# Description

Zabbix templates for Microsoft SQL Server (MS SQL).

# Features

* MS SQL performance counters.
* MS SQL instance Low Level Discovery.
* MS SQL database Low Level Discovery.
* MS SQL agent job Low Level Discovery.
* MS SQL database mirroring monitoring.
* MS SQL Always On monitoring.
* MS SQL Log Shipping monitoring.

# Supported versions

Tested on Microsoft SQL Server 2012, 2014 and 2016. It may work with earlier versions, but some items (with missing performance counters) may be unsupported. For the extensive overview on the performance counters difference between MS SQL 2008 and MS SQL 2012 you can read here (https://blog.dbi-services.com/sql-server-2012-new-perfmon-counters/).

Tested on Zabbix 3.4.0. It may work with earlier versions, but some items (for example service.info[service,<param>]) may be unsupported. The template was started on Zabbix 2.4.0 but after each new Zabbix version, objects were modified or new things were added.

# Assumptions

These templates are not a one-click solution. It may require some tinkering, so I assume that:

* You are familiar with a Microsoft Window OS.
* You are familiar with a Microsoft SQL server.
* You are familiar with a Zabbix.
* You have a correctly working Zabbix server and Zabbix agents.
* You have tested other Zabbix template with an item type “Active” and it works.

# Template includes

Templates:

* “Template SQL Server DE Baseline.xml” – Template for Microsoft SQL Server Database Engine.
* “Template SQL Server SA Baseline.xml” – Template for Microsoft SQL Server Agent.

Value mapping:

* “SQL Agent Job status.xml” – Zabbix value mapping for Microsoft SQL Server Agent Job status.
* “SQL Database status.xml” – Zabbix value mapping for Microsoft SQL Server Database status.

Scripts:

* “Discovery.mssql.databasename.ps1”– PowerShell script for Low Level Discovery.
* “Discovery.mssql.databasestatus.ps1”– PowerShell script for Low Level Discovery.
* “Discovery.mssql.instanceagentname.ps1”– PowerShell script for Low Level Discovery.
* “Discovery.mssql.instancename.ps1”– PowerShell script for Low Level Discovery.
* “Discovery.mssql.jobname.ps1”– PowerShell script for Low Level Discovery.
* “Discovery.mssql.jobstatus.ps1”– PowerShell script for Low Level Discovery.

User parameters”

* “zabbix\_agentd.userparams.conf” – Example of user parameters.

# Deployment. Step by step

1. Import templates via Configuration >> Templates:

* “Template Microsoft SQL Server DE Baseline.xml”
* “Template Microsoft SQL Server SA Baseline.xml”

1. Import value mappings via Administration >> General >> Value mapping:

* “SQL Agent Job status.xml”
* “SQL Database status.xml”

1. Copy PowerShell scripts (\*.ps1) to a location a Zabbix Agent can access (by default “C:\Zabbix\bin”).
2. Update UserParameter in zabbix\_agentd.conf with a provided example. Change paths to a PowerShell scripts (\*.ps1) if needed.
3. Grant rights for Zabbix Agent service account. It needs read rights on tables msdb.dbo.sysjobhistory, msdb.dbo.sysjobs and msdb.dbo.log\_shipping\_monitor\_secondary. By default, Zabbix Agent service account is NT AUTHORITY\SYSTEM which is already in SQL Server.
4. If you need to monitor mirrored databases or databases in Always On, you will have to give Zabbix Agent’s service account (NT AUTHORITY\SYSTEM by default) sysadmin rights. More about it [here](https://docs.microsoft.com/en-us/sql/relational-databases/system-catalog-views/sys-database-mirroring-transact-sql?view=sql-server-2017).
5. Restart Zabbix Agent.
6. Add templates to a Host.
7. By default, you should add both templates to a host. Unless you have Microsoft SQL Server Express edition. In that case add only “Template SQL Server DE Baseline.xml” template.

# Notes

MS SQL system databases (master, msdb etc.) are discovered and monitored by default. I saw templates, where system databases there excluded from discovery and monitoring, but I am against it. You need to know the status (Online, Offline) and performance parameters of system databases as well.

MS SQL Agent job status is queried from table msdb.dbo.sysjobhistory. If the job is never run and no record exists in this table, Zabbix item will be unsupported.

The triggers settings are configured for an ideal MS SQL performance, which is hardly achieved in a real environment. Therefore, I strongly advise you to review all the triggers, before going live. You can either lower severity or change trigger value.

Items Update interval (30 seconds), History storage period (90 days) and Trend storage period (365 days) are set accordingly to the rule “as much as possible, for as long as possible”. If you have ~10 instances with an average ~8 databases per instance you will be fine. But if you have more than that, you should check “Zabbix calculations for MS SQL template.xlsx”. Enter your values and see how much disk space for a Zabbix database you will need. Also, depending on the number of processed values per second (NVS), you may consider upgrading your Zabbix server or increasing Items Update interval from 30 seconds to any value you feel comfortable. All calculations are based on Zabbix [manual](https://www.zabbix.com/documentation/3.4/manual/installation/requirements) and they provide only average values, not the exact numbers.

# Roadmap

* Fix URL’s in Triggers prototypes. Microsoft updated their URLs and current links leads to a page 404.
* Create a one-click solution to simplify deployment of the scripts.
* Review Triggers prototypes values.
* Create a template for monitoring the MS SQL logs.
* Create a template for monitoring the MS SQL backups.
* Create the templates for monitoring the MS SQL High Availability (Always On, database mirroring etc.) performance.
* Create a template for monitoring the MS SQL security best practices (based on CIS Benchmarks from [www.cisecurity.org](http://www.cisecurity.org) )

# Thanks

MS SQL performance counters selection is based on a 2010 Quest Software poster written by Kevin Kline (MVP) with Brent Ozar (MCM, MVP) and contributions by Christian Bolton (MCM, MVP), Bob Ward (Microsoft), Rod Colledge (MVP), and Raoul Illyaos. You can find original poster by googling “quest sql server performance counters of interest”.

PowerShell scripts are based on the scripts of Nate Jones, Jon Pangburna, Александр Александров (<https://share.zabbix.com/databases/microsoft-sql-server/mssql-2008-2016-multiinstance>).

Thanks for the additional ideas and help with the scripts to Domantas P., Jonas S., Pavel L., Oleg D., Nerijus P., Klaidas I.

# Contacts

Let me know if you find any errors. Or maybe you just have a great idea which really, really must be added to the templates.

mantas.tumenas@gmail.com