axians

Monitoring (Azure)

SQL DB with Zabbix

[TGIF 21-01-2022]

VINCI ENERGIES

Why?

- Baselines
- Issue detection
- History
- What is my instance doing

But, why?

- Proactive management and maintenance
- Tackle problems before they become an issue
- Explain past issues

What

- Azure SQL DB != SQL Server
- Every database is different
- Different interpretations

What are your objectives

- Reliability
- Performance
- Uptime
- Jobs
- Your own objective

How?

- SQL Server Management Studio or Azure Data Studio?
- There's the Azure portal...
- You can use (expensive) software
- You can try the TICK Stack
- Zabbix

axians

Hello!

Reitse Eskens

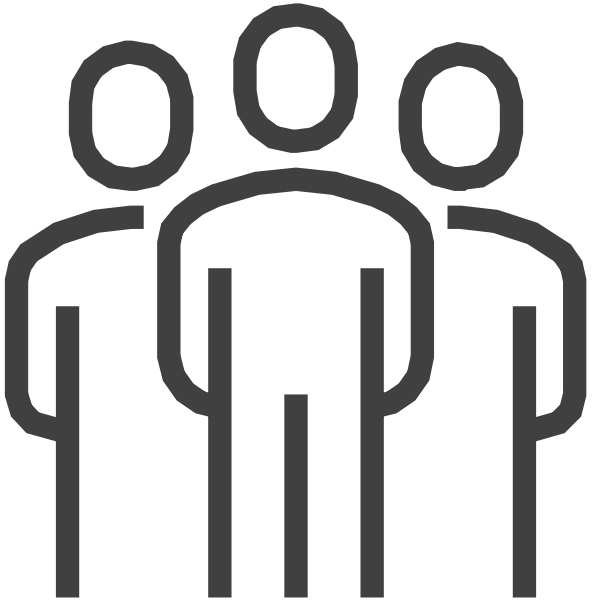
I am here because I love my DBA job and share knowledge.

MCSA, MCSE, MCT and a number of Azure certs

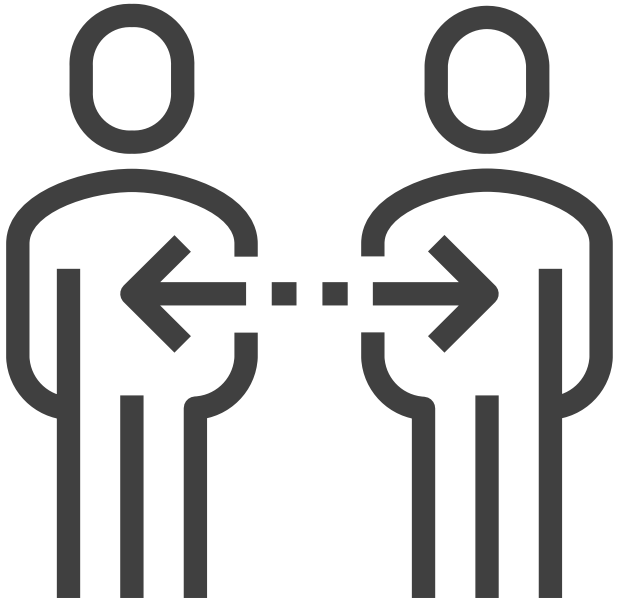
You can reach me at
@2meterdba (twitter)

reitse.eskens@axians.com

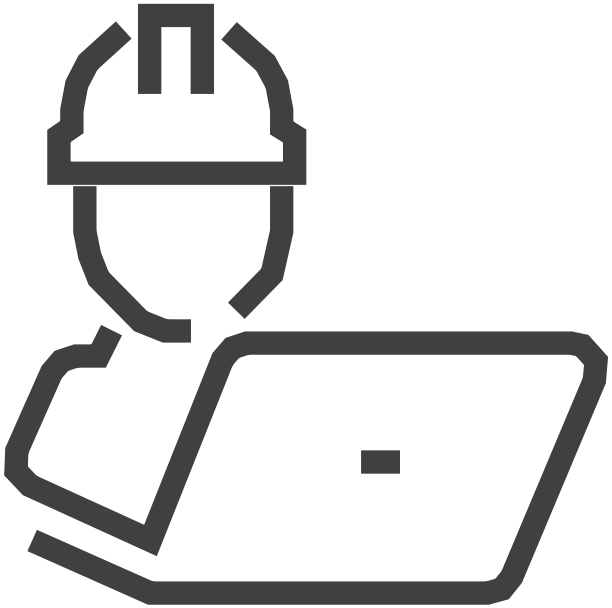
- Server infrastructure monitoring
- Runs on Linux
- Graphical interface
- Many templates from the community



- Might already run in your organisation
- Prebuilt templates for SQL Server, Oracle and a lot more
- Easy to learn and create your own templates
- Show query results on the dashboard



- Installation guides are online, many!
- Configure the firewall on the Linux machine
- Create user accounts with correct rights
- Create your SQL ODBC connection in Linux, take care who can access that folder!




axians
DEMO!

```
sudo su
curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -

curl https://packages.microsoft.com/config/ubuntu/20.04/prod.list > /etc/apt/sources.list.d/mssql-release.list

exit
sudo apt-get update
sudo ACCEPT_EULA=Y apt-get install -y msodbcsql17
# optional: for bcp and sqlcmd
sudo ACCEPT_EULA=Y apt-get install -y mssql-tools
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc
source ~/.bashrc
# optional: for unixODBC development headers
sudo apt-get install -y unixodbc-dev
```

ini

Copy

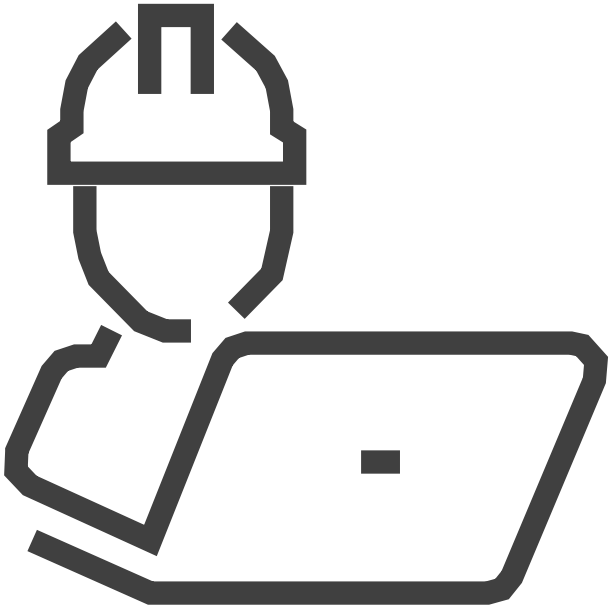
```
# [DSN name]
[MSSQLTest]
Driver = ODBC Driver 17 for SQL Server
# Server = [protocol:]server[,port]
Server = tcp:localhost,1433
#
# Note:
# Port is not a valid keyword in the odbc.ini file
# for the Microsoft ODBC driver on Linux or macOS
#
```

https://docs.microsoft.com/en-us/sql/connect/odbc/linux-mac/connection-string-keywords-and-data-source-names-dsns?view=sql-server-ver15

```
[DEMODB]
Driver = ODBC Driver 17 for SQL Server
Server = MyDemoDB.database.windows.net
Database = myDemoDatabase
Port = 1433
```

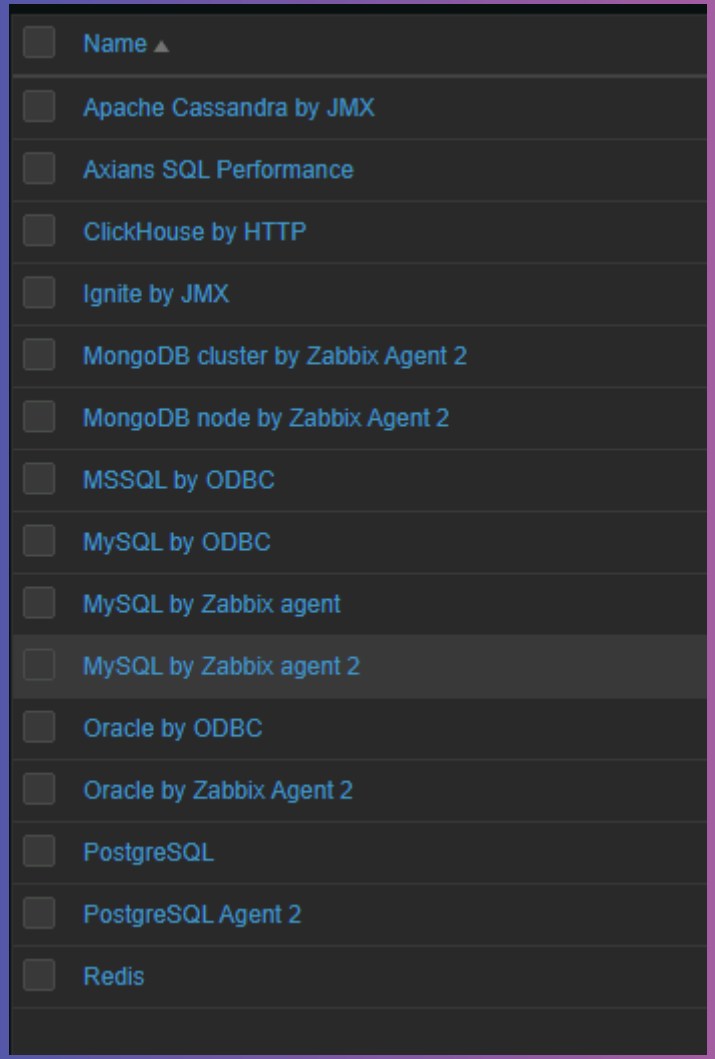
You can add extra options to the ini file to suit your needs

- Most basic things are included
- Check the triggers. They might fire to fast. Or not
- The templates are incomplete, add your own metrics
- Check the configuration of the scripts

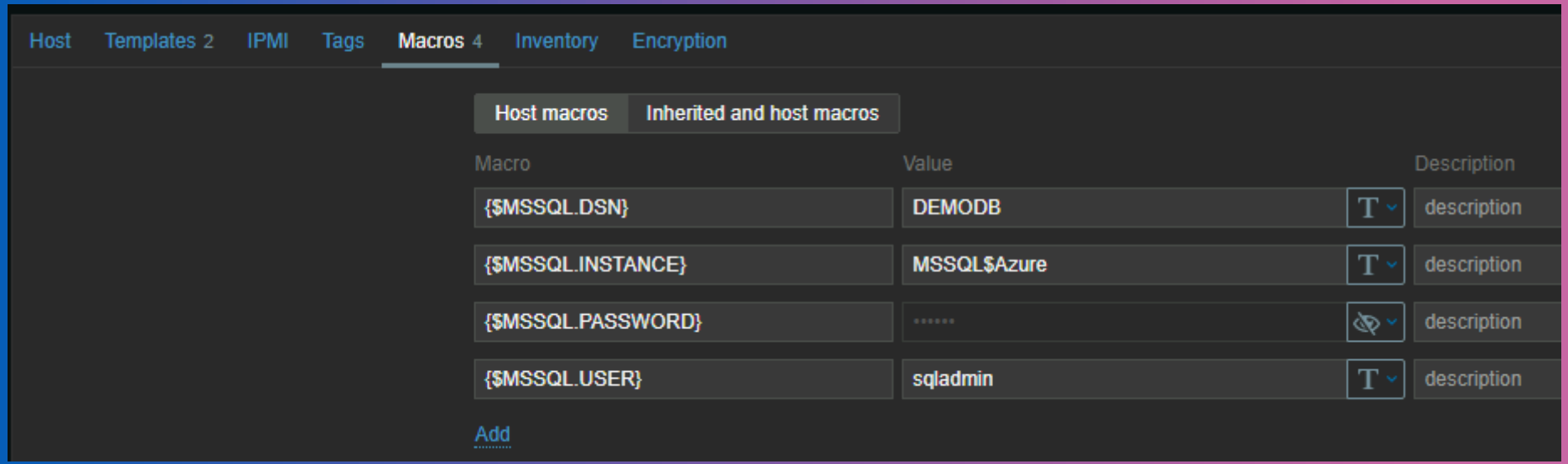



axians
DEMO!

templates


Name ▲
Apache Cassandra by JMX
Axians SQL Performance
ClickHouse by HTTP
Ignite by JMX
MongoDB cluster by Zabbix Agent 2
MongoDB node by Zabbix Agent 2
MSSQL by ODBC
MySQL by ODBC
MySQL by Zabbix agent
MySQL by Zabbix agent 2
Oracle by ODBC
Oracle by Zabbix Agent 2
PostgreSQL
PostgreSQL Agent 2
Redis

connection


Host
Templates 2
IPMI
Tags
Macros 4
Inventory
Encryption
Host macros
Inherited and host macros
Macro
Value
Description
{$MSSQL.DSN}
DEMODB
description
{$MSSQL.INSTANCE}
MSSQL$Azure
description
{$MSSQL.PASSWORD}
description
{$MSSQL.USER}
sqladmin
description
Add

Not everything is supported on Azure

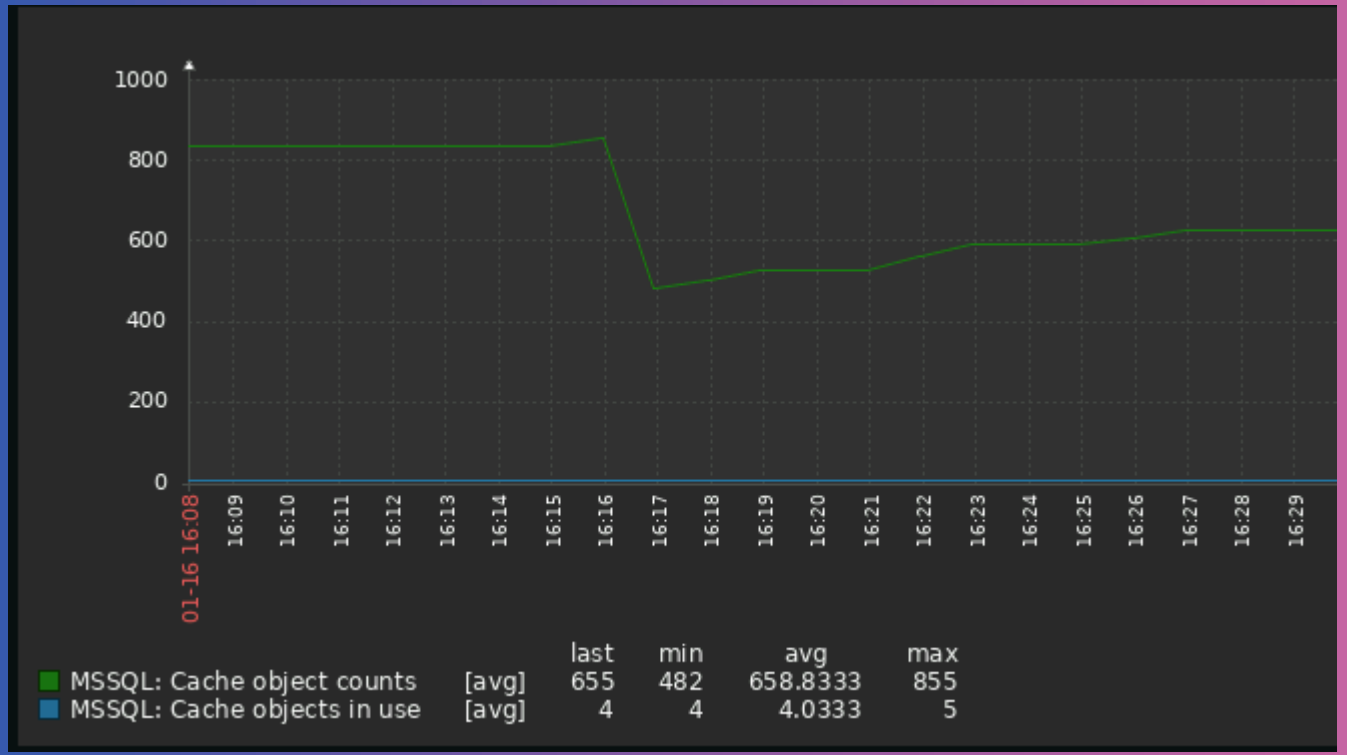
Name ▲	Interval	Type	Status	Info
MSSQL by ODBC: Availability groups discovery	1h	Database monitor	Not supported	i
MSSQL by ODBC: Database discovery	1h	Database monitor	Enabled	
MSSQL by ODBC: Local database discovery	1h	Database monitor	Not supported	i
MSSQL by ODBC: Mirroring discovery	1h	Database monitor	Not supported	i
MSSQL by ODBC: Non-local database discovery	1h	Database monitor	Not supported	i
MSSQL by ODBC: Replication discovery	1h	Database monitor	Not supported	i

Name
MSSQL: Service's TCP port state
MSSQL: Get performance counters
MSSQL: Percent of Adhoc queries running
MSSQL: Full scans to Index searches ratio
MSSQL: Percent of Recompiled Transact-SQL Objects
MSSQL: Total average wait time
MSSQL: Average latch wait time

Items

- MSSQL: Get performance counters: MSSQL: Cache objects in use
- MSSQL: Get performance counters: MSSQL: Database pages
- MSSQL: Get performance counters: MSSQL: Total data file size
- MSSQL: Get performance counters: MSSQL: Checkpoint pages per second
- MSSQL: Get performance counters: MSSQL: Cache pages

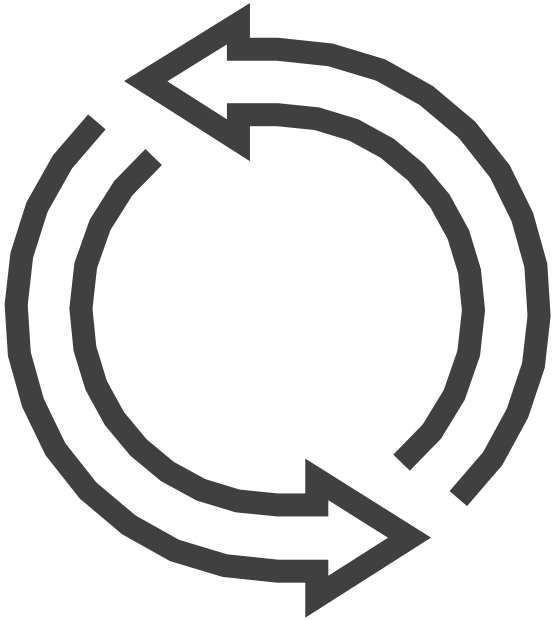
Items


1000
800
600
400
200
0
01-16 16:08
16:09
16:10
16:11
16:12
16:13
16:14
16:15
16:16
16:17
16:18
16:19
16:20
16:21
16:22
16:23
16:24
16:25
16:26
16:27
16:28
16:29
last min avg max
MSSQL: Cache object counts [avg] 655 482 658.8333 855
MSSQL: Cache objects in use [avg] 4 4 4.0333 5

triggers

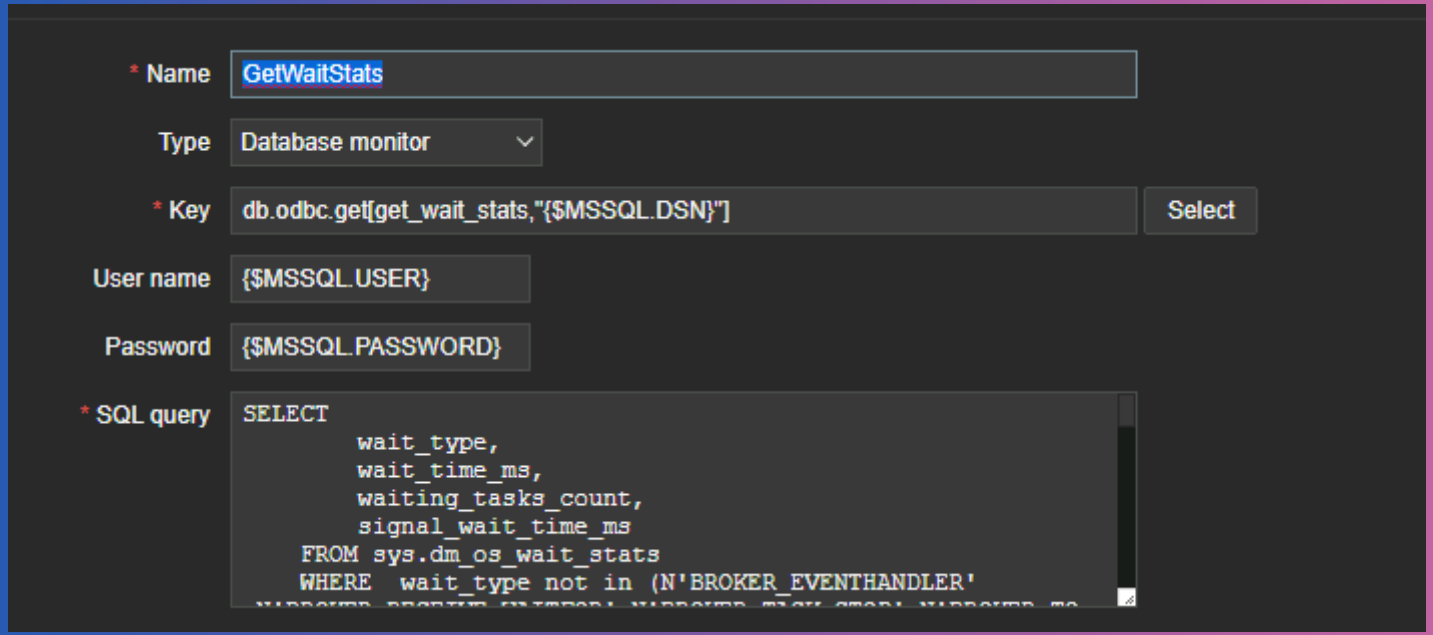
☐	Warning	MSSQL: Number of physical database page writes per second is high (over {$MSSQL.PAGE_WRITES.MAX} for 5m)
☐	High	MSSQL: Percentage of the buffer cache efficiency is low (below {$MSSQL.BUFFER_CACHE_RATIO.MIN.CRIT}% for 5m)
☐	Disaster	MSSQL: Service is unavailable

- Create your own template
- Configure the measurements and triggers
- Add dashboards
- Evaluate the results, get peer reviews and share!

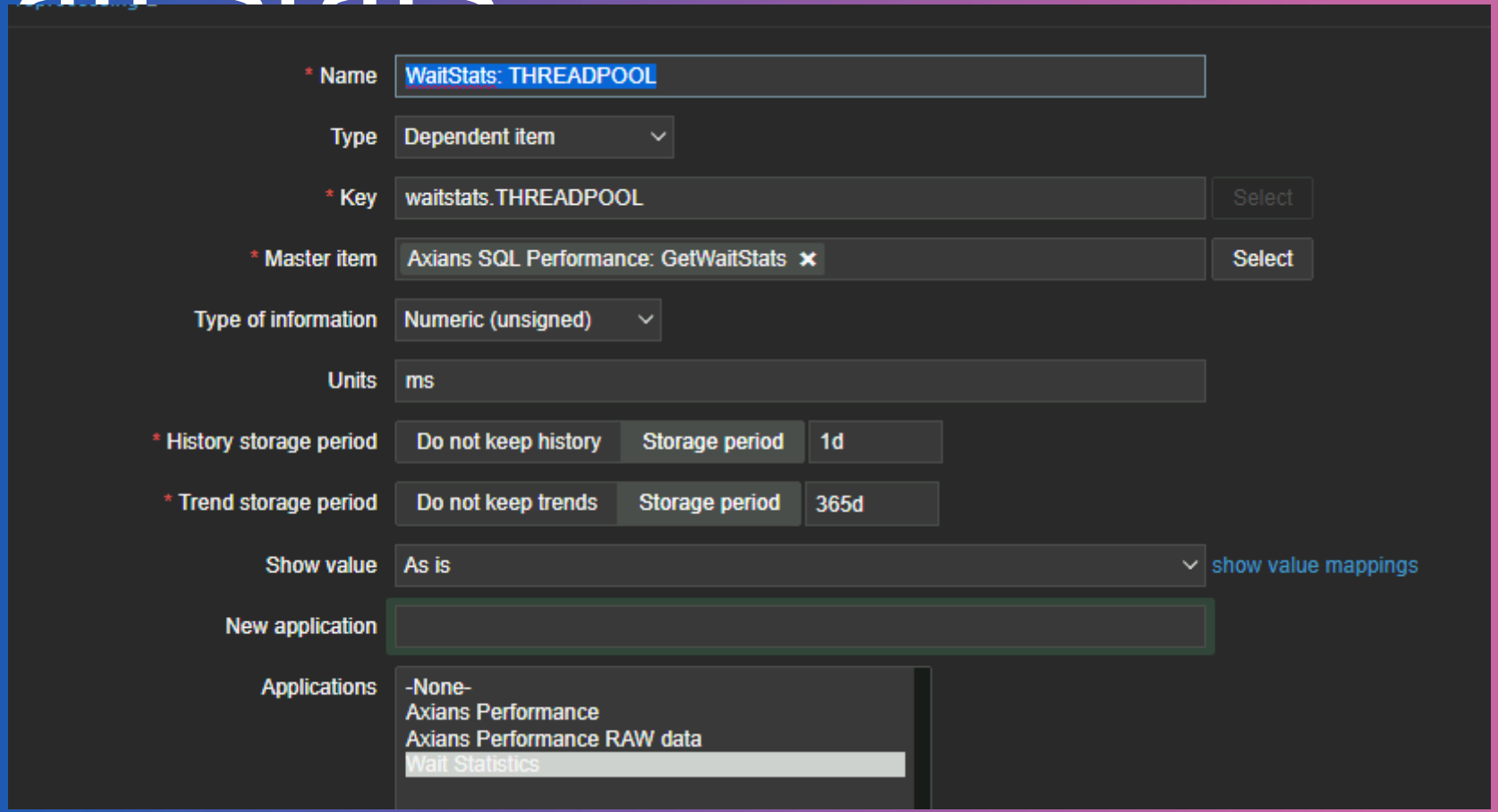



axians
DEMO!

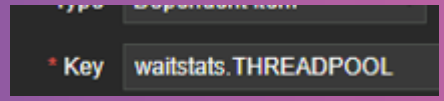
Wait stats

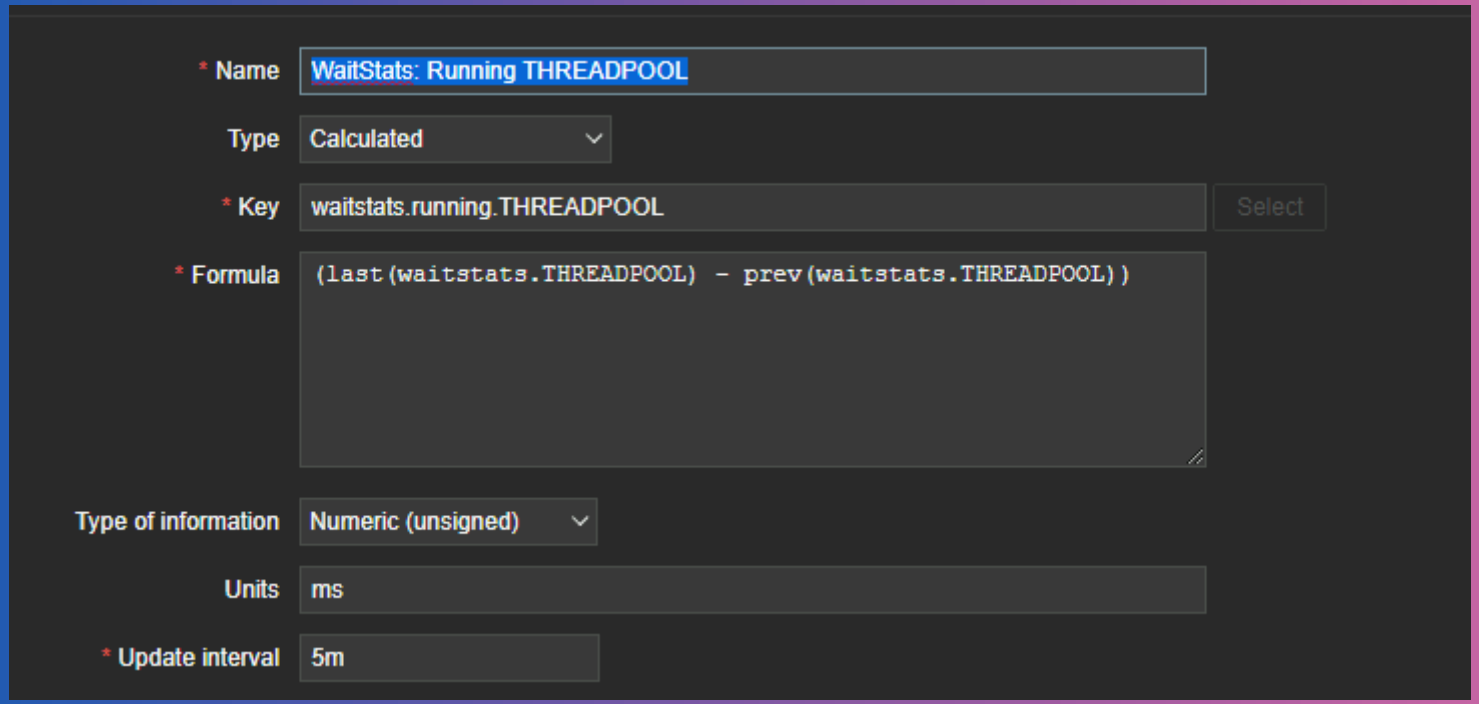

* Name
GetWaitStats
Type
Database monitor
* Key
db.odbc.get[get_wait_stats,"{$MSSQL.DSN}"]
Select
User name
{$MSSQL.USER}
Password
{$MSSQL.PASSWORD}
* SQL query
SELECT
wait_type,
wait_time_ms,
waiting_tasks_count,
signal_wait_time_ms
FROM sys.dm_os_wait_stats
WHERE wait_type not in (N'BROKER_EVENTHANDLER'

Wait stats

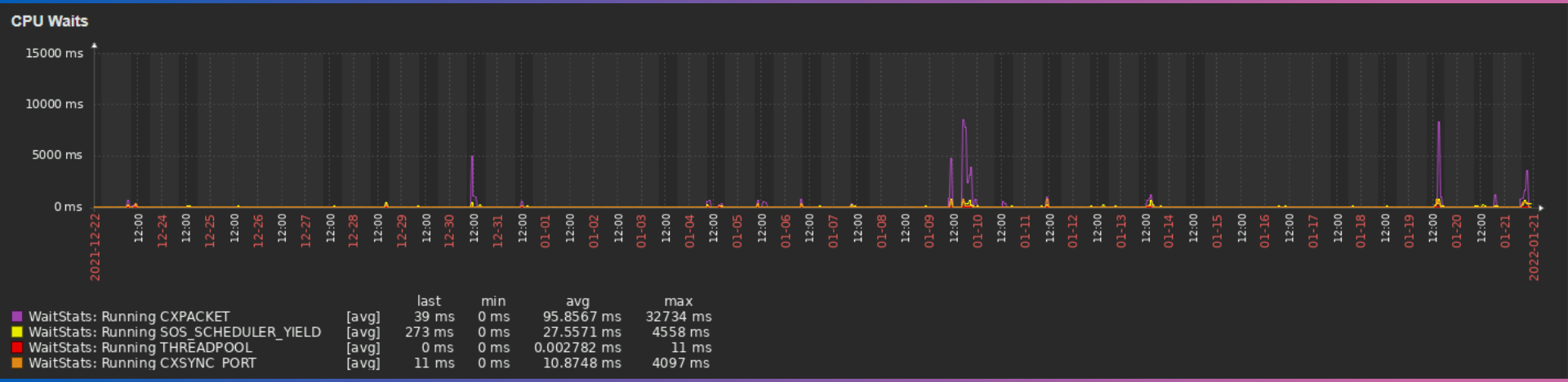

* Name
WaitStats: THREADPOOL
Type
Dependent item
* Key
waitstats.THREADPOOL
Select
* Master item
Axians SQL Performance: GetWaitStats
Select
Type of information
Numeric (unsigned)
Units
ms
* History storage period
Do not keep history
Storage period
1d
* Trend storage period
Do not keep trends
Storage period
365d
Show value
As is
show value mappings
New application
Applications
-None-
Axians Performance
Axians Performance RAW data
Wait Statistics

Wait stats


* Key
waitstats.THREADPOOL


* Name
WaitStats: Running THREADPOOL
Type
Calculated
* Key
waitstats.running.THREADPOOL
Select
* Formula
(last(waitstats.THREADPOOL) - prev(waitstats.THREADPOOL))
Type of information
Numeric (unsigned)
Units
ms
* Update interval
5m

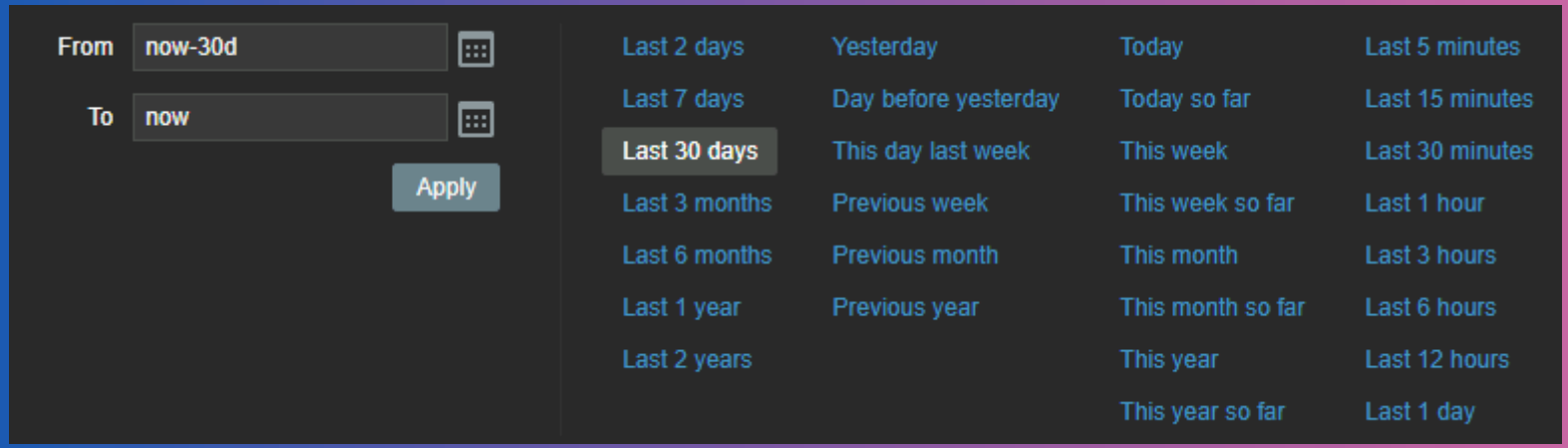
Wait stats


CPU Waits
15000 ms
10000 ms
5000 ms
0 ms
2021-12-22
2022-01-21
last min avg max
WaitStats: Running CXPACKET [avg] 39 ms 0 ms 95.8567 ms 32734 ms
WaitStats: Running SOS_SCHEDULER_YIELD [avg] 273 ms 0 ms 27.5571 ms 4558 ms
WaitStats: Running THREADPOOL [avg] 0 ms 0 ms 0.002782 ms 11 ms
WaitStats: Running CXSYNC_PORT [avg] 11 ms 0 ms 10.8748 ms 4097 ms

Wait stats


cpu cores
20
15
10
5
0
2021-12-22
12-25
12-28
01-01
01-04
01-07
01-10
01-13
01-16
01-19
2022-01-21
last min avg max
cpu limit [avg] 14.2353 2 4.0019 16
cpu load
60 percent
40 percent
20 percent
0 percent
2021-12-22
12-24
12-25
12-26
12-27
12-28
12-29
12-30
12-31
01-01
01-02
01-03
01-04
01-05
01-06
01-07
01-08
01-09
01-10
01-11
01-12
01-13
01-14
01-15
01-16
01-17
01-18
01-19
01-20
2022-01-21
last min avg max
avg cpu percent [avg] 9.6521 percent 0 percent 2.9846 percent 99.67 percent
avg instance CPU Percent [avg] 15.9562 percent 0 percent 5.3305 percent 92.98 percent
CPU Waits
15000 ms
10000 ms
5000 ms
0 ms
2021-12-22
12-24
12-25
12-26
12-27
12-28
12-29
12-30
12-31
01-01
01-02
01-03
01-04
01-05
01-06
01-07
01-08
01-09
01-10
01-11
01-12
01-13
01-14
01-15
01-16
01-17
01-18
01-19
01-20
01-21
2022-01-21
last min avg max
WaitStats: Running CXPACKET [avg] 39 ms 0 ms 95.8567 ms 32734 ms
WaitStats: Running SOS_SCHEDULER_YIELD [avg] 125 ms 0 ms 27.6926 ms 4558 ms
WaitStats: Running THREADPOOL [avg] 0 ms 0 ms 0.002782 ms 11 ms
WaitStats: Running CXSYNC_PORT [avg] 11 ms 0 ms 10.8748 ms 4097 ms

Wait stats


From
now-30d
To
now
Apply
Last 2 days
Last 7 days
Last 30 days
Last 3 months
Last 6 months
Last 1 year
Last 2 years
Yesterday
Day before yesterday
This day last week
Previous week
Previous month
Previous year
Today
Today so far
This week
This week so far
This month
This month so far
This year
This year so far
Last 5 minutes
Last 15 minutes
Last 30 minutes
Last 1 hour
Last 3 hours
Last 6 hours
Last 12 hours
Last 1 day

Wait stats

```
<item>
    <name>WaitStats: Running THREADPOOL</name>
    <type>CALCULATED</type>
    <key>waitstats.running.THREADPOOL</key>
    <delay>5m</delay>
    <history>7d</history>
    <units>ms</units>
    <params>(last(waitstats.THREADPOOL) - prev(waitstats.THREADPOOL))</params>
    <applications>
        <application>
            <name>Wait Statistics</name>
        </application>
    </applications>
</item>
```

Wait stats

```
select cast('<item>' + char(13) + '<name> WaitStats: ' + wait_type + '</name>' + char(13) +
        '<type>DEPENDENT</type>' + char(13) +
        '<key>waitstats.' + wait_type + '</key>' + char(13) +
        '<delay>0</delay>' + char(13) +
        '<history>1d</history>' + char(13) +
        '<units>ms</units>' + char(13) +
        '<applications>' + char(13) + char(9) +
        '<application>' + char(13) + char(9) + char(9) +
        '<name>Wait Statistics</name>' + char(13) + char(9) +
        '</application>' + char(13) +
        '</applications>' + char(13) +
        '<preprocessing>' +char(13) + char(9) +
        '<step>' + char(13) + char(9) + char(9) +
        '<type>JSONPATH</type>' + char(13) + char(9) + char(9) +
        '<parameters>' + char(13) + char(9) + char(9) +char(9) +
        '<parameter>$[?(@.wait_type==''' + wait_type + ''')].wait_time_ms.first()</parameter>' + char(13) + char(9) + char(9) +
        '</parameters>' + char(13) + char(9) + char(9) +
        '</step>' +char(13) + char(9) +
        '<step>' + char(13) + char(9) + char(9) +
        '<type>CHANGE_PER_SECOND</type>'  + char(13) + char(9) + char(9) +
        '<parameters>'  + char(13) + char(9) + char(9) +char(9) +
        '<parameter/>' + char(13) + char(9) + char(9) +
        '</parameters>' + char(13) + char(9) + char(9) +
        '</step>' +char(13) + char(9) +
        '</preprocessing>' +char(13) +
        '<master_item>' + char(13) + char(9) +
        '<key>db.odbc.get[get_wait_stats,&amp;quot;{$MSSQL.DSN}&amp;quot;]</key>' +char(13) +
        '</master_item>' + char(13) +
        '</item>' as xml) as [dependent item],
```

Why Zabbix?

- It might already be there in your organization
- Create your own templates
- Active community
- It’s free

More info?

- https://github.com/reitse/Speaking
- https://wordpress.com/post/sqlreitse.home.blog/574

Thank you so much for listening!

Remember to provide feedback for this session and the event as a whole.

Any questions? You can reach me at:

@2meterDBA / reitse.eskens@axians.com