CSCI 5533 - Distributed Information Systems
Distributed Clusters in Elasticsearch
University of Houston - Clear Lake

Spencer Riner

January 26, 2020

Contents

1 Introduction

Elasticsearch is a distributed and open source search engine used to index various types of unstructured data [1]. Each independent machine running an instance of Elasticsearch is referred to in this lab as a **server**. Many servers who coordinate with one another comprise a **cluster**. In this lab, you will create a three-node cluster that is able to recover from an unexpected outage without any user intervention. The Elasticsearch documentation is extensive and will prove useful during the lab. You can find it here: `https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html`

1.1 Indices

The essential components of an Elasticsearch cluster are **indices**, **shards**, and **replicas**. Indices are a logical namespace which map to one or more primary shards and have zero or more replica shards [2]. An index, in some cases, can be thought of similarly to a relational database. See the table below for a correlation of the terms used in each. The advantage that Elasticsearch provides is in the sharding of indices. Depending on your architecture, shards can be distributed across multiple servers and replicated.

Table 1: Comparison between Elasticsearch and relational database terms.

Structured DBMS	**Elasticsearch**
Database	Index
Tables	Types
Rows	Documents
Attributes	Properties

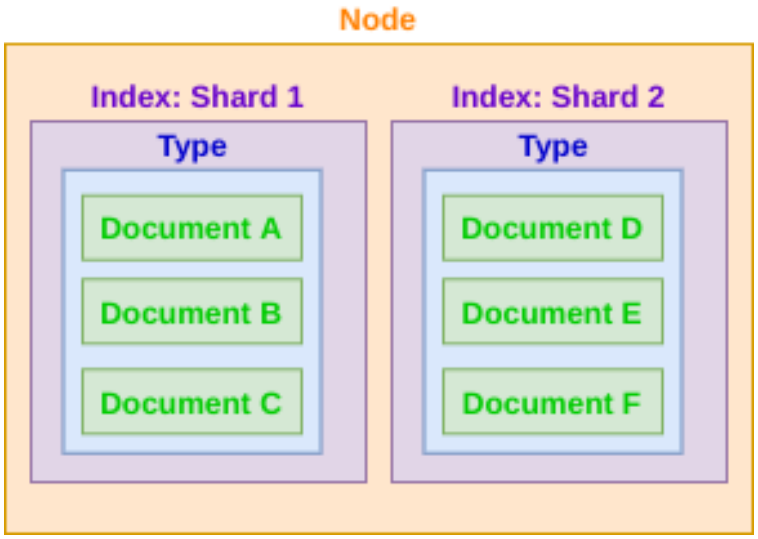

Figure 1: The nested structure of documents in an Elasticsearch index.

1.2 Shards and Replicas

The concept of shards and replicas is important to understand. A shard is a self-contained index that contains a subset of that index's documents [3]. This is how Elasticsearch distributes its data across multiple physical nodes. There are two types of shards: **primary** and **replica**.

Primary shards are the original copy of the shard. Primary shards are then copied to other machines as replicas. When a server is powered off or loses its network connection, the replica shards are used to create new primary shards to ensure the availability of the index data.

In this lab, each index will have three primary shards (one for each node) and one replica of each of the primary shards. See Figure 2.

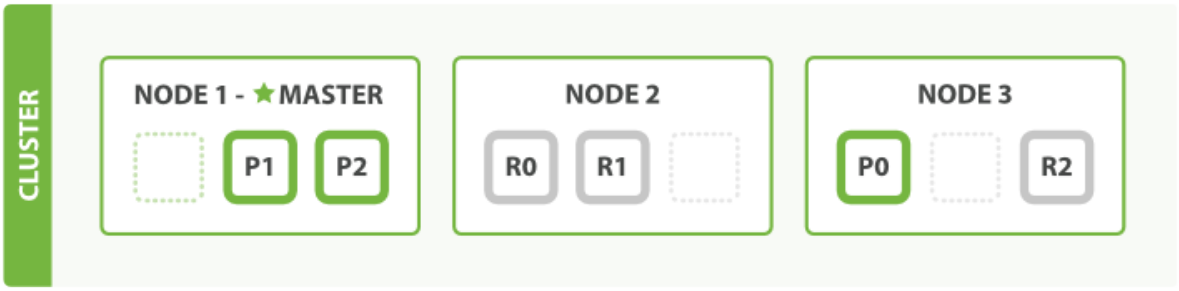

Figure 2: An Elasticsearch cluster with three nodes.

The squares labelled **P0**, **P1**, and **P2** all represent primary shards. These are the original copies of the documents contained in the index. The squares labelled **R0**, **R1**, and **R2** are the respective replicas of each primary shard. You can see that if any one of the nodes were to go down unexpectedly, any of the shards could be reliably replicated to recreate the 0 through 2 shards.

1.2.1 A Simple Shard Example

A car dealership has decided to use Elasticsearch to manage their inventory. An **index** called `car` is created. Within the car index, there exists a **type** for each car manufacturer. Within each type, there are multiple **documents** that represent the available models. The documents have **properties** such as VIN number, license plate number, and color.

Similar indices for trucks, SUVs, and vans may also be created. After the indices have been created and populated, **shards** and **replicas** are created according to the configuration set in the cluster. In this example, each index has three shards and one replica of each shard. The cluster has three nodes to distribute the shards across. See Figure 3.

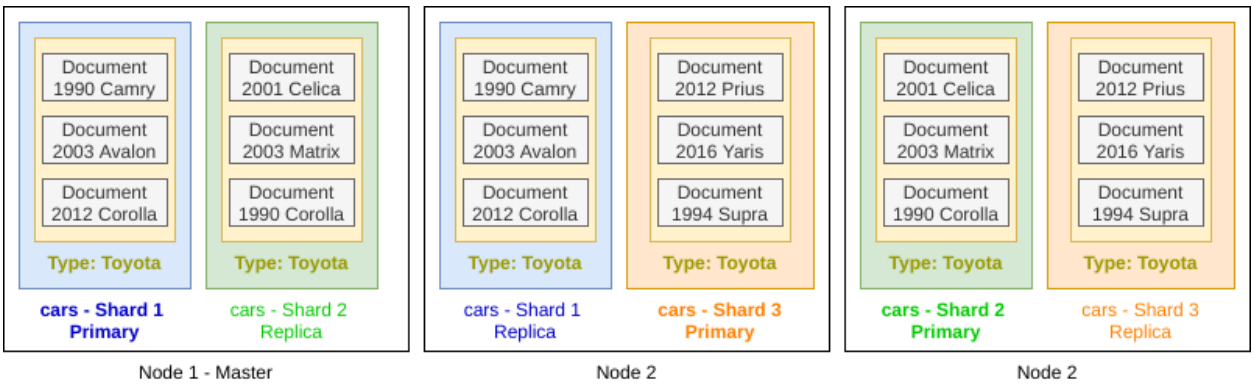

Figure 3: The `car` index distributed across three nodes.

Each shard contains its own set of documents from the index, in this case multiple models of cars available at the dealership. It's important to know that the shards do not overlap, and each shard is required for the complete index. Because of the replicas, any of the three nodes could go down unexpectedly and the entire index (via Shards 1, 2, and 3) would still be available. This is one of the advantages to using a multi-node cluster with Elasticsearch.

1.3 Node Roles

There are many types of Elasticsearch nodes, but we are only going to use two: **Master-eligible nodes** and **Data nodes**.

Master-eligible nodes are responsible for index creation and deletion, as well as deciding which shards are allocated to which servers [4]. In the case of an unexpected outage on a Master node, the cluster has an election process that will designate a new master from the remaining master-eligible nodes. This presents the potential for split brain [5] within the cluster. Because of this, a special role called "Voting-only master-eligible" exists. This terminology can be confusing, because the node is actually *not* eligible to become the master, but exists only to resolve election conflicts. See the Elasticsearch docs for more info.

Data nodes store the shards and perform resource-intensive operations requested by the Master node.

1.4 RESTful APIs and curl

Elasticsearch has a RESTful API available for performing create, read, update, and delete (CRUD) actions on its documents. REST stands for Representational State Transfer [6] and is easily accessed using the command line tool `curl`. Curl does Internet transfers for resources specified as URLs using Internet protocols [7] and will be the primary method of creating and deleting Elasticsearch indices in this lab. See the example below for creating a document in the car index.

```
curl -XPOST "http://localhost:9200/car/toyota" -d'
  {
    "model": "Camry"
    "year": "2009"
    "color": "green"
  }'
```

2 Lab Instructions

This lab will be separated into two separate portions: installing and configuring Elasticsearch, and inserting and deleting data in the cluster.

The Elasticsearch cluster for this lab will consist of three nodes, each running on an independent virtual machine with its own IP address (your IP addresses may differ):

Table 2: Elasticsearch cluster overview.

Host Name	Role	IP Address
es-master-a	Master-Eligible Node	192.168.128.4
es-master-b	Master-Eligible Node	192.168.128.7
es-data-a	Data Node	192.168.128.8

In production clusters, the master nodes generally do not do any data processing. However, for the purposes of this lab, the master nodes will process data using the `node.data: true` option in `/etc/elasticsearch/elasticsearch.yml`.

2.0.1 Prerequisites

This lab will use Oracle VirtualBox as a hypervisor for three virtual machines. You are free to use different software if you prefer. You will also need 60 GB of free space on your drive and at least 8GB of memory.

2.0.2 Downloads

You will need Oracle VirtualBox installed on your computer.
`https://www.virtualbox.org/wiki/Downloads`
You also need an image of Ubuntu 18.04 LTS Server.
`https://ubuntu.com/download/server`

2.1 Part 1 - Cluster Setup

See Figure 4 for the cluster architecture.

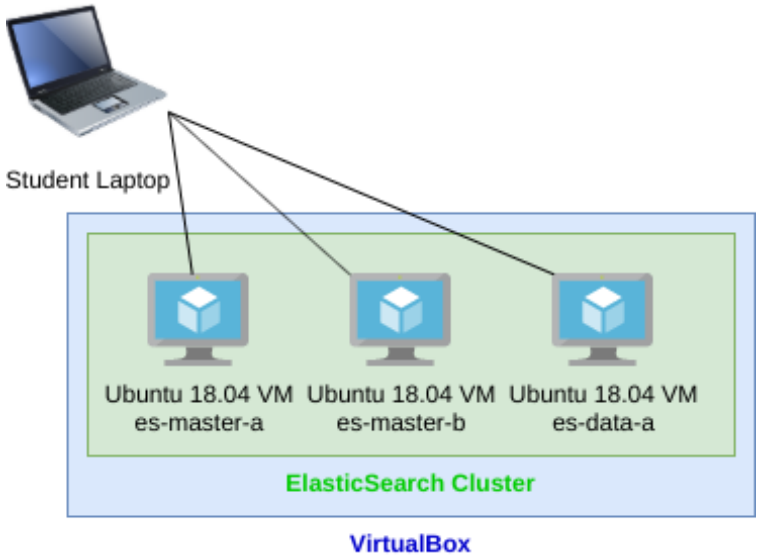

Figure 4: Elasticsearch cluster architecture.

2.1.1 Learning Objectives

There are many reasons an organization may want to employ Elasticsearch. Many use what is known as the Elastic Stack, which is a collection of three separate software packages called Logstash, Elasticsearch, and Kibana, to aggregate log information from client machines. This project will introduce you to the concepts of virtualization, installing a Linux operating system, and installing software required to configure an Elasticsearch cluster.

2.1.2 Virtual Machine Creation

1. Create an initial virtual machine with these attributes (leave **Create a virtual hard disk now** selected):
   - Name: `es-master-a`
   - Type: Linux
   - Version: Ubuntu (64-bit)
   - Memory Size: 2560 MB
   - File Size: 20.0 GB
2. Open the *Preferences* window found in the *File* menu and select *Network* on the sidebar. Click the icon to add a NAT [1] Network.
3. Set the Network Name to **dis** and the Network CIDR to **192.168.128.0/24**.
4. Open the Settings window for your newly created VM.

[1] Network Address Translation

5. Choose *Network* on the sidebar and select *NAT Network* from the *Attached to:* drop-down menu. Choose the **dis** NAT Network.

2.1.3 Install Ubuntu 18.04

1. Start the `es-master-a` virtual machine with the Start button.
2. Choose the Ubuntu image file you downloaded previously when prompted.
3. Choose the default options during the installer. When you get to the *Profile setup* dialog, enter these details:
   - Your name: **dis**
   - Your server's name: **es-master-a**
   - Pick a username: **dis**
   - Choose a password: **dislab2020**
4. Do not enable ssh or install any suggested server snaps.
5. Reboot when prompted.

2.1.4 Install Elasticsearch

1. Log in to the machine using the username and password you set during OS installation (`dis` and `dislab2020`).
2. Update the apt repository: `sudo apt update`
3. Install the `apt-transport-https` package necessary to access a repository over HTTPS: `sudo apt install apt-transport-https`
4. Install OpenJDK 8: `sudo apt install openjdk-8-jdk`
5. Import the Elasticsearch's repository's GNU Privacy Guard (GPG) [2] key with `wget`:

   ```
   wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
   ```

   You should see an OK message.
6. Add the Elasticsearch repository:

   ```
   sudo sh -c 'echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" > /etc/apt/sources.list.d/elastic-7.x.list'
   ```
7. Use `cat` to verify the string `deb https://artifacts.elastic.co/packages/7.x/apt stable main` is present in the file `/etc/apt/sources.list.d/elastic-7.x.list`.

   ```
   cat /etc/apt/sources.list.d/elastic-7.x.list
   ```

[2] GnuPG is a complete and free implementation of the OpenPGP standard as defined by RFC4880 (also known as PGP) [8]

8. Update the repository information again with `sudo apt update` and then install Elasticsearch with `sudo apt install elasticsearch`

2.1.5 Initial Elasticsearch config

1. Open the Elasticsearch config file using the text editor of your choice.

   ```
   sudo vim /etc/elasticsearch/elasticsearch.yml
   ```

2. Note that most of the lines in this file are commented. Find the lines listed below, uncomment them if needed, and change their values as shown. The lines after the line break beginning with `node.` will need to be added yourself.

   ```
   cluster.name: discluster
   node.name: es-master-a
   # You need to run ifconfig to find your node's IP address
   network.host: 192.168.128.4
   discovery.seed_hosts: ["192.168.128.4", "192.168.128.7", "192.168.128.8"]
   cluster.initial_master_nodes: ["192.168.128.4", "192.168.128.8"]

   node.master: true
   node.voting_only: false
   node.data: true
   node.ingest: true
   node.ml: false
   xpack.ml.enabled: false
   cluster.remote.connect: false
   ```

2.1.6 Clone es-master-a

1. Power off `es-master-a`.
2. Right click `es-master-a` in the sidebar.
3. Name the clones `es-master-b` and `es-data-a`.
4. Use the *Generate new MAC addresses for all network adapters* option when cloning.
5. Power on `es-master-a`, `es-master-b`, and `es-data-a`.

2.1.7 Configure Elasticsearch on the new nodes

1. You will notice the host name on both clones is still `es-master-a`. Run `sudo hostnamectl set-hostname es-master-b` and `sudo hostnamectl set-hostname es-data-a`.
2. Run `sudo truncate -s 0 /etc/machine-id` on both clones and reboot.

3. Make a note of the IP addresses on the clones after the reboot. Go to the cluster array lines in the Elasticsearch configuration file and make sure the IP addresses match the three VMs.

4. The configuration file will still be present on the clones, but you must change the following lines to be unique for each node.

```
# es-master-b
node.name: es-master-b
# Use ifconfig to find IP address
network.host: 192.168.128.7

# es-data-a
node.name: es-data-a
# Use ifconfig to find IP address
network.host: 192.168.128.8
node.voting_only: true
```

5. Start and enable the Elasticsearch service on each of the virtual machines:

```
sudo systemctl enable elasticsearch
sudo systemctl start elasticsearch
```

6. If there is no output, the service started successfully. You can check with `systemctl status elasticsearch`.

2.1.8 Submission

1. Take a screenshot with your name and student ID visible of the following commands to verify the cluster formation was successful. You will need to use the host's IP address on each VM, e.g. 192.168.128.4 on es-master-a, etc. See example below. Send this screenshot to your TA.

```
curl -XGET "http://192.168.128.4:9200/_cluster/health?pretty"
curl -XGET "http://192.168.128.4:9200/_cat/nodes?pretty"
```

```
dis@es-master-a:~$ echo "0123456 Student Name"
0123456 Student Name
dis@es-master-a:~$ curl -XGET "http://192.168.128.4:9200/_cluster/health?pretty"
{
  "cluster_name" : "discluster",
  "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
}
dis@es-master-a:~$ curl -XGET "http://192.168.128.4:9200/_cat/nodes?pretty"
192.168.128.7  6 79 2 0.02 0.14 0.12 dim - es-master-b
192.168.128.4 11 80 1 0.00 0.07 0.07 dim * es-master-a
192.168.128.8 10 79 1 0.00 0.09 0.09 dim - es-data-a
dis@es-master-a:~$
```

Figure 5: Example of lab submission screenshot - es-master-a.

```
dis@es-master-b:~$ echo "0123456 Student Name"
0123456 Student Name
dis@es-master-b:~$ curl -XGET "http://192.168.128.7:9200/_cluster/health?pretty"
{
  "cluster_name" : "discluster",
  "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
}
dis@es-master-b:~$ curl -XGET "http://192.168.128.7:9200/_cat/nodes?pretty"
192.168.128.8 10 79 0 0.00 0.08 0.09 dim - es-data-a
192.168.128.4 12 80 0 0.00 0.06 0.07 dim * es-master-a
192.168.128.7  7 80 0 0.02 0.13 0.12 dim - es-master-b
dis@es-master-b:~$ _
```

Figure 6: Example of lab submission screenshot - es-master-b.

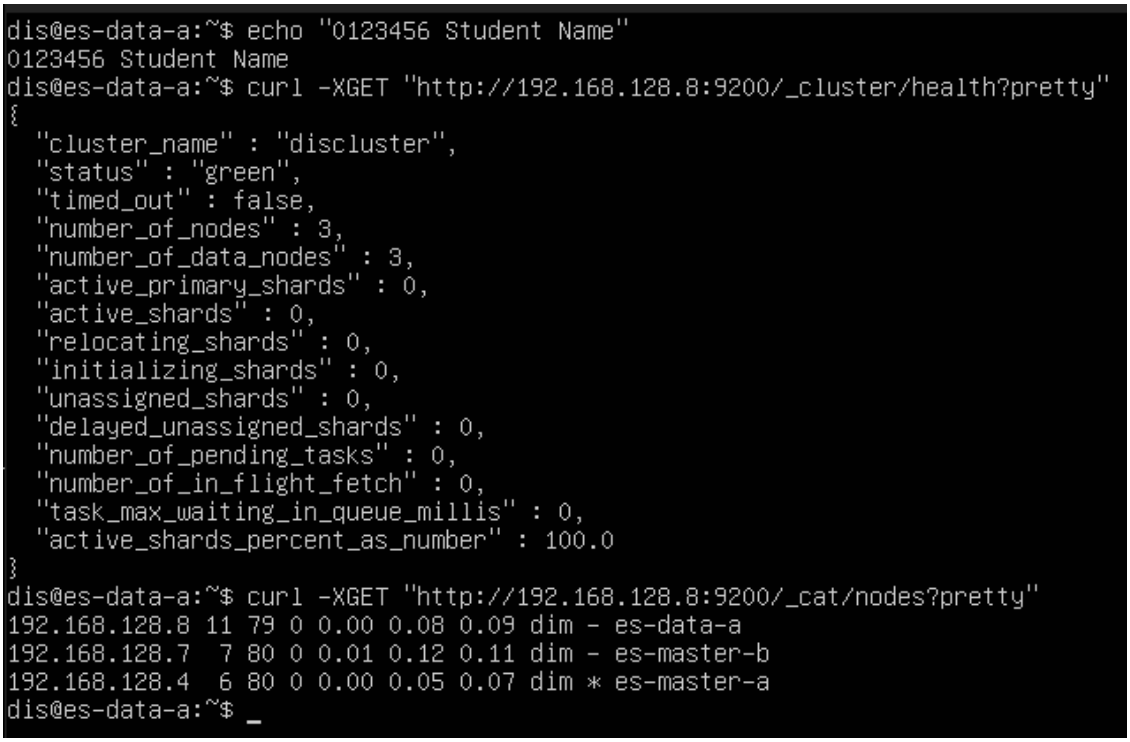

Figure 7: Example of lab submission screenshot - es-data-a.

2. Demonstrate the running cluster to your TA by running the following commands with successful return codes:

```
sudo systemctl status elasticsearch
curl -XGET "http://192.168.128.4:9200/_cluster/health?pretty"
curl -XGET "http://192.168.128.4:9200/_cat/nodes?pretty"
```

2.2 Part 2 - Inserting and Deleting Elasticsearch Data

2.2.1 Learning Objectives

The benefits of using a NoSQL database such as Elasticsearch are being free to store unstructured data. In this section, we will use the `curl` tool to perform HTTP requests such as `GET` and `POST` on our Elasticsearch cluster. We will also observe how nodes react when a member of the cluster is powered off unexpectedly.

3 Conclusion

References

[1] Elastic, "What is elasticsearch?," January 2020. [Online]. Available: `https://www.elastic.co/what-is/elasticsearch`. [Accessed January 28, 2020].

[2] Zachary Tong, "What is an elasticsearch index?," February 2013. [Online]. Available: https://www.elastic.co/blog/what-is-an-elasticsearch-index. [Accessed January 26, 2020].

[3] Elastic, "Scalability and resilience: clusters, nodes, and shards," December 2019. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/scalability.html. [Accessed January 28, 2020].

[4] Elastic, "Node," January 2020. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-node.html. [Accessed January 26, 2020].

[5] Adam Vanderbush, "Avoiding the split brain problem in elasticsearch," June 2017. [Online]. Available: https://qbox.io/blog/split-brain-problem-elasticsearch. [Accessed February 2, 2020].

[6] "Rest api tutorial," February 2020. [Online]. Available: https://restfulapi.net/. [Accessed February 4, 2020].

[7] Daniel Stenberg, *Everything curl*. February 2020. [Online]. Available: https://ec.haxx.se/. [Accessed February 4, 2020].

[8] "Gnupg," January 2020. [Online]. Available: https://gnupg.org. [Accessed February 8, 2020].

[9] Elastic, "Adding nodes to your cluster," December 2019. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/add-elasticsearch-nodes.html. [Accessed January 26, 2020].