Omgamganapathayenamaha

Omgamganapathayenamaha

Omgamganapathayenamaha

Omgamganapathayenamaha

Omgamganapathayenamaha

1. Elasticsearch is a distributed search engine
2. ES Cluster is a group of nodes
3. ES Cluster is a group of indices
4. Index/\_type\_doc/doc
5. Each Index is a group of shards and each shard is a group of documents (primary shards and replica shards)
6. One shard is equal to one lucene index and it has maximum limit to hold documents (limit is 2,147,483,519 (= Integer.MAX\_VALUE - 128))
7. Sharding is important for two primary reasons:
   1. It allows you to horizontally split/scale your content volume
   2. It allows you to distribute and parallelize operations across shards (potentially on multiple nodes) thus increasing performance/throughput

Points about JDK8 and JDK11

1. JDK11
   1. Elasticsearch 6.5.0 works with JDK11
   2. Kibana 6.5.0 works with JDK11
   3. Logstash 6.5.0 will not workwith JDK11
2. JDK8
   1. Elasticsearch 6.5.0 works with JDK8
   2. Kibana 6.5.0 works with JDK8
   3. Logstash 6.5.0 works with JDK8

export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64            (setting JAVA\_HOME to JDK11)

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64               (setting JAVA\_HOME to JDK8)

Installing JDK 8 and setting it up (Source way. There are other ways too)

tar –zxvf jdk-8u161-linux-x64.tar.gz

cd jdk1.8.0\_161

sudo update-alternatives --install /usr/bin/java java /homes/kvsr/jdk/jdk1.8.0\_161/bin/java 100

sudo update-alternatives --config java

sudo update-alternatives --install /usr/bin/javac javac /homes/kvsr/jdk/jdk1.8.0\_161/bin/javac 100

sudo update-alternatives --config javac

sudo update-alternatives --install /usr/bin/jar jar /homes/kvsr/jdk/jdk1.8.0\_161/bin/jar 100

sudo update-alternatives --config jar

export JAVA\_HOME=/homes/kvsr/jdk/jdk1.8.0\_161 (in ~/.bashrc)

setenv JAVA\_HOME /homes/kvsr/jdk/jdk1.8.0\_161 (in ~/.cshrc)

source ~/.bashrc

source ~/.cshrc

Before Setting up Elasticsearch Server

Increasing Memory Map Count

1. Elasticsearch needs memory map count (mmap count) more than or equal to 262144. OS by default gives only 65530
2. So, login as root and increase the mmap count as below:

    sysctl -w vm.max\_map\_count=262144

1. Check the mmap count a below:

                    sysctl vm.max\_map\_count

1. Alternatively, you can set this value permanently, update the vm.max\_map\_count setting in /etc/sysctl.conf. To verify after rebooting, run sysctl vm.max\_map\_count

Increasing Heap Space

1. Open config/jvm.options and edit as below:

# Xms represents the initial size of total heap space

# Xmx represents the maximum size of total heap space

-Xms30g

-Xmx30g

Locking Elasticsearch Heap Space in Memory (i.e., No swapping of this Elasticsearch Heap)

1. In config/elasticsearch.yml, enable

bootstrap.memory\_lock: true

1. You have to give permission to ‘regress’ user to lock the memory by adding the below at /etc/security/limits.conf

regress soft memlock 34359738368           # Specified value is 32G in bytes

regress hard memlock 34359738368         # Specified value is 32G in bytes

1. For ‘regress’ user, increase the open file count from default 1024 to 65536 as below:

regress - nofile 65536

1. You have to logout and re-login for the above limits to take effect
2. You can check the revised values by using “sysctl –all”

Creating more nodes on same machine (if you have more than 64GB RAM)

1. For each extra node, you want to create, please create separate config directory, data directory and logs directory

[regress@mx86-reg-blr9:~/elastic/elasticsearch-6.5.0$](mailto:regress@mx86-reg-blr9:~/elastic/elasticsearch-6.5.0$) mkdir config\_node1 logs\_node1 data\_node1

[regress@mx86-reg-blr9:~/elastic/elasticsearch-6.5.0$](mailto:regress@mx86-reg-blr9:~/elastic/elasticsearch-6.5.0$) cp config/\* config\_node1/

[regress@mx86-reg-blr9:~/elastic/elasticsearch-6.5.0$](mailto:regress@mx86-reg-blr9:~/elastic/elasticsearch-6.5.0$)

1. Set the below in every node’s elasticsearch.yml

cluster.name: bng-elnath-01

node.name: node1

path.data: /home/regress/elastic/elasticsearch-6.5.0/data\_node1

path.logs: /home/regress/elastic/elasticsearch-6.5.0/logs\_node1

http.port: 9201

1. Start the elasticsearch on all nodes as below:                       (Recommend to start each node in separate Terminal)

./bin/elasticsearch &      [ node0 - default ]

export ES\_PATH\_CONF=/home/regress/elastic/elasticsearch-6.5.0/config\_node1

./bin/elasticsearch &      [ other than node0 ]

Setting up Elasticsearch Server

                Make sure ‘java –version’ is working

                             qnc-jdi-regression:~# java -version

java version "1.8.0\_161"

Java(TM) SE Runtime Environment (build 1.8.0\_161-b12)

Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)

qnc-jdi-regression:~#

                Make sure ‘JAVA\_HOME’ is working

                             qnc-jdi-regression:~# echo $JAVA\_HOME/

/homes/kvsr/jdk/jdk1.8.0\_161/

qnc-jdi-regression:~#

                tar -zxvf elasticsearch-6.2.0.tar.gz

                cd elasticsearch-6.2.0

                vi config/elasticsearch.yml

                                network.host: 0.0.0.0

                                transport.host: localhost

                                network.bind\_host: 10.47.227.191

                ./bin/elasticsearch &

               [kvsr@qnc-ng-crm-vm02 ~/elastic/elasticsearch-6.2.0] ps auxxx | grep elastic

kvsr      3176  5.1 16.2 4674928 1331412 pts/0 Sl   04:40   0:18 /homes/kvsr/jdk/jdk1.8.0\_161/bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -XX:+UseCMSInitiatingOccupancyOnly -XX:+AlwaysPreTouch -Xss1m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djna.nosys=true -XX:-OmitStackTraceInFastThrow -Dio.netty.noUnsafe=true -Dio.netty.noKeySetOptimization=true -Dio.netty.recycler.maxCapacityPerThread=0 -Dlog4j.shutdownHookEnabled=false -Dlog4j2.disable.jmx=true -Djava.io.tmpdir=/tmp/elasticsearch.Wrvshgr5 -XX:+HeapDumpOnOutOfMemoryError -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintTenuringDistribution -XX:+PrintGCApplicationStoppedTime -Xloggc:logs/gc.log -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=32 -XX:GCLogFileSize=64m -Des.path.home=/homes/kvsr/elastic/elasticsearch-6.2.0 -Des.path.conf=/homes/kvsr/elastic/elasticsearch-6.2.0/config -cp /homes/kvsr/elastic/elasticsearch-6.2.0/lib/\* org.elasticsearch.bootstrap.Elasticsearch

kvsr      3318  0.0  0.0  12948   944 pts/0    S+   04:46   0:00 grep elastic

[kvsr@qnc-ng-crm-vm02 ~/elastic/elasticsearch-6.2.0]

                NOTE: Now, you can access the elasticsearch server on the default http port 9200 as (<http://qnc-ng-crm-vm02:9200/>)

Setting up Kibana

                Make sure ‘java –version’ is working

                                qnc-jdi-regression:~# java -version

java version "1.8.0\_161"

Java(TM) SE Runtime Environment (build 1.8.0\_161-b12)

Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)

qnc-jdi-regression:~#

                Make sure ‘JAVA\_HOME’ is working

                                qnc-jdi-regression:~# echo $JAVA\_HOME/

/homes/kvsr/jdk/jdk1.8.0\_161/

qnc-jdi-regression:~#

tar –zxf kibana-6.2.0-linux-x86\_64.tar.gz

cd kibana-6.2.0-linux-x86\_64

vi config/kibana.yml

                server.host: 10.47.227.191           # To allow connections from remote users, set this parameter to a non-loopback address

                elasticsearch.url: <http://10.47.227.191:9200>

[root@qnc-ng-crm-vm02:~/kvsr/elastic/kibana-6.2.0-linux-x86\_64#](mailto:root@qnc-ng-crm-vm02:~/kvsr/elastic/kibana-6.2.0-linux-x86_64) ./bin/kibana &

[1] 12467

[root@qnc-ng-crm-vm02:~/kvsr/elastic/kibana-6.2.0-linux-x86\_64#](mailto:root@qnc-ng-crm-vm02:~/kvsr/elastic/kibana-6.2.0-linux-x86_64)

NOTE: Now, you can access the Kibana on the port 5601 as (<http://qnc-ng-crm-vm02:5601>)

Kibana Console

1. As soon as you open the Kibana page, click on “Dev Tools” link on the left side bar
2. That will open the “Kibana Console” for you

Setting up logstash (logstash is a Data ingestion Tool for Elasticsearch)

                Make sure ‘java –version’ is working

                             qnc-jdi-regression:~# java -version

java version "1.8.0\_161"

Java(TM) SE Runtime Environment (build 1.8.0\_161-b12)

Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)

qnc-jdi-regression:~#

                Make sure ‘JAVA\_HOME’ is working

                             qnc-jdi-regression:~# echo $JAVA\_HOME/

/homes/krishnar/kvsr/jdk/jdk1.8.0\_161/

qnc-jdi-regression:~#

                tar -xzf logstash-6.2.1.tar.gz

                cd logstash-6.2.1

                ./bin/logstash -e 'input { stdin { } } output { stdout {} }'      (This will bring up logstash as well as allows us to test our Logstash installation by running the most basic Logstash pipeline)

Sending Logstash's logs to /root/kvsr/elastic/logstash-6.2.1/logs which is now configured via log4j2.properties

[2018-02-17T08:33:55,890][INFO ][logstash.modules.scaffold] Initializing module {:module\_name=>"netflow", :directory=>"/root/kvsr/elastic/logstash-6.2.1/modules/netflow/configuration"}

[2018-02-17T08:33:55,905][INFO ][logstash.modules.scaffold] Initializing module {:module\_name=>"fb\_apache", :directory=>"/root/kvsr/elastic/logstash-6.2.1/modules/fb\_apache/configuration"}

[2018-02-17T08:33:55,972][INFO ][logstash.setting.writabledirectory] Creating directory {:setting=>"path.queue", :path=>"/root/kvsr/elastic/logstash-6.2.1/data/queue"}

[2018-02-17T08:33:55,976][INFO ][logstash.setting.writabledirectory] Creating directory {:setting=>"path.dead\_letter\_queue", :path=>"/root/kvsr/elastic/logstash-6.2.1/data/dead\_letter\_queue"}

[2018-02-17T08:33:56,385][WARN ][logstash.config.source.multilocal] Ignoring the 'pipelines.yml' file because modules or command line options are specified

[2018-02-17T08:33:56,477][INFO ][logstash.agent           ] No persistent UUID file found. Generating new UUID {:uuid=>"87acef59-71ea-454a-9da9-bec0deeb8488", :path=>"/root/kvsr/elastic/logstash-6.2.1/data/uuid"}

[2018-02-17T08:33:56,980][INFO ][logstash.runner          ] Starting Logstash {"logstash.version"=>"6.2.1"}

[2018-02-17T08:33:57,351][INFO ][logstash.agent           ] Successfully started Logstash API endpoint {:port=>9600}

[2018-02-17T08:33:58,552][INFO ][logstash.pipeline        ] Starting pipeline {:pipeline\_id=>"main", "pipeline.workers"=>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50}

[2018-02-17T08:33:58,673][INFO ][logstash.pipeline        ] Pipeline started succesfully {:pipeline\_id=>"main", :thread=>"#<Thread:0x2a5784c0 run>"}

The stdin plugin is now waiting for input:

[2018-02-17T08:33:58,778][INFO ][logstash.agent           ] Pipelines running {:count=>1, :pipelines=>["main"]}

hello world

2018-02-17T16:38:19.404Z qnc-jdi-regression hello world

                ^ctrl-D

                Creating a pipeline with beats as input (Filebeat should be setup before this)

                                vi first-pipeline.conf

input {

    beats {

        port => "5044"

    }

}

output {

    stdout { codec => rubydebug }

}

                                bin/logstash -f first-pipeline.conf --config.test\_and\_exit

bin/logstash -f first-pipeline.conf --config.reload.automatic

[root@qnc-ng-crm-vm01:~/kvsr/elastic/logstash-6.2.1#](mailto:root@qnc-ng-crm-vm01:~/kvsr/elastic/logstash-6.2.1) ./bin/logstash -f logstash.conf --config.reload.automatic &

[1] 11063

[root@qnc-ng-crm-vm01:~/kvsr/elastic/logstash-6.2.1#](mailto:root@qnc-ng-crm-vm01:~/kvsr/elastic/logstash-6.2.1)

[root@qnc-ng-crm-vm01:~/kvsr/elastic/logstash-6.2.1#](mailto:root@qnc-ng-crm-vm01:~/kvsr/elastic/logstash-6.2.1) netstat -na | grep 10514

udp        0      0 10.47.227.190:10514     0.0.0.0:\*

[root@qnc-ng-crm-vm01:~/kvsr/elastic/logstash-6.2.1#](mailto:root@qnc-ng-crm-vm01:~/kvsr/elastic/logstash-6.2.1)

Creating Snapshots

1. We can create Snapshots for entire cluster (or) for some indices in the cluster
2. Before creating snapshots, we have to register a repository with elasticsearch as below:

#Creating a Repository

PUT /\_snapshot/es\_indices\_backup

{

    "type": "fs",

    "settings": {

        "location": "/es\_snapshots/6.5.0",

        "compress": true

    }

}

1. To retrieve the information about a repository, we can use the below command:

GET /\_snapshot/es\_indices\_backup

1. We can create a snapshot for an index (comma separated indices can be provided as well) as below:

#Creating a Snapshot for one index

PUT /\_snapshot/es\_indices\_backup/%3Csnapshot-%7Bnow%2Fd%7D%3E

{

  "indices": "logstash-2019.01.09",

  "ignore\_unavailable": true,

  "include\_global\_state": false,

  "partial": true

}

1. To retrieve information about a snapshot, we can use the below command:

GET /\_snapshot/es\_indices\_backup/snapshot-2019.01.09

1. We can delete a snapshot as below:

DELETE /\_snapshot/es\_indices\_backup/snapshot-2019.01.09

1. We can unregister a repository from elasticsearch as below:

DELETE /\_snapshot/es\_indices\_backup

Disk Space related Parameters

                cluster.routing.allocation.disk.watermark.low   (Default: 85%)

1. When 85% of Disk Space is consumed, ES will not allocate shards to nodes
2. This setting has no effect on the primary shards of newly-created indices or, specifically, any shards that have never previously been allocated

                cluster.routing.allocation.disk.watermark.high  (Default: 90%)

1. When 90% of Disk Space is consumed, ES will attempt to relocate the shards to another node
2. This setting affects the allocation of all shards, whether previously allocated or not

                cluster.routing.allocation.disk.watermark.flood\_stage (Default: 95%)

1. When 95% of Disk Space is consumed, all indexes will become read-only
2. Manually, indexes have to be released from read-only blockage as below after Disk Space is brought to limits:

PUT /logstash-\*/\_settings

{

  "index.blocks.read\_only\_allow\_delete": null

}

Increasing/Decreasing Shard Replicas

PUT /logstash-\*/\_settings

{

    "index" : {

        "number\_of\_replicas" : 0

    }

}

1. When you make the “number of replicas” to ZERO, unassigned replica shards will get deleted (This is a Trick to remove unassigned shards)
2. Later, you can increase the “number of replicas” to ONE or MORE, to create replica shards 

Passing Filters over the Dashboard URL

Shared Link without filters: <http://qnc-ng-crm-vm02:5601/app/kibana#/dashboard/44f6b3e0-8dd1-11e8-b843-27a3466db4ec?_g=(refreshInterval%3A('%24%24hashKey'%3A'object%3A48456'%2Cdisplay%3AOff%2Cpause%3A!f%2Csection%3A0%2Cvalue%3A0)%2Ctime%3A(from%3Anow-15m%2Cmode%3Aquick%2Cto%3Anow))>

Adding filter to a shared link: [http://qnc-ng-crm-vm02:5601/app/kibana#/dashboard/44f6b3e0-8dd1-11e8-b843-27a3466db4ec?\_g=(refreshInterval%3A('%24%24hashKey'%3A'object%3A48456'%2Cdisplay%3AOff%2Cpause%3A!f%2Csection%3A0%2Cvalue%3A0)%2Ctime%3A(from%3Anow-15m%2Cmode%3Aquick%2Cto%3Anow))&\_a=(query:(language:lucene,query:'project\_id.keyword:JPR12188'))](http://qnc-ng-crm-vm02:5601/app/kibana#/dashboard/44f6b3e0-8dd1-11e8-b843-27a3466db4ec?_g=(refreshInterval%3A('%24%24hashKey'%3A'object%3A48456'%2Cdisplay%3AOff%2Cpause%3A!f%2Csection%3A0%2Cvalue%3A0)%2Ctime%3A(from%3Anow-15m%2Cmode%3Aquick%2Cto%3Anow))&_a=(query:(language:lucene,query:'projec)

                The above syntax is called Lucene Query Syntax

OR syntax to pass more than one device filter in the URL

[http://10.207.149.195:5601/app/kibana#/discover?\_g=(refreshInterval:('$$hashKey':'object:1061',display:'5%20seconds',pause:!f,section:1,value:5000),time:(from:now-15m,mode:quick,to:now))&\_a=(columns:!(sysloghost,severity,message),filters:!(),index:ceef58f0-c863-11e8-ae85-3df2321b895e,interval:auto,query:(language:lucene,query:'sysloghost.keyword:bull OR sysloghost.keyword:vale-ptxa'),sort:!('@timestamp',desc))](http://10.207.149.195:5601/app/kibana#/discover?_g=(refreshInterval:('$$hashKey':'object:1061',display:'5%20seconds',pause:!f,section:1,value:5000),time:(from:now-15m,mode:quick,to:now))&_a=(columns:!(sysloghost,severity,message),filters:!(),index:ceef58f0-c863-11e8-ae85-3df2321b895e,interval)

Setting up Filebeat

                tar -zxf filebeat-6.2.1-linux-x86\_64.tar.gz

                cd filebeat-6.2.1-linux-x86\_64

                vi filebeat.yml

filebeat.prospectors:

- type: log

enabled: true

paths:

   - /root/kvsr/elastic/logstash-tutorial.log

output.logstash:

   hosts: ["localhost:5044"]

                                #output.elasticsearch    (Make sure you comment out output.elasticsearch)

#  hosts: ["localhost:9200"]

                chmod go-w filebeat.yml

Linking rsyslog, logstash, elasticsearch and kibana

                rsyslog server    :                               qnc-jdi-regression           :               10.50.40.70

                logstash server :                               qnc-jdi-regression           :               10.50.40.70

                elasticsearch server        :               qnc-jdi-regression           :               10.50.40.70

Elasticsearch REST APIs

GET /\_cat/health?v

GET /\_cat/nodes?v

GET /\_cat/indices?v&pretty

GET /\_cat/shards?v

PUT /index1?pretty

DELETE /index1?pretty

PUT /index2/\_doc/1?pretty

{

  "name": "ganesha"

}

GET /index2/\_doc/1

DELETE /index2/\_doc/1?pretty

              POST /customer/\_doc?pretty

{

  "name": "Jane Doe"

}

POST /customer/\_doc/\_bulk?pretty

{"index":{"\_id":"1"}}

{"name":"John Doe"}

{"index":{"\_id":"2"}}

{"name":"Jane Doe"}

POST /customer/\_doc/\_bulk?pretty

{"update":{"\_id":"1"}}

{"doc":{"name":"John Doe becomes Jane Doe"}}

{"delete":{"\_id":"2"}}

                Loading a json file to Elasticsearch

       curl -H "Content-Type: application/json"-XPOST "localhost:9200/bank/account/\_bulk?pretty&refresh"--data-binary "@accounts.json"

       curl "localhost:9200/\_cat/indices?v"

              curl -H 'Content-Type: application/x-ndjson' -XPOST 'http://qnc-ng-crm-vm02:9200/tc\_exec/doc/\_bulk?pretty' --data-binary @elastic\_new\_load.json

# match\_all query

GET /tc\_exec/\_search

{

  "query": {

    "match\_all": {}

  }

}

# match\_all query with more attrs

GET /tc\_exec/\_search

{

  "query": {

    "match\_all": {}

  },

  "size": 3,

  "from": 0,

  "\_source": ["project\_id", "npi\_id", "rli\_num", "pr"]

}

# match query

GET /tc\_exec/\_search

{

  "query": {

    "match": {

      "project\_id": "JPR12188" # match takes only one attribute

    }

  }

}

# Delete Query

POST tc\_exec/\_delete\_by\_query

{

  "query": {

    "match\_all": {}

  }

}

# regexp

GET /tc\_exec/\_search

{

  "query": {

    "regexp": {

      "project\_id": "JPR1\*"

    }

  }

}

# bool query takes “should”, “must”, “filter” etc. inside it

# “should” is like “OR” …. “must” is like “AND”

# below query is equivalent to “select \* from index where message ~ (‘event subscription’) OR message ~ (‘commit operation’) OR message ~ (‘UI\_CMDLINE\_READ\_LINE’);”

{

"query": {

"bool": {

"should": [

{

"match": {

"message": {

"query": "event subscription",

"operator": "and"

}

}

},

{

"match": {

"message": {

"query": "commit operation",

"operator": "and"

}

}

},

{

"match": {

"message": {

"query": "UI\_CMDLINE\_READ\_LINE",

"operator": "and"

}

}

}

]

}

}

}

{

"query": {

"bool": {

"must": [

{

"terms": {

"sysloghost": [

"pushkin",

"skah"

]

}

},

{

"match": {

"message": {

"query": "User",

"operator": "and"

}

}

},

{

"match": {

"message": {

"query": "'regress'",

"operator": "and"

}

}

},

{

"match": {

"message": {

"query": "UI\_CMDLINE\_READ\_LINE",

"operator": "and"

}

}

}

]

}

}

}

# full text match query

GET /logstash-2018.12.05/\_search

{

    "query": {

      "bool": {

        "filter": [

          { "terms":  { "sysloghost.keyword": ["karbonn","karbonn1"] }},

          { "range": {

            "@timestamp": {

              "gte": "2018-12-05T16:36:00",

              "lte": "2018-12-05T16:37:00",

              "time\_zone": "+05:30"

          }}},

          { "bool": {

           "should": [

            { "match": {

              "message": {

                "query": "notify to other modules",

                "operator": "and"

              }

            }},

            { "match": {

              "message": {

                "query": "Port Alarms: Cleared",

                "operator": "and"

              }

            }}

            ],

            "minimum\_should\_match": 1

          }}

        ]

      }

    },

    "size": 3000,

    "sort" : [

          { "@timestamp" : {"order" : "desc"}}

      ]

}

# Combining regexp, exact matches and ranges

GET /logstash-2018.12.05/\_search

{

    "query": {

      "bool": {

        "filter": [

          { "terms":  { "sysloghost.keyword": ["karbonn","karbonn1"] }},

          { "range": {

            "@timestamp": {

              "gte": "2018-12-05T16:36:00",

              "lte": "2018-12-05T16:37:00",

             "time\_zone": "+05:30"

          }}},

          {

            "regexp" : {

              "message" : "notify"

            }

          }

        ]

      }

    },

    "size": 3000,

    "sort" : [

          { "@timestamp" : {"order" : "desc"}}

      ]

}

Retrieving and Updating Cluster settings when the Cluster is UP and Running

GET /\_cluster/settings

PUT /\_cluster/settings

{

  "transient": {

    "logger.org.elasticsearch.transport": "debug"

  }

}

Ingest Node

1. Use an ingest node to pre-process documents before the actual document indexing happens
2. All nodes enable ingest by default, so any node can handle ingest tasks
3. To disable ingest for a node, configure the following setting in the elasticsearch.yml file:

node.ingest: false

1. To pre-process documents before indexing, [define a pipeline](https://www.elastic.co/guide/en/elasticsearch/reference/current/pipeline.html) that specifies a series of [processors](https://www.elastic.co/guide/en/elasticsearch/reference/current/ingest-processors.html)
2. Each processor transforms the document in some specific way
3. To use a pipeline, simply specify the pipeline parameter on an index or bulk request
   1. PUT my-index/\_doc/my-id?pipeline=my\_pipeline\_id
   2. {
   3. "foo":"bar"
   4. }

[http://qnc-ng-crm-vm02:5601/app/kibana#/discover?\_g=(refreshInterval:('$$hashKey':'object:2067',display:'5%20seconds',pause:!f,section:1,value:5000),time:(from:now-15m,mode:quick,to:now))&\_a=(columns:!(sysloghost,programname,message,severity),index:'8881b480-7444-11e8-be16-576710754716',interval:auto,query:(language:lucene,query:''),sort:!('@timestamp',desc))](http://qnc-ng-crm-vm02:5601/app/kibana#/discover?_g=(refreshInterval:('$$hashKey':'object:2067',display:'5%20seconds',pause:!f,section:1,value:5000),time:(from:now-15m,mode:quick,to:now))&_a=(columns:!(sysloghost,programname,message,severity),index:'8881b480-7444-11e8-be16-576710754716',interv)

Cluster Saviour Commands

                python3.5 cluster\_saviour.py --type kibana --java\_home /usr/lib/jvm/java-11-openjdk-amd64 --kibana\_home /root/kvsr/elastic/kibana-6.5.0-linux-x86\_64

                python3.5 cluster\_saviour.py --type logstash --ls\_java\_home /home/regress/jdk/jdk1.8.0\_161 --ls\_home /root/kvsr/elastic/logstash-6.5.0

                python3.5 cluster\_saviour.py --type elastic --java\_home /usr/lib/jvm/java-11-openjdk-amd64 --es\_home /home/regress/elastic/elasticsearch-6.5.0 --node\_config\_dir config\_node4

python3.5 cluster\_saviour.py --type elastic --java\_home /usr/lib/jvm/java-11-openjdk-amd64 --es\_home /home/regress/elastic/elasticsearch-6.5.0 --node\_config\_dir config\_node5

python3.5 cluster\_saviour.py --type elastic --java\_home /usr/lib/jvm/java-11-openjdk-amd64 --es\_home /home/regress/elastic/elasticsearch-6.5.0 --node\_config\_dir config\_node4

python3.5 cluster\_saviour.py --type elastic --java\_home /usr/lib/jvm/java-11-openjdk-amd64 --es\_home /home/regress/elastic/elasticsearch-6.5.0 --node\_config\_dir config\_node3

python3.5 cluster\_saviour.py --type elastic --java\_home /usr/lib/jvm/java-11-openjdk-amd64 --es\_home /home/regress/elastic/elasticsearch-6.5.0 --node\_config\_dir config\_node2

python3.5 cluster\_saviour.py --type kibana --java\_home /usr/lib/jvm/java-11-openjdk-amd64 --kibana\_home /root/kvsr/elastic/kibana-6.5.0-linux-x86\_64

python3.5 cluster\_saviour.py --type elastic --java\_home /usr/lib/jvm/java-11-openjdk-amd64 --es\_home /home/regress/elastic/elasticsearch-6.5.0 --node\_config\_dir config\_node1

python3.5 cluster\_saviour.py --type elastic --java\_home /usr/lib/jvm/java-11-openjdk-amd64 --es\_home /home/regress/elastic/elasticsearch-6.5.0 --node\_config\_dir config

ElasticSearch Configuration Options

cluster.name: bng-elnath-01

node.name: node1

bootstrap.memory\_lock: true # To lock the memory only for ES use

network.host: 0.0.0.0 # The node will bind to this hostname or IP address and *publish* (advertise) this host to other nodes in the cluster

transport.host: localhost # Transport module is used for internal communication of nodes in the cluster

network.bind\_host: 10.207.149.195 # Defaults to network.host. This specifies which network interface(s) a node should bind to in order to listen for incoming requests

discovery.zen.minimum\_master\_nodes: 3 #These many master node are required for cluster to start working as well as for master node election

#This is to avoid split-brain scenario and also to avoid the data loss. It is better to have more than 2 master eligible nodes

#minimum\_master\_nodes = (master\_eligible\_nodes / 2) + 1

path.data: /home/regress/elastic/elasticsearch-6.5.0/data\_node1

path.logs: /home/regress/elastic/elasticsearch-6.5.0/logs\_node1

node.master: false

http.port: 9201 # On this port, The http module allows to expose **Elasticsearch** APIs over HTTP

path.repo: ["/es\_snapshots/6.5.0"]