

# LAB – ELASTICSEARCH

**Pre-Requisite:**

1. Install JDK 1.6 or higher in your machine.
2. Install curl in your machine. If you are using windows machine download the curl from the below link:

<https://curl.haxx.se/download.html>

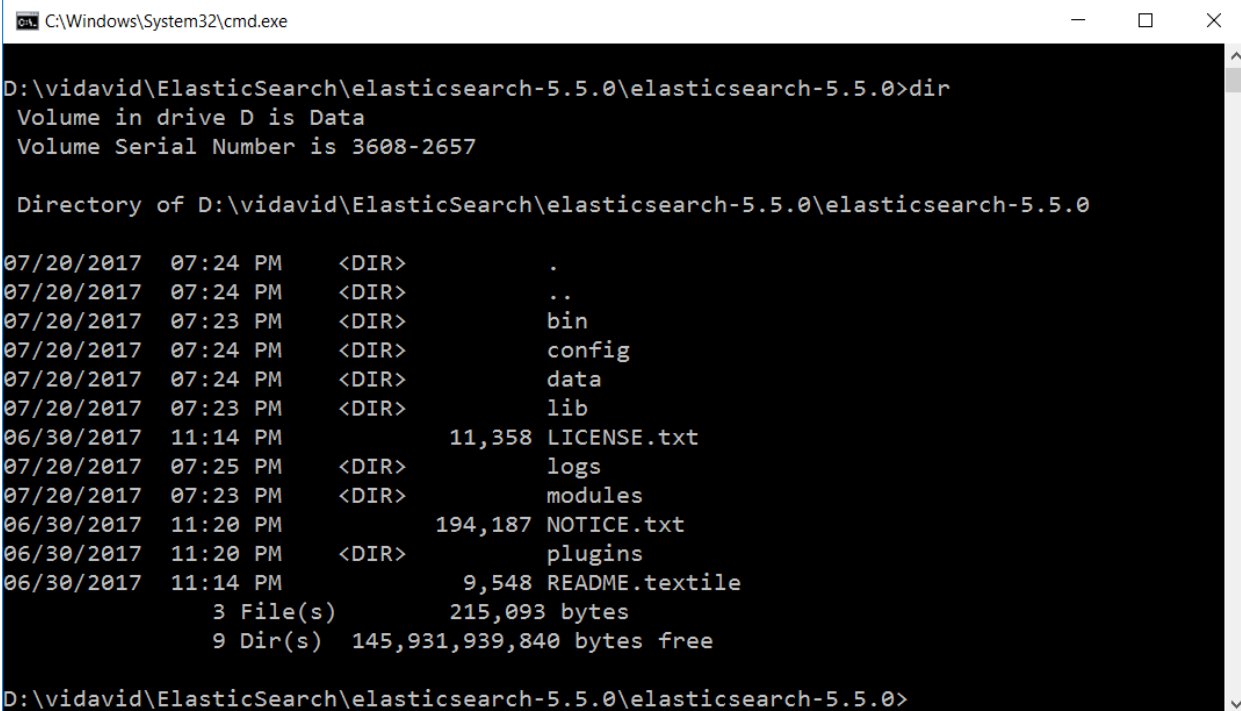
Copy the curl.exe into "C:\Windows" path. Now you can use curl in your command prompt itself.

3. Download Elasticsearch from below link: (I have downloaded "elasticsearch-5.5.0" version throughout my demo)

<https://www.elastic.co/downloads/elasticsearch>

Extract the zip. Locate the bin directory where you can see bat file list.

- a. You can open the command prompt which is locating the Elasticsearch installation directory



```
C:\Windows\System32\cmd.exe
D:\vidavid\ElasticSearch\elasticsearch-5.5.0\elasticsearch-5.5.0>dir
Volume in drive D is Data
Volume Serial Number is 3608-2657

Directory of D:\vidavid\ElasticSearch\elasticsearch-5.5.0\elasticsearch-5.5.0

07/20/2017  07:24 PM    <DIR>          .
07/20/2017  07:24 PM    <DIR>          ..
07/20/2017  07:23 PM    <DIR>          bin
07/20/2017  07:24 PM    <DIR>          config
07/20/2017  07:24 PM    <DIR>          data
07/20/2017  07:23 PM    <DIR>          lib
06/30/2017  11:14 PM             11,358 LICENSE.txt
07/20/2017  07:25 PM    <DIR>          logs
07/20/2017  07:23 PM    <DIR>          modules
06/30/2017  11:20 PM             194,187 NOTICE.txt
06/30/2017  11:20 PM    <DIR>          plugins
06/30/2017  11:14 PM             9,548 README.textile
               3 File(s)              215,093 bytes
               9 Dir(s)  145,931,939,840 bytes free

D:\vidavid\ElasticSearch\elasticsearch-5.5.0\elasticsearch-5.5.0>
```

- b. Use below command to start Elasticsearch cluster.

```
.\bin\elasticsearch
```

This will start the elastic search with default cluster name, as you see in the below window: the default node name here is pBtoz7C.

You can kill the below terminal window by press **Ctrl + C**

```

Elasticsearch 5.5.0
[2017-07-27T15:55:58,850][INFO ][o.e.n.Node] JVM arguments [-Xms2g, -Xmx2g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancyFraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -XX:+DisableExplicitGC, -XX:+AlwaysPreTouch, -Xss1m, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -Djdk.io.permissionsUseCanonicalPath=true, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.disable.jmx=true, -Dlog4j.skipJansi=true, -XX:+HeapDumpOnOutOfMemoryError, -Delasticsearch, -Des.path.home=D:\vidavid\ElasticSearch\elasticsearch-5.5.0\elasticsearch-5.5.0]
[2017-07-27T15:56:00,613][INFO ][o.e.p.PluginsService] [pBtoz7C] loaded module [aggs-matrix-stats]
[2017-07-27T15:56:00,615][INFO ][o.e.p.PluginsService] [pBtoz7C] loaded module [ingest-common]
[2017-07-27T15:56:00,618][INFO ][o.e.p.PluginsService] [pBtoz7C] loaded module [lang-expression]
[2017-07-27T15:56:00,621][INFO ][o.e.p.PluginsService] [pBtoz7C] loaded module [lang-groovy]
[2017-07-27T15:56:00,627][INFO ][o.e.p.PluginsService] [pBtoz7C] loaded module [lang-mustache]
[2017-07-27T15:56:00,630][INFO ][o.e.p.PluginsService] [pBtoz7C] loaded module [lang-painless]
[2017-07-27T15:56:00,635][INFO ][o.e.p.PluginsService] [pBtoz7C] loaded module [parent-join]
[2017-07-27T15:56:00,640][INFO ][o.e.p.PluginsService] [pBtoz7C] loaded module [percolator]
[2017-07-27T15:56:00,644][INFO ][o.e.p.PluginsService] [pBtoz7C] loaded module [reindex]
[2017-07-27T15:56:00,648][INFO ][o.e.p.PluginsService] [pBtoz7C] loaded module [transport-netty3]
[2017-07-27T15:56:00,652][INFO ][o.e.p.PluginsService] [pBtoz7C] loaded module [transport-netty4]
[2017-07-27T15:56:00,661][INFO ][o.e.p.PluginsService] [pBtoz7C] no plugins loaded
[2017-07-27T15:56:03,262][INFO ][o.e.d.DiscoveryModule] [pBtoz7C] using discovery type [zen]
[2017-07-27T15:56:04,049][INFO ][o.e.n.Node] [pBtoz7C] initialized
[2017-07-27T15:56:04,051][INFO ][o.e.n.Node] [pBtoz7C] starting ...
[2017-07-27T15:56:04,701][INFO ][o.e.t.TransportService] [pBtoz7C] publish_address {127.0.0.1:9300}, bound_addresses {127.0.0.1:9300}, {[::1]:9300}
[2017-07-27T15:56:07,794][INFO ][o.e.c.s.ClusterService] [pBtoz7C] new_master {pBtoz7C}{pBtoz7C}{pBtoz7C}{hHNgS1HLThSEKXNXAOvXNQ}{127.0.0.1}{127.0.0.1:9300}, reason: zen-disco-elected-as-master ([0] nodes joined)
[2017-07-27T15:56:08,187][INFO ][o.e.g.GatewayService] [pBtoz7C] recovered [2] indices into cluster_state
[2017-07-27T15:56:08,291][INFO ][o.e.h.n.Netty4HttpServerTransport] [pBtoz7C] publish_address {127.0.0.1:9200}, {[::1]:9200}
[2017-07-27T15:56:08,299][INFO ][o.e.n.Node] [pBtoz7C] started
[2017-07-27T15:56:10,121][INFO ][o.e.c.r.a.AllocationService] [pBtoz7C] Cluster health status changed from [RED] to [YELLOW] (reason: [shards started [[products][3]] ...]).

```

- If you want to create cluster with cluster name and node name you can use the below command:

```
.\bin\elasticsearch -Ecluster.name=capgemini_es -Enode.name=cap_node
```

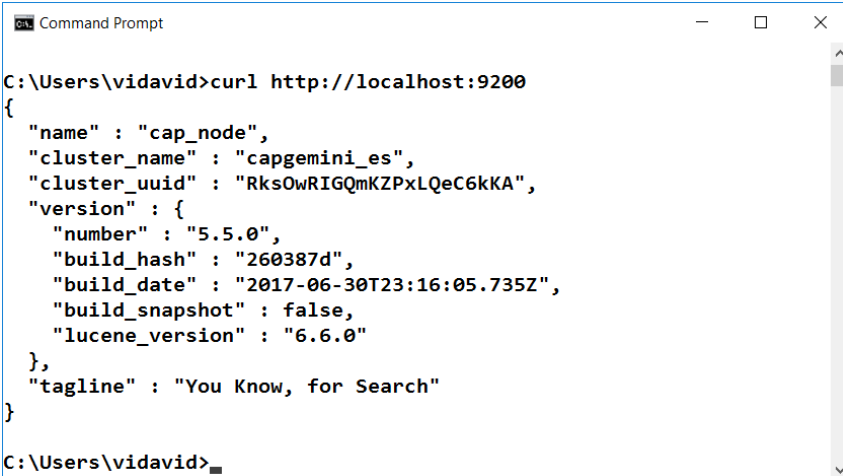
- Now you can see the below terminal with node details:

```
Elasticsearch 5.5.0
[2017-07-27T16:03:10,862][INFO ][o.e.n.Node] [cap_node] node name [cap_node], node ID [pBtoz7CbRqCeJPyceefM9A]
[2017-07-27T16:03:10,878][INFO ][o.e.n.Node] [cap_node] version[5.5.0], pid[3572], build[260387d/2017-06-30T23:16:05
.735Z], OS[Windows 8.1/6.3/amd64], JVM[Oracle Corporation/Java HotSpot(TM) 64-Bit Server VM/1.8.0_45/25.45-b02]
[2017-07-27T16:03:10,878][INFO ][o.e.n.Node] [cap_node] JVM arguments [-Xms2g, -Xmx2g, -XX:+UseConcMarkSweepGC, -XX:
CMSInitiatingOccupancyFraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -XX:+DisableExplicitGC, -XX:+AlwaysPreTouch, -Xss1m, -Djava.
awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -Djdk.io.permissionsUseCanonicalPath=true, -Dio.netty.noUnsafe=true, -D
io.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.disable
.jmx=true, -Dlog4j.skipJansi=true, -XX:+HeapDumpOnOutOfMemoryError, -Delasticsearch, -Des.path.home=D:\vidavid\ElasticSearch\elasti
csearch-5.5.0\elasticsearch-5.5.0]
[2017-07-27T16:03:11,972][INFO ][o.e.p.PluginsService] [cap_node] loaded module [aggs-matrix-stats]
[2017-07-27T16:03:11,974][INFO ][o.e.p.PluginsService] [cap_node] loaded module [ingest-common]
[2017-07-27T16:03:11,977][INFO ][o.e.p.PluginsService] [cap_node] loaded module [lang-expression]
[2017-07-27T16:03:11,979][INFO ][o.e.p.PluginsService] [cap_node] loaded module [lang-groovy]
[2017-07-27T16:03:11,981][INFO ][o.e.p.PluginsService] [cap_node] loaded module [lang-mustache]
[2017-07-27T16:03:11,984][INFO ][o.e.p.PluginsService] [cap_node] loaded module [lang-painless]
[2017-07-27T16:03:11,986][INFO ][o.e.p.PluginsService] [cap_node] loaded module [parent-join]
[2017-07-27T16:03:11,988][INFO ][o.e.p.PluginsService] [cap_node] loaded module [percolator]
[2017-07-27T16:03:11,991][INFO ][o.e.p.PluginsService] [cap_node] loaded module [reindex]
[2017-07-27T16:03:11,993][INFO ][o.e.p.PluginsService] [cap_node] loaded module [transport-netty3]
[2017-07-27T16:03:11,997][INFO ][o.e.p.PluginsService] [cap_node] loaded module [transport-netty4]
[2017-07-27T16:03:12,001][INFO ][o.e.p.PluginsService] [cap_node] no plugins loaded
[2017-07-27T16:03:13,914][INFO ][o.e.d.DiscoveryModule] [cap_node] using discovery type [zen]
[2017-07-27T16:03:14,720][INFO ][o.e.n.Node] [cap_node] initialized
[2017-07-27T16:03:14,722][INFO ][o.e.n.Node] [cap_node] starting ...
[2017-07-27T16:03:15,235][INFO ][o.e.t.TransportService] [cap_node] publish_address {127.0.0.1:9300}, bound_addresses {127.0.0.1
:9300}, {[::1]:9300}
[2017-07-27T16:03:18,289][INFO ][o.e.c.s.ClusterService] [cap_node] new_master {cap_node}{pBtoz7CbRqCeJPyceefM9A}{YvfpC0b1RQC5LH
r8R1cKJg}{127.0.0.1}{127.0.0.1:9300}, reason: zen-disco-elected-as-master ([0] nodes joined)
[2017-07-27T16:03:18,644][INFO ][o.e.h.n.Netty4HttpServerTransport] [cap_node] publish_address {127.0.0.1:9200}, bound_addresses {1
27.0.0.1:9200}, {[::1]:9200}
[2017-07-27T16:03:18,644][INFO ][o.e.n.Node] [cap_node] started
[2017-07-27T16:03:18,673][INFO ][o.e.g.GatewayService] [cap_node] recovered [2] indices into cluster_state
```

## Demo: Health Check Up

1. Once the cluster is up and running by default it will take 9200 port, you can use below command to check the cluster details:

```
curl http://localhost:9200
```



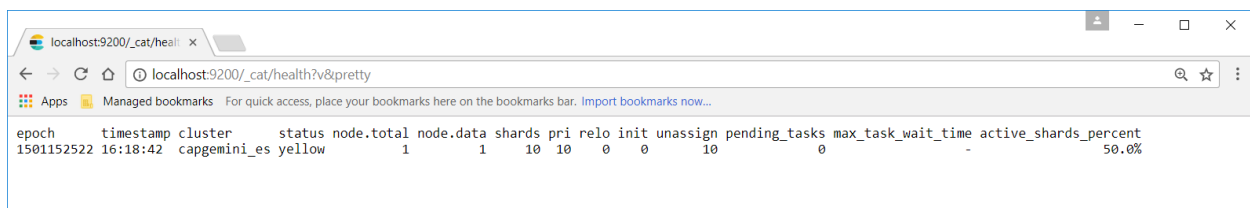
```

C:\Users\vidavid>curl http://localhost:9200
{
  "name" : "cap_node",
  "cluster_name" : "capgemini_es",
  "cluster_uuid" : "RksOwRIGQmKZPxLQeC6kKA",
  "version" : {
    "number" : "5.5.0",
    "build_hash" : "260387d",
    "build_date" : "2017-06-30T23:16:05.735Z",
    "build_snapshot" : false,
    "lucene_version" : "6.6.0"
  },
  "tagline" : "You Know, for Search"
}
C:\Users\vidavid>

```

2. To know the **health** details use the below command:

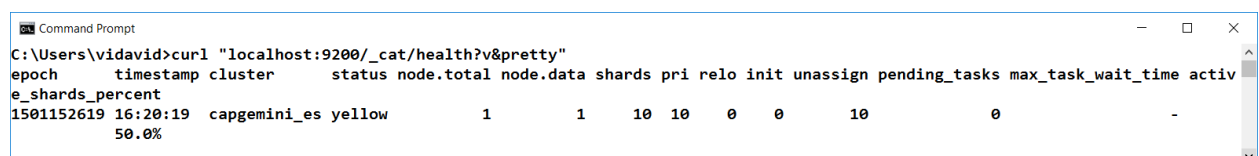
localhost:9200/\_cat/health?v&pretty



epoch	timestamp	cluster	status	node.total	node.data	shards	pri	relo	init	unassign	pending_tasks	max_task_wait_time	active_shards_percent
1501152522	16:18:42	capgemini_es	yellow	1	1	10	10	0	0	10	0	-	50.0%

3. You can also try with curl in your command prompt:

```
curl "localhost:9200/_cat/health?v&pretty"
```



```

C:\Users\vidavid>curl "localhost:9200/_cat/health?v&pretty"
epoch      timestamp  cluster      status node.total node.data shards pri relo init unassign pending_tasks max_task_wait_time activ
e_shards_percent
1501152619 16:20:19  capgemini_es yellow      1         1      10  10   0   0    10         0         -
50.0%

```

**Note :** I will use curl in my command prompt for REST API request

4. You can use below command to check node details:

```
curl "localhost:9200/_cat/nodes?v&pretty"
```

#### Demo: List All Indices:

1. To list all the available indices in your cluster use the below command:

```
curl -XGET "localhost:9200/_cat/indices?v&pretty"
```

#### Demo: Create an index:

2. To create new index use the below command:

```
curl -XPUT "localhost:9200/products?&pretty"  
curl -XGET "localhost:9200/_cat/indices?v&pretty"
```

```
curl -XPUT "localhost:9200/customers?&pretty"  
curl -XPUT "localhost:9200/orders?&pretty"  
  
curl -XGET "localhost:9200/_cat/indices?v&pretty"
```

By using the above command we have created products, customers and orders indices.

#### Demo: Index and Query a document

1. You can create/update document by using below command:

```
curl -XPUT 'localhost:9200/products/mobiles/1?pretty' -d'  
{  
  "name": "iPhone 7",  
  "camera": "12MP",  
  "storage": "256GB",  
  "display": "4.7inch",  
  "battery": "1,960mAh",  
  "reviews": ["Incredibly happy after having used it for one week", "Best iPhone so far", "Very expensive,  
stick to Android"]  
}
```

Note: Single quote will not be parsed by your windows. Hence we have to use double quotes(""). For convenience I have used .json file to provide document.

```
curl -XPUT "localhost:9200/products/mobiles/1?pretty" -d @mobile.json
curl -XPUT "localhost:9200/products/mobiles/2?pretty" -d @mobile1.json
curl -XPUT "localhost:9200/products/mobiles/3?pretty" -d @mobile2.json
curl -XPUT "localhost:9200/products/laptops/1?pretty" -d @laptop.json
curl -XPUT "localhost:9200/products/laptops/2?pretty" -d @laptop1.json
curl -XPOST "localhost:9200/products/mobiles?pretty" -d @mobiles.json
curl -XGET "localhost:9200/_cat/indices?v&pretty"
```

#### Demo: Fetching whole and partial documents

```
curl -XGET "localhost:9200/products/mobiles/1?pretty"
curl -XGET "localhost:9200/products/laptops/1?pretty"
curl -XGET "localhost:9200/products/laptops/10?pretty"
```

#### Partial documents

```
curl -XGET "localhost:9200/products/mobiles/1?pretty&_source=false"
curl -XGET "localhost:9200/products/mobiles/1?pretty&_source=name,reviews"
curl -XGET "localhost:9200/products/mobiles/1?pretty&_source=name,reviews,storage"
curl -XGET "localhost:9200/products/laptops/1?pretty&_source=name,RAM,storage"
```

#### Demo: Updating whole and partial documents

```
curl -XGET "localhost:9200/products/mobiles/3?pretty"
curl -XPUT "localhost:9200/products/mobiles/3?pretty" -d @mobiles3.json
```

#### Updates using the \_update API with "doc"

```
curl -XGET "localhost:9200/products/mobiles/2?pretty"
curl -XPOST "localhost:9200/products/mobiles/2/_update?pretty" -d @doc.json
curl -XGET "localhost:9200/products/mobiles/2?pretty"
curl -XPOST "localhost:9200/products/mobiles/2/_update?pretty" -d @mobile_update.json
```

#### Updates using the `_update` API with "script"

```
curl -XPUT "localhost:9200/products/shoes/1?pretty" -d @shoes1.json
curl -XPUT "localhost:9200/products/shoes/2?pretty" -d @shoes2.json
curl -XGET "localhost:9200/products/shoes/1?pretty"
curl -XPOST "localhost:9200/products/shoes/1/_update?pretty" -d @shoes_update.json
curl -XGET "localhost:9200/products/shoes/1?pretty"
curl -XGET "localhost:9200/products/shoes/2?pretty"
curl -XPOST "localhost:9200/products/shoes/2/_update?pretty" -d @shoes_update1.json
curl -XGET "localhost:9200/products/shoes/2?pretty"
```

#### Demo: Deleting an index

```
curl -XDELETE "localhost:9200/products/mobiles/2?pretty"
curl -XGET "localhost:9200/products/mobiles/2?pretty"
curl -i -XHEAD "localhost:9200/products/mobiles/2?pretty"
curl -i -XHEAD "localhost:9200/products/mobiles/1?pretty"
curl -XGET "localhost:9200/_cat/indices?v&pretty"
curl -XDELETE "localhost:9200/customerss?pretty"
curl -XGET "localhost:9200/_cat/indices?v&pretty"
```

#### Demo: Bulk indexing documents

##### Multi-get

```
curl -XGET "localhost:9200/_mget?pretty" -d @get.json
curl -XGET "localhost:9200/products/_mget?pretty" -d @get1.json
curl -XGET "localhost:9200/products/laptops/_mget?pretty" -d @get2.json
```



**Index multiple documents**

```
curl -XPOST "localhost:9200/_bulk?pretty" --data-binary @bulk.json  
curl -XPOST "localhost:9200/products/_bulk?pretty" --data-binary @bulk.json  
curl -XPOST "localhost:9200/products/shoes/_bulk?pretty" --data-binary @shoes_bulk.json
```

**Auto-generate ids**

```
curl -XPOST "localhost:9200/products/shoes/_bulk?pretty" --data-binary @autoid.json
```

(or)

```
curl -XPOST "localhost:9200/products/shoes/_bulk?pretty" -d"  
{ \"index\" : {} }  
{ \"name\": \"Puma\", \"size\": 9, \"color\": \"black\" }  
{ \"index\" : {} }  
{ \"name\": \"New Balance\", \"size\": 8, \"color\": \"black\" }  
"
```

*Note: this may not work in windows, you can go with above command.*

**Bulk operations in one go (paste these one operation at a time)**

```
curl -XPOST "localhost:9200/products/shoes/_bulk?pretty" -H "Content-Type: application/json" -d"  
{ \"index\" : { \"_id\" : \"3\" } }  
{ \"name\": \"Puma\", \"size\": 9, \"color\": \"black\" }  
{ \"index\" : { \"_id\" : \"4\" } }  
{ \"name\": \"New Balance\", \"size\": 8, \"color\": \"black\" }  
{ \"delete\" : { \"_id\" : \"2\" } }  
{ \"create\" : { \"_id\" : \"5\" } }  
{ \"name\": \"Nike Power\", \"size\": 12, \"color\": \"black\" }  
{ \"update\" : { \"_id\" : \"1\" } }  
{ \"doc\" : { \"color\" : \"orange\" } }
```

"

## Demo: Bulk indexing documents from a JSON file

Create customers.json

```
curl -H "Content-Type: application/x-ndjson" -XPOST  
"localhost:9200/customers/personal/_bulk?pretty&refresh" --data-binary @"customers.json"  
curl -XGET "localhost:9200/_cat/indices?v&pretty"
```