

Die einzelnen Starter

```
# zookeeper.properties
# the directory where the snapshot is stored.
dataDir=/tmp/zookeeper
# the port at which the clients will connect
clientPort=2181
# disable the per-ip limit on the number of
# connections since this is a non-production
# config
maxClientCnxns=0
# Zookeeper starten
bin/zookeeper-server-start.sh config/zoo
-
keeper.properties
# server.properties
# The minimum age of a log file to be eligible for
# deletion due to age
logRetentionHours=168
# A size-based retention policy for logs. Segments
# are pruned from the log unless the remaining
# segments drop below logRetentionBytes.
# Functions independently of logRetentionHours.
#logRetentionBytes=1073741824
# The maximum size of a log segment file. When
# this size is reached a new log segment will be
# created.
logSegmentBytes=1073741824
# The interval at which log segments are checked to
# see if they can be deleted according
# to the retention policies
logRetentionCheckIntervalMs=30000
##### Zookeeper #####
#####
'zookeeper.connect=localhost:2181'
# Timeout in ms for connecting to zookeeper
zookeeper.connection.timeout.ms=6000
# Broker (Server) starten
bin/kafka-server-start.sh config/server.p
-
roperties
# Topic test erstellen
bin/kafka-topics.sh \
--create --zookeeper localhost:2181 \
--replICA tion-factor 1 \
--partitions 1 \
```

Die einzelnen Starter (cont)

```
--topic test \
# Topics auflisten
bin/kafka-topics.sh --list --zoo
-
keeper
localhost:2181
```

Setup / Installation

Kafka wurde von LinkedIn entwickelt und ist inzwischen ein Open Source Projekt der Apache Software Foundation. Die Firma hinter Kafka heißt Confluent, sie wurde von den Entwickler aus LinkedIn ausgegründet. Das bedeutet aber nicht, dass es nicht noch viele weitere Firmen gibt welche Teile des Kafka Universums bereitstellen.

Quickstart: <https://kafka.apache.org/quickstart>

Download: <https://www.apache.org/dyn/closer.cgi?path=/kafka/>

Docker Container: <https://hub.docker.com/r/confluent/kafka>

Ökosystem: <https://cwiki.apache.org/confluence/display/KAFKA/Ecosystem>

Offene Punkte: Mit ein wenig Historie fragt man sich warum es Cocoon nie zu einem Status wie Kafka geschafft hat oder was sind eigentlich die Unterschiede zwischen Cocoon und Kafka und lässt sich Kafka Streaming mit einer Cocoon Pipeline kombinieren. Hier gibts Infos zu [Cocoon](#)

Systemaufbau / Hierarchie

Im Kafka Universum wird die Welt wie folgt von abstrakt zu konkret herunter gebrochen:

- * Cluster
- * DataCenter
- * Rack
- * Instanz



By **Huluvu424242**
(FunThomas424242)

Published 26th May, 2019.
Last updated 2nd June, 2019.
Page 1 of 1.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish
Yours!
<https://apollopad.com>