### **ELASTICSEARCH TASKS:**



#### **1.1 Install elasticsearch cluster version 7.17.8**

Bash



wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.17.8-linux-x86\_64.tar.gz

tar -xzf elasticsearch-7.17.8-linux-x86\_64.tar.gz

cd elasticsearch-7.17.8/

*# Repeat on all 3 nodes.*

#### **1.2. Configure JVM Settings**

# **Edit config/jvm.options. Best practice: set Xms and Xmx to the same value, no more than 50% of system RAM.**

Bash



# config/jvm.options

-Xms1g

-Xmx1g

#### **1.3. Generate TLS Certificates**

# **Run these steps on one node only.**

Bash



# Create certificates directory

mkdir config/certs

# Generate CA

./bin/elasticsearch-certutil ca --pem --out config/certs/elastic-stack-ca.zip

unzip config/certs/elastic-stack-ca.zip -d config/certs/

# Create instances.yml to define all nodes for the certificates

cat > config/certs/instances.yml << EOL

instances:

- name: "es1"

ip: ["192.168.56.10"]

- name: "es2"

ip: ["192.168.56.11"]

- name: "es3"

ip: ["192.168.56.12"]

EOL

# Generate node certificates from the instances file

./bin/elasticsearch-certutil cert --ca config/certs/elastic-stack-ca/ca.crt --ca-key config/certs/elastic-stack-ca/ca.key --pem --in config/certs/instances.yml --out config/certs/nodes.zip

unzip config/certs/nodes.zip -d config/certs/

# **Now, securely copy the config/certs/elastic-stack-ca directory and the specific node certificate directory (e.g., config/certs/es2) to the other nodes.**

#### **1.4 Configure elasticsearch.yml and Start**

# **First, create data/log dirs and set ownership before starting. Run on all nodes.**

Bash



sudo mkdir -p /var/lib/elasticsearch /var/log/elasticsearch

sudo chown -R elasticsearch:elasticsearch /var/lib/elasticsearch /var/log/elasticsearch

# **Configure elasticsearch.yml on es1**

YAML



cluster.name: es-cluster

node.name: es1

path.data: /var/lib/elasticsearch

path.logs: /var/log/elasticsearch

network.host: 192.168.56.10

http.port: 9200

transport.port: 9300

discovery.seed\_hosts: ["192.168.56.10", "192.168.56.11", "192.168.56.12"]

cluster.initial\_master\_nodes: ["es1", "es2", "es3"]

xpack.security.enabled: true

xpack.security.transport.ssl.enabled: true

xpack.security.transport.ssl.verification\_mode: certificate

xpack.security.transport.ssl.certificate\_authorities: [ "certs/elastic-stack-ca/ca.crt" ]

xpack.security.transport.ssl.key: certs/es1/es1.key

xpack.security.transport.ssl.certificate: certs/es1/es1.crt

xpack.security.http.ssl.enabled: true

xpack.security.http.ssl.certificate\_authorities: [ "certs/elastic-stack-ca/ca.crt" ]

xpack.security.http.ssl.key: certs/es1/es1.key

xpack.security.http.ssl.certificate: certs/es1/es1.crt

# **Similar Config for other 2 Nodes just with change of node.name, network.host, and certificate paths (e.g., certs/es2/es2.key).**

# **Start Elasticsearch on all nodes**

Bash



./bin/elasticsearch -d

# **Set passwords for built-in users (run on one node only, after cluster has formed)**

Bash



./bin/elasticsearch-setup-passwords auto

# **SAVE THE OUTPUT! Then update your ELASTIC\_PASSWORD environment variable.**

#### **Verify Cluster Formation**

Bash



curl --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD "https://$ES\_HOST:9200/\_cluster/health?pretty"

JSON



{

"cluster\_name" : "es-cluster",

"status" : "green",

"timed\_out" : false,

"number\_of\_nodes" : 3,

"number\_of\_data\_nodes" : 3,

"active\_primary\_shards" : 0,

"active\_shards" : 0,

"relocating\_shards" : 0,

"initializing\_shards" : 0,

"unassigned\_shards" : 0,

"delayed\_unassigned\_shards" : 0,

"number\_of\_pending\_tasks" : 0,

"number\_of\_in\_flight\_fetch" : 0,

"task\_max\_waiting\_in\_queue\_millis" : 0,

"active\_shards\_percent\_as\_number" : 100.0

}

#### **3. Create Indices and Insert Data**

# **The commands you provided for creating indices (books, authors, publishers) and for bulk inserting data were correct and are ready to be used.**

# **Example: Create books index**

Bash



curl -X PUT "https://$ES\_HOST:9200/books" -u elastic:$ELASTIC\_PASSWORD --cacert $CACERT\_PATH -H 'Content-Type: application/json' -d'

{

"settings": { "number\_of\_shards": 3, "number\_of\_replicas": 1 },

"mappings": { "properties": { "title": {"type": "text"}, "author": {"type": "text"}, "published\_year": {"type": "integer"}, "publisher\_id": {"type": "keyword"}, "isbn": {"type": "keyword"} }}

}'

# 

#### **Create Snapshot**

# **Step 1: Set Up an NFS Server**

# **Step 2: Mount the NFS Share on all ES nodes and configure path.repo**

# **Update elasticsearch.yml on all nodes:**

path.repo: ["/mnt/es\_backups"]

# **Restart all Elasticsearch nodes after this change.**

# **Step 3: Register the Snapshot Repository** Bash



curl --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD -X PUT "https://$ES\_HOST:9200/\_snapshot/my\_backup" -H 'Content-Type: application/json' -d'

{

"type": "fs",

"settings": {

"location": "/mnt/es\_backups"

}

}'

# **Step 4: Create the Snapshot**

Bash



curl --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD -X PUT "https://$ES\_HOST:9200/\_snapshot/my\_backup/snapshot\_1?wait\_for\_completion=true"

# **Step 5: Simulate Disaster - Delete Your Indices**

Bash



curl --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD -X DELETE "https://$ES\_HOST:9200/books,authors,publishers"

# **Step 6: Restore the Indices from the Snapshot**

Bash



curl --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD -X POST "https://$ES\_HOST:9200/\_snapshot/my\_backup/snapshot\_1/\_restore?wait\_for\_completion=true"

# **Step 7: Verify Restore**

Bash



curl --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD "https://$ES\_HOST:9200/\_cat/indices?v"

#### **Upgrade to Elasticsearch 8.15.3**

# **Step 1: Run Pre-Upgrade Check from your 7.17.8 cluster**

Bash



curl --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD "https://$ES\_HOST:9200/\_migration/deprecations"

# 

# **Step 2: Disable Shard Allocation**

Bash



curl --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD -X PUT "https://$ES\_HOST:9200/\_cluster/settings" -H 'Content-Type: application/json' -d'

{ "persistent": { "cluster.routing.allocation.enable": "primaries" } }'

# **Step 3: Upgrade Each Node One by One (e.g., es3 -> es2 -> es1)**

# **For each node:**

# **a. Stop the node: pkill -f "java.\*elasticsearch"**

# **b. Download CORRECT architecture package and install**

Bash



wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.15.3-linux-x86\_64.tar.gz

tar -xzf elasticsearch-8.15.3-linux-x86\_64.tar.gz

# c. Copy config directory from old install to new one (elasticsearch-8.15.3/config/)

# d. Start the new node: `cd elasticsearch-8.15.3/ && ./bin/elasticsearch -d`

# e. Wait for it to join the cluster. Check `\_cat/nodes`.

# **Step 4: Re-enable Allocation after all nodes are upgraded**

Bash



curl --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD -X PUT "https://$ES\_HOST:9200/\_cluster/settings" -H 'Content-Type: application/json' -d'

{ "persistent": { "cluster.routing.allocation.enable": "all" } }'

#### **Capture Slow logs & Manually Allocate shard**

#### **Apply Settings to All *Existing* Indices**

This command updates the settings for your current books, authors, and publishers indices.

Bash

# Update settings for all indices

curl -k --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD -X PUT "https://\_ES\_HOST\_:9200/\_all/\_settings" -H 'Content-Type: application/json' -d'

{

"index.search.slowlog.threshold.query.warn": "1ms",

"index.search.slowlog.threshold.fetch.warn": "1ms",

"index.search.slowlog.level": "info",

"index.indexing.slowlog.threshold.index.warn": "1ms",

"index.indexing.slowlog.level": "info",

"index.indexing.slowlog.source": "1000"

}'



**Run the Query to find slow logs:**

curl -k --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD -X PUT "https://192.168.56.10:9200/\_index\_template/slowlog\_template" -H 'Content-Type: application/json' -d'

{

"index\_patterns": ["\*"],

"template": {

"settings": {

"index.search.slowlog.threshold.query.trace": "1ms",

"index.indexing.slowlog.threshold.index.trace": "1ms"

}

},

"priority": 500

}'

**Run Query:**

curl -k --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD -X GET "https://\_ES\_HOST\_:9200/books/\_search?pretty" -H 'Content-Type: application/json' -d'

{

"query": {

"script": {

"script": {

"source": "Thread.sleep(10); return true;"

}

}

}

}'

**Watch the Logs**

sudo tail -f /var/log/elasticsearch/es-cluster\_index\_search\_slowlog.json | jq

**Manually Allocate shard**

**curl -k --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD "https://$ES\_HOST:9200/\_cat/shards/books?v"**

****

**Disable Automatic Shard Allocation**

curl -k --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD -X PUT "https://$ES\_HOST:9200/\_cluster/settings" -H 'Content-Type: application/json' -d'

{

"persistent": {

"cluster.routing.allocation.enable": "none"

}

}'

Manually Move a Shard

curl -k --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD -X POST "https://$ES\_HOST:9200/\_cluster/reroute" -H 'Content-Type: application/json' -d'

{

"commands": [

{

"move": {

"index": "books",

"shard": 0,

"from\_node": "es3",

"to\_node": "es1"

}

}

]

}'

**Verify the New Location**

curl -k --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD "https://\_ES\_HOST\_:9200/\_cat/shards/books?v"

**Re-enable Automatic Allocation**

curl -k --cacert $CACERT\_PATH -u elastic:$ELASTIC\_PASSWORD -X PUT "https://$ES\_HOST:9200/\_cluster/settings" -H 'Content-Type: application/json' -d'

{

"persistent": {

"cluster.routing.allocation.enable": "all"

}

}'

#### **Set Up New Standalone Cluster & Reindex Data**

# **Step 1: Install new standalone node (e.g., Elasticsearch 8.x) as before.**

# **Step 2: Generate self-signed cert for HTTP**

Bash



# In the new ES 8.x directory

./bin/elasticsearch-certutil http --self-signed --out config/certs/http.p12

# **Step 3: Configure elasticsearch.yml**

YAML



cluster.name: standalone-cluster

node.name: standalone-node

path.data: /var/lib/elasticsearch

path.logs: /var/log/elasticsearch

network.host: 192.168.56.12

discovery.type: single-node

xpack.security.enabled: true

xpack.security.http.ssl.enabled: true

xpack.security.http.ssl.keystore.path: config/certs/http.p12

# **Step 4: Start the node. On first start, it will generate a password for the 'elastic' user.**

# **Step 5: Get data from the old cluster using Reindex API**

# **Must be run for each index.**

Bash



curl -k -u elastic:$DEST\_PASSWORD -X POST "https://$DEST\_HOST:9200/\_reindex?wait\_for\_completion=true" -H 'Content-Type: application/json' -d'

{

"source": {

"remote": {

"host": "https://<SOURCE\_HOST\_IP>:9200",

"username": "elastic",

"password": "'"$SOURCE\_PASSWORD"'"

},

"index": "books"

},

"dest": {

"index": "books"

}

}'

# **Step 6: Verify**

Bash



curl -k -u elastic:$DEST\_PASSWORD "https://localhost:9200/\_cat/indices?v"

#### **Configure a Custom JDK**

export ES\_JAVA\_HOME=/path/to/custom/jdk