Kafka Quick Start Guide

Posted on Wednesday, September 30, 2015 · 2 Comments (http://sookocheff.com/post/kafka/kafka-quick-start/#disqus_thread)

If you've read the previous article describing Kafka in a Nutshell (http://sookocheff.com/post/kafka/kafka-in-a-nutshell/) you may be itching to write an application using Kafka as a data backend. This article will get you part of the way there by describing how to deploy Kafka locally using Docker (https://www.docker.com/) and test it using kafkacat (https://github.com/edenhill/kafkacat).

Running Kafka Locally

First, if you haven't already, download and install Docker (http://docs.docker.com/mac/step_one/). Once you have Docker installed, create a default virtual machine that will host your local Docker containers.

```
> docker-machine create --driver virtualbox default
```

Clone the Kafka docker repository

You could use the Docker pull command here but I find it instructive to be able to view the source files for your container.

```
> git clone https://github.com/wurstmeister/kafka-docker
```

Set a default topic

Open `docker-compose-single-broker.yml` and set a default topic and advertised name. You will want to use the IP address of your default Docker machine. Copy it to the clipboard with the following command.

```
> docker-machine ip default | pbcopy
```

In `docker-compose-single-broker.yml`, edit the `KAFKA_ADVERTISED_HOST_NAME` with the IP address you copied above and the `KAFKA_CREATE_TOPICS` with the name of the default topic you would like created. The `1:1` refers to the number of partition and the replication factor for your partition.

```
environment:
  KAFKA\_ADVERTISED\_HOST\_NAME: 192.168.99.100  # IP address pasted to clipboard above
  KAFKA\_CREATE\_TOPICS: "test:1:1"
```

Run your Docker container

Running the Docker container will start an instance of Zookeeper and Kafka that you can test locally.

```
> docker-compose -f docker-compose-single-broker.yml up -d
```

You will receive a message saying the Zookeeper and Kafka are up and running How come?

```
Starting kafkadocker\_zookeeper\_1...
Starting kafkadocker\_kafka\_1...
```

Interacting with Kafka

kafkacat (https://github.com/edenhill/kafkacat) provides a generic command line interface to
Kafka for producing and consuming data. You can install kafkacat with Homebrew
(http://brew.sh/).

```
> brew install kafkacat
```

Once installed, interacting with Kafka is relatively simple. kafkacat provides two modes,
consumer and producer. You enter producer mode with the `-P` option. The `-b` option specifies
the Kafka broker to talk to and the `-t` option specifies the topic to produce to. Kafka runs on
port `9092` with an IP address machine that of our Virtual Machine. Running the following
command will open stdin to receive messages, simply type each message followed by `Enter` to
produce to your Kafka broker.

```
> kafkacat -P -b $(docker-machine ip default):9092 -t test
Kafka
quick
start
guide.
^C
```

To consume messages, you run kafkacat with the `-C` option. Kafka will replay the messages you
have sent as a producer.

```
> kafkacat -C -b $(docker-machine ip default):9092 -t test
Kafka
quick
start
guide.
```

Where to go from here?

Now that you have a running Kafka broker there are a number of things you can do to become more familiar with Kafka. First, I recommend reading through the kafkacat examples (https://github.com/edenhill/kafkacat) to become familiar with producing and consuming messages. Next, you can use one of the many Kafka clients (https://cwiki.apache.org/confluence/display/KAFKA/Clients) to produce and consume messages from your application.

Share this post:
🐦 (https://twitter.com/intent/tweet?url=http%3a%2f%2fsookocheff.com%2fpost%2fkafka%2fkafka-quick-start%2f&text=Kafka%20Quick%20Start%20Guide&via=soofaloofa)
f (https://facebook.com/sharer.php?u=http%3a%2f%2fsookocheff.com%2fpost%2fkafka%2fkafka-quick-start%2f)
g+ (https://plus.google.com/share?url=http%3a%2f%2fsookocheff.com%2fpost%2fkafka%2fkafka-quick-start%2f)
in (http://www.linkedin.com/shareArticle?mini=true&url=http%3a%2f%2fsookocheff.com%2fpost%2fkafka%2fkafka-quick-start%2f)
✉ (mailto:?subject=Check out this link&body=http%3a%2f%2fsookocheff.com%2fpost%2fkafka%2fkafka-quick-start%2f)

Think this post was worth a dollar? If so, I'd love to have it. Support this site quickly and easily using the Paypal button below.

Amount of Support:
Hat Tip $1.00 CAD ▾

Pay Now

About Kevin Sookocheff

I am a Software Developer. I work at Workiva (http://www.workiva.com) with a bunch of great people. I run a business as Red Finch Software (http://redfinchsoftware.com). You can find me online as soofaloofa (https://twitter.com/soofaloofa). I love to teach, learn, speak, and write about business and software. I'm interested in distributed systems, cloud computing and machine learning. I currently work on data engineering, data workflows, and data processing at scale.

On the web: (http://www.twitter.com/soofaloofa)
 (http://www.facebook.com/kevin.sookocheff)
 (http://www.linkedin.com/in/kevinsookocheff)   (https://github.com/soofaloofa)
 (http://sookocheff.com/index.xml)

2 Comments   **Kevin Sookocheff**   ❶ Login ▾

♥ Recommend   ↱ Share   Sort by Best ▾

Join the discussion…

Zhang Jiaqiang • 3 months ago

Hi Kevin Sookocheff,

I followed the instructions and get the following error msg when I launched kafkacat producer
command:

%3|1465956180.100|FAIL|rdkafka#producer-1| 192.168.99.100:9092/bootstrap: Connect to
ipv4#192.168.99.100:9092 failed: Connection refused

This error msg keep poping in the screen so I have to kill the kafkacat by force.

If this issue is related to docker network env setting ?

Would you please give me some suggestion on how to investigate further on this.

Thanks for this article.

∧ | ∨ • Reply • Share ›

> **Kevin Sookocheff** Mod ↗ Zhang Jiaqiang • 3 months ago
>
> Are you using VirtualBox as your container host? You will have to make sure the correct port
> is forwarded. Here's a tutorial that might help: http://www.howtogeek.com/12264...
>
> ∧ | ∨ • Reply • Share ›

✉ **Subscribe**   Ⓓ **Add Disqus to your site Add Disqus Add**   🔒 **Privacy**   **DISQUS**

App Engine

MapReduce API (/series/mapreduce-api/)
Pipelines API (/series/pipelines-api/)
Cloud Endpoints (/series/cloud-endpoints/)

APIs

On Choosing a Hypermedia Format (/posts/2014-03-11-on-choosing-a-hypermedia-format/)
When to use HTTP PUT (/posts/2014-03-27-when-to-use-http-put-and-http-post/)

Random

Unity3d on the Mac App Store (http://sookocheff.com/posts/2013-04-09-submitting-a-unity3d-game-to-
the-mac-app-store/)
Saving Canvas Data to an Image (http://sookocheff.com/posts/2011-07-27-saving-canvas-data-to-an-
image-file-with-javascript-and-php/)

Social

Twitter (http://www.twitter.com/soofaloofa)
Facebook (http://www.facebook.com/kevin.sookocheff)
LinkedIn (http://www.linkedin.com/in/kevinsookocheff)

© Kevin Sookocheff · Top