

# Kafka Topics CLI Tutorial

How to create, delete, describe, or change a Kafka topic?

Kafka Topics CLI, i.e., `kafka-topics` is used to create, delete, describe, or change a topic in Kafka. Make sure you have [started Kafka](#) beforehand.

## CLI Extensions

Use CLI commands with appropriate extensions for your platform, e.g., `kafka-topics.bat` for windows, `kafka-topics.sh` for Linux and Mac

## How to create a Kafka topic?

To create a Kafka topic, we need to provide the mandatory parameters:

If Kafka v2.2+, use the Kafka hostname and port e.g., `localhost:9092`

If older version of Kafka, use the Zookeeper URL and port e.g. `localhost:2181`

Provide the mandatory parameters: topic name, number of partitions and replication factor.

Use the `kafka-topics.sh` CLI with the `--create` option

### Important recommendation: if you use Kafka v2.2+

If you have a recent version of Kafka, we strongly recommend you to use the command with the `--bootstrap-server` option because the Zookeeper option is now deprecated and is removed as part of Kafka v3.

## Example

Creating the Kafka topic `first_topic` with 3 partitions and a replication factor of 1 when my Kafka broker is running at `localhost:9092`

### Kafka v2.2+:

`.sh`

```
localhost:9092 --topic first_topic --create --partitions 3 --replication-factor 1
```

### Kafka v2.1 or less:

`.sh`

```
localhost:2181 --topic first_topic --create --partitions 3 --replication-factor 1
```

## Command Output

```
1  WARNING: Due to limitations in metric names, topics with a period ('.') or unde
2  Created topic first_topic.
```

The WARNING is saying that you shouldn't mix underscores and periods in topic name

## Gotchas

Here are the common mistakes and caveats with the `kafka-topics.sh --create` command:

**You cannot specify a replication factor greater than the number of brokers you have**

You can specify as many partitions as you want, 3 is a good number to get started with in dev

If you see the documentation after running a command, this means your command is wrong. Scroll up to see details about the error message

There are no default for partitions and replication factor and you must specify those explicitly

The topic name must contain only ASCII alphanumeric, '.', '\_' and '-'

## Extra Important options you can set (advanced)

`--config`

You can set [topic-level configurations](#) for example `--config max.message.bytes=64000`

`--disable-rack-aware`

Disable rack aware replica assignment (not recommended, set only if you know what you're doing)

## How to list Kafka Topics?

To list Kafka topics, we need to provide the mandatory parameters:

If Kafka v2.2+, use the Kafka hostname and port e.g., `localhost:9092`

If older version of Kafka, use the Zookeeper URL and port e.g. `localhost:2181`

Use the `kafka-topics.sh` CLI with the `--list` option

### Important recommendation: if you use Kafka v2.2+

If you have a recent version of Kafka, we strongly recommend you to use the command with the `--bootstrap-server` option because the Zookeeper option is now deprecated and is removed as part of Kafka v3.

## Example

Listing topics when my Kafka broker is running at `localhost:9092`

## Kafka v2.2+:

```
.sh
```

```
1 kafka-topics.sh --bootstrap-server localhost:9092 --list
```

## Kafka v2.1 or less:

```
.sh
```

```
1 kafka-topics.sh --zookeeper localhost:2181 --list
```

## Command Output

```
1 __consumer_offsets
2 first_topic
3 second_topic
4 test
```

## Gotchas

Here are the common mistakes and caveats with the `kafka-topics.sh --list` command:

This command lists internal topics that you won't have created (such as `__consumer_offsets`). Do not try to delete these topics. Use `--exclude-internal` if you want to hide these topics

If you see the documentation after running a command, this means your command is wrong. Scroll up to see details about the error message

## Extra Important options you can set (advanced)

`--exclude-internal`

Exclude internal topics when running list or describe command (they're listed by default)

### To filter topics based on replication status:

`--at-min-isr-partitions`

If set when describing topics, only show partitions whose isr count is equal to the configured minimum.

`--unavailable-partitions`

Only show partitions whose leader is not available.

`--under-min-isr-partitions`

Only show partitions whose isr count is less than the configured minimum.

`--under-replicated-partitions`

Only show under replicated partitions

## How to describe a Kafka topic?

To describe a Kafka topic and get partition details, we need to provide the mandatory parameters

If Kafka v2.2+, use the Kafka hostname and port e.g., `localhost:9092`

If older version of Kafka, use the Zookeeper URL and port e.g. `localhost:2181`

Use the `kafka-topics.sh` CLI with the `--describe` option

### Important recommendation: if you use Kafka v2.2+

If you have a recent version of Kafka, we strongly recommend you to use the command with the `--bootstrap-server` option because the Zookeeper option is now deprecated and is removed as part of Kafka v3.

## Example

Listing topics when my Kafka broker is running at `localhost:9092`

### Kafka v2.2+:

```
.sh
```

```
1 kafka-topics.sh --bootstrap-server localhost:9092 --describe --topic first_top:
```

### Kafka v2.1 or less:

```
.sh
```

```
1 kafka-topics.sh --zookeeper localhost:2181 --describe --topic first_topic
```

You can specify a comma delimited list of topics to describe more than one at a time.

## Command Output

```
1 Topic: first_topic      TopicId: fN9mf1UDSmiCdKIDFyMEIQ PartitionCount: 3
2 Topic: first_topic      Partition: 0    Leader: 1      Replicas: 1    Isr: 1
3 Topic: first_topic      Partition: 1    Leader: 1      Replicas: 1    Isr: 1
4 Topic: first_topic      Partition: 2    Leader: 1      Replicas: 1    Isr: 1
```

Leader: 1 means that for partition 0, the broker with the ID 1 is the leader.

Replicas: 1 means that for partition 0, the broker with the ID 1 is a replica.

Isr: 1 means that for partition 0, the broker with the ID 1 is an in-sync replica.

## In case the command is running against a Kafka cluster, you may see

```

1 Topic: third_topic      TopicId: 84yqCErzTG27J4wv44dkPQ PartitionCount: 4
2 Topic: third_topic      Partition: 0      Leader: 2      Replicas: 2,3,1 Isr: 2
3 Topic: third_topic      Partition: 1      Leader: 3      Replicas: 3,1,2 Isr: 3
4 Topic: third_topic      Partition: 2      Leader: 1      Replicas: 1,2,3 Isr: 1
5 Topic: third_topic      Partition: 3      Leader: 2      Replicas: 2,1,3 Isr: 2

```

Leader: 2 means that for partition 0, the broker with the ID 2 is the leader.

Replicas: 2,3,1 means that for partition 0, the brokers with the ID 2, 3 and 1 are replicas.

Isr: 2,3,1 means that for partition 0, the brokers with the ID 2, 3 and 1 are replicas in-sync replica.

## Gotchas

Here are the common mistakes and caveats with the `kafka-topics.sh --describe` command:

It's easy to get partition ID and broker ID confused. Look at the Command Output section above to make sense of what the numbers mean in the command output

If you see the documentation after running a command, this means your command is wrong. Scroll up to see details about the error message

You can specify a comma delimited list to describe a list of topics

If you don't specify a `--topic` option, it will describe all topics

## Extra Important options you can set (advanced)

### To describe all topics (usually combined with the options `describe after`)

Do not set the `--topic` argument and you will describe all Kafka topics

## To filter partitions based on replication status:

`--at-min-isr-partitions`

If set when describing topics, only show partitions whose ISR count is equal to the configured minimum.

`--unavailable-partitions`

Only show partitions whose leader is not available

`--under-min-isr-partitions`

Only show partitions whose ISR count is less than the configured minimum.

`--under-replicated-partitions`

Only show under replicated partitions

## How to increase the number of partitions of a Kafka topic?

To increase the number of partitions in a Kafka topic, we need to provide the mandatory parameters.

If Kafka v2.2+, use the Kafka hostname and port e.g., `localhost:9092`

If older version of Kafka, use the Zookeeper URL and port e.g. `localhost:2181`

Provide the mandatory parameters: topic name and number of partitions

Use the `kafka-topics.sh` CLI with the `--alter` option

### **Danger: Increasing Partitions**

Increasing the number of partitions in a Kafka topic a DANGEROUS OPERATION if your applications are relying on key-based ordering. In that case, create a new topic and copy all data there instead to have keys properly re-distributed.

## Example



Altering the Kafka topic `first_topic` to have 5 partitions when my Kafka broker is running at `localhost:9092`

### Kafka v2.2+:

```
.sh
```

```
1 kafka-topics.sh --bootstrap-server localhost:9092 --alter --topic first_topic
```

### Kafka v2.1 or less:

```
.sh
```

```
1 kafka-topics.sh --zookeeper localhost:2181 --alter --topic first_topic --parti
```

## Command Output

```
1 (no output)
```

The command does not have any output, although you can verify afterwards with a `--describe` command.

```
.sh
```

```
1 $ kafka-topics.sh --bootstrap-server localhost:9092 --describe --topic first_t  
2 Topic: first_topic      TopicId: fN9mf1UDSmiCdKIDFyMEIQ PartitionCount: 5  
3 Topic: first_topic      Partition: 0      Leader: 1      Replicas: 1      Isr: 1  
4 Topic: first_topic      Partition: 1      Leader: 1      Replicas: 1      Isr: 1  
5 Topic: first_topic      Partition: 2      Leader: 1      Replicas: 1      Isr: 1
```

```

6 Topic: first_topic      Partition: 3      Leader: 1      Replicas: 1      Isr: 1
7 Topic: first_topic      Partition: 4      Leader: 2      Replicas: 2      Isr: 2

```

## Gotchas

Here are the common mistakes and caveats with the `kafka-topics.sh --alter` command:

**Th** based ordering as changing the number of partitions changes the key-hashing technique

You can only add partitions, not remove partitions

## How to Delete a Kafka Topic?

To delete a Kafka topic, we need to provide the mandatory parameters.

If Kafka v2.2+, use the Kafka hostname and port e.g., `localhost:9092`

If older version of Kafka, use the Zookeeper URL and port e.g. `localhost:2181`

Use the `kafka-topics.sh` CLI with the `--delete` option

Ensure that the Kafka brokers allow for topic deletion `delete.topic.enable=true` (default)

### Important recommendation: if you use Kafka v2.2+

If you have a recent version of Kafka, we strongly recommend you to use the command with the `--bootstrap-server` option because the Zookeeper option is now deprecated and is removed as part of Kafka v3.

## Example

Deleting the topic `first_topic` when my Kafka broker is running at `localhost:9092`

## Kafka v2.2+:

```
.sh
```

```
1 kafka-topics.sh --bootstrap-server localhost:9092 --delete --topic first_topic
```

## Kafka v2.1 or less:

```
.sh
```

```
1 kafka-topics.sh --zookeeper localhost:2181 --delete --topic first_topic
```

### Multiple Topics

You can specify a comma delimited list of topics to delete more than one topic at a time

## Command Output

```
1 (no output)
```

You can verify the outcome of the command with a `kafka-topics.sh --describe` command.

It may take some time (depending on the topic size) to see the effect of the command.

## Gotchas

Here are the common mistakes and caveats with the `kafka-topics.sh --delete` command:

In case topic deletion is not enabled (see the `delete.topic.enable` broker setting) then the topics will be "marked for deletion" but will not be deleted



conduktor

Kafka Options Explorer

Pricing

Contact

Try for free



You can specify a comma delimited list to delete a list of topics

### Windows Users (if not running on WSL2): do NOT delete topics

Windows has a long-standing bug (KAFKA-1194) that makes Kafka crash if you delete topics.

The only way to recover from this error is to manually delete the topic folders in

`data/kafka`.

### If you are running Kafka on WSL2, you can safely delete topics

Was this content helpful? 



6



0

[PREVIOUS](#)[NEXT](#)

Share this page.



Apache, Apache Kafka, Kafka and the Kafka logo are trademarks of the **Apache Software Foundation**. All other trademarks, servicemarks, and copyrights are the property of their respective owners.

© 2023 Conductor Inc. All Rights Reserved.

[Privacy Policy](#)[Terms of Service](#)[Cookie Policy](#)[DPA](#)[EULA](#)[EULA Enterprise](#)

Status: All Systems Operational