KIP-264: Add a consumer metric to record raw fetch size

Status

Current state: *Under Discussion*

Discussion thread: *here*

Vote thread: *here*

JIRA:	⚠️ Unable to render Jira issues macro, execution error.

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

Motivation

The existing consumer's `bytes-fetched` metric uses the size of decompressed message. It would be useful to have another metric `raw-bytes-fetched` to track the size of uncompressed (raw) fetched message.

Note that, as Jason Gustafson mentioned on the existing PR, there is already a `bytes-received` metric available in `Selector`, but that metric also takes into account the overhead of the Kafka protocol. Having both metrics in place would allow for better examining the overhead.

Public Interfaces

No public interface will be introduced or changed.

Proposed Changes

A new consumer metric sensor called `raw-bytes-fetched` will be introduced to help with tracking the average, max, consumption rate, and consumption total of raw message sizes received, similar to how these metrics are in place for size of decompressed message (via `bytes-fetched` metric sensor). The newly added metrics include:

- `raw-fetch-size-avg`: The average number of raw bytes fetched per request
- `raw-fetch-size-max`: The maximum number of raw bytes fetched per request
- `raw-bytes-consumed-rate`: The average number of raw bytes consumed per second
- `raw-bytes-consumed-total`: The total number of raw bytes consumed

Sample code in FetcherMetricsRegistry.java that defines the new metrics

```java
public FetcherMetricsRegistry(Set<String> tags, String metricGrpPrefix) {
    /***** Client level *****/
    String groupName = metricGrpPrefix + "-fetch-manager-metrics";

    this.rawFetchSizeAvg = new MetricNameTemplate("raw-fetch-size-avg", groupName,
            "The average number of raw bytes fetched per request", tags);
    this.rawFetchSizeMax = new MetricNameTemplate("raw-fetch-size-max", groupName,
            "The maximum number of raw bytes fetched per request", tags);
    this.rawBytesConsumedRate = new MetricNameTemplate("raw-bytes-consumed-rate", groupName,
            "The average number of raw bytes consumed per second", tags);
    this.rawBytesConsumedTotal = new MetricNameTemplate("raw-bytes-consumed-total", groupName,
            "The total number of raw bytes consumed", tags);

    ...
}
```

Compatibility, Deprecation, and Migration Plan

There is no backward compatibility issue to address for this change. The existing metrics will remain in place.

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.