Elasticsearch Search and analytics engine

CONCEPTS

Reading this sheet
Entries are METHOD path. Run them as curl -X <METHOD> $ES/<path>, with --json '<body>' when a body is given (curl 7.82+ sets the JSON headers), or paste them into Kibana Dev Tools. Set ES=localhost:9200 first.

Elasticsearch 8+ enables security by default: use ES=https://localhost:9200 and add -u elastic:<password> --cacert <http_ca.crt> to every curl.

Cluster nodes, one health status
- **Index** mapping + settings
  - **Shard** primary + replicas
    - **Document** JSON with an _id

Health: green = all shards allocated, yellow = replicas missing, red = a primary is missing.

New documents become searchable after the next refresh (about 1 s by default), not instantly.

CLUSTER AND CAT

GET /
Node name, cluster, version

GET _cluster/health?pretty
Green / yellow / red status

GET _cat/indices?v
Indices with docs and size

GET _cat/nodes?v
Nodes and load

GET _cat/shards/*<index>*?v
Where shards live

GET _cat/indices?v&s=store.size:desc
Sort cat output by a column

GET _cluster/allocation/explain
Why a shard is unassigned

INDICES

PUT *<index>*
Create an index

INDICES (cont.)

DELETE *<index>*
Delete an index and its data

GET *<index>*/_mapping
Show field mappings

GET *<index>*/_settings
Show index settings

PUT *<index>*/_mapping
Add fields to a mapping

PUT *<index>*/_settings
Change a dynamic setting

POST _aliases
Add or swap aliases atomically

DOCUMENTS

POST *<index>*/_doc
Index with a generated id

PUT *<index>*/_doc/*<id>*
Index or replace by id

GET *<index>*/_doc/*<id>*
Get a document

DELETE *<index>*/_doc/*<id>*
Delete a document

POST *<index>*/_update/*<id>*
Partial update: {"doc":{...}}

GET *<index>*/_count
Count documents

POST *<index>*/_delete_by_query
Delete all matching a query

SEARCH

GET *<index>*/_search?q=*<field>*:*<text>*
URI search for quick checks

POST *<index>*/_search
Search with a query body

{"match": {"*<field>*": "*<text>*"}}
Full-text match

{"term": {"*<field>*": "*<value>*"}}
Exact value (keyword, number)

{"bool": {"must": [], "filter": [], "must_not": []}}
Combine clauses

{"range": {"*<field>*": {"gte": 10, "lt": 20}}}
Range filter

SEARCH (cont.)

{"exists": {"field": "*<field>*"}}
Field is present

{"match_phrase": {"*<field>*": "*<text>*"}}
Exact phrase

"size": 20, "from": 40
Paginate

"sort": [{"*<field>*": "desc"}]
Sort results

"_source": ["*<field>*"]
Return only some fields

AGGREGATIONS

"size": 0, "aggs": {"by": {"terms": {"field": "*<f>*"}}}
Top values (terms)

{"avg": {"field": "*<field>*"}}
Metric: avg, min, max, sum

{"date_histogram": {"field": "*<f>*", "calendar_interval": "day"}}
Count per day

BULK AND REINDEX

POST _bulk
Many writes, NDJSON body

POST _reindex
Copy: {"source":{...},"dest":{...}}

RECIPES

Create index, bulk load, search

```
curl -X PUT $ES/books --json
'{"mappings": {"properties":
{"year": {"type": "integer"}}}}'
curl -X POST "$ES/books/_bulk?
refresh" \
  -H 'Content-Type: application/
x-ndjson' --data-binary
@books.ndjson
curl -s "$ES/books/_search?q=
title:dune&pretty"
```

Swap to a new index behind an alias

```
# alias books → books_v1
curl -X PUT $ES/books_v2 --json
@mapping.json
curl -X POST $ES/_reindex --json
\
  '{"source": {"index":
"books_v1"}, "dest": {"index":
"books_v2"}}'
curl -X POST $ES/_aliases --json
@swap.json
```