KIP-697: Stricter parsing of addresses in configs

Status

Current state: *Under Discussion*

Discussion thread: *here [Change the link from the KIP proposal email archive to your own email thread]*

JIRA: KAFKA-10713

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

This KIP concerns the following configs, but we'll talk about `bootstrap.servers` for concreteness.

- StreamsConfig.APPLICATION_SERVER_CONFIG
- RaftConfig.QUORUM_VOTERS_CONFIG
- ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG
- WorkerConfig.BOOTSTRAP_SERVERS_CONFIG
- AdminClientConfig.BOOTSTRAP_SERVERS_CONFIG
- ProducerConfig.BOOTSTRAP_SERVERS_CONFIG

The client `bootstrap.servers` config is documented as being a list of `host:port` pairs (comma separated, as with other configs of `LIST` type). However, that differs from how this config is actually validated, which allows a protocol part to be included. The protocol part is never used, only the host and port, as obtained using the non-public `Utils.getHost` and `Utils.getPort` methods. This can be confusing in a number of ways:

- Because a protocol is accepted users might mistakenly think that it is meaningful.
- As reported in KAFKA-10713, if a user mistakenly separates the list items with the wrong character (e.g. a semicolon) then the `bootstrap.servers` will be parsed incorrectly (because the protocol part of the validation regex is `.*?` which is too lax), silently resulting in only the last server in the list being used. The user is unaware of their mistake until they application is not able to bootstrap when that server is not available.
- The `Utils.getHost` and `Utils.getPort` have, over time, come to be used to validate other configs as well: `application.server` and `quorum.voters`, compounding the scope for future problems.

We should tighten up what is accepted by `Utils.getHost` and `Utils.getPort` so that a protocol is not accepted.

Public Interfaces

No changes, since the implementation of validation for `bootstrap.servers` will be aligned with the existing documentation.

Proposed Changes

Remove the `.*?` from the pattern in `Utils.HOST_PORT_PATTERN`.

Compatibility, Deprecation, and Migration Plan

Any configs listed in the motivation section which currently include a protocol will become invalid. Users will have to remove the protocol from their configs if they have included one. Since this is a breaking change it will be implemented in Kafka 3.0 to minimise surprises.

Rejected Alternatives

-