Elasticsearch – Setup

Download the elasticsearch release from the Elasticsearch site:

**SEAL ID**: 2052625 – elastic – Elasticsearch v5.6.x **Lifecycle Status**: Maintain

<https://www.elastic.co/downloads/past-releases/elasticsearch-5-6-10>

**SEAL ID**: 2056189 – elastic – Elasticsearch v6.x **Lifecycle Status**: Research

<https://www.elastic.co/downloads/past-releases/elasticsearch-6-3-2>

I also recommend downloading the Kibana release to assist within the elasticsearch database.

**SEAL ID**: 2053881 – Elasticsearch Kibana v5.6.5 **Lifecycle Status**: Maintain

<https://www.elastic.co/downloads/past-releases/kibana-5-6-5>

**SEAL ID**: 2056015 – Elasticsearch Kibana v6.2.2 **Lifecycle Status**: Research

<https://www.elastic.co/downloads/past-releases/kibana-6-2-2>

**Install Elasticsearch:**

For Windows users, I recommend downloading the elasticsearch MSI file, as it provides a guided installation. In any case, you can follow the installation instructions at the bottom of the linked page.

If using the **MSI** file, I recommend installing as a service and setting to start automatically.

Using the **Zip**, you can install the service by navigating to the bin folder in a command prompt and executing the following commands:

elasticsearch-sevice.bat install

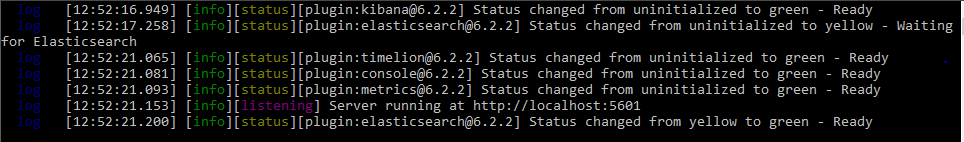
elasticsearch-service.bat start

the service can be stopped and uninstalled with the ‘stop’ and ‘remove’ properties. Using the zip, you may need to go to your services.msc to set to start automatically.

Once the elasticsearch has started, you can navigate to <http://localhost:9200> (port 9200 used by default) and you should see a JSON response describing your elasticsearch instance.

**Install Kibana:**

For kibana, ZIP is the only option. Firs extract the zip file to you desired location, then navigate to the bin folder and execute the kibana.bat file. Once executed, you should a console displaying a similar result:



You should be able to now navigate to <http://localhost:5601> (default: verify your address and port in the console (on the purple [listening] line)).

**Adding an Index (filestore) to elasticsearch:**

Before you can add data to an elasticsearch instance, you must create a container for all the records (called an index). You can create an index but sending an HTML PUT command to the elasticsearch instance. This can be done in Kibana, under Dev Tools > Console.

Type “PUT /product” and press the green play sign on the right of the input console. If successful, you will receive a success JSON message on the right. (“product” is the name of the index we are creating. You can name it anything you want, but I will use product for this explanation.)

You can also use a tool like postman to send a PUT command: <http://localhost:9200/product>

\*Sometimes Kibana has a difficult time loading with no indexes in the elasticsearch datastore, so setting up an index can resolve the issue, and the postman command makes this easier.

Note: Setting up restrictions, permissions, and mappings are outside the scope of this document, but the index setup can be complex and restrictive or very flexible (as it is in this basic setup). If you want to explore the index design and setup possibilities, please visit the documentation for your version: <https://www.elastic.co/guide/en/elasticsearch/reference/5.6/indices.html>

**Adding Data to an Index:**

There are several ways to add to the index, but I will discuss adding/updating an individual record and bulk adding several records.

Adding/updating a single record:

Using the REST API you can add (or update) a record to the index using the following command structure.

PUT /{index}/{type}/{id}

{

"{field1}" : "{value1}",

"{field2}" : "{value2}",

"{field3}" : "{value3}"

}

Using our product index, we could add the following record:

PUT /product/default/1001

{

"name": "Juice - V8 Tomato Kiwi Blend",

"price": 169,

"in\_stock": 4,

"sold": 3,

"tags": [

"Fruit"

],

"description": "Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Mauris viverra diam vitae quam. Suspendisse potenti. Nullam porttitor lacus at turpis. Donec posuere metus vitae ipsum.",

"is\_active": true,

"created": "2008/04/12"

}

Notice that we use the default {type}. Types will be going away in future versions of elasticsearch. It is a way that allows multiple types of documents to reside in the same index, but with the way the elasticsearch works, the storage size of each record will increase if multiple types are used in an index. This is due to the system reserving all the fields assigned to any items in an index for all documents. If we had 2 types with 3 different fields each, each record would technically have 6 fields (although the value would be null).



The next way to add records is through the bulk API and using cURL to read a file. We will use the above json data file to add 1000 records to the index. You my need to install cUrl to your computer to complete these next steps.

1. Copy and Paste the file to a location on your computer.
2. Open a command prompt and navigate to the location where you saved the file.
3. Type the following command into the console:

curl -H “Content-Type: application/json” -XPOST <http://localhost:9200/product/default/_bulk?pretty> --data-binary “@test-data.json”

1. At this point you should see a large amount of json pass your screen, and the you can now test that the data has been added.
   1. Visit <http://localhost:9200/product/default/500?pretty>
   2. You should see a JSON response ("name" : "Lobster - Tail 3 - 4 Oz") for item id 500.
2. If you have got this far, you should have sample data within your index to use in a programming environment.

I have included my REST Jetty server and demo using elasticsearch for one possible implementation method, inside the google drive location that you accessed this file in. <https://drive.google.com/open?id=1HT2bIt9TK1jj3HXs38OAvuFxk-FfLNUa>