Configuring JMX exporter for Kafka and Zookeeper

May 12, 2018

I've been using Prometheus for quite some time and really enjoying it. Most of the things are quite simple – installing and configuring Prometheus is easy, setting up exporters is launch and forget, instrumenting your code (/blog/go-prometheus-service/) is a bliss. But there are 2 things that I've really struggled with:

1. Grokking data model and PromQL to get meaningful insights.
2. Configuring jmx-exporter.

In this post, I'll share the JMX part because I don't feel that I've fully understood the data model and PromQL. So let's dive into that jmx-exporter thing.

What is jmx-exporter

jmx-exporter (https://github.com/prometheus/jmx_exporter) is a program that reads JMX data from JVM based applications (e.g. Java and Scala) and exposes it via HTTP in a simple text format that Prometheus understand and can scrape.

JMX is a common technology in Java world for exporting statistics of running application and also to control it (you can trigger GC with JMX, for example).

jmx-exporter is a Java application that uses JMX APIs to collect the app and JVM metrics. It is Java agent which means it is running inside the same JVM. This gives you a nice benefit of not exposing JMX remotely – jmx-exporter will just collect the metrics and exposes it over HTTP in read-only mode.

Installing jmx-exporter

Because it's written in Java, jmx-exporter is distributed as a jar, so you just need to download it from maven (https://repo1.maven.org/maven2/io/prometheus/jmx/jmx_prometheus_javaagent/0.3.0/jmx_prometheus_javaagent-0.3.0.jar) and put it somewhere on your target host.

I have an Ansible role for this – https://github.com/alexdzyoba/ansible-jmx-exporter (https://github.com/alexdzyoba/ansible-jmx-exporter). Besides downloading the jar it'll also put the configuration file for jmx-exporter.

This configuration file contains rules for rewriting JMX MBeans to the Prometheus exposition format metrics. Basically, it's a collection of regexps to convert MBeans strings to Prometheus strings.

The example_configs directory (https://github.com/prometheus/jmx_exporter/tree/master/example_configs) in jmx-exporter sources contains examples for many popular Java apps including Kafka and Zookeeper.

Configuring Zookeeper with jmx-exporter

As I've said jmx-exporter runs inside other JVM as java agent to collect JMX metrics. To demonstrate how it all works, let's run it within Zookeeper.

Zookeeper is a crucial part of many production systems including Hadoop, Kafka and Clickhouse, so you really want to monitor it. Despite the fact that you can do this with 4lw commands (https://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_zkCommands) (`mntr` , `stat` , etc.) and that there are (https://github.com/dln/zookeeper_exporter) dedicated (https://github.com/lucianjon/zk-exporter) exporters (https://github.com/dabealu/zookeeper-exporter) I prefer to use JMX to avoid constant Zookeeper querying (they add noise to metrics because 4lw commands counted as normal Zookeeper requests).

To scrape Zookeeper JMX metrics with jmx-exporter you have to pass the following arguments to Zookeeper launch:

```
-javaagent:/opt/jmx-exporter/jmx-exporter.jar=7070:/etc/jmx-exporter/zookeeper.yml
```

If you use the Zookeeper that is distributed with Kafka (you shouldn't) then pass it via `EXTRA_ARGS` :

```
$ export EXTRA_ARGS="-javaagent:/opt/jmx-exporter/jmx-exporter.jar=7070:/etc/jmx-exporter/zookeeper.yml"
$ /opt/kafka_2.11-0.10.1.0/bin/zookeeper-server-start.sh /opt/kafka_2.11-0.10.1.0/config/zookeeper.properties
```

If you use standalone Zookeeper distribution then add it as SERVER_JVMFLAGS to the zookeeper-env.sh:

```
# zookeeper-env.sh
SERVER_JVMFLAGS="-javaagent:/opt/jmx-exporter/jmx-exporter.jar=7070:/etc/jmx-exporter/zookeeper.yml"
```

Anyway, when you launch Zookeeper you should see the process listening on the specified port (7070 in my case) and responding to `/metrics` queries:

```
$ netstat -tlnp | grep 7070
tcp        0      0 0.0.0.0:7070            0.0.0.0:*               LISTEN      892/java

$ curl -s localhost:7070/metrics | head
# HELP jvm_threads_current Current thread count of a JVM
# TYPE jvm_threads_current gauge
jvm_threads_current 16.0
# HELP jvm_threads_daemon Daemon thread count of a JVM
# TYPE jvm_threads_daemon gauge
jvm_threads_daemon 12.0
# HELP jvm_threads_peak Peak thread count of a JVM
# TYPE jvm_threads_peak gauge
jvm_threads_peak 16.0
# HELP jvm_threads_started_total Started thread count of a JVM
```

Configuring Kafka with jmx-exporter

Kafka is a message broker written in Scala so it runs in JVM which in turn means that we can use jmx-exporter for its metrics.

To run jmx-exporter within Kafka, you should set `KAFKA_OPTS` environment variable like this:

```
$ export KAFKA_OPTS='-javaagent:/opt/jmx-exporter/jmx-exporter.jar=7071:/etc/jmx-exporter/kafka.yml'
```

Then launch the Kafka (I assume that Zookeeper is already launched as it's required by Kafka):

```
$ /opt/kafka_2.11-0.10.1.0/bin/kafka-server-start.sh /opt/kafka_2.11-0.10.1.0/conf/server.properties
```

Check that jmx-exporter HTTP server is listening:

```
$ netstap -tlnp | grep 7071
tcp6       0      0 :::7071                 :::*                    LISTEN      19288/java
```

And scrape the metrics!

```
$ curl -s localhost:7071 | grep -i kafka | head
# HELP kafka_server_replicafetchermanager_minfetchrate Attribute exposed for management (kafka.server<type=ReplicaFet
# TYPE kafka_server_replicafetchermanager_minfetchrate untyped
kafka_server_replicafetchermanager_minfetchrate{clientId="Replica",} 0.0
# HELP kafka_network_requestmetrics_totaltimems Attribute exposed for management (kafka.network<type=RequestMetrics,
# TYPE kafka_network_requestmetrics_totaltimems untyped
kafka_network_requestmetrics_totaltimems{request="OffsetFetch",} 0.0
kafka_network_requestmetrics_totaltimems{request="JoinGroup",} 0.0
kafka_network_requestmetrics_totaltimems{request="DescribeGroups",} 0.0
kafka_network_requestmetrics_totaltimems{request="LeaveGroup",} 0.0
kafka_network_requestmetrics_totaltimems{request="GroupCoordinator",} 0.0
```

Here is how to run jmx-exporter java agent if you are running Kafka under systemd:

```
...
[Service]
Restart=on-failure
Environment=KAFKA_OPTS=-javaagent:/opt/jmx-exporter/jmx-exporter.jar=7071:/etc/jmx-exporter/kafka.yml
ExecStart=/opt/kafka/bin/kafka-server-start.sh /etc/kafka/server.properties
ExecStop=/opt/kafka/bin/kafka-server-stop.sh
TimeoutStopSec=600
User=kafka
...
```

Recap

With jmx-exporter you can scrape the metrics of running JVM applications. jmx-exporter runs as a Java agent (inside the target JVM) scrapes JMX metrics, rewrite it according to config rules and exposes it in Prometheus exposition format.

For a quick setup check my Ansible role for jmx-exporter (https://github.com/alexdzyoba/ansible-jmx-exporter) alexdzyoba.jmx-exporter (https://galaxy.ansible.com/alexdzyoba/jmx-exporter/).

That's all for now, stay tuned by subscribing to the RSS (https://alex.dzyoba.com/feed) or follow me on Twitter @AlexDzyoba (https://twitter.com/alexdzyoba).
