

## 14 JMX мониторинг

### Обзор

Мониторинг JMX можно использовать для наблюдения за счетчиками JMX в Java приложениях.

В Zabbix 2.0 добавлена встроенная поддержка мониторинга JMX, был выпущен новый Zabbix демон, так называемый “Zabbix Java gateway”.

Когда Zabbix сервер хочет узнать значение конкретного счетчика JMX у узла сети, он опрашивает Zabbix **Java gateway**, который в свою очередь используя [API управление JMX](#), удаленно опрашивает интересующее приложение.

Для получения более подробных сведений, включая где можно взять Zabbix Java gateway и как его настроить, смотрите [этот раздел](#) руководства.

Связь между Java gateway и наблюдаемым JMX приложением не должна быть закрыта брандмауэром.

### Включение удаленного JMX мониторинга для Java приложений

Приложению Java не требуется какое-либо дополнительно установленное программное обеспечение, но для поддержки удаленного мониторинга JMX приложение должно быть запущено с указанными ниже параметрами командной строки.

Как минимум, если вы просто хотите начать наблюдение за простым приложением Java на локальном хосте без каких либо защиты, запустите его со следующими опциями:

```
java \  
-Dcom.sun.management.jmxremote \  
-Dcom.sun.management.jmxremote.port=12345 \  
-Dcom.sun.management.jmxremote.authenticate=false \  
-Dcom.sun.management.jmxremote.ssl=false \  
-jar /usr/share/doc/openjdk-6-jre-headless/demo/jfc/Notepad/Notepad.jar
```

С этими аргументами Java будет слушать входящие соединения JMX на порту 12345, только с локальных хостов, без обязательных аутентификации или SSL.

Если вы хотите разрешить подключения с другого интерфейса, укажите параметр -Djava.rmi.server.hostname равным IP адресу этого интерфейса.

Если вы хотите иметь более строгую проверку в плане безопасности, есть много других опций в Java, которые вам доступны. Например, следующая иллюстрация запускает приложение с более универсальным набором опций и открывает это приложение для более широкой сети, не только для локального компьютера.

```
java \  
-Djava.rmi.server.hostname=192.168.3.14 \  
-Dcom.sun.management.jmxremote \  
-Dcom.sun.management.jmxremote.port=12345 \  
-jar /usr/share/doc/openjdk-6-jre-headless/demo/jfc/Notepad/Notepad.jar
```

```
-Dcom.sun.management.jmxremote.authenticate=true \  
-Dcom.sun.management.jmxremote.password.file=/etc/java-6-  
openjdk/management/jmxremote.password \  
-Dcom.sun.management.jmxremote.access.file=/etc/java-6-  
openjdk/management/jmxremote.access \  
-Dcom.sun.management.jmxremote.ssl=true \  
-Djavax.net.ssl.keyStore=$ВАШЕ_ХРАНИЛИЩЕ_КЛЮЧЕЙ \  
-Djavax.net.ssl.keyStorePassword=$ВАШ_ПАРОЛЬ_К_ХРАНИЛИЩУ_КЛЮЧЕЙ \  
-Djavax.net.ssl.trustStore=$ВАШЕ_ДОВЕРЕННОЕ_ХРАНИЛИЩЕ \  
-Djavax.net.ssl.trustStorePassword=$ВАШ_ПАРОЛЬ_К_ДОВЕРЕННОМУ_ХРАНИЛИЩУ \  
-Dcom.sun.management.jmxremote.ssl.need.client.auth=true \  
-jar /usr/share/doc/openjdk-6-jre-headless/demo/jfc/Notepad/Notepad.jar
```

Значительное количество (если не все) этих настроек можно указать в `/etc/java-6-openjdk/management/management.properties` (или там, где этот файл расположен на вашем компьютере).

Обратите внимание, если вы желаете использовать SSL, то вы должны изменить `startup.sh` скрипт `java gateway`, добавив в него опции `-Djavax.net.ssl.*` так, чтобы он знал где искать хранилище ключей и доверенное хранилище.

Смотрите [Мониторинг и Управление с использованием JMX \[en\]](#) для получения более подробной информации.

## Настройка JMX интерфейсов и элементов данных в веб-интерфейсе Zabbix

Когда `Java Gateway` запущен, сервер знает где его искать и `Java` приложение запущено с поддержкой удаленного JMX мониторинга, самое время настроить интерфейсы и элементы данных в Веб-интерфейсе Zabbix.

### Настройка JMX интерфейса

Начнем с создания интерфейса JMX-типа у интересующего узла сети.

### Hosts

Host Templates IPMI Macros Host inventory Encryption

\* Host name

Visible name

\* Groups

In groups:

Other groups:

New group

\* At least one interface must exist.

Agent interfaces	IP address	DNS name	Connect to	Port	Default
<input type="text" value="127.0.0.1"/>	<input type="text"/>	<input type="text"/>	<input checked="" type="radio"/> IP <input type="radio"/> DNS	<input type="text" value="10050"/>	<input checked="" type="radio"/> <a href="#">Remove</a>
<a href="#">Add</a>					

SNMP interfaces [Add](#)

JMX interfaces	IP address	DNS name	Connect to	Port	Default
<input type="text" value="127.0.0.1"/>	<input type="text"/>	<input type="text"/>	<input checked="" type="radio"/> IP <input type="radio"/> DNS	<input type="text" value="12345"/>	<input checked="" type="radio"/> <a href="#">Remove</a>
<a href="#">Add</a>					

#### Добавление элемента данных JMX агента

Для каждого интересующего вас счетчика JMX вам необходимо добавить элемент данных с типом **JMX агент** присоединенный к этому интерфейсу.

Ключ на снимке экрана ниже имеет следующий вид `jmx["java.lang:type=Memory", "HeapMemoryUsage.used"]`.

Item Preprocessing

\* Name

Type

\* Key

\* Host interface

\* JMX endpoint

User name

Password

Type of information

Units

\* Update interval

Custom intervals

Type	Interval	Period	Action
Flexible Scheduling	50s	1-7,00:00-24:00	<a href="#">Remove</a>

[Add](#)

\* History storage period

\* Trend storage period

Show value  [show value mappings](#)

New application

Applications

Populates host inventory field

Description

Enabled

Все обязательные поля ввода отмечены красной звёздочкой.

Поля, требующие специфичной информации для JMX элементов данных:

Тип	Укажите здесь <b>JMX агент</b> .
-----	----------------------------------

Ключ	Ключ элемента данных <code>jmx[ ]</code> состоит из двух параметров: <b>имя объекта</b> - имя объекта MBean; <b>имя атрибута</b> - имя атрибута MBean с опциональными составными данными имен полей, разделенных точками. Смотрите ниже для получения более подробных сведений о ключах элементов данных JMX. Начиная с Zabbix 3.4, вы можете обнаруживать MBeans и MBean атрибуты, используя элемент данных <code>jmx.discovery[ ]</code> <a href="#">низкоуровневого обнаружения</a> .
JMX endpoint	Вы можете указать пользовательский JMX endpoint. Убедитесь, что параметры подключения JMX endpoint совпадают с JMX интерфейсом. Это можно сделать при помощи макросов <code>{HOST.*}</code> , как это сделано в JMX endpoint по умолчанию. Это поле поддерживается начиная с 3.4.0. Поддерживаются <a href="#">макросы</a> <code>{HOST.*}</code> и пользовательские макросы.
Имя пользователя	Укажите имя пользователя, если вы настроили аутентификацию у вашего Java приложения. Поддерживаются пользовательские макросы.
Пароль	Укажите пароль, если вы настроили аутентификацию у вашего Java приложения. Поддерживаются пользовательские макросы.

Если вы хотите наблюдать за Логическим счетчиком, который может быть “true” или “false”, вы должны указать тип информации “Числовой (целое положительное)” и “Логический” тип данных. Сервер будет записывать Логические значения как 1 или 0, соответственно.

## Детальная информация о ключах JMX элементов данных

### Простые атрибуты

Имя объекта MBean неважно, кроме строки, которую вы определили в вашем Java приложении. Имя атрибута, с другой стороны, может быть более сложным. В случае, если атрибут возвращает простой тип данных (число, строку и т.п.), то не стоит волноваться об этом, ключ будет выглядеть примерно так:

```
jmx[com.example:Type=Hello,weight]
```

В этом примере именем объекта является “com.example:Type=Hello”, именем атрибута будет являться “weight” и, скорее всего, тип возвращаемого значения должен быть “Числовой (с плавающей точкой)”.

### Атрибуты возвращающие составные данные

Ключ становится более сложным, когда ваш атрибут возвращает составные данные. Например: именем вашего атрибута является “apple” и он возвращает хэш представляющих его параметров, таких как “weight”, “color” и прочее. Тогда ваш ключ может выглядеть примерно так:

```
jmx[com.example:Type=Hello,apple.weight]
```

Этот пример показывает как разделяются с помощью точки имя атрибута и ключ хэша. Точно также, если атрибут возвращает часть вложенных составных данных, их нужно снова разделить точкой:

```
jmx[com.example:Type=Hello,fruits.apple.weight]
```

### Проблема с точками

Пока все хорошо. Но что, если имя атрибута или ключ хэша содержит символ точки? Вот пример:

```
jmx[com.example:Type=Hello,all.fruits.apple.weight]
```

Это проблема. Как сказать Zabbix'у, что имя атрибута "all.fruits", а не просто "all"? Как отличить точку, которая является частью имени, от точки которая разделяет имя атрибута и ключи хэшей?

До **2.0.4** Zabbix Java gateway был не способен справиться с такими ситуациями и пользователи оставались с НЕПОДДЕРЖИВАЕМЫМИ элементами данных. Начиная с 2.0.4 проблема была исправлена, все что вам требуется сделать - экранировать точки, которые являются частью имени, обратной косой чертой:

```
jmx[com.example:Type=Hello,all\.fruits.apple.weight]
```

Аналогично, если ваш ключ хэша содержит точку вам необходимо её экранировать:

```
jmx[com.example:Type=Hello,all\.fruits.apple.total\.weight]
```

### Другие проблемы

Символ обратной косой черты тоже должен быть экранирован:

```
jmx[com.example:type=Hello,c:\\documents]
```

Для обработки любых других символов в ключе JMX элемента данных, пожалуйста, смотрите [раздел](#) формата ключа элементов данных.

На самом деле это все, что нужно сделать. Успешного мониторинга JMX!

### Непримитивные типы данных

Начиная с Zabbix 4.0.0 имеется возможность работы с пользовательскими MBean, которые возвращают непримитивные типы данных, которые переопределяют метод **toString()**.

## Пример пользовательского endpoint с JBoss EAP 6.4

Пользовательские endpoint позволяют работать с различными транспортными протоколами, которые отличаются от протокола по умолчанию RMI.

Для иллюстрации этой возможности в качестве примера давайте попытаемся настроить JBoss EAP 6.4. Во-первых, давайте сделаем некоторые предположения:

- У вас уже имеется установленный Zabbix Java gateway. Если нет, тогда вам нужно сделать это в соответствии с [документацией](#).
- Zabbix сервер и Java gateway установлены с префиксом /usr/local/.
- JBoss уже установлен в /opt/jboss-eap-6.4/ и запущен в автономном режиме.
- Мы будем считать, что все эти компоненты работают на одном и том же хосте.
- Брандмауэр и SELinux отключены (или настроены соответствующим образом).

Давайте выполним некоторые простые настройки в zabbix\_server.conf:

```
JavaGateway=127.0.0.1
StartJavaPollers=5
```

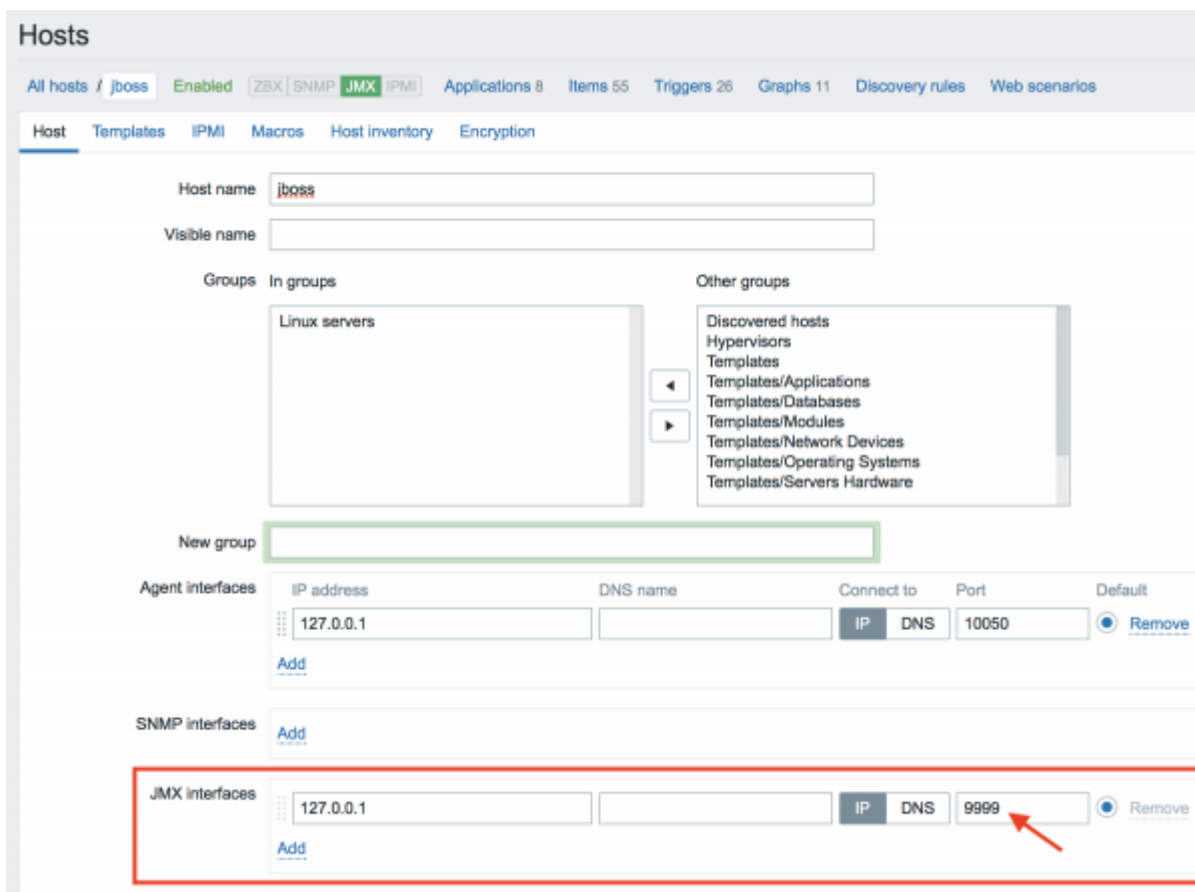
И в файле конфигурации zabbix\_java/settings.sh (или zabbix\_java\_gateway.conf):

```
START_POLLERS=5
```

Проверьте, что JBoss слушает свой стандартный порт управления:

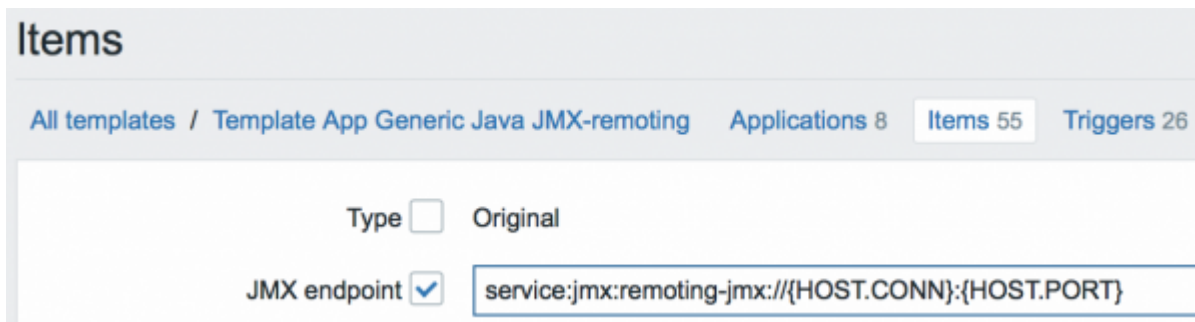
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999      0.0.0.0:*           LISTEN
10148/java
```

Теперь давайте создадим в Zabbix узел сети с JMX интерфейсом 127.0.0.1:9999.



Как мы знаем эта версия JBoss использует протокол JBoss Remoting вместо RMI, мы можем использовать массовое обновление параметра JMX endpoint в нашем шаблоне JMX в соответствии:

```
service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}
```



Давайте обновим кэш конфигурации:

```
$ /usr/local/sbin/zabbix_server -R config_cache_reload
```

Обратите внимание, что сначала может возникнуть ошибка.



```

3. mc [root@centos7-dev]~/home/vagrant/zabbix-3.2.6/src/zabbix_java (ssh)
com.zabbix.gateway.ZabbixException: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
    at com.zabbix.gateway.JMXItemChecker.getValues(JMXItemChecker.java:97) ~[zabbix-java-gateway-3.4.2.jar:na]
    at com.zabbix.gateway.SocketProcessor.run(SocketProcessor.java:63) ~[zabbix-java-gateway-3.4.2.jar:na]
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_144]
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_144]
    at java.lang.Thread.run(Thread.java:748) [na:1.8.0_144]
Caused by: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
    at javax.management.remote.JMXConnectorFactory.newJMXConnector(JMXConnectorFactory.java:359) ~[na:1.8.0_144]
    at javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:269) ~[na:1.8.0_144]
    at com.zabbix.gateway.ZabbixJMXConnectorFactory$1.run(ZabbixJMXConnectorFactory.java:76) ~[zabbix-java-gatewa
-3.4.2.jar:na]
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ~[na:1.8.0_144]
    at java.util.concurrent.FutureTask.run(FutureTask.java:266) ~[na:1.8.0_144]
    ... 3 common frames omitted
2017-11-07 13:52:12.644 [pool-1-thread-1] WARN com.zabbix.gateway.SocketProcessor - error processing request
com.zabbix.gateway.ZabbixException: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
    at com.zabbix.gateway.JMXItemChecker.getValues(JMXItemChecker.java:97) ~[zabbix-java-gateway-3.4.2.jar:na]
    at com.zabbix.gateway.SocketProcessor.run(SocketProcessor.java:63) ~[zabbix-java-gateway-3.4.2.jar:na]
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_144]
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_144]
    at java.lang.Thread.run(Thread.java:748) [na:1.8.0_144]
Caused by: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
    at javax.management.remote.JMXConnectorFactory.newJMXConnector(JMXConnectorFactory.java:359) ~[na:1.8.0_144]
    at javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:269) ~[na:1.8.0_144]
    at com.zabbix.gateway.ZabbixJMXConnectorFactory$1.run(ZabbixJMXConnectorFactory.java:76) ~[zabbix-java-gatewa
-3.4.2.jar:na]
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ~[na:1.8.0_144]
    at java.util.concurrent.FutureTask.run(FutureTask.java:266) ~[na:1.8.0_144]
    ... 3 common frames omitted
2017-11-07 13:52:14.889 [Thread-0] INFO com.zabbix.gateway.JavaGateway - Zabbix Java Gateway 3.4.2 (revision 72885)
as stopped
2017-11-07 13:52:26.167 [main] INFO com.zabbix.gateway.JavaGateway - Zabbix Java Gateway 3.4.2 (revision 72885) has
tarted

```

“Unsupported protocol: remoting-jmx” означает, что Java gateway не знает как работать с указанным протоколом. Эту ошибку можно исправить создав файл ~/needed\_modules.txt со следующим содержимым:

```

jboss-as-remoting
jboss-logging
jboss-logmanager
jboss-marshalling
jboss-remoting
jboss-sasl
jcl-over-slf4j
jul-to-slf4j-stub
log4j-jboss-logmanager
remoting-jmx
slf4j-api
xnio-api
xnio-nio</pre>

```

и затем выполнив эту команду:

```

$ for i in $(cat ~/needed_modules.txt); do find /opt/jboss-eap-6.4 -iname
${i}*.jar -exec cp {} /usr/local/sbin/zabbix_java/lib/ \; ; done

```

Таким образом, у Java gateway будут в наличии все необходимые модули для работы с jmx-remoting. Осталось только перезапустить Java gateway, немного подождать и, если вы все сделали правильно, вы увидите, что эти данные JMX мониторинга начинают поступать в Zabbix:

Latest data

Filter

Name	Last check	Last value	Change
<b>Classes (3 items)</b>			
<input type="checkbox"/> d Loaded Class Count	2017-11-07 14:00:13	7968	+2
<input type="checkbox"/> d Total Loaded Class Count	2017-11-07 14:00:09	7968	+2
<input type="checkbox"/> d Unloaded Class Count	2017-11-07 14:00:13	0	
<b>Compilation (2 items)</b>			
<input type="checkbox"/> comp Accumulated time spent in compilation	2017-11-07 14:00:13	40s 750ms	+1s 440ms
<input type="checkbox"/> comp Name of the current JIT compiler	2017-11-07 14:00:38	HotSpot 64-Bit Tiered Compiler	
<b>Garbage Collector (4 items)</b>			
<input type="checkbox"/> gc Copy accumulated time spent in collection	2017-11-07 14:00:09	0	
<input type="checkbox"/> gc Copy number of collections per second	2017-11-07 14:00:09	0	
<input type="checkbox"/> gc MarkSweepCompact accumulated time spent in collection	2017-11-07 14:00:13	372ms	
<input type="checkbox"/> gc MarkSweepCompact number of collections per second	2017-11-07 14:00:13	0	
<b>Memory (5 items)</b>			
<input type="checkbox"/> mem Heap Memory committed	2017-11-07 14:00:13	1.23 GB	
<input type="checkbox"/> mem Heap Memory max	2017-11-07 14:00:38	1.23 GB	
<input type="checkbox"/> mem Heap Memory used	2017-11-07 14:00:09	271.07 MB	+4.01 MB
<input type="checkbox"/> mem Non-Heap Memory committed	2017-11-07 14:00:13	66.36 MB	+364 KB
<input type="checkbox"/> mem Non-Heap Memory used	2017-11-07 14:00:13	89.5 MB	+128.1 KB
<input type="checkbox"/> mem Object Pending Finalization Count	2017-11-07 14:00:13	0	
<b>Memory Pool (6 items)</b>			
<input type="checkbox"/> mp Code Cache committed	2017-11-07 14:00:09	12.31 MB	+128 KB
<input type="checkbox"/> mp Code Cache max	2017-11-07 14:00:40	240 MB	
<input type="checkbox"/> mp Code Cache used	2017-11-07 14:00:09	12.23 MB	+145.44 KB
<input type="checkbox"/> mp Tenured Gen committed	2017-11-07 14:00:13	899.38 MB	
<input type="checkbox"/> mp Tenured Gen max	2017-11-07 14:00:40	899.38 MB	
<input type="checkbox"/> mp Tenured Gen used	2017-11-07 14:00:09	32.25 MB	

From: <https://www.zabbix.com/documentation/4.2/> - Zabbix Documentation 4.2

Permanent link: [https://www.zabbix.com/documentation/4.2/ru/manual/config/items/itemtypes/jmx\\_monitoring](https://www.zabbix.com/documentation/4.2/ru/manual/config/items/itemtypes/jmx_monitoring)

Last update: 2018/07/25 02:03

