kcat Kafka on the shell

CONCEPTS

> The mode flag picks the job: -P produce, -C consume, -G group consume, -L metadata, -Q offset query. Without one, kcat guesses from stdin and stdout.

stdin one message per line
↓ `-P -t -p`
Topic partition ordered log
↓ `-C -o -e`
stdout payload or -f / -J
↓ `pipe to jq`
Your tools jq, grep, files

> Put -b, -X and credentials in a config file (-F or ~/.config/kcat.conf) so commands stay short.

METADATA

`kcat -b <broker> -L`
List brokers, topics and partitions

`kcat -b <broker> -L -t <topic>`
Show one topic only

CONSUME

`kcat -b <broker> -t <topic> -C`
Read all partitions and wait

`kcat -b <broker> -t <topic> -C -e`
Exit at end of partition

`kcat -b <broker> -t <topic> -C -p <n>`
Read one partition

`kcat -b <broker> -t <topic> -C -o beginning`
Start from the oldest message

`kcat -b <broker> -t <topic> -C -o -<n>`
Last n messages per partition

`kcat -b <broker> -t <topic> -C -c <n>`
Stop after n messages

OUTPUT FORMAT

`kcat -b <broker> -t <topic> -C -J`
One JSON envelope per message

`kcat -b <broker> -t <topic> -C -K:`
Print key:value

`kcat -b <broker> -t <topic> -C -f '<fmt>'`
Custom output format

`kcat -b <broker> -t <topic> -C -f '%t[%p]@%o %k=%s\n'`
Topic, partition, offset, key, value

Format tokens (-f)

Token	Prints
`%s`	Payload
`%S`	Payload length
`%k`	Key
`%K`	Key length
`%t`	Topic
`%p`	Partition
`%o`	Offset
`%T`	Timestamp (ms)
`%h`	Headers
`%R`	Payload length, then payload (binary)

PRODUCE

`kcat -b <broker> -t <topic> -P`
Send each stdin line as a message

`kcat -b <broker> -t <topic> -P -l <file>`
Send a file, one message per line

`kcat -b <broker> -t <topic> -P -K:`
Split key:value on the colon

CONSUMER GROUPS

`kcat -b <broker> -G <group> <topic>`
Balanced consume with a group

CONFIG AND SECURITY

`kcat -F <file>`
Read settings from a config file

`kcat -b <broker> -X <prop>=<value>`
Set any librdkafka property