC Tab
- Kusto Telemetry
- List of Error messages
- ICM Queue for Maintenance Window Issues
- Public Docs
- CSS Mentoring Series Maintenance Windows & Advance ...

ASC Tab

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.

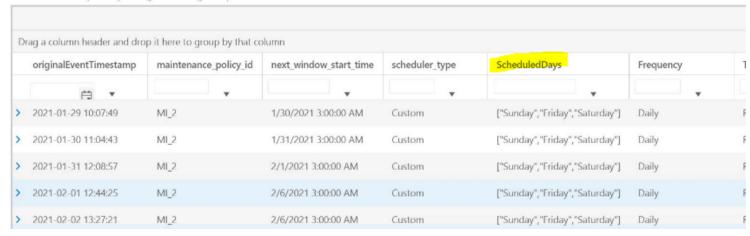


Maintenance Policy Operations

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



Kusto Telemetry

Note: Please use ASC as the primary source

Elastic Pools:

```
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
```

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
```

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
```

List of Error messages

(CLI) Old version of client module	Unrecognized arguments: maint-config-id
(PowerShell)Old version of client module	Set-AzSqlInstance: A parameter cannot be found that matches parameter name 'MaintenanceConfigurationId'
(CLI) Non-existing window id	Invalid value given for parameter MaintenancePolicyld. Specify a valid parameter value.
(PowerShell) Non-existing window id	Invalid value given for parameter MaintenancePolicyld. Specify a valid parameter value
MW not enabled in Azure region(CLI)	Maintenance windows are not available in this region
MW not enabled in Azure region(PowerShell)	Maintenance schedule MI_1 is temporary not available in this region

ICM Queue for Maintenance Window Issues

https://portal.microsofticm.com/imp/v3/administration/teamdashboard/details?id=54216

Public Docs

https://aka.ms/SQLMaintenanceWindow I https://aka.ms/SQLAdvNotification I

CSS Mentoring Series Maintenance Windows & Advance Notification

<u>Deck</u> ☑

How good have you found this content?



