JBug OWL

OMD/Nagios – Monitoring & Plug-In‘s

Dennis Wiegand
dwiegand@s-und-n.de

Agenda

- Was ist OMD und was hat es mit Nagios zu tun
  - Inhalte der OMD Distribution
- Nutzung von OMD / Vorteile
- Plug-In Beispiele (check_logfiles / jolokia)

Was ist OMD und was hat es mit Nagios zu tun

- OMD = The Open Monitoring Distribution
- Komplettes Paket aus diversen Einzelbausteinen(bereits kompiliert)
- Ermöglicht auf einem Host verschiedene Sites aufzubauen (z.B. Prod, Uat, Dev)
- Was hat das mit Nagios zu tun? Es beinhaltet bereits den Nagios Kern und ist ebenfalls Open Source

Inhalte der OMD Distribution

- Nagios
  - Nsca
  - check_nrpe
  - Check_disk, ssh, dns, check_http usw.
- Icinga
- Shinken
- NagVis
- pnp4nagios
- rrdtool/rrdcached
- Check_MK
- MK Livestatus
- Multisite
- Dokuwiki
- Thruk
- Mod-Gearman
- check_logfiles
- check_oracle_health
- check_mysql_health
- jmx4perl
- check_webinject
- check_multi

Nutzung von OMD / Vorteile

- Zusammenstellung von verschiedenen Modulen und Plug-In‘s die in einem Paket bereits aufeinander abgestimmt sind
- Open Source
- Hohe Anzahl von Services und Hosts möglich dank mod-gearman
- Site basiert. Alle separat konfigurierbar und via Frontend zugreifbar(eigene User,URL)
- Einfaches Anlegen der site‘s mit Kommando: omd create <SITENAME>
- Anlegen einer Site Kopie ebenso schnell mit: omd cp <SITENAME_ALT> <SITENAME_NEU>
- Update der Version schnell und einfach. OMD Update Paket wird installiert und via Befehl beliebiger up oder downgrade für eine site oder mehrere site‘s
- Enorme Zeitersparnis um eine Nagios Instanz aufzusetzen

Plug-In Beispiele – check_logfiles

- Logfiles auf Basis von Patterns(regexp) prüfen und Alarme versenden (WARNING,CRITICAL)
- Merkt sich letztes Offset und prüft dort weiter um nicht ein Log komplett neu zu scannen
- Kann Logfilerotationen berücksichtigen
- Folgeaktionen möglich ehe das Event in Nagios reported wird, z.B. starte den Server durch
- Mehrere „Profile“ in einem Prüfaufruf möglich

```
nagios$ check_logfiles --tag=ssh --logfile=/var/adm/messages \
     --rotation SOLARIS \
     --criticalpattern 'Failed password for root'
CRITICAL - (1 errors in check_logfiles.protocol-2007-04-25-20-59-20) - Apr 25 20:59:15 srvweb8 sshd[10849]: [ID 8
00047 auth.info] Failed password for root from 172.16.224.11 port 24206 ssh2 |ssh=2831;0;1;0
```

Plug-In Beispiele - jolokia

- Ausflug zu JMX(Java Management Extension)
  - Ab Java 1.5 im „Bauch“ (keine extra lib nötig)
  - Bereits mit JAVA Service-Req-3(Jahr 2000) vorhanden
  - Konzept ist die Applikationen von innen heraus zu Monitoren oder zu Managen
  - JSR-160 bietet API um mit JMX-enabled-applications zu kommunizieren

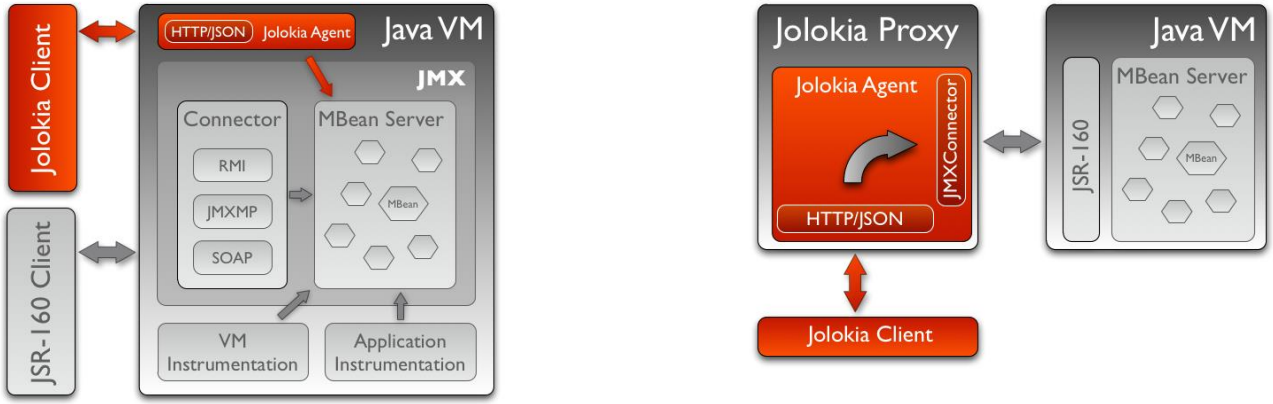
- Jolokia-Agent kann innerhalb einer JVM auf den JMX Server zugreifen und ist via HTTP/HTTPS von außen erreichbar

Plug-In Beispiele - jolokia

- Agent Jolokia Vorteile:
  - Er kann Bulk Requests senden (1 request mit n Anfragen)
  - Security Features erlauben es die Kommunikation in https laufen zu lassen, Hosts, IP‘s oder Subnetzte zu beschränken, Operationen zu blockieren(z.B. exec) oder User + PW festzulegen
  - Benötigt keine eigene Java VM und ist somit schnell(kein JVM start nötig)
  - Nutzt die json-simple library für schnelle Verarbeitung von Request/Response
  - Nutzung durch Firewall einfacher da HTTP/HTTPS mit einem Port genutzt wird. JSR-160 nutzt Port Ranges

Plug-In Beispiele - jolokia

- Kann über Agent auf Ziel-Server oder via Proxy Server genutzt werden. Letzteres ermöglicht Nutzung von Jolokia bei System die keine zusätzlichen Agent erlauben



Plug-In Beispiele - jolokia

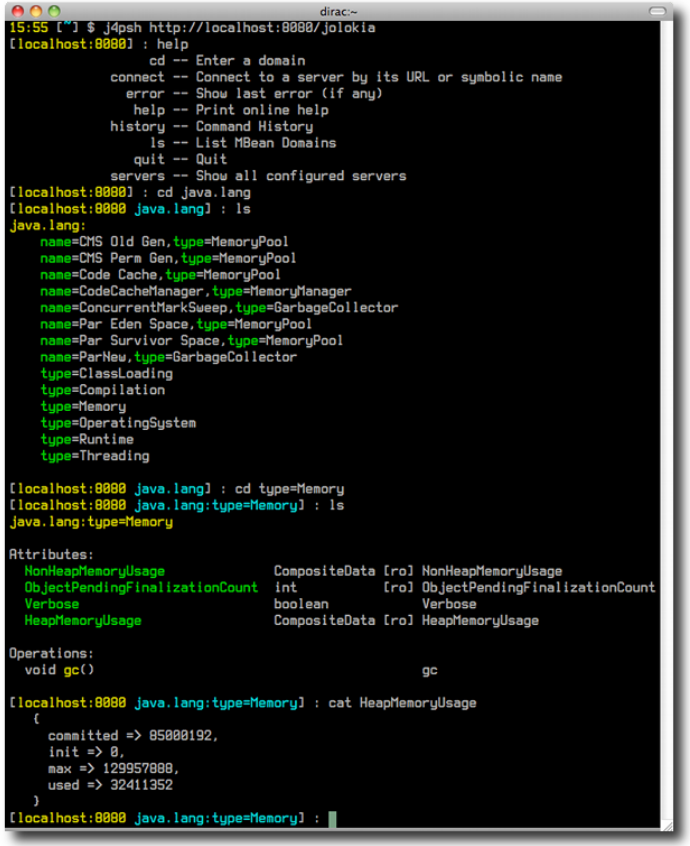
- Installiert wird z.B. das war File in einem tomact. Wenn das war File deployed wurde kann direkt via HTTP Aufruf die Abfrage der JMX Daten ausgeführt werden

```
http://localhost:8080/jolokia/read/java.lang:type=Memory/HeapMemoryUsage
```


```
{
  "timestamp":1285531108,
  "status":200,
  "request": {
              "mbean":"java.lang:type=Memory",
              "attribute":"HeapMemoryUsage",
              "type":"read"
            },
   "value": {
              "max":"129957888",
              "committed":"85000192",
              "init":"0",
              "used":"7713440"
            }
}
```

Plug-In Beispiele - jolokia

- Ist der Agent deployed kann mit dem OMD Paket das Plug-In jmx4perl genutzt werden um Nagios Checks zu definieren oder via j4psh auf die JMX Mbean‘s zugegriffen werden(analog zu http://localhost:8080/jolokia/list)



Plug-In Beispiele - jolokia

- Es lassen sich Check‘s erstellen die via Bulk Request z.B. den Heap/NonHeap abfragen können und in einer Ausgabe darstellen
- 2 Server mit verschiedener Speicherausstattung können mit dem gleichen Service abgefragt werden, da die Speicherbelegung used/max zu einer % Zahl berechnet werden kann
- Somit ist nur der gewünscht Schwellwert wichtig und nicht der vorhandene Speicher des Servers

Plug-In Beispiele - jolokia

- Beispiel für eine Nagios jmx4perl Konfiguration um Heap/NonHeap zu prüfen (Bulk-Request)

```
<Check memory_heap>
   Use = relative_base($0,$1)
   Unit = B
   Value = java.lang:type=Memory/HeapMemoryUsage/used
   Base = java.lang:type=Memory/HeapMemoryUsage/max
   Label = Heap-Memory: $BASE
   Name = Heap
</Check>

<Check memory_non_heap>
   Use = relative_base($0,$1)
   Unit = B
   Value = java.lang:type=Memory/NonHeapMemoryUsage/used
   Base = java.lang:type=Memory/NonHeapMemoryUsage/max
   Label = Non-Heap-Memory: $BASE
   Name = Non-Heap
</Check>

<MultiCheck memory>
   Check memory_heap
   Check memory_non_heap
</MultiCheck>
```

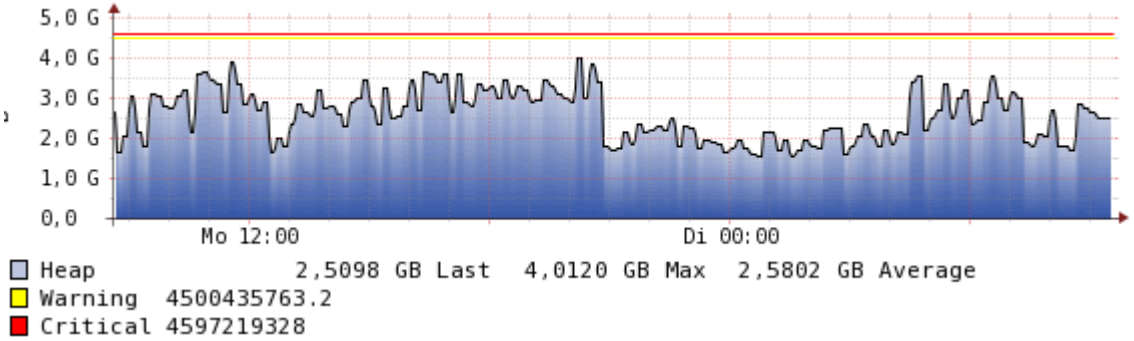
Use = Nutze relativen Check(Max/Current)
Unit = Formatierung B = Byte
$0 / $1 = Übergebne Schwellwerte
Value = MBean um den aktuell belegten Speicher zu zeigen
Base = MBean um den maximal verfügbaren Speicher zu zeigen
Label = Ausgabe formartieren
Name = Performance Daten Label für Nagios

MultiCheck fasst memory_heap & memory_non_heap zusammen

Plug-In Beispiele - jolokia

Current Status:	OK (for 10d 16h 57m 4s)
Status Information:	OK - All 2 checks OK
	[1] OK memory_heap: Heap-Memory: 51.89% used (2.34 GB / 4.51 GB)
	[2] OK memory_non_heap: Non-Heap-Memory: 71.98% used (218.81 MB / 304 MB)
Performance Data:	Heap=2511189344B;4500435763.2;4597219328;0;4839178240 Non-Heap=229436080B; 296453406.72;302828748.8;0;318767104

- Mit pnp4Nagios lassen sich die Ergebnisse visualisieren



Vielen Dank !

Dennis Wiegand
dwiegand@s-und-n.de