Securing and sharing Elasticsearch resources with Read-onlyREST

Ulrich Schwickerath[1,*], *Pablo* Saiz[1], and *Zhechka* Toteva[1]

[1]CERN, 1, Esplanade des Particules, 1211 Geneva 23

Abstract.
In early 2016 CERN IT created a new project to consolidate and centralise Elasticsearch instances across the site, with the aim to offer a production quality new IT services to experiments and departments. We present the solutions we adapted for securing the system using open source only tools, which allows us to consolidate up to 20 different use cases on a single Elasticsearch cluster.

1 Introduction

Before CERN established a centralised Elasticsearch service, many people ran their own, privately managed Elasticsearch instances, resulting in a large use of resources. The idea was born to offer a centralised service to improve this situation.

In the first surveys which were performed during the initial stage of the project, use cases for Elasticsearch turned out to be wide spread and partly conflicting. While some communities insisted on strict data privacy, others asked for public access to their data. In some cases, access to internal networks was required. Other use cases required heavy indexing activity and a huge amount of disk space, while some focus on searching a few but very large indices only. An additional challenge was that some projects were still at an early phase, with unclear final requirements in accessibility and storage.

The new service had to fulfil the requirements and standards for any IT service, being scalable, virtualised, secure, centrally managed, and cost effective. The following sections describe how each of these requirements were addressed.

2 A new scalable Elasticsearch service

This section describes how the challenges related to setting up a centralised service for Elasticsearch were addressed. With the described architecture it was possible to satisfy the bulk of the requirements. Table 1 shows an overview of terms used.

The service consists of a set of independent clusters which can differ in size, hardware, connectivity and accessibility. As an example, two clusters use disk servers as storage backend for a demanding use case which indexes a lot of data. In this specific case searches are rare and not time critical. Other clusters can be accessed from CERN's internal network but not from the internet. All our clusters provide ACLs, some clusters are shared by different communities and host several smaller use cases, others are dedicated to specific applications

*e-mail: Ulrich.Schwickerath@cern.ch

term	rank	explanation
cluster	top level structure	Elasticsearch installation unit. Can be dedicated to a community (eg LHC experiment), eg *https://es-atlas.cern.ch*
entry point	cluster sub-structure	load balanced DNS alias pointing to a specific cluster, eg *https://es-usecase1.cern.ch*
path	path inside an entry point	URL path within an entry point, allowing for different authentication schemes or services, eg */krb* in *https://es-usecase1.cern.ch/krb*

Table 1. Glossary of terms.

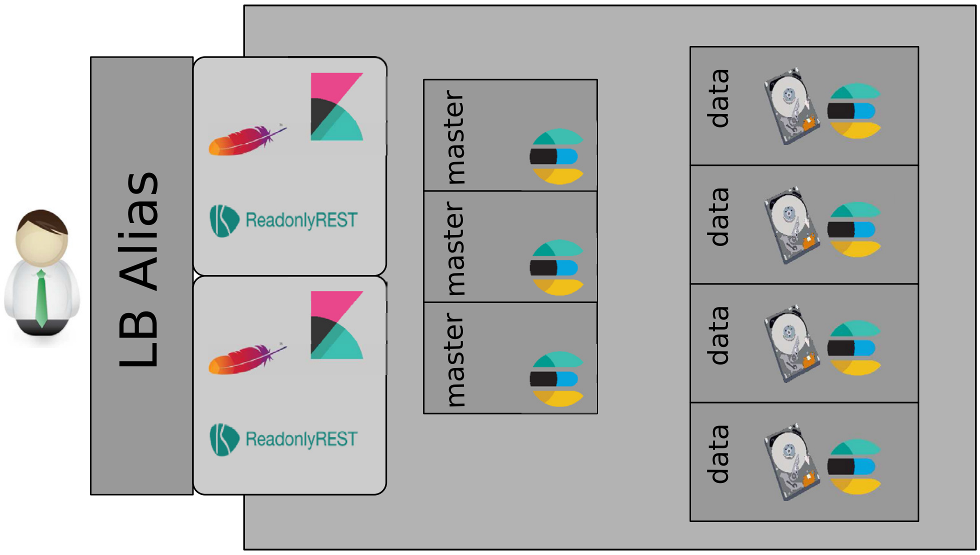

Figure 1. Cluster layout. Users connect through a load balanced alias of at least two search nodes. Apache takes care of the user authentication, and proxies the request to Elasticsearch or Kibana. Each cluster has exactly three master nodes out of which one is active, and at least 2 data nodes.

or communities. Examples there are IT service monitoring, or LHC experiments. All clusters are setup in the same way and managed in the same way. A schematic view of the general setup can be found in Fig. 1, a glossary of the node types can be found in table 2.

Clusters are isolated from each other by appropriate firewall rules between the nodes which are automatically created by Puppet [1]. Each cluster consists of:

- exactly three Elasticsearch master nodes
- at least two data nodes
- at least two search nodes

term	explanation
data node	Elasticsearch node providing storage not accessible by the users
master node	machine acting as Elasticsearch master not accessible by the users
search node	machine acting as Elasticsearch client, running Apache, Elasticsearch and Kibana, Apache accessible by the users with SSL

Table 2. Summary of node types used

All nodes run CERN CentOS7 [2][3]. Search nodes are setup with dual stack network configuration and thus provide Elasticsearch services also on IPv6. Virtual machine flavors are defined in such a way that per CPU about 2 GB of RAM are allocated.

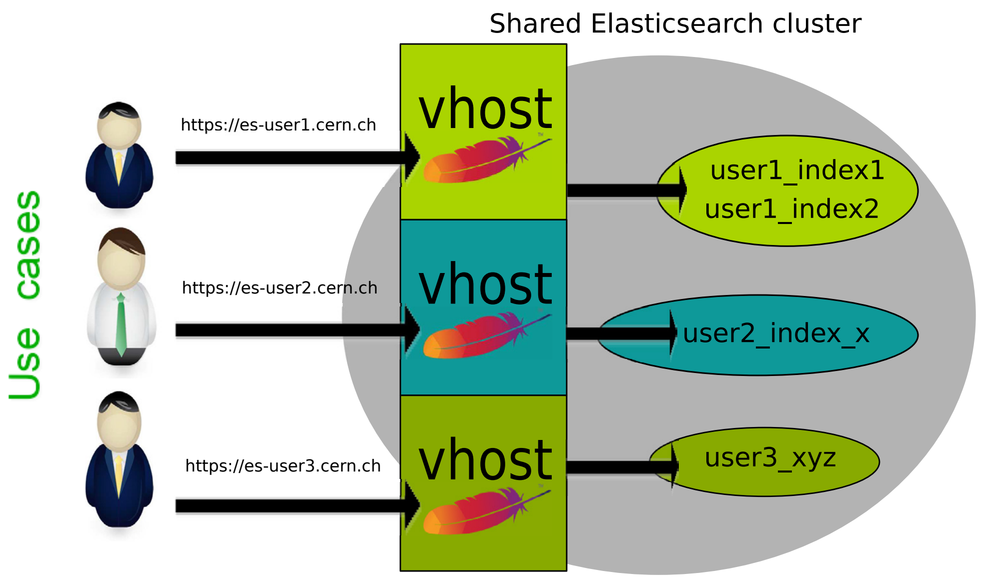

Figure 2. ACL implementation: Users connect through a load balanced alias to the cluster. Apache takes care of user authentication, and redirects the request locally to Elasticsearch. Authorisation is done with ReadonlyRest and is based on index name patterns. The index names typically have to match the alias name. In the shown example, *https://es-user3.cern.ch* can only access indices starting with *user3*.

2.1 Master nodes

In order to avoid a split brain situation, at least 2 masters nodes are required to be available at any time. Master nodes are typically medium sized virtual machines, providing 2 CPU cores. They are required to live in different availability zones. Master nodes do not provide any data storage, and are not accessible by the users.

2.2 Data nodes

Data nodes run on dedicated hypervisors which have been equipped with solid state disk (SSD) storage. These hypervisors provide 128 GB of RAM, 40 CPU cores (HT enabled) and about 4 TB of raw SSD disk space. Since Elasticsearch already takes care of keeping replicas, the disks have been organised in a Raid 0 configuration to maximise the available space for Elasticsearch.

In order to optimise memory usage, each hypervisor runs two data nodes, each providing 20 CPU cores and about 60 GB of memory, out of which 32 GB are give to Elasticsearch, and the rest to the operating system for buffering. About 4 GB are reserved for the hypervisor itself. With this setup, each data node provides about 1.7 TB of storage available for Elasticsearch. For test clusters and other small clusters smaller data nodes are used, in order to optimise the resource usage.

The hypervisors are organised in four different tenants, and the tenants in turn use different up-link switches. Elasticsearch shard placing policies are set up in such a way that replicas are required to be located on a data node in a different tenant.

2.3 Search nodes

For redundancy reasons at least two search nodes are needed per cluster. They are virtualised and located in different availability zones. The search nodes provide users access to the clusters through Apache proxies. All the traffic from the user to the search nodes of any Elasticsearch cluster is secured with SSL. Search nodes act as Elasticsearch clients, use the free and open source version of ReadonlyREST Elasticsearch plugin [4] for ACLs, and provide Kibana [5] visualisation capabilities. As they run both Elasticsearch and Kibana, they need to be larger than the masters. Depending on the use cases virtual machines with four or eight CPUs are used.

3 Centralised cluster management

The service is entirely integrated into CERN's Agile Infrastructure [7][8]. Apache and ReadonlyREST configuration files are automatically created from Hiera [1] based configurations in Puppet. This allows for sharing the base setup while opening the possibility for per cluster or per endpoint customisations, and allows for a light weight configuration of even complex setups.

As mentioned earlier, search and master nodes are spread over three different OpenStack [9] availability zones, and data nodes are spread over four distinct tenants. In addition, at least one copy of the data in Elasticsearch is required by default, which is allocated on a data node in a different tenant by scheduling policies. Apart from improving the redundancy against network outages, this also helps managing the clustes. If ever an intrusive intervention has to be performed on the hypervisors, this can be done by availability zone or tenant. During the intervention the clusters will be degraded. However, at least one of the copies of the data will stay available, which allows users to continue to work.

In addition, intrusive interventions on the service can be paralleled on the cluster level as well. Examples are kernel upgrades or Elasticsearch version updates. These are fully automated using external scripts which ensure that the interventions on the nodes are performed in the optimal order. The clusters stay available to users while being upgraded or rebooted.

Since the monitoring of the Agile infrastructure uses Elasticsearch which is now provided by the described service, it was decided to have a dedicated Elasticsearch cluster for internal service monitoring. No commercial tools are required to perform this task. For dedicated clusters a restricted subset of this monitoring is available for specific users which allows them to trouble shoot their activity on their own.

4 Resource sharing, access control mechanisms and security

Access control is the key for enabling sharing of the resources. It is implemented based on index name patterns using ReadonlyREST [4]. For each use case on a cluster, a load balanced alias is created, see fig. 2. A user connecting to a specific entry point can only access indices which match the name of the alias. This is ensured by setting up appropriate rules for ReadonlyREST. The mechanism allows for a strict separation of users from each other. They cannot even see other users indices.

ReadonlyREST only works on the REST API of Elasticsearch. Therefore, access to the Java API of Elasticsearch is generally blocked.

As the cluster entry points also provide visualisation via Kibana, it has to be ensured that the Kibana index used also differs for each entry point. This is done by using an additional plugin for Kibana, called Kibana Ownhome [6]. The Kibana index name is determined automatically via Apache, depending on the entry point name which the user used to connect to the cluster. This allows to run only one Kibana instance per search node which can be shared by different users. They can create and view their own visualisations and dashboards, without noticing the other users of the same cluster.

A caveat is that access to the development tools console of Kibana cannot be fully restricted. Therefore, it is disabled in the setup.

Setting up the Elasticsearch clusters in the described way allows to share resources between very different user communities, respecting their privacy requirements. The effect of consolidation and adoption of new use cases can be seen in Fig. 3. While the number of supported use cases grew with a rate of about three new use cases per month, the number of required nodes to support them only slowly increased, and the number of clusters decreased due to consolidation of the use cases hosted on those clusters.

As of May 2018 additional resources were deployed which resulted in a significant increase of the number of nodes in the service. In addition, we added support for Elasticsearch 6 by offering replacement capacity, allowing the users to test the new version and move over when they are confident.

5 Special features

The use of Apache proxies opens the door for additional features which are described in this section. They allow to implement different authentication methods without the need of changes to Elasticsearch itself. Using virtual hosts we define different paths which grant access to Elasticsearch using different authentication schemes:

- basic authentication. A user name matching the alias name and a randomised password is created and securely transferred to the user. The default path is $/es$.
- kerberos authentication based on LDAP groups [1]. Two paths are provided per entry point. */krb* provides read only access, while */krb_rw* grants read/write access for the configured groups, respectively.
- for special cases, in addition Single-Sign-On Access (SSO) can be configured. This can be useful if Elasticsearch has to be queried from an external application which already uses SSO. This is typically configured using the path $/sso$

[1] via CERN egroups

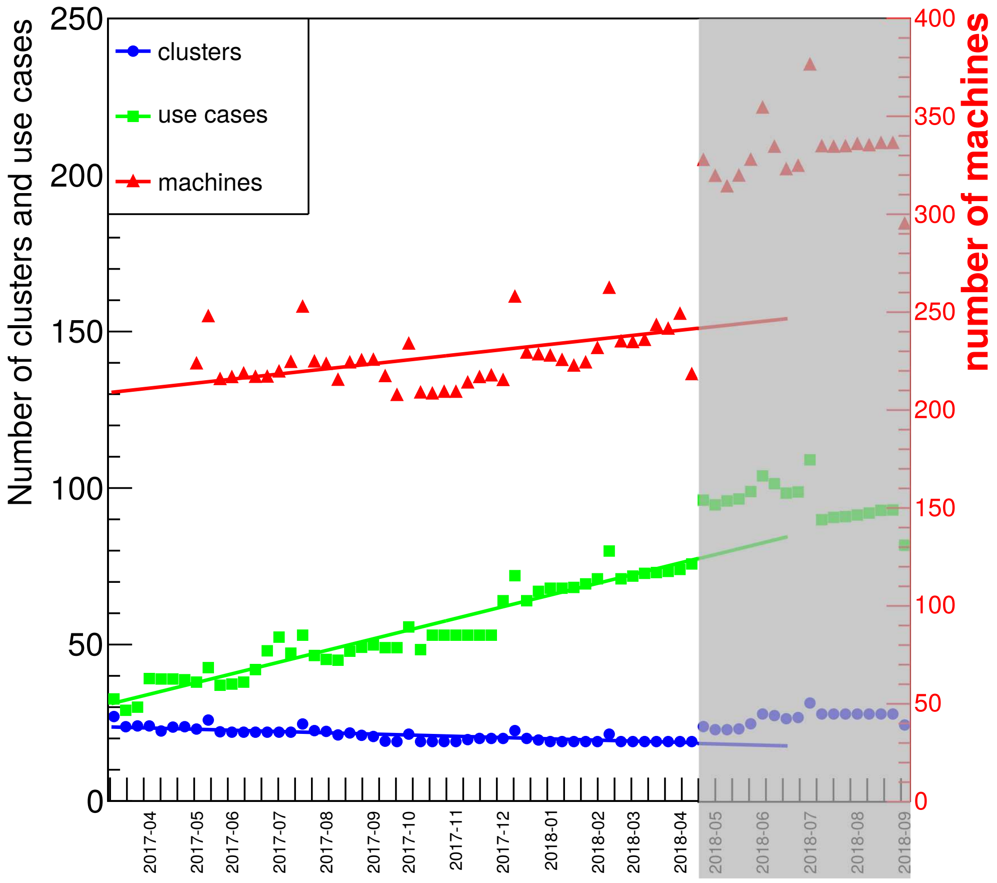

Figure 3. Service evolution between April 2017 and April 2018 when the service consolidation was done. The number of nodes used for the service slowly increased (red line) while we deployed about 3 new use cases per month (green line). At the same time, the number of clusters decreased (blue line). As of May 2018 a significant amount of additional resources were provided to the users, partly to allow them to test a new version of Elasticsearch, Elasticsearch 6 (grey area).

The way the cluster is managed it is easy to add additional paths, either to all clusters in general, individual clusters or even only for specific entry points. Individual paths can be enabled or disabled on request by entry point.

On request, access to Kibana can be restricted using two factor authentication. It is also possible to enable two factor authentication when accessing the service from outside CERN while using simple SSO from inside the CERN network. This feature is used for the internal monitoring and allows service managers to trouble shoot the service from home outside working hours.

6 Future work

Document level security is currently not implemented yet. Recent version of ReadonlyREST allow for this feature though, and it is planned to offer this feature to the users in the future if they have a valid use case for it.

7 Summary

During the past few years a new centralised service providing access to Elasticsearch and Kibana for visualisation has been setup and moved into production. Making use of only freely available software and virtualisation, it was possible to consolidate a up to 20 different use cases on a single Elasticsearch cluster, and support up to 100 use cases on the service with a moderate to small increase of resources.

References

[1] Puppet Labs: IT Automation Software for System Administrators [software], version 4.9.4, 2018. Available from *http://puppetlabs.com* [accessed 2019-02-02]

[2] CentOS Linux distribution [software], version 7.5, 2018. Available from *https://www.centos.org* [accessed 2018-10-02];

[3] CentOS7 with CERN extensions [software], version 7.5, 2018. Available from *http://linux.web.cern.ch/linux/centos7* [accessed 2018-10-02]

[4] ReadonlyREST: Security for Elasticsearch and Kibana [software], version 2.16.20. Available from *https://github.com/sscarduzio/elasticsearch-readonlyrest-plugin* [accessed 2018-10-02]

[5] Kibana: Your Window into the Elastic Stack [software], versions 5.5.2, 5.6.9, 6.2.4. Available from *https://www.elastic.co/products/kibana* [accessed 2019-02-02]

[6] Kibana Own Home plugin by Wataru Takase [software], versions v5.5.2, v5.6.9, v6.2.4, Available from *https://github.com/wtakase/kibana-own-home* [accessed 2019-02-02]

[7] Review of CERN Data Centre Infrastructure, P Andrade et al 2012 J. Phys.: Conf. Ser. **396** 042002

[8] CERN's Agile infrastructure, B Jones et al 2015, J. Phys.: Conf. Ser.**664** 022026

[9] OpenStack Open Source Cloud Computing Software [software], version Ocata. Available from *https:/s/openstack.org* [accessed 2019-02-02]