

Monitoring Microsoft SQL Server with Zabbix Agent2

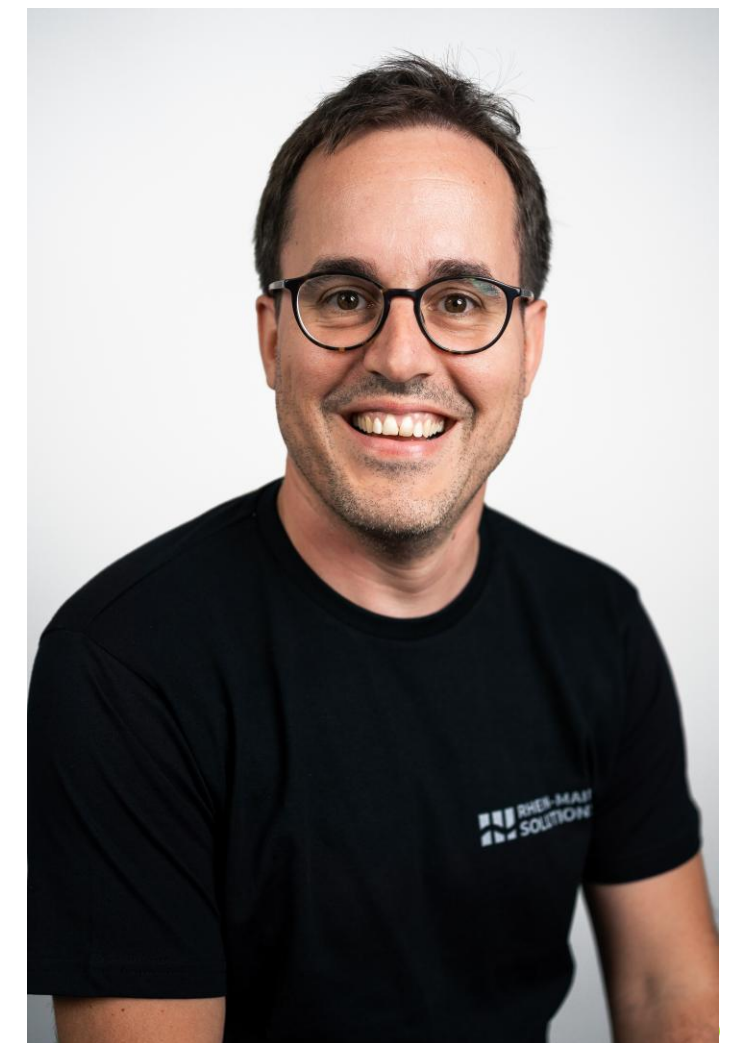
Michael Lindauer

Rhein-Main Solutions GmbH



ZABBIX '26
CONFERENCE

GERMANY



Warum Agent2 statt ODBC?

Weniger Abhängigkeiten

Es ist kein ODBC-Treiber notwendig

Query-Sicherheit

Custom Queries können nur am Agent definiert werden.

Sicherheit

Daten können lokal abgefragt werden. Windows Authentication ist möglich.

Besser Skalierbar

Die Abfragen werden von Agents ausgeführt. Server/Proxies werden entlastet.

Plugin-Setup

Einrichten des Plugins

ZABBIX '26
CONFERENCE

GERMANY

 RHEIN-MAIN
SOLUTIONS

Installation in Linux

Die Agent-Plugins sind schon Teil der Packages.

Hier mit RHEL als Beispiel:

c. Install Zabbix agent 2 plugins

You may want to install Zabbix agent 2 plugins.

```
# dnf install zabbix-agent2-plugin-mongodb zabbix-agent2-plugin-mssql zabbix-agent2-plugin-postgresql
```

Installation in Windows

Download des Installers von der Webseite:

Im Bereich "Verschlüsselung mit OpenSSL" sind keine Plugins.

OS DISTRIBUTION	BETRIEBSSYSTEM-VERSION	HARDWARE	ZABBIX VERSION	VERSCHLÜSSELUNG	PAKET
Windows	11, 10	amd64	7.4	OpenSSL	MSI
Linux	Server 2016 +	i386	7.0 LTS	No encryption	Archive
macOS	Server 2003 +		6.0 LTS		
AIX	XP (64bit) +				
FreeBSD					
OpenBSD					
Solaris					

Zabbix Release: **7.4.9**

Zabbix agent v7.4.9 [Read manual](#)

Packaging: MSI
 Encryption: OpenSSL
 Linkage: Dynamic
 Checksum: sha256: 7dc4517e12baae2cb3982a6b2e2a1d6a49e64ca498b8621d3cffd9304e1bf677
 sha1: 9e6f73fe189ca674f91c7ce6e265abff16d2d6ab
 md5: 1d106415b08284548ed035878d94355

DOWNLOAD https://cdn.zabbix.com/zabbix/binaries/stable/7.4/7.4.9/zabbix_agent-7.4.9-windows-amd64-openssl.msi

Zabbix agent 2 v7.4.9 [Read manual](#)

Packaging: MSI
 Encryption: OpenSSL
 Linkage: Dynamic
 Checksum: sha256: ca9f8ed3854ac10b539ac1506419ea18701d9f0b001c66ad41d3fd9a190ab832
 sha1: d70c-f3a6d3d40f9a0d08f09b3680775b9d3ced8f
 md5: f41cda7bce074be4d68d5c6885a96a19

DOWNLOAD https://cdn.zabbix.com/zabbix/binaries/stable/7.4/7.4.9/zabbix_agent2-7.4.9-windows-amd64-openssl.msi

Installation in Windows

Download des Installers von der Webseite:
Im Bereich "No encryption" sind die Plugins als
MSI-Datei.

OS DISTRIBUTION	BETRIEBSSYSTEM-VERSION	HARDWARE	ZABBIX VERSION	VERSCHLÜSSELUNG	PAKET
Windows	11, 10	amd64	7.4	OpenSSL	MSI
Linux	Server 2016 +	i386	7.0 LTS	No encryption	Archive
macOS	Server 2003 +		6.0 LTS		
AIX	XP (64bit) +				
FreeBSD					
OpenBSD					
Solaris					

Zabbix Release:

Zabbix agent2 plugins v7.4.9 [Read manual](#)

Packaging: MSI
Encryption: No encryption
Linkage: Static
Checksum: sha256: 9d76b3ac3d7b916e09272f1c575c1dc9fab215534454d5b14d79bde33b887b55
 sha1: c226a6c1259a7fd1e53467cd72283117cb2425dc
 md5: a2185cc7d72ae00f4b9f0430c7888e9

[DOWNLOAD](#) https://cdn.zabbix.com/zabbix/binaries/stable/7.4/7.4.9/zabbix_agent2_plugins-7.4.9-windows-amd64.msi

Plugin-Konfiguration

- Der Pfad zur Plugin-Konfiguration ist standardmäßig schon in der Zabbix_agent.conf verlinkt:
Include=.|zabbix_agent2.d|plugins.d|.conf*
- Entsprechend befindet sich die Plugin-Konfiguration in Windows standardmäßig in:
C:|Program Files|Zabbix Agent 2|zabbix_agent2.d|plugins.d|mssql.conf
- In Linux-Systemen ist der Pfad äquivalent:
/etc/zabbix/zabbix_agent2.d/plugins.d/mssql.conf
- Hinweis: In Windows wird der Ordner "*C:|Program Files|Zabbix Agent 2|plugins|*" angelegt. Hier werden aber keine Konfigurationsdateien sondern die ausführbaren Plugin-Dateien abgelegt.

Plugin-Konfiguration

Das MSSQL-Plugin kennt drei Gruppen von Konfigurationsparametern:

Plugin settings

Grundlegende Einstellungen zum Plugin und Custom Queries.

Session settings

Verbindungseinstellungen die gruppiert unter einem Namen zur Verfügung stehen.

Default settings

Standard-Einstellungen wenn keine benannte Session genutzt wird.

Plugin settings

- Plugins.MSSQL.System.Path
Pfad zur ausführbaren Datei (Pflichtfeld, vorbelegt)
- Plugins.MSSQL.Timeout
Timeout für Serveranfragen (Default: 10s)
- Plugins.MSSQL.KeepAlive
Aufbewahrungszeit offener Verbindungen (Default: 600s)
- Plugins.MSSQL.CustomQueriesDir
Pfad für Custom Queries
- Plugins.MSSQL.CustomQueriesEnabled
Schalter zum Aktivieren/Deaktivieren von Custom Queries (Default: false)

Default settings

- Plugins.MSSQL.Default.Uri
Verbindungs-Pfad (Connection-String)
- Plugins.MSSQL.Default.User
User für SQL Authentication
- Plugins.MSSQL.Default.Password
Passwort für SQL Authentication
- Plugins.MSSQL.Default.Database
Standard-Datenbankverbindung

Default settings – Uri

- Format: <protocol://host:port>
- Alternativ: <protocol://host/instance_name> (seit Zabbix 7.2.0; erfordert SQL Server Browser Service)
- Als Protokoll steht ausschließlich "sqlserver" zur Verfügung.
- Wird das Protokoll nicht gesetzt, wird automatisch "sqlserver" ausgewählt.
- Wird der Port nicht gesetzt, wird automatisch "1433" ausgewählt.
- Statt "sqlserver://mssql.domain.com:1433" ist also auch einfach „mssql.domain.com“ möglich.

Default settings

- Plugins.MSSQL.Default.CACertPath
Pfad für CA-Zertifikat
- Plugins.MSSQL.Default.TrustServerCertificate
Schaltet bei "true" die Validierung des Server-Zertifikats aus
- Plugins.MSSQL.Default.HostNameInCertificate
Hostname des DB-Servers im Zertifikat
- Plugins.MSSQL.Default.Encrypt
Stufe der Verschlüsselung zwischen Plugin und DB-Server (true, false, strict, disable)
- Plugins.MSSQL.Default.TLSMinVersion
Minimal verwendbare TLS-Version

Session settings

- Plugins.MSSQL.Sessions.*.Uri
- Plugins.MSSQL.Sessions.*.User
- Plugins.MSSQL.Sessions.*.Password
- Plugins.MSSQL.Sessions.*.CACertPath
- Plugins.MSSQL.Sessions.*.TrustServerCertificate
- Plugins.MSSQL.Sessions.*.HostNameInCertificate
- Plugins.MSSQL.Sessions.*.Encrypt
- Plugins.MSSQL.Sessions.*.TLSMinVersion
- Plugins.MSSQL.Sessions.*.Database

Session settings - Beispiel

- Plugins.MSSQL.Sessions.Prod.Uri
- Plugins.MSSQL.Sessions.Prod.User
- Plugins.MSSQL.Sessions.Prod.Password
- Plugins.MSSQL.Sessions.Prod.CACertPath
- Plugins.MSSQL.Sessions.Prod.TrustServerCertificate
- Plugins.MSSQL.Sessions.Prod.HostNameInCertificate
- Plugins.MSSQL.Sessions.Prod.Encrypt
- Plugins.MSSQL.Sessions.Prod.TLSMinVersion
- Plugins.MSSQL.Sessions.Prod.Database

Metric Keys

- Jeder MSSQL-Plugin-Key hat mindestens drei Parameter für Uri, User und Password.
- Diese Parameter sind keine Pflichtangaben.
- Wenn Session-Konfiguration verwendet werden soll, dann wird statt der Uri der Sessionname gesetzt.

Ohne Session

```
mssql.version["${MSSQL.URI}","${MSSQL.USER}","${MSSQL.PASSWORD}"]
```

Mit Session

```
mssql.version[Prod]
```

Host-Setup

Einrichten des Hosts im Zabbix-Frontend

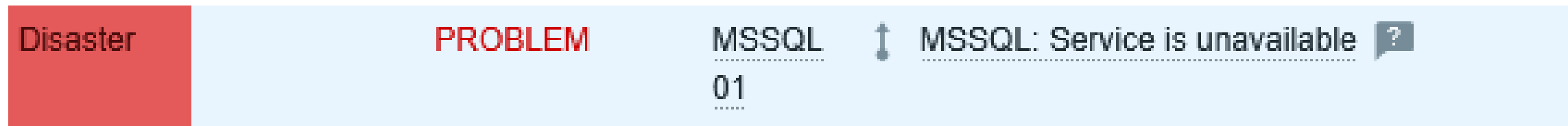
ZABBIX '26
CONFERENCE

GERMANY

 RHEIN-MAIN
SOLUTIONS

Host Setup

1. Template "MSSQL by Zabbix agent 2" verlinken.
2. Macros setzen:
 {\$MSSQL.URI}
 {\$MSSQL.USER}
 {\$MSSQL.PASSWORD}
3. Über Metriken freuen.
4. Sich ärgern:



5. In der Doku nachlesen und die Macros {\$MSSQL.HOST} und {\$MSSQL.PORT} setzen:

* Name

Type

* Key

Custom Queries

Noch mehr Metriken!

ZABBIX '26
CONFERENCE

GERMANY

 RHEIN-MAIN
SOLUTIONS

Custom Queries

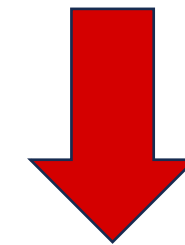
Jede Datei mit der Endung .sql in dem Pfad unter
"Plugins.MSSQL.CustomQueriesDir"
wird vom Zabbix Agent **beim Start** als Custom Query
geladen.

```
beispiel.sql × +
  Bearbeiten  Ansicht
1  USE zabbix;
2  SELECT hostid, name, status
3  FROM dbo.hosts
4  WHERE host LIKE '%' + @p1 + '%'
5  ORDER BY hostid;
```

Custom Queries

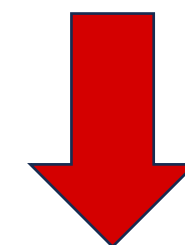
Das Ergebnis ist immer ein JSON-Objekt in einem Array.

mssql.custom.query[Prod,,,beispiel, MSSQL]



```

beispiel.sql × +
  Bearbeiten  Ansicht
1  USE zabbix;
2  SELECT hostid, name, status
3  FROM dbo.hosts
4  WHERE host LIKE '%' + @p1 + '%'
5  ORDER BY hostid;
  
```

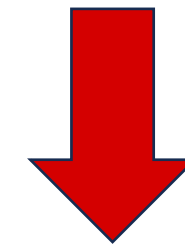


```
[{"hostid":10327,"name":"MSSQL by ODBC","status":3}, {"hostid":10571,"name":"GCP Cloud SQL MSSQL by HTTP","status":3}, {"hostid":10572,"name":"GCP Cloud SQL MSSQL Replica by HTTP","status":3}, {"hostid":10609,"name":"MSSQL by Zabbix agent 2","status":3}]
```

Custom Queries

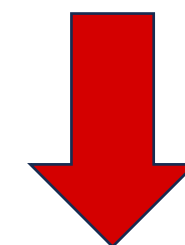
Das Ergebnis ist immer ein JSON-Objekt in einem Array.
Mit dem passenden Item Key lassen sich auch
Parameter übergeben.

mssql.custom.query[Prod,,,beispiel,MSSQL]



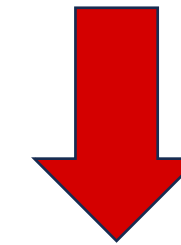
```

beispiel.sql × +
  Bearbeiten  Ansicht
1  USE zabbix;
2  SELECT hostid, name, status
3  FROM dbo.hosts
4  WHERE host LIKE '%' + @p1 + '%'
5  ORDER BY hostid;
  
```



```
[{"hostid":10327,"name":"MSSQL by ODBC","status":3},{"hostid":10571,"name":"GCP Cloud SQL MSSQL by HTTP","status":3},
{"hostid":10572,"name":"GCP Cloud SQL MSSQL Replica by HTTP","status":3},{"hostid":10609,"name":"MSSQL by Zabbix agent 2","status":3}]
```

mssql.custom.query[Prod,,,beispiel,German Conference]

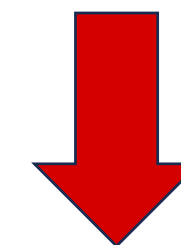


Custom Queries

Das Ergebnis ist immer ein JSON-Objekt in einem Array.
Mit dem passenden Item Key lassen sich auch
Parameter übergeben.

```
C: > Program Files > Zabbix Agent 2 > custom-queries > mssql > beispiel.sql

1  SELECT
2  | 'Hello world!' AS message,
3  | @p1 AS sender
```



```
[{"message":"Hello world!","sender":"German Conference"}]
```

ZABBIX '26

CONFERENCE

GERMANY

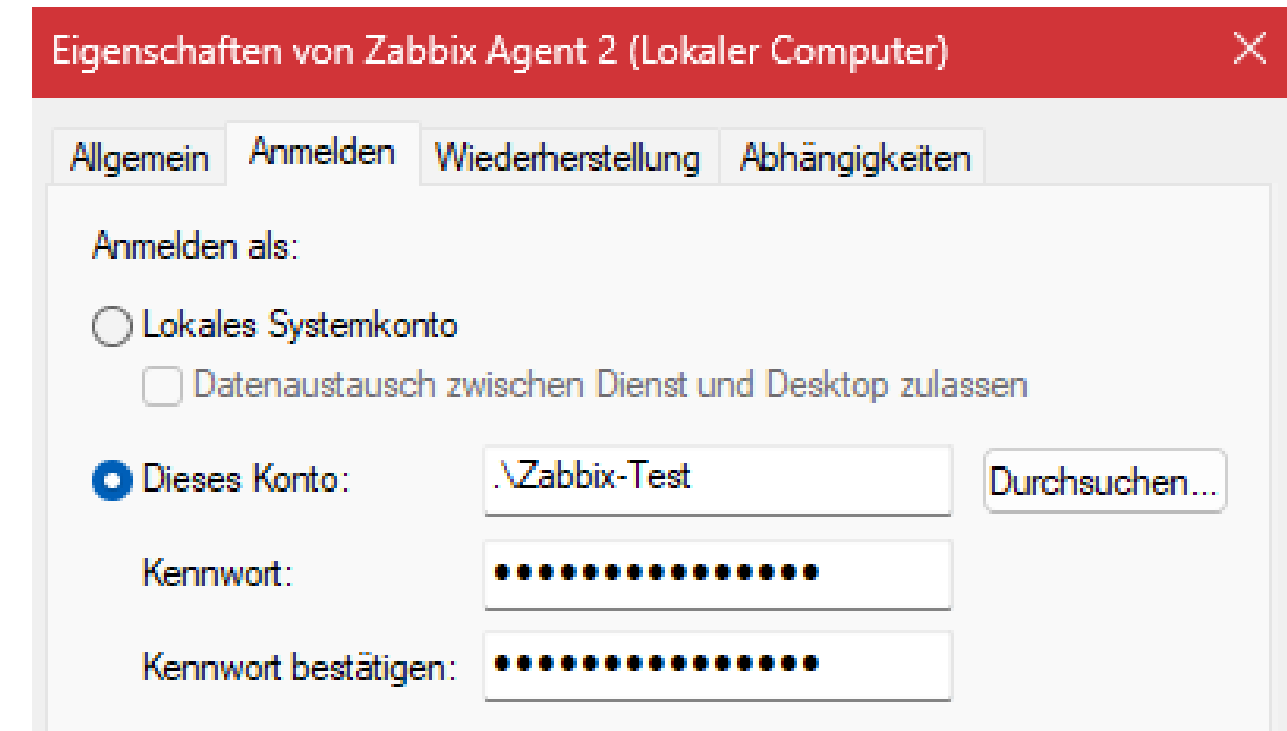
Windows Authentication

Geht das auch local?

 RHEIN-MAIN
SOLUTIONS

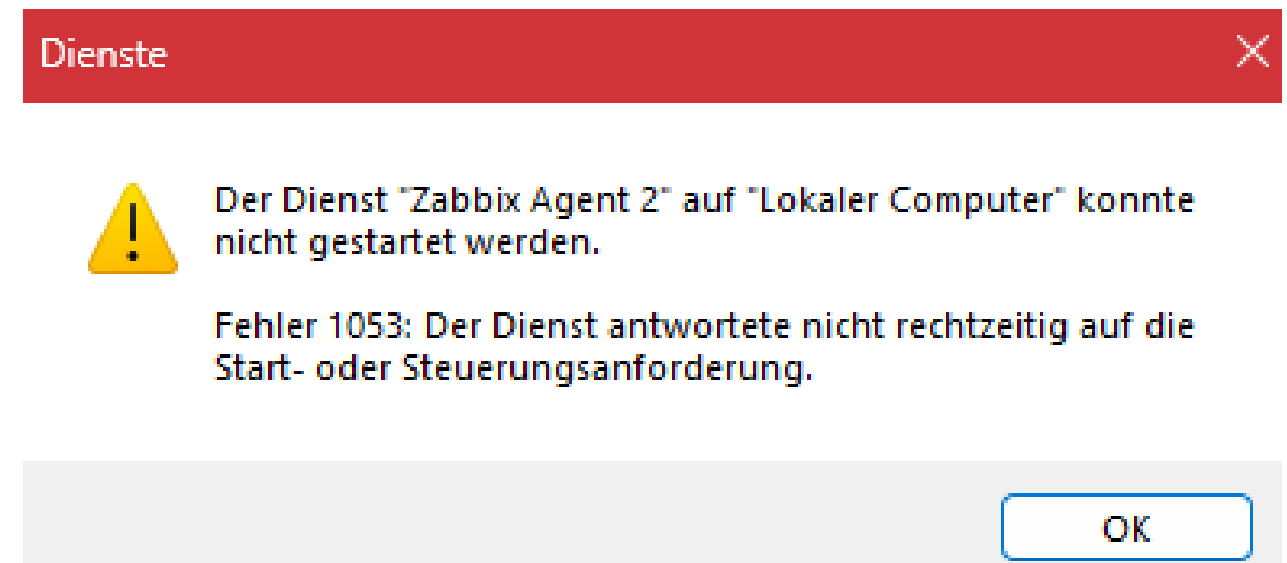
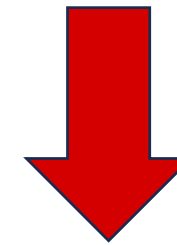
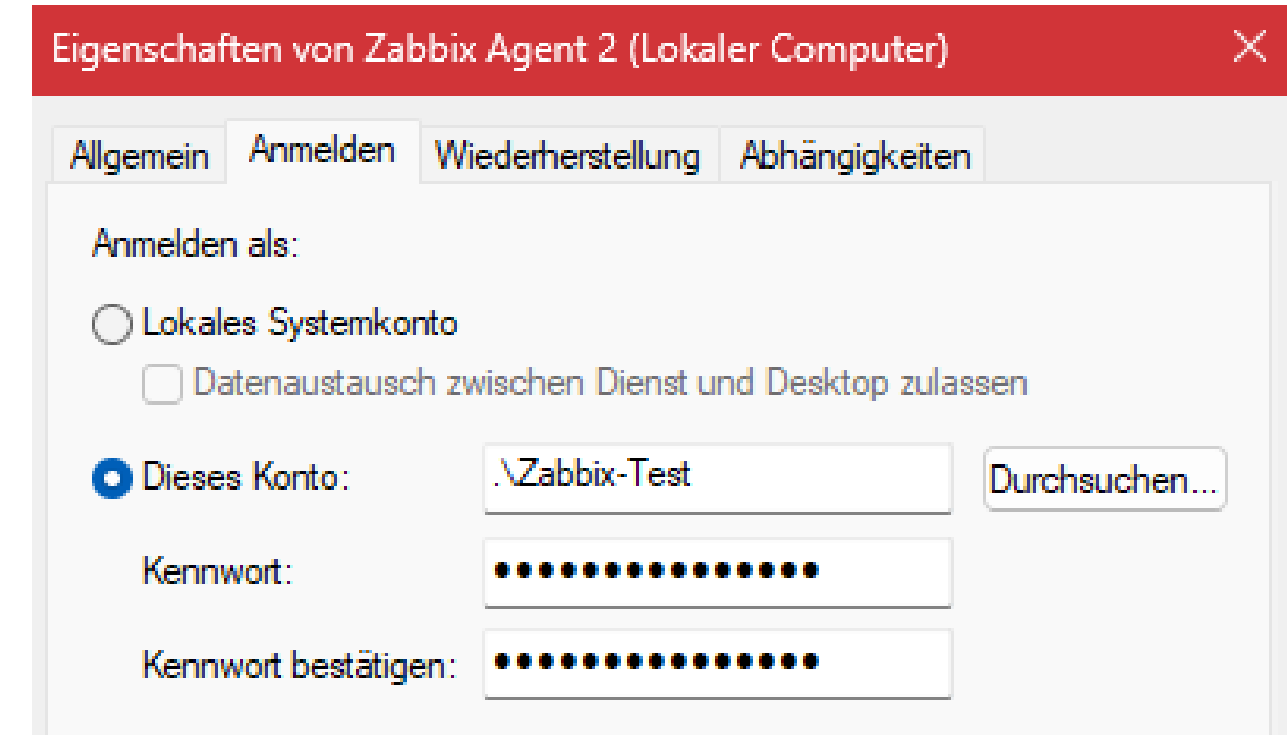
Windows Authentication

Um sich mit einem Windows-Account lokal am MSSQL-Server anzumelden, muss der Dienst Zabbix Agent2 mit diesem Account gestartet werden.



Windows Authentication

Um sich mit einem vorhandenen Account lokal am MSSQL-Server anzumelden, muss der Zabbix Agent2 mit diesem Account gestartet werden.



Windows Authentication – User-Rechte

Um den Zabbix Agent2 im Kontext eines lokalen Users oder eines Domänenusers zu betreiben, benötigt dieser zusätzliche Rechte:

- Schreibrecht auf den Log-Ordner und die schon bestehende(n) Log-Datei(en) des Zabbix Agent2
- Falls Logs ausgelagert sind: Leserechte auf den Konfig-Ordner und sämtliche Dateien.
- Performance Counter:
Computerverwaltung → Lokale Benutzer und Gruppen → Gruppen → Leistungsüberwachungsbenutzer
(Computer Management → Local Users and Groups → Groups → Performance Monitor Users)
- Event Logs:
Computerverwaltung → Lokale Benutzer und Gruppen → Gruppen → Ereignisprotokollleser
(Computer Management → Local Users and Groups → Groups → Event Log Readers)

Ergebnis: Zabbix Agent2 läuft und ist bereit für Windows Authentication.

Windows Authentication

Vorteile

- Ermöglicht das Einhalten von Unternehmensrichtlinien
- Rechte werden über Windows-User und -Gruppen verwaltet
- Auf Seite von Zabbix muss kein Passwort gespeichert werden
- Der Nutzerkontext ermöglicht entsprechende Abfragen (z.B. SMB-Shares)

Nachteile

- Die Einrichtung ist aufwendig und fehleranfällig
- Falls ein Domain-User genutzt wird, bekommt dieser Zugriff auf möglicherweise sensible Daten.
- Durch das Fehlen der Rechte von "Local System" können Einschränkungen beim Monitoring und Ausführen von Scripten auftreten. Das ist über Einstellungen und "Run as..." lösbar.

ZABBIX '26

CONFERENCE

GERMANY

Potentielle Fallstricke

Was ist zu beachten?

Potentielle Fallstricke

Verbindungseinstellungen:

- Der MSSQL-Server muss auf den überwachten Ports erreichbar sein.
- Der MSSQL Server Browser Service muss aktiv sein.
- Der in MSSQL konfigurierte User muss über ausreichende Rechte verfügen.
(entsprechend der Anleitung in https://www.zabbix.com/integrations/mssql#mssql_agent2)
- Für die Anmeldung per Windows Authentication muss der Login folgendermaßen erzeugt werden:
CREATE LOGIN [MYDOMAIN\zabbix] FROM WINDOWS;
- Für die Nutzung von Windows Authentication darf kein User/Passwort in Zabbix eingetragen sein.
- Priorität bei Angaben von Uri, User und Passwort beachten!

Metriken

Was steckt im Standard-Template?

ZABBIX '26
CONFERENCE

GERMANY

 RHEIN-MAIN
SOLUTIONS

Metriken

mssql.db.get

→ Liefert eine Liste aller Datenbanken für Low-Level-Discovery).

mssql.job.status.get

→ Gibt Status- und Laufzeitinformationen zu allen SQL Server Agent Jobs zurück.

mssql.last.backup.get

→ Zeigt pro Datenbank detaillierte Informationen zum letzten Backup.

mssql.local.db.get

→ Liefert detaillierte Infos zu Always-On-Datenbanken auf dem lokalen Server.

mssql.mirroring.get

→ Gibt detaillierte Informationen zum Datenbank-Mirroring zurück

mssql.nonlocal.db.get

→ Liefert Informationen zu Always-On-Datenbanken auf anderen Servern.

Metriken

mssql.perfcounter.get

→ Gibt eine Auswahl an Performance-Zählern mit aktuellen Werten zurück.

mssql.ping

→ Prüft die Erreichbarkeit der Verbindung zum SQL Server.

mssql.quorum.get

→ Liefert Details zur Quorum-Konfiguration der Always-On-Umgebung.

mssql.quorum.member.get

→ Gibt Informationen zu den einzelnen Quorum-Mitgliedern zurück.

mssql.replica.get

→ Liefert detaillierte Informationen zu Always-On-Replikaten.

mssql.version

→ Gibt Versionsinformationen des Microsoft SQL Server zurück.

Fragen

Michael Lindauer
Rhein-Main Solutions GmbH



ZABBIX '26
CONFERENCE

GERMANY

