

Upgrading Elasticsearch from Version 5.x to 6.x (Linux)

Summary

Elasticsearch is a required component of ALM Octane. This document is provided as a service and provides instructions on upgrading Elasticsearch on Centos version 7.

Topic

This article provides the steps for upgrading the Elasticsearch database from version 5.x to 6.x. Elasticsearch is a required database component for ALM Octane. Elasticsearch version 6.x is required for ALM Octane 12.60.47 and later.

Overview

The upgrade is divided into the following stages:

- **Before upgrading**
Delete multi-type indices.
Delete indices of version older than 5.x.
- **Upgrade**
Perform the upgrade.

Before upgrading

1. Back up your data

You must back up your Elasticsearch before dropping any index.

```
curl -XPUT
"http://<hostname>:9200/_snapshot/<my_repo>/<my_snapshot>?wait_for_completion=true" -H 'Content-Type: application/json' -d'
{
  "indices": "mqm*",
  "ignore_unavailable": true,
  "include_global_state": false
}'
```

2. Drop multi-type indices

```
curl -XGET http://<hostname>:9200/_cat/indices/mqm* | grep close | awk
'{print $2}' > indices.lst
cat indices.lst | awk '{print "curl -XDELETE http://<hostname>:9200/"$1}' >
delete_ix.sh
chmod +x delete_ix.sh
./delete_ix.sh
```

3. Drop other indices with versions lower than 5.0

```
curl http://<hostname>:9200/ settings/index.version.created?pretty=true |awk
'NR%9==2||NR%9==6'|awk 'NR%2 {printf "%s ",$0;next;}1'|tr -d '{}":,'|awk
'$3<5000000 && $3 {printf "curl -XDELETE http://<hostname>:9200/%s\n",$1}' >
delete_old.sh
chmod +x delete_old.sh
./delete_old.sh
```

4. Back up configuration files

```
cp -R /etc/elasticsearch /tmp
```

5. Check OS definitions

Check that the limits are set correctly.

- **Limits.conf**

```
vi /etc/security/limits.conf
elasticsearch soft memlock unlimited
elasticsearch hard memlock unlimited
elasticsearch soft nproc 4096
elasticsearch hard nproc 4096
elasticsearch - nofile 65536
```

- **java**

Elasticsearch requires at least Java 8

```
java -version
```

6. Back up ELS indices

Take a snapshot backup of the Elasticsearch server indices: `mqm_*`

7. Download ELS installation

```
cd /tmp
```

```
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.8.1.rpm
```

8. Stop ALM Octane servers

Windows installations:

Stop the ALM Octane service.

Linux installations:

```
service octane stop
```

9. Disable shard allocation before shutting the node

```
curl -XPUT '{hostname}:9200/_cluster/settings?pretty' -H 'Content-Type: application/json' -d '{
  "persistent": {
    "cluster.routing.allocation.enable": "none"
  }
}'
```

10. Perform a synced flush and stop indexing

```
curl -XPOST '{hostname}:9200/_flush/synced?pretty'
```

A synced flush request is a “best effort” operation. It will fail if there are any pending indexing operations, *but it is safe to reissue the request multiple times if necessary.*

11. Shut down all cluster nodes

```
systemctl stop elasticsearch.service
```

Upgrade

1. Uninstall plugins

```
cd /usr/share/elasticsearch/bin
./elasticsearch-plugin list
./elasticsearch-plugin remove <plugin name>
```

2. Upgrade all nodes

Use the rpm downloaded earlier in “Before upgrading” to install the new package.

```
rpm -U /tmp/elasticsearch-6.8.1.rpm
```

3. Edit /etc/elasticsearch/jvm.options

```
vi /etc/elasticsearch/jvm.options
```

Change from	Change to
-Xms<XXX>g	Change the XXX to half of the memory available on the machine minus 1.
-Xmx<XXX>g	Change the XXX to half of the memory available on the machine minus 1.
-XX:+UseConcMarkSweepGC	-XX:+UseG1GC

4. Edit elasticsearch.yml

Verify the following Elasticsearch parameters.

Note that **elasticsearch.yml** is usually located in the **/etc/elasticsearch** folder.

```
vi /etc/elasticsearch/elasticsearch.yml
```

```
cluster.name: <Your cluster name>
node.name: <Your node name>
network.host: <your network interface name>

###Path parameters
path.data: /<DATA path>/elasticsearch
path.logs: /LOGS path
path.repo: ["/REPOSITORY path"]

bootstrap.memory_lock: true

# In case you 3 nodes in the cluster
discovery.zen.ping_timeout: 10s
discovery.zen.minimum_master_nodes: 2
gateway.recover_after_nodes: 2
gateway.expected_nodes: 3

##Disable XPack unlicensed features
xpack.security.enabled: false

node.ml: false
xpack.ml.enabled: false
```

5. Enable Memory lock

Validate that memory lock is enabled in the service:

```
systemctl edit elasticsearch
```

```
[Service]
LimitMEMLOCK=infinity
```

6. Start the cluster

If you have dedicated master nodes, start them first. Dedicated master nodes are nodes with **node.master** set to **true** (default) and **node.data** set to **false**.

The number of nodes started should be at least the number defined in **elasticsearch.yml**:

```
discovery.zen.minimum_master_nodes: xxx
```

If not, the recovery will not start until the number of nodes reaches the number defined in **discovery.zen.minimum_master_nodes**.

To start Elasticsearch, run the following on **all cluster nodes**:

```
systemctl daemon-reload
systemctl start elasticsearch
```

Check that nodes respond:

```
curl -XGET 'http://{hostname}:9200/_cat/health?pretty'
curl -XGET 'http://{hostname}:9200/_cat/recovery?pretty'
```

7. Wait for yellow

Run the commands below and wait until all primary shards have been recovered, but not all replica shards are allocated. This is to be expected because allocation is still disabled.

```
curl -XGET '{hostname}:9200/_cat/health?pretty'
curl -XGET '{hostname}:9200/_cat/recovery?pretty'
```

8. Re-enable allocation

When all nodes have joined the cluster and the cluster is yellow, re-enable the allocation.

```
curl -XPUT '{hostname}:9200/_cluster/settings?pretty' -H 'Content-Type:
application/json' -d'
{
  "persistent": {
    "cluster.routing.allocation.enable": "all"
  }
}'
```

9. Upgrade internal indices

Run the upgrade:

```
curl -XPOST "http://<hostname>:9200/_xpack/migration/upgrade/.watches"
curl -XPOST "http://<hostname>:9200/_xpack/migration/upgrade/.triggered_watches"
```

10. Back up ELS indexes

Take a snapshot backup of the Elasticsearch server indexes after upgrading.

```
curl -XPUT
"http://<hostname>:9200/_snapshot/<my_repo>/<my_snapshot>?wait_for_completion=
true" -H 'Content-Type: application/json' -d'
{
  "indices": "mqm*",
  "ignore_unavailable": true,
```

```
"include_global_state": false
}'
```

11. Wait for green (when you have more than one node)

All primary and replica shards are successfully allocated.

```
curl -XGET '{hostname}:9200/_cat/health?pretty'
```

```
curl -XGET '{hostname}:9200/_cat/recovery?pretty'
```