Check enabled Feature Switches and Trace Flags for server

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:26 AM PST

Contents

Check what Feature Switches have been enabled for an Azure SQL Database server

This is a "How-To" article which helps you to see individual feature configuration options for a SQL server.

Issue

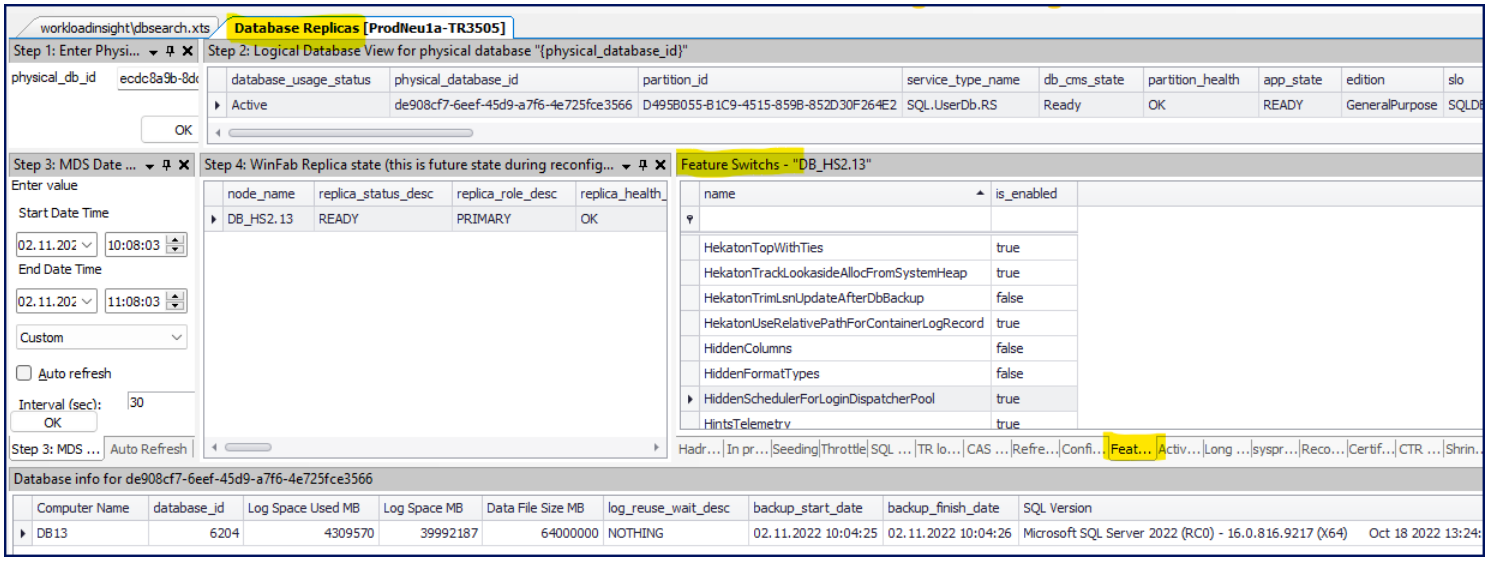
Feature Switches can be set by the PG to mitigate issues or to force specific database behaviour based on a customer scenario. For example, the "SQL.Config_DMVCollector_ViewOverrides" feature switch was set to DISABLED by the PG for a server that hosted several Elastic Pools with thousands of databases in total. Some DMVs that depend on live telemetry became very slow, allocated excessive resources and thus caused database unavailability. This feature switch disabled the DMV data collection for those busy servers (the DMV issue has been fixed now). Another scenario is if a planned upgrade maintenance contained a feature-based regression; it is then often easier to deploy a disabling feature switch than rolling back the regressed code itself.

Investigation / Analysis

You can check for feature switches either through XTS or Kusto.

Feature Switches in XTS

You can see the active Feature Switch configuration through the `Database Replicas.xts` view. The easiest way to open this is to search for the server through `DBSearch.xts`, then select one of the databases and click on the `Database Replicas` link. The output looks similar to this:



Feature Switches in Kusto

Note that Kusto will only show the non-default feature switches; you will only see configuration settings that are specific for a server and that deviate from its default configuration. As an advantage, Kusto will also return the enabled SQL Server trace flags which you don't see on XTS.

Use this simple version of the query if you know the server name. Remove the `Property_Name` line if you don't know the exact name of the feature switch.

```
// check feature switch for one server
let startTime = datetime(2022-11-01 00:00:00);
let endTime = datetime(2022-11-01 23:00:00);
let srv = "servername";
MonConfigLogicalServerOverrides
| where TIMESTAMP >= startTime
| where TIMESTAMP <= endTime
| where Logical_Server_Name == srv
//| where Property_Name has "SQL.Config_FeatureSwitches_HiddenSchedulerForLoginDispatcherPool"
//| where Property_Name has "HiddenSchedulerForLoginDispatcherPool"
//| where Property_Name has "DMVCollector"
| project TIMESTAMP, code_package_version, Logical_Server_Name, Property_Name, Property_Value, Category
| summarize by Logical_Server_Name, code_package_version, Category, Property_Name,  Property_Value
```

Sample output:

Logical_Server_Name	code_package_version	Category	Property_Name
servername	16.0.816.9217-DB-08c62705	LogicalServerOverride	SQL.Config_FeatureSwitches_
servername	16.0.816.9217-DB-08c62705	LogicalServerOverride	SQL.Config_DMVCollector_Vi
servername	16.0.816.9217-DB-08c62705	LogicalServerOverride	SQL.Config_RgSettings_DiffBa
servername	16.0.816.9217-DB-08c62705	LogicalServerOverride	SQL.Config_BackupService_D
servername	16.0.816.9217-DB-08c62705	LogicalServerOverride	SQL.Config_FeatureSwitches_
servername	16.0.816.9217-DB-08c62705	LogicalServerOverride	SQL.Config_FeatureSwitches_
servername	16.0.816.9217-DB-08c62705	TraceFlag	4138

If you need to investigate several servers in the same subscription, but only know the Subscription ID and/or a single server name, you can retrieve a list of servers in a subscription through the following query:

```
// list all servers in the subscription
let startTime = datetime(2022-11-02 00:00:00);
let endTime = datetime(2022-11-02 23:00:00);
let subID = "18CE6AE1-4706-4A18-B3CD-546DC78BD4B2";
let srv = "servername";
MonAnalyticsDBSnapshot
| where  TIMESTAMP >= startTime
| where  TIMESTAMP <= endTime
| where customer_subscription_id =~ subID
//| where logical_server_name =~ srv
//| order by TIMESTAMP desc
//| limit 1
| distinct logical_server_name
```

You can use the folloing Join query to feed the server list directly to the `MonConfigLogicalServerOverrides` Kusto table:

```
// list all servers in the subscription and feed them to the FS table
let startTime = datetime(2022-11-01 00:00:00);
let endTime = datetime(2022-11-02 23:00:00);
let subID = "571f6eff-f864-41ef-9f93-0f7fd9628e13";
MonAnalyticsDBSnapshot
| where  TIMESTAMP >= startTime
| where  TIMESTAMP <= endTime
| where customer_subscription_id =~ subID
| distinct logical_server_name
| join kind=inner
(
    MonConfigLogicalServerOverrides
    | where TIMESTAMP >= startTime
    | where TIMESTAMP <= endTime
    | where Property_Name has "HiddenSchedulerForLoginDispatcherPool"
    | project TIMESTAMP, code_package_version, Logical_Server_Name, Property_Name, Property_Value, Category
    | summarize by Logical_Server_Name, code_package_version, Category, Property_Name,  Property_Value
) on $left.logical_server_name == $right.Logical_Server_Name
| project logical_server_name, code_package_version, Category, Property_Name, Property_Value
| order by logical_server_name asc
```

Sample output:

logical_server_name	code_package_version	Category	Property_Name
servername1	16.0.816.9217-DB-08c62705	LogicalServerOverride	SQL.Config_FeatureSwitches_H
servername2	16.0.816.9217-DB-08c62705	LogicalServerOverride	SQL.Config_FeatureSwitches_H
servername3	16.0.816.9217-DB-08c62705	LogicalServerOverride	SQL.Config_FeatureSwitches_H

How good have you found this content?