

Überwachen von Applikationslogik mittels Enterprise Manager 12c

Bernhard Wesely, Trivadis Delphi GmbH

Der Oracle Enterprise Manager überwacht standardmäßig sämtliche Infrastruktur-Komponenten einer Umgebung auf ihre Funktionsfähigkeit. Dies ist allerdings nur eine Seite der Medaille, denn auch die Applikationslogik selbst sollte überwacht werden, um Fehlern möglichst frühzeitig auf die Schliche zu kommen. Da hier Standard-Metriken nicht greifen, ist eine Möglichkeit gesucht, selbst Metriken zu erstellen.

Nach der Installation des Enterprise Manager (EM) verfügen wir bereits über eine beachtliche Anzahl an mitgelieferten Standard-Metriken. Damit lassen sich Oracle-Komponenten wie Server, Datenbanken und Web-Logic Server, aber auch Systeme von Drittherstellern wie etwa der Microsoft SQL Server überwachen. Es ist jedoch ein Leichtes, sich Fälle auszumalen, in denen die Datenbank innerhalb ihrer normalen Parameter läuft, die Applikation ihre Aufgabe dennoch nicht ordnungsgemäß wahrnimmt. Um auch diese abdecken zu können, unterstützt der EM die sogenannten „Metric Extensions“. Damit ist es möglich, eigene Metriken zu erstellen und damit zu überwachen, zu alarmieren und aufzuzeichnen.

User Defined Metrics

Von früheren EM-Versionen kennen wir bereits das Konzept der User Defined Metrics. Metric Extensions sind eine Erweiterung dieses Konzepts. Die wichtigsten Änderungen im Überblick:

- **Metric Extension Lifecycle**
Wie die meisten Software-Projekte durchläuft eine Metric Extension nun einen Entwicklung/Test/Produktions-Zyklus
- **Erweiterte Protokoll- und Ziel-Unterstützung**
Es sind jetzt mehr Protokolle und Ziel-Typen unterstützt
- **Integrierte Software-Verteilung**
Sollte es nötig sein, für die Erfassung einer Metrik etwa ein Perl-Script auszurollen, so kann dies jetzt automatisch geschehen
- **Zentrale Definition**
User Defined Metrics wurden pro

Ziel definiert. Hat man das Ziel oder den Agenten entfernt, wurden die UDMs ebenfalls gelöscht. Metric Extensions hingegen sind zentral definiert und hängen nicht mehr an einem Ziel.

Entwickeln einer Metric Extension

Wie bereits erwähnt, durchlaufen Metric Extensions nun einen Entwicklungszyklus mit folgenden Stufen (siehe [Abbildung 1](#)):

- **Entwicklung**
In dieser Stufe wird die Metrik in Version 1 erstellt. Anpassungen sind hier jederzeit möglich. Getestet wird die Metrik im Zuge des Erstellungs-Wizard. Ist die Entwicklung abgeschlossen, wird die Metrik als „Deployable Draft“ gespeichert und geht nun in den Test-Zustand über.
- **Test**
Dieser Deployable Draft kann wie jede andere Metrik auf beliebige Zie-

le ausgerollt werden, um das Erfassen der Daten sowie das Senden von Alarmen zu testen. Die Metrik lässt sich an dieser Stelle nicht mehr verändern; dazu ist eine neue Version erforderlich. Um die Metrik nun allgemein verwenden zu können, muss sie veröffentlicht werden und geht damit in den Published-Status über.

Produktion

Befindet sich die Metric Extension nun im Published-Status, kann sie über Monitoring-Templates automatisch auf den Zielsystemen ausgerollt werden. Wie auch im vorhergegangenen Test-Status lassen sich Metric Extension hier manuell auf einzelne Ziele ausrollen.

[Abbildung 2](#) zeigt eine Metric Extension in allen drei Status.

Credential Sets

Um Metric Extensions auf dem jeweiligen Zielsystem ausführen zu

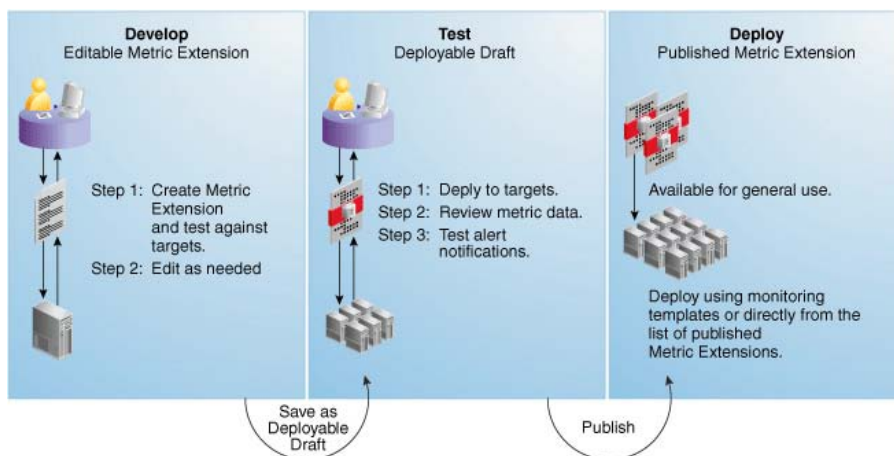


Abbildung 1: Metric Extension Lifecycle (Quelle: Oracle)

| Name | Target Type | Display Name | Version | Description | Status | Deployed Targets |
|-----------------|-------------------|--------------|---------|-------------|------------------|------------------|
| ME\$email_queue | Database Instance | E-Mail Queue | 1 | | Published | 1 |
| ME\$email_queue | Database Instance | E-Mail Queue | 2 | | Deployable Draft | 0 |
| ME\$email_queue | Database Instance | E-Mail Queue | 3 | | Editable | 0 |

Abbildung 2: Metric Extension in unterschiedlichen Status

```
oracle@oms01:~/ [oms12c] emcli create_credential_set
  -set_name="Metric Extension Monitoring Credentials"
  -target_type=oracle_database
  -supported_cred_types=DBCreds -monitoring

Credential set "Metric Extension Monitoring Credentials"
created successfully.
```

Listing 1: „emcli“-Kommando, um ein Monitoring Credential Set anzulegen

können, werden natürlich Zugangsdaten benötigt. Die Standard-Enterprise-Manager-Metriken verwenden hierzu die sogenannten „Default Monitoring Credentials“. Für Datenbanken werden diese während der Veröffentlichung von Zielen im Credential Set „Monitoring Database Credentials“ gespeichert. Typischerweise ist dies der Datenbank-Benutzer. Dieser hat jedoch nur eingeschränkte Rechte in der Datenbank und kann in den meisten Fällen nicht auf die Applikations-Tabellen zugreifen. Um nun einen eigenen Benutzer spezifizieren zu können, muss ein neues Credential Set angelegt werden. Dies geschieht über das Enterprise Manager Command Line Interface „emcli“ (siehe Listing 1).

Um dem nun angelegten Credential Set eine Username/Passwort/Ziel-Kombination zuzuweisen, findet sich unter „Setup -> Security -> Monitoring Credentials“ eine grafische Oberfläche. Hier wählt man die passende Kombination von Zielsystem und Credential Set und vergibt über den Wizard hinter dem Button „Set Credential“ einen Benutzernamen und ein Passwort (siehe Abbildung 3).

Während der Erstellung der Metric Extension kann nun das zu verwendende Credential Set ausgewählt werden (siehe Abbildung 4). Achtung: Username und Passwort werden erst während der Ausführung gegen ein

Zielsystem ausgewertet; es ist daher zwingend notwendig, Username und Passwort vor der Verteilung der Metrik für jedes System zu definieren.

Adapter

Adapter stellen die Kommunikationsmethode zwischen Metrik Extension und Zielsystem dar. Es sind folgende Adapter verfügbar:

- OS Command – Single Column
- OS Command – Multiple Values
- OS Command – Multiple Columns
- SQL
- SNMP
- Java Management Extensions (JMX)

Unterschiedliche Ziel-Typen unterstützen immer nur ihr passendes Subset an Adaptern. Es ergibt beispielsweise wenig Sinn, eine Datenbank mittels „JMX“ oder einen Host mittels „SQL“ kontaktieren zu wollen. Die Definition der möglichen Adapter pro Zieltyp ist bereits vorgegeben.

Ein Fallbeispiel

Angenommen, ein Web-Shop verschickt nach einer Bestellung automatisch Bestätigungs-E-Mails über die bestellte Ware. Gesteuert wird dieser Versand über eine Tabelle namens „WEBSHOP.EMAIL“. Diese besteht aus dem Datum, an dem der Datensatz erzeugt wurde, dem Empfänger und dem Text der E-Mail.

Nach einer Bestellung wird ein Datensatz in dieser Tabelle erzeugt. Asynchron dazu läuft ein Job, der die Tabelle ausliest und E-Mails generiert. Für jede gesendete E-Mail wird die entsprechende Zeile anschließend gelöscht. Dieses einfach gehaltene Beispiel lässt sich natürlich auf jede beliebige Komplexität erweitern.

Bevor wir mit der Implementierung angefangen, muss definiert werden, was überwacht werden soll. Zwei Fragen bieten sich hier vordergründig an:

- *Ist der Versand der E-Mails schnell genug oder stauen sich Mails auf?*
Die zu erfassende Metrik ist dann die Gesamtzahl der E-Mails in der Queue.
- *Wächst die Tabelle zu schnell an, obwohl das Limit von Punkt 1 noch nicht erreicht ist?*
Die zu erfassende Metrik ist dann die Anzahl der in den letzten fünf Minuten hinzugekommenen Einträge.

Erstellen der Metric Extension

Über die Menüpunkte „Enterprise -> Monitoring -> Metrik Extensions“ gelangt man zu der zentralen Administrationsseite für Metric Extensions. Hier können neue Metriken erstellt, editiert, ausgerollt und gelöscht werden. Über den Button „Create“ lässt sich hier eine neue Metrik erstellen. Auf der ersten Seite geben wir allgemeine Informationen wie den Namen der Metric Extension oder den Ziel-Typ an. In diesem Fall ist das Ziel vom Typ „Database Instance“.

Eine Metric Extension hat immer einen kurzen (intern verwendet) und einen langen Namen (Display-Name). Der Display-Name darf beispielsweise Leerzeichen enthalten und ist auch der Name, der eigentlich immer angezeigt wird. Passend zu unserem Ziel wählen wir noch den Adapter namens

„SQL“ aus. Im unteren Teil der Seite kann noch eingestellt werden, ob die Metrik aktiv ist beziehungsweise wie oft sie vom Zielsystem gesammelt werden soll. Diese Frequenz stellen wir auf fünf Minuten. Wollen wir die erhobenen Daten im Repository historisiert speichern, wählen wir unter „Use of Metric Data“ noch den Punkt „Alerting and Historical Trending“ an. Durch Klicken des „Next“-Buttons gelangen wir auf die nächste Seite.

Nun wird der gewählte Adapter konfiguriert. Im Falle des SQL-Adapters geht es hier um das auszuführende Statement. In unserem einfachen Beispiel reicht die simple SQL-Query „select count(*) from webshop.email“ aus, um die Metrik zu erfassen. Weitere Konfigurationen wie die Steuerung von PL/SQL oder die Verwendung von Bind-Variablen können weiter unten auf der Seite vorgenommen werden. Wir klicken den „Next“-Button.

Hier definieren wir die Spalten, die unser Statement zurückliefern wird. Aktuell ist dies nur eine, das „count(*)“. Über die Menüpunkte „Add -> new metric column“ erreichen wir ein Fenster, das uns einige Konfigurationmöglichkeiten bietet. Wieder werden ein kurzer sowie ein langer Name verlangt. Zusätzlich können wir hier gleich Schwellwerte und angepasste Alarmmeldungen definieren. Für dieses Beispiel verwenden wir „anzahl_emails“ als Name, setzen den Comparison Operator auf „>“ (größer) und die Schwellwerte auf 5.000 („Warning“) und 10.000 („Critical“). Da sich dieser Alarm von selbst auflöst, sobald die Schwellwerte wieder unterschritten werden, lassen wir das Feld „Manually Clearable Alert“ auf „false“. Metriken, die keinen definierten „OK“-Zustand haben, wie Fehlermeldungen in Logfiles, würde man hier auf „true“ stellen. Mittels „OK“-Button kommen wir wieder zur vorhergehenden Ansicht zurück.

Der aufmerksame Beobachter fragt sich nun, wo die zweite Metrik (Rate der hinzugekommenen E-Mails pro fünf Minuten) abgeblieben ist. Schließlich erfassen wir diese Information ja nicht gesondert mit unserem „count(*)“. Hier hilft uns der Enter-

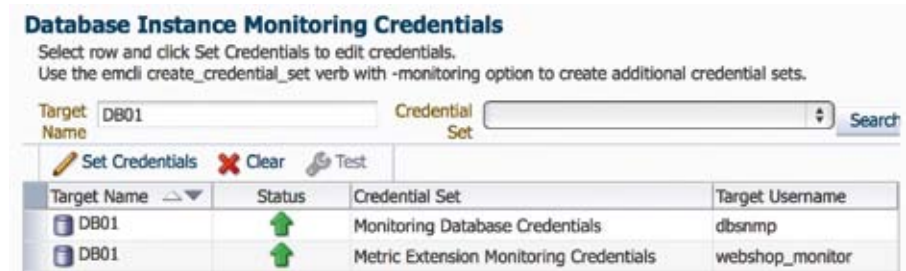


Abbildung 3: GUI zur Verwaltung der Monitoring Credentials

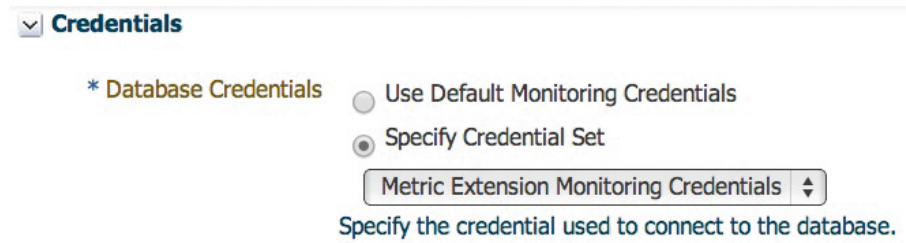


Abbildung 4: Auswahl des Credential Set während der Metric-Extension-Erstellung

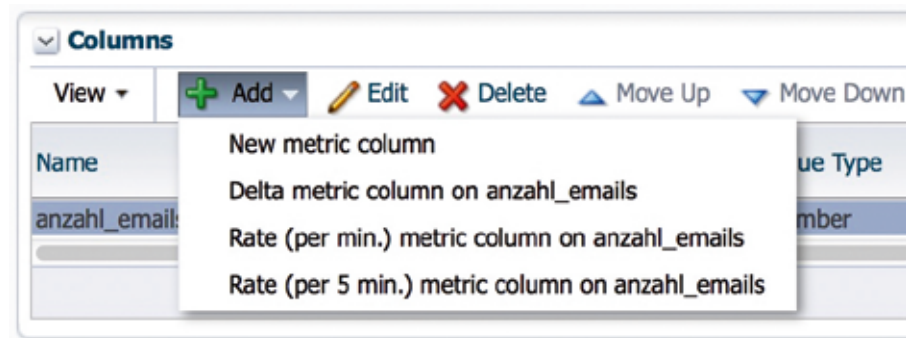


Abbildung 5: Hinzufügen der Rate-Spalte

| Columns | | | | | | |
|-----------------------------|-----------------|-------------|------------|---------------------|---------|----------|
| Name | Display Name | Column Type | Value Type | Alert Threshold | | |
| | | | | Comparison Operator | Warning | Critical |
| anzahl_emails | Anzahl E-Mails | Data Column | Number | > | 5000 | 10000 |
| Rate_Per_5Min_anzahl_emails | Rate ueber 5min | Data Column | Number | > | 100 | 200 |

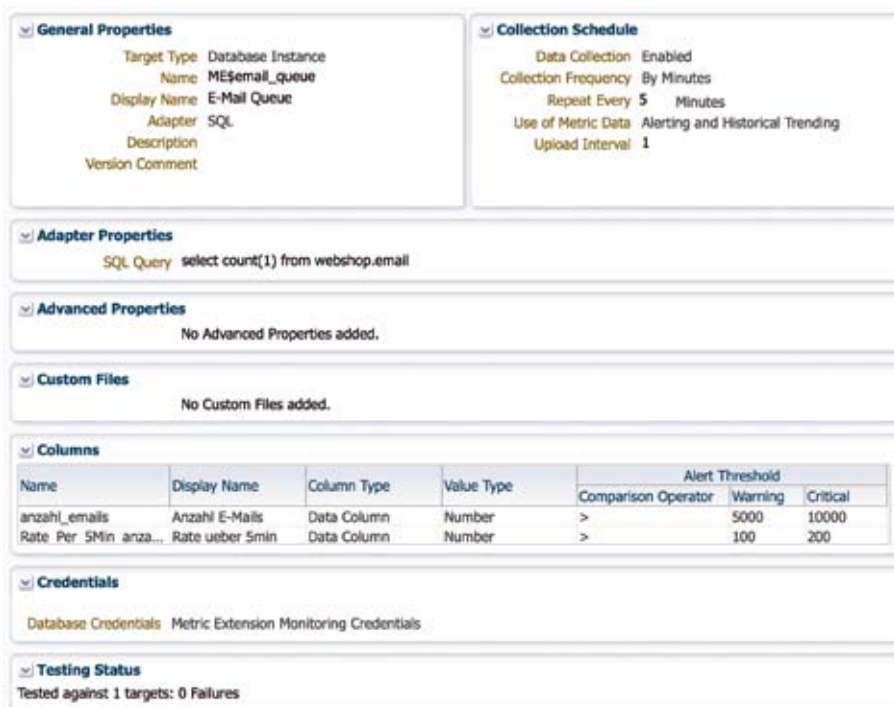
Abbildung 6: Spalten-Definition der Metric Extension

prise Manager, der diese Werte für uns errechnet. Wir definieren eine zweite Spalte auf Basis der ersten. Diese erzeugen wir über den Menüpunkt „Add -> Rate (per 5 min.) metric column on anzahl_emails“ (siehe Abbildung 5).

Der Name der Spalte ist hier schon vorgegeben, wir definieren nur den Display-Namen und den Schwellwert. Als Beispiel nehmen wir hier 100 („Warning“) und 200 („Critical“). Wir klicken auf „OK“. Abbildung 6 zeigt die fertige Spalten-Definition.

Auf der nächsten Seite geben wir die Credentials an, die der SQL-Adapter zum Einloggen in die Datenbank verwendet. Sollte der Benutzer alle erforderlichen Rechte besitzen, ist hier nichts zu verändern. Falls diese nicht ausreichen, muss ein Credential Set, wie im Kapitel „Credential Sets“ beschrieben, angelegt, konfiguriert und ausgewählt werden.

Im vorletzten Schritt kann die Metrik gegen ein oder mehrere Ziele getestet werden. Durch Klicken auf „Add“



| Name | Display Name | Column Type | Value Type | Comparison Operator | Warning | Critical |
|-----------------------|-----------------|-------------|------------|---------------------|---------|----------|
| anzahl_emails | Anzahl E-Mails | Data Column | Number | > | 5000 | 10000 |
| Rate Per 5Min anza... | Rate ueber 5min | Data Column | Number | > | 100 | 200 |

Abbildung 7: Zusammenfassung der Metric Extension

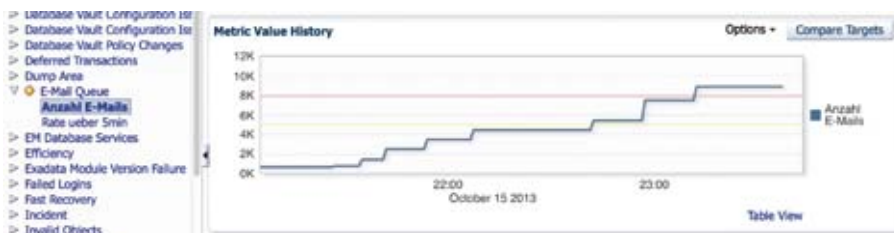


Abbildung 8: Graph der historisierten Metrikdaten

können Datenbanken hinzugefügt werden. Wir wählen hier unsere Webshop-Datenbank aus. Mittels „Run Test“ wird die Metrik auf dem Zielsystem ausgeführt. In der unteren Hälfte der Seite sehen wir nach kurzer Zeit das Resultat der Auswertung. Hier wird nur die selektierte Spalte dargestellt, die Rate wird ja über einen vergangenen Zeitraum berechnet und kann hier nicht angezeigt werden. Sollte alles passen, kommen wir über einen Klick auf „Next“ zur Zusammenfassung (siehe Abbildung 7); über einen weiteren Klick auf „Finish“ wird die Metric Extension angelegt.

Ausrollen der Metric Extension

Nach Erstellung der Metric Extension wollen wir sie natürlich am konkreten Objekt testen. Hierzu erstellen wir einen Deployable Draft. Wir wählen die Metrik aus und verwenden den Punkt

„Actions -> Save as Deployable Draft“. Nun lässt sich die Metrik via „Actions -> Deploy to Targets“ ausrollen. Im nun folgenden Assistenten fügen wir durch „Add“ die Datenbank des Webshops hinzu. Nun wird die Metrik ausgerollt und entsprechend dem Collection Schedule gesammelt.

Events, die von Metric Extensions ausgelöst werden, unterscheiden sich für das Incident System nicht von Events der Oracle-eigenen Metriken. Genau wie die Standard-Metriken können Events über ein Ruleset, das auf „Targets“ zutrifft, in Kombination mit Rules des Typs „Metric Alert“ sowie „Metric Evaluation Error“ in das Incident System gebracht werden. „Metric Evaluation Error“ wird gern vergessen, ist aber gerade im Umfeld der Metric Extensions wichtig, um herauszufinden, ob der letzte Applikations-Rollout die Tabellenstruktur so verändert hat,

dass die Metrik nicht mehr ausgewertet werden kann.

Anzeigen der Metrikwerte

Sofern wir bei der Erstellung der Metric Extension angegeben haben, dass wir die Daten auch für „Historical Trending“ speichern wollen, werden die Metrikdaten im Repository historisiert gespeichert. Diese Werte lassen sich in jeder Ansicht grafisch darstellen. Ob dies nun auf der Datenbank-Homepage über „Database -> Monitoring -> All Metrics“ erfolgt (siehe Abbildung 8), über Reports oder im „Charts“-Feature von Gruppen, ist ganz dem EM-Administrator überlassen. Über Reports lassen sich die Werte, Graphen und Alarme auch an Benutzer, die keinen Zugriff auf den Enterprise Manager haben, verteilen. Dies kann per Web-Zugriff auf eine spezielle Enterprise-Manager-Seite oder per E-Mail passieren.

Fazit

Es hat sich viel seit den User Defined Metrics der vergangenen EM-Versionen getan. Auch wenn die Kernfunktionalität kaum verändert wurde, gibt es rund herum nun viel mehr Möglichkeiten, eigene Anforderungen abzubilden. Mithilfe der EM-Standard-Metriken und auch der Metric Extensions ist es nun möglich, die eigene Infrastruktur lückenlos zu überwachen. Aber Achtung: Metric Extensions sind Teil des jeweiligen zielspezifischen Performance-Diagnose-Pack, wie zum Beispiel des Database-Diagnostics-Pack für Datenbanken, und damit lizenzpflichtig.

Bernhard Wesely
Bernhard.Wesely@trivadis.com

