

5 Brama Java

Przegląd

Istnieje natywne wsparcie dla monitorowania aplikacji JMX wykorzystujące demona Zabbix zwanego "Brama Java Zabbix", dostępne jest od wersji Zabbix 2.0. Brama Java Zabbix jest demonem napisanym w Java. Żeby uzyskać wartość jakiegoś licznika JMX na hoście, serwer Zabbix pyta o to bramę Java Zabbix, która używa [API zarządzania JMXI](#) do zdalnego odpytania aplikacji. Aplikacja nie wymaga żadnego dodatkowego oprogramowania, jedynie musi być uruchomiona z opcją linii komend `-Dcom.sun.management.jmxremote`.

Brama Java akceptuje połączenia przychodzące z serwera lub proxy Zabbix i może być użyta jedynie jako "pasywne proxy". W przeciwieństwie do proxy Zabbix, może również być używana przez proxy Zabbix (proxy Zabbix nie mogą być łączone w łańcuch). Dostęp do każdej bramy Java konfigurowany jest bezpośrednio w pliku konfiguracyjnym serwera lub proxy Zabbix, z tym, że można skonfigurować jedynie jedną bramę Java dla serwera lub proxy Zabbix. Jeżeli host posiada zarówno pozycje typu **Agent JMX** jak i innych typów, tylko pozycje **Agent JMX** zostaną przekazane do bramy Java.

Gdy następuje aktualizacja pozycji przez bramę Java, serwer lub proxy Zabbix łączy się z bramą Java i żąda wartości, które brama Java pobiera i odsyła z powrotem do serwera lub proxy. Zatem, brama Java nie cache'uje żadnych wartości.

Serwer lub proxy Zabbix posiada specyficzny typ procesu, który łączy się do bramy Java, jest on kontrolowany przez opcję **StartJavaPollers**. Wewnętrznie, brama Java uruchamia wiele wątków, zgodnie z opcją **START_POLLERS**. Po stronie serwera, jeżeli połączenie zajmie więcej czasu niż ilość sekund z opcji **Timeout**, zostanie ono zakończone, ale brama Java może nadal odbierać wartości z licznika JMX. Żeby to rozwiązać, od Zabbix 2.0.15 i Zabbix 2.2.10 w bramie Java istnieje opcja **TIMEOUT**, która pozwala ustalić opóźnienie dla operacji sieciowych JMX.

Serwer lub proxy Zabbix spróbują zgrupować zapytania do jednego docelowego JMX (zgodnie z interwałami pozycji) i wysłać je do Bramy Java w jednym połączeniu (dla zwiększenia wydajności).

Sugerujemy, by wartość **StartJavaPollers** była mniejsza lub równa wartości **START_POLLERS**, w przeciwnym razie może nastąpić sytuacja, gdy nie będzie dostępny żaden wolny wątek w bramie Java do obsłużenia przychodzących zapytań.

Rozdziały poniżej opisują jak pobrać i uruchomić bramę Java Zabbix, jak skonfigurować serwer Zabbix (lub proxy Zabbix), by używał bramy Java Zabbix do monitorowania JMX, i jak skonfigurować pozycje Zabbix w interfejsie użytkownika, by odpowiadały konkretnym licznikom JMX.

5.1 Pobieranie bramy Java

Istnieją dwa sposoby pobrania bramy Java. Pierwszym jest pobranie pakietu bramy Java ze strony Zabbix, drugim jest kompilacja bramy Java ze źródeł.

5.1.1 Pobieranie ze strony Zabbix

Pakiety bramy Java Zabbix (RHEL, Debian, Ubuntu) dostępne są do pobrania z

<http://www.zabbix.com/download.php>.

5.1.2 Kompilacja ze źródeł

Żeby skompilować bramę Java, po pierwsze trzeba uruchomić skrypt `./configure` z opcją `--enable-java`. Zalecamy ustawienie opcji `--prefix` (ustawienie ścieżki instalacji) na ścieżkę inną niż domyślna (`/usr/local`), ze względu na to, że brama Java podczas instalacji tworzy całą strukturę katalogów, a nie jeden plik wykonywalny.

```
$ ./configure --enable-java --prefix=$PREFIX
```

Żeby skompilować pakiet bramy Java do pliku JAR, należy uruchomić `make`. Krok ten wymaga, by programy `javac` i `jar`, były dostępne ze ścieżki wyszukiwania.

```
$ make
```

Po tym w katalogu `src/zabbix_java/bin` powinien się pojawić plik `zabbix-java-gateway-$VERSION.jar`. W tym momencie można już uruchamiać bramę Java z katalogu dystrybucji w `src/zabbix_java`, i jeżeli ma tak być to można od razu przejść do instrukcji konfiguracji i uruchamiania bramy Java. W przeciwnym razie, należy się upewnić, że posiadamy odpowiednie uprawnienia do uruchomienia `make install`.

```
$ make install
```

5.2 Przegląd plików w dystrybucji bramy Java

Niezależnie od sposobu pobrania bramy Java, powinniśmy mieć kolekcję skryptów powłoki, plików JAR i plików konfiguracyjnych w katalogu `$PREFIX/sbin/zabbix_java`. Znaczenie poszczególnych plików jest opisane krótko poniżej.

```
bin/zabbix-java-gateway-$VERSION.jar
```

Główny plik JAR bramy Java.

```
lib/logback-core-0.9.27.jar  
lib/logback-classic-0.9.27.jar  
lib/slf4j-api-1.6.1.jar  
lib/android-json-4.3_r3.1.jar
```

Zależności bramy Java: [Logback](#), [SLF4J](#), i biblioteka [Android JSON](#) (do wersji Zabbix 2.2.5 używana była biblioteka [JSON.org](#)).

```
lib/logback.xml  
lib/logback-console.xml
```

Pliki konfiguracji dla Logback.

```
shutdown.sh
```

```
startup.sh
```

Skrypty uruchamiające i zatrzymujące bramę Java.

```
settings.sh
```

Plik konfiguracji wykorzystywany przez powyższe skrypty uruchamiania i zatrzymywania.

5.3 Konfiguracja i uruchamianie bramy Java

Domyślnie brama Java słucha na porcie 10052. Jeżeli planujemy uruchomienie bramy Java na innym porcie, można go ustawić w skrypcie settings.sh. Opis ustawianie tej i innych opcji można zobaczyć w [plik konfiguracji bramy Java](#).

Port 10052 nie jest portem [zarejestrowanym w IANA](#).

Po ustawieniu wszystkich opcji, można uruchomić bramę Java skrypcem uruchomieniowym:

```
$ ./startup.sh
```

Natomiast, gdy już nie jest potrzebna brama Java, należy uruchomić skrypt zatrzymujący ją:

```
$ ./shutdown.sh
```

Należy zauważyć, że w przeciwieństwie do serwera czy proxy, brama Java nie potrzebuje bazy danych.

5.4 Konfiguracja serwera do użycia bramy Java

Gdy brama Java jest już uruchomiona, należy wskazać serwerowi Zabbix gdzie ma szukać bramy Java. Można to zrobić ustawiając parametry JavaGateway i JavaGatewayPort w [pliku konfiguracyjnym serwera](#). Jeżeli host, na którym jest uruchomiona aplikacja JMX, jest monitorowany przez proxy Zabbix, zamiast tego należy ustawić parametry połączenia w [pliku konfiguracyjnym proxy](#).

```
JavaGateway=192.168.3.14  
JavaGatewayPort=10052
```

Domyślnie serwer nie uruchamia żadnych procesów związanych z monitorowaniem JMX. Jeżeli chcemy ich użyć, należy ustawić liczbę wstępnie uruchomionych instancji kolejek Java. Można to zrobić w taki sam sposób, w jaki się ustawia zwykłe kolejki i pułapki.

```
StartJavaPollers=5
```

Nie należy zapominać o zrestartowaniu serwera lub proxy po ukończeniu konfiguracji.

5.5 Śledzenie błędów bramy Java

W przypadku wystąpienia błędów w bramie Java lub gdy komunikat wyświetlany w interfejsie użytkownika dla pozycji nie jest wystarczająco jasny, można się przyjrzeć plikowi logu bramy Java.

Domyślnie brama Java loguje swoje działania do pliku logu /tmp/zabbix_java.log z poziomem "info". Czasami informacje te nie są wystarczające i trzeba przestawić poziom logowania na "debug". W przypadku konieczności zwiększenia poziomu logowania, należy zmodyfikować plik lib/logback.xml i zmienić atrybut level w tagu <root> na "debug":

```
<root level="debug">
  <appender-ref ref="FILE" />
</root>
```

Należy zauważyć, że w przeciwieństwie do serwera lub proxy Zabbix, nie ma konieczności restartowania bramy Java Zabbix po zmianie pliku logback.xml - zmiany w pliku logback.xml zostaną zaaplikowane automatycznie. Po zakończeniu śledzenia można wrócić z poziomem logowania na "info".

Jeżeli chcemy logować do innego pliku, lub na kompletnie inne medium (baza danych), należy zmienić plik logback.xml zgodnie z wymaganiami. Więcej szczegółów można znaleźć w [Podręczniku Logback](#).

Czasami w celu śledzenia błędów lepiej jest uruchomić bramę Java z konsoli zamiast jako demon. Żeby to zrobić, wystarczy zakomentować zmienną PID_FILE w settings.sh. Jeżeli nie ma PID_FILE, skrypt startup.sh uruchomi bramę Java jako aplikację konsoli a Logback używać będzie pliku lib/logback-console.xml, w którym ustawione jest logowanie z poziomem "debug", na konsolę.

Na koniec, należy zauważyć, że brama Java używa logowania SLF4J, więc można zamienić Logback na inny wybrany pakiet, umieszczając przy tym odpowiedni plik JAR w katalogu lib. Więcej szczegółów można zobaczyć w [Podręczniku SLF4J](#).

From:

<https://www.zabbix.com/documentation/2.2/> - **Zabbix Documentation 2.2**

Permanent link:

<https://www.zabbix.com/documentation/2.2/pl/manual/concepts/java>

Last update: **2016/01/27 13:53**

