Not mention

<http://www.elasticsearch.org/guide/en/elasticsearch/reference/1.4/search-request-highlighting.html>

<http://www.elasticsearch.org/guide/en/elasticsearch/reference/1.4/search-suggesters.html>

<http://www.elasticsearch.org/guide/en/elasticsearch/reference/1.4/search-percolate.html>

Install Marvel

./bin/plugin -i elasticsearch/marvel/latest

You probably don’t want Marvel to monitor your local cluster, so you can disable data collection with this command:

echo 'marvel.agent.enabled: false' >> ./config/elasticsearch.yml

./bin/elasticsearch

Add -d if you want to run it in the background as a daemon.

hen Elasticsearch is running in the foreground, you can stop it by pressing Ctrl-C, otherwise you can shut it down with the shutdown API:

curl -XPOST 'http://localhost:9200/\_shutdown'

Relational DB ⇒ Databases ⇒ Tables ⇒ Rows ⇒ Columns

Elasticsearch ⇒ Indices ⇒ Types ⇒ Documents ⇒ Fields

**Search with Query DSL**

request body is built with JSON, and uses a match query

GET /megacorp/employee/\_search

{

"query" : {

"match\_phrase" : {

"about" : "rock climbing"

}

},

"highlight": {

"fields" : {

"about" : {}

}

}

}

It is really just scratching the surface, and many features—such as suggestions, geolocation, percolation, fuzzy, and partial matching

### Retrieving part of a document

By default, a GET request will return the whole document, as stored in the \_source field. But perhaps all you are interested in is the title field. Individual fields can be requested using the \_sourceparameter. Multiple fields can be specified in a comma-separated list:

GET /website/blog/123?\_source=title,text

Or if you want **just** the \_source field without any metadata, you can use the \_source endpoint:

GET /website/blog/123/\_source

which returns just:

{

"title": "My first blog entry",

"text": "Just trying this out...",

"date": "2014/01/01"

}

POST /website/blog/

{ ... }

And the second uses the /\_create endpoint in the URL:

PUT /website/blog/123/\_create

{ ... }

POST /website/blog/1/\_update

{

"doc" : {

"tags" : [ "testing" ],

"views": 0

}

}

POST /\_bulk

{ "delete": { "\_index": "website", "\_type": "blog", "\_id": "123" }} 1

{ "create": { "\_index": "website", "\_type": "blog", "\_id": "123" }}

{ "title": "My first blog post" }

{ "index": { "\_index": "website", "\_type": "blog" }}

{ "title": "My second blog post" }

{ "update": { "\_index": "website", "\_type": "blog", "\_id": "123", "\_retry\_on\_conflict" : 3} }

{ "doc" : {"title" : "My updated blog post"} } 2

## Installing the ICU plugin

The [ICU analysis plugin](https://github.com/elasticsearch/elasticsearch-analysis-icu) for Elasticsearch uses the International Components for Unicode (ICU) libraries (see [site.project.org](http://site.icu-project.org/)) to provide a rich set of tools for dealing with Unicode. These include the icu\_tokenizer which is particularly useful for Asian languages, and a number of token filters which are essential for correct matching and sorting in all languages other than English.

### NOTE

The ICU plugin is an essential tool for dealing with languages other than English, and it is highly recommended that you install and use it. Unfortunately, because it is based on the external ICU libraries, different versions of the ICU plugin may not be compatible with previous versions. When upgrading, you may need to reindex your data.

To install the plugin, first shutdown your Elasticsearch node, then run the following command from the Elasticsearch home directory:

./bin/plugin -install elasticsearch/elasticsearch-analysis-icu/$VERSION

The current $VERSION can be found at [github.com/elasticsearch/elasticsearch-analysis-icu](https://github.com/elasticsearch/elasticsearch-analysis-icu).

Once installed, restart Elasticsearch, and you should see a line similar to the following in the startup logs:

[INFO][plugins] [Mysterio] loaded [marvel, analysis-icu], sites [marvel]

+

gateway.recover\_after\_nodes: 8

gateway.expected\_nodes: 10

gateway.recover\_after\_time: 5m

discovery.zen.ping.multicast.enabled: false

discovery.zen.ping.unicast.hosts: ["host1", "host2:port"]

The standard recommendation is to give 50% of the available memory to Elasticsearch heap, while leaving the other 50% free. It won’t go unused…Lucene will happily gobble up whatever is leftover.+

, try to avoid crossing the 32gb Heap boundary

GET /\_nodes/process fd count

Elasticsearch also uses a mix of NioFS and MMapFS for the various files. Ensure that the maximum map count so that there is ample virtual memory available for mmapped files. This can be set temporarily with:

sysctl -w vm.max\_map\_count=262144

Or permanently by modifying vm.max\_map\_count setting in your /etc/sysctl.conf