**Elasticsearch Upgrade Plan Note**

--

I looked at Apoorva's runbook, and it is an Elasticsearch x-pack-based setup. If we use Elasticsearch x-pack v7.11 or higher, we can use it for only one month so will need a license. if you are using the Elasticsearch commercial license, then there are some things that you need to be aware of. Firstly, it is important to note that Elasticsearch can stop working if your license expires. All data operations such as read and write can stop. If you use it without a license, all use will be stopped after one month.

So, we will have to use search guard as open-source security plugin instead of x-pack, and the es version will use a search guard-supported version. (Currently 8.17.0)

Search Guard is an Open-Source security plugin for Elasticsearch, Kibana and the entire ELK stack.

--

--

Considering the future support date, I would like to use JDK version 17 or 21. I wonder which of these two versions can support Kafka middleware. I also verified two versions in Elasticsearch/Logstash version. Both versions are possible.

--

--

As for version 8.12.2, I installed ES v.8.12.2 and search guard on Dev(https://tsgvm00877:9201) last year. I also used the tool to generate certs and make it work. That's why I said 8.12.2. Currently, Elasticsearch provides up to version 8.17.1(https://www.elastic.co/downloads/past-releases#elasticsearch), but search guard, a security open source, provides up to version 8.17.0 (https://docs.search-guard.com/latest/search-guard-versions). If you want version 8.17.0, I will try to set up with this version at this time if this version of search guard with ES v.8.17.0 works well. The search guard version has been updated in a few months.

--

--

Once the new ES to be upgraded is setup, you need to perform the index creation and data copy steps. The first is to create an index in the target cluster. I implemented a script that creates an index with mappings from the source ES cluster and also changes the mappings to ES v.8. Additionally, the script also updates the alias values ​​for each index simultaneously to the target cluster. If there are any errors, I will fix them at this time.

After index creation is complete, the second step is to check the mapping for each index within the target cluster. At this time, I will use the mapping comparison script to check the mappings between the source cluster and the target cluster. I will also check for any errors in the mapping comparison script and check if it is working properly.

The third step is data copying. I will also use the reindexing script to copy all WMx/OMx related indexes in the source cluster to the target cluster. Afterwards, I plan to compare the counts.

I will also start time of reindexing to allow ES team to properly re-queue ES queue data when running the reindexing script.

--

--

When I was reindexing with a python script in Dev, I used the basic auth value in the Header of the HTTPS URL. Basic authentication is a very simple authentication scheme that is built into the HTTP protocol. The client sends HTTP requests with the Authorization header that contains the Basic word followed by a space and a base64-encoded username:password string (: -> colon).

In the ssl certificate step, search guard provides a tool to create node certificate, rest api certificate, etc. In the dev environment, I created a ssl certificate through this. That's why the login window appears when we access es in Dev.

--