

# SQL Server überwachen mit CheckMK – Teil 2

In Teil 1 des Themas SQL Server Überwachen mit CheckMK haben wir erklärt, wie wir eine SQL Server Instanz in unsere Überwachung in CheckMK aufnehmen können. Wir haben den CheckMK Agenten und das **mssql.vbs** Plugin installiert und somit den Grundstein für die Überwachung unserer Instanz gelegt. Darauf aufbauend, wollen wir weiter in die Tiefe gehen und uns ansehen, wie wir die vom **mssql.vbs** Plugin mitgelieferten Checks selbst konfigurieren können. Hierfür werden wir den **MSSQL database properties** Check so konfigurieren, dass wir eine Benachrichtigung erhalten, sobald wir eine Datenbank auf unserer überwachten Instanz in den Emergency Mode versetzen.

Kategorie  
SQL Server

Simon  
Schlappfer

## Definieren einer Regel

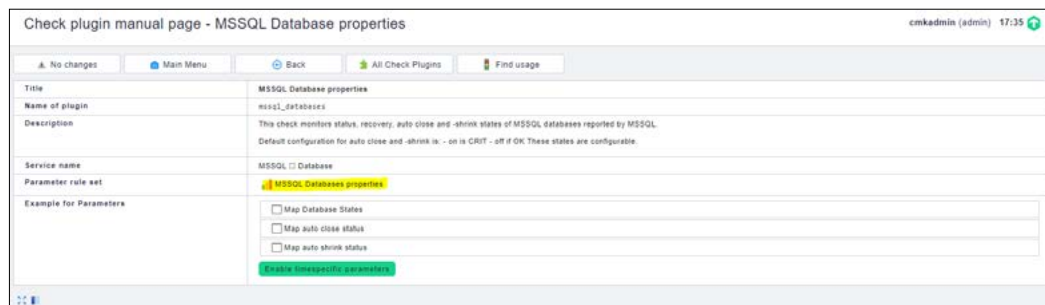
Wir verwenden in diesem Beispiel eine SQL Server 2017 Instanz. Zum Zwecke dieses Beispiels haben wir eine Datenbank mit dem Namen **TestBase** erstellt und werden an dieser die Konfiguration des **MSSQL database properties** Checks testen. Das **mssql.vbs** Plugin ist bereits installiert und die Instanz wurde in unsere Überwachung aufgenommen. Wenn wir nun einen Blick auf die Checks dieser Instanz auf unserer CheckMK Local Site werfen, können wir sehen dass unsere Datenbank überwacht wird:

OK	MSSQL SQL_2017_1 instance	OK - Version: Microsoft SQL Server 2017 (RTM) (14.0.2027.2) - Developer Edition (64-bit)	29 m	32.5 s
OK	MSSQL SQL_2017_1 master Database	OK - Status: ONLINE. Recovery: SIMPLE. Auto close: off. Auto shrink: off	29 m	32.5 s
OK	MSSQL SQL_2017_1 model Database	OK - Status: ONLINE. Recovery: FULL. Auto close: off. Auto shrink: off	29 m	32.5 s
OK	MSSQL SQL_2017_1 msdb Database	OK - Status: ONLINE. Recovery: SIMPLE. Auto close: off. Auto shrink: off	29 m	32.5 s
OK	MSSQL SQL_2017_1 tempdb Database	OK - Status: ONLINE. Recovery: SIMPLE. Auto close: off. Auto shrink: off	29 m	32.5 s
OK	MSSQL SQL_2017_1 TestBase Database	OK - Status: ONLINE. Recovery: FULL. Auto close: off. Auto shrink: off	32.6 s	37.5 s

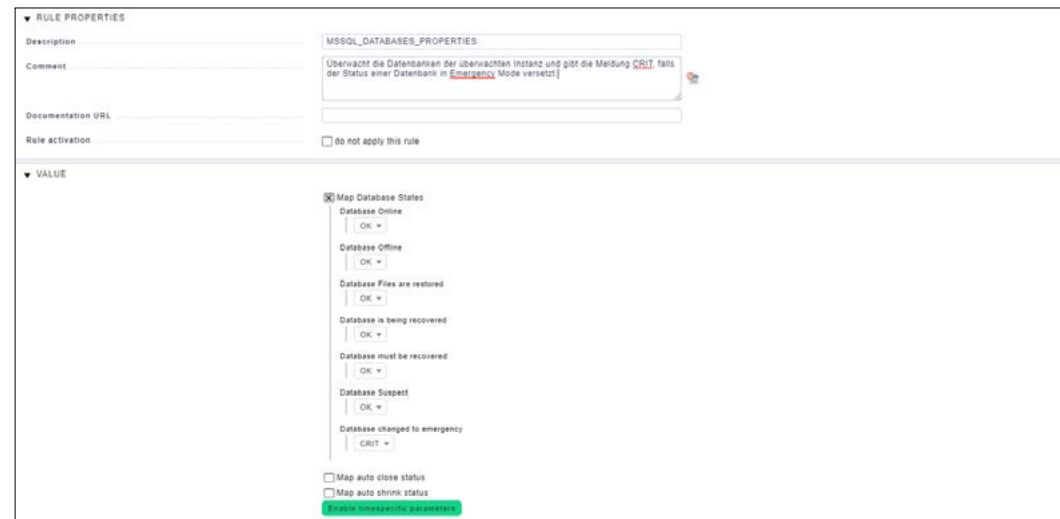
Der Status der Datenbank ist **ONLINE** und die Meldung von CheckMK ist **OK**. Wir wollen den Check nun so konfigurieren, dass wir die Meldung **CRIT** erhalten, sobald eine Datenbank unserer Instanz in den Emergency Mode versetzt wurde.

Dafür müssen wir eine sogenannte **Regel** (oder Rule) erstellen, die vorgibt, wie der CheckMK Agent in unterschiedlichen Situationen reagieren soll

- × Wähle in der WATO - Configuration **Check Plugins** aus
- × Klicke anschließend auf **Applications**
- × Hier finden wir eine große Anzahl Check Plugins. Unter der Rubrik **Microsoft SQL Server** ist nun der Check **MSSQL database properties** zu finden, diesen wählst Du mit einem Linksklick aus
- × In dem erscheinenden Fenster kannst Du die standardmäßig eingestellten Regeln einsehen
- × Zum Konfigurieren der Regeln unter der Sektion **Parameter rule set** klickst Du auf den Namen des Checks:



- × Anschließend wählst Du **Create rule in folder** aus
- × Unter **Rule properties** kannst du einen Namen für die neue Regel sowie eine kurze Beschreibung angeben
- × Unter **Value** und **Map Database States** kannst Du die Regel konfigurieren. Hierfür wählst du unter **Database changed to emergency** die Meldung CRIT aus:



- × Nun klicke auf **Safe** und anschließend auf den **Change** Button im oberen linken Rand
- × Mit einem Klick auf **Activate affected** ist die Definition der Regel abgeschlossen

## Datenbank in den Emergency Mode versetzen

Nachdem wir nun unsere neue Regel konfiguriert haben, wollen wir sie testen. Hierfür verbinden wir uns mit der überwachten Instanz und versetzen unsere Datenbank mit dem folgenden Kommando in den Emergency Mode.

```
ALTER DATABASE TestBase SET EMERGENCY
```

Der CheckMK Agent überprüft die zu überwachende Instanz in Intervallen, weshalb wir nicht sofort eine CRIT Meldung erhalten. Wir können aber eine Überprüfung durch den Agenten erzwingen. Hierfür gehen wir wie folgt vor:

- × Unter Tactical Overview klickst Du auf **Hosts**
- × Anschließend wählst Du den gewünschten Host aus
- × Klicke nun auf dem TestBase Check auf das dreigestrichene Icon und **Reschedule Check**

Und in der Tat bekommen wir nun eine CRIT Meldung für unsere TestBase:

OK	MSSQL SQL_2017_1 Instance	OK - Version: Microsoft SQL Server 2017 (RTM) (14.0.2027.2) - Developer Edition (64-bit)	55 m	40.8 s
OK	MSSQL SQL_2017_1 master Database	OK - Status: ONLINE, Recovery: SIMPLE, Auto close: off, Auto shrink: off	55 m	40.8 s
OK	MSSQL SQL_2017_1 model Database	OK - Status: ONLINE, Recovery: FULL, Auto close: off, Auto shrink: off	55 m	40.8 s
OK	MSSQL SQL_2017_1 model Database	OK - Status: ONLINE, Recovery: SIMPLE, Auto close: off, Auto shrink: off	55 m	40.8 s
OK	MSSQL SQL_2017_1 tempdb Database	OK - Status: ONLINE, Recovery: SIMPLE, Auto close: off, Auto shrink: off	55 m	40.8 s
CRIT	MSSQL SQL_2017_1 TestBase Database	CRIT - Status: EMERGENCY, Recovery: FULL, Auto close: off, Auto shrink: off	7 m	40.8 s

Das Erstellen und Konfigurieren der neuen Regel war also erfolgreich.