

---

# Python

*Release*

**Aug 31, 2017**



---

# Contents

---

<b>1</b>	<b>Getting Started</b>	<b>3</b>
1.1	Prerequisites . . . . .	3
1.2	Quick Install . . . . .	3
1.3	Source Install . . . . .	3
<b>2</b>	<b>Command List</b>	<b>5</b>
2.1	kafka-assigner . . . . .	5
<b>3</b>	<b>Contribution guide</b>	<b>7</b>
3.1	Bug fixes . . . . .	7
3.2	Code guidelines . . . . .	7
<b>4</b>	<b>API Documentation</b>	<b>9</b>
4.1	kafka.tools . . . . .	9
4.2	kafka.tools.assigner . . . . .	10
4.3	kafka.tools . . . . .	13
<b>5</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>



kafka-tools is a collection of various tools for managing Apache Kafka.



This document describes how to install and configure kafka-tools.

### Prerequisites

Software that must be installed:

- Python (<https://www.python.org/>)
- Kazoo (<https://kazoo.readthedocs.io/en/latest/>)
- Kafka (<https://kafka.apache.org/>)

In addition, you will need to run it on a host that has:

- Access to the Zookeeper ensemble for the Kafka cluster.
- SSH access to the Kafka brokers (with credentials preferably loaded into ssh-agent).

### Quick Install

1. Download the Kafka binaries from <https://kafka.apache.org/downloads.html>.
2. Use pip to install the kafka-tools package from pypi.

### Source Install

1. Download the Kafka binaries from <https://kafka.apache.org/downloads.html>.
2. Clone the kafka-tools repository from <https://github.com/linkedin/kafka-tools>
3. Run the tests using tox.

4. Install kafka-tools using setup.py.



#### **kafka-assigner**

kafka-assigner is used for performing partition reassignments and preferred replica elections. It uses the admin CLI utilities provided with Kafka and layers on additional logic to perform tasks like removing a broker, rebalancing partitions, fixing partition replication factors, and performing preferred replica elections.



We're always open to fixes and new features! Please open a PR for any changes that you have and someone will review and merge it. If you're not up for writing the code, open an issue for any problems or requests.

### Bug fixes

If you find a bug in “kafka-tools” we welcome patches to help fix the problem.

A few things to keep in mind when submitting bug fixes:

- Please include details on reproducing the bug
- Explain how your patch resolves the issue
- Test cases for the fix are extremely helpful
- Would your patch affect existing expected behaviors? For example, catching an exception may fix an issue for your code, but another caller may expect that exception in order to handle the error case differently.

### Code guidelines

Please follow these Python best practices:

- Provide documentation for new or modified APIs
- Provide test cases for new or modified APIs



### **kafka.tools**

#### **kafka.tools.exceptions**

#### **kafka.tools.models.broker**

#### **kafka.tools.models.cluster**

#### **kafka.tools.models.partition**

#### **kafka.tools.models.topic**

#### **kafka.tools.modules**

#### **kafka.tools.utilities**

`kafka.tools.utilities.check_java_home()`

Make sure that JAVA\_HOME in the current environment is specified and is valid.

**Raises** ConfigurationException if JAVA\_HOME is not set or does not contain java

`kafka.tools.utilities.find_path_containing(fname)`

Search the PATH for the given executable filename

**Parameters** `fname` – the filename to check

**Returns** the path that contains the filename

**Raises** ConfigurationException if the filename cannot be found, or if it is not executable

`kafka.tools.utilities.get_tools_path(tools_path=None)`

Find the Kafka admin utilities, either from the provided arg or the PATH.

**Parameters** `tools_path` – the path to use for locating the Kafka admin utilities.

**Returns** the path that contains Kafka admin utilities

**Raises** `ConfigurationException` if the path cannot be determined

`kafka.tools.utilities.is_exec_file` (*fname*)

Check if the given filename is a regular file and is executable.

**Parameters** `fname` – the filename to check.

**Returns** True if the filename given exists and is executable, False otherwise

`kafka.tools.utilities.json_loads` (*json\_str*)

Load the provided string as JSON data. Make sure to try the python2 way and the python3 way

**Parameters** `json_str` – The JSON encoded string

**Returns** The decoded JSON object

`kafka.tools.utilities.synchronized` (*item*)

Decorator that synchronizes access to the instance method it decorates using a preexisting lock in the `_lock` attribute of the instance

## kafka.tools.assigner

### kafka.tools.assigner.actions.balance

### kafka.tools.assigner.actions.balancemodules.count

### kafka.tools.assigner.actions.balancemodules.even

### kafka.tools.assigner.actions.balancemodules.leader

### kafka.tools.assigner.actions.balancemodules.rackaware

`kafka.tools.assigner.actions.balancemodules.rackaware.check_partition_swappable` (*replicas\_a*,  
*replicas\_b*,  
*pos*)

Check if the broker at position `pos` in the first replica list can be swapped with the replica at position `pos` in the second list 1. `replicas_a[pos]` must not be in the `replicas_b` list 2. `replicas_b[pos]` must not be in the `replicas_a` list 3. `replicas_a[pos]` must have a different rack than the replicas in `replicas_b` (except for `replicas_b[pos]`) 4. `replicas_b[pos]` must have a different rack than the replicas in `replicas_a` (except for `replicas_a[pos]`)

**Params** `replicas_a` the first replica list

**Params** `replicas_b` the second replica list

**Params** `pos` the position in the replica list to be replaced

**Returns** True if the broker can be swapped into this replica list, False otherwise

`kafka.tools.assigner.actions.balancemodules.rackaware.difference_in_size_to_last_partition`

Return the difference in size between the specified Partition and the last Partition in the provided list. If the list is empty, return infinity.

**Params partition** a Partition object to use for calculating the difference

**Params partitions** a list of Partition objects

**Returns** The difference in size between partition and the last Partition in the partitions list, or infinity

`kafka.tools.assigner.actions.balancemodules.rackaware.racks_for_replica_list` (*replicas*,  
*pos=None*)

Returns a set of racks for each of the given replicas in the list Skip the replica at position pos, if specified

**Params replicas** a list of Broker objects

**Params pos** a replica position to skip, or None to not skip a replica

**Returns** a list of racks





**kafka.tools.assigner.actions.balancemodules.size**

**kafka.tools.assigner.actions.clone**

**kafka.tools.assigner.actions elect**

**kafka.tools.assigner.actions.remove**

**kafka.tools.assigner.actions.reorder**

**kafka.tools.assigner.actions.setrf**

**kafka.tools.assigner.actions.trim**

**kafka.tools.assigner.arguments**

**kafka.tools.assigner.batcher**

**kafka.tools.assigner.models.reassignment**

**kafka.tools.assigner.models.replica\_election**

**kafka.tools.assigner.sizers.ssh**

## **kafka.tools**

**kafka.tools.protocol.arguments**

**kafka.tools.protocol.errors**

**kafka.tools.protocol.help**

**kafka.tools.protocol.types.boolean**

**kafka.tools.protocol.types.bytes**

**kafka.tools.protocol.types.integers**

**kafka.tools.protocol.types.sequences**

**kafka.tools.protocol.types.string**

**kafka.tools.protocol.requests.api\_versions\_v0**

**kafka.tools.protocol.requests.controlled\_shutdown\_v1**

**kafka.tools.protocol.requests.create\_topics\_v0**

**kafka.tools.protocol.requests.delete\_topics\_v0**

**kafka.tools.protocol.requests.describe\_groups\_v0**

**kafka.tools.protocol.requests.group\_coordinator\_v0**



## CHAPTER 5

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



---

## Python Module Index

---

### k

kafka.tools.assigner.actions.balance, 10  
kafka.tools.assigner.actions.balancemodules.count, 10  
kafka.tools.assigner.actions.balancemodules.even, 10  
kafka.tools.assigner.actions.balancemodules.leader, 10  
kafka.tools.assigner.actions.balancemodules.rackaware, 10  
kafka.tools.assigner.actions.balancemodules.size, 13  
kafka.tools.assigner.actions.clone, 13  
kafka.tools.assigner.actions elect, 13  
kafka.tools.assigner.actions.remove, 13  
kafka.tools.assigner.actions.reorder, 13  
kafka.tools.assigner.actions.setrf, 13  
kafka.tools.assigner.actions.trim, 13  
kafka.tools.assigner.arguments, 13  
kafka.tools.assigner.batcher, 13  
kafka.tools.assigner.models.reassignment, 13  
kafka.tools.assigner.models.replica\_election, 13  
kafka.tools.assigner.sizers.ssh, 13  
kafka.tools.exceptions, 9  
kafka.tools.models.broker, 9  
kafka.tools.models.partition, 9  
kafka.tools.models.topic, 9  
kafka.tools.modules, 9  
kafka.tools.protocol.arguments, 13  
kafka.tools.protocol.errors, 13  
kafka.tools.protocol.help, 13  
kafka.tools.protocol.requests.api\_versions\_v0, 13  
kafka.tools.protocol.requests.controlled\_shutdown\_v1, 13  
kafka.tools.protocol.requests.create\_topics\_v0, 13  
kafka.tools.protocol.requests.delete\_topics\_v0, 13  
kafka.tools.protocol.requests.describe\_groups\_v0, 13  
kafka.tools.protocol.requests.group\_coordinator\_v0, 13  
kafka.tools.protocol.requests.heartbeat\_v0, 13  
kafka.tools.protocol.requests.join\_group\_v0, 13  
kafka.tools.protocol.requests.leader\_and\_isr\_v0, 13  
kafka.tools.protocol.requests.leave\_group\_v0, 13  
kafka.tools.protocol.requests.list\_groups\_v0, 13  
kafka.tools.protocol.requests.list\_offset\_v0, 13  
kafka.tools.protocol.requests.offset\_commit\_v0, 13  
kafka.tools.protocol.requests.offset\_commit\_v1, 13  
kafka.tools.protocol.requests.offset\_commit\_v2, 13  
kafka.tools.protocol.requests.offset\_fetch\_v0, 13  
kafka.tools.protocol.requests.offset\_fetch\_v1, 13  
kafka.tools.protocol.requests.sasl\_handshake\_v0, 13  
kafka.tools.protocol.requests.stop\_replica\_v0, 13  
kafka.tools.protocol.requests.sync\_group\_v0, 13  
kafka.tools.protocol.requests.topic\_metadata\_v0, 13  
kafka.tools.protocol.requests.topic\_metadata\_v1, 13



## C

check\_java\_home() (in module kafka.tools.utilities), 9

check\_partition\_swappable() (in module kafka.tools.assigner.actions.balancemodules.rackaware), 10

## D

difference\_in\_size\_to\_last\_partition() (in module kafka.tools.assigner.actions.balancemodules.rackaware), 10

## F

find\_path\_containing() (in module kafka.tools.utilities), 9

## G

get\_tools\_path() (in module kafka.tools.utilities), 9

## I

is\_exec\_file() (in module kafka.tools.utilities), 10

## J

json\_loads() (in module kafka.tools.utilities), 10

## K

kafka.tools.assigner.actions.balance (module), 10

kafka.tools.assigner.actions.balancemodules.count (module), 10

kafka.tools.assigner.actions.balancemodules.even (module), 10

kafka.tools.assigner.actions.balancemodules.leader (module), 10

kafka.tools.assigner.actions.balancemodules.rackaware (module), 10

kafka.tools.assigner.actions.balancemodules.size (module), 13

kafka.tools.assigner.actions.clone (module), 13

kafka.tools.assigner.actions.elect (module), 13

kafka.tools.assigner.actions.remove (module), 13

kafka.tools.assigner.actions.reorder (module), 13

kafka.tools.assigner.actions.setrf (module), 13

kafka.tools.assigner.actions.trim (module), 13

kafka.tools.assigner.arguments (module), 13

kafka.tools.assigner.batcher (module), 13

kafka.tools.assigner.models.reassignment (module), 13

kafka.tools.assigner.models.replica\_election (module), 13

kafka.tools.assigner.sizers.ssh (module), 13

kafka.tools.exceptions (module), 9

kafka.tools.models.broker (module), 9

kafka.tools.models.partition (module), 9

kafka.tools.models.topic (module), 9

kafka.tools.modules (module), 9

kafka.tools.protocol.arguments (module), 13

kafka.tools.protocol.errors (module), 13

kafka.tools.protocol.help (module), 13

kafka.tools.protocol.requests.api\_versions\_v0 (module), 13

kafka.tools.protocol.requests.controlled\_shutdown\_v1 (module), 13

kafka.tools.protocol.requests.create\_topics\_v0 (module), 13

kafka.tools.protocol.requests.delete\_topics\_v0 (module), 13

kafka.tools.protocol.requests.describe\_groups\_v0 (module), 13

kafka.tools.protocol.requests.group\_coordinator\_v0 (module), 13

kafka.tools.protocol.requests.heartbeat\_v0 (module), 13

kafka.tools.protocol.requests.join\_group\_v0 (module), 13

kafka.tools.protocol.requests.leader\_and\_isr\_v0 (module), 13

kafka.tools.protocol.requests.leave\_group\_v0 (module), 13

kafka.tools.protocol.requests.list\_groups\_v0 (module), 13

kafka.tools.protocol.requests.list\_offset\_v0 (module), 13

kafka.tools.protocol.requests.offset\_commit\_v0 (module), 13

kafka.tools.protocol.requests.offset\_commit\_v1 (module), 13

kafka.tools.protocol.requests.offset\_commit\_v2 (module), 13  
kafka.tools.protocol.requests.offset\_fetch\_v0 (module), 13  
kafka.tools.protocol.requests.offset\_fetch\_v1 (module), 13  
kafka.tools.protocol.requests.sasl\_handshake\_v0 (module), 13  
kafka.tools.protocol.requests.stop\_replica\_v0 (module), 13  
kafka.tools.protocol.requests.sync\_group\_v0 (module), 13  
kafka.tools.protocol.requests.topic\_metadata\_v0 (module), 13  
kafka.tools.protocol.requests.topic\_metadata\_v1 (module), 13  
kafka.tools.protocol.requests.update\_metadata\_v0 (module), 13  
kafka.tools.protocol.responses.api\_versions\_v0 (module), 13  
kafka.tools.protocol.responses.controlled\_shutdown\_v1 (module), 13  
kafka.tools.protocol.responses.create\_topics\_v0 (module), 13  
kafka.tools.protocol.responses.delete\_topics\_v0 (module), 13  
kafka.tools.protocol.responses.describe\_groups\_v0 (module), 13  
kafka.tools.protocol.responses.group\_coordinator\_v0 (module), 13  
kafka.tools.protocol.responses.heartbeat\_v0 (module), 13  
kafka.tools.protocol.responses.join\_group\_v0 (module), 13  
kafka.tools.protocol.responses.leader\_and\_isr\_v0 (module), 13  
kafka.tools.protocol.responses.leave\_group\_v0 (module), 13  
kafka.tools.protocol.responses.list\_groups\_v0 (module), 13  
kafka.tools.protocol.responses.list\_offset\_v0 (module), 13  
kafka.tools.protocol.responses.metadata\_v0 (module), 13  
kafka.tools.protocol.responses.metadata\_v1 (module), 13  
kafka.tools.protocol.responses.offset\_commit\_v0 (module), 13  
kafka.tools.protocol.responses.offset\_commit\_v1 (module), 13  
kafka.tools.protocol.responses.offset\_commit\_v2 (module), 13  
kafka.tools.protocol.responses.offset\_fetch\_v0 (module), 13  
kafka.tools.protocol.responses.offset\_fetch\_v1 (module), 13  
kafka.tools.protocol.responses.sasl\_handshake\_v0 (module), 13  
kafka.tools.protocol.responses.stop\_replica\_v0 (module), 13  
kafka.tools.protocol.responses.sync\_group\_v0 (module), 13  
kafka.tools.protocol.responses.update\_metadata\_v0 (module), 13  
kafka.tools.protocol.types.boolean (module), 13  
kafka.tools.protocol.types.bytes (module), 13  
kafka.tools.protocol.types.integers (module), 13  
kafka.tools.protocol.types.sequences (module), 13  
kafka.tools.protocol.types.string (module), 13  
kafka.tools.utilities (module), 9

**R**  
racks\_for\_replica\_list() (in module kafka.tools.assigner.actions.balancemodules.rackaware), 11

**S**  
synchronized() (in module kafka.tools.utilities), 10