**Index templates:**

* Index templates specify settings and mappings
* They are applied to indices that match one or more patterns
* Pattern may include wildcards(\*)

**PUT /\_templates/name\_of\_index\_template**

Priorities:

A new index may match multiple index templates

Any order parameter can be used to define the priority of index templates

* The value is simply as an integer
* Templates with lower values are merged first

Updating an index template:

PUT /\_templates/access\_logs

{

#Full configuration

}

Index templates can be retrieved and detailed using GET and DELETE verbs with same endpoint respectively

**Elastic Common Schema (ECS):**

A specification of common fields and how they should be mapped.

ECS was created to overcome the challenges of having different field names for the same things.

It defines a lot of common fields in various groups these are known as field slots.

Here documents are referred as field sets

Here documents are referred as events

Mostly used for standard events

e.g. webserver logs, os metrics etc.,

**Dynamic mapping:**

Automatically assigns data types

If a number is given It is mapped it to “long” data type as it doesn’t know how much bigger a number can be so it assigns long data type

If we give text it will be assigned both as text and keyword data type as well

**String**  to text field with keyword mapping

**Integer** tolong

**Float** to float

Boolean to Boolean

Object to object

Array depends on the first non null value

**Combining explicit and dynamic mapping:**

The unmapped fields will be automatically mapped to the default mappings

**Configure dynamic mapping** :

In mapping field giving

“dynamic” : false

disables dynamic mapping

If a value is given without mapping it externally it will be started in the \_source but it will not be indexed i.e it cannot be searched.

New filed are ignored (not rejected)

Setting “dynamic” : “strict” would return an error while we assign a new field

We can also enable dynamic mapping to a particular field

Numeric detection

In mappings filed

“numeric\_detection” : true

Then it checks the strings for numeric values then it will set it to long or float

Disabling in mapping fields “date\_detection” : false

Configuring date format

“dynamic\_date\_formats” : [“dd-MM-yyyy”]

**Stemming and stop words:**

* Stemming reduces words to their form
* Stop words are the words that are filtered out during text analysis.

E.g. a, the, at, of, on, etc.,

Stemming analyser changes all to the root words

When a search query is performed then that word will be also changed to its root word and then searched.

**Built-in-analysers**:

Standard analyser

* Split by word
* Splits text words (standard tokenizer)
* Lower case filter

Contains the stop token filter (disabled by default)

Simple analyser:

* Splits when encountered other than letters.
* Lower case filter

White space analyser:

* Splits when white space
* Doesn’t contain lower case filter

Keyword analyser:

* Same as the exact input values are stored

Pattern analyser:

* A regular expression is used to match token separators
* It should match whatever should split
* The default pattern matches all non word characters
* Lowercases by default

OPEN & CLOSED INDICES:

An open index is available for indexing and search requests

A closed index will refuse requests

* Read and write requests are blocked
* Necessary for performing some operations

Dynamic and static settings:

* Dynamic settings can be changed without closing the index first
* Static settings requires the index to be closed first

Analysers can be updated

The documents may be using the old version of the analyser

Reindex these documents to avoid problems