KIP-5 - Broker Configuration Management

Brokers should be able to pull from a meta data store to get their configurations.

Status

Current state: *Superseded by KIP-21*

Discussion thread: *here*

JIRA: *here*

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

Motivation

It is dangerous to have the possibility of different configuration on different brokers. Even if folks use orchestration system it could be disastrous for a Kafka cluster if that system didn't work/update right. It takes an unnecessary amount of effort making builds for every target platform to help keep changes consistent throughout the cluster(s). Instead we can provide a way for Kafka brokers to handle their own configurations.

Public Interfaces

- *Command line tools and arguments*

Proposed Changes

You can keep using properties as you do today. If you decide to use the global configuration feature then only a few properties will be able to be overridden e.g. host and port. All other properties will either be default or set in the central meta store for retrieving the settings.

These changes use and are based on KAFKA-1845 (KafkaConfig should use ConfigDef).

Applying Global Configuration to the Broker

The new proposed workflow is the following:

1. The broker is being started (entry point - Kafka.scala class)
2. The local configuration file server.properties is parsed
3. Global configuration settings (if present) are picked up from zookeeper
4. Global configuration take precedence over local except for predefined list (currently - *brokerId, host, port, advertisedHost, advertisedPort*)
5. KafkaServer is started with "merged" configuration

The global configuration is stored in the same format as a topic-level configuration:

```
{
    "version" : "1",
    "config"  : {
                    "message.max.bytes" : "200000",
                    "num.network.threads" : "5"
                }
}
```

Dedicated zookeeper path to store global configuration: `/brokers/config`

Updating Global Configuration

To change global configuration the new Wire Protocol message is introduced. Proposed schema:

Request:

```
ConfigChangeRequest => [ConfigData]
     ConfigData => ConfigKey ConfigValue
          ConfigKey => string
          ConfigValue => string
```

Response:

```
ConfigChangeResponse => ErrorCode
     ErrorCode => int16
```

To change global configuration:

1. User issues ChangeConfigRequest to Controller with global configuration that needs to be updated
2. Controller fetches current global configuration from zk, merges with supplied and validates merged config (this part relies on KAFKA-1845)
3. If successful, configuration is updated in zk
4. New global configuration is picked up by broker once restarted

Compatibility, Deprecation, and Migration Plan

In 0.8.3 we will still support the property files but it will be deprecated and we will then remove support for property files in 0.9.0.

Rejected Alternatives

If there are alternative ways of accomplishing the same thing, what were they? The purpose of this section is to motivate why the design is the way it is and not some other way.