

# Introduction into the Java HTTP REST client for Elasticsearch

Alexander Reelsen

Community Advocate

[alex@elastic.co](mailto:alex@elastic.co) | [@spinscale](https://twitter.com/spinscale)

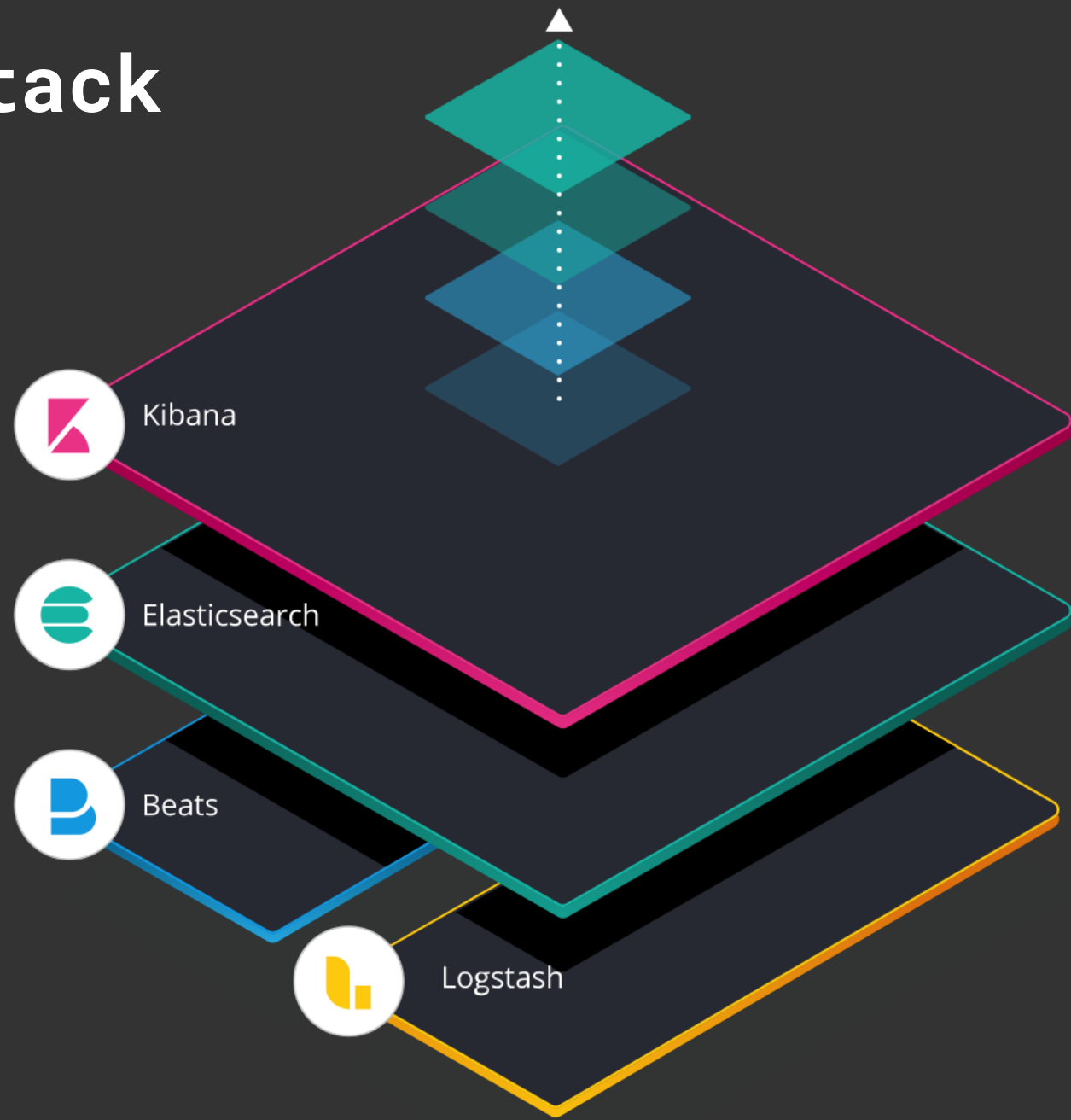


elastic

# Agenda

- History & Background
- Getting up and running
- Executing operations

# Elastic Stack



# 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)

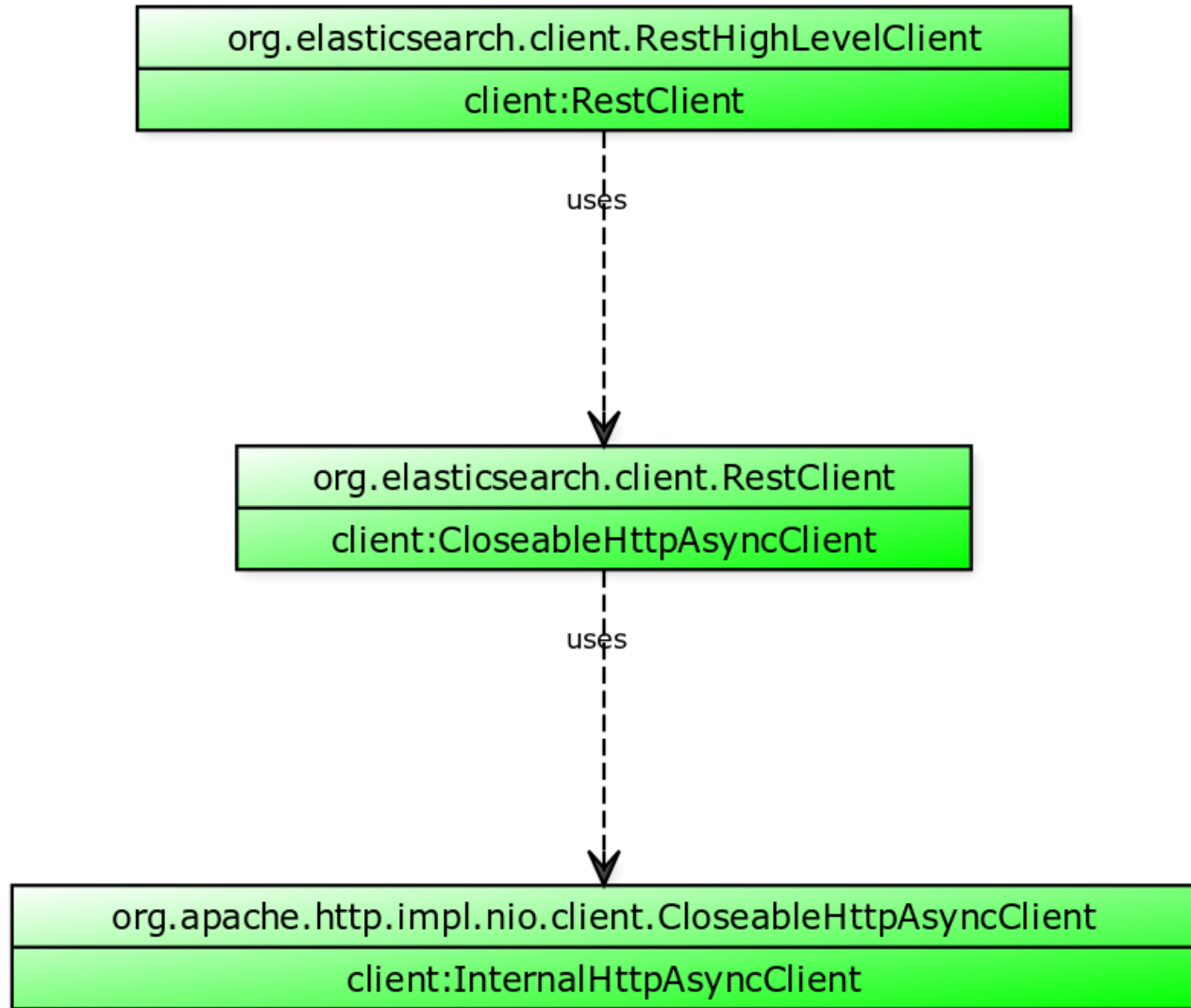
# History

- `TransportClient` has existed since earliest versions
- Uses port 9300 just like the node to node communication
- Uses custom serialization (BWC guarantees)
- All the query builders available, all POJOs usable

# Implementation

- Depends on the Elasticsearch core project
- Based on Apache HTTP Client (works on java 8), might want to consider shading
- Supports synchronous calls & cancellable async calls
- Threadsafe
- `RestLowLevelClient`
- `RestHighLevelClient`

# Architecture



# LowLevelRestClient

- TLS setup
- Basic Auth
- Sniffing
- Node Selectors



# HighLevelRestClient

- Requests classes for all endpoints
- Builders for queries/aggregations

# Compatibility

- Requires java 8
- forward compatible, 7.0 with 7.x, 7.1 with 7.1 - 7.x
- Also upgrade the REST client when upgrading ES cluster (bugfixes, REST API breaking changes over majors)
- Upgrade the client last

# Instantiating a client

```
RestHighLevelClient client = new RestHighLevelClient(
    RestClient.builder(new HttpHost("localhost", 9200)));

ClusterHealthResponse response =
    client.cluster().health(new ClusterHealthRequest(), RequestOptions.DEFAULT);

ActionListener<ClusterHealthResponse> listener =
    ActionListener.<ClusterHealthResponse>wrap(
        r -> System.out.println(r.getStatus()),
        Throwable::printStackTrace);

client.cluster()
    .healthAsync(new ClusterHealthRequest(), RequestOptions.DEFAULT, listener);
```

# Basic Auth

```
final CredentialsProvider credentialsProvider =
    new BasicCredentialsProvider();
credentialsProvider.setCredentials(AuthScope.ANY,
    new UsernamePasswordCredentials("user", "password"));

RestClientBuilder builder = RestClient.builder(
    new HttpHost("localhost", 9200))
    .setHttpClientConfigCallback(new HttpClientConfigCallback() {
        @Override
        public HttpAsyncClientBuilder customizeHttpClient(
            HttpAsyncClientBuilder httpClientBuilder) {
            return httpClientBuilder
                .setDefaultCredentialsProvider(credentialsProvider);
        }
    });
```

# TLS

```
KeyStore truststore = KeyStore.getInstance("jks");
try (InputStream is = Files.newInputStream(keyStorePath)) {
    truststore.load(is, keyStorePass.toCharArray());
}
SSLContextBuilder sslBuilder = SSLContexts.custom()
    .loadTrustMaterial(truststore, null);
final SSLContext sslContext = sslBuilder.build();
RestClientBuilder builder = RestClient.builder(
    new HttpHost("localhost", 9200, "https"))
    .setHttpClientConfigCallback(new HttpClientConfigCallback() {
        @Override
        public HttpAsyncClientBuilder customizeHttpClient(
            HttpAsyncClientBuilder httpClientBuilder) {
            return httpClientBuilder.setSSLContext(sslContext);
        }
    });
```

# Indexing

```
final byte[] bytes = ...;
final IndexRequest request = new IndexRequest(index);
// optional
request.id("my_id");
request.source(bytes, XContentType.JSON);

final IndexResponse response = client.index(request, RequestOptions.DEFAULT);
```

# Bulk indexing

```
BulkRequest request = new BulkRequest();

request.add(new IndexRequest("my_index").id("1")
    .source(XContentType.JSON, "name", "My first product"));
request.add(new IndexRequest("my_index").id("2")
    .source(XContentType.JSON, "name", "My second product"));
request.add(new IndexRequest("my_index").id("3")
    .source(XContentType.JSON, "name", "My third product"));

client.bulk(request, RequestOptions.DEFAULT);
```

# Bulk Processor

- Bulk requests should be limited by size/number of documents
- Requires manual checking for constant indexing
- Use the built-in `BulkProcessor` for that.



# Bulk Processor

```
BulkProcessor bulkProcessor = BulkProcessor.builder(
    client,
    new BulkProcessor.Listener() {
        @Override
        public void beforeBulk(long executionId,
                               BulkRequest request) { ... }

        @Override
        public void afterBulk(long executionId,
                              BulkRequest request,
                              BulkResponse response) { ... }

        @Override
        public void afterBulk(long executionId,
                              BulkRequest request,
                              Throwable failure) { ... }
    })
    .setBulkActions(10000)
    .setBulkSize(new ByteSizeValue(5, ByteSizeUnit.MB))
    .setFlushInterval(TimeValue.timeValueSeconds(5))
    .setConcurrentRequests(1)
    .setBackoffPolicy(
        BackoffPolicy.exponentialBackoff(TimeValue.timeValueMillis(100), 3))
    .build();
```

# Searching

```
final String input = "my nice query";
final SearchRequest searchRequest = new SearchRequest();
final SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();
searchSourceBuilder.query(QueryBuilders.multiMatchQuery(input, "name", "description"));
searchRequest.source(searchSourceBuilder);

final SearchResponse response = client.search(searchRequest, RequestOptions.DEFAULT);

// no hits
if (response.getHits().getTotalHits().value == 0) {
    return;
}

for (SearchHit hit : response.getHits().getHits()) {
    // per hit processing
}
```

# Use builders for Queries

- `QueryBuilders.*` for queries
- `AggregationBuilders.*` for aggs
- `PipelineAggregatorBuilders.*` for pipeline aggs

# Pagination using `from / offset`

```
private SearchRequest createSearchRequest(String input, int from, int size) {  
    final SearchRequest searchRequest = new SearchRequest();  
    final SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();  
    searchSourceBuilder.from(from);  
    searchSourceBuilder.size(size);  
    searchSourceBuilder.query(QueryBuilders.multiMatchQuery(input, "name", "description"));  
    searchRequest.source(searchSourceBuilder);  
  
    return searchRequest;  
}
```

# Pagination using `search_after`

```
SearchRequest searchRequest = new SearchRequest(INDEX);
searchRequest.source().query(QueryBuilders.matchQuery("name", "Name"));
searchRequest.source().sort(SortBuilders.fieldSort("price").order(SortOrder.DESC));
final SearchResponse response = client.search(searchRequest, RequestOptions.DEFAULT);

SearchRequest searchAfterRequest = new SearchRequest(INDEX);
searchAfterRequest.source().query(QueryBuilders.matchQuery("name", "Name"));
searchAfterRequest.source().sort(SortBuilders.fieldSort("price").order(SortOrder.DESC));

SearchHit lastHit = response.getHits().getHits()[response.getHits().getHits().length-1];
searchAfterRequest.source().searchAfter(lastHit.getSortValues());

final SearchResponse searchAfterResponse =
    client.search(searchAfterRequest, RequestOptions.DEFAULT);
```

# Node Selectors (Low Level REST client)

```
final NodeSelector INGEST_NODE_SELECTOR = nodes -> {
    final Iterator<Node> iterator = nodes.iterator();
    while (iterator.hasNext()) {
        Node node = iterator.next();
        // roles may be null if we don't know, thus we keep the node in then...
        if (node.getRoles() != null && node.getRoles().isIngest() == false) {
            iterator.remove();
        }
    }
};
```

```
HttpHost host = new HttpHost("localhost", 9200, "http");
final RestClientBuilder builder = RestClient.builder(host);
builder.setNodeSelector(INGEST_NODE_SELECTOR);
```

# Sniffers (Low Level REST client)

- Fetches list of current nodes
- Uses a background thread (requires proper shutdown)
- Configurable interval
- Optional: Sniffing on failure
- Own dependency

# Sniffers (Low Level REST client)

```
RestClient restClient = RestClient.builder(  
    new HttpHost("localhost", 9200, "http"))  
    .build();  
Sniffer sniffer = Sniffer.builder(restClient).build();  
  
// don't forget to close both  
restClient.close();  
sniffer.close();
```



# Remember, remember...

- Consider **shading** of the client
- Modular, requires dependencies for certain features
- Dependencies

# Deprecation warnings

- Error on deprecation warnings (useful when testing upgrades)

```
final RestClientBuilder builder = RestClient.builder(host);  
builder.setStrictDeprecationMode(true);
```

# Using against Elastic Cloud

```
String cloudId = "optionalHumanReadableName:" +  
    "dXMtZWFzdC0xLmF3cy5mb3VuZC5pbyRlbGFzdGljc2VhcmNoJGtpYmFuYQ==" ;  
final RestClientBuilder builder = RestClient.builder(cloudId);
```

# Debugging & Logging

- Apache HTTPClient uses JCL
- Set `org.elasticsearch.client=DEBUG` for more logging
- Set `org.apache.http.wire=TRACE` for whole HTTP request
- ^^ Privacy!

# Demo

- Uses `testcontainers`
- Demo code available on `github`

# Summary

# Summary

- Think async!
- Sniffing is your friend!
- Documentation
- With a bit of code change, you can either start a docker container or run against another cluster!

# Elastic Cloud






elastic Products Learn Company Pricing Contact [Try Free](#) [Login](#)

SAAS **Elastic Cloud** STANDALONE Elastic on-prem ORCHESTRATION Elastic on-prem

## Elastic Cloud pricing

Pricing for our suite of SaaS offerings, which make it easy to deploy, operate, and scale Elastic products in the cloud.

 <b>Elasticsearch Service</b> <p>Easily spin up a deployment on AWS, GCP or Azure with Kibana and features you can't get anywhere else.</p>	<b>AS LOW AS</b> <b>\$16/month</b>	<a href="#">See pricing</a>	<a href="#">Start free trial</a>
 <b>App Search Service</b> <p>Build a fast, relevant, search experience for your custom application in just a few minutes.</p>	<b>AS LOW AS</b> <b>\$49/month</b>	<a href="#">See pricing</a>	<a href="#">Start free trial</a>
 <b>Site Search Service</b> <p>Everything you need to deliver a powerful search experience for your website — without the learning curve.</p>	<b>AS LOW AS</b> <b>\$79/month</b>	<a href="#">See pricing</a>	<a href="#">Start free trial</a>



# Elastic Support Subscriptions



elastic [Products](#) [Learn](#) [Company](#) [Pricing](#) [Contact](#) [Try Free](#) [Login](#) [🔍](#)

SAAS **Elastic Cloud** **STANDALONE Elastic on-prem** ORCHESTRATION **Elastic on-prem**

## Elastic Stack subscriptions

The Elastic Stack — Elasticsearch, Kibana, Beats, and Logstash — powers a variety of use cases. And we have flexible plans to help you get the most out of your on-prem subscriptions.

FREE		Gold	Platinum	Enterprise
<b>Open Source</b> Apache 2.0: Now and always.	<b>Basic</b> The forever-free plan.	More features. Dedicated support.	Advanced functionality. Around the clock support.	Stack orchestration and endpoint protection by default.
<b>Feature highlights include:</b>	<b>Everything in Open Source plus:</b>	<b>Everything in Basic plus:</b>	<b>Everything in Gold plus:</b>	<b>Everything in Platinum plus:</b>
<ul style="list-style-type: none"><li>✓ Clustering &amp; high availability</li><li>✓ Powerful search and analysis</li><li>✓ Data visualization and dashboarding</li><li>✓ And more</li></ul>	<ul style="list-style-type: none"><li>✓ Core security features</li><li>✓ Solutions such as APM, SIEM, Maps, and more</li><li>✓ Canvas</li><li>✓ And more</li></ul>	<ul style="list-style-type: none"><li>✓ Alerting</li><li>✓ Reporting</li><li>✓ Ingest management</li><li>✓ Business hours support</li><li>✓ And more</li></ul>	<ul style="list-style-type: none"><li>✓ Advanced security features</li><li>✓ Machine learning</li><li>✓ Cross-cluster replication</li><li>✓ 24/7/365 support</li><li>✓ And more</li></ul>	<ul style="list-style-type: none"><li>✓ Endpoint prevention</li><li>✓ Endpoint detection and response mapped to MITRE ATT&amp;CK</li><li>✓ Endpoint event collection</li><li>✓ Access to ECE &amp; ECK orchestration features</li></ul>
<a href="#">Free download</a>		<a href="#">Contact us</a>	<a href="#">Contact us</a>	<a href="#">Contact us</a>

# Getting more help

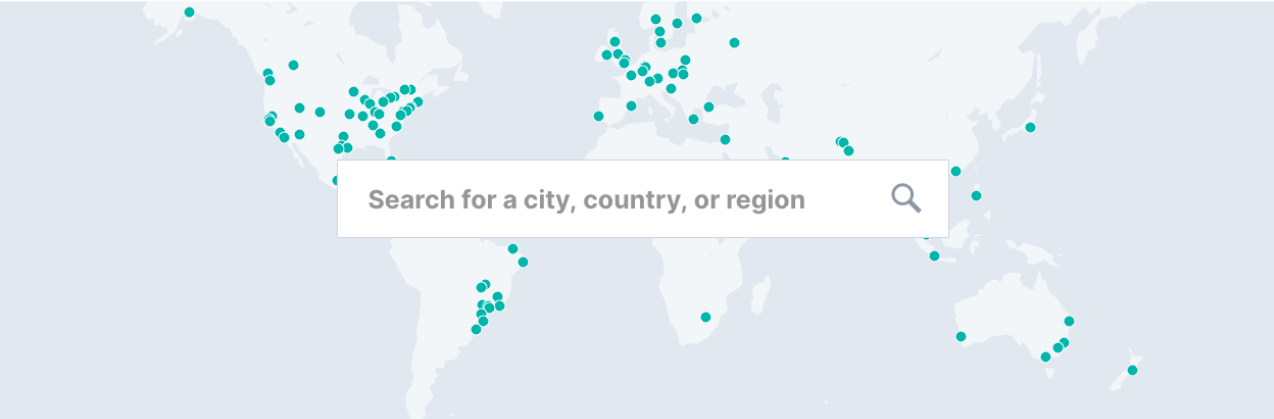
# Discuss Forum

<https://discuss.elastic.co>



all categories ▾ all tags ▾ Categories Latest New (117) Unread (917) Top [+ New Topic](#)

Category	Topics	Latest
<b>Announcements</b> Release announcements, end of life notifications and other bits about Elastic products that we think will be useful to everyone. <b>Community Ecosystem</b>	385 5 unread	<b>Notes on Using These Forums</b> <span>2</span> <b>Meta Elastic</b> Apr 17
<b>Beats</b> Any questions regarding Beats, forwarders and shippers for various types of data.	61 / week 1 unread 15 new	Couldn't push logs to elasticsearch using filebeat <span>1</span> <b>Filebeat</b> 3m
<b>Elasticsearch</b> Any questions related to Elasticsearch, including specific features, language clients and plugins. <b>Rally</b> 1 unread	178 / week 831 unread 36 new	<b>&lt;BarSeries&gt; configuration</b> <span>0</span> <b>Kibana</b> 6m
<b>Logstash</b> Everything related to your favorite centralized logging platform, including plugins and recipes.	95 / week 29 unread 24 new	Dec 15th, 2019: [EN] Elasticsearch Snapshot Lifecycle Management (SLM) with Minio.io S3 <span>0</span> <b>advent-staging</b> 7m
<b>Kibana</b> All things about visualizing data in Elasticsearch & Logstash, including how to use Kibana and extending the platform.	113 / week 42 unread 19 new	Invalid IP network, skipping {network=>"10.13.7.0/10.13.7.24"} <span>0</span> <b>Logstash</b> 10m
<b>APM</b> Everything related to APM - whether it is the APM Server, the Kibana dashboards, or the agents.	12 / week 5 new	FScrawler stuck at 2.6gb index size <span>2</span> <b>Elasticsearch</b> 11m
<b>Logs</b> Everything related to the Logs app - setup with Filebeat, Filebeat modules, and using the Kibana Logs app.	55	Elastic APM Java agent - sanitize_fields_names on application/json* data <span>1</span> <b>APM</b> java 21m
<b>Metrics</b> Everything related to metrics - Metricbeat, integrations and modules, Kibana dashboards and the Metrics app.	1 / week	Metricbeat Failed to connect EOF <span>5</span> <b>Metricbeat</b> 22m
		Mix free and paid licenses <span>0</span> <b>Elasticsearch</b> license 23m
		Filebeat CPU utilization metrics are not normalized by default <span>2</span> <b>Beats</b> stack-monitoring 23m
		How do i aggregate these documets <span>6</span> <b>Logstash</b> 26m
		Metricbeat error <span>1</span> <b>Metricbeat</b> 28m



# Community & Meetups

<https://community.elastic.co>



### Explore by region

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

<b>ELASTIC - BARCELONA</b> Spain 🇪🇸	<b>ELASTIC - COPENHAGEN</b> Denmark 🇩🇰	<b>ELASTIC - GOTEBORG</b> Sweden 🇸🇪	<b>ELASTIC - SCOTLAND</b> United Kingdom 🇬🇧
<b>ELASTIC - STOCKHOLM</b> Sweden 🇸🇪	<b>ELASTIC - TEL AVIV</b> Israel 🇮🇱	<b>ELASTIC - TURKEY</b> Turkey 🇹🇷	<b>ELASTIC BONN USER GROUP</b> Germany 🇩🇪
<b>ELASTIC CAMBRIDGE &amp; EAST ANGLIA USER GROUP</b> United Kingdom 🇬🇧	<b>ELASTIC DUBAI USER GROUP</b> United Arab Emirates 🇦🇪	<b>ELASTIC FR</b> France 🇫🇷	<b>ELASTIC GREECE</b> Greece 🇬🇷
<b>ELASTIC HELSINKI</b> Finland 🇫🇮	<b>ELASTIC KRAKOW USER GROUP</b> Poland 🇵🇱	<b>ELASTIC LONDON USER GROUP</b> United Kingdom 🇬🇧	<b>ELASTIC LUXEMBOURG USER GROUP</b> Luxembourg 🇱🇺
<b>ELASTIC MANCHESTER USER GROUP</b> United Kingdom 🇬🇧	<b>ELASTIC MOSCOW</b> Russian Federation 🇷🇺	<b>ELASTIC NIGERIA</b> Nigeria 🇳🇮	<b>ELASTIC OSLO USER GROUP</b> Norway 🇳🇴
<b>ELASTIC PORTUGAL</b> Portugal 🇵🇹	<b>ELASTIC RHEINRUHR</b> Germany 🇩🇪	<b>ELASTIC SLOVAK USER GROUP</b> Slovakia 🇸🇰	<b>ELASTIC USER GROUP - CZ</b> Czech Republic 🇨🇪
<b>ELASTIC USER GROUP - DUBLIN</b> Ireland 🇮🇪	<b>ELASTIC USER GROUP ABIDJAN</b> Côte d'Ivoire 🇨🇮	<b>ELASTIC WARSAW USER GROUP</b> Poland 🇵🇱	<b>ELASTIC ZAGREB</b> Croatia 🇭🇷
<b>ELASTICSEARCH - SOUTH AFRICA</b> South Africa 🇿🇦	<b>ELASTICSEARCH SWITZERLAND</b> Switzerland 🇨🇭	<b>ELASTICSEARCH USER GROUP PAKISTAN</b> Pakistan 🇵🇰	<b>SEARCH MEETUP MUNICH</b> Germany 🇩🇪

# Official Elastic Training

<https://training.elastic.co>



[contact](#)

Elastic Training

[Training Subscriptions](#) [Private Training](#) [Specializations](#) [Certification](#) [Catalog](#) | [Cart](#) [Login](#)

## Official Elastic Training

Purchase two in-classroom training seats in select cities on the same order and get **50% off** the second seat.

Don't wait. Save 25% by purchasing your [Elastic Certified Engineer Exam](#) attempt by January 31, 2020!



Metrics



Elasticsearch  
Advanced Search



Logging



Data Science



Security Analytics



Elastic Stack  
Management



APM

Click on one of the above **specializations** to explore its course offerings.

Course

Location

[Search](#)

[Reset Filters](#)

Elasticsearch Engineer I

Munich, Germany

Mar 2, 2020 -  
Mar 3, 2020

[Register Now](#)

Early bird expires 6 Jan

50% off second seat

Elasticsearch Engineer II

Munich, Germany

Mar 4, 2020 -  
Mar 5, 2020

[Register Now](#)

Early bird expires 6 Jan

50% off second seat

**Thanks for listening**

**Q & A**

Alexander Reelsen

Community Advocate

[alex@elastic.co](mailto:alex@elastic.co) | [@spinscale](https://twitter.com/spinscale)



**elastic**