

# IT Monitoring

[massimo.paladin@cern.ch](mailto:massimo.paladin@cern.ch)

CERN IT-CF

HEPiX Fall 2013

28<sup>th</sup> October 2013



## Motivation

- Several **independent monitoring activities** in CERN IT
- Combination of data from different groups necessary
- Understanding performance became more important
- Move to a virtualized dynamic infrastructure

## Challenges

- Implement a **shared architecture** and **common tool-chain**
- Delivered under a common collaborative effort

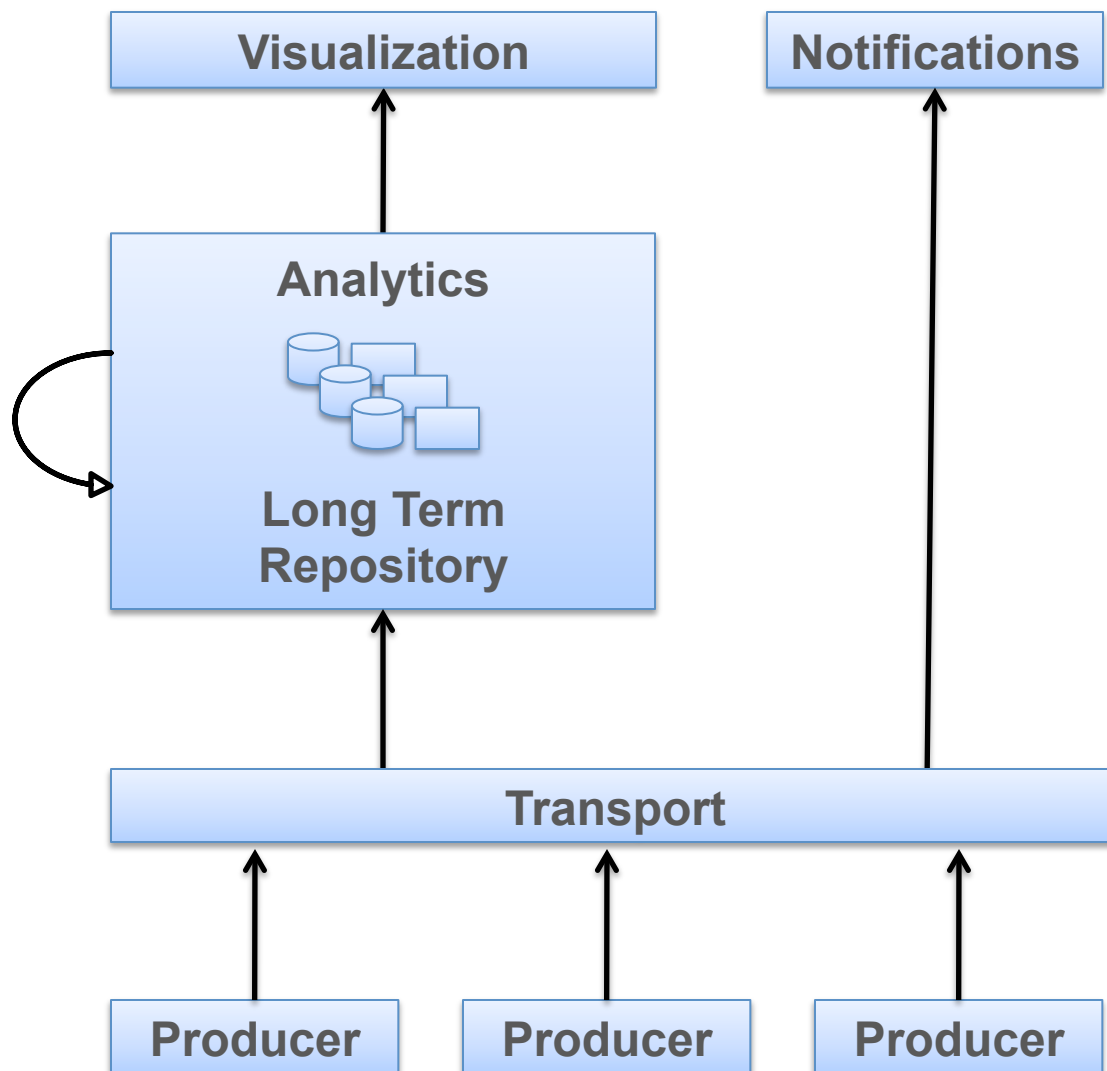
## Adopt open source tools

- For each architecture block look outside for solutions
- Large adoption and strong community support
- Fast to adopt, test, and deliver
- Easily replaceable by other (better) future solutions

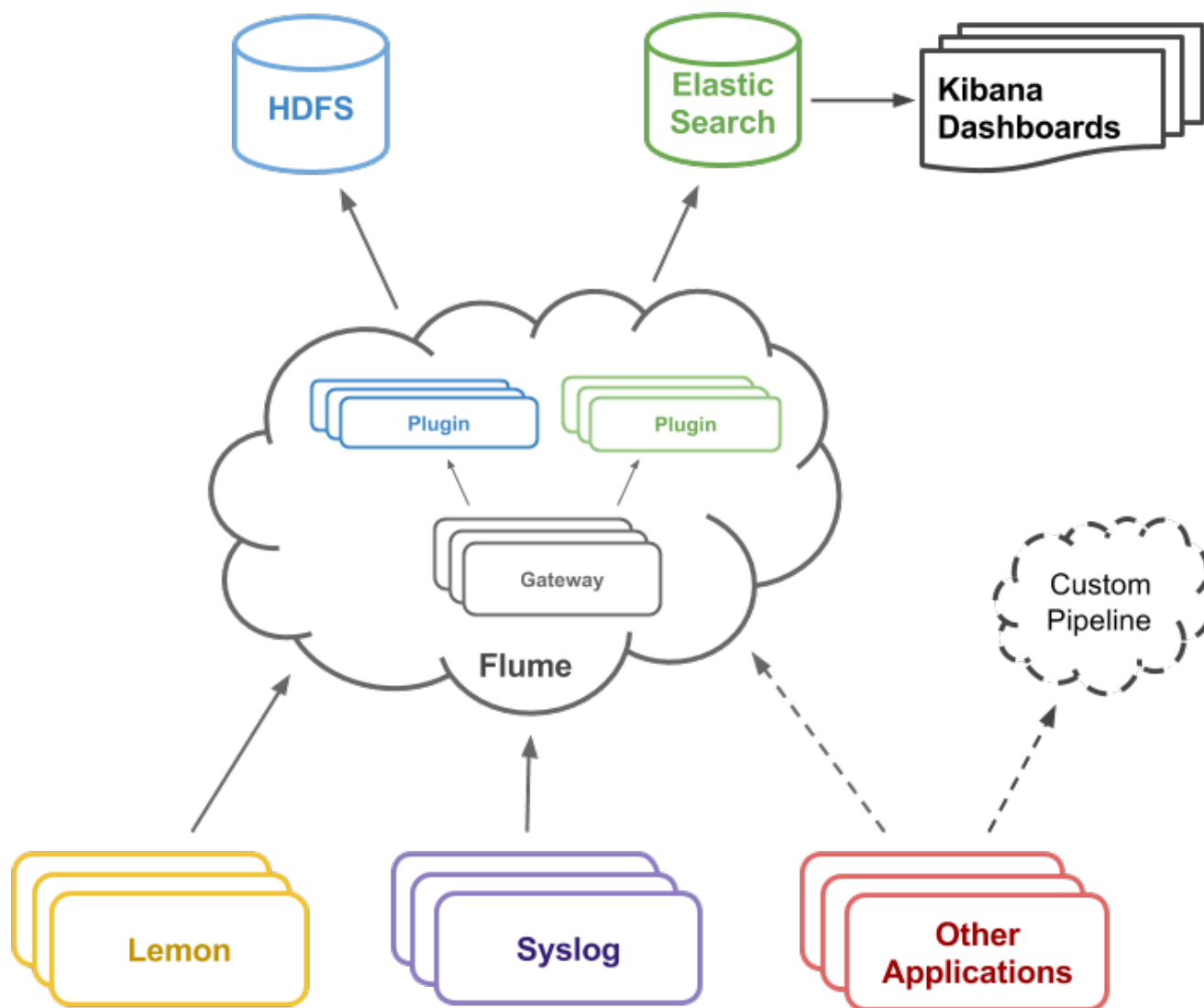
## Integrate with new CERN infrastructure

- AI project, OpenStack, Puppet, Roger, etc.

## Focus on simple adoption (e.g. puppet modules)



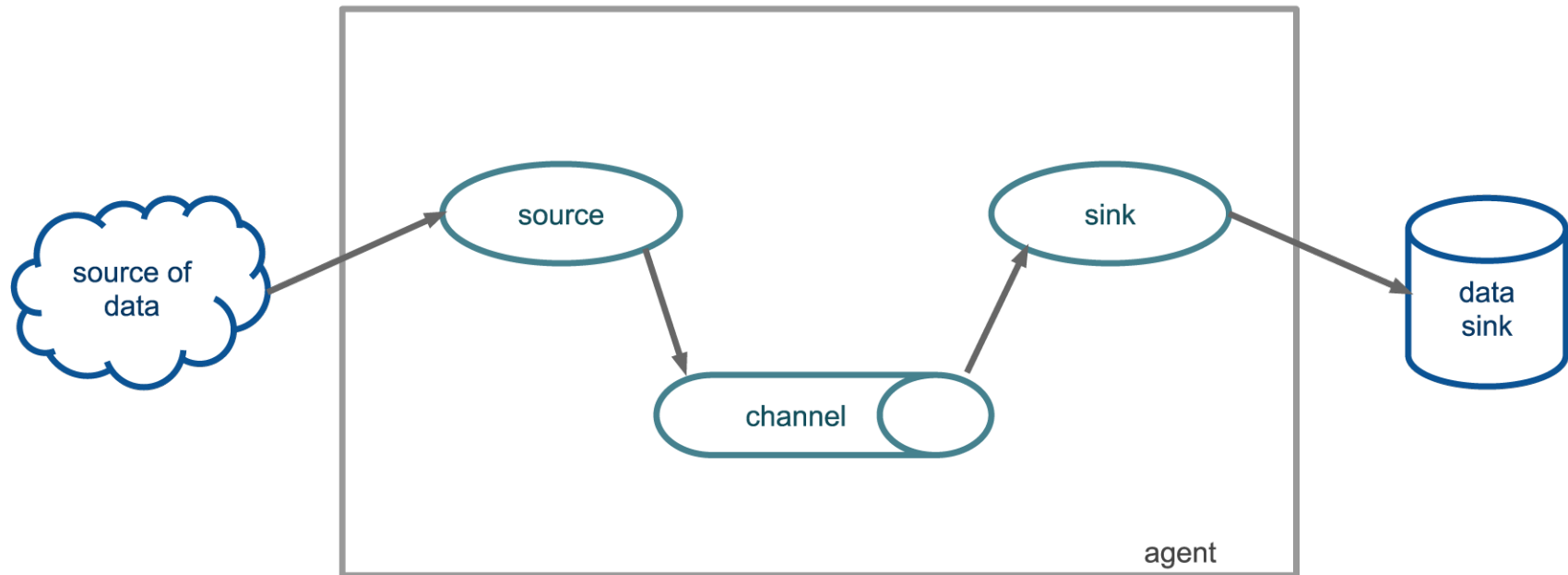
- Integrate data from multiple producers
- Scalable transport collect operations data
- Long term archival for offline processing
- Analytics: real time queries, limited data retention
- Visualization: dynamic and user-friendly dashboards



## Distributed service for collecting large amounts of data

- Robust and fault tolerant
- Horizontally scalable, multi-tier deployment
- Many ready to be used input and output plugins
  - Avro, Thrift, JMS, Syslog, HTTP, ES, HDFS, Custom, ...
- Java based, Apache license





## Flume event

- Byte payload + set of string headers

## Flume agent

- JVM process hosting “source -> sink” flow(s)



## Routing is static

- On demand subscriptions are not possible
- Requires reconfiguration and restart

## No authentication/authorization features

- Secure transport available

## Java process on client side

- Smaller memory footprint would be nicer

## Needs tuning to correctly size flume layers

## Available sources and sinks saved a lot of time

## Horizontally scalable

## In-flight data filtering possible





Distributed framework for large data sets processing

Distributed filesystem designed for commodity HW

- Suitable for applications with large data sets
- Cluster provided by other IT group (DSS)
- Data stored by cluster (might be revised)
- Daily jobs to aggregate data by month

## Feedback

- Large analytics potential to explore
- Reliable external long term repository



## Distributed RESTful search and analytics engine

- Real time acquisition, data is indexed in real time
- Automatically balanced shards and replicas
- Schema free, document oriented (JSON)
  - No prior data declaration required
  - Automatic data type discovery
- Based on Lucene (full-featured IR library)
  - Full text search
- RESTful JSON API

```
$ curl -XGET http://es-search:9200/_cluster/health?pretty=true
{
  "cluster_name" : "itmon-es",
  "status" : "green",
  "timed_out" : false,
  "number_of_nodes" : 11,
  "number_of_data_nodes" : 8,
  "active_primary_shards" : 2990,
  "active_shards" : 8970,
  "relocating_shards" : 0,
  "initializing_shards" : 0,
  "unassigned_shards" : 0
}
```



## Used by many large companies

- Soundcloud
  - “To provide immediate and relevant results for their online audio distribution platform reaching 180 million people”
- Github
  - “20TB of data using ElasticSearch, including 1.3 billion files and 130 billion lines of code”
- Foursquare, Stackoverflow, Salesforce, ...

## Distributed under Apache license



Requires a lot of RAM (Java), IO intensive

- Take into account when planning deployment

Shards re-initialisation takes some time (~1h)

- Not frequent operation, only after full cluster reboot

Authentication not built-in

- Done with Jetty plugin: Access control, SSL

Monitoring: many plugins available

- ElasticHQ, BigDesk, Head, Paramedic, ...

Easy to deploy and manage

Robust, fast, and rich API

More features coming with aggregation framework



## Visualize time-stamped data from ElasticSearch

- Designed to analyse log, perfectly fits time stamped data
- No code, point & click to build your own dashboard
- Built with AngularJS (from google)
- Open source, community driven
  - Supported by ElasticSearch
  - Provided code/feature contribution
- Easy to install & configure
  - “git clone” OR “tar -xvzf” OR ElasticSearch plugin
  - 1-line config file to point to the ElasticSearch cluster



Easy to install and configure

Very cool user interface

Fits many use cases (e.g. text, metrics)

Still limited feature set, but active growing community



## Producers

- From all puppet-based data centre nodes
- Infrastructure & Application monitoring

## Flume

- 10 aggregator nodes, 5 nodes to HDFS + 5 nodes to ES

## HDFS

- ~500 TB cluster, 1.8 TB collected since mid July 2013

## ElasticSearch

- 1 master node, 1 search node, 8 data nodes
- 90 days TTL, 10 shards/index, 2 replicas/shards
- Running ElasticSearch Kibana plugin

Lemon 

Kibana 3 milestone 3

Controls 



5m 15m 1h 6h 12h 24h **2d** 7d 30d

**Relative** | Absolute | Since | ☐ Auto-refresh



Dashboard Control





Filters 

field must    
field : @fields.submitter\_cluster  
query : "aimon/flume/gw/dev"

field must    
field : @fields.environment  
query : "production"

field must    
field : @fields.entity  
query : ""

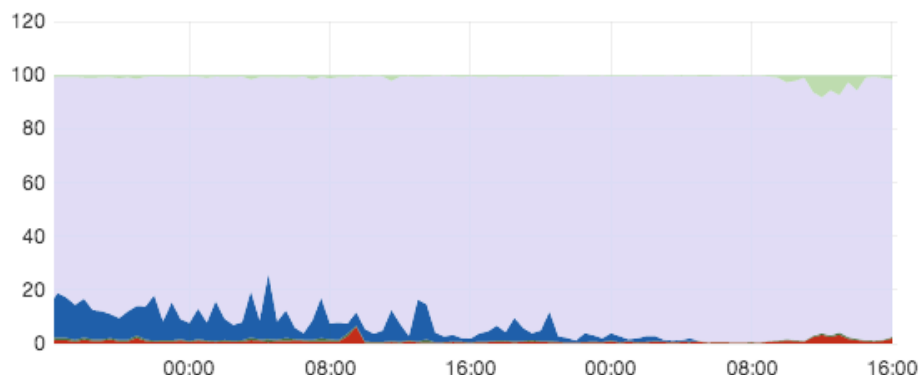
time must    
field : @timestamp  
from : "2013-10-23T14:17:44.804Z"  
to : "2013-10-25T14:17:44.804Z"

Queries 

Graphs 

## CPU Utilization (percentage)

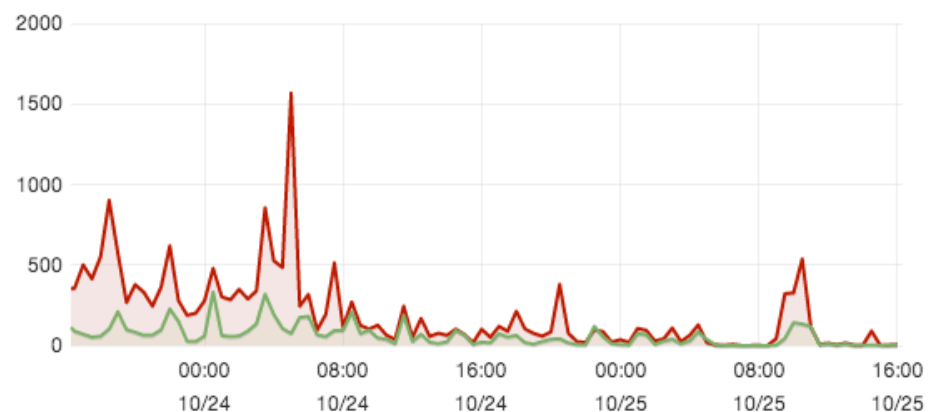
[Zoom Out](#) | ● User (2931269) ● System (2931269) ● Nice (2931269) ● Idle (2931269)  
● IOWait (2931269) ● IRQ (2931269) ● SoftIRQ (2931269) per 30m | (20518883 hits)



## Network Utilization (KB/s)

histogram 

[Zoom Out](#) | ● Input (3055987) ● Output (3055987) per 30m | (6111974 hits)







Every IT service needs monitoring

Similar technologies fit different use cases

Community makes you stronger

- Does not impose solutions
- Leverages the effort of adopting, deploying and running

Examples of what other teams in IT have been doing with these tools

- IT-OIS Cloud Service
- IT-PES Batch service

## CERN Cloud Infrastructure

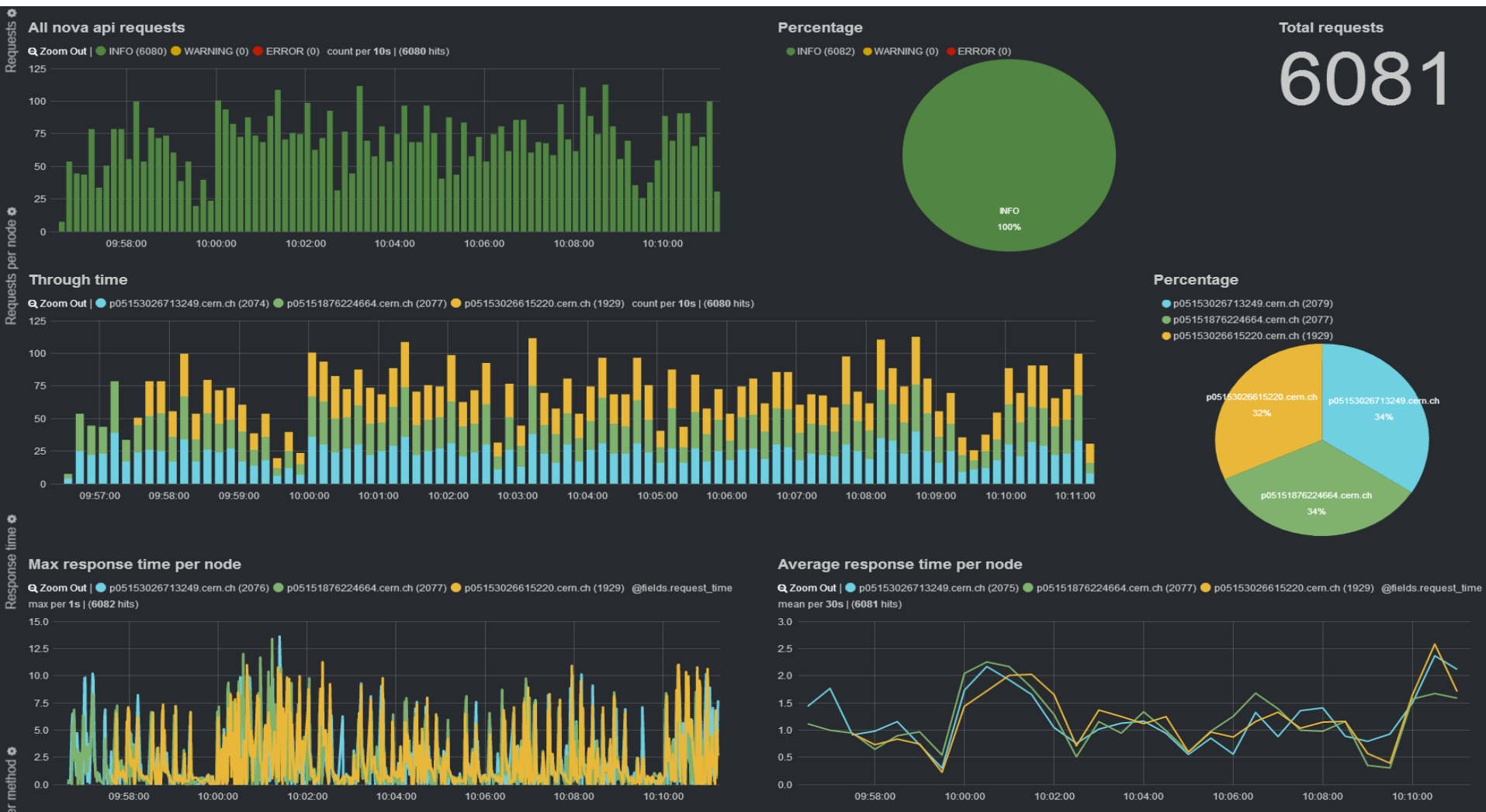
- Based on OpenStack
- Consists of several services (Keystone, Nova, Glance, Cinder, ...)

## Production deployment

- 664 nodes, 20224 cores, ~ 1800 VMs

## Requirements

- Centralized copy of logs for investigation
- Display OpenStack usage statistics and functional tests
- Maintain a long term history of the infrastructure status



## Running LSF

4000 servers with over 50000 CPU cores

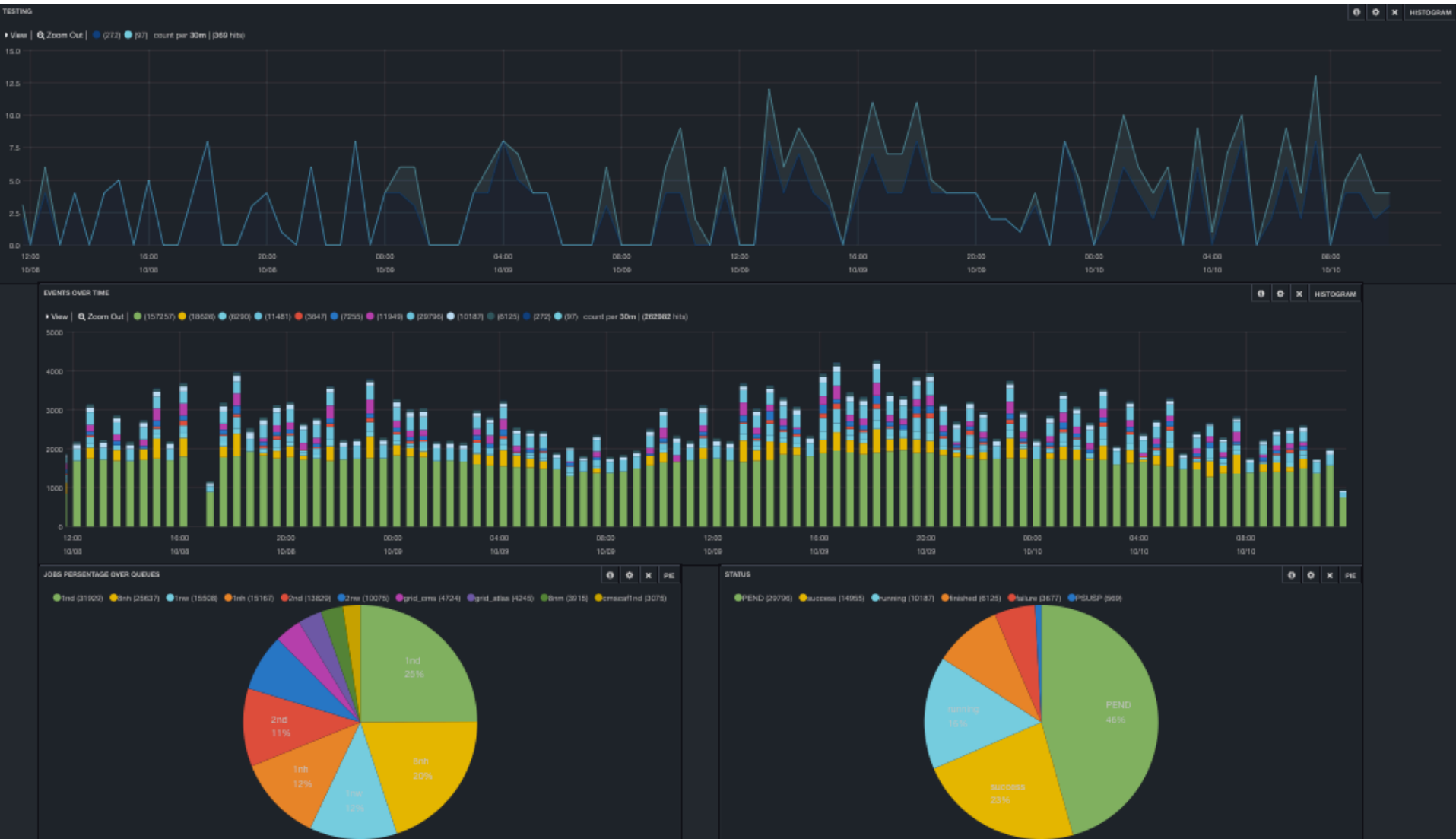
400000 jobs/day

## Requirements

- Tool enabling 2nd line support for end users
- More specific/relevant information display with Kibana
- Kibana dashboards opened to Helpdesk, users
- No need for engineer-level personnel replying to requests



# Example



Several interesting **technologies tested** and deployed

**Full workflow deployed** for concrete monitoring needs

Verified by **different monitoring** producers

Improve monitoring tools under a **common effort**

# Questions??

[itmon-team@cern.ch](mailto:itmon-team@cern.ch)

<http://cern.ch/itmon>