

kafka 参数详解

```
# Licensed to the Apache Software Foundation (ASF) under one or
more
# contributor license agreements.  See the NOTICE file
distributed with
# this work for additional information regarding copyright
ownership.
# The ASF licenses this file to You under the Apache License,
Version 2.0
# (the "License"); you may not use this file except in compliance
with
# the License.  You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
software
# distributed under the License is distributed on an "AS IS"
BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
# See the License for the specific language governing permissions
and
# limitations under the License.
# see kafka.server.KafkaConfig for additional details and defaults
##### Server Basics
#####
#每一个broker在集群中的唯一表示，要求是正数。当该服务器的IP地址发
生改变时，broker.id没有变化，则不会影响consumers的消息情况
```

```
# The id of the broker. This must be set to a unique integer for
each broker.
broker.id=0
# Switch to enable topic deletion or not, default value is false
#delete.topic.enable=true
##### Socket Server Settings
#####
#broker server服务端口
# The address the socket server listens on. It will get the value
returned from
# java.net.InetAddress.getCanonicalHostName() if not configured.
#   FORMAT:
#       listeners = security_protocol://host_name:port
#   EXAMPLE:
#       listeners = PLAINTEXT://your.host.name:9092
#listeners=PLAINTEXT://:9092
# Hostname and port the broker will advertise to producers and
consumers. If not set,
# it uses the value for "listeners" if configured.  Otherwise, it
will use the value
# returned from java.net.InetAddress.getCanonicalHostName().
#advertised.listeners=PLAINTEXT://your.host.name:9092
#broker处理消息的最大线程数，一般情况下数量为cpu核数
# The number of threads handling network requests
num.network.threads=3
#broker处理磁盘IO的线程数，数值为cpu核数2倍
# The number of threads doing disk I/O
num.io.threads=8
#socket的发送缓冲区，socket的调优参数SO_SNDBUFF
# The send buffer (SO_SNDBUF) used by the socket server
```

```
socket.send.buffer.bytes=102400
#socket的接受缓冲区, socket的调优参数SO_RCVBUF
# The receive buffer (SO_RCVBUF) used by the socket server
socket.receive.buffer.bytes=102400
#socket请求的最大数值, 防止serverOOM, message.max.bytes必然要小于
socket.request.max.bytes, 会被topic创建时的指定参数覆盖
# The maximum size of a request that the socket server will accept
(protection against OOM)
socket.request.max.bytes=104857600
##### Log Basics
#####
#kafka数据的存放地址, 多个地址的话用逗号分割, 多个目录分布在不同磁
盘上可以提高读写性能 /data/kafka-logs-1, /data/kafka-logs-2
# A comma seperated list of directories under which to store log
files
log.dirs=/tmp/kafka-logs
#每个topic的分区个数, 若是在topic创建时候没有指定的话会被topic创建
时的指定参数覆盖
# The default number of log partitions per topic. More partitions
allow greater
# parallelism for consumption, but this will also result in more
files across
# the brokers.
num.partitions=1
#用于在启动时, 用于日志恢复的线程个数, 默认是1.
# The number of threads per data directory to be used for log
recovery at startup and flushing at shutdown.
# This value is recommended to be increased for installations with
data dirs located in RAID array.
num.recovery.threads.per.data.dir=1
```

```
##### Log Flush Policy
#####
#表示每当消息记录数达到1000时flush一次数据到磁盘
# Messages are immediately written to the filesystem but by
default we only fsync() to sync
# the OS cache lazily. The following configurations control the
flush of data to disk.
# There are a few important trade-offs here:
#     1. Durability: Unflushed data may be lost if you are not
using replication.
#     2. Latency: Very large flush intervals may lead to latency
spikes when the flush does occur as there will be a lot of data to
flush.
#     3. Throughput: The flush is generally the most expensive
operation, and a small flush interval may lead to excessive
seeks.
# The settings below allow one to configure the flush policy to
flush data after a period of time or
# every N messages (or both). This can be done globally and
overridden on a per-topic basis.
# The number of messages to accept before forcing a flush of data
to disk
#log.flush.interval.messages=10000
#表示每间隔1000毫秒flush一次数据到磁盘
# The maximum amount of time a message can sit in a log before we
force a flush
#log.flush.interval.ms=1000
##### Log Retention Policy
#####
```

```
#数据文件保留多长时间， 存储的最大时间超过这个时间会根据
log.cleanup.policy设置数据清除策略log.retention.bytes和
log.retention.minutes或log.retention.hours任意一个达到要求， 都会执
行删除
# The following configurations control the disposal of log
segments. The policy can
# be set to delete segments after a period of time, or after a
given size has accumulated.
# A segment will be deleted whenever *either* of these criteria
are met. Deletion always happens
# from the end of the log.
# The minimum age of a log file to be eligible for deletion
log.retention.hours=168
#topic每个分区的最大文件大小， 一个topic的大小限制 = 分区数
*log.retention.bytes。 -1没有大小限log.retention.bytes和
log.retention.minutes任意一个达到要求， 都会执行删除， 会被topic创建
时的指定参数覆盖
# A size-based retention policy for logs. Segments are pruned from
the log as long as the remaining
# segments don't drop below log.retention.bytes.
#log.retention.bytes=1073741824
#topic的分区是以一堆segment文件存储的， 这个控制每个segment的大小，
会被topic创建时的指定参数覆盖
# The maximum size of a log segment file. When this size is
reached a new log segment will be created.
log.segment.bytes=1073741824
#文件大小检查的周期时间， 是否处罚 log.cleanup.policy中设置的策略
# The interval at which log segments are checked to see if they
can be deleted according
# to the retention policies
```

```
log.retention.check.interval.ms=300000
##### Zookeeper
#####
#zookeeper集群的地址，可以是多个，多个之间用逗号分割
hostname1:port1,hostname2:port2,hostname3:port3
# Zookeeper connection string (see zookeeper docs for details).
# This is a comma separated host:port pairs, each corresponding to
a zk
# server. e.g. "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002".
# You can also append an optional chroot string to the urls to
specify the
# root directory for all kafka znodes.
zookeeper.connect=localhost:2181
#ZooKeeper的连接超时时间
# Timeout in ms for connecting to zookeeper
zookeeper.connection.timeout.ms=6000
```