Item 61
git_comments:

1. * topic level metrics *
2. **** Client level ****
3. We can't create the MetricName up front for these, because we don't know the topic name yet.
4. **** Topic level ****

git_commits:

1. **summary:** KAFKA-5597; Improve Metrics docs generation
   message: KAFKA-5597; Improve Metrics docs generation Instead of having the metrics registry and the org.apache.kafka.common.metrics.Metrics object be separate things, have the metrics registry hold a copy of the Metrics object. That way, all the metricInstance stuff is hidden, and we don't have to make sure that the metrics registry and the Metrics object are configured identicailly (with the same tags). I personally think this looks a little better. Author: James Cheng <jylcheng@yahoo.com> Reviewers: Ismael Juma <ismael@juma.me.uk> Closes #3799 from wushujames/producer_sender_metrics_docs_different

github_issues:

github_issues_comments:

github_pulls:

1. **title:** KAFKA-5597 Alternate way to do Metrics docs generation
   body: I was about to start on the next round of autogeneration of metrics docs, but I wanted to @guozhangwang 's opinion on this first. This is a possible alternate way to do autogeneration of metrics docs, that possibly looks a little nicer for the developer. Posting this to get some feedback on if the original way looks better, or if this new way looks better. Instead of having the metrics registry and the org.apache.kafka.common.metrics.Metrics object be separate things, have the metrics registry hold a copy of the Metrics object. That way, all the metricInstance stuff is hidden, and we don't have to make sure that the metrics registry and the Metrics object are configured identicailly (with the same tags). I personally think this looks a little better.

github_pulls_comments:

1. @guozhangwang Can I get your thoughts on this? I'd like to get the next round of metrics docs into 1.0.0, and I'd like some feedback on whether I should change my technique or not.
2. I like the idea of consolidating the logic in a single place / class. cc @ijuma to take another look.
3. Also cc @rajinisivaram who's done a bunch of metrics work recently
4. I merged in the latest trunk. I also changed the MetricNameTemplate references to private, which means metrics can only be obtained using accessors. The effect of this is that users of the MetricsRegistry objects can *only* obtained specific named metrics, which themselves were derived from the templates. Which kind of achieves my main goal, which is that all metrics have to be predefined and therefore will have their documentation autogenerated. This works so well, in fact, that in Sender.java, you can see that it doesn't even have to import org.apache.kafka.common.metrics.Metrics anymore.
5. @guozhangwang: I think this PR is done, and I'd like to submit it for merge in order to get it into 1.0.0. Do I submit a new PR without "[WIP]"? Or do I just edit the title of the PR? @ijuma @rajinisivaram: In this PR, I also improved the MetricsRegistry class so that you don't need to keep the getAllTemplates() array in sync with the class variables anymore. Btw, after I get this PR in, I have an equivalent PR that refactors the ConsumerMetrics to follow the same pattern. So, heads up for that. Should I file a JIRA for that? Thanks.
6. You can edit the title of this PR. Sure, feel free to file a JIRA for the `ConsumerMetrics` side.
7. Test failure is not due to this PR. The failure is on trunk. ``` org.apache.kafka.common.protocol.ErrorsTest > testUniqueErrorCodes FAILED java.lang.AssertionError: Error codes must be unique expected:<61> but was:<62> at org.junit.Assert.fail(Assert.java:88) at org.junit.Assert.failNotEquals(Assert.java:834) at org.junit.Assert.assertEquals(Assert.java:645) at org.apache.kafka.common.protocol.ErrorsTest.testUniqueErrorCodes(ErrorsTest.java:38) ``` It was

introduced in https://github.com/apache/kafka/commit/5f6393f9b17cce17ded7a00e439599dfa77deb2d#diff-b119227df7efa3ffeb7fe69e49ff1afeR541
8. Hopefully trunk will be fixed soon. When it is, I will update this PR. Can I get this merged to trunk, as well as to 1.0.0? I want to try to get https://issues.apache.org/jira/browse/KAFKA-5951 in, and it will layer on top of this PR.
9. As @ijuma replied on the mailing list, I think it should be fine to get in 1.0.0 if it is in mergable state in a week.
10. @guozhangwang @ijuma: This is ready for review.
11. @guozhangwang @ijuma Can you review? I'd like to get this and https://issues.apache.org/jira/browse/KAFKA-5951 into 1.0.0. My PR for https://issues.apache.org/jira/browse/KAFKA-5951 is ready but it layers on top of the changes in this PR, and so I can't submit it yet. I'd like to make sure that I have enough time to get it reviewed so it can merge in to 1.0.0. Thanks!
12. @ijuma: Thanks for the feedback. I updated the PR with all your suggestions.

github_pulls_reviews:

1. Seems like most (all?) fields should be `final`.
2. We generally avoid `getters` in Kafka.
3. In this case, the getters aren't standard "getters" which just return the internal variable. These getters are actually instantiating the MetricName object. For many MetricNames, this isn't *entire* necessary since they are the same each time. But, for MetricNames that have tags (like per-topic MetricNames), it's necessary to instantiate them at get-time, because the tag may vary per metric.
4. @ijuma: Thanks for the feedback. Since you made me realize the getters aren't necessary all of the time, and you also pointed out that almost all fields should be final, I came up with a way to get rid of the getters, which also gets rid of a lot of the boilerplate code. I'll upload that shortly, so you can see how it looks.
5. Look at how clean and concise it is now. Direct variable access, and no need to call any methods, no need to use the Metrics object, and nothing is exposed about the metricsInstance or MetricNameTemplates.
6. The ones above are accessing a variable, whereas this is calling a method. This is needed because I need to pass in the tags.
7. Notice that this method is named the same as the instance variable that it is using. Is that okay?
8. Both should be `final`.
9. Maybe inline this and other similar cases?
10. We can probably have one helper method for the `METRIC_GROUP_NAME` case and another for the `TOPIC_GROUP_NAME` case. Since these methods are called so many times, it's worth a little convenience. Also, the lines are too long at the moment. They should fit in the GitHub review window.
11. Rename to `allTemplates()`?
12. Nit: should this be simply `tags` since we call them `tags` in createMetricName. Using a different name can make it sound like there's a difference.
13. Worth adding a comment explaining why we can't create the `metricName` for these (i.e. we don't know the topic name).

jira_issues:

1. **summary:** Autogenerate Producer sender metrics
   description:

jira_issues_comments:

1. GitHub user wushujames opened a pull request: https://github.com/apache/kafka/pull/3535 KAFKA-5597: Autogenerate producer sender metrics. Subtask of https://issues.apache.org/jira/browse/KAFKA-3480 The changes are very similar to what was done for the consumer in https://issues.apache.org/jira/browse/KAFKA-5191 (pull request https://github.com/apache/kafka/pull/2993) A screenshot of the docs are here: ![producer metrics docs](https://user-images.githubusercontent.com/677529/28245950-96b0c20a-69c6-11e7-8631-11fd55a0ad92.png) You can merge this pull request into a Git repository by running: $ git pull https://github.com/wushujames/kafka producer_sender_metrics_docs Alternatively you can review and

apply these changes as the patch at: https://github.com/apache/kafka/pull/3535.patch To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #3535 ---- commit 33f876fa471048ec6ddd7c05dbd5935d0157eca3 Author: James Cheng <jylcheng@yahoo.com> Date: 2017-07-16T08:29:30Z Autogenerate producer sender metrics. ----

2. Issue resolved by pull request 3535 [https://github.com/apache/kafka/pull/3535]
3. Github user asfgit closed the pull request at: https://github.com/apache/kafka/pull/3535
4. GitHub user wushujames opened a pull request: https://github.com/apache/kafka/pull/3799 KAFKA-5597 [WIP] Alternate way to do Metrics docs generation I was about to start on the next round of autogeneration of metrics docs, but I wanted to @guozhangwang 's opinion on this first. This is a possible alternate way to do autogeneration of metrics docs, that possibly looks a little nicer for the developer. Posting this to get some feedback on if the original way looks better, or if this new way looks better. Instead of having the metrics registry and the org.apache.kafka.common.metrics.Metrics object be separate things, have the metrics registry hold a copy of the Metrics object. That way, all the metricInstance stuff is hidden, and we don't have to make sure that the metrics registry and the Metrics object are configured identicailly (with the same tags). I personally think this looks a little better. You can merge this pull request into a Git repository by running: $ git pull https://github.com/wushujames/kafka producer_sender_metrics_docs_different Alternatively you can review and apply these changes as the patch at: https://github.com/apache/kafka/pull/3799.patch To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #3799 ---- commit 7b48ab7030c58879103bccf8c3f7dff59444a6a6 Author: James Cheng <jylcheng@yahoo.com> Date: 2017-09-06T07:23:23Z Instead of having the metrics registry and the org.apache.kafka.common.metrics.Metrics object be separate things, have the metrics registry hold a copy of the Metrics object. That way, all the metricInstance stuff is hidden, and we don't have to make sure they are configured identicailly (with the same tags). ----
5. Github user asfgit closed the pull request at: https://github.com/apache/kafka/pull/3799