

KAFKA version 2.11-2.0.0

RPM PACKAGE: VZkafka-2.11.2-Linux64.rpm

1. This package requires the id kafka with the group kafka to exist prior to installation. All commands should be run as a member of the kafka group or as the kafka id.
2. This package requires one SSI file of the form host.localhost.kafka with the contents (example only):

```
SERVERS=serv1,serv2
```

```
SERVER_JAVA_HOME="/apps/opt/jdk180_181_64"
```

3. This package is distributed with zookeeper and kafka-manager-1.3.3.16.
4. Middleware team has created this package to be used in AWS with automated rehydration by providing Kafka Servers to be defined in the host.localhost.kafka and specific configuration of this server to be defined in the host.localhost.SERVER_NAME where SERVER_NAME is a unique name.

The host.localhost.SERVER_NAME optional ssi file is to be the /apps/opt/kafka/config/server.properties file with the changes specific for this SERVER.

The host.localhost.zookeeper optional ssi file is to be the /apps/opt/kafka/config/zookeeper.properties file which changes specific for this application's zookeeper.

5. Additionally, Middleware has added a menu (/apps/opt/kafka/scripts/menu) which allows easy usage of the kafka software. Here are the menu items:

```
***** Kafka Utilities Menu *****
**                                     **
** A) Start Zookeeper (one per server) **
** B) Stop Zookeeper                  **
** C) List Running Zookeeper process  **
** D) Start Kafka broker(s) (one or many per server) **
** E) Stop all Kafka broker(s)        **
** F) List Running Kafka brokers      **
** G) Run change_config.sh (creates new brokers and sets **
      JAVA_HOME, host.localhost.KAFKaname)
** Q) Quit menu
*****
```

The scripts behind this menu can be found in the /apps/opt/kafka/scripts directory. In here I also provide example consumer and message sending scripts which can be modified as needed for testing. Additionally, middleware has provided the script to list the topics which can also be modified as you see fit for testing.

6. Please contact the middleware group for any enhancement and we will add them so others can benefit from your ideas.
7. Additional information on Kafka and it's usage can be found on the Web.

Example of the SSI file host.localhost.SERVER_NAME:

```
# 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 #####

# The id of the broker. This must be set to a unique integer for each broker.
broker.id=0
##### Socket Server Settings #####
##

# The address the socket server listens on. It will get the value returned from
# java.net.InetAddress.getCanonicalHostName() if not configured.
# FORMAT:
# listeners = listener_name://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 v
# alue
# returned from java.net.InetAddress.getCanonicalHostName().
#advertised.listeners=PLAINTEXT://your.host.name:9092

# Maps listener names to security protocols, the default is for them to be the s
# ame. See the config documentation for more details
#listener.security.protocol.map=PLAINTEXT:PLAINTEXT,SSL:SSL,SASL_PLAINTEXT:SASL_
PLAINTEXT,SASL_SSL:SASL_SSL

# The number of threads that the server uses for receiving requests from the net
# work and sending responses to the network
num.network.threads=3

# The number of threads that the server uses for processing requests, which may
# include disk I/O
```

num.io.threads=8

The send buffer (SO_SNDBUF) used by the socket server
socket.send.buffer.bytes=102400

The receive buffer (SO_RCVBUF) used by the socket server
socket.receive.buffer.bytes=102400

The maximum size of a request that the socket server will accept (protection against OOM)
socket.request.max.bytes=104857600

Log Basics

A comma separated list of directories under which to store log files
log.dirs=/tmp/kafka-logs

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

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

Internal Topic Settings

####

The replication factor for the group metadata internal topics "__consumer_offsets" and "__transaction_state"
For anything other than development testing, a value greater than 1 is recommended to ensure availability such as 3.
offsets.topic.replication.factor=1
transaction.state.log.replication.factor=1
transaction.state.log.min.isr=1

Log Flush Policy

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

Log Retention Policy

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 due to age
log.retention.hours=168

A size-based retention policy for logs. Segments are pruned from the log unless the remaining
segments drop below log.retention.bytes. Functions independently of log.retention.hours.
#log.retention.bytes=1073741824
The maximum size of a log segment file. When this size is reached a new log segment will be created.
log.segment.bytes=1073741824

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 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

Timeout in ms for connecting to zookeeper
zookeeper.connection.timeout.ms=6000

```
# 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
```

```
# Timeout in ms for connecting to zookeeper
zookeeper.connection.timeout.ms=6000
```

```
##### Group Coordinator Settings #####
#####
```

```
# The following configuration specifies the time, in milliseconds, that the GroupCoordinator will delay the initial consumer rebalance.
# The rebalance will be further delayed by the value of group.initial.rebalance.delay.ms as new members join the group, up to a maximum of max.poll.interval.ms.
# The default value for this is 3 seconds.
# We override this to 0 here as it makes for a better out-of-the-box experience for development and testing.
# However, in production environments the default value of 3 seconds is more suitable as this will help to avoid unnecessary, and potentially expensive, rebalances during application startup.
group.initial.rebalance.delay.ms=0
```

Example of the SSI file host=localhost.zookeeper:

```
# 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.
# 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
```