

Date \_\_\_\_\_ 20 \_\_\_\_\_

## Elastic Search Basics

Documents - JSON data, every document has unique id

Indices - an index powers search into all document within a collection of types.

They contain inverted indices that let you search across everything within them & mappings that define schemas for the data within.

	<u>inverted index</u>
document 1	word 1 : 1, 2
	" 2 : 1
document 2	" 3 : 1, 2

## TF-IDF

term frequency \* Inverse document freq

Term frequency how often term appears in a given doc

Document frequency how often term appears in all documents

TF/IDF relevance of term in a document

Date \_\_\_\_\_ 20 \_\_\_\_\_

Using Indices

Restful API

Client API

Analytic Tools - Kibana

How Elastic search scales

- Index is split into shards
  - documents are hashed to a particular shard
  - Each shard may be on a different node in a cluster
  - Every shard has self contained Lucene index of its own

Primary & Replica Shards

app round robin req. amongst nodes

node → install of elasticsearch usually 1 per physical server

write req. routed to primary shard then replicated  
read req. routed to any shard

Date \_\_\_\_\_ 20 \_\_\_\_\_

Elastic search manages redundancy makes it fault tolerant

odd # nodes a good idea

Number of primary shards cannot be changed later

can add more replicas for read throughput

worst case - need to reindex data

# of shards setup front via PUT

number of replicas defined as per primary

Ques

- schema for documents are defined by the index
- What purpose do inverted index serve?  
quickly map search terms to documents.