

Ambari Metrics - Whitelisting

In large clusters (500+ nodes), sometimes there are performance issues seen in AMS aggregations. In the ambari-metrics-collector log file, we can see log lines that look like

```
20:51:30,952 INFO 2080712366@qtp-974606690-381 AsyncProcess:1597 - #1, waiting for 13948 actions to finish
20:51:31,601 INFO 1279097595@qtp-974606690-359 AsyncProcess:1597 - #1, waiting for 19376 actions to finish
```

In Ambari 3.0.0, we are tackling these performance issues through a complete schema and aggregation logic revamp. Until then, we can use AMS whitelisting to reduce the number of metrics tracked by AMS, thereby solving this scale problem.

How do we enable whitelisting in AMS.

Until Ambari 2.4.3

A metric whitelist file can be used to track the set of metrics in AMS. All other metrics will be discarded.

STEPS

- Metric whitelist file is present in `/etc/ambari-metrics-collector/conf`. If not present in older Ambari versions, it can be downloaded from https://github.com/apache/ambari/blob/trunk/ambari-metrics/ambari-metrics-timelinecollector/conf/unix/metrics_whitelist to the collector host.
- Adding config `ams-site : timeline.metrics.whitelist.file = <path_to_whitelist_file>`
- Restart AMS collector
- Verify whitelisting config was used. In `ambari-metrics-collector` log file, verify the line `'Whitelisting # metrics'`.

From Ambari 2.5.0 onwards

From Ambari 2.5.0, more refinements for whitelisting were included.

- **App Blacklisting** - Blacklist metrics from one or more services. Other service metrics will be entirely allowed or controlled through a whitelist file.

```
ams-site : timeline.metrics.apps.blacklist = hbase,namenode
```

- **App Whitelisting** - Whitelist metrics from one or more services.

```
ams-site:timeline.metrics.apps.whitelist = nimbus,datanode
```

NOTE : The App name can be found from the metadata URL - `http:<metrics_collector_host>:6188/ws/v1/timeline/metrics/metadata`

- **Metric Whitelisting** - Same as the whitelisting method in Ambari 2.4.3 (through a whitelist file).

In addition to supplying metric names in the whitelist file, patterns can also be supplied using the `._p_` prefix. For example, a pattern can be specified as follows

```
._p_dfs.FSNamesystem.*
```

```
._p_jvm.JvmMetrics*
```

An example of a metric whitelisting file that has both metrics and patterns - https://github.com/apache/ambari/blob/trunk/ambari-metrics/ambari-metrics-timelinecollector/src/test/resources/test_data/metric_whitelist.dat.

These whitelisting/blacklisting techniques can be used together.

- If you just have `timeline.metrics.whitelist.file = <some_file>`, only metrics in that file will be allowed (irrespective of whatever apps might be sending them).
- If you just have `timeline.metrics.apps.blacklist = datanode`, all datanode metrics will be disallowed. Metrics from all other services will be allowed.
- If you just have `timeline.metrics.apps.whitelist = namenode`, it is not useful since there is no blacklisting at all.
- If you have metric whitelisting enabled (through a file), and have `timeline.metrics.apps.blacklist = datanode`, all datanode metrics will be disallowed. The whitelisted metrics from other services will be allowed.
- If you have `timeline.metrics.apps.blacklist = datanode`, `timeline.metrics.apps.whitelist = namenode` and metric whitelisting enabled (through a file), datanode metrics will be blacklisted, all namenode metrics will be allowed, and whitelisted metrics from other services will be allowed.

Known Issues

1. The Kafka Topics Grafana dashboard is unable to discover the Kafka Topics by default if whitelisting is enabled. The reason is that 'kafka.log.Log.*' metrics are used to filter the available Kafka Topics but those metrics are not enabled by default in the whitelist file. The issue can be

remediated by adding '._p_kafka.log.Log.*' to the whitelist file and restarting the Metrics Collector.

 Unable to render Jira issues macro, execution error.

2. The following metrics are erroneously filtered by the whitelisting despite they are defined in the whitelist file. Due to this issue the Kafka Topics Grafana dashboard is not showing data.

```
kafka.server.BrokerTopicMetrics.BytesInPerSec.topic.*.count
kafka.server.BrokerTopicMetrics.BytesOutPerSec.topic.*.count
kafka.server.BrokerTopicMetrics.MessagesInPerSec.topic.*.count
kafka.server.BrokerTopicMetrics.TotalProduceRequestsPerSec.topic.*.count
```

The issue can be worked around by adding the '._p_' prefix to the corresponding metrics in the whitelist file, e.g. `._p_kafka.server.BrokerTopicMetrics.BytesInPerSec.topic.*.count`.

 Unable to render Jira issues macro, execution error.