

ELASTICSEARCH SYNC MODULE

New in version Kraken.

This sync module writes the metadata from other zones to **ElasticSearch**. As of luminous this is a json of data fields we currently store in ElasticSearch.

```
{
  "_index" : "rgw-gold-ee5863d6",
  "_type" : "object",
  "_id" : "34137443-8592-48d9-8ca7-160255d52ade.34137.1:object1:null",
  "_score" : 1.0,
  "_source" : {
    "bucket" : "testbucket123",
    "name" : "object1",
    "instance" : "null",
    "versioned_epoch" : 0,
    "owner" : {
      "id" : "user1",
      "display_name" : "user1"
    },
    "permissions" : [
      "user1"
    ],
    "meta" : {
      "size" : 712354,
      "mtime" : "2017-05-04T12:54:16.462Z",
      "etag" : "7ac66c0f148de9519b8bd264312c4d64"
    }
  }
}
```

ELASTICSEARCH TIER TYPE CONFIGURABLES

- endpoint

Specifies the Elasticsearch server endpoint to access

- num_shards (integer)

The number of shards that Elasticsearch will be configured with on data sync initialization. Note that this cannot be changed after init. Any change here requires rebuild of the Elasticsearch index and reinit of the data sync process.

- num_replicas (integer)

The number of the replicas that Elasticsearch will be configured with on data sync initialization.

- explicit_custom_meta (true | false)

Specifies whether all user custom metadata will be indexed, or whether user will need to configure (at the bucket level) what custom metadata entries should be indexed. This is false by default

- index_buckets_list (comma separated list of strings)

If empty, all buckets will be indexed. Otherwise, only buckets specified here will be indexed. It is possible to provide bucket prefixes (e.g., foo*), or bucket suffixes (e.g., *bar).

- approved_owners_list (comma separated list of strings)

If empty, buckets of all owners will be indexed (subject to other restrictions), otherwise, only buckets owned by specified owners will be indexed. Suffixes and prefixes can also be provided.

- override_index_path (string)

If not empty, this string will be used as the elasticsearch index path. Otherwise the index path will be determined and generated on sync initialization.

New in version Luminous.

Since the ElasticSearch cluster now stores object metadata, it is important that the ElasticSearch endpoint is not exposed to the public and only accessible to the cluster administrators. For exposing metadata queries to the end user itself this poses a problem since we'd want the user to only query their metadata and not of any other users, this would require the ElasticSearch cluster to authenticate users in a way similar to RGW does which poses a problem.

As of Luminous RGW in the metadata master zone can now service end user requests. This allows for not exposing the elasticsearch endpoint in public and also solves the authentication and authorization problem since RGW itself can authenticate the end user requests. For this purpose RGW introduces a new query in the bucket apis that can service elasticsearch requests. All these requests must be sent to the metadata master zone.

SYNTAX

GET AN ELASTICSEARCH QUERY

```
GET /{bucket}?query={query-expr}
```

request params:

- max-keys: max number of entries to return
- marker: pagination marker

expression := [(]<arg> <op> <value> [)] [<and|or> ...]

op is one of the following: <, <=, ==, >=, >

For example

```
GET /?query=name==foo
```

Will return all the indexed keys that user has read permission to, and are named 'foo'.

Will return all the indexed keys that user has read permission to, and are named 'foo'.

The output will be a list of keys in XML that is similar to the S3 list buckets response.

CONFIGURE CUSTOM METADATA FIELDS

Define which custom metadata entries should be indexed (under the specified bucket), and what are the types of these keys. If explicit custom metadata indexing is configured, this is needed so that rgw will index the specified custom metadata values. Otherwise it is needed in cases where the indexed metadata keys are of a type other than string.

```
POST /{bucket}?mdsearch
x-amz-meta-search: <key [; type]> [, ...]
```

Multiple metadata fields must be comma separated, a type can be forced for a field with a ;. The currently allowed types are string(default), integer and date

eg. if you want to index a custom object metadata x-amz-meta-year as int, x-amz-meta-date as type date and x-amz-meta-title as string, you'd do

```
POST /mybooks?mdsearch
x-amz-meta-search: x-amz-meta-year;int, x-amz-meta-release-date;date, x-amz-meta-title;string
```

DELETE CUSTOM METADATA CONFIGURATION

Delete custom metadata bucket configuration.

DELETE /<bucket>?mdsearch

GET CUSTOM METADATA CONFIGURATION

Retrieve custom metadata bucket configuration.

GET /<bucket>?mdsearch