Abto Software

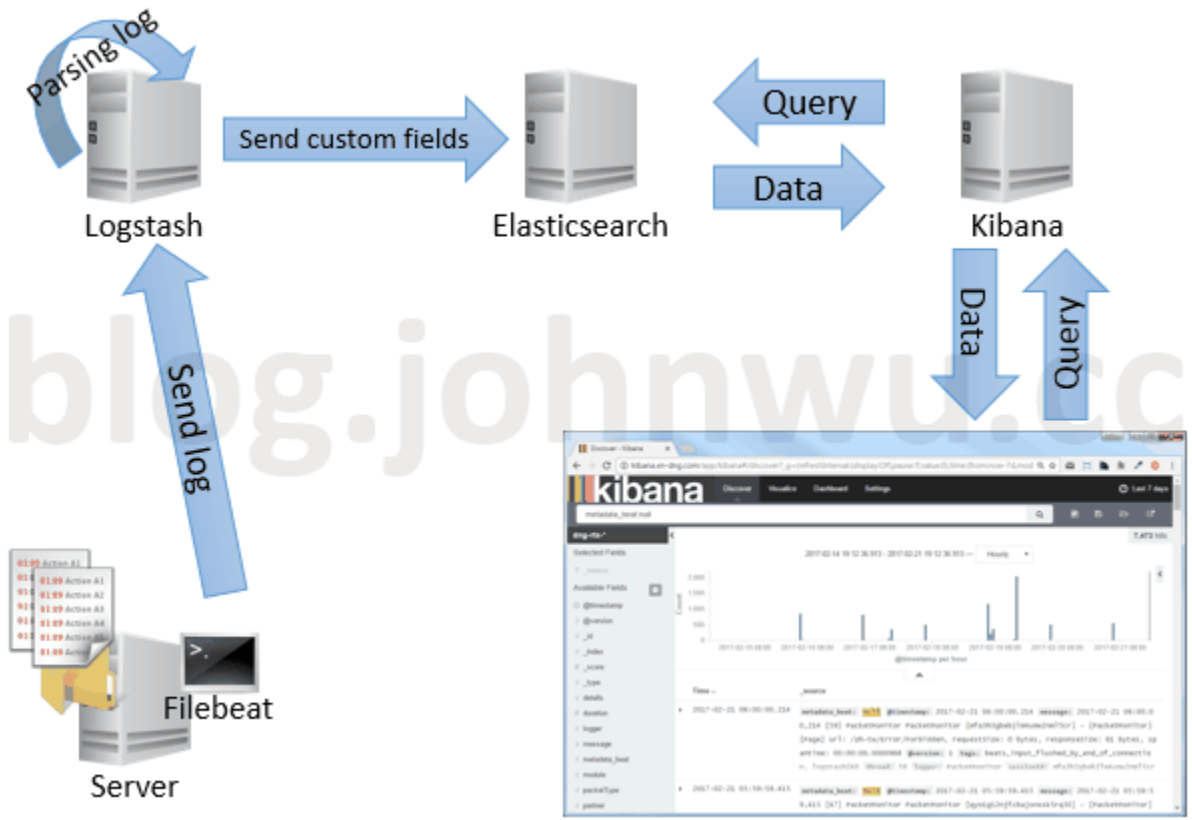
just answer

Побудова і оптимізація Elasticsearch Cluster

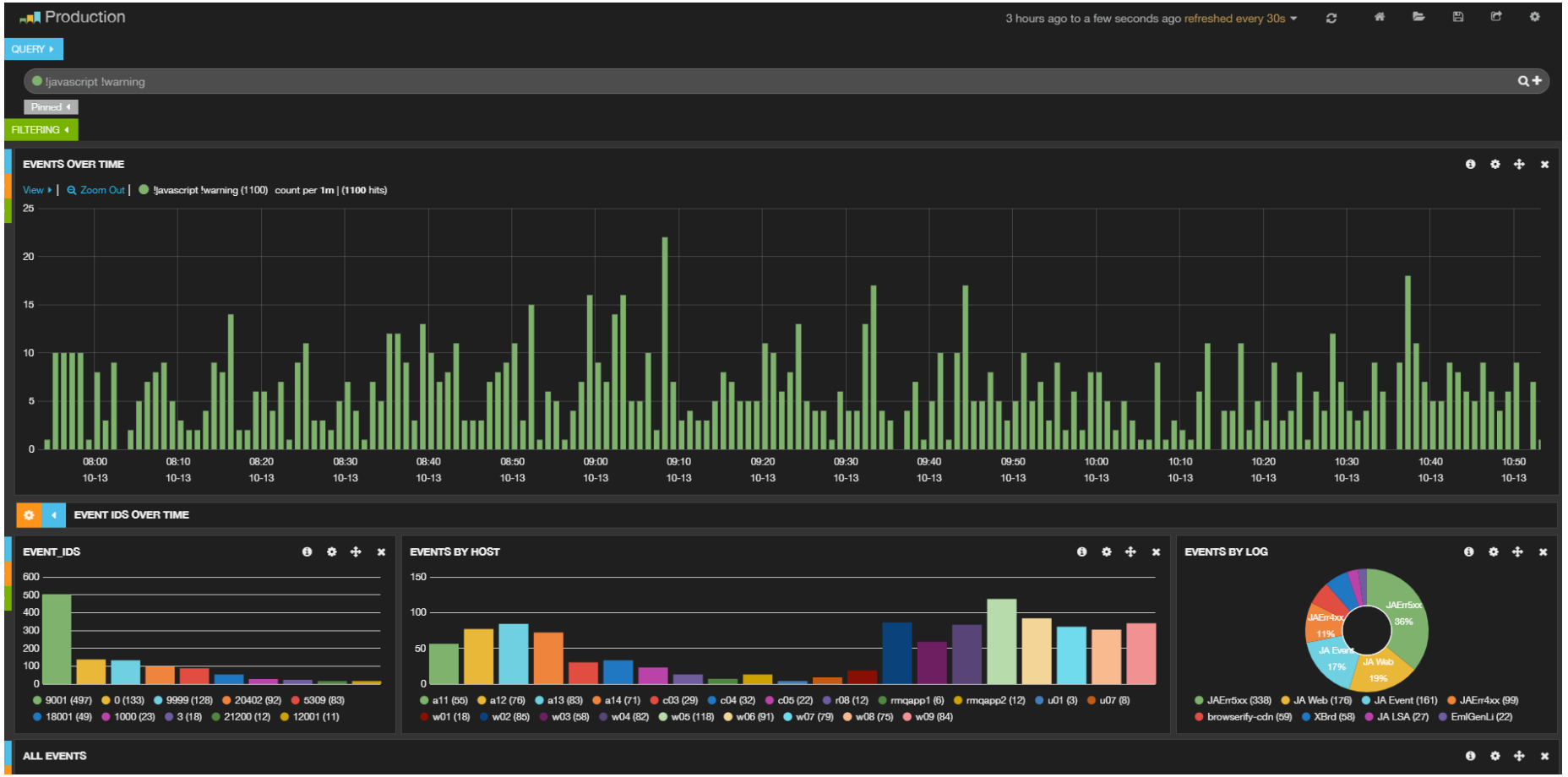
Elasticsearch

- Elasticsearch is a distributed, RESTful search and analytics engine capable of solving a growing number of use cases. As the heart of the Elastic Stack, it centrally stores your data so you can discover the expected and uncover the unexpected.

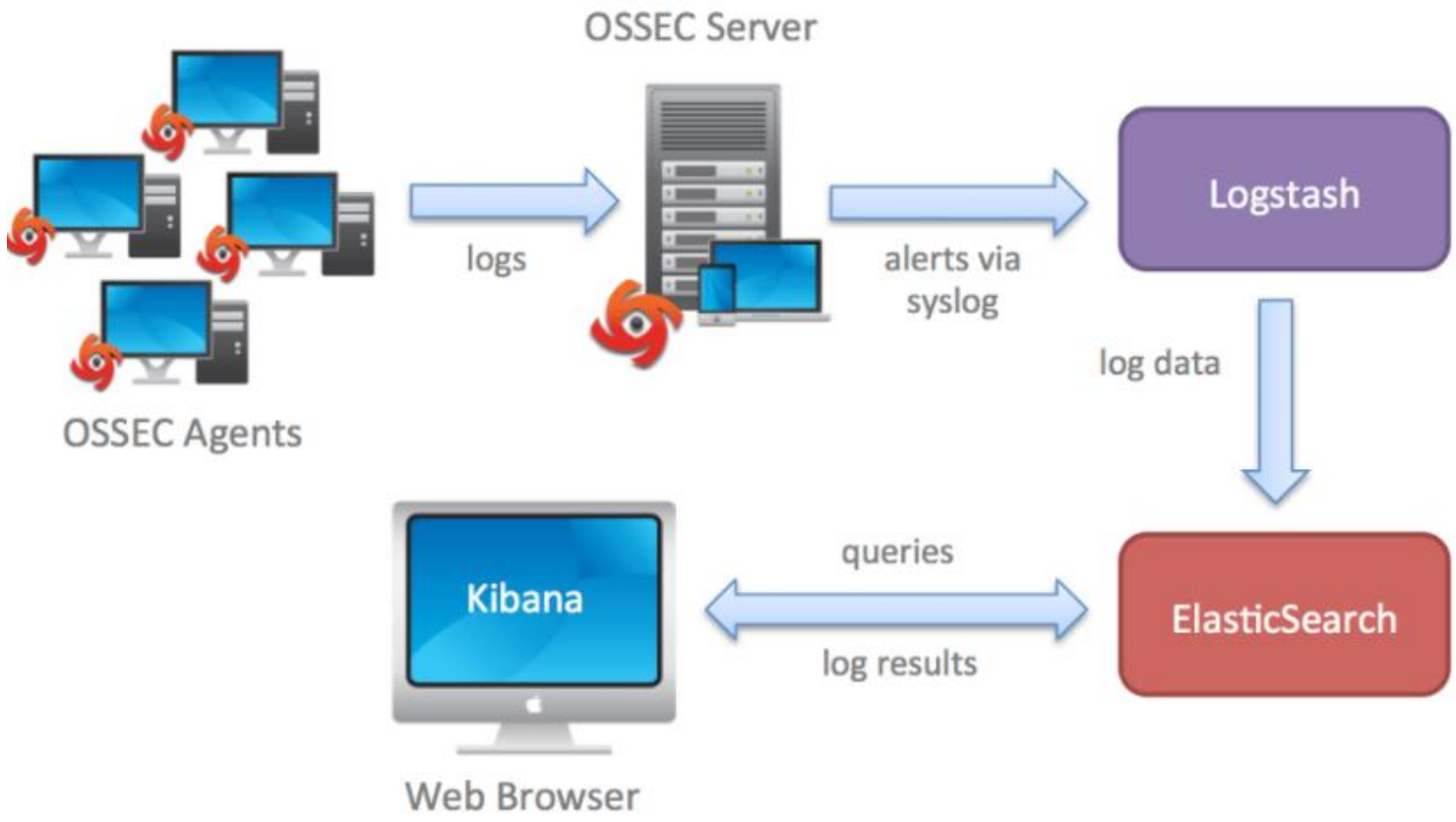
Elasticsearch – ELK stack



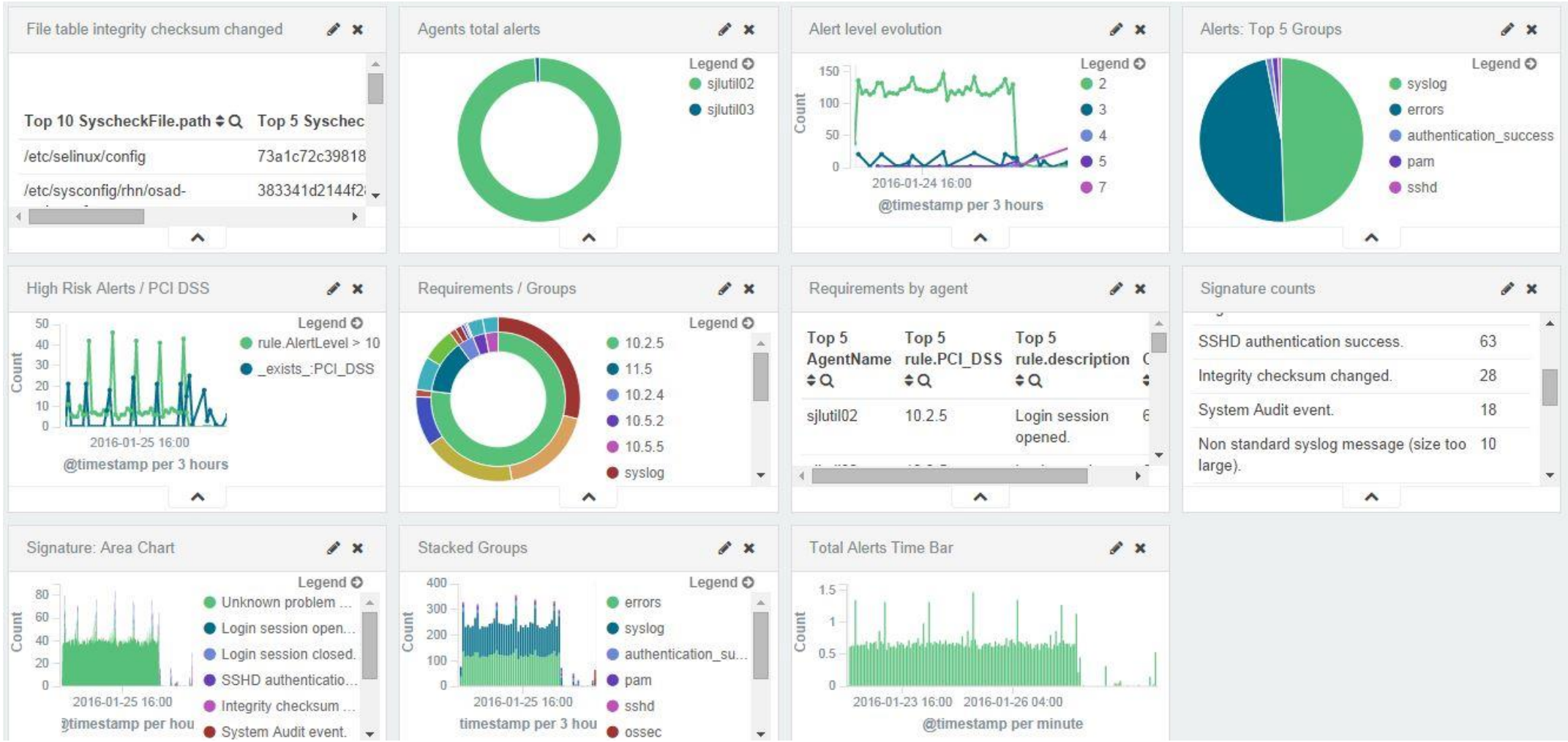
Elasticsearch – ELK stack



Elasticsearch - OSSEC



Elasticsearch - OSSEC



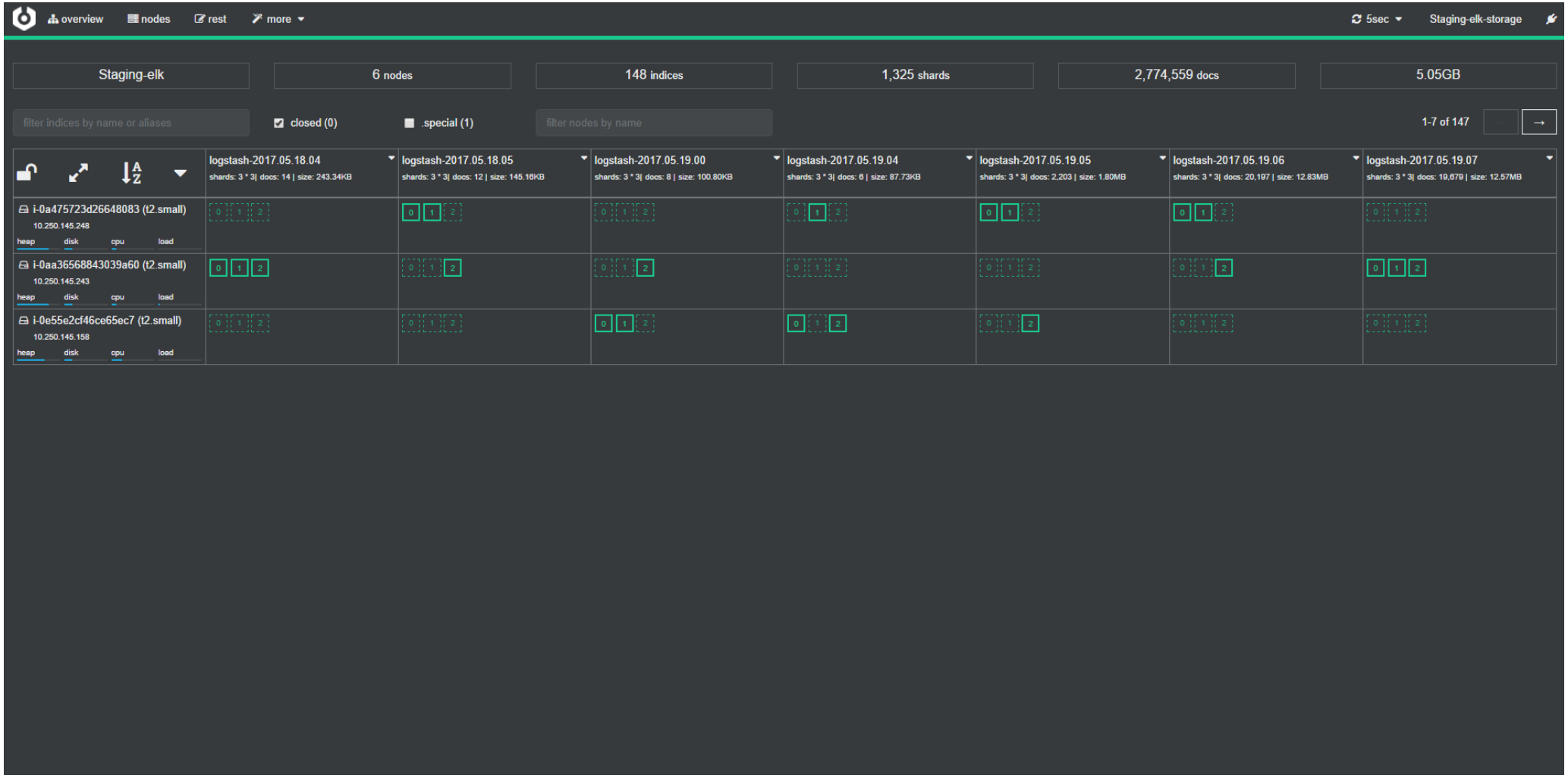
Elasticsearch Terminology

- **Elasticsearch**: It is a horizontally distributed, data storage, search server, aggregation engine, based on Lucene library. It is written in java.
- **Cluster**: A cluster consists of one or more nodes which share the same cluster name. Each cluster has a single master node which can be replaced if the current master node fails.
- **Node**: A node is a running instance of Elasticsearch which belongs to a cluster. Multiple nodes can be started on a single server. At startup, a node will use unicast to discover an existing cluster with the same cluster name and will try to join that cluster.

Elasticsearch Terminology

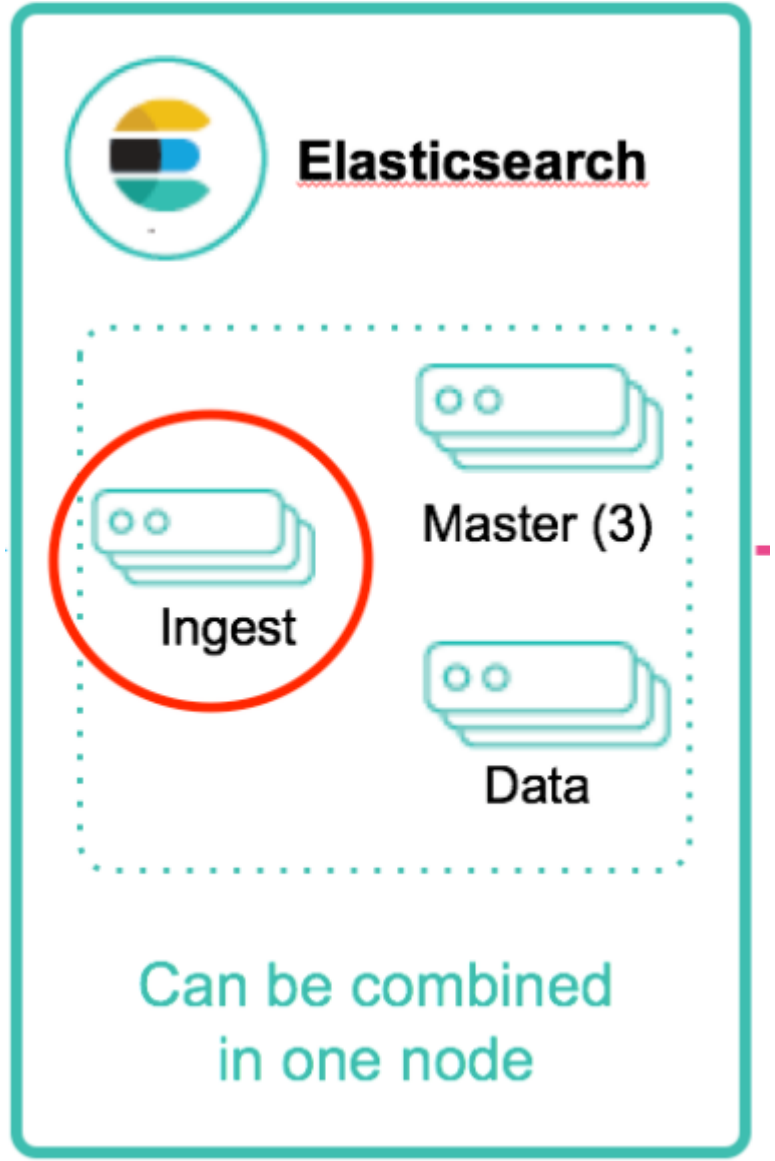
- **Index:** An index is a collection of documents that have somewhat similar characteristics. For example, you can have an index for customer data, another index for a product catalog, and yet another index for order data.
- **Primary Shard**: Each document is stored in a single primary shard. When you index a document, it is indexed first on the primary shard, then on all replicas of the primary shard. By default, an index has 5 primary shards.
- **Replica Shard**: Each primary shard can have zero or more replicas. A replica is a copy of the primary shard. By Default there are 1 replica for each primary shards.
- **Document**: A document is a JSON document which is stored in Elasticsearch. It is like a row in a table in a relational database. Each document is stored in an index and has a type and an id. A document is a JSON object which contains zero or more fields, or key-value pairs.

Elasticsearch Terminology

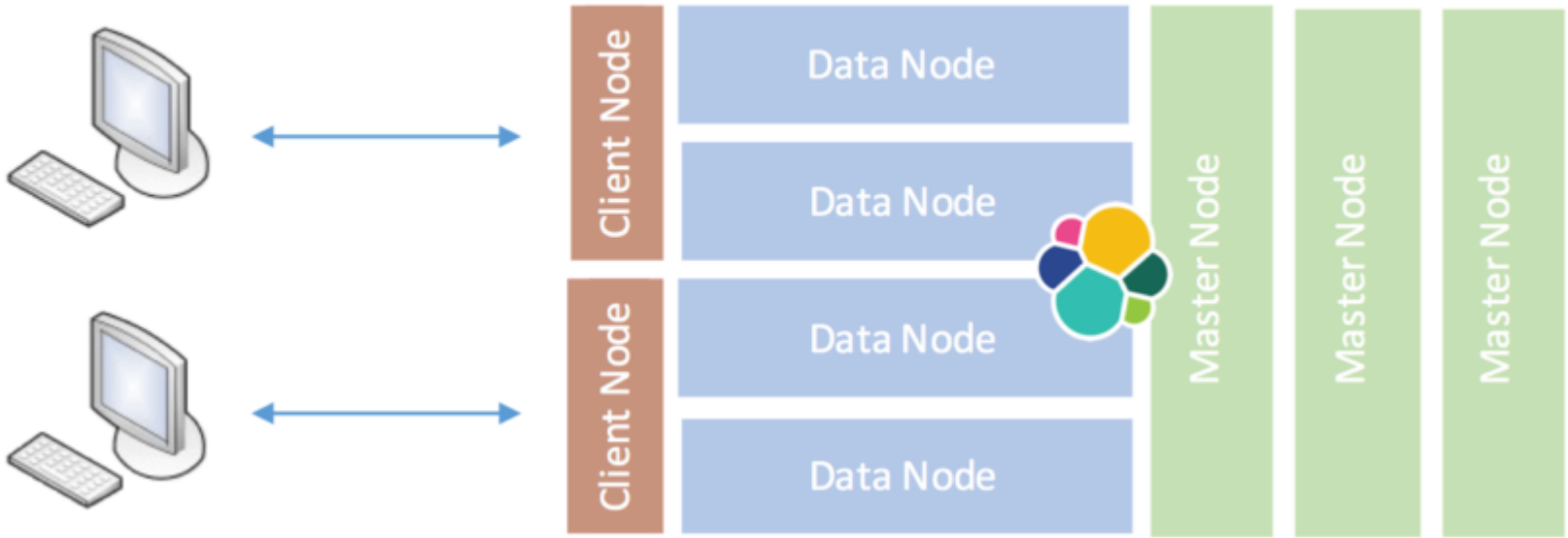


ElasticSearch Cluster Nodes

- **Master** only nodes take place in updating cluster state as well as master elections. They should never handle query or index loads.
- **Data** only nodes store data that is indexed into Elasticsearch. These can also handle querying and indexing.
- **Client/Ingest** only nodes are used as load balancers for indexing and searching.



ElasticSearch Cluster

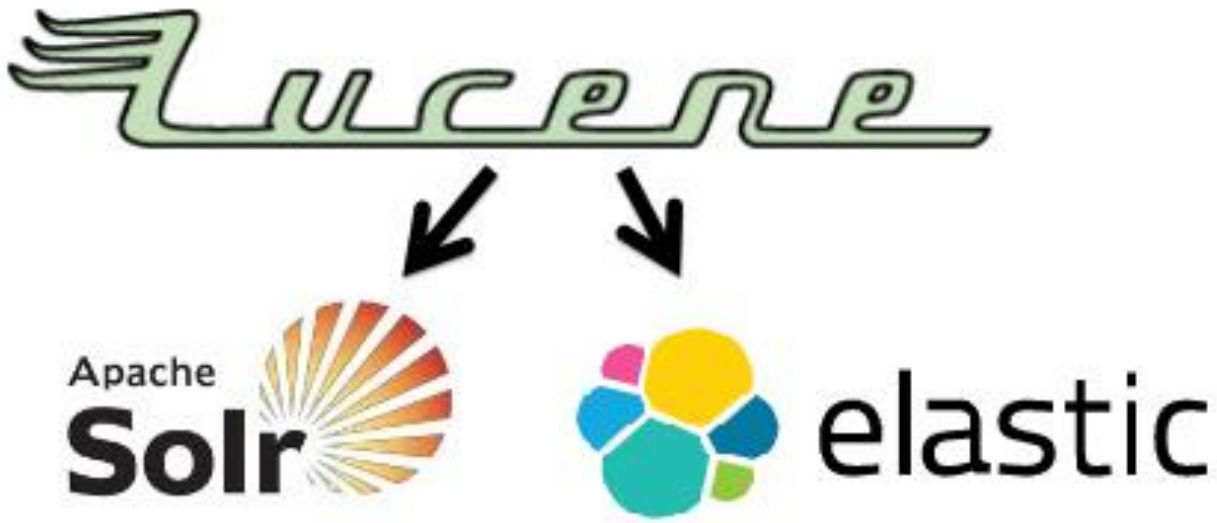



REALITY-CHECK

Real Case

- Platform for helping people each other
- 12000 Experts in different areas
- 24+ millions hits per day
- 1 million of unique visitors
- 16+ millions of answered questions

Real Case



Product Case

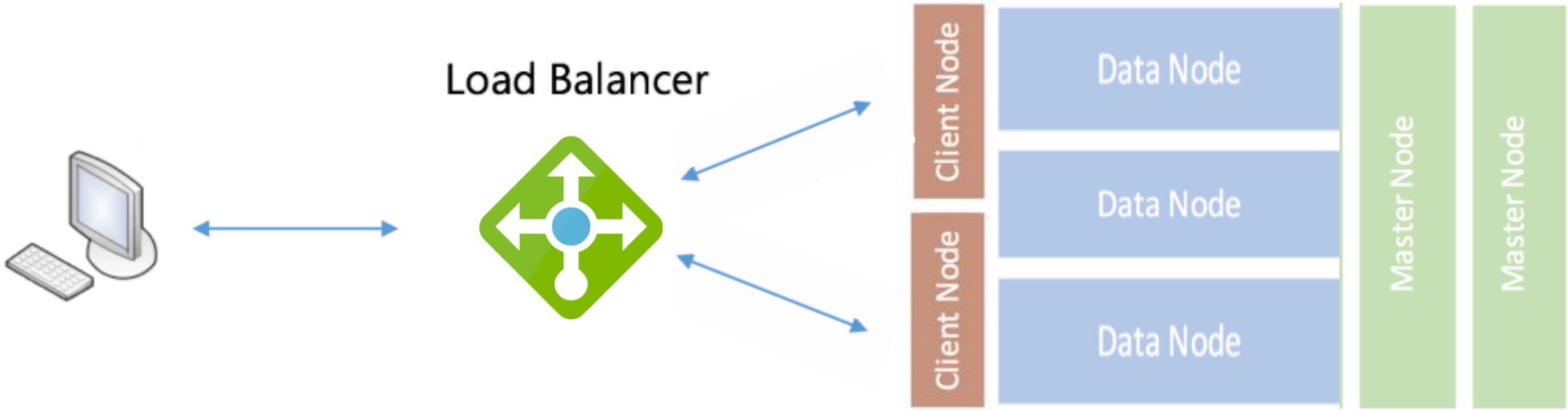
Basically - sisters

SOLR state on project

Elasticsearch state on project (after migration)

https://www.flickr.com/photos/franzfume/11530902934/

Production Cluster



Cluster Configuration

- cluster.name: cluster
- node.name: node1
- bootstrap.memory_lock: true
- network.bind_host: 192.168.1.1
- discovery.zen.ping.unicast.hosts: [“node1", “node2"]
- discovery.zen.minimum_master_nodes: 2
- node.master: false
- node.data: true
- node.ingest: false

Hardware Specification

Master node:

- 2GB RAM
- 2vCPU
- 40GB Disk
- SSD

Client node:

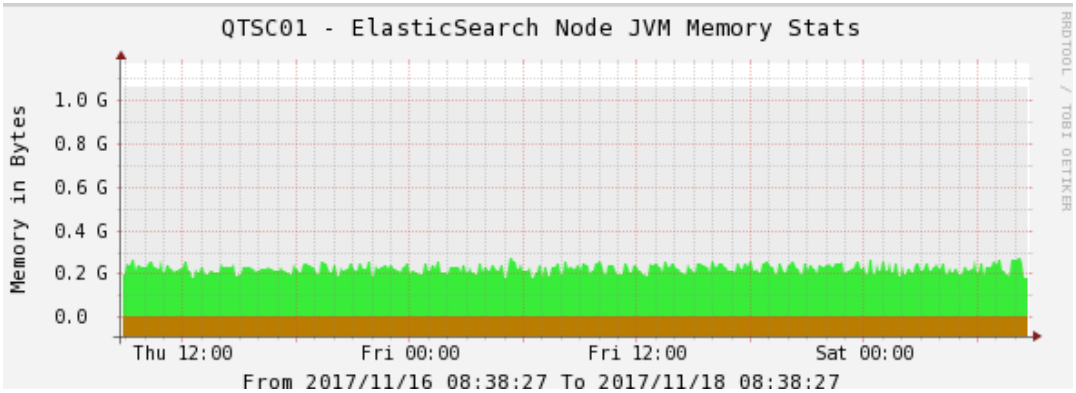
- 2GB RAM
- 2vCPU
- 40GB Disk
- SSD

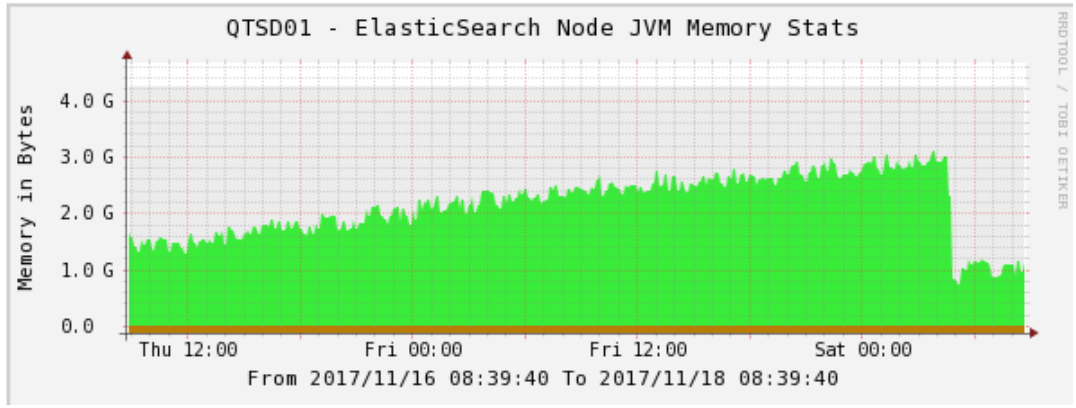
Data node:

- 4GB RAM
- 4vCPU
- 80GB Disk
- SSD

Performance - Memory

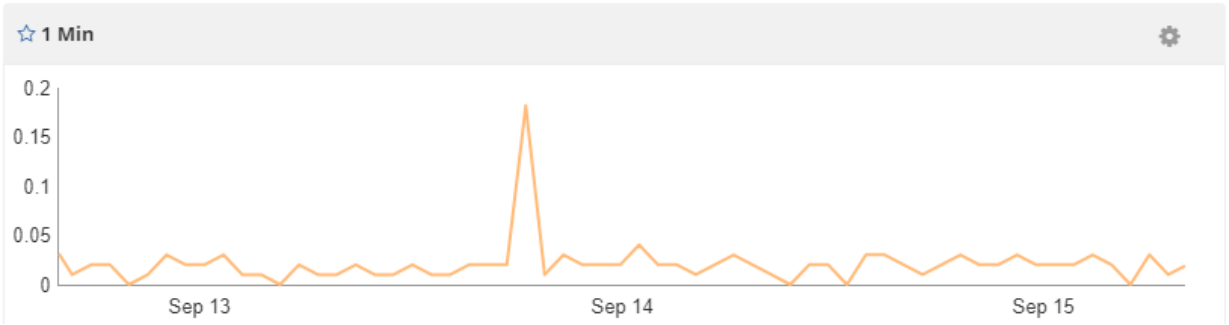
- 5 000 000 documents
- 90 requests per second
- 10 threads

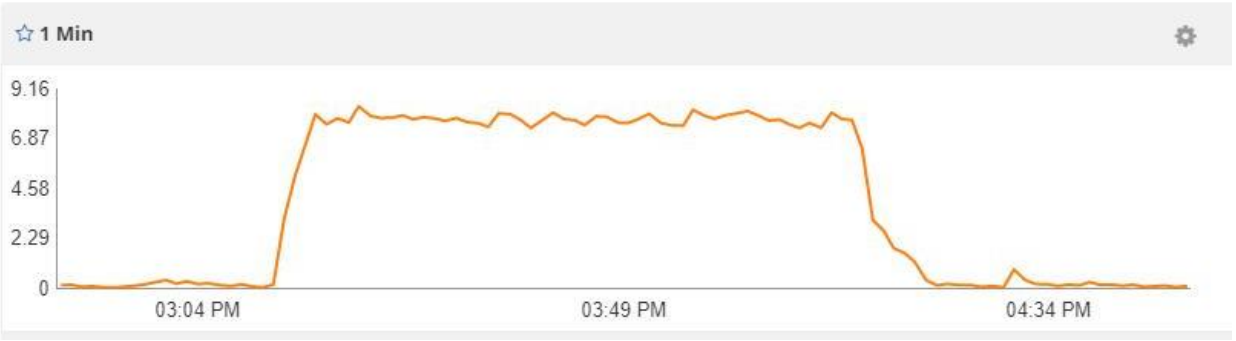




Performance Load

- 5 000 000 documents
- 90 requests per second
- 10 threads



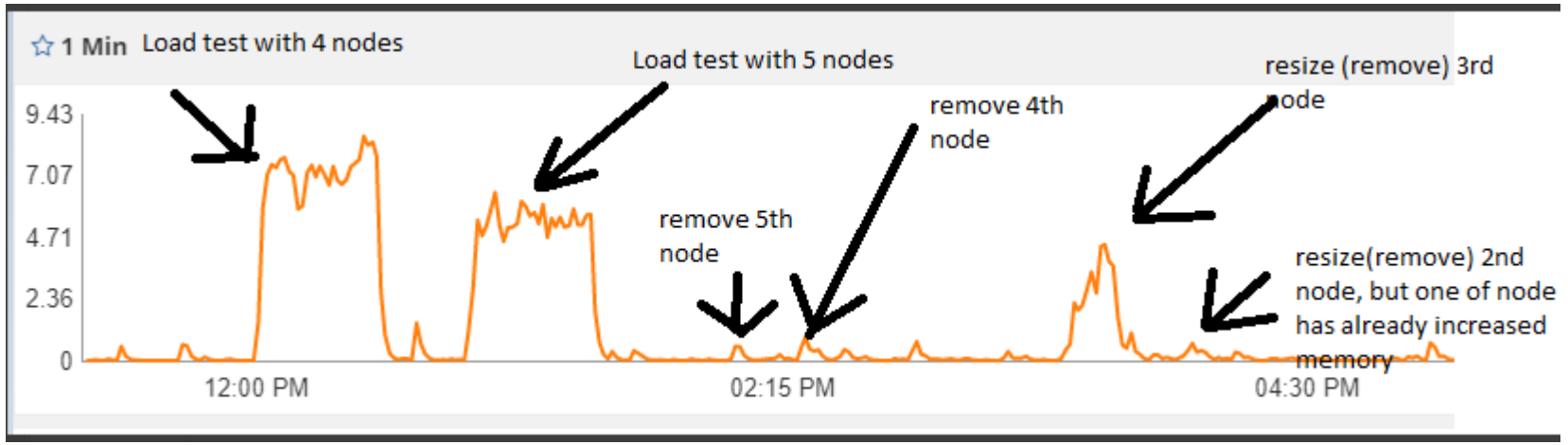


Performance Improving

Step 1: Added extra 2 data nodes. Test with 5 data nodes.

Step 2: Increase memory on existing 3 data nodes. Test.

Step 3: Increase memory on all 5 data nodes. Test.



Performance

Tested impact of adding new nodes vs adding ram

- 3 nodes - response time - 22.3s
- 4 nodes - response time - 16.1s
- 5 nodes - response time - 13.4s
- 3 nodes with increased ram - 7.9s
- 5 nodes with increased ram – 4.1s

Conditions

- 90 requests per second
- 10 threads
- 10 millions documents

Hardware Specification

Master node:

- 2GB RAM
- 2vCPU
- 40GB Disk
- SSD

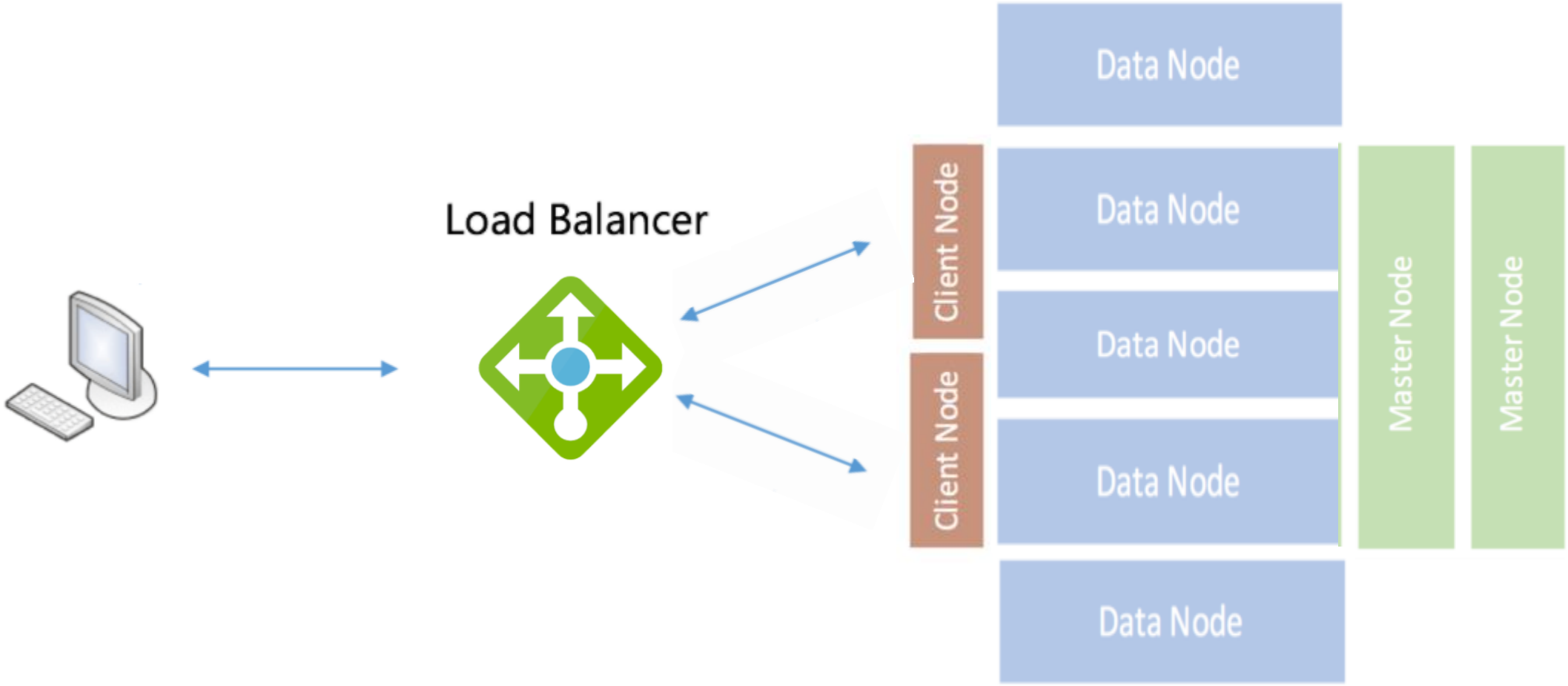
Client node:

- 2GB RAM
- 2vCPU
- 40GB Disk
- SSD

Data node:

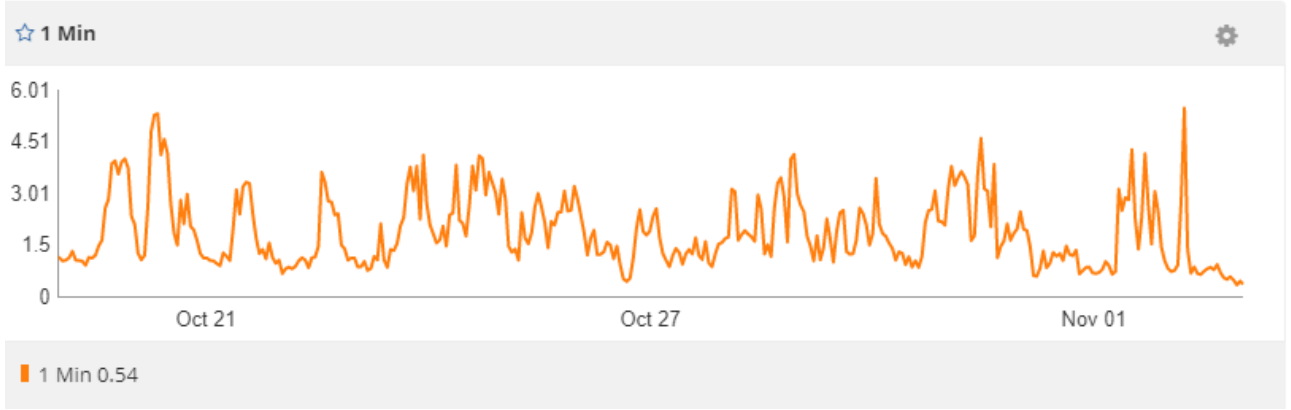
- 8GB RAM
- 8vCPU
- 160GB Disk
- SSD

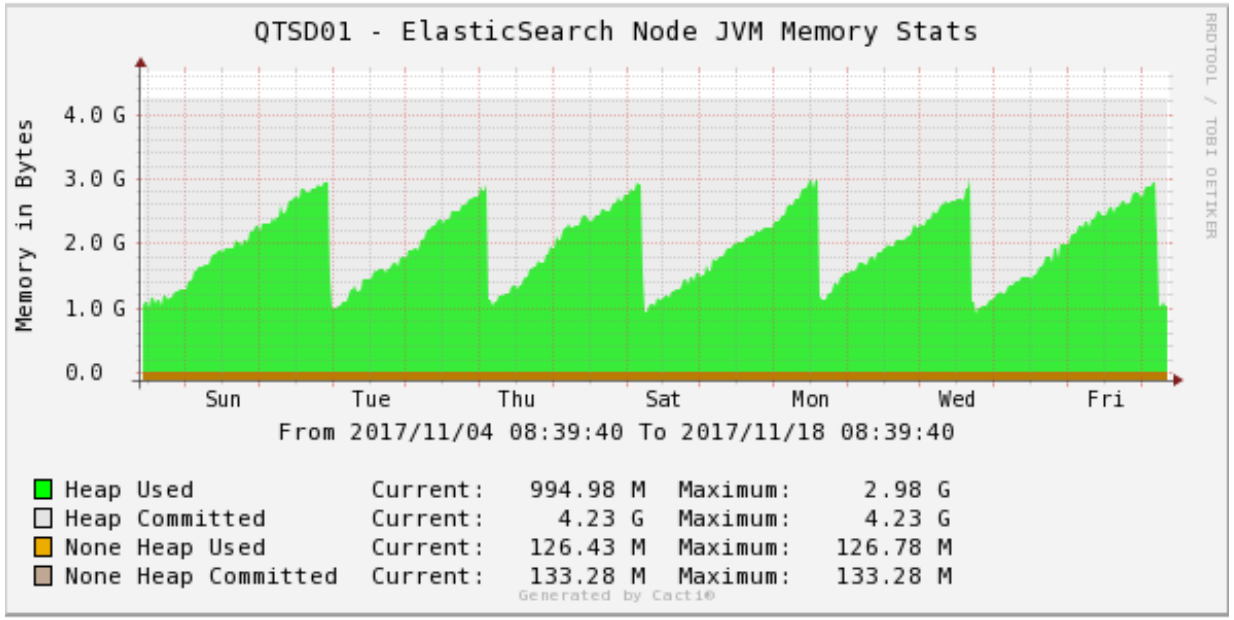
Production Cluster Final Architecture



Performance

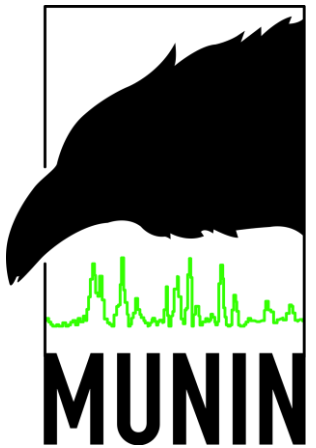
- 21 000 000 documents
- 85 GB repo
- 90ms of indexing
- 5 data nodes



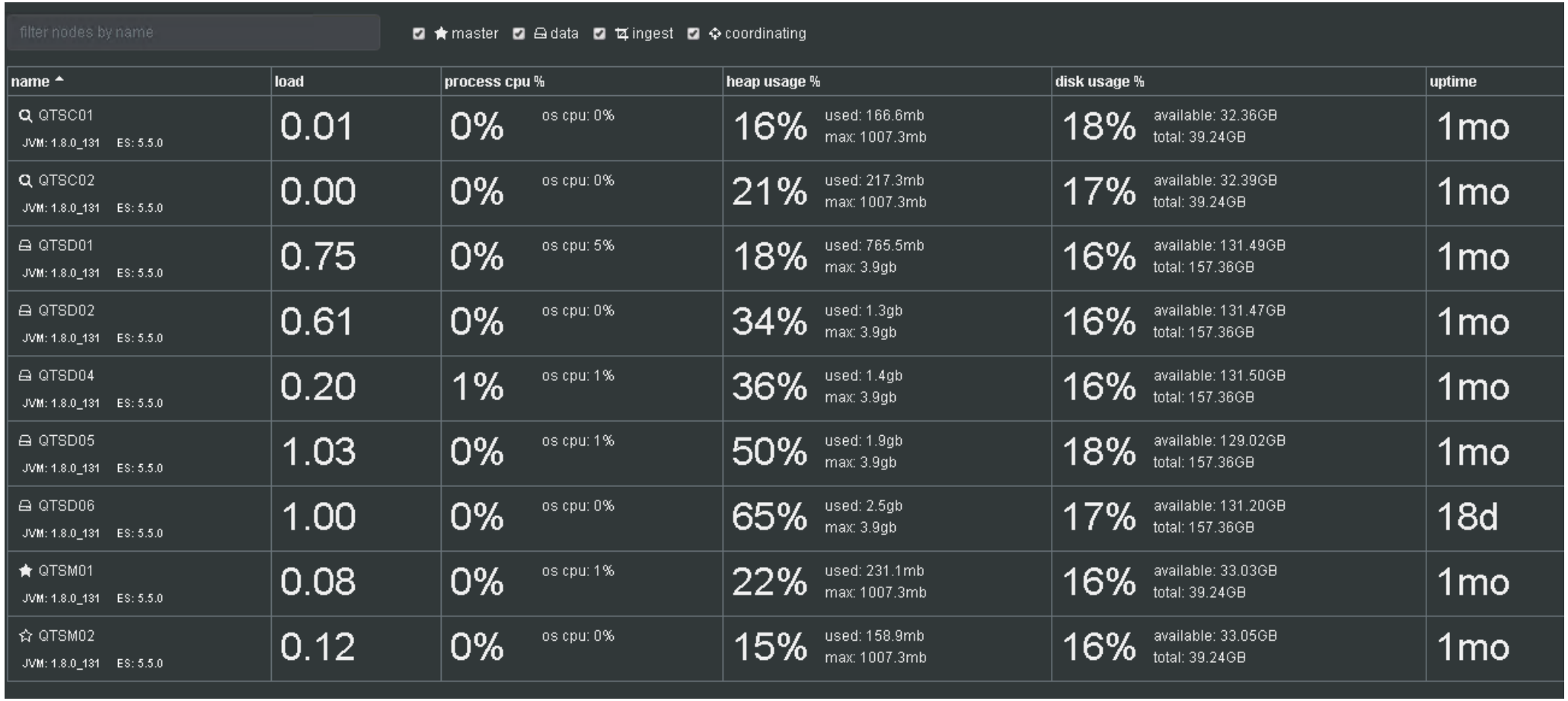


Monitoring ES Cluster

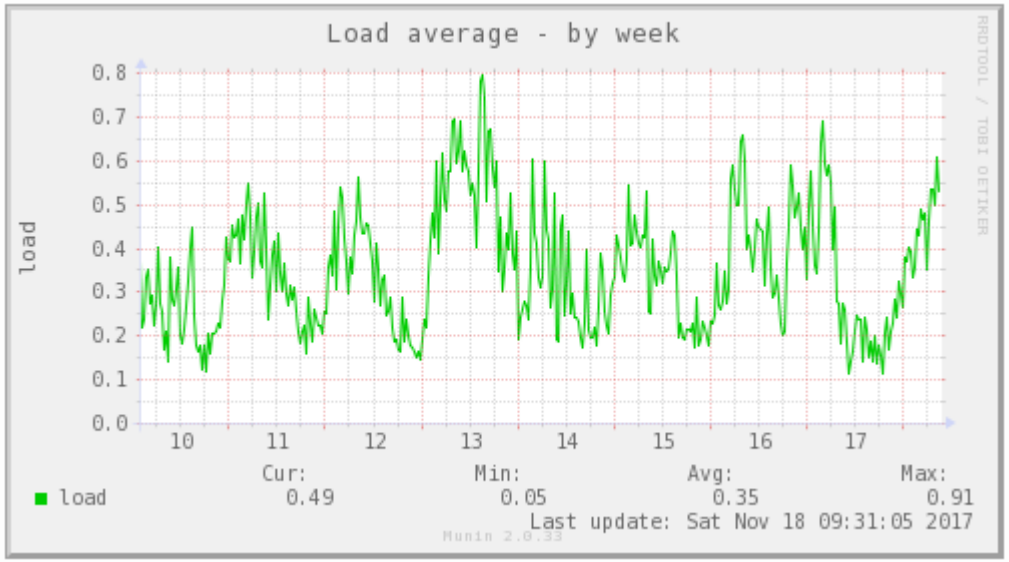
- Cerebro plugin
- X-Pack plugin
- Cacti
- Munin
- ELK stack ☺

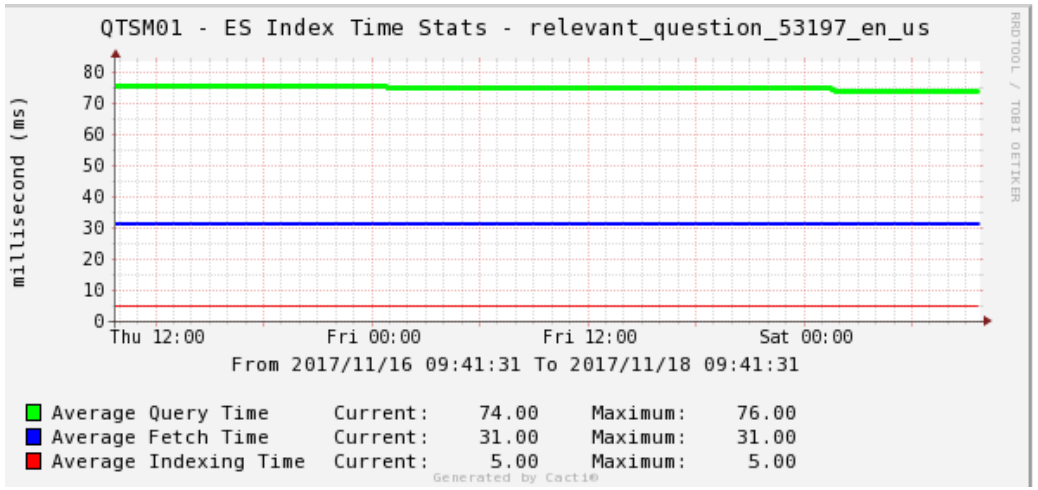


Cerebro

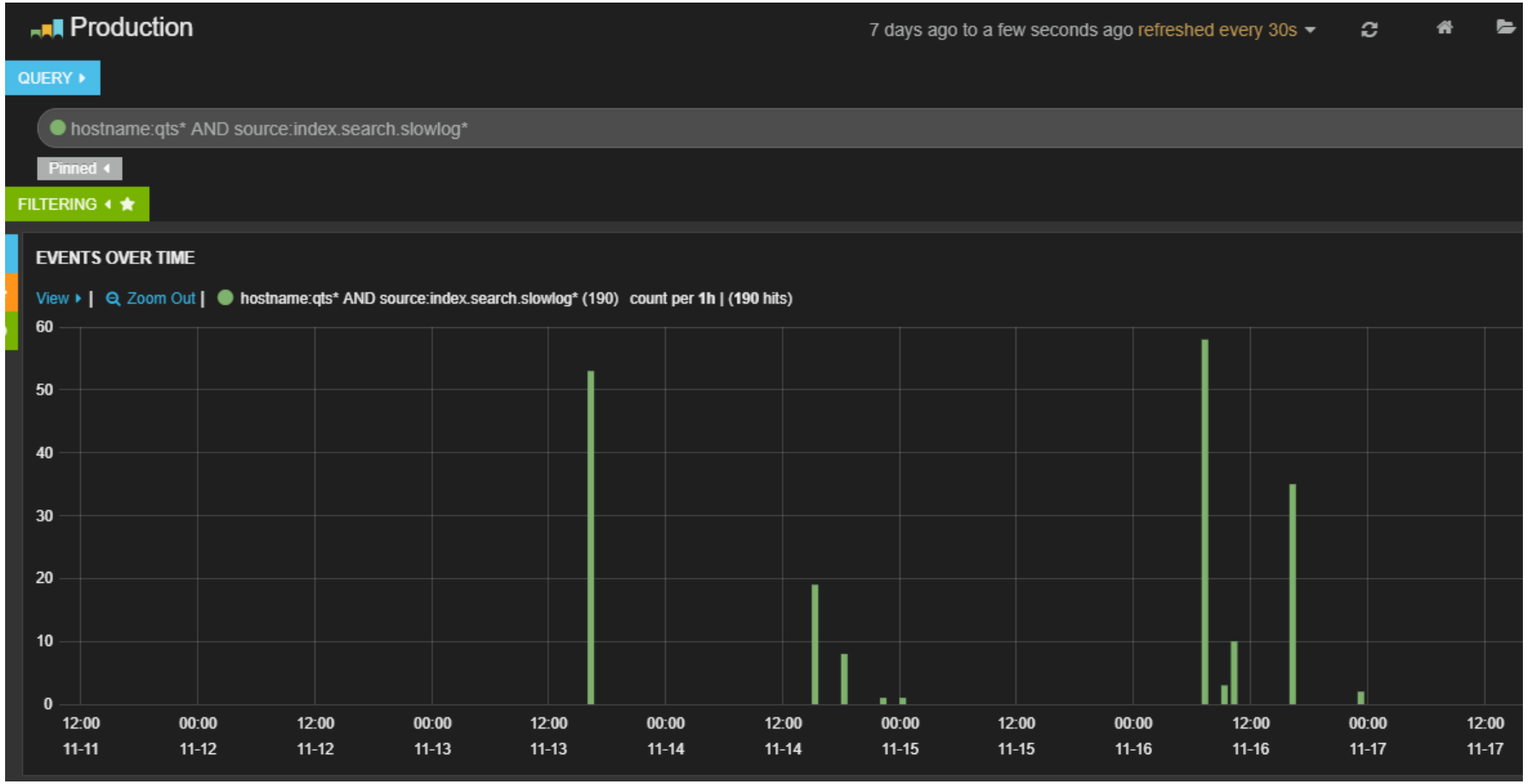

filter nodes by name
master
data
ingest
coordinating
name
load
process cpu %
heap usage %
disk usage %
uptime
QTSC01
JVM: 1.8.0_131 ES: 5.5.0
0.01
0%
os cpu: 0%
16%
used: 166.6mb
max: 1007.3mb
18%
available: 32.36GB
total: 39.24GB
1mo
QTSC02
JVM: 1.8.0_131 ES: 5.5.0
0.00
0%
os cpu: 0%
21%
used: 217.3mb
max: 1007.3mb
17%
available: 32.39GB
total: 39.24GB
1mo
QTSD01
JVM: 1.8.0_131 ES: 5.5.0
0.75
0%
os cpu: 5%
18%
used: 765.5mb
max: 3.9gb
16%
available: 131.49GB
total: 157.36GB
1mo
QTSD02
JVM: 1.8.0_131 ES: 5.5.0
0.61
0%
os cpu: 0%
34%
used: 1.3gb
max: 3.9gb
16%
available: 131.47GB
total: 157.36GB
1mo
QTSD04
JVM: 1.8.0_131 ES: 5.5.0
0.20
1%
os cpu: 1%
36%
used: 1.4gb
max: 3.9gb
16%
available: 131.50GB
total: 157.36GB
1mo
QTSD05
JVM: 1.8.0_131 ES: 5.5.0
1.03
0%
os cpu: 1%
50%
used: 1.9gb
max: 3.9gb
18%
available: 129.02GB
total: 157.36GB
1mo
QTSD06
JVM: 1.8.0_131 ES: 5.5.0
1.00
0%
os cpu: 0%
65%
used: 2.5gb
max: 3.9gb
17%
available: 131.20GB
total: 157.36GB
18d
QTSM01
JVM: 1.8.0_131 ES: 5.5.0
0.08
0%
os cpu: 1%
22%
used: 231.1mb
max: 1007.3mb
16%
available: 33.03GB
total: 39.24GB
1mo
QTSM02
JVM: 1.8.0_131 ES: 5.5.0
0.12
0%
os cpu: 0%
15%
used: 158.9mb
max: 1007.3mb
16%
available: 33.05GB
total: 39.24GB
1mo

Cacti , Munin


Load average - by week
load
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
10
11
12
13
14
15
16
17
Cur:
Min:
Avg:
Max:
load
0.49
0.05
0.35
0.91
Last update: Sat Nov 18 09:31:05 2017
RRDTOOL / TOBI OETIKER


QTSM01 - ES Index Time Stats - relevant_question_53197_en_us
millisecond (ms)
80
70
60
50
40
30
20
10
0
Thu 12:00
Fri 00:00
Fri 12:00
Sat 00:00
From 2017/11/16 09:41:31 To 2017/11/18 09:41:31
Average Query Time Current: 74.00 Maximum: 76.00
Average Fetch Time Current: 31.00 Maximum: 31.00
Average Indexing Time Current: 5.00 Maximum: 5.00
RRDTOOL / TOBI OETIKER

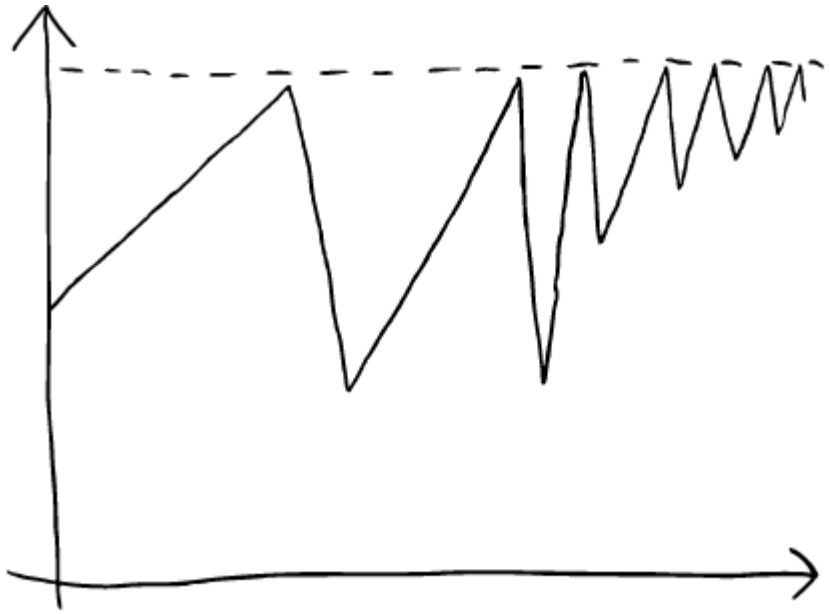
Slowlog in ELK



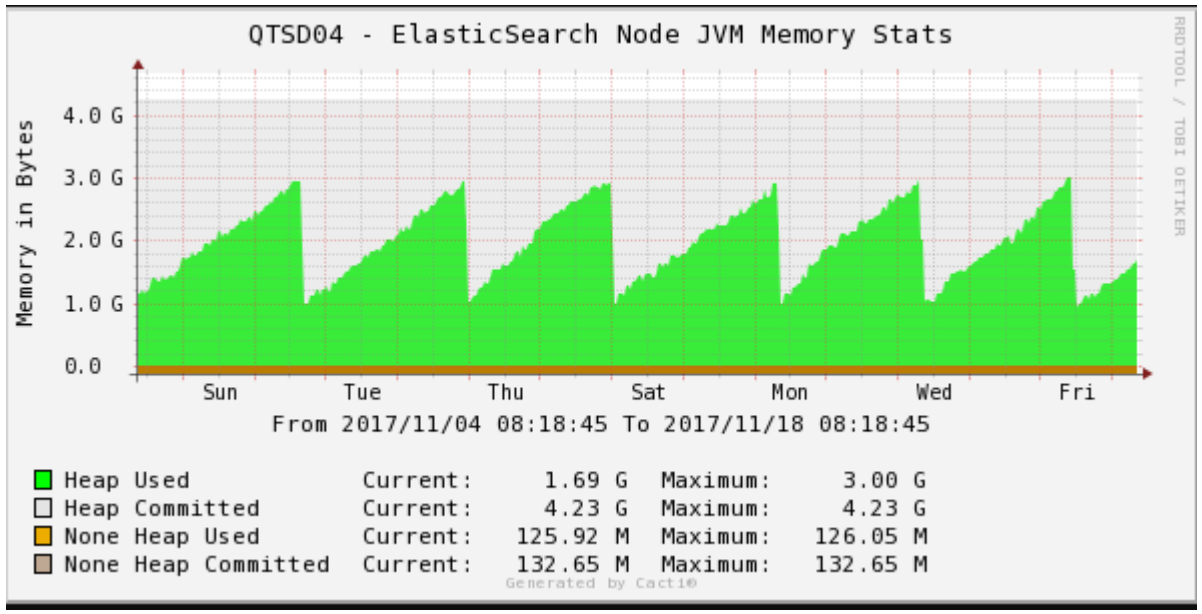
Performance
where to keep an eye

- Bootstrap parameters in config
- Enough free disk space
- Garbage collection behavior
- Slowlog Monitoring
- Rolling restart

Garbage Collection

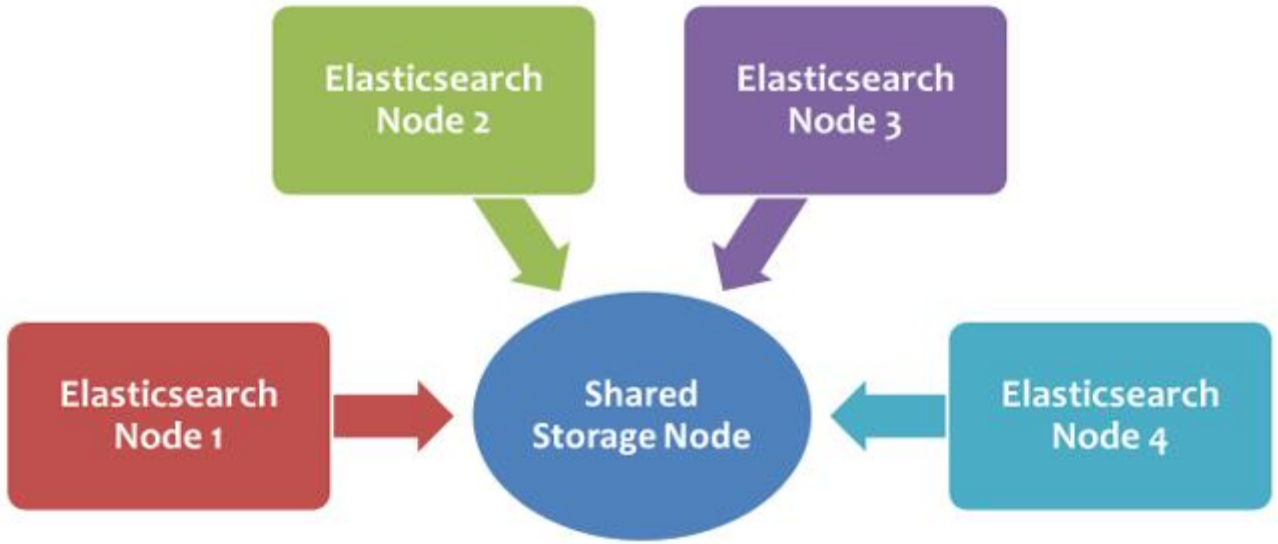


Going to be problem



Normal behavior

Elasticsearch Backup



Abto Software

just answer

Дякую!

Thank You!