SETUP AN ELASTICSEARCH CLUSTER ON AWS EC2

Here we explain how to setup an ElasticSearch 6.x (ES) cluster on Amazon EC2.

The main difference between Amazon and non-Amazon is Amazon considers unicast to be a security weakness, since it broadcasts the existence of servers across the network. So they have their own mechanism for node discovery, the ElasticSearch EC2 Discovery Plugin. With Amazon, you use ec2 instead of zen, which is the ElasticSearch default.

Prerequisites and Installation
You will obviously need at least two Amazon EC2 instances to make a cluster. Here we use Ubuntu 16.04, but the instructions will be the same for most Linux distributions.

First, you need Java:

```
sudo apt install default-jdk
```

Then download and install ElasticSearch.

```
wget
https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.4.2.deb
sudo dpkg -i elasticsearch-6.4.2.deb
sudo apt-get install -f
```

Next install the ElasticSearch EC2 Discovery Plugin.

```
cd /usr/share/elasticsearch/bin

sudo ./elasticsearch-plugin install discovery-ec2
```

Now set the JVM memory options to at least ½ of the memory on the machine. ES says anything less than that will result in poor performance.

```
sudo vim /etc/elasticsearch/jvm.options
```

On an 8 gb machine, 4 is half:

```
-Xms4g
-Xmx4g
```

Use Amazon Security rules to open Firewall Ports 9200 through 9300. ES uses those to communicate with each other and for http, by which you communicate with it.

Paris

/etc/elasticsearch/elasticsearch.yml

Next, I have set up two Amazon servers, one named Paris and the other named Paris2.

On each server edit the file /etc/elasticsearch/elasticsearch.yml.

In the following steps you will want to use the private IP address, meaning the one attached to the eth0 driver shown when you run ifconfig. This is not the same public IP address that you use to ssh to the machine. That is shown in the AWS console. ES will bind to that address, meaning any localhost commands will have to use that address instead.

In the example below, paris is the master node and paris 2 is a data node. The config items are:

cluster.name	Any name. Must be the same across all nodes.
node.name	Any name.
node.master	Whether or not this is a master node, meaning one that coordinates the activity of data nodes. In our example, we have 1 master and 2 data nodes. The master can also serve as a data node.
node.data	Indicates whether this node can store data.
node.ingest	Used to pre-process documents before they are indexed.
discovery.zen.hosts_provider	Use ec2 for node discovery.
discovery.zen.ping.unicast.hosts	List the IP address of all master and data nodes in the cluster. Enclose in commas.

network.host	The private IP address of this machine, i.e., the one shown by the ifconfig -a command. This is **not** the public address that you use to ssh into the machine from your laptop.

```
cluster.name: paris
node.name: paris
node.master: true
node.data: true
node.ingest: true
discovery.zen.hosts_provider: ec2
discovery.zen.ping.unicast.hosts:
network.host:
```

Paris2 /etc/elasticsearch/elasticsearch.yml
Here is the Paris 2 config file. The only difference is the network.host and node.master.

```
cluster.name: paris
node.name: paris2
node.master: false
node.data: true
node.ingest: true
discovery.zen.hosts_provider: ec2
discovery.zen.ping.unicast.hosts:
network.host:
```

Now start the servers. Use the service command. You cannot run ES as root, so this runs ES as user elasticsearch.

```
sudo service elasticsearch start
sudo service elasticsearch status
```

If you want to look at logs there are here, where paris.log is the name I assigned to the cluster.

```
sudo more /var/log/elasticsearch/paris.log
```

Now you can check the cluster to see how many nodes there are. Since this is a cluster we want a number > 1. Status should be green.

The status could be yellow meaning you have shards (i.e., portions of the index) not assigned to any node. A shard is a subdivision of indexes, this allowing an index to span more than 1 machine.

```
curl -XGET http://172.31.46.15:9200/_cluster/health?pretty
{
  "cluster_name" : "paris",
  "status" : "green",
  "timed_out" : false,
  "number_of_nodes" : 2,
  "number_of_data_nodes" : 2,
  "active_primary_shards" : 15,
```

```
  "active_shards" : 30,
  "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
}
```