Elasticsearch Commands

### Cluster Settings

* Set mmap
  + sysctl -w vm.max\_map\_count=262144
* [Optional] Set Min & Max Heap Size *(jvm.options)*
  + Xms512M
  + Xmx512M

### Cluster Info Retrieval

* Retrieve cluster info
  + curl -X GET "localhost:9200/?pretty"
  + GET \_cluster/state
  + GET \_cluster/health
  + curl -X GET "http://10.0.0.196:9200/\_cluster/stats?pretty"
  + Voting Configuration
    - http://MtShavano:9200/\_cluster/state?filter\_path=metadata.cluster\_coordination.last\_committed\_config
* Retrieve aggregated indices info
  + curl "localhost:9200/\_cat/indices?v=true"
* Retrieve node info
  + curl -X GET "localhost:9200/\_cat/nodes?pretty"
  + curl -X GET "localhost:9200/\_cat/nodes?v=true&pretty"
* Retrieve shard info
  + curl -X GET "localhost:9200/\_cat/shards?v&pretty"
  + curl -X GET "localhost:9200/\_cat/shards/*<index>*?v&pretty"
  + Shard Refresh  
    [*(Shard Refresh vs. Shard Flush)*](https://stackoverflow.com/questions/19963406/refresh-vs-flush)
    - ES refresh default: 1/sec
    - Force Refresh
      * curl -X POST "localhost:9200/*<index>*/\_refresh"
      * curl -X PUT "localhost:9200/my\_refresh\_test/\_doc/4?refresh=true" - force refresh @ doc index *(!recommended - perf issues)*
* Retrieve tasks
  + curl -X GET "localhost:9200/\_tasks?pretty"
  + curl -X GET "localhost:9200/\_tasks?nodes=es01&pretty"
* CAT data
  + curl -X GET "localhost:9200/\_cat
  + curl -X GET "localhost:9200/\_cat?v=true
  + curl -X GET "localhost:9200/\_cat/nodes?pretty&v=true&s=heap.percent:desc" - sorted output *(“s=<field> - descending in this ex)*
  + curl -X GET "localhost:9200/\_cat/nodes?pretty&v=true&s=node.role:desc,heap.percent:asc"
  + curl -X GET "localhost:9200/\_cat/aliases?v=true&pretty"
  + Can use “?v&h=...” for setting displayed headers
* Cluster Issues Debugging
  + curl -X GET "<http://10.0.0.196:9200/_cluster/stats?pretty>"
  + GET \_cluster/health?level=indices&pretty - find prob index(s)
  + GET \_cluster/health?level=shards&pretty - find prob shards
  + GET \_cat/shards/test4?v - find prob shards
  + GET \_cluster/allocation/explain - Explain shard allocations in cluster
  + GET \_cluster/allocation/explain {...}
* Cluster Monitoring
  + Enable monitoring
    - PUT /\_cluster/settings {"persistent" : { "xpack.monitoring.collection.enabled" : true }}

### Authentication

* curl statement w/authentication
  + curl --user *<user:password>* ...
  + curl --user elastic:password -X GET "http://10.0.0.196:9200/\_cat/nodes?v&pretty"

### Index

* [*Create index API*](https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-create-index.html)
  + PUT /*<index>*
* [*Index settings*](https://www.elastic.co/guide/en/elasticsearch/reference/current/index-modules.html#index-modules-settings)
  + [*Update Index settings API*](https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-update-settings.html)
    - PUT /*<index>*/\_settings {...}
* [*Close [open] index*](https://www.elastic.co/guide/en/elasticsearch/reference/7.11/indices-close.html)
  + POST *<index>*/\_close | \_open
* [*Reindex index API*](https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-reindex.html)
  + Copies (all | subset) docs src index → destination index  
    *(src & dest - can be index, alias, or data stream)*
  + POST \_reindex {...}
    - wait\_for\_completion=false
      * ES returns task id - track progress &/or cancel job
* [*Rollover index API*](https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-rollover-index.html)
* [*Refresh index API*](https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-refresh.html)
  + POST /*<index>*/\_refresh*<?query parms>*

##### ILM

* PUT /\_cluster/settings
* {
* "persistent" : {
* "indices.lifecycle.poll\_interval": "5s"
* }
* }

#### Mapping

* [*Mapping parms*](https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-params.html)
  + [*Properties*](https://www.elastic.co/guide/en/elasticsearch/reference/current/properties.html)
    - Type mappings, object fields, & nested fields contain properties
    - Properties: any [*data type*](https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-types.html)
* Metadata
  + Identity fields
    - \_id
      * Doc unique identifier
      * Accessible in queries
    - \_index
      * Used in queries - aggs, sorting, scripting
    - \_type
      * Deprecated??
  + Doc source fields
    - [*\_source*](https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-source-field.html)
      * Contains original JSON doc body
      * Can be disabled - ! recommended
      * Not indexed or searchable
    - \_size
      * Size of \_source
      * M/b enabled to access
  + \_meta
    - App-specific custom metadata
    - Retrieved via [*Get mapping API*](https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-get-mapping.html)
* [*Create index w/explicit mapping*](https://www.elastic.co/guide/en/elasticsearch/reference/current/explicit-mapping.html#create-mapping)
* [*Updata existing mapping*](https://www.elastic.co/guide/en/elasticsearch/reference/current/explicit-mapping.html#add-field-mapping)

### Analysis

* Analyze text
  + GET \_analyze { "text": "The United Kingdom" }

### Data Insertion

* Insert single datum
  + curl -X PUT "localhost:9200/customer/\_doc/1?pretty" -H 'Content-Type: application/json' -d'
  + {
  + "name": "John Doe"
  + }
  + '
  + Single Line
  + curl -X PUT "localhost:9200/customer/\_doc/1?pretty" -H 'Content-Type: application/json' -d' {"name": "John Doe"}'
* Insert bulk data
  + curl -H "Content-Type: application/json" -XPOST "localhost:9200/bank/\_bulk?pretty&refresh" --data-binary "@accounts.json"
  + curl -H "Content-Type: application/json" -XPOST "localhost:9200/denormalized\_blogs/\_bulk?pretty&refresh" --data-binary "@accounts.json"

### Update Data

* [*Update API*](https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-update.html)
  + Reindexes the doc “under-the-covers”
* POST /<index>/\_update/<\_id>
* **Note:** to replace a doc, use the index API

### Retrieve Data

* Retrieve mapping
  + curl -X GET "localhost:9200/*<index>*/\_mapping"
* Retrieve single datum
  + curl -X GET "localhost:9200/customer/\_doc/3?pretty"
* Retrieve data from an index *(\_search)*
  + Note: implicit \_search query is match\_all {}
  + curl -X GET "localhost:9200/*<index>*/\_search?pretty"
  + curl -X GET "localhost:9200/nested\_blogs/\_search?pretty"
* Retrieve data counts
  + GET blogs/\_count
* [*Term query*](https://www.elastic.co/guide/en/elasticsearch/reference/7.11/query-dsl-term-query.html)
  + Retrieve docs containing **exact** term in designated field
    - Precise values *(e.g. price, userID, prodID)*
* [*Match query*](https://www.elastic.co/guide/en/elasticsearch/reference/7.11/query-dsl-match-query.html)
  + Retrieve docs containing match’d text, number, date, or boolean
  + Perform full-text search *(incl fuzzy-matching)*
  + Query text is **analyzed** before matching
    - Can search text fields for analyzed tokens vs. exact term
    - match query - type boolean
      * operator parm *(defaults to “or”)*
        + Can be set to “and” to and the analyzed query string