KIP-998: Give ProducerConfig(props, doLog) constructor protected access

Status

Current state: Voting

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

JIRA: **KAFKA-15781** - Getting issue details... STATUS

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

Motivation

In applications that construct the various client configs multiple times, the config logging can be pretty extreme, so it's nice to be able to disable this for subsequent config objects that are created. We don't expose the constructors that accept a `doLog` parameter in the public API, but all the other clients at least make this constructor `protected` so it's possible to extend the class and suppress the excessive logging. See the ConsumerConfig for example

The ProducerConfig, however, is the one case where this constructor is package-private. It would be nice to align it with the other client config classes and make this `protected` to allow turning off logging

Public Interfaces

The following ProducerConfig constructor will be changed from package-private to protected:

ProducerConfig

```
package org.apache.kafka.clients.producer;

public class ProducerConfig {

    // add the `protected` modifier
    protected ProducerConfig(Map<?, ?> props, boolean doLog);
}
```

Proposed Changes

As described above, we are just changing the access modifier of this constructor from package-private to protected. See https://github.com/apache/kafka/pull/14681

Compatibility, Deprecation, and Migration Plan

N/A

Test Plan

N/A

Rejected Alternatives

1. there was some debate around whether or not this change required a KIP based on the established definition of a public API. Ultimately it seems that we have not stated any project-specific conventions that would apply in this case and therefore we default to the basic java accessibility rules. In this case, since the class itself is non-final and we are adding a protected API that could be leveraged by a custom class that extends ProducerConfig, this is considered a public API. Separately, we determined that a KIP is necessary for this case due to the change itself being one that affects the javadocs of a public API. See this thread for the full discussion
2. We could go a step further and make this, as well as the other config constructors, a fully public method instead of just protected. This was rejected on the basis that these classes are really not intended to be used or instantiated outside of the corresponding client, even if we don't outright prevent this, so we don't want to open the classes up any more than is necessary. This KIP is really just about bringing the ProducerConfig in alignment with the other client config classes, and allowing those who do already instantiate their own instances of these config classes, for whatever reason, to be able to disable the logging if they so choose.