Elasticsearch, Logstash, and Kibana

Duration: 5 Days

Pre-requisites

No prior knowledge of the Elastic Stack is required. This course is designed for individuals new to the Elastic Stack Comfort using the terminal or command line is recommended.

Requirements

- Participants must use their own desktop or laptop system.
- Internet connection capable of streaming video.
- Windows 7 or later.
- Java version 1.8u20 or later installed.
- A modern web browser.
- At least 20% free disk space.

Basic flow of data in Elasticsearch

- What is Elasticsearch and typical use-cases.
- Shards and replicas; packaging
- Installation; configuration files
- Indexing; what is an index, type, and ID.
- Mappings; stored and indexed fields; _source and _all
- Analysis basics
- Realtime get.
- Search: how searches are distributed to shards
- Ranking by TF/IDF and BM25
- Aggregations and doc values introduction
- Updates; versioning
- Deletes, introduction to Lucene segment merges.
- Lab
    - CRUD operations
    - query and filter
    - pagination

Controlling how data is indexed and stored.

- Mappings and mapping types
- Multi-field definitions
- Default mappings; dynamic mappings
- Texts, keywords, integers, and other core types
- Date formats
- Pre-defined fields, when to store fields separately vs using _source.

- Analysers: using the Analyze API
- Char filters
- Tokenizers: standard vs whitespace.
- Token filters: lowercase, stop words, synonyms, ngrams and shingles.
- Lab:
    - Exact match vs full-text search
    - Using the ASCII folding token filter for better internationalization
    - Using language analyzers to support stemming.

Searching through your data

- Selecting fields, source filtering and field data fields
- Sorting and pagination
- Search basics: term, range, and bool queries
- Enable caching through the filter context.
- Match query: configuring the analyzer, operator, common terms, and fuzziness.
- Query string and simple query string queries.
- Lab:
    - Using various ways of selecting fields.
    - Configure sorting and pagination.
    - Using a bool query to combine different match, range, and term queries.
    - Boosting exact matches above stemmed ones.

Aggregations

- Relationships between queries and aggregations; post filter, global aggregations.
- General optimizations: avoid script fields, set result size to 0 to cache.
- Metrics aggregations: stats, cardinality, percentiles.
- Why terms, cardinality and percentiles are approximate.
- Multi-bucket aggregations: terms, ranges, and histograms.
- Single-bucket aggregations and nesting; how nesting works.
- Lab:
    - Configure sizes of results, per-shard, and overall buckets.
    - Computing the cardinality of a field.
    - Sorting buckets by results of sub-aggregations.
    - Optimizing terms queries by configuring collect mode.
    - Nest the sum and histogram aggregations.

Data Visualization through Kibana

- Installation and configuration.
- Index patterns; refreshing the fields list.
- Discovering and searching raw data.

- Lucene query syntax.
- Visualizing data: types of visualizations and their use.
- Timelion charts: using the Timelion query language.
- Building dashboards.
- Lab:
    - Building complex queries through the Lucene query syntax.
    - Digging deeper into data through sub-aggregations.
    - Building dashboards on top of saved searches and visualizations.
    - Comparing different data series in Timelion (raw average vs moving average).

Data ingestion through Logstash

- Installation
- Inputs: popular input plugins and their configuration options.
- Codecs: parsing JSON and multiline logs.
- Filters: using grok and Geo IP to parse and enrich data.
- Outputs: popular output plugins and their options.
- Pipeline pattern: using Logstash on every logging box.
- Using Logstash with Kafka and Redis as a buffer.
- Adjusting pipeline workers and batch sizes.
- Adjusting Logstash heap size.
- Specific Plugin Tunables.
- Lab:
    - Configuring Logstash to parse and enrich Apache logs.
    - Tuning Logstash for throughput.
    - Using Logstash with Kafka.

Data collection using Beats.

- Installation: Packetbeat, TopBeat, Filebeat
- Filebeat Tunables
- Parsing JSON logs
- Sending logs directly to Elasticsearch
- Using Ingest nodes.
- Sending logs directly to Logstash
- Sending logs to Logstash via Kafka
- Lab:
    - Setting up TopBeat to push metrics to Elasticsearch.
    - Shipping parsing Apache logs via Filebeat and Ingest node.
    - Shipping and parsing Apache logs via Filebeat and Logstash

Data collection using rsyslog.

- Installation
- Plugins: main input modules and their configurations
- Message modifiers: using mm normalize to parse unstructured data in a scalable way.
- Parsing JSON logs.
- Using grok in rsyslog.
- Tuning queues, workers, and batch sizes
- Rainer script: variables, conditionals, loops, and lookup tables
- Using rulesets to manage multiple data flows.
- Writing data to Elasticsearch
- Coupling rsyslog with Logstash via Redis/Kafka
- Lab
  - Sending local syslog to Elasticsearch
  - Tailing files with rsyslog and sending them to Kafka
  - Using rulesets to separate local and remote logs.
  - Parsing logs with mm normalize and sending them to Elasticsearch.

Data Collection using Log agent-js.

- Installation
- Running on-demand or as a service
- Parsing rules
- Geo IP matching and database updates.
- UDP syslog and other listeners
- Lab
  - Parsing and sending local Apache and syslog to Elasticsearch
  - Build a pipeline from rsyslog to Elasticsearch through Log agent.

Relevancy tuning

- Analysis: stop words, synonyms, ngrams and shingles and their alternatives
- Using the Reindex API when mappings need to be changed.
- A deep look into BM25
- Multi-match query: choosing between best fields, most fields, and cross fields modes.
- Tweaking the score with the function score query
- Lab
  - Using the letter tokenizer as an option for URL matching
  - Using ngrams to tolerate typos.
  - Using shingles to match compound words.
  - Implement hashtag search via the word delimiter token filter.
  - Searching across multiple fields.
  - Boosting documents based on date and number of views.
  - Typo tolerance without using ngrams.

  - reducing the impact of common words without using stop words.

Advanced aggregations

- Finding trends and outliers with the significant terms aggregation
- Cheaper and more representative results with the sampler aggregation
- Field collapsing with the top hits aggregations.
- Pipeline aggregations; moving averages.
- Lab
  - Checking trends, the significant terms aggregation
  - Show the latest hit per category.
  - Using the moving average aggregation

Working with relational data

- Arrays and objects; why the offer the best performance and when they fail.
- Nested documents
- Nested queries; using inner hits.
- Parent-child relations
- Denormalizing and application-side joins
- Deciding on which feature/technique to use
- Lab
  - Model a one-to-one relationship.
  - Model a query-heavy one-to-many relationship.
  - Model an update-heavy one-to-many relationship.
  - Model a many-to-many relationship.

Percolator

- Percolator basics
- Configuring mappings for percolation
- Using routing, filters, sorting and aggregations with the Percolator Query
- Lab
  - Using Percolator to trigger alerts.
  - Using metadata to filter and aggregate matching queries.

Suggesters

- Overview of types and requests
- Term vs. phrase suggester
- How the phrase suggester collects candidates
- Using a shingle field to score candidate phrases.
- Completion vs context suggesters
- Completion suggesters vs prefix queries

- Mapping for completion suggesters
- Weights and fuzzy matches
- Payloads for instant-search kind of autocomplete.
- Lab
  - Using the term suggester to suggest single word corrections.
  - Using the phrase suggester against a shingle field for multi-word suggestions
  - Using a separate index for autocomplete
  - Using the _suggest endpoint instead of _search.
  - Boosting suggestions via static weights
  - Add fuzzy support for suggestions.
  - Filtering suggestions
  - Using metadata for ranking suggestions (terms, location)

Geo-spatial Research

- Basics: geo-point and geo-shape types
- How shape matching is done via geohashes
- Distance, distance range and bounding box queries
- Lab
  - Indexing geo-points and searching them via bounding box and polygon queries
  - Filtering and aggregating geo-points by distance.
  - Matching a shape against a point

Highlighting

- How the default highlighter works
- Common highlighter options: size, order, and number of fragments
- Postings highlighter: overhead, use-cases, mapping
- Fast vector highlighter: using term vectors for extra flexibility.
- Lab
  - Selecting fields to highlight and disabling _source from the response.
  - Choosing highlight tags, number of fragments, their size and order.
  - Using the postings highlighter for long natural language fields.
  - Using the fast vector highlighter for multi-fields.

High Availability and Kibana Security

- Managing a cluster, including how to configure shard filtering, shard allocation awareness and forced awareness.
- Hands-on Lab
  - Learn about designing for scale, scaling with replicas, scaling with Indices, capacity planning use cases, and working with time-based data.
  - Hands-on Lab

- We discuss of monitoring options, including the Stats API, task monitoring, the cat API, the X-Pack Monitoring component, and guidelines for monitoring a cluster and setting up alerts.
- Hands-on Lab