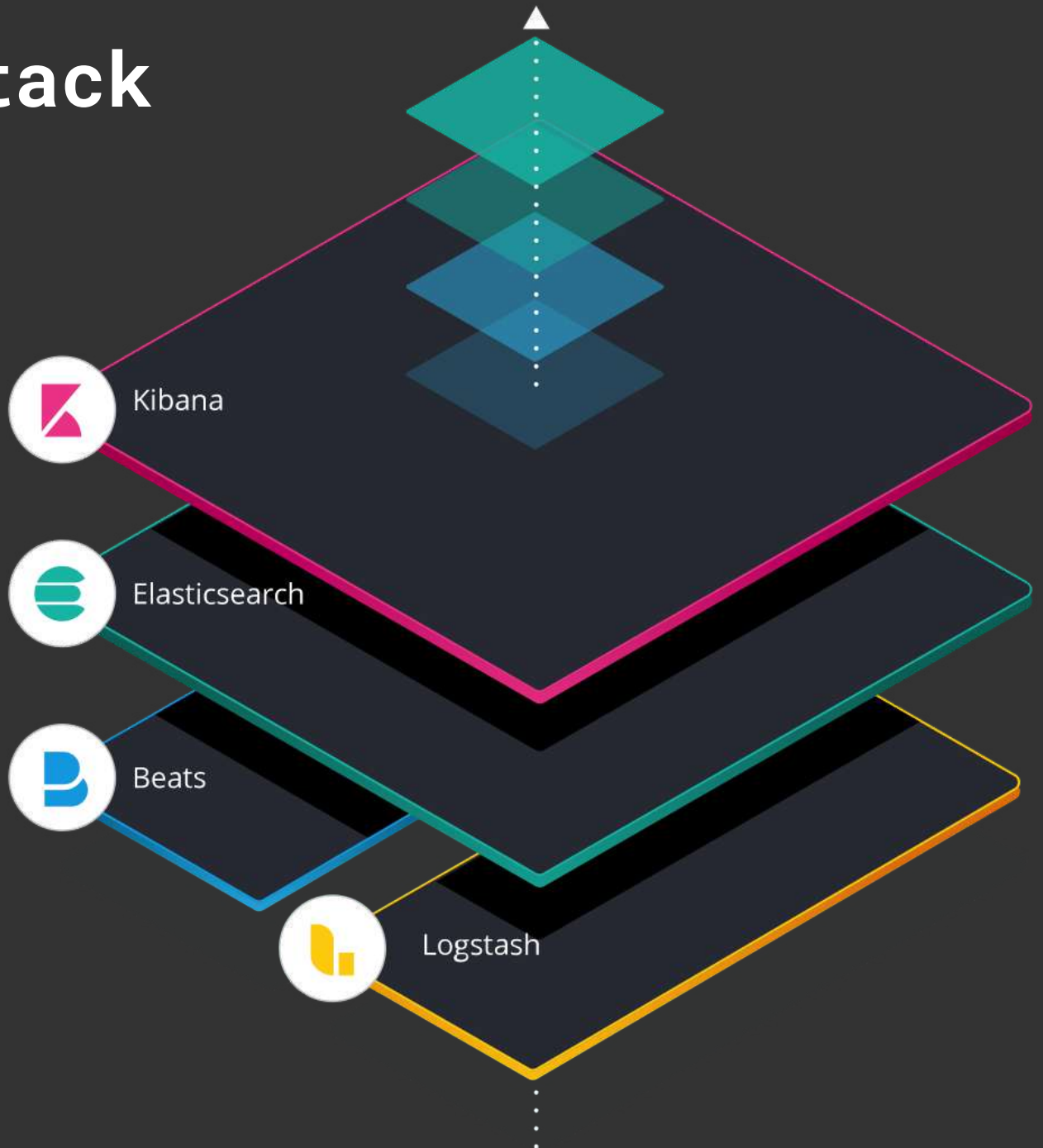
Elasticsearch - Digging deeper into full text search

Alexander Reelsen
Community Advocate
alex@elastic.co | @spinscale

TOC

- How to run the Elastic Stack
- Datatypes: `range_`, `date_nanos`, `search-as-you-type`, `flattened`
- Search: Field collapsing/ `top_hits`, `distance_feature` query, vector search, phonetic search
- Processors: `dissect`, `enrich`
- Index lifecycle management

Elastic Stack



elastic

Elasticsearch in 10 seconds

- Search Engine (FTS, Analytics, Geo), near real-time
- Distributed, scalable, highly available, resilient
- Interface: HTTP & JSON
- Heart of the Elastic Stack (Kibana, Logstash, Beats)

Installation & Start

```
# https://www.elastic.co/downloads/elasticsearch
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.5.2-darwin-x86_64.tar.gz
# wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.5.2-linux-x86_64.tar.gz
# wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.5.2-windows-x86_64.zip

tar zxf elasticsearch-7.5.2-darwin-x86_64.tar.gz
cd elasticsearch-7.5.2

./bin/elasticsearch-plugin install analysis-phonetic
./bin/elasticsearch
```

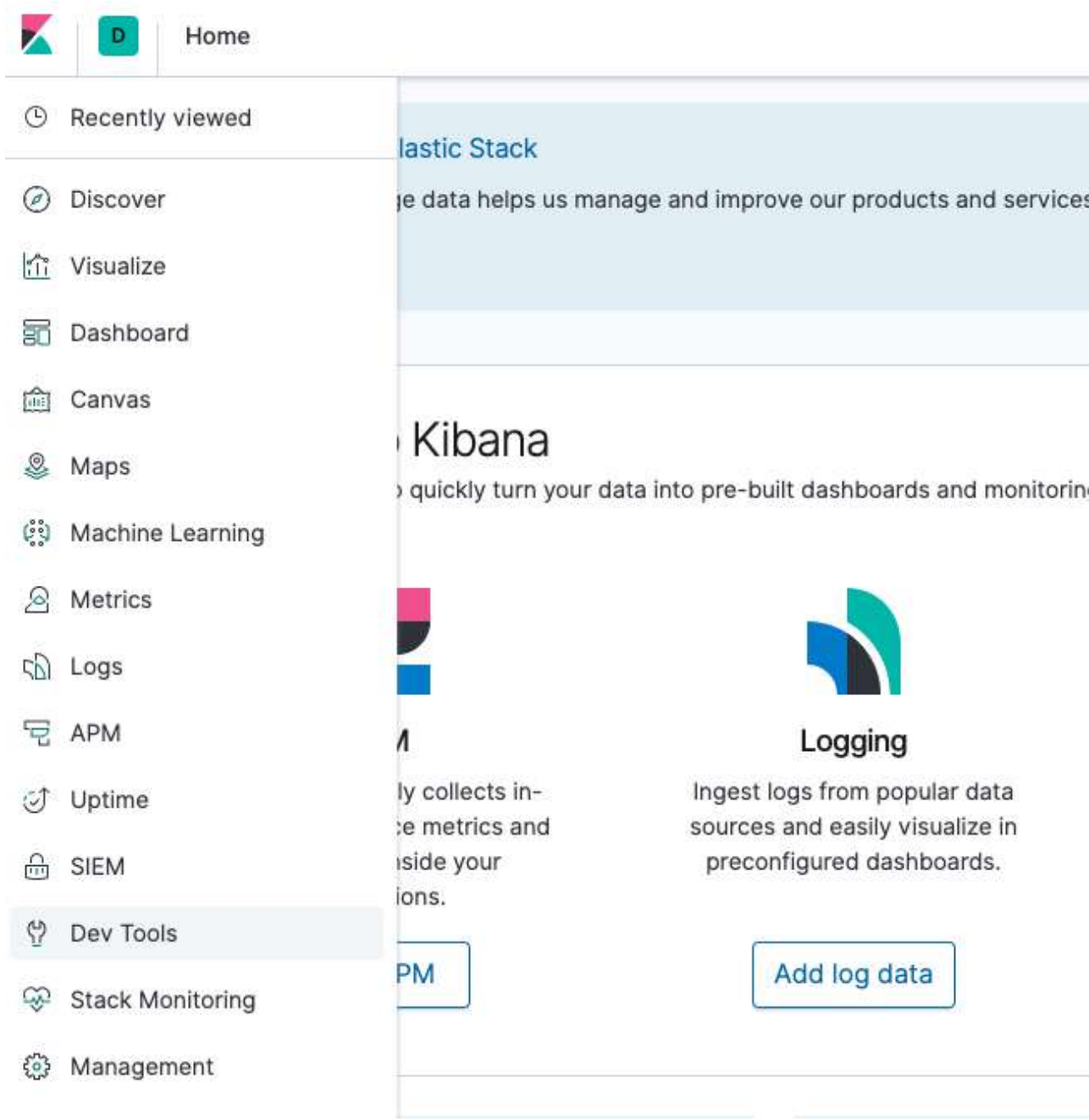
```
wget https://artifacts.elastic.co/downloads/kibana/kibana-7.5.2-darwin-x86_64.tar.gz
# wget https://artifacts.elastic.co/downloads/kibana/kibana-7.5.2-linux-x86_64.tar.gz
# wget https://artifacts.elastic.co/downloads/kibana/kibana-7.5.2-windows-x86_64.zip

tar zxf kibana-7.5.2-darwin-x86_64.tar.gz
cd kibana-7.5.2
./bin/kibana
```

Point your browser to http://localhost:5601/

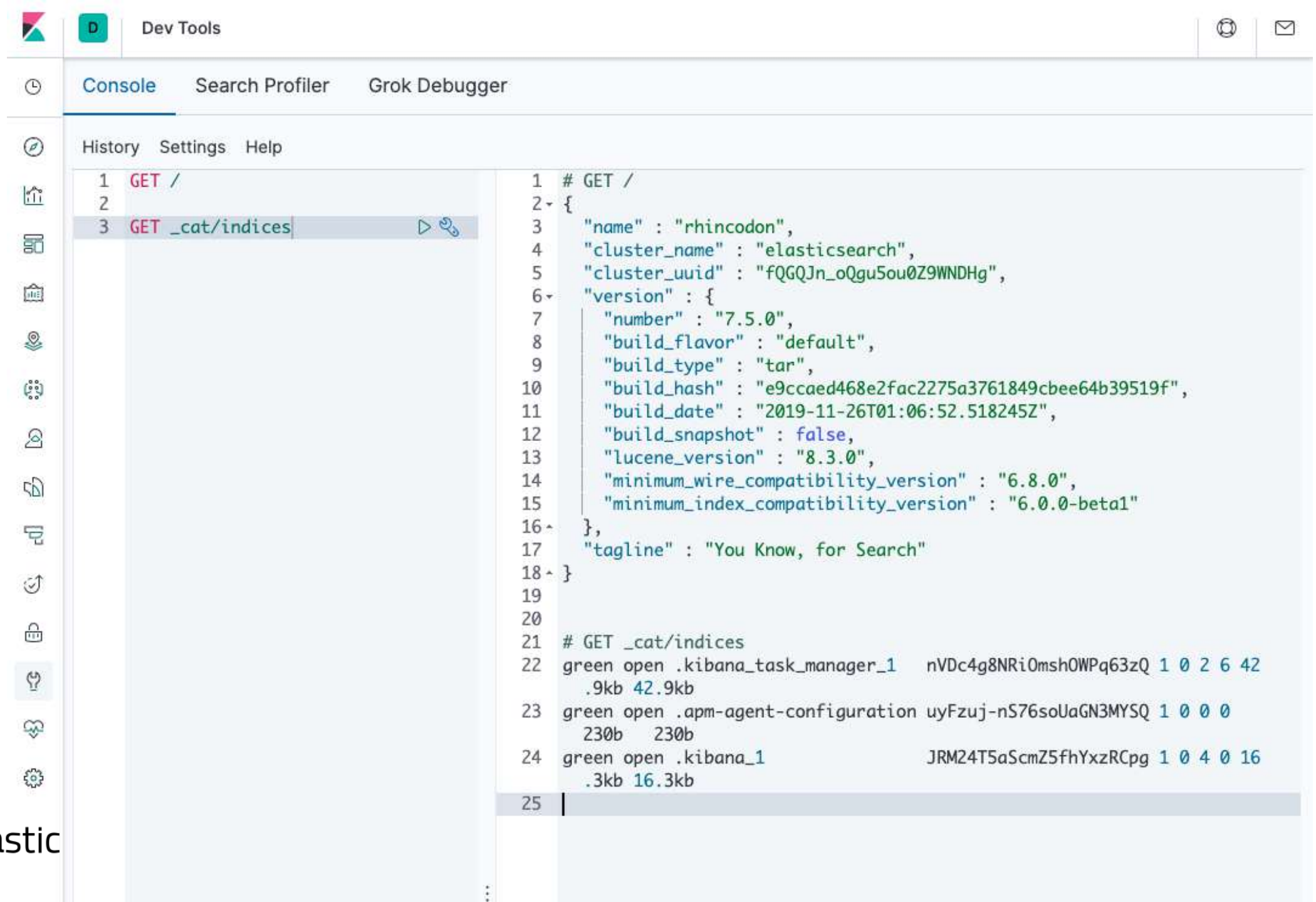
Click Dev-Tools

Samples in Kibana

Samples in Github



elastic


Dev Tools
Console
Search Profiler
Grok Debugger
History
Settings
Help
1 GET /
2
3 GET _cat/indices
1 # GET /
2 {
3 "name" : "rhincodon",
4 "cluster_name" : "elasticsearch",
5 "cluster_uuid" : "fQGQJn_oQgu5ou0Z9WNDHg",
6 "version" : {
7 "number" : "7.5.0",
8 "build_flavor" : "default",
9 "build_type" : "tar",
10 "build_hash" : "e9ccaed468e2fac2275a3761849cbee64b39519f",
11 "build_date" : "2019-11-26T01:06:52.518245Z",
12 "build_snapshot" : false,
13 "lucene_version" : "8.3.0",
14 "minimum_wire_compatibility_version" : "6.8.0",
15 "minimum_index_compatibility_version" : "6.0.0-beta1"
16 },
17 "tagline" : "You Know, for Search"
18 }
19
20
21 # GET _cat/indices
22 green open .kibana_task_manager_1 nVDc4g8NRiOmshOWPq63zQ 1 0 2 6 42
.9kb 42.9kb
23 green open .apm-agent-configuration uyFzuj-nS76soUaGN3MYSQ 1 0 0 0
230b 230b
24 green open .kibana_1 JRM24T5aScmZ5fhYxzRCpg 1 0 4 0 16
.3kb 16.3kb
25

elastic

Datatypes

elastic

`range_` datatype

- Search in ranges
- Supported types: `integer`, `float`, `long`, `double`, `date`, `ip`
- Example: Model hotel room availabilities

`date_range` datatype

```
# date_range datatype
PUT range_index
{
  "mappings": {
    "properties": {
      "time_frame": { "type": "date_range" }
    }
  }
}

PUT range_index/_doc/hotel_1_room_404
{
  "time_frame" : [
    { "gte" : "2015-10-31", "lte" : "2015-11-02" },
    { "gte" : "2015-11-04", "lte" : "2015-11-05" }
  ]
}
```

Search

```
GET range_index/_search
{
  "query": {
    "range": {
      "time_frame": {
        "gte": "2015-11-04",
        "lte": "2015-11-05",
        "relation": "contains"
      }
    }
  }
}
```

`date_nanos` datatype

```
# date_nanos datatype
PUT nanos_index
{
  "mappings": {
    "properties": {
      "time_in_nanos": {
        "type": "date_nanos",
        "format" : "yyyy-MM-dd'T'HH:mm:ss.nX"
      }
    }
  }
}

PUT nanos_index/_bulk?refresh
{ "index" : {}}
{"time_in_nanos":"2019-12-31T23:59:59.999999999Z"}
{ "index" : {}}
{"time_in_nanos":"2019-12-31T23:59:59.999Z"}
```

`date_nanos` datatype

```
GET nanos_index/_search
{
  "query": {
    "range": {
      "time_in_nanos": {
        "gt": "2019-12-31T23:59:59.999Z"
      }
    }
  }
}
```

`search-as-you-type` datatype

- One of the most requested features
- Very fast, but requires maintenance: `completion` suggester
- Since Elasticsearch 7.0: Lucene Block max WAND

`search-as-you-type` datatype

```
# search_as_you_type datatype
PUT search_index
{
  "mappings": {
    "properties": {
      "title": {
        "type": "search_as_you_type"
      }
    }
  }
}
```

`search-as-you-type` datatype

```
PUT search_index/_bulk?refresh
{ "index" : {} }
{ "title" : "This is it!" }
{ "index" : {} }
{ "title" : "This or that?" }
{ "index" : {} }
{ "title" : "Thin or thick?" }
{ "index" : {} }
{ "title" : "This is eval!" }
{ "index" : {} }
{ "title" : "Thick is not sick" }
```

`search-as-you-type` datatype

```
GET search_index/_search
{
  "query": {
    "match_phrase_prefix": { "title": "thi" }
  }
}

GET search_index/_search
{
  "query": {
    "match_phrase_prefix": { "title": "this i" }
  }
}

GET search_index/_search
{
  "query": {
    "match_phrase_prefix": { "title": "this is e" }
  }
}
```

`search-as-you-type` datatype

```
# no need for terms to be next to each other
GET search_index/_search
{
  "query": {
    "multi_match": {
      "query": "thick s",
      "type": "bool_prefix",
      "operator": "and",
      "fields": [
        "title",
        "title._2gram",
        "title._3gram"
      ]
    }
  }
}
```

`flattened` datatype

- Maps an entire object as a single field
- Prevents mapping explosion
- Allows only for some basic queries
- Searching: Think of a specialized `keyword` datatype

`flattened` datatype

```
# flattened datatype
PUT bug_reports
{
  "mappings": {
    "properties": {
      "labels": {
        "type": "flattened"
      }
    }
  }
}
```

`flattened` datatype

```
POST bug_reports/_doc/1?refresh
{
  "title": "Results are not sorted correctly.",
  "labels": {
    "priority": "urgent",
    "release": ["v1.2.5", "v1.3.0"],
    "timestamp": {
      "created": 1541458026,
      "closed": 1541457010
    }
  }
}
```

`flattened` datatype

```
POST bug_reports/_search
{
  "query": {
    "term": {"labels": "urgent"}
  }
}

POST bug_reports/_search
{
  "query": {
    "term": {"labels.release": "v1.3.0"}
  }
}
```

Searching

elastic

Bulk indexing

```
# index some book data to play around with
PUT books/_bulk
{ "index" : { "_id" : "database-internals" } }
{"isbn13":"978-1492040347","author":"Alexander Petrov", "title":"Database Internals: A deep-dive into how distributed data systems work","publisher":"O'Reilly","category":["databases","information systems"],"pages":350,"price":47.28,"format":"paperback","rating":4.5}
{ "index" : { "_id" : "designing-data-intensive-applications" } }
{"isbn13":"978-1449373320", "author":"Martin Kleppmann", "title":"Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems","publisher":"O'Reilly","category":["databases" ],"pages":590,"price":31.06,"format":"paperback","rating":4.4}
{ "index" : { "_id" : "kafka-the-definitive-guide" } }
{"isbn13":"978-1491936160","author":[ "Neha Narkhede", "Gwen Shapira", "Todd Palino"], "title":"Kafka: The Definitive Guide: Real-time data and stream processing at scale", "publisher":"O'Reilly","category":["databases" ],"pages":297,"price":37.31,"format":"paperback","rating":3.9}
{ "index" : { "_id" : "effective-java" } }
{"isbn13":"978-1491936160","author": "Joshua Block", "title":"Effective Java", "publisher":"Addison-Wesley", "category":["programming languages", "java" ],"pages":412,"price":27.91,"format":"paperback","rating":4.2}
{ "index" : { "_id" : "daemon" } }
{"isbn13":"978-1847249616","author":"Daniel Suarez", "title":"Daemon","publisher":"Quercus","category":["dystopia","novel"],"pages":448,"price":12.03,"format":"paperback","rating":4.0}
{ "index" : { "_id" : "cryptonomicon" } }
{"isbn13":"978-1847249616","author":"Neal Stephenson", "title":"Cryptonomicon","publisher":"Avon","category":["thriller", "novel" ],"pages":1152,"price":6.99,"format":"paperback","rating":4.0}
{ "index" : { "_id" : "garbage-collection-handbook" } }
{"isbn13":"978-1420082791","author": [ "Richard Jones", "Antony Hosking", "Eliot Moss" ], "title":"The Garbage Collection Handbook: The Art of Automatic Memory Management","publisher":"Taylor & Francis","category":["programming algorithms" ],"pages":511,"price":87.85,"format":"paperback","rating":5.0}
{ "index" : { "_id" : "radical-candor" } }
{"isbn13":"978-1250258403","author": "Kim Scott", "title":"Radical Candor: Be a Kick-Ass Boss Without Losing Your Humanity","publisher":"Macmillan","category":["human resources","management", "new work"],"pages":404,"price":7.29,"format":"paperback","rating":4.0}
{ "index" : { "_id" : "never-split-the-difference" } }
{"isbn13":"978-1847941497","author": "Chris Voss", "title":"Never Split the Difference: Negotiating as if Your Life Depended on It","publisher":"Random House Business","category":["negotiation", "sales"],"pages":288,"price":10.49,"format":"paperback","rating":4.3}
{ "index" : { "_id" : "not-giving-a-fsck" } }
{"isbn13":"978-0062641540","author": "Mark Manson", "title":"The Subtle Art of Not Giving a F*ck: A Counterintuitive Approach to Living a Good Life","publisher":"Harper","category":["success", "motivation"],"pages":224,"price":12.99,"format":"paperback","rating":4.4}
{ "index" : { "_id" : "permanent-record" } }
{"isbn13":"978-1250756541","author": "Edward Snowden", "title":"Permanent Record","publisher":"Macmillan","category":["politics", "biography"],"pages":339,"price":12.99,"format":"paperback","rating":4.7}
```

Field Collapsing

```
# field collapsing
GET books/_search
{
  "query": {
    "bool": {
      "must_not": [
        { "term": { "category.keyword": "novel" } }
      ]
    }
  },
  "collapse": {
    "field": "publisher.keyword"
  },
  "sort": [
    { "rating": { "order": "desc" } }
  ]
}
```

`top_hits` Aggregation

```
# top_hits aggregation
GET books/_search
{
  "size": 0,
  "aggs": {
    "by_format": {
      "terms": {
        "field": "format.keyword"
      },
      "aggs": {
        "by_rating": {
          "top_hits": {
            "size": 1,
            "sort": [ { "rating": "desc" } ]
          }
        }
      }
    }
  }
}
```

`distance_feature` datatype & query

```
# Add a new release_year field
PUT books/_mapping
{
  "properties" : {
    "release_year" : {
      "type" : "date",
      "format" : "strict_year"
    }
  }
}
```

`distance_feature` datatype & query

```
# update release_year of all books
PUT books/_bulk
{ "update" : { "_id" : "database-internals" } }
{ "doc" : { "release_year" : "2019" } }
{ "update" : { "_id" : "designing-data-intensive-applications" } }
{ "doc" : { "release_year" : "2017" } }
{ "update" : { "_id" : "kafka-the-definitive-guide" } }
{ "doc" : { "release_year" : "2017" } }
{ "update" : { "_id" : "effective-java" } }
{ "doc" : { "release_year" : "2017" } }
{ "update" : { "_id" : "daemon" } }
{ "doc" : { "release_year" : "2011" } }
{ "update" : { "_id" : "cryptonomicon" } }
{ "doc" : { "release_year" : "2002" } }
{ "update" : { "_id" : "garbage-collection-handbook" } }
{ "doc" : { "release_year" : "2011" } }
{ "update" : { "_id" : "radical-candor" } }
{ "doc" : { "release_year" : "2018" } }
{ "update" : { "_id" : "never-split-the-difference" } }
{ "doc" : { "release_year" : "2017" } }
{ "update" : { "_id" : "not-giving-a-fsck" } }
{ "doc" : { "release_year" : "2016" } }
{ "update" : { "_id" : "permanent-record" } }
{ "doc" : { "release_year" : "2019" } }
```

`distance_feature` datatype & query

```
# newer books are more relevant
# like function score, but waaaay faster
GET /books/_search
{
  "query": {
    "bool": {
      "filter": {
        "range": { "pages": { "gte": 500 } }
      },
      "should": {
        "distance_feature": {
          "field": "release_year",
          "pivot": "2555d",
          "origin": "now"
        }
      }
    }
  }
}
```

Vector based scoring

- Scoring based on features
- Two datatypes: `sparse` & `dense`
- vectors can be used for scoring using vector field functions
- query vector required

Feature modelling

- Prefers long books: Range 0-10
- Prefers good rated one: Range 0-5
- Prefers cheaper books: 0-10 (inverse, 0 more than 100 EUR, 10 less than 10 EUR)

Mapping update

```
# add vector field
PUT books/_mapping
{
  "properties": {
    "vector_recommendation": {
      "type": "dense_vector",
      "dims": 3
    }
  }
}
```

Add vectors to documents

```
# Add a vector for each document
PUT books/_bulk
{ "update" : { "_id" : "database-internals" } }
{ "doc" : { "vector_recommendation" : [3.5, 4.5, 5.2] } }
{ "update" : { "_id" : "designing-data-intensive-applications" } }
{ "doc" : { "vector_recommendation" : [5.9, 4.4, 6.8] } }
{ "update" : { "_id" : "kafka-the-definitive-guide" } }
{ "doc" : { "vector_recommendation" : [2.97, 3.9, 6.2] } }
{ "update" : { "_id" : "effective-java" } }
{ "doc" : { "vector_recommendation" : [4.12, 4.2, 7.2] } }
{ "update" : { "_id" : "daemon" } }
{ "doc" : { "vector_recommendation" : [4.48, 4.0, 8.7] } }
{ "update" : { "_id" : "cryptonomicon" } }
{ "doc" : { "vector_recommendation" : [10.0, 4.0, 9.3] } }
{ "update" : { "_id" : "garbage-collection-handbook" } }
{ "doc" : { "vector_recommendation" : [5.1, 5.0, 1.3] } }
{ "update" : { "_id" : "radical-candor" } }
{ "doc" : { "vector_recommendation" : [4.0, 4.0, 9.2] } }
{ "update" : { "_id" : "never-split-the-difference" } }
{ "doc" : { "vector_recommendation" : [4.0, 4.3, 8.9] } }
{ "update" : { "_id" : "not-giving-a-fsck" } }
{ "doc" : { "vector_recommendation" : [2.8, 4.4, 8.9] } }
{ "update" : { "_id" : "permanent-record" } }
{ "doc" : { "vector_recommendation" : [3.3, 4.7, 8.7] } }
```

Search for short, cheap books with a good rating

```
# short, good rating, cheap
GET books/_search
{
  "query": {
    "script_score": {
      "query" : {
        "match_all" : {}
      },
      "script": {
        "source": "cosineSimilarity(params.query_vector, doc['vector_recommendation']) + 1.0",
        "params": {
          "query_vector": [1.0, 5.0, 10.0]
        }
      }
    }
  }
}
```

Search for long, any priced books with a good rating

```
# long, good rating, any price
GET books/_search
{
  "query": {
    "script_score": {
      "query" : {
        "match_all" : {}
      },
      "script": {
        "source": "cosineSimilarity(params.query_vector, doc['vector_recommendation']) + 1.0",
        "params": {
          "query_vector": [10.0, 5.0, 5.0]
        }
      }
    }
  }
}
```

Phonetic search

- Find similar terms by converting terms to their phonetic representation

```
# phonetic mapping
PUT phonetic_sample
{
  "mappings": {
    "properties": {
      "name": {
        "type": "text",
        "fields": {
          "metaphone": { "type": "text", "analyzer": "metaphone_analyzer" },
          "koelner": { "type": "text", "analyzer": "koelner_analyzer" },
          "soundex": { "type": "text", "analyzer": "soundex_analyzer" }
        }
      }
    }
  },
  "settings": {
    "index": {
      "analysis": {
        "analyzer": {
          "metaphone_analyzer": { "tokenizer": "standard", "filter": [ "lowercase", "phonetic_filter" ] },
          "soundex_analyzer": { "tokenizer": "standard", "filter": [ "lowercase", "soundex_filter" ] },
          "koelner_analyzer": { "tokenizer": "standard", "filter": [ "lowercase", "koelner_filter" ] }
        },
        "filter": {
          "phonetic_filter": { "type": "phonetic", "encoder": "metaphone", "replace": false },
          "soundex_filter": { "type": "phonetic", "encoder": "soundex", "replace": false },
          "koelner_filter": { "type": "phonetic", "encoder": "koelnerphonetik", "replace": false
          }
        }
      }
    }
  }
}
```

Metaphone/Soundex

```
POST phonetic_sample/_analyze
{
  "field": "name.metaphone",
  "text": "Joe Blocks"
}

POST phonetic_sample/_analyze
{
  "field": "name.soundex",
  "text": "Joe Blocks"
}
```

Koelner phonetik

```
POST phonetic_sample/_analyze
{
  "field": "name.koelner",
  "text": "Aleksander"
}

POST phonetic_sample/_analyze
{
  "field": "name.koelner",
  "text": "Alexander"
}
```

Meier/Maier/Mayer/Meyer

```
PUT phonetic_sample/_bulk?refresh
{ "index" : { "_id" : 1 }}
{"name":"Peter Meyer"}
{ "index" : { "_id" : 2 }}
{"name":"Peter Meier"}
{ "index" : { "_id" : 3 }}
{"name":"Peter Maier"}
{ "index" : { "_id" : 4 }}
{"name":"Peter Mayer"}

GET phonetic_sample/_search
{ "query": { "match": { "name.metaphone": "Maier" } } }

GET phonetic_sample/_search
{ "query": { "match": { "name.koelner": "Maier" } } }
```

Ingest Processors

elastic

Dissect processor

- Grok processor is hard to configure for simple cases
- regular expressions are complex and CPU heavy
- `dissect` does not use regexes, syntax is simpler

Dissect processor

```
# dissect processor
POST _ingest/pipeline/_simulate
{
  "pipeline": {
    "processors": [
      {
        "dissect": {
          "field": "input",
          "pattern": "%{url}?%{param_string}"
        }
      },
      {
        "kv": {
          "field": "param_string",
          "target_field": "params",
          "field_split": "&",
          "value_split": "="
        }
      }
    ]
  },
  "docs": [
    {"_source" : { "input" : "https://example.org?foo=bar&spam=eggs" } }
  ]
}
```

Enrich processor

- Enrich documents with data from another index
- Processor uses an `enrich policy`
- Since: 7.5

Enrich processor: zip to city lookup

```
# enrich processor
PUT cities/_bulk?refresh
{ "index" : { "_id" : "munich"} }
{"zip":"80331","city":"Munich"}
{ "index" : { "_id" : "berlin"} }
{"zip":"10965","city":"Berlin"}

PUT /_enrich/policy/zip-policy
{
  "match": {
    "indices": "cities",
    "match_field": "zip",
    "enrich_fields": [ "city" ]
  }
}

POST /_enrich/policy/zip-policy/_execute

GET _cat/indices/.enrich-*
```

Enrich processor: zip to city lookup

```
POST /_ingest/pipeline/_simulate
{
  "pipeline": {
    "processors": [
      {
        "enrich": {
          "policy_name": "zip-policy",
          "field": "zip",
          "target_field": "city",
          "max_matches": "1"
        }
      }
    ]
  },
  "docs": [
    { "_id": "first", "_source" : { "zip" : "80331" } } ,
    { "_id": "second", "_source" : { "zip" : "50667" } }
  ]
}
```

Index Lifecycle Management

elastic

Index Lifecycle Management

- control aging indices
- configuration via a lifecycle policy
- policy split into phases per action
- `hot` action: set priority, unfollow, rollover
- `warm` action: set priority, unfollow, read-only, allocate, shrink, force merge
- `cold` action: set priority, allocate, freeze
- `delete` action: delete

Sample policy

```
# index lifecycle management
PUT _ilm/policy/full_policy
{
  "policy": {
    "phases": {
      "hot": {
        "actions": { "rollover": { "max_age": "7d", "max_size": "50G" } }
      },
      "warm": {
        "min_age": "30d",
        "actions": {
          "forcemerge": { "max_num_segments": 1 },
          "shrink": { "number_of_shards": 1 },
          "allocate": { "number_of_replicas": 2 }
        }
      },
      "cold": {
        "min_age": "60d",
        "actions": { "allocate": { "require": { "type": "cold" } } }
      },
      "delete": {
        "min_age": "90d",
        "actions": { "delete": {} }
      }
    }
  }
}
```

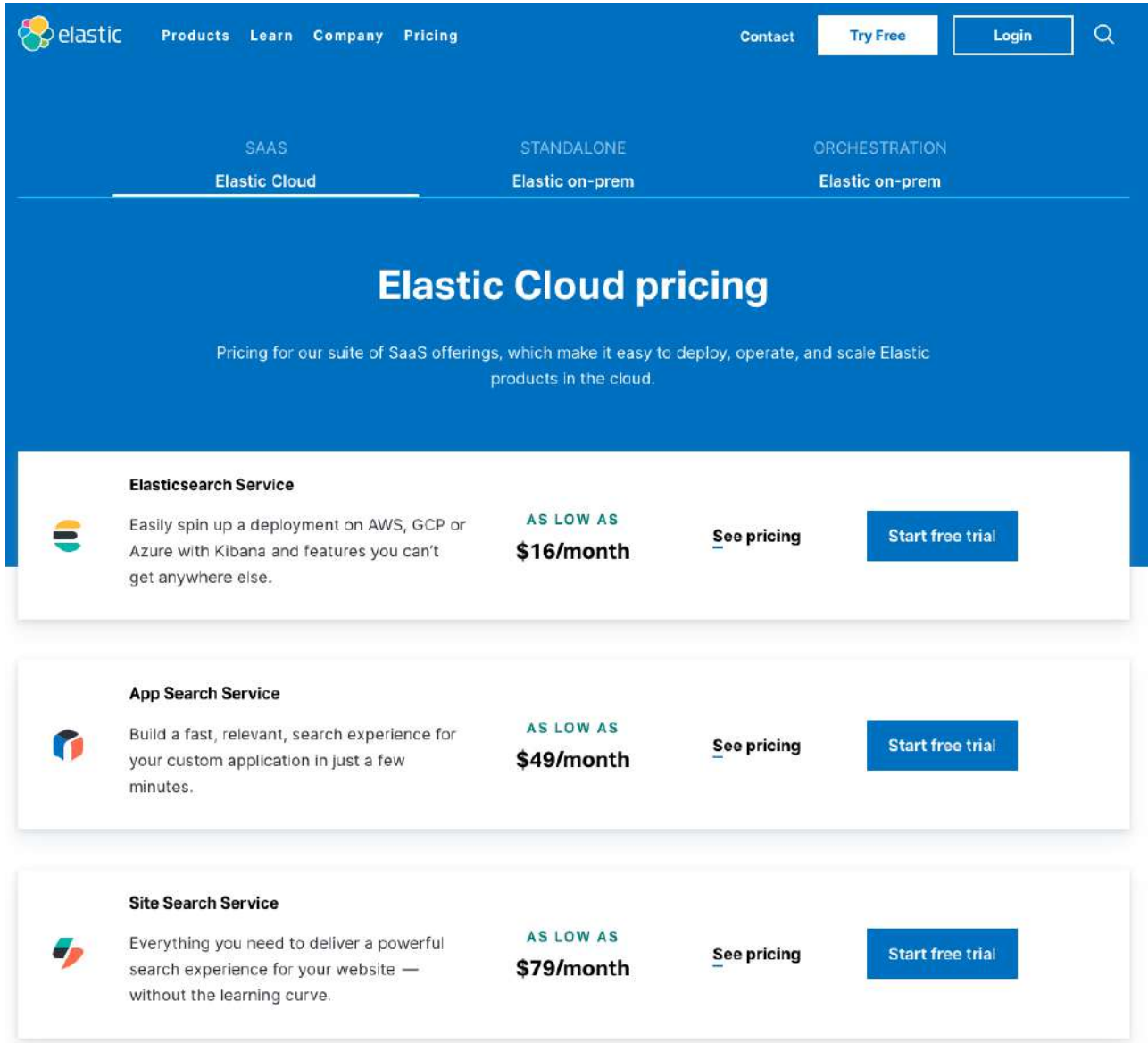
Summary

elastic

Summary

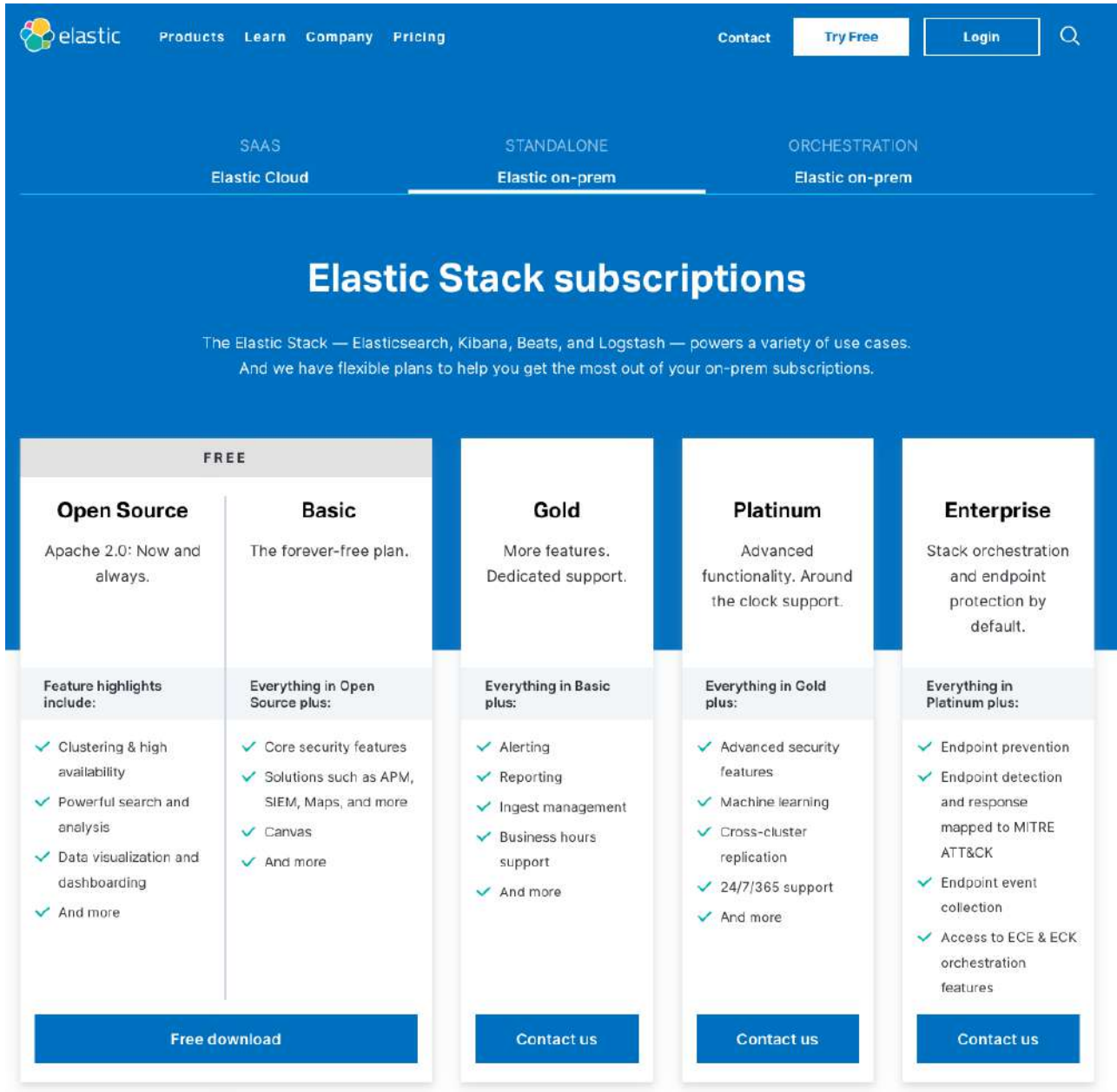
- Understanding search is hard
- Use the reference documentation
- Ask your users about expectations, do not guess!

Elastic Cloud



elastic

Elastic Support Subscriptions



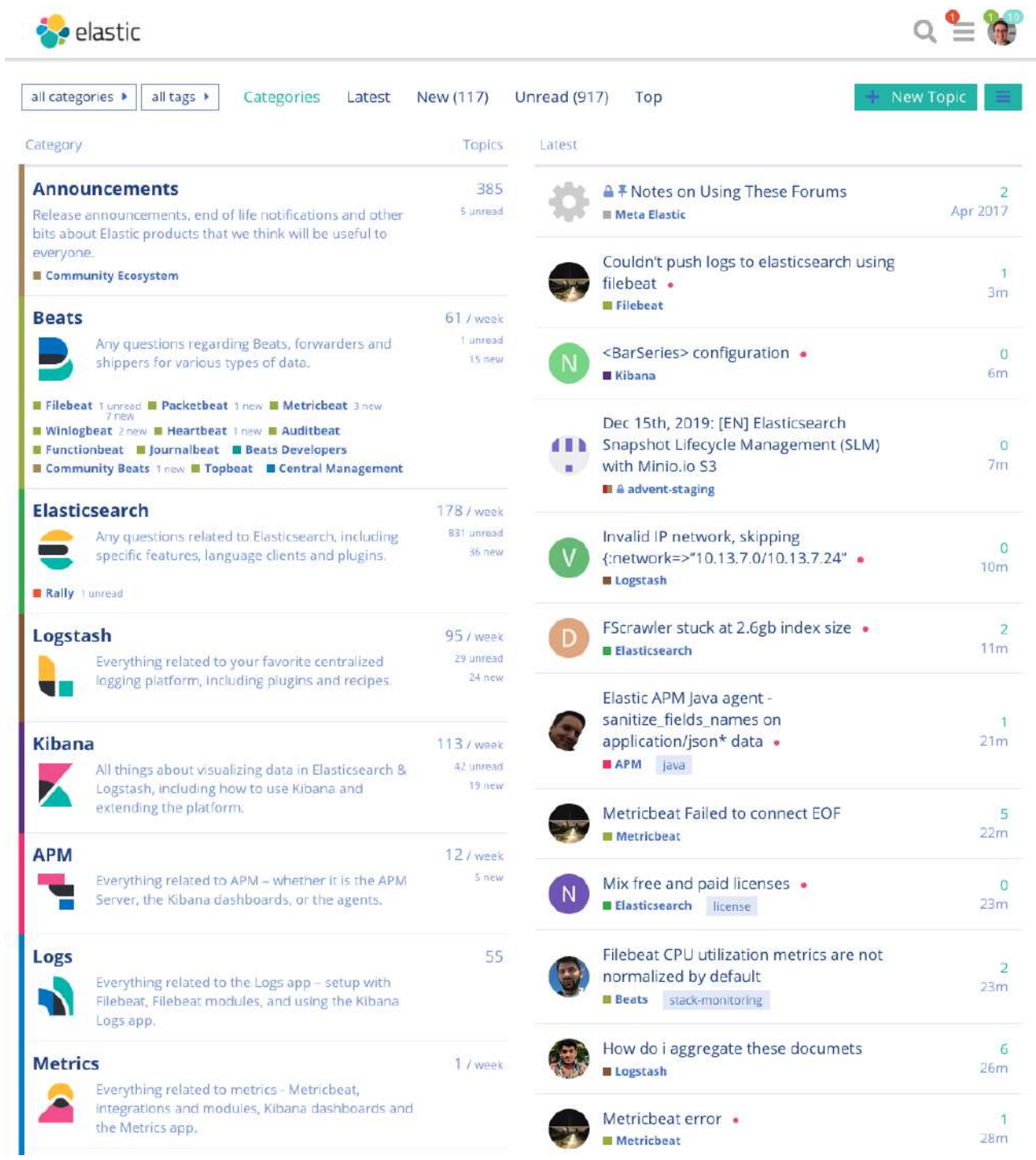
elastic

Getting more help

elastic

Discuss Forum

https://discuss.elastic.co



elastic

Community & Meetups

https://community.elastic.co

Explore by region

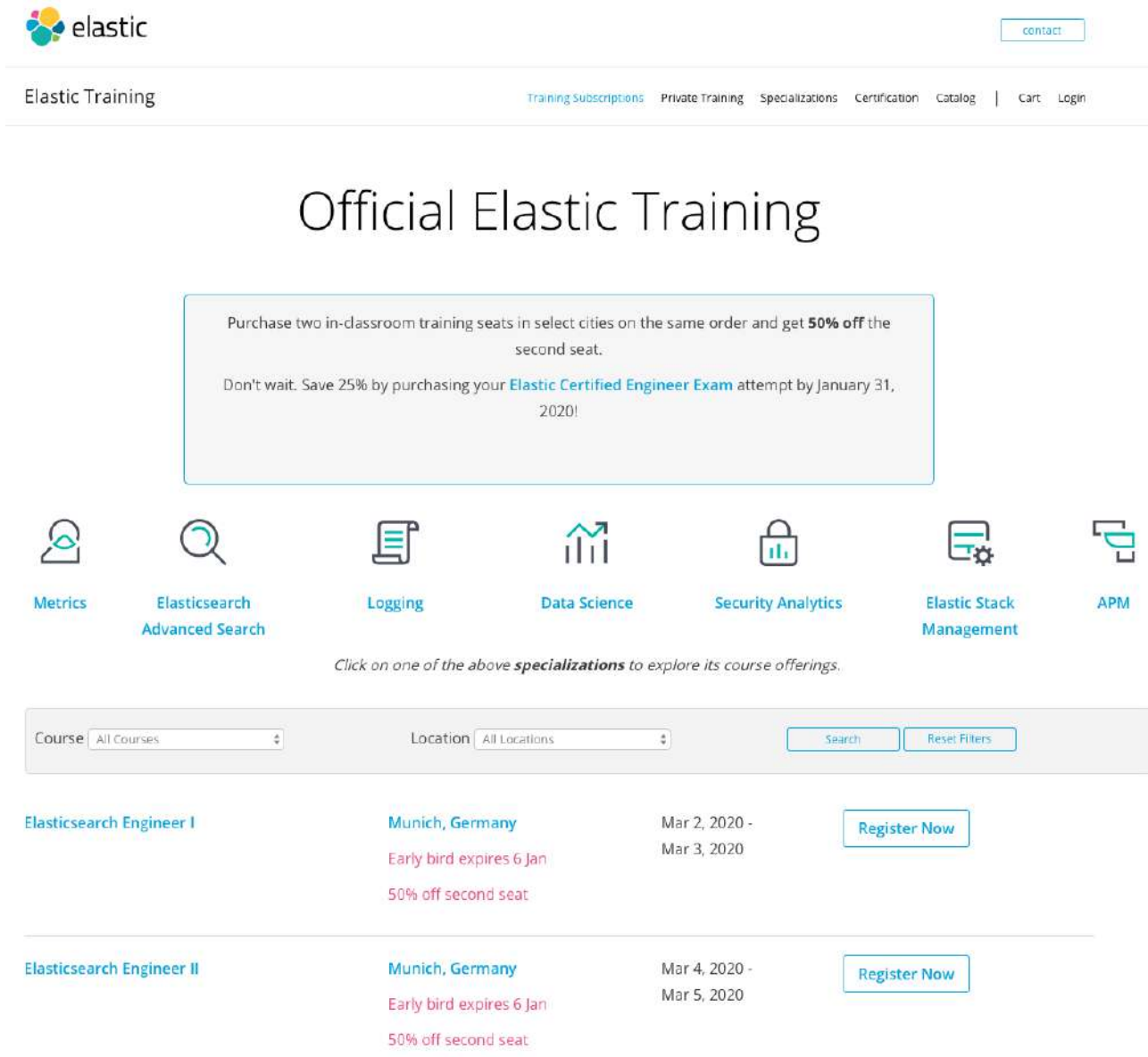
Asia Pacific and Japan | Europe, Middle East and Africa | North and South America | Virtual

ELASTIC - BARCELONA Spain	**ELASTIC - COPENHAGEN** Denmark	**ELASTIC - GOTEBORG** Sweden	**ELASTIC - SCOTLAND** United Kingdom
ELASTIC - STOCKHOLM Sweden	**ELASTIC - TEL AVIV** Israel	**ELASTIC - TURKEY** Turkey	**ELASTIC BONN USER GROUP** Germany
ELASTIC CAMBRIDGE & EAST ANGLIA USER GROUP United Kingdom	**ELASTIC DUBAI USER GROUP** United Arab Emirates	**ELASTIC FR** France	**ELASTIC GREECE** Greece
ELASTIC HELSINKI Finland	**ELASTIC KRAKOW USER GROUP** Poland	**ELASTIC LONDON USER GROUP** United Kingdom	**ELASTIC LUXEMBOURG USER GROUP** Luxembourg
ELASTIC MANCHESTER USER GROUP United Kingdom	**ELASTIC MOSCOW** Russian Federation	**ELASTIC NIGERIA** Nigeria	**ELASTIC OSLO USER GROUP** Norway
ELASTIC PORTUGAL Portugal	**ELASTIC RHEINRUHR** Germany	**ELASTIC SLOVAK USER GROUP** Slovakia	**ELASTIC USER GROUP - CZ** Czech Republic
ELASTIC USER GROUP - DUBLIN Ireland	**ELASTIC USER GROUP ABIDJAN** Côte d'Ivoire	**ELASTIC WARSAW USER GROUP** Poland	**ELASTIC ZAGREB** Croatia
ELASTICSEARCH - SOUTH AFRICA South Africa	**ELASTICSEARCH SWITZERLAND** Switzerland	**ELASTICSEARCH USER GROUP PAKISTAN** Pakistan	**SEARCH MEETUP MUNICH** Germany

elastic

Official Elastic Training

https://training.elastic.co



elastic

Thanks for listening

Q & A

Alexander Reelsen
Community Advocate
alex@elastic.co | @spinscale