SmartCat

DATA ⟶ KNOWLEDGE ⟶ POWER

Challenges of Monitoring Distributed Systems

May 2017

Nenad Bozic
@NenadBozicNs
nenad.bozic@smartcat.io

SmartCat
www.smartcat.io
@SmartCat_io

SmartCat

SmartCat

SmartCat

Agenda

- Monitoring 101
- Metric data stream and tools
- Log data stream and tools
- Combine metrics and logs for full control
- Alerting

Monitoring 101

- Monitoring domain consists of:

  Metrics data stream

  Log data stream

  Alerting

SmartCat

Metrics Data Stream

Metric data stream

- Metrics are indicators that everything is working within expected boundaries
- Easily forgotten and pushed aside when chasing deadlines
- Good dashboard has enough information (not too much, not too little)

Distributed system -> many graphs to watch -> information overload trap

SmartCat

Metric data stream - decision

- SaaS solutions vs self-managed solutions
- Paying solutions vs free solutions
- Decision based on:

  technical team skillset

  level of control

  security of data

SmartCat

Metric data stream - stack

- Riemann as sink that handles events and sends them to Riemann server
- InfluxDB as NoSQL store which is build for measurements
- Grafana as visualization tool (flexible configurable graphs from many data sources)

Cluster Cassandra diagnostics

Zoom Out | Last 1 hour | Refresh every 5s

Cassandra host: All

Cluster health

Last 15 minutes

Nodes up: 12 / 12

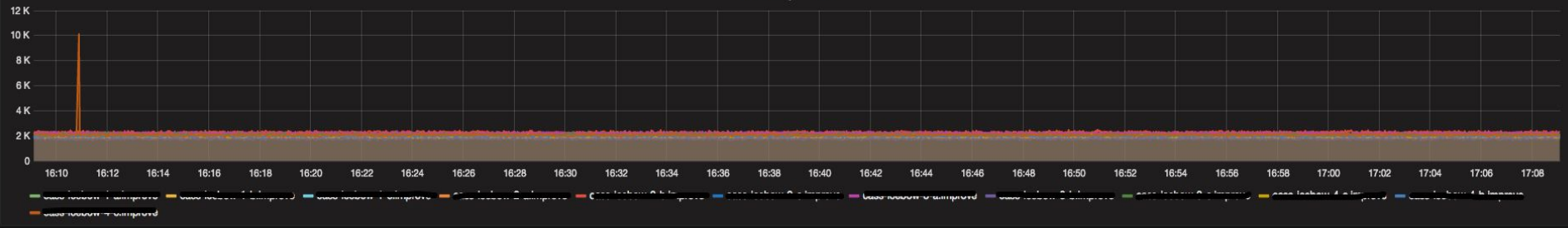
Request rate UPDATE

Last 3 seconds

Request rate SELECT

Last 3 seconds

Write request rate



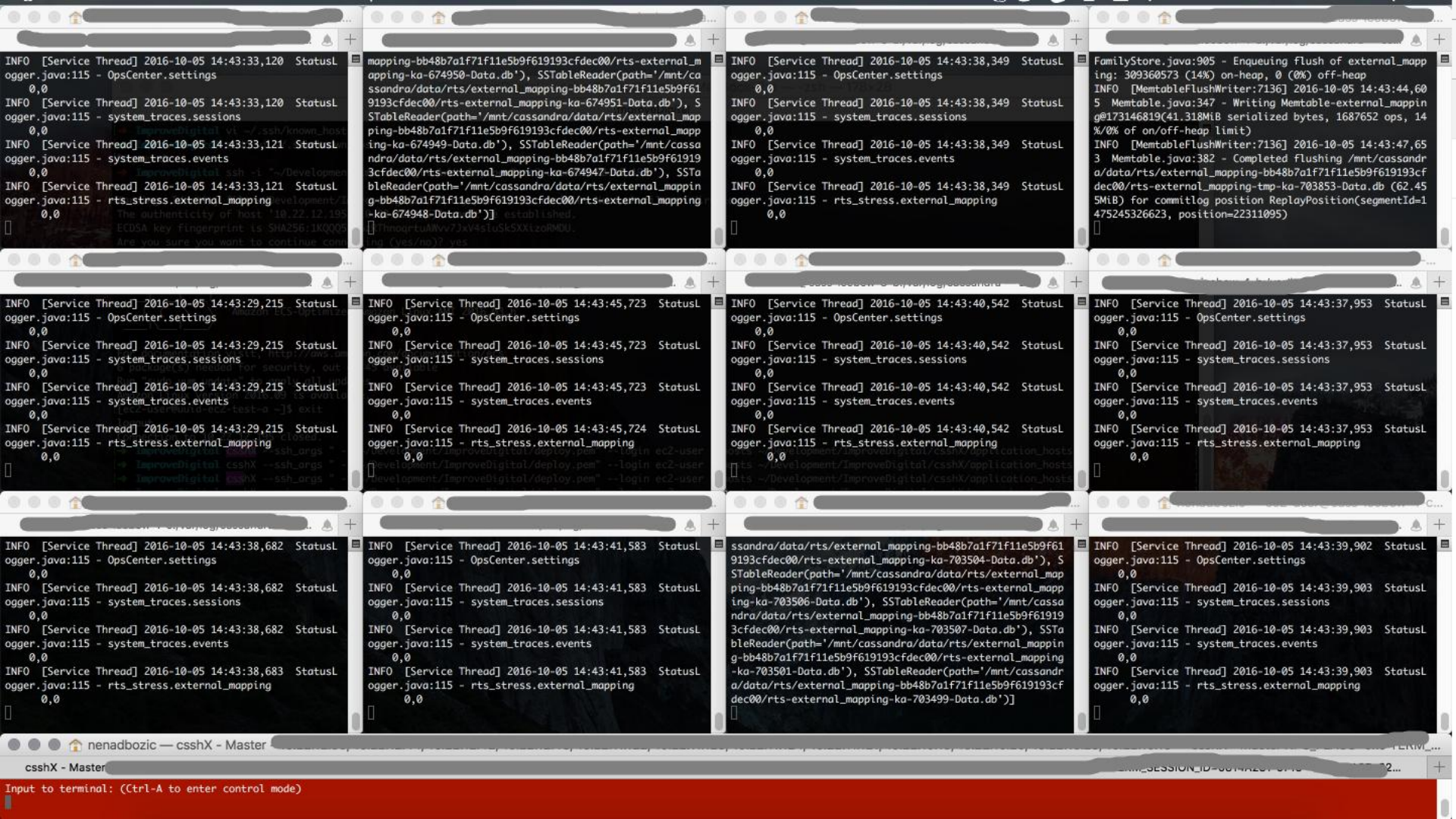
Read request rate

Log Data Stream

Log data stream

- Metrics are indicator that something happened and logs provide context (what happened)
- Log monitoring on single machine requires skill and knowledge
- Same challenges as with metrics (not too much, not too little)

Distributed system -> many terminals open -> information overload trap


INFO [Service Thread] 2016-10-05 14:43:33,120 StatusLogger.java:115 - OpsCenter.settings 0,0
INFO [Service Thread] 2016-10-05 14:43:33,120 StatusLogger.java:115 - system_traces.sessions 0,0
INFO [Service Thread] 2016-10-05 14:43:33,121 StatusLogger.java:115 - system_traces.events 0,0
INFO [Service Thread] 2016-10-05 14:43:33,121 StatusLogger.java:115 - rts_stress.external_mapping 0,0
mapping-bb48b7a1f71f11e5b9f619193cfdec00/rts-external_mapping-ka-674950-Data.db'), SSTableReader(path='/mnt/cassandra/data/rts/external_mapping-bb48b7a1f71f11e5b9f619193cfdec00/rts-external_mapping-ka-674951-Data.db'), SSTableReader(path='/mnt/cassandra/data/rts/external_mapping-bb48b7a1f71f11e5b9f619193cfdec00/rts-external_mapping-ka-674949-Data.db'), SSTableReader(path='/mnt/cassandra/data/rts/external_mapping-bb48b7a1f71f11e5b9f619193cfdec00/rts-external_mapping-ka-674947-Data.db'), SSTableReader(path='/mnt/cassandra/data/rts/external_mapping-bb48b7a1f71f11e5b9f619193cfdec00/rts-external_mapping-ka-674948-Data.db')]
INFO [Service Thread] 2016-10-05 14:43:38,349 StatusLogger.java:115 - OpsCenter.settings 0,0
INFO [Service Thread] 2016-10-05 14:43:38,349 StatusLogger.java:115 - system_traces.sessions 0,0
INFO [Service Thread] 2016-10-05 14:43:38,349 StatusLogger.java:115 - system_traces.events 0,0
INFO [Service Thread] 2016-10-05 14:43:38,349 StatusLogger.java:115 - rts_stress.external_mapping 0,0
FamilyStore.java:905 - Enqueuing flush of external_mapping: 309360573 (14%) on-heap, 0 (0%) off-heap
INFO [MemtableFlushWriter:7136] 2016-10-05 14:43:44,605 Memtable.java:347 - Writing Memtable-external_mapping@173146819(41.318MiB serialized bytes, 1687652 ops, 14%/0% of on/off-heap limit)
INFO [MemtableFlushWriter:7136] 2016-10-05 14:43:47,653 Memtable.java:382 - Completed flushing /mnt/cassandra/data/rts/external_mapping-bb48b7a1f71f11e5b9f619193cfdec00/rts-external_mapping-tmp-ka-703853-Data.db (62.455MiB) for commitlog position ReplayPosition(segmentId=1475245326623, position=22311095)
INFO [Service Thread] 2016-10-05 14:43:29,215 StatusLogger.java:115 - OpsCenter.settings 0,0
INFO [Service Thread] 2016-10-05 14:43:29,215 StatusLogger.java:115 - system_traces.sessions 0,0
INFO [Service Thread] 2016-10-05 14:43:29,215 StatusLogger.java:115 - system_traces.events 0,0
INFO [Service Thread] 2016-10-05 14:43:29,215 StatusLogger.java:115 - rts_stress.external_mapping 0,0
INFO [Service Thread] 2016-10-05 14:43:45,723 StatusLogger.java:115 - OpsCenter.settings 0,0
INFO [Service Thread] 2016-10-05 14:43:45,723 StatusLogger.java:115 - system_traces.sessions 0,0
INFO [Service Thread] 2016-10-05 14:43:45,723 StatusLogger.java:115 - system_traces.events 0,0
INFO [Service Thread] 2016-10-05 14:43:45,724 StatusLogger.java:115 - rts_stress.external_mapping 0,0
INFO [Service Thread] 2016-10-05 14:43:40,542 StatusLogger.java:115 - OpsCenter.settings 0,0
INFO [Service Thread] 2016-10-05 14:43:40,542 StatusLogger.java:115 - system_traces.sessions 0,0
INFO [Service Thread] 2016-10-05 14:43:40,542 StatusLogger.java:115 - system_traces.events 0,0
INFO [Service Thread] 2016-10-05 14:43:40,542 StatusLogger.java:115 - rts_stress.external_mapping 0,0
INFO [Service Thread] 2016-10-05 14:43:37,953 StatusLogger.java:115 - OpsCenter.settings 0,0
INFO [Service Thread] 2016-10-05 14:43:37,953 StatusLogger.java:115 - system_traces.sessions 0,0
INFO [Service Thread] 2016-10-05 14:43:37,953 StatusLogger.java:115 - system_traces.events 0,0
INFO [Service Thread] 2016-10-05 14:43:37,953 StatusLogger.java:115 - rts_stress.external_mapping 0,0
INFO [Service Thread] 2016-10-05 14:43:38,682 StatusLogger.java:115 - OpsCenter.settings 0,0
INFO [Service Thread] 2016-10-05 14:43:38,682 StatusLogger.java:115 - system_traces.sessions 0,0
INFO [Service Thread] 2016-10-05 14:43:38,682 StatusLogger.java:115 - system_traces.events 0,0
INFO [Service Thread] 2016-10-05 14:43:38,683 StatusLogger.java:115 - rts_stress.external_mapping 0,0
INFO [Service Thread] 2016-10-05 14:43:41,583 StatusLogger.java:115 - OpsCenter.settings 0,0
INFO [Service Thread] 2016-10-05 14:43:41,583 StatusLogger.java:115 - system_traces.sessions 0,0
INFO [Service Thread] 2016-10-05 14:43:41,583 StatusLogger.java:115 - system_traces.events 0,0
INFO [Service Thread] 2016-10-05 14:43:41,583 StatusLogger.java:115 - rts_stress.external_mapping 0,0
ssandra/data/rts/external_mapping-bb48b7a1f71f11e5b9f619193cfdec00/rts-external_mapping-ka-703504-Data.db'), SSTableReader(path='/mnt/cassandra/data/rts/external_mapping-bb48b7a1f71f11e5b9f619193cfdec00/rts-external_mapping-ka-703506-Data.db'), SSTableReader(path='/mnt/cassandra/data/rts/external_mapping-bb48b7a1f71f11e5b9f619193cfdec00/rts-external_mapping-ka-703507-Data.db'), SSTableReader(path='/mnt/cassandra/data/rts/external_mapping-bb48b7a1f71f11e5b9f619193cfdec00/rts-external_mapping-ka-703501-Data.db'), SSTableReader(path='/mnt/cassandra/data/rts/external_mapping-bb48b7a1f71f11e5b9f619193cfdec00/rts-external_mapping-ka-703499-Data.db')]
INFO [Service Thread] 2016-10-05 14:43:39,902 StatusLogger.java:115 - OpsCenter.settings 0,0
INFO [Service Thread] 2016-10-05 14:43:39,903 StatusLogger.java:115 - system_traces.sessions 0,0
INFO [Service Thread] 2016-10-05 14:43:39,903 StatusLogger.java:115 - system_traces.events 0,0
INFO [Service Thread] 2016-10-05 14:43:39,903 StatusLogger.java:115 - rts_stress.external_mapping 0,0
nenadbozic — csshX - Master
csshX - Master
Input to terminal: (Ctrl-A to enter control mode)

SmartCat

Log data stream - decision

- SaaS solutions vs self-managed solutions
- Paying solutions and free solutions
- Decision based on:

  technical team skillset

  level of control

  security of your data

Log data stream - ELK stack

- ELK - ElasticSearch, LogStash, Kibana
- Filebeat is sending log messages from instances
- Logstash can filter, manipulate and transform messages
- ElasticSearch indexes log messages for easier searching
- Kibana is visualization tool with filtering capabilities

kibana Discover Visualize Dashboard Settings

Time range control → Last 15 minutes

* ← Search field to query elasticsearch

filebeat-*

Selected Fields

? _source

Available Fields

@timestamp

t _id

t _index

_score

t _type

t beat.hostname

t beat.name

count

? fields

t input_type

t message

offset

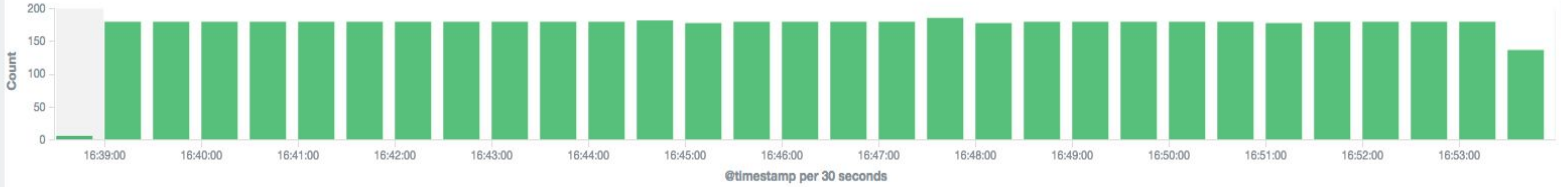
t source

t type

Filtering messages based on indexed fields

5,365 hits

October 5th 2016, 16:38:59.174 - October 5th 2016, 16:53:59.174 — by 30 seconds



Messages from all hosts

Time	_source
October 5th 2016, 16:53:53.106	**@timestamp:** October 5th 2016, 16:53:53.106 **beat.hostname:** ip-172-31-23-27 **beat.name:** ip-172-31-23-27 **count:** 1 **fields:** - **input_type:** log **message:** INFO [request-rate-module] 2016-10-05 14:53:52,163 LogReporter.java:35 - Measurement REQUESTRATE_UPDATE [time=1475679232163, value=0.0, tags={}, fields={}] **offset:** 1,944,813 **source:** /var/log/cassandra/system.log **type:** log **_id:** AVeVViUYO5gEzeSC86Ld **_type:** log **_index:** filebeat-2016.10.05 **_score:**
October 5th 2016, 16:53:53.029	**@timestamp:** October 5th 2016, 16:53:53.029 **beat.hostname:** ip-172-31-23-25 **beat.name:** ip-172-31-23-25 **count:** 1 **fields:** - **input_type:** log **message:** INFO [request-rate-module] 2016-10-05 14:53:53,011 LogReporter.java:35 - Measurement REQUESTRATE_UPDATE [time=1475679233011, value=0.0, tags={}, fields={}] **offset:** 1,949,781 **source:** /var/log/cassandra/system.log **type:** log **_id:** AVeVViWSO5gEzeSC86Lt **_type:** log **_index:** filebeat-2016.10.05 **_score:**
October 5th 2016, 16:53:52.105	**@timestamp:** October 5th 2016, 16:53:52.105 **beat.hostname:** ip-172-31-23-27 **beat.name:** ip-172-31-23-27 **count:** 1 **fields:** - **input_type:** log **message:** INFO [request-rate-module] 2016-10-05 14:53:51,163 LogReporter.java:35 - Measurement REQUESTRATE_UPDATE [time=1475679231163, value=0.0, tags={}, fields={}] **offset:** 1,944,499 **source:** /var/log/cassandra/system.log **type:** log **_id:** AVeVViUYO5gEzeSC86Lb **_type:** log **_index:** filebeat-2016.10.05 **_score:**
October 5th 2016, 16:53:52.105	**@timestamp:** October 5th 2016, 16:53:52.105 **beat.hostname:** ip-172-31-23-27 **beat.name:** ip-172-31-23-27 **count:** 1 **fields:** - **input_type:** log **message:** INFO [request-rate-module] 2016-10-05 14:53:51,164 LogReporter.java:35 - Measurement REQUESTRATE_SELECT [time=1475679231163, value=0.0, tags={}, fields={}] **offset:** 1,944,656 **source:** /var/log/cassandra/system.log **type:** log **_id:** AVeVViUYO5gEzeSC86Lc **_type:** log **_index:** filebeat-2016.10.05 **_score:**
October 5th 2016, 16:53:52.028	**@timestamp:** October 5th 2016, 16:53:52.028 **beat.hostname:** ip-172-31-23-25 **beat.name:** ip-172-31-23-25 **count:** 1 **fields:** - **input_type:** log **message:** INFO [request-rate-module] 2016-10-05 14:53:52,012 LogReporter.java:35 - Measurement REQUESTRATE_SELECT [time=1475679232011, value=0.0, tags={}, fields={}] **offset:** 1,949,624 **source:** /var/log/cassandra/system.log **type:** log **_id:** AVeVViWSO5gEzeSC86Ls **_type:** log **_index:** filebeat-2016.10.05 **_score:**
October 5th 2016, 16:53:52.028	**@timestamp:** October 5th 2016, 16:53:52.028 **beat.hostname:** ip-172-31-23-25 **beat.name:** ip-172-31-23-25 **count:** 1 **fields:** - **input_type:** log **message:** INFO [request-rate-module] 2016-10-05 14:53:52,011 LogReporter.java:35 - Measurement REQUESTRATE_UPDATE [time=1475679232011, value=0.0, tags={}, fields={}] **offset:** 1,949,467 **source:** /var/log/cassandra/system.log **type:** log **_id:** AVeVViWSO5gEzeSC86Lr **_type:** log **_index:** filebeat-2016.10.05 **_score:**

kibana

Discover | Visualize | Dashboard | Settings

Last 7 days

This visualization is linked to a saved search: **Application Errors (staging)**

filebeat-*

Data | Options

metrics

Y-Axis — Count

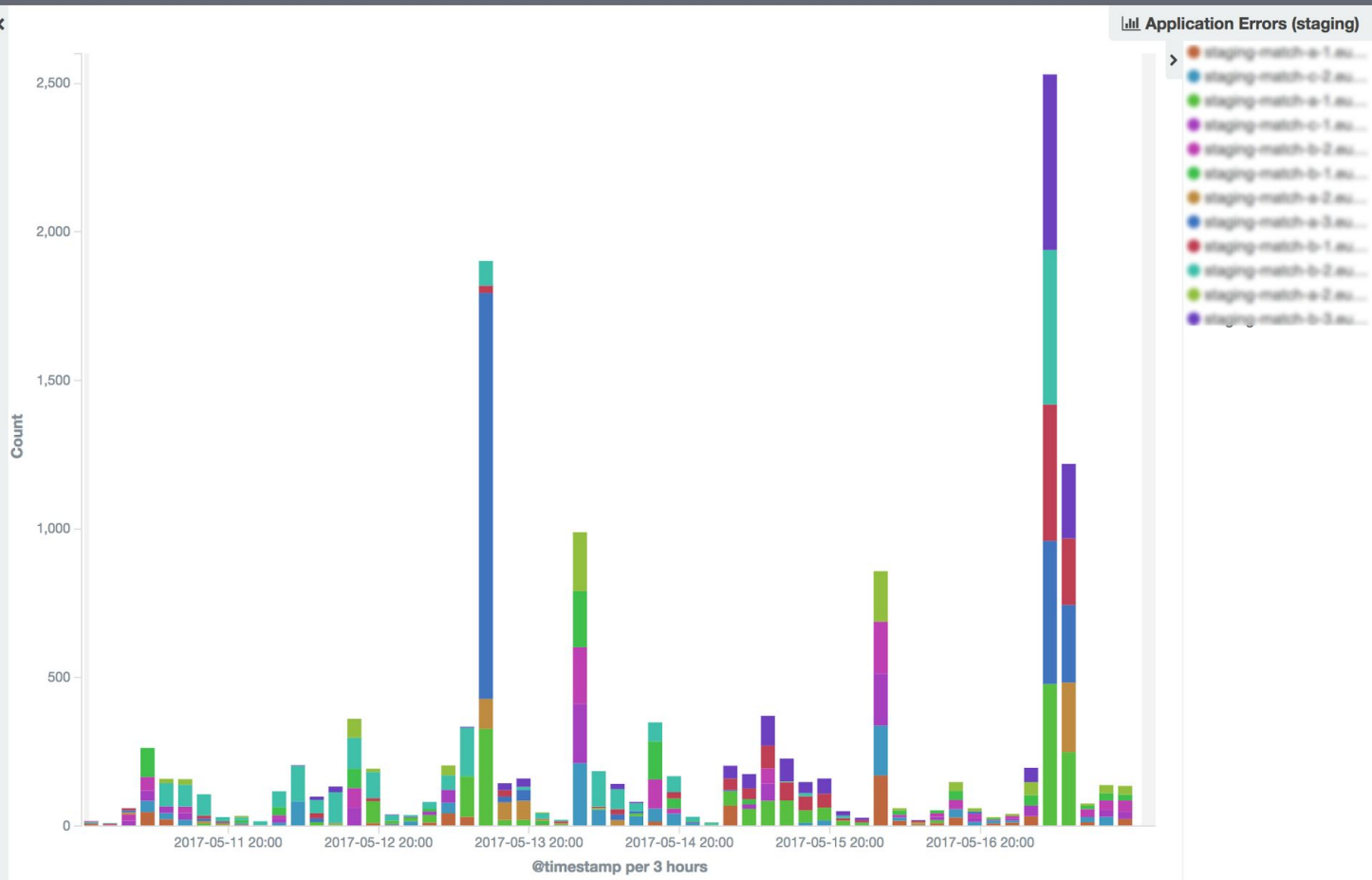
+ Add metrics

buckets

X-Axis — @timestamp per 3 hours

Split Bars — beat.name: Descending

Add sub-buckets



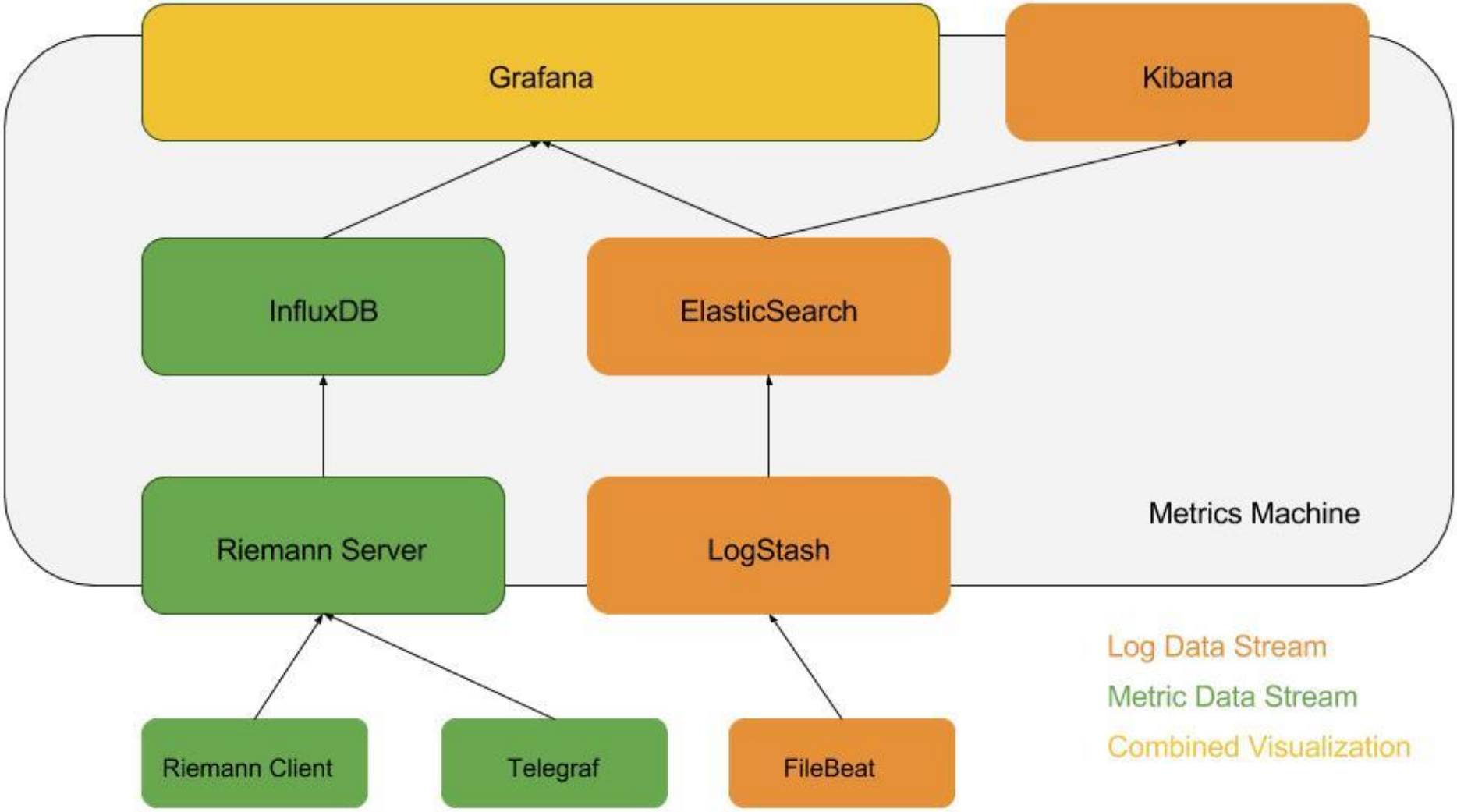
Combine logs and metrics

Real world example

- Provide reliable latency guarantee for 99.999% request
- Whole infrastructure deployed on AWS
- Lot of metrics transferred to metrics machine
- We needed fine grained diagnostics for queries to database both on cluster and application level among other things

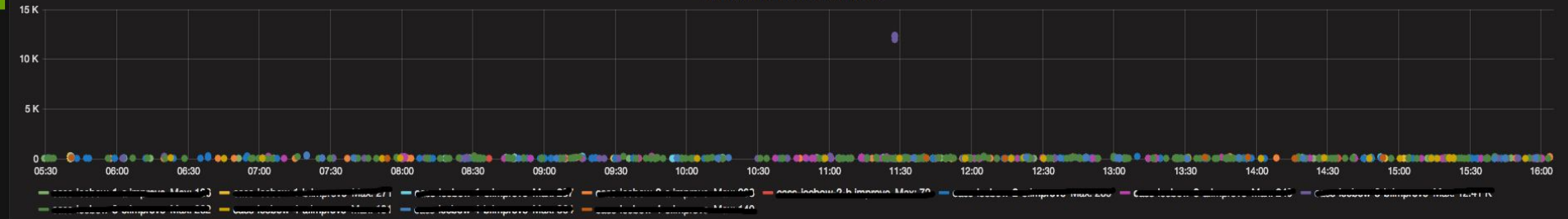
Combine logs and metrics

- It is much easier to look at graphs than logs
- Good metric coverage can pinpoint exact cause of problems
- Usually we need log messages to bring the context
- Grafana can combine InfluxDB (measurement data store) and ElasticSearch (log index)

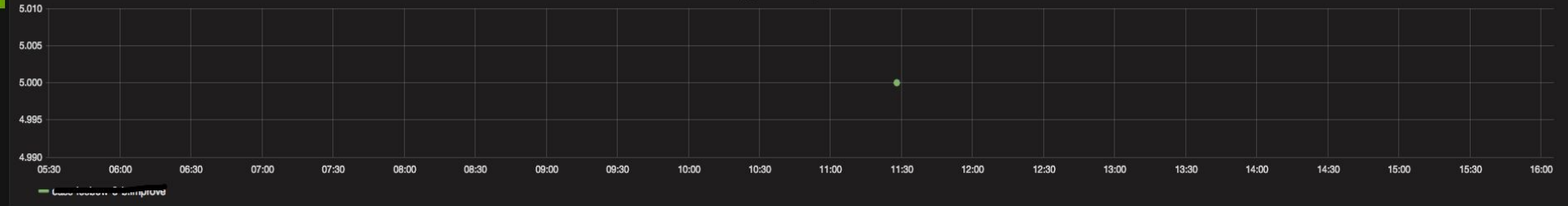

Grafana
Kibana
InfluxDB
ElasticSearch
Riemann Server
LogStash
Metrics Machine
Riemann Client
Telegraf
FileBeat
Log Data Stream
Metric Data Stream
Combined Visualization

appname: All cassandrahost: All

Slow queries (external_mapping)



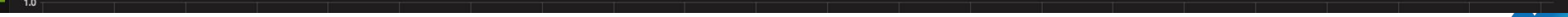
Dropped message count



Dropped messages

beat.hostname	message
[redacted]	INFO [ScheduledTasks:1] 2016-10-04 09:28:00,091 MessagingService.java:929 - REQUEST_RESPONSE messages were dropped in last 5000 ms: 8 for internal timeout and 0 for cross node timeout
[redacted]	INFO [ScheduledTasks:1] 2016-10-04 09:28:00,091 MessagingService.java:929 - MUTATION messages were dropped in last 5000 ms: 27347 for internal timeout and 0 for cross node timeout
[redacted]	WARN [SharedPool-Worker-129] 2016-10-04 09:27:58,581 AbstractEventProcessor.java:76 - Event queue overflown. Until relaxed, further events will be dropped.
[redacted]	WARN [SharedPool-Worker-54] 2016-10-04 09:27:58,546 AbstractEventProcessor.java:76 - Event queue overflown. Until relaxed, further events will be dropped.
[redacted]	INFO [ScheduledTasks:1] 2016-10-04 09:28:00,091 MessagingService.java:929 - READ messages were dropped in last 5000 ms: 1374 for internal timeout and 0 for cross node timeout

Retry count



Alerting


CHECK
mph
SERVICE
ENGINE
SOON
x1000
r/min

SmartCat

Alerting

- Alerting is giving you freedom not to look at graphs
- Someone else placed domain knowledge about alerts
- Alerting must not be frequent since you will end up ignoring alerts

Distributed system -> many alerts -> information overload trap


Grafana
The ~~Boy~~ Who Cried Wolf

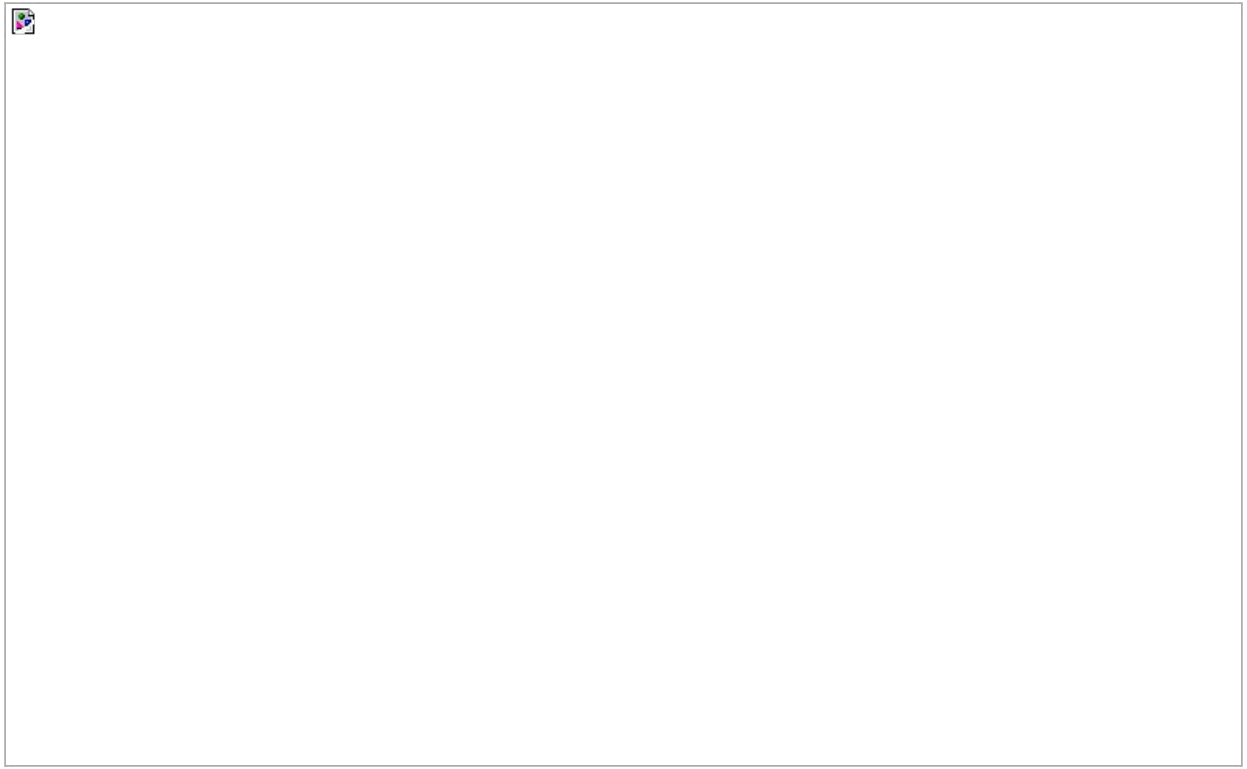
SmartCat

Sentinel - SMART Alerting

- Alerts are build by humans, we make false assumptions
- Correlation between features in most alerting system is not supported
- Why not let the machine find anomalies
- Have snapshot of the system at moment something happened
- Have diagnostic messages with cause of error

SmartCat

Sentinel - SMART Alerting



SmartCat

Sentinel - SMART Alerting

ALERT - Anomaly detected Inbox x

[redacted]

11:37 AM (2 minutes ago)

Time: 2016-10-21 11:07:00

Snapshot of the system:
avg(mem_used_percent) -> 11.481133733107036
avg(diskio_writes) -> 90474.97142857143
avg(cpu_usage_user) -> 19.8143671812
avg(diskio_read_time) -> 95.44593425605537
time_slot -> 1477040820
avg(diskio_read_bytes) -> 0.6123595505617978
avg(diskio_write_bytes) -> 10965.333333333334
avg(queryReport_value) -> 84200.27777777778
avg(diskio_write_time) -> 3.842264177777778E8
avg(requestRate_select_value) -> 22.01406040986498
avg(cpu_usage_system) -> 12041.555555555555
avg(requestRate_update_value) -> 54.957909850772374
avg(cpu_usage_idle) -> 3862055.6470588236
avg(diskio_reads) -> 0.17503878265256603
avg(disk_used_percent) -> 8.538556188444445E9
avg(cpu_usage_steal) -> 62.93553057481667
avg(diskio_io_time) -> 212.26966292134833
avg(cpu_usage_iowait) -> 6.056117222060926

Further Investigation:
Disk I/O write time value was high, please check Cluster Disk Stats.

Conclusion

Takeaways

- Have right amount of information, not too much, not too little
- Having good selection of metrics and logs is iterative process
- Do not end up fixing monitoring machine instead of fixing application code
- Be proactive, not reactive
- Tailor metrics by your needs, build tools if there are not any that suite your use case

Links

- Monitoring stack for distributed systems - SmartCat blog post
- Distributed logging - SmartCat blog post
- Metrics collection stack for distributed systems - SmartCat blog post
- Monitoring machine ansible project (Riemann, Influx, Grafana, ELK) - SmartCat github project

Twitter
@NenadBozicNs

Q&A

SmartCat.io

Thank you

Nenad Bozic
@NenadBozicNs

SmartCat.io

SmartCat
www.smartcat.io
@SmartCat_io