

KIP-979: Allow independently stop KRaft processes

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Test Plan](#)
- [Rejected Alternatives](#)

Status

Current state: *Accepted*

Discussion thread: <https://lists.apache.org/thread/mwcljxdnyobthsszy4n2qr2tqcf9cxcf>

JIRA:

 Unable to render Jira issues macro, execution error.

Motivation

Some users run KRaft controllers and brokers on the same machine (not containerized, but through tarballs, etc). Prior to KRaft, when running ZooKeeper and Kafka on the same machine, users could independently stop the ZooKeeper node and Kafka broker since there were specific shell scripts for each (zookeeper-server-stop and kafka-server-stop, respectively).

However in KRaft mode, they can't stop the KRaft controllers independently from the Kafka brokers or any independent process because there is just a single script that doesn't distinguish between processes and signals all of them. We need to provide a way for users to kill either controllers or brokers, and even better, any individual process.

Public Interfaces

The command line for stopping Kafka nodes will include a pair of optional and mutually exclusive parameters "[--process-role]" OR "[--node-id]" to support identifying a set of processes or a specific process to stop based on the contents of the node's configuration file.

Instead of simply running

```
./bin/kafka-server-stop.sh
```

The new script will accept an optional parameter [--process-role=value], with the value indicating whether to kill Kafka broker processes or controller processes.

OR

The new script will accept an optional parameter [--node-id=value], with the value indicating the specific process, identified by its node id, that the user wishes to kill.

Example 1: the new command to kill all the broker processes will look like:

```
./bin/kafka-server-stop.sh --process-role=broker
```

Example 2: the command to kill the process with node ID = 1 will look like:

```
./bin/kafka-server-stop.sh --node-id=1
```

If both parameters are provided, the value for node-id parameter will take precedence, i.e, the process with node id specified will be killed, no matter what's the process role provided. Also, a log message will be shown to indicate that precedence.

When neither a "process-role" nor a "node-id" field is provided, the behavior remains unchanged -- all Kafka processes are stopped.

Proposed Changes

I'm changing the "kafka-server-stop.sh" file to accept an optional field, either "process-role" or "node-id". When the user specifies a process role or a node id, I will retrieve the absolute path to the configuration file, and search for the value of that field. If the value retrieved from the configuration file matches the input, that process will be killed, otherwise it will be skipped.

Compatibility, Deprecation, and Migration Plan

Existing users won't need to change any behavior if they want to continue killing both controller and broker processes. Otherwise, if they want to kill only the broker/controller, or an individual broker/controller, they'll need to specify a "process-role" or a "node-id" field in the script.

Test Plan

The change can be tested through command line.

Rejected Alternatives

One rejected alternative is using an optional argument "[--required-config <name = value>]" to allow users to stop Kafka processes with any arbitrary name = value based on the node's configuration file. This argument gives the users freedom to choose any field in the configuration file to indicate the process they want to kill. However, upon discussion, we think this approach based on configs is a bit too open-ended and not very user friendly. Therefore, we decided to simply provide flags for the things a user may care about the most, i.e, two specific and mutually exclusive parameters `process-role` and `node-id`. We agree that it is more user-friendly than the `--required-config` argument, and it comes at the possible expense of generality.