**Creation/Update/Deletion of documents within Index**

**Setup ElasticSearch in my PC**

1. Install Oracle Virtual Box
2. Get the recent most ubuntu image from online
3. Map the ubuntu image to the Virtual box and then launch the linux
4. After a series of installation steps ubuntu will be up

**Create a ssh connection and use the elasticsearch**

1. Install openssh server in ubuntu using the command: sudo apt-get install openssh-server
   1. Type password
   2. Press Y

1. Open Putty in windows and connect to 127.0.0.1. Give the username and password and there you go elastisearch can be used.

1. This is to resemble how you connect in real life to a server through ssh.

**Mapping Creation**

1. How to check if Elasticsearch is running
   1. curl 127.0.0.1:9200
2. Create a new mapping. Below loading the document into the index first lets create the mapping.

curl -XPUT 127.0.0.1:9200/movies -d '

> {

> "mappings": {

> "movie": {

> "\_all" : {"enabled": false}, -- We are disabling from index all the words because movie database wont be used to search for single words.

> "properties": {

> "year" : {

> "type" : "date" -- We are saying to interpret the year column as date.

> }

> }

> }

> }

> }'

{"acknowledged":true,"shards\_acknowledged":true,"index":"movies"}

1. How to see if the mapping is stored. Use the GET command and also use pretty after the index name so that you can see the json in indented format.

curl -XGET 127.0.0.1:9200/movies/\_mapping/movie?pretty

{

"movies" : {

"mappings" : {

"movie" : {

"\_all" : {

"enabled" : false

},

"properties" : {

"year" : {

"type" : "date"

}

}

}

}

}

}

1. Just for knowing how to insert a document through CURL. Use the below command

curl -XPUT 127.0.0.1:9200/movies/movie/109487 -d'

> {

> "genre" : ["IMAX","Sci-Fi"],

> "title" : "Interstellar",

> "year" : 2014

> }'

{"\_index":"movies","\_type":"movie","\_id":"109487","\_version":1,"result":"created","\_shards":{"total":2,**"successful":1,"**failed":0},"created":true}

1. Retrieving all the documents in the index

curl -XGET 127.0.0.1:9200/movies/movie/\_search?pretty

{

"took" : 105,

"timed\_out" : false,

"\_shards" : {

"total" : 5,

"successful" : 5,

"skipped" : 0,

"failed" : 0

},

"hits" : {

"total" : 1,

"max\_score" : 1.0,

"hits" : [

{

"\_index" : "movies",

"\_type" : "movie",

"\_id" : "109487",

"\_score" : 1.0,

"\_source" : {

"genre" : [

"IMAX",

"Sci-Fi"

],

"title" : "Interstellar",

"year" : 2014

}

}

]

}

}

1. Importing More than one movie into the index using REST query over JSON
   1. Get the file into the ubuntu side. If the file is in any website you can use wget <http://media.sundog-soft.com/es/movies.json>
   2. Insert the file into the index using the curl command with the bulk command

curl -XPUT 127.0.0.1:9200/\_bulk?pretty --data-binary @movies.json

{

"took" : 10,

"errors" : true,

"items" : [

{

"create" : {

"\_index" : "movies",

"\_type" : "movie",

"\_id" : "135569",

"status" : 409,

"error" : {

"type" : "version\_conflict\_engine\_exception",

"reason" : "[movie][135569]: version conflict, document already exists (current version [1])",

"index\_uuid" : "h3ZlNoV3QsSm5adM1z1lKg",

"shard" : "3",

"index" : "movies"

}

}

},

{

"create" : {

"\_index" : "movies",

"\_type" : "movie",

"\_id" : "122886",

"status" : 409,

"error" : {

"type" : "version\_conflict\_engine\_exception",

"reason" : "[movie][122886]: version conflict, document already exists (current version [1])",

"index\_uuid" : "h3ZlNoV3QsSm5adM1z1lKg",

"shard" : "1",

"index" : "movies"

}

}

},

{

"create" : {

"\_index" : "movies",

"\_type" : "movie",

"\_id" : "109487",

"status" : 409,

"error" : {

"type" : "version\_conflict\_engine\_exception",

"reason" : "[movie][109487]: version conflict, document already exists (current version [1])",

"index\_uuid" : "h3ZlNoV3QsSm5adM1z1lKg",

"shard" : "1",

"index" : "movies"

}

}

},

{

"create" : {

"\_index" : "movies",

"\_type" : "movie",

"\_id" : "58559",

"status" : 409,

"error" : {

"type" : "version\_conflict\_engine\_exception",

"reason" : "[movie][58559]: version conflict, document already exists (current version [1])",

"index\_uuid" : "h3ZlNoV3QsSm5adM1z1lKg",

"shard" : "1",

"index" : "movies"

}

}

},

{

"create" : {

"\_index" : "movies",

"\_type" : "movie",

"\_id" : "1924",

"status" : 409,

"error" : {

"type" : "version\_conflict\_engine\_exception",

"reason" : "[movie][1924]: version conflict, document already exists ( current version [1])",

"index\_uuid" : "h3ZlNoV3QsSm5adM1z1lKg",

"shard" : "4",

"index" : "movies"

}

}

}

]

}

1. Check if all movies are loaded

curl -XGET 127.0.0.1:9200/movies/movie/\_search?pretty

{

"took" : 10,

"timed\_out" : false,

"\_shards" : {

"total" : 5,

"successful" : 5,

"skipped" : 0,

"failed" : 0

},

"hits" : {

"total" : 5,

"max\_score" : 1.0,

"hits" : [

{

"\_index" : "movies",

"\_type" : "movie",

"\_id" : "109487",

"\_score" : 1.0,

"\_source" : {

"genre" : [

"IMAX",

"Sci-Fi"

],

"title" : "Interstellar",

"year" : 2014

}

},

{

"\_index" : "movies",

"\_type" : "movie",

"\_id" : "122886",

"\_score" : 1.0,

"\_source" : {

"id" : "122886",

"title" : "Star Wars: Episode VII - The Force Awakens",

"year" : 2015,

"genre" : [

"Action",

"Adventure",

"Fantasy",

"Sci-Fi",

"IMAX"

]

}

},

{

"\_index" : "movies",

"\_type" : "movie",

"\_id" : "58559",

"\_score" : 1.0,

"\_source" : {

"id" : "58559",

"title" : "Dark Knight, The",

"year" : 2008,

"genre" : [

"Action",

"Crime",

"Drama",

"IMAX"

]

}

},

{

"\_index" : "movies",

"\_type" : "movie",

"\_id" : "135569",

"\_score" : 1.0,

"\_source" : {

"id" : "135569",

"title" : "Star Trek Beyond",

"year" : 2016,

"genre" : [

"Action",

"Adventure",

"Sci-Fi"

]

}

},

{

"\_index" : "movies",

"\_type" : "movie",

"\_id" : "1924",

"\_score" : 1.0,

"\_source" : {

"id" : "1924",

"title" : "Plan 9 from Outer Space",

"year" : 1959,

"genre" : [

"Horror",

"Sci-Fi"

]

}

}

]

}

}

1. How to update a document that is already loaded since we have already loaded it twice and showed the failure message.
   1. Check if the movie is there that you wanted to update.

curl -XGET 127.0.0.1:9200/movies/movie/109487?pretty

{

"\_index" : "movies",

"\_type" : "movie",

"\_id" : "109487",

**"\_version" : 1, --This is the initial version. Once you update a new version will be created**

"found" : true,

"\_source" : {

"genre" : [

"IMAX",

"Sci-Fi"

],

"title" : "Interstellar",

"year" : 2014

}

}

1. Post command should be used to any action.

curl -**XPOST** 127.0.0.1:9200/movies/movie/109487/\_**update** -d '

{ "doc": { "title": "Interstellar Alright!!!" } }'

{"\_index":"movies","\_type":"movie","\_id":"109487","\_version":2,"result":"updated","\_shards":{"total":2,"successful":1,"failed":0}}

1. Curl Command is now showing the recent most version

curl -XGET 127.0.0.1:9200/movies/movie/109487?pretty {

"\_index" : "movies",

"\_type" : "movie",

"\_id" : "109487",

"\_version" : 2,

"found" : true,

"\_source" : {

"genre" : [

"IMAX",

"Sci-Fi"

],

"title" : "Interstellar Alright!!!",

"year" : 2014

}

}

1. If we try to create the same id document again it will by default update the document and give the version as 3 now. The others cannot be retrieved because already they are marked for "deletion" so its not safe to get them back. When next cleanup is scheduled in elasticsearch, those will be wiped out.

1. Deletion of a document
   1. Search for dark knight movie first and get the id and use it for deletion

curl -XGET 127.0.0.1:9200/movies/movie/\_search?q=Dark

{"took":19,"timed\_out":false,"\_shards":{"total":5,"successful":5,"skipped":0,"failed":0},"hits":{"total":1,"max\_score":0.9808292,"hits":[{"\_index":"movies","\_type":"movie","\_id":"58559","\_score":0.9808292,"\_source":{ "id": "58559", "title" : "Dark Knight, The", "year":2008 , "genre":["Action", "Crime", "Drama", "IMAX"] }}]}}

1. Delete the actual document using the below command

curl -XDELETE 127.0.0.1:9200/movies/movie/58559?pretty

{

"found" : true,

"\_index" : "movies",

"\_type" : "movie",

"\_id" : "58559",

"\_version" : 2,

**"result" : "deleted", -- actually the deletion wont happen immediately. Its marked and later during cleanup sweep it will be wiped out**

"\_shards" : {

"total" : 2,

"successful" : 1,

"failed" : 0

}

}