

## 2-NODE DOCKERIZED ELASTICSEARCH CLUSTER

-Designed by Yogesh Darji

### How to create ElasticSearch Cluster

**Step1: Run the Docker Image with following command on one box say - 192.168.56.101**

```
root@osboxes:/home/osboxes# docker run -d my_es:v1
12fb36ce4de45a352136ba0e551f1d36a8cf25f10201384ef9161dd32c7e0a6d
```

```
root@osboxes:/home/osboxes#
root@osboxes:/home/osboxes# docker images
REPOSITORY          TAG          IMAGE ID          CREATED           SIZE
my_es               v1          7732a4e7da7d     6 days ago       346.2 MB
java                8-jre       76fd51ceaa2e     13 days ago      312.2 MB
hello-world        latest      693bce725149     2 weeks ago      967 B
ubuntu             latest      2fa927b5cdd3     3 weeks ago      122 MB
root@osboxes:/home/osboxes#
root@osboxes:/home/osboxes#
root@osboxes:/home/osboxes# docker run -d my_es:v1
12fb36ce4de45a352136ba0e551f1d36a8cf25f10201384ef9161dd32c7e0a6d
root@osboxes:/home/osboxes#
root@osboxes:/home/osboxes# docker ps
CONTAINER ID        IMAGE          COMMAND                  CREATED           STATUS           PORTS                    NAMES
12fb36ce4de4        my_es:v1      "/docker-entrypoint.s"  16 seconds ago   Up 15 seconds   9200/tcp, 9300/tcp     suspicious_engelbart
root@osboxes:/home/osboxes#
root@osboxes:/home/osboxes#
root@osboxes:/home/osboxes# █
```

**Step2: Run the Docker Image with following command on other box say - 192.168.56.102**

```
root@osboxes:/home/osboxes# docker run -d my_es:v1
8a923ff11d07ce640811e9110a8fd8064381294a1897be5b9ae9f9ef47519028
```

```
[root@osboxes:/home/osboxes#
[root@osboxes:/home/osboxes# docker images
REPOSITORY          TAG          IMAGE ID          CREATED           SIZE
my_es               v1          7732a4e7da7d     6 days ago       346.2 MB
java                8-jre       76fd51ceaa2e     13 days ago      312.2 MB
hello-world        latest      693bce725149     2 weeks ago      967 B
ubuntu             latest      2fa927b5cdd3     3 weeks ago      122 MB
[root@osboxes:/home/osboxes#
[root@osboxes:/home/osboxes# docker run -d my_es:v1
8a923ff11d07ce640811e9110a8fd8064381294a1897be5b9ae9f9ef47519028
[root@osboxes:/home/osboxes#
[root@osboxes:/home/osboxes# docker ps
CONTAINER ID        IMAGE          COMMAND                  CREATED           STATUS           PORTS                    NAMES
8a923ff11d07        my_es:v1      "/docker-entrypoint.s"  3 seconds ago    Up 2 seconds    9200/tcp, 9300/
root@osboxes:/home/osboxes# █
```

**Step3: Inspect both the Nodes**

```
$docker inspect 12fb36ce4de4
```

You will find "IPAddress": "172.17.0.2" in the Network Section at the bottom

```
    "IPAddress": "172.17.0.2",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "MacAddress": "02:42:ac:11:00:02",
    "Networks": {
      "bridge": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": null,
        "NetworkID": "a7ac4ee1b4a6371e3ef66ebb2cc34ce531661f1c1053e5774779e8c93d1900b0",
        "EndpointID": "6d77d678c87534ae8237aa597981ff19f7afa4773d9efc3d69fb1674de92ef49",
        "Gateway": "172.17.0.1",
        "IPAddress": "172.17.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:ac:11:00:02"
      }
    }
  }
}
]
root@osboxes:/home/osboxes#
```

```
$docker inspect 5dcc1cc47729
```

You will find "IPAddress": "172.17.0.3" in the Network Section at the bottom

## 2-NODE DOCKERIZED ELASTICSEARCH CLUSTER

-Designed by Yogesh Darji

```
    "IPv4Gateway": "172.17.0.1",
    "MacAddress": "02:42:ac:11:00:03",
    "Networks": {
      "bridge": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": null,
        "NetworkID": "0151d4c7cde61a5f62a3f16569a18221dd536d1d50345d9c6c7224ca259b622d",
        "EndpointID": "cc2004188eea49c3efe9cfba44d9eb0b5b4476daffff14b678d35eef7e8d9bb90",
        "Gateway": "172.17.0.1",
        "IPAddress": "172.17.0.3",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:ac:11:00:03"
      }
    }
  }
}
```

```
ot@osboxes:/home/osboxes#
```

### **Step 4: Check the cluster health on both the boxes**

```
root@osboxes:/VZElastic# curl -XGET http://172.17.0.2:9200/\_cluster/health/?pretty
```

```
root@osboxes:/VZElastic# curl -XGET http://172.17.0.2:9200/_cluster/health/?pretty
{
  "cluster_name" : "elasticsearch",
  "status" : "green",
  "timed_out" : false,
  "number_of_nodes" : 1,
  "number_of_data_nodes" : 1,
  "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
}
root@osboxes:/VZElastic#
```

## 2-NODE DOCKERIZED ELASTICSEARCH CLUSTER

-Designed by Yogesh Darji

```
root@osboxes:/VZElastic# curl -XGET http://172.17.0.3:9200/\_cluster/health/?pretty
```

```
[root@osboxes:/VZElastic# curl -XGET http://172.17.0.3:9200/_cluster/health/?pretty
{
  "cluster_name" : "elasticsearch",
  "status" : "green",
  "timed_out" : false,
  "number_of_nodes" : 1,
  "number_of_data_nodes" : 1,
  "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
}
[root@osboxes:/VZElastic# _
```

### Step 5: Start the dockerized Elastic Search Instance on both boxes

```
docker exec -it 12fb36ce4de4 bash
```

```
bin/elasticsearch -Des.insecure.allow.root=true
```

```
[root@osboxes:/VZElastic#
[root@osboxes:/VZElastic# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
12fb36ce4de4       my_es:v1           "/docker-entrypoint.s"  18 minutes ago     Up 18 minutes      9200/tcp, 9300/tcp  suspicious_engelbart
[root@osboxes:/VZElastic#
[root@osboxes:/VZElastic# docker exec -it 12fb36ce4de4 bash
[root@12fb36ce4de4:/usr/share/elasticsearch#
[root@12fb36ce4de4:/usr/share/elasticsearch# bin/elasticsearch -Des.insecure.allow.root=true
[2016-06-23 22:04:04,456] [WARN ] [bootstrap                ] running as ROOT user. this is a bad idea!
[2016-06-23 22:04:04,785] [INFO ] [node                  ] [Cecelia Reyes] version[2.3.3], pid[99], build[218bdf1/2016-05-17T15:40:04Z]
[2016-06-23 22:04:04,787] [INFO ] [node                  ] [Cecelia Reyes] initializing ...
[2016-06-23 22:04:05,617] [INFO ] [plugins                ] [Cecelia Reyes] modules [reindex, lang-expression, lang-groovy], plugins [], sites []
[2016-06-23 22:04:05,641] [INFO ] [env                    ] [Cecelia Reyes] using [1] data paths, mounts [[/usr/share/elasticsearch/data (/dev/disk/by-1e5e7fd18af)]], net usable_space [85.8gb], net total_space [94.9gb], spins? [possibly], types [ext4]
[2016-06-23 22:04:05,642] [INFO ] [env                    ] [Cecelia Reyes] heap size [1015.6mb], compressed ordinary object pointers [true]
[2016-06-23 22:04:08,566] [INFO ] [node                  ] [Cecelia Reyes] initialized
[2016-06-23 22:04:08,567] [INFO ] [node                  ] [Cecelia Reyes] starting ...
[2016-06-23 22:04:08,752] [INFO ] [transport              ] [Cecelia Reyes] publish_address {172.17.0.2:9301}, bound_addresses [{::}:9301]
[2016-06-23 22:04:08,757] [INFO ] [discovery              ] [Cecelia Reyes] elasticsearch/1u7cNDuJT-GG8-Ne6M6LLA
[2016-06-23 22:04:11,909] [INFO ] [cluster.service        ] [Cecelia Reyes] detected_master {Photon}{pe1RKZHUR3aNV9S_q5tgcg}{172.17.0.2}{172.17.0.2:9301}
aNV9S_q5tgcg}{172.17.0.2}{172.17.0.2:9300}}, reason: zen-disco-receive{from master [{Photon}{pe1RKZHUR3aNV9S_q5tgcg}{172.17.0.2}{172.17.0.2:9300}]}
[2016-06-23 22:04:11,992] [INFO ] [http                   ] [Cecelia Reyes] publish_address {172.17.0.2:9201}, bound_addresses [{::}:9201]
[2016-06-23 22:04:11,993] [INFO ] [node                  ] [Cecelia Reyes] started
```

## 2-NODE DOCKERIZED ELASTICSEARCH CLUSTER

-Designed by Yogesh Darji

**Step 6: Since, the Elasticsearch is running on both the windows, open a new window, to set up a cluster**

**Step 7: sudo apt-get update**

**Step 8: Edit etc/hosts file of all nodes and make entries for all nodes for the hostnames as follows:**

vi etc/hosts in the root

172.17.0.2 node-1

172.17.0.3 node-2

```
127.0.0.1          localhost
127.0.1.1          osboxes

172.17.0.2 node-1
172.17.0.3 node-2

# The following lines are desi
::1          ip6-localhost ip6-loop
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
~
~
~
~
~
```

**Step 9: All elastic search configuration files are in elasticsearch.yml which is located in /etc/elasticsearch folder**

```
apt-get update
apt-get install vim
```

```
cd /etc/elasticsearch/
```

## 2-NODE DOCKERIZED ELASTICSEARCH CLUSTER

-Designed by Yogesh Darji

```
root@127b36ce4de4:/usr/share/elasticsearch/bin# cd ..
root@127b36ce4de4:/usr/share/elasticsearch# ls
NOTICE.txt README.textile bin config data lib logs modules plugins
root@127b36ce4de4:/usr/share/elasticsearch#
root@127b36ce4de4:/usr/share/elasticsearch# cd /etc/elasticsearch/
root@127b36ce4de4:/etc/elasticsearch# ls
elasticsearch.yml logging.yml scripts
root@127b36ce4de4:/etc/elasticsearch#
```

**Step 10: Now edit the `elasticsearch.yml` file for configuring nodes**

**`vim elasticsearch.yml`**

i. Under cluster section:

```
cluster.name: ES-cluster
```

ii. Under node section, change the node name parameter and add

other parameters as shown below.

```
node.name: node-1
```

```
node.client: true
```

```
node.data: false
```

```
..
# ----- Cluster -----
#
# Use a descriptive name for your cluster:
#
cluster.name: ES-cluster
#
# ----- Node -----
#
# Use a descriptive name for the node:
#
[ node.name: node-1
[ node.client: true
    node.data: false
#
# Add custom attributes to the node:
#
```

Under network section, change the “`network.host`” parameter with IP address of your client node.

```
network.host: 172.17.0.2
```

## 2-NODE DOCKERIZED ELASTICSEARCH CLUSTER

-Designed by Yogesh Darji

iii. Under discovery Section:

```
discovery.zen.ping.multicast.enabled:false
```

```
discovery.zen.ping.unicast.hosts: ["node-1", "node-2"]
```

```
# ----- Discovery -----  
#  
# Pass an initial list of hosts to perform discovery when new node is started:  
# The default list of hosts is ["127.0.0.1", "[::1]"]  
#  
discovery.zen.ping.unicast.hosts: ["node-1", "node-2"]  
  
discovery.zen.ping.multicast.enabled:false  
  
# Prevent the "split brain" by configuring the majority of nodes (total number of nodes / 2 + 1):  
#  
# discovery.zen.minimum_master_nodes: 3  
#  
# For more information, see the documentation at:  
# <http://www.elastic.co/guide/en/elasticsearch/reference/current/modules-discovery.html>  
#
```

iv. Save the file and restart the elastic search service for changes.

v. `sudo service elasticsearch restart`

Now when I run the

`curl -XGET http://172.17.0.2:9200/\_cluster/health?pretty`

```
[root@osboxes:/home/osboxes# docker ps  
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES  
12fb36ce4de4       my_es:v1           "/docker-entrypoint.s"   About an hour ago   Up About an hour   9200/tcp, 9300/tcp   su:  
[root@osboxes:/home/osboxes#  
[root@osboxes:/home/osboxes#  
[root@osboxes:/home/osboxes# curl -XGET http://172.17.0.2:9200/_cluster/health?pretty  
{  
  "cluster_name" : "elasticsearch",  
  "status" : "green",  
  "timed_out" : false,  
  "number_of_nodes" : 2,  
  "number_of_data_nodes" : 2,  
  "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  
}
```

### Playing with your Cluster

#### Step 1. Create an index named – customer in the node

```
[root@osboxes:/home/osboxes#  
[root@osboxes:/home/osboxes# curl -XPUT '172.17.0.2:9200/customer?pretty'  
{  
  "acknowledged" : true  
}  
[root@osboxes:/home/osboxes#  
[root@osboxes:/home/osboxes#  
[root@osboxes:/home/osboxes# curl '172.17.0.2:9200/_cat/indices?v'  
health status index      pri rep docs.count docs.deleted store.size pri.store.size  
green open   customer  5   1         0           0       1.2kb          650b  
[root@osboxes:/home/osboxes#  
[root@osboxes:/home/osboxes#
```

#### Step 2: Now increase the number of replicas of shard by 2:

```
curl -XPUT '172.17.0.2:9200/customer/_settings' -d '  
{  
  "index" : {  
    "number_of_replicas" : 2  
  }  
}'
```

As you can see in the SS, the number of unassigned shards have become 5 now. Also the "active\_shards\_percent\_as\_number" : 66.66666666666666" has reduced and the "status" : "yellow", "unassigned\_shards" : 5,



## 2-NODE DOCKERIZED ELASTICSEARCH CLUSTER

-Designed by Yogesh Darji

```
[root@osboxes:/home/osboxes#  
[root@osboxes:/home/osboxes# curl -XGET http://172.17.0.2:9200/_cluster/health/?pretty  
{  
  "cluster_name" : "elasticsearch",  
  "status" : "yellow",  
  "timed_out" : false,  
  "number_of_nodes" : 2,  
  "number_of_data_nodes" : 2,  
  "active_primary_shards" : 5,  
  "active_shards" : 10,  
  "relocating_shards" : 0,  
  "initializing_shards" : 0,  
  "unassigned_shards" : 5,  
  "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" : 66.66666666666666  
}
```

### Some ElasticSearch Concepts:

Faster searches when compared to traditional DB. It is distributed in nature.

#### **Node:**

Node is an instance of Elasticsearch. A collection of connected nodes is called Cluster. If you are running a single node of Elasticsearch, then you have cluster of one node.

Every node in the cluster can handle HTTP and Transport traffic by default. The transport layer is used exclusively for communication between nodes and between nodes and the **Java TransportClient**; the HTTP layer is used only by external REST clients.

All nodes know about all the other nodes in the cluster and can forward client requests to the appropriate node. Besides that, each node serves one or more purpose:

#### **Master-eligible node**

A node that has `node.master` set to true (default), which makes it eligible to be elected as the *master* node, which controls the cluster.

## 2-NODE DOCKERIZED ELASTICSEARCH CLUSTER

-Designed by Yogesh Darji

### **Data Node:**

A node that has `node.data` set to `true` (default). Data nodes hold data and perform data related operations such as CRUD, search, and aggregations.

### **Client Node:**

A client node has both `node.master` and `node.data` set to `false`. It can neither hold data nor become the master node. It behaves as a “smart router” and is used to forward cluster-level requests to the master node and data-related requests (such as search) to the appropriate data nodes.

### **Tribe node**

A tribe node, configured via the `tribe.*` settings, is a special type of client node that can connect to multiple clusters and perform search and other operations across all connected clusters.

### **Master Eligible Node**

The master node is responsible for lightweight cluster-wide actions such as creating or deleting an index, tracking which nodes are part of the cluster, and deciding which shards to allocate to which nodes. It is important for cluster health to have a stable master node.

`node.master: true`

`node.data: false`

### **Sharding Rules:**

It is horizontal Scaling.

This module provides per-index settings to control the allocation of shards to nodes:

- Shard allocation filtering: Controlling which shards are allocated to which nodes.
- Delayed allocation: Delaying allocation of unassigned shards caused by a node leaving.
- Total shards per node: A hard limit on the number of shards from the same index per node.

**Shard Allocation Filtering**, allows you to specify which nodes are allowed to host the shards of a particular index.

```
PUT test/_settings
```

```
{
```

```
"index.routing.allocation.include.size": "big,medium"  
}
```

PUT test/\_settings

```
{  
  "index.routing.allocation.include.size": "big",  
  "index.routing.allocation.include.rack": "rack1"  
}
```

### **Delaying Allocation when a node leaves:**

When a node leaves the cluster for whatever reason, intentional or otherwise, the master reacts by:

- Promoting a replica shard to primary to replace any primaries that were on the node.
- Allocating replica shards to replace the missing replicas (assuming there are enough nodes).
- Rebalancing shards evenly across the remaining nodes.

### ***Monitoring delayed unassigned shards***

The number of shards whose allocation has been delayed by this timeout setting can be viewed with the [cluster health API](#):

```
GET _cluster/health
```

### ***Removing a node permanently***

If a node is not going to return and you would like Elasticsearch to allocate the missing shards immediately, just update the timeout to zero:

```
PUT /_all/_settings  
{
```

```
"settings": {  
  "index.unassigned.node_left.delayed_timeout": "0"  
}
```

### **Total Shards per node:**

The cluster-level shard allocator tries to spread the shards of a single index across as many nodes as possible. However, depending on how many shards and indices you have, and how big they are, it may not always be possible to spread shards evenly.

The following *dynamic* setting allows you to specify a hard limit on the total number of shards from a single index allowed per node:

```
index.routing.allocation.total_shards_per_node
```

### **How to Replicate Data?**

At index time, a replica shard does the same amount of work as the primary shard. New documents are first indexed on the primary and then on any replicas. Increasing the number of replicas does not change the capacity of the index.

However, replica shards can serve read requests. If, as is often the case, your index is search heavy, you can increase search performance by increasing the number of replicas, but only if you also *add extra hardware*

Adding more nodes would not help us to add indexing capacity, but we could take advantage of the extra hardware at search time by increasing the number of replicas:

## 2-NODE DOCKERIZED ELASTICSEARCH CLUSTER

-Designed by Yogesh Darji

```
[root@osboxes:/home/osboxes#
[root@osboxes:/home/osboxes#
root@osboxes:/home/osboxes# curl -XPUT '172.17.0.2:9200/customer/_settings' -d '
> {
>   "index" : {
>     "number_of_replicas" : 2
>   }
[> }'
[{"acknowledged":true}root@osboxes:/home/osboxes#
[root@osboxes:/home/osboxes#
[root@osboxes:/home/osboxes# curl -XGET 'http://172.17.0.2:9200/'
{
  "name" : "Photon",
  "cluster_name" : "elasticsearch",
  "version" : {
    "number" : "2.3.3",
    "build_hash" : "218bdf10790eef486ff2c41a3df5cfa32dadcfde",
    "build_timestamp" : "2016-05-17T15:40:04Z",
    "build_snapshot" : false,
    "lucene_version" : "5.5.0"
  },
  "tagline" : "You Know, for Search"
}
```

```
[root@osboxes:/home/osboxes#
[root@osboxes:/home/osboxes# curl -XGET http://172.17.0.2:9200/_cluster/health?pretty
{
  "cluster_name" : "elasticsearch",
  "status" : "yellow",
  "timed_out" : false,
  "number_of_nodes" : 2,
  "number_of_data_nodes" : 2,
  "active_primary_shards" : 5,
  "active_shards" : 10,
  "relocating_shards" : 0,
  "initializing_shards" : 0,
  "unassigned_shards" : 5,
  "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" : 66.66666666666666
}
root@osboxes:/home/osboxes# █
```