
CERN

(Centralised) Elasticsearch Service

- A bit of History
- Project goals
- Security model
- Cluster management
- Experiences
- Summary

Elasticsearch at CERN in the past

- ES in use for the last years, also in IT (eg for monitoring)
- Many small clusters around
- Privately run, not always very well managed or secured
- Project to upgrade and **consolidate ES instances** launched in Jan 2016
  - Production status planned for Q1 2017

Centralised Elasticsearch Service

Goals:

- Consolidation of various use cases in a single place
  - Spare hardware resources by sharing where possible
- Setup of a reliable service covering as many use cases as possible
- Provide Elasticsearch and Kibana access

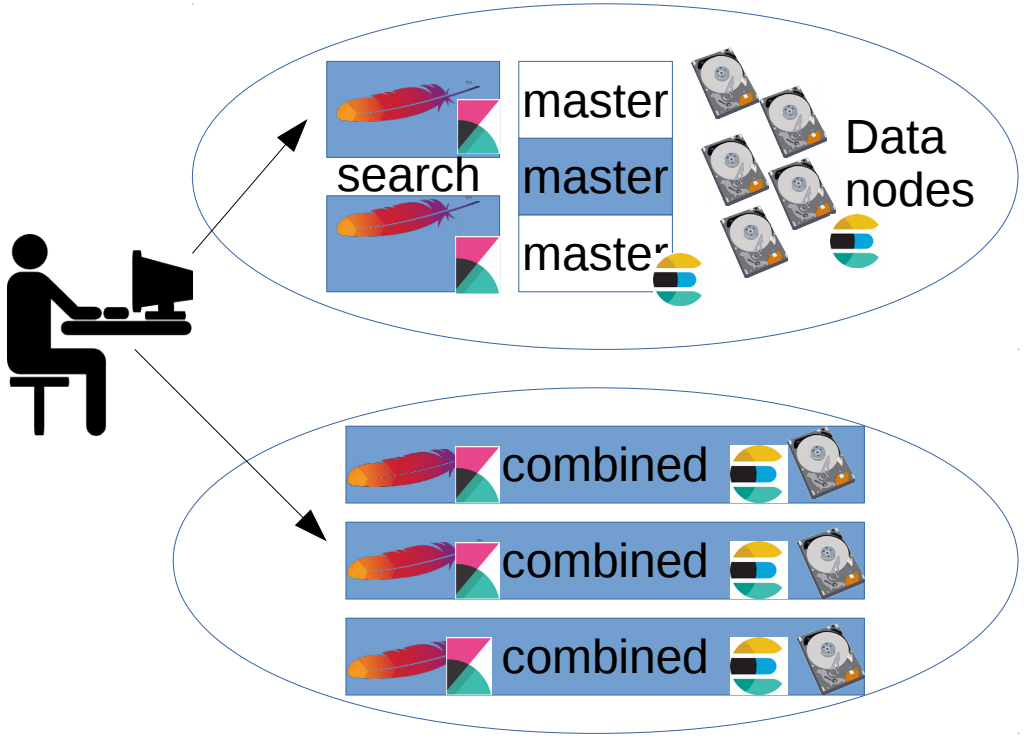
Centralised ES service: management

Architecture:

- Full integration into the existing infrastructure
- Puppet managed and monitored as any other cluster
- Shared and dedicated cluster
  - Preferred way is to go via the central monitoring
  - Offer shared clusters for users with public data
  - Dedicated clusters for use cases with sensitive data
- No commercial plugins (including security)

Node organisation: models in use

- Made to spare resources
- 4 node types:
  - ES Master nodes (3)
  - Search nodes (2+)
    - Http proxy (apache)
    - Kibana services
  - Data nodes (3+)
  - Combined nodes:
    - Master, search and data node
    - For small installations



Current status, numbers

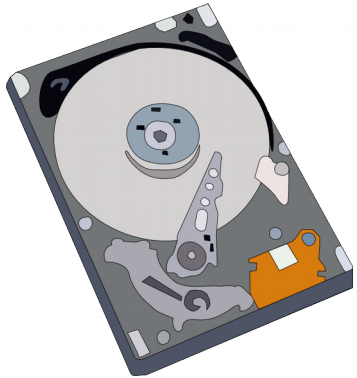
- Currently **21** independent ES clusters, **30** use cases
- Up to **5** use cases on a single cluster
- Elasticsearch versions ranging from **2.3.3** to **5.1.1**
- Kibana versions **4.5.1** to **5.1.1**
  - Plugins and version configurable by cluster
  - Allows to upgrade at different speed, as needed by the users

Hardware

Virtualized hardware

- Higher flexibility and better resource consolidation
- Allows to give smaller nodes to small use cases
- Allows for rapid access and replacement
- **114** VMs for search, master and combined nodes

openstack™
CLOUD SOFTWARE



Work horse for data nodes:

- Virtualised data nodes on special hypervisors
- Spinning disks with SSD cache (b-cache)
- 630GB/60GB RAM per full node
- Use full or half nodes (315GB/30GB RAM), **115** nodes in use

Security model and access

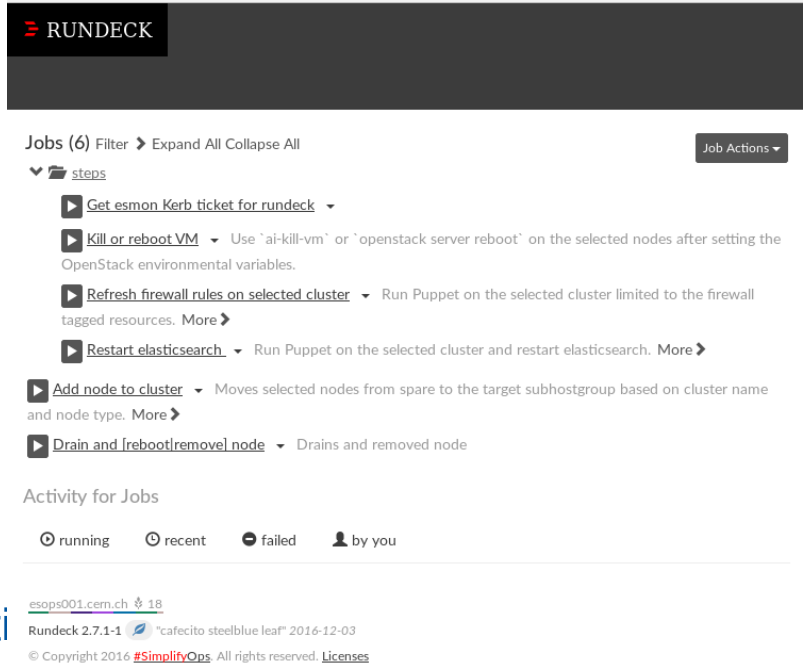
Access SSL only

- 443 Kibana, via CERN SSO
- Possibility to have read-only Kibana access
  (ES 2.X only, via readonlyrest plugin)
- Port 9103 REST access with basic authentication
- Java API is in general closed
- For shared clusters no access to ES plugins (head, …)
- Firewall rules to separate clusters from each other
- Different access pattern rules for different entry points into the same cluster

Cluster management

Automation and monitoring:

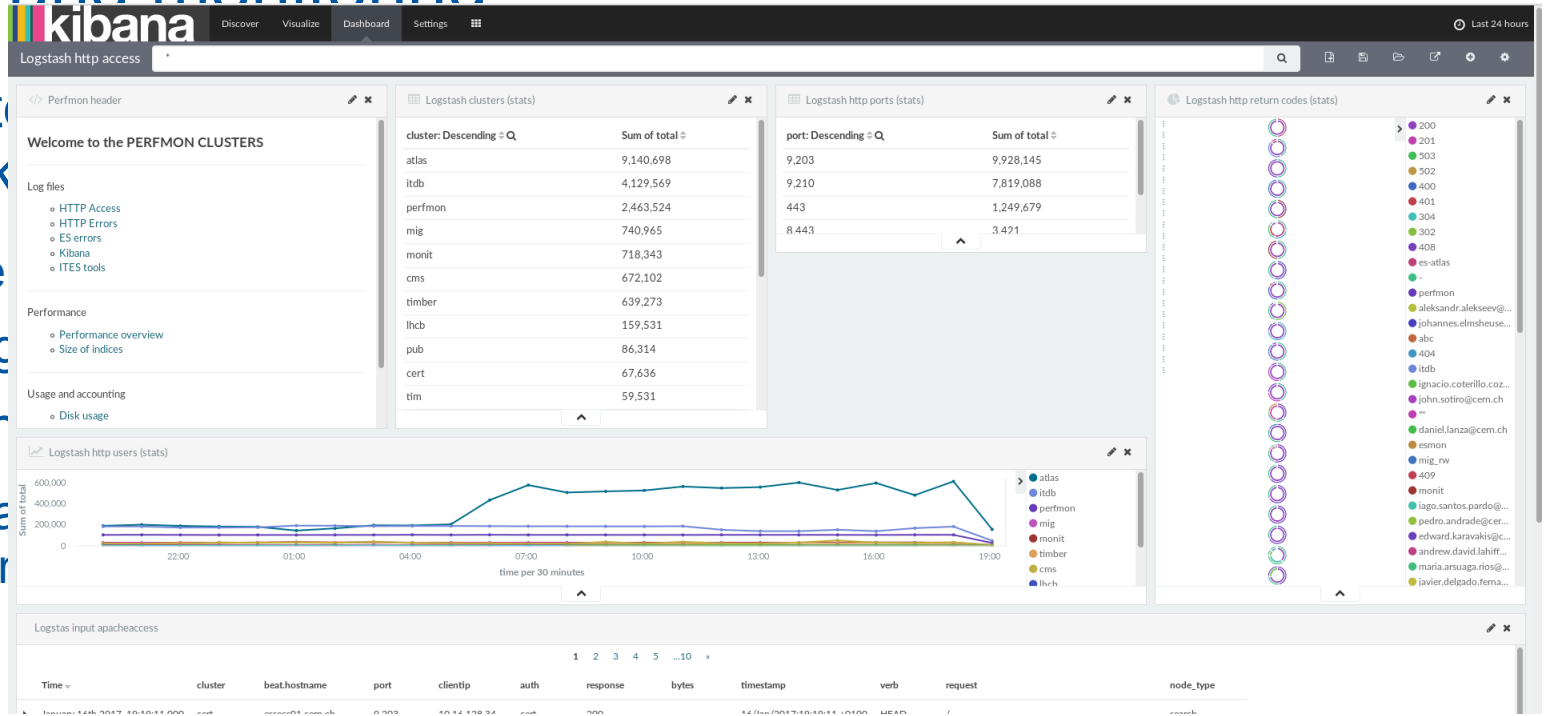
- **Automated** work flows
  - Cluster restart, Kernel upgrades
  - ES updates
  - Adding/removing data nodes
  - ...
- Self-monitoring
  - Kibana dashboard with health and accounti information



Cluster management

Automation and monitoring:

- Automat
  Rundeck
  - Cluste
  - Adding
- Self-mor
  - Kibana
    accour



Automation and

- Automat
  Rundeck
  - Cluste
  - Adding
- Self-mor
  - Kibana
    accour

Experiences so far

- ES is very flexible and dynamic
- User expectations follow that pattern closely ...
- Hard to fit everybody with a single solution
  - Hence split into independent clusters
  - Also for performance reasons
- Resource consolidation tricky
  - Requires good control of ACLs
  - Work in progress

www.cern.ch