

# CSCE 5300 Introduction to Big Data and Data Science

## Lesson 6 Apache Lucene Apache Solr

# Overview

- Apache Lucene
- Apache Solr

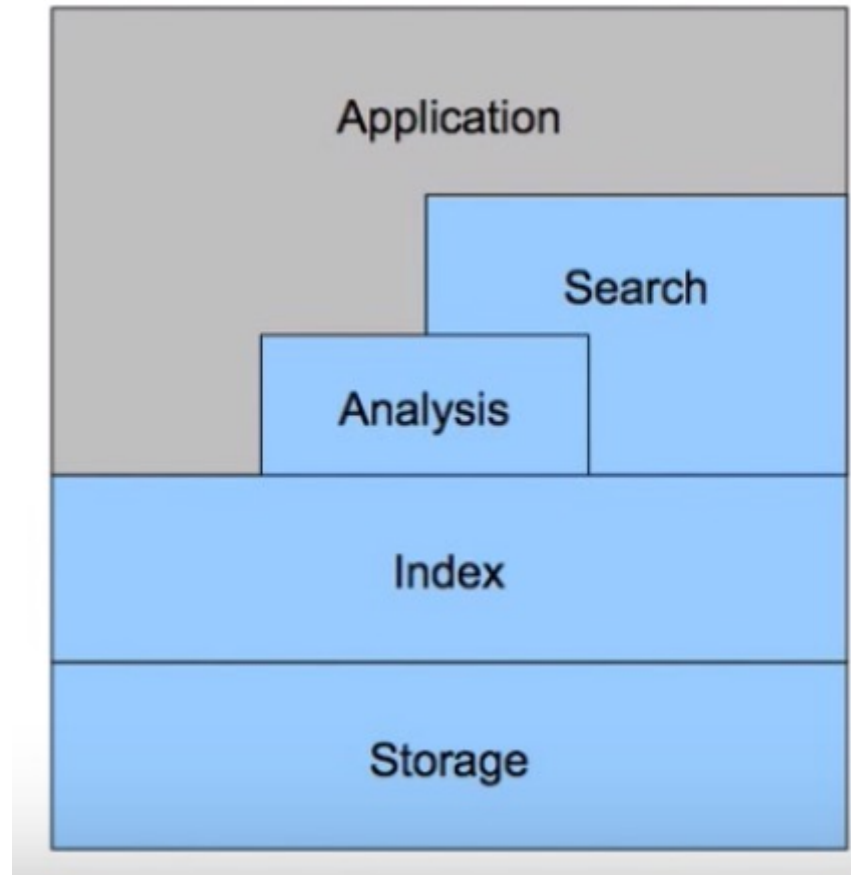
# Apache Lucene



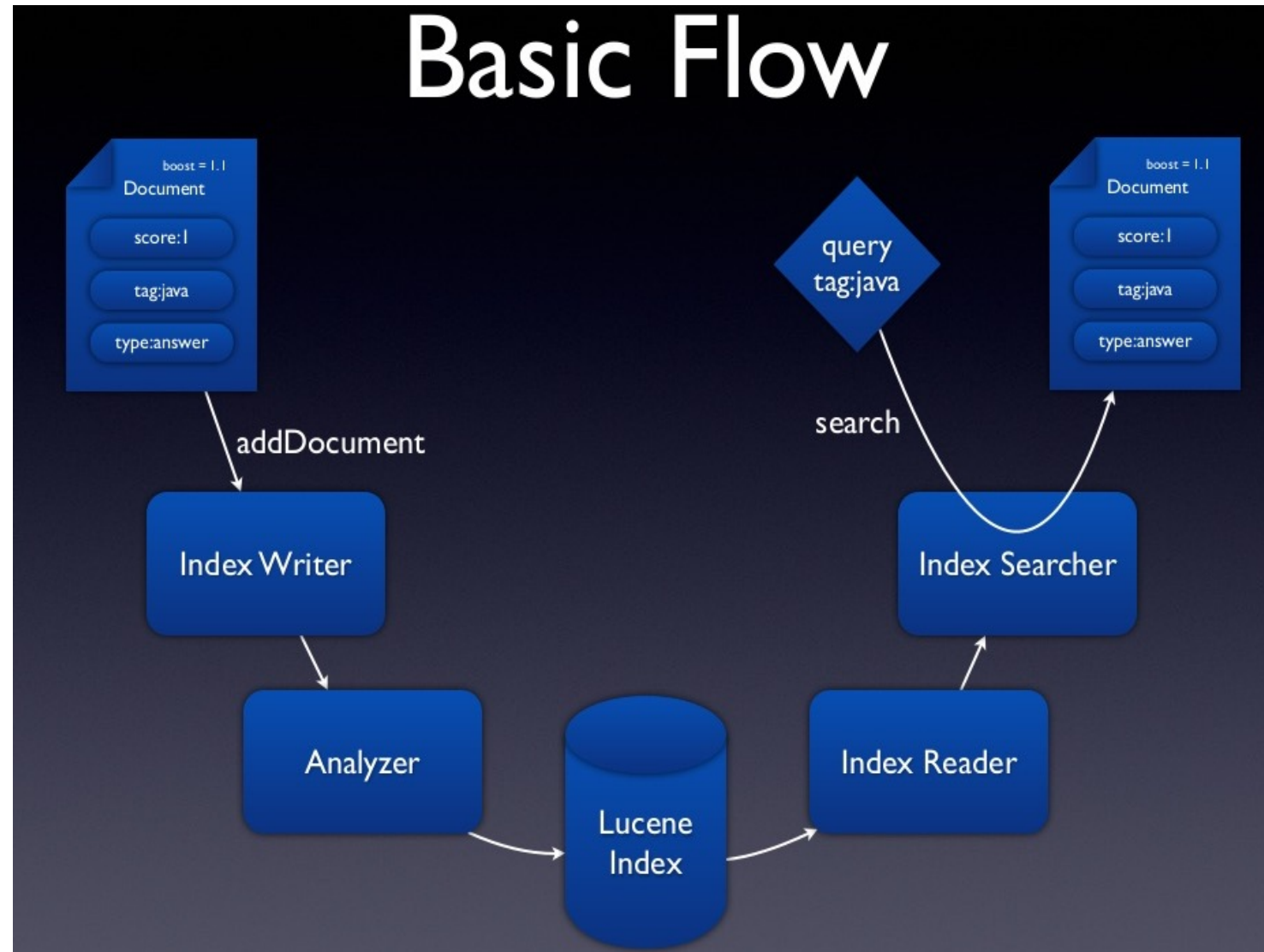
# Apache Lucene Highlights

- Fast, high performance, scalable search/IR library
- Open source
- Initially developed by Doug Cutting (Also author of Hadoop)
- Indexing and Searching
- Inverted Index of documents
- Provides advanced Search options like **synonyms**, stopwords, based on **similarity**, **proximity**.
- <http://lucene.apache.org/>

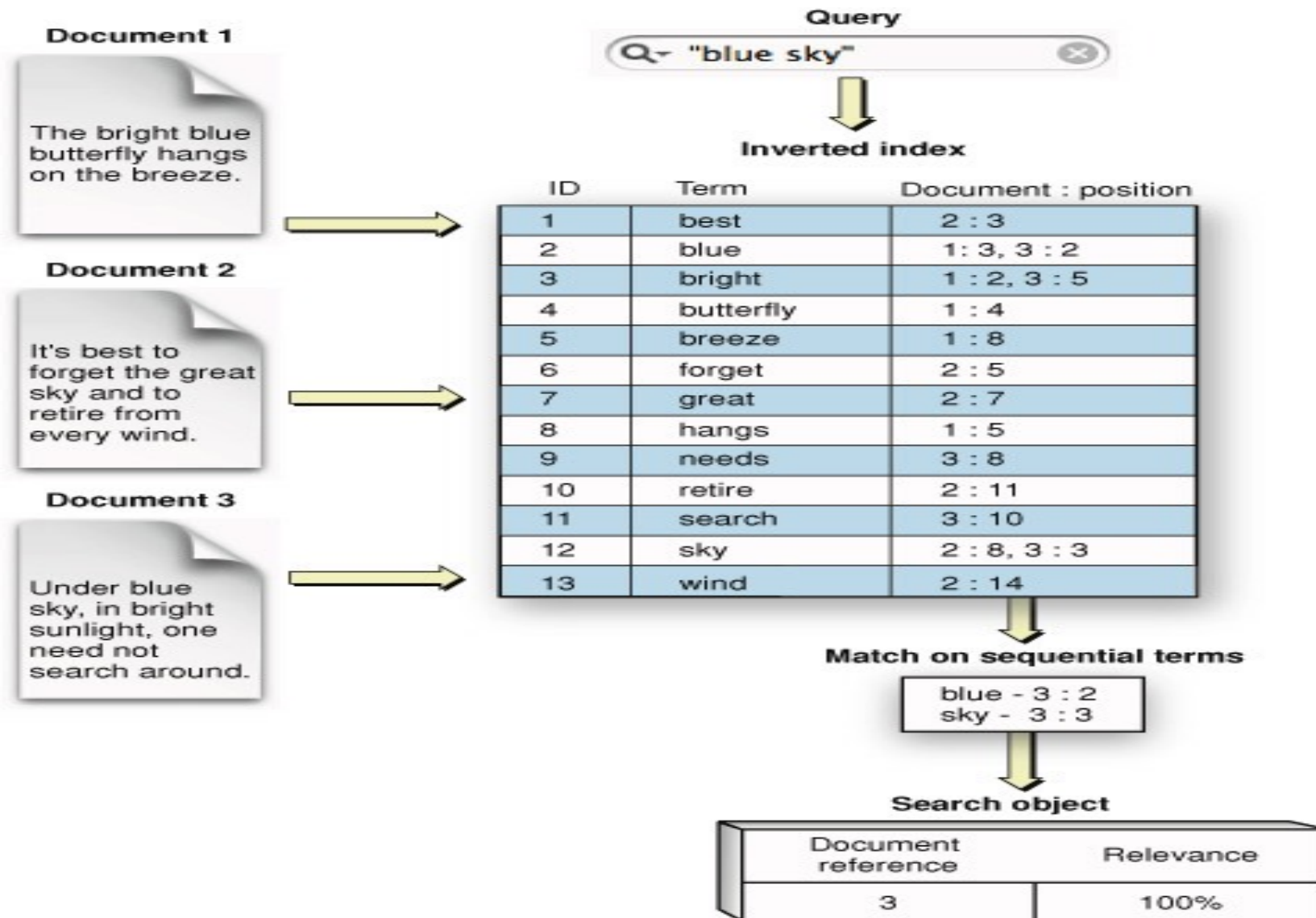
# Lucene - Architecture



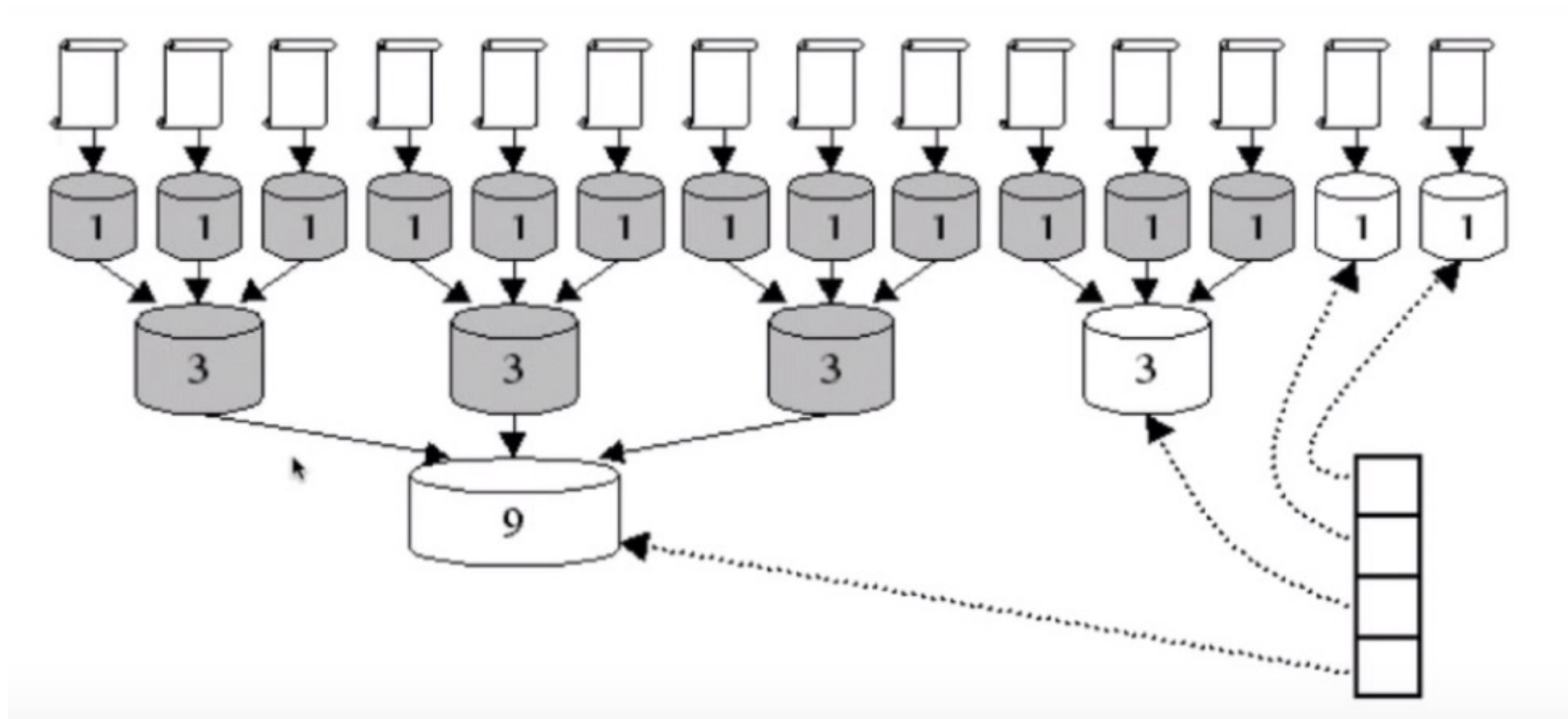
# Lucene – Work Flow



# Lucene Internals - Inverted Index

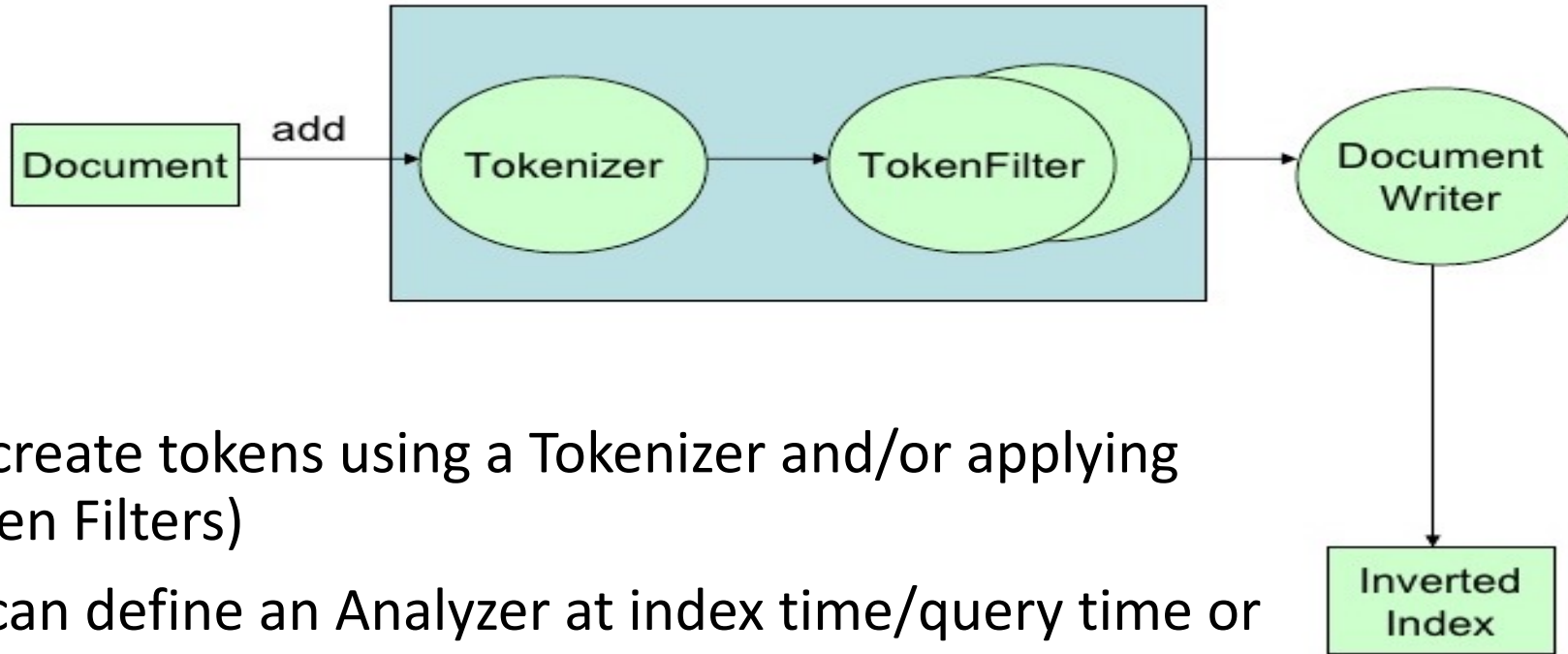


# Lucene - Indexing





# Indexing Pipeline



- Analyzer : create tokens using a Tokenizer and/or applying Filters (Token Filters)
- Each field can define an Analyzer at index time/query time or the both at same time.

Credit : <http://www.slideshare.net/otisg/lucene-introduction>

# Analysis Process - Tokenizer

## WhitespaceAnalyzer

Simplest built-in analyzer

The quick brown fox jumps over the lazy dog.



[The] [quick] [brown] [fox] [jumps] [over] [the] [lazy] [dog.]

Tokens

# Analysis Process - Tokenizer

## SimpleAnalyzer

Lowercases, split at non-letter boundaries

The quick brown fox jumps over the lazy dog.



[The] [quick] [brown] [fox] [jumps] [over] [the] [lazy] [dog.]

Tokens

# Some common analyzer

- **WhitespaceAnalyzer** : Splits text at whitespaces, just as the name indicates. In fact, this is the only thing this analyzer does.
- **SimpleAnalyzer** : Splits text at non-letter characters and lowercases resulting tokens.
- **StopAnalyzer** : Splits text at non-letter characters, lowercases resulting tokens, and removes stopwords.
- **StandardAnalyzer** : Splits text using a grammar-based tokenization, normalizes and lowercases tokens, removes stopwords, and discards punctuations. It can be used to extract company names, e-mail addresses, model numbers, and so on. This analyzer is great for general usage.
- **SnowballAnalyzer**: This analyzer is similar to StandardAnalyzer with an additional SnowballFilter for stemming.

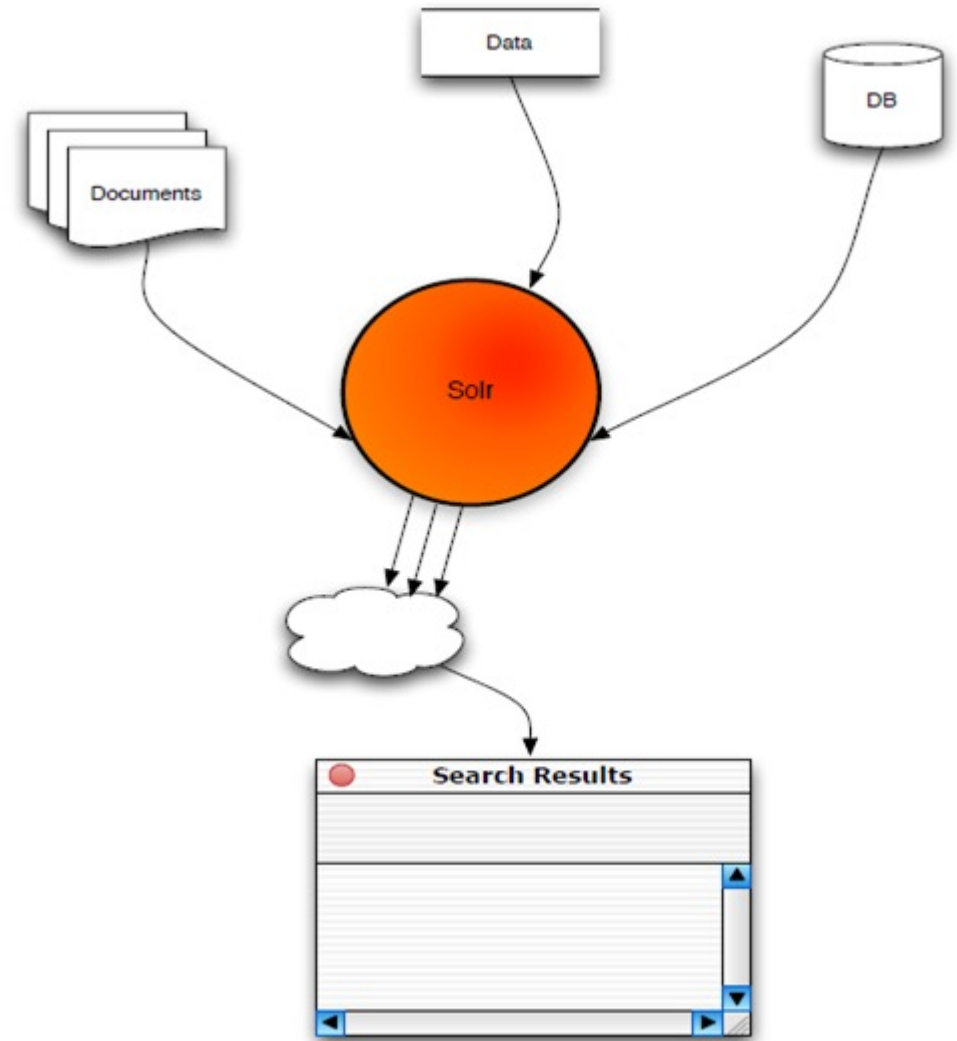
# Apache Solr



# Apache Solr

- Created by Yonik Seeley for CNET
- Enterprise Search platform for Apache Lucene
- Open source
- Highly reliable, scalable, fault tolerant
- Support distributed Indexing (SolrCloud), Replication, and load balanced querying
- <http://lucene.apache.org/solr>

# High level overview



Source: <http://www.slideshare.net/erikhatcher/solr-search-at-the-speed-of-light>

# Apache Solr - Features

- Full-text search
- Faceted search (similar to groupby clause in RDBMS)
- Scalability
  - Caching
  - Replication
  - Distributed search
- Near real-time indexing
- Geospatial search
- And many more : highlighting, database integration, rich document (e.G., Word, PDF) handling



# Solr – schema.xml

- Types with index and query Analyzers - similar to data type
- Fields with name, type and options
- **Unique Key** : Unique Identifier of a document. For e.g. “id”
- **Dynamic Fields** : *Dynamic fields* allow Solr to index fields that you did not explicitly define in your schema. For e.g. fieldName: \*\_i or \*\_txts
- **Copy Fields** : Solr has a mechanism for making copies of fields so that you can apply several distinct field types to a single piece of incoming information. field ‘a’ populates field ‘b’ with its value before tokenizing (having different analyzer/filter).

# Solr – Content Analysis

- Field Attributes

- **Name** : Name of the field
- **Type** : Data-type (FieldType) of the field
- **Indexed** : Should it be indexed (indexed="true/false")
- **Stored** : Should it be stored (stored="true/false")
- **Required** : is it a mandatory field (required="true/false")
- **Multi-Valued** : Would it will contains multiple values e.g. text: pizza, food (multiValued="true/false")

e.g. `<field name="id" type="string" indexed="true" stored="true" required="true" multiValued="false" />`

# Solr – solrconfig.xml

- Data dir: where all index data will be stored
- Index configuration
- Cache configurations
- Request Handler configuration
- Search components, response writers, query parsers

# Query Types

- Single and multi term queries
  - ex fieldname:value or title: software engineer
- +, -, AND, OR NOT operators.
  - ex. title: (software AND engineer)
- Range queries on date or numeric fields,
  - ex: timestamp: [ \* TO NOW ] or price: [ 1 TO 100 ]
- Boost queries:
  - e.g. title:Engineer ^1.5 OR text:Engineer
- Fuzzy search : is a search for words that are similar in spelling
  - e.g. roam~0.8 => noam
- Proximity Search : with a sloppy phrase query. The close together the two terms appear, higher the score.
  - ex “apache lucene”~20 : will look for all documents where “apache” word occurs within 20 words of “lucene”

# Solr/Lucene Use-cases

- Search
- Analytics
- NoSQL datastore
- Auto-suggestion / Auto-correction
- Recommendation Engine (MoreLikeThis)
- Relevancy Engine (Feedback to other applications)
- Solr as a White-List
- GeoSpatial based Search

# Search

- **Application**
  - Eclipse, Hibernate search
- **E-Commerce :**
  - Flipkart.com, Infibeam.com, Buy.com, Netflix.com, ebay.com
- **Jobs**
  - Indeed.com, Simplyhired.com, Naukri.com
- **Auto**
  - AOL.com
- **Travel**
  - Cleartrip.com
- **Social Network**
  - Twitter.com, LinkedIn.com, mylife.com

Source: <http://www.quora.com/Which-major-companies-are-using-Solr-for-search>

# Search (Contd.)

- **Search Engine**
  - Yandex.ru, DuckDuckGo.com
- **News Paper**
  - Guardian.co.uk
- **Music/Movies**
  - Apple.com, Netflix.com
- **Events**
  - Stubhub.com, Eventbrite.com
- **Cloud Log Management**
  - Loggly.com
- **Others**
  - Whitehouse.gov

# Faceting

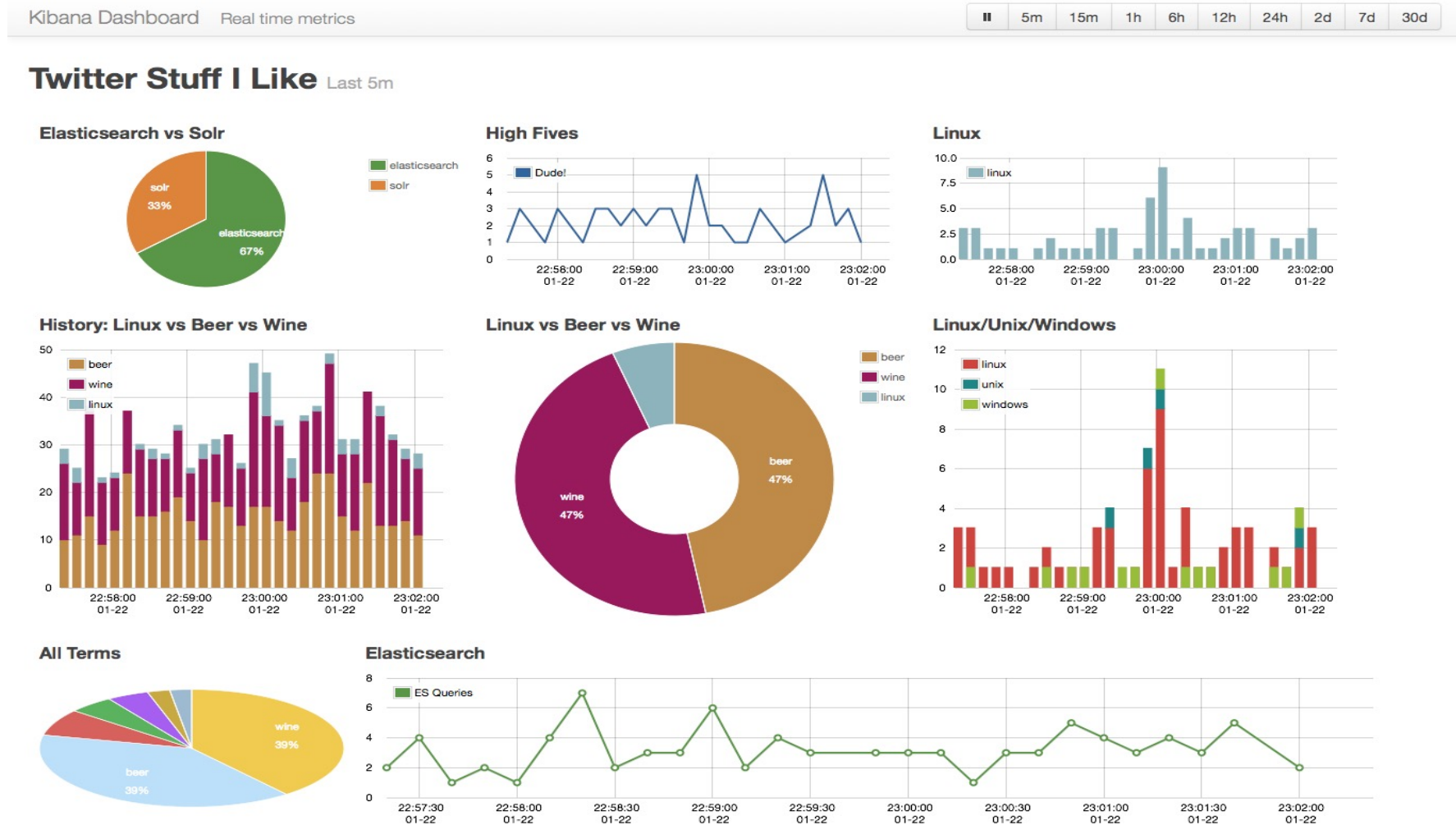
- Grouping results based on field value
- Facet on: field terms, queries, date ranges
- &facet=on  
&facet.field=job\_title  
&facet.query=salary:[30000 TO 100000]
- <http://wiki.apache.org/solr/SimpleFacetParameters>

Filter your search	
<b>Publication date</b>	
▸ This week (17)	
▸ Last week (3)	
<b>Cities</b>	
▸ Hyderabad, India (96)	
▸ Mumbai, India (53)	
▸ Bangalore, India (48)	
▸ Chennai, India (24)	
▸ Jodhpur, India (24)	
▸ Pune, India (18)	
▸ Indore, India (8)	
▸ Noida, India (8)	
▸ New Delhi, India (5)	
▸ Noida Area, India (2)	
▸ Pune Area, India (2)	
▸ Ahmedabad Area, India (1)	
▸ Navi Mumbai, India (1)	
<b>▼ Salary Estimate</b>	
\$50,000+ (56176)	
\$70,000+ (40059)	
\$90,000+ (20686)	
\$110,000+ (9094)	
\$130,000+ (3942)	
<b>▼ Title</b>	
Java Developer (1911)	
Software Engineer (1334)	
Senior Software Developer (752)	
Senior Software Engineer (694)	
Senior Java Developer (575)	
Software Developer (469)	
Web Developer (345)	
Sr. Java Developer (304)	
Software Development Engineer (250)	
Android Developer (229)	
Web Application Developer (216)	
Principal Software Engineer (20)	
Sr. Software Engineer (197)	
Application Developer (177)	

Source: [www.career9.com](http://www.career9.com), [www.indeed.com](http://www.indeed.com)



# Analytics



- Analytics source : [Kibana.org](http://Kibana.org) based on [ElasticSearch](http://ElasticSearch) and [Logstash](http://Logstash)
- Image Source : <http://semicomplete.com/presentations/logstash-monitorama-2013/#/8>

# Autosuggestion

Enter your keywords:

teach

Did you mean: **teaching**

<b>teach</b>	17
teachers	2
teacher	1
teach book	15
teach world	11
teach wide	11
teach teaching	9
teach computer	9

Find dinn|

<b>dinner</b>
<b>dinner</b> restaurant
<b>dinner</b> and drinks
<b>dinner</b> cruise
<b>dinner</b> and dancing
<b>dinner</b> date
<b>dinner</b> theater
<b>dinner</b> show
<b>dinner</b> buffet
<b>dinner</b> and live jazz

Source: [www.drupal.org](http://www.drupal.org) , [www.yelp.com](http://www.yelp.com)

# Integration

- Clustering (Solr-Carrot2)
- Named Entity extraction (Solr-UIMA)
- SolrCloud (Solr-Zookeeper)
- Parsing of many Different File Formats (Solr-Tika)
- Machine Learning/Data Mining (Apache Mahout)
- Large scale Indexing (Hadoop)

# SolrCtl Command

- The solrctl utility is a wrapper shell script included with Cloudera Search for managing collections, instance directories, configs, Apache Sentry permissions, and more.

## Syntax

The general `solrctl` command syntax is:

```
solrctl [options] command [command-arg] [command [command-arg]] ...
```

Source: [https://www.cloudera.com/documentation/enterprise/5-14-x/topics/search\\_solrctl\\_ref.html](https://www.cloudera.com/documentation/enterprise/5-14-x/topics/search_solrctl_ref.html)

# SolrCtl Collection Commands

Source: [https://www.cloudera.com/documentation/enterprise/5-14-x/topics/search\\_solrctl\\_ref.html](https://www.cloudera.com/documentation/enterprise/5-14-x/topics/search_solrctl_ref.html)

```
collection [--create <name> -s <numShards>
            [-a]
            [-c <configName>]
            [-r <replicationFactor>]
            [-m <maxShardsPerHost>]
            [-n <createHostSet>]]
            [--delete <name>]
            [--reload <name>]
            [--stat <name>]
            [--deletedocs <name>]
            [--list]
            [--create-snapshot <snapshotName> -c <collectionName>]
            [--delete-snapshot <snapshotName> -c <collectionName>]
            [--list-snapshots <collectionName>]
            [--describe-snapshot <snapshotName> -c <collectionName>]
            [--prepare-snapshot-export <snapshotName> -c <collectionName> -d <destDir>]
            [--export-snapshot <snapshotName> [-s <sourceDir>] [-c <collectionName>] -d
            [--restore name -b <backupName> -l <backupLocation> -i <requestId>
            [-a]
            [-c <configName>]
            [-r <replicationFactor>]
            [-m <maxShardsPerNode>]]
            [--request-status <requestId>]
```

- solrctl collection --list

***Lists the collection***

- solrctl config --create logs\_config predefinedTemplate -p immutable=false

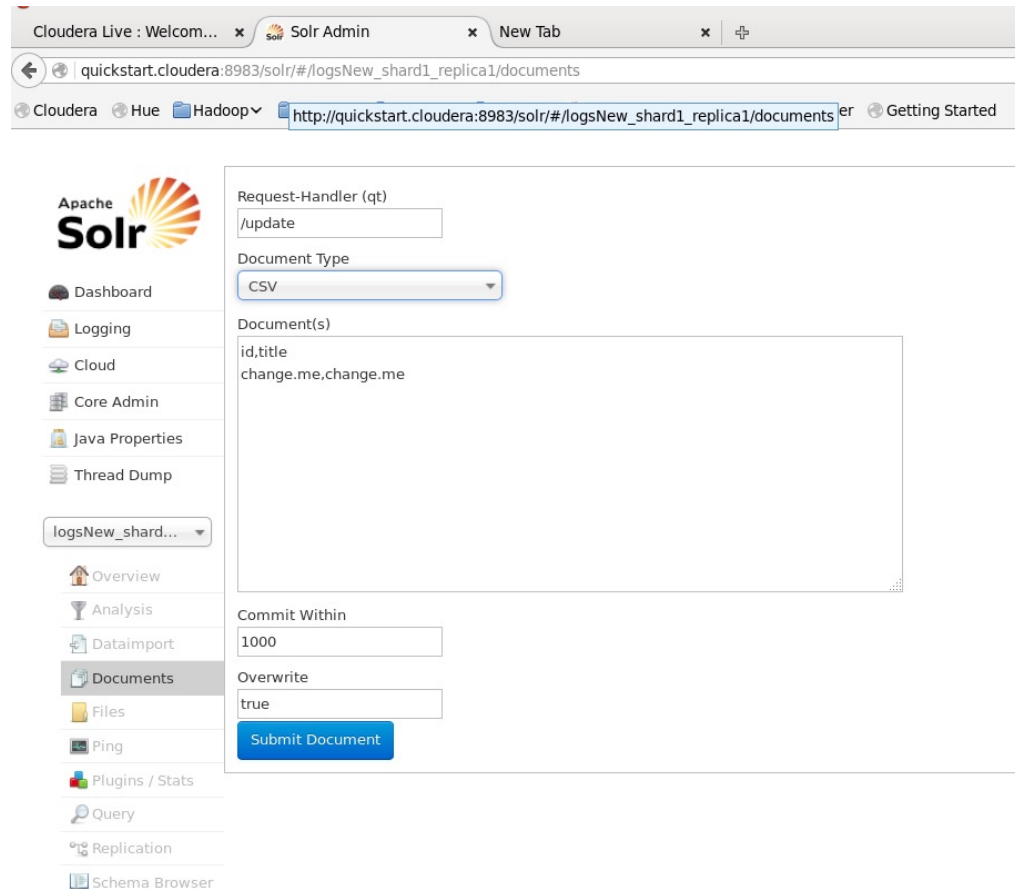
***logs\_config => config name***

***predefinedTemplate => existing config template***

- solrctl instancedir --generate \$HOME/logs\_config
- solrctl collection --create logNew2 -c logs\_config

***logNew2 => collection Name***

# Update or add data to collection



The screenshot shows the Apache Solr Admin interface in a web browser. The browser tabs include 'Cloudera Live : Welcom...', 'Solr Admin', and 'New Tab'. The address bar shows the URL 'quickstart.cloudera:8983/solr/#/logsNew\_shard1\_replica1/documents'. The browser's bookmark bar contains 'Cloudera', 'Hue', 'Hadoop', and a link to the current page.

The Solr Admin interface features a sidebar on the left with the following navigation links: Dashboard, Logging, Cloud, Core Admin, Java Properties, Thread Dump, logsNew\_shard1\_replica1 (selected), Overview, Analysis, Dataimport, Documents (highlighted), Files, Ping, Plugins / Stats, Query, Replication, and Schema Browser.

The main content area displays the 'Documents' page for the selected shard. It includes the following fields and controls:

- Request-Handler (qt):** A text input field containing '/update'.
- Document Type:** A dropdown menu with 'CSV' selected.
- Document(s):** A large text area containing the text 'id,title' and 'change.me,change.me'.
- Commit Within:** A text input field containing '1000'.
- Overwrite:** A text input field containing 'true'.
- Submit Document:** A blue button to submit the document.

# Query Syntax

Solr Admin - Mozilla

Cloudera Live : Welcom... x Solr Admin x New Tab x

quickstart.cloudera:8983/solr/#/logsNew\_shard1\_replica1/query

Cloudera Hue Hadoop HBase Impala Spark Solr Oozie Cloudera Manager Getting Started

Apache Solr

Dashboard  
Logging  
Cloud  
Core Admin  
Java Properties  
Thread Dump

logsNew\_shard1\_replica1

Overview  
Analysis  
Dataimport  
Documents  
Files  
Ping  
Plugins / Stats  
Query  
Replication  
Schema Browser

Request-Handler (qt)

/select

common

q

\*:\*

fq

sort

start, rows

0 10

fl

df

Raw Query Parameters

key1=val1&key2=val2

wt

json

☒ indent

☐ debugQuery

☐ dismax

☐ edismax

☐ hl

Cloudera Live : Welcom... x Solr Admin x New Tab x

quickstart.cloudera:8983/solr/#/logsNew\_shard1\_replica1/query

Cloudera Hue Hadoop HBase Impala Spark Solr Oozie Cloudera Manager Getting Started

Apache Solr

Dashboard  
Logging  
Cloud  
Core Admin  
Java Properties  
Thread Dump

logsNew\_shard1\_replica1

Overview  
Analysis  
Dataimport  
Documents  
Files  
Ping  
Plugins / Stats  
Query  
Replication  
Schema Browser

fq

sort

start, rows

0 10

fl

df

Raw Query Parameters

key1=val1&key2=val2

wt

json

☒ indent

☐ debugQuery

☐ dismax

☐ edismax

☐ hl

☐ facet

☐ spatial

☐ spellcheck

Execute Query



# Results

The screenshot displays the Apache Solr Admin web interface. The browser tabs include 'Cloudera Live: Welcom...', 'Solr Admin', and 'New Tab'. The address bar shows the URL: `http://quickstart.cloudera:8983/solr/#/logsNew_shard1_replica1/query`. The left sidebar contains navigation links: Dashboard, Logging, Cloud, Core Admin, Java Properties, Thread Dump, logsNew\_shard1\_replica1 (selected), Overview, Analysis, Dataimport, Documents, Files, Ping, Plugins / Stats, Query (selected), Replication, and Schema Browser.

The main content area is divided into two panels. The left panel, titled 'Request-Handler (qt)', shows the query configuration: `/select` under 'common', `q=*` in the query field, `fq` in the filter field, `sort` in the sort field, `start=0` and `rows=10` in the start/rows fields, `fl` in the field list field, `df` in the distribution field, `key1=val1&key2=val2` in the Raw Query Parameters field, `wt=json` in the wrapper type dropdown, and checkboxes for `indent` (checked), `debugQuery`, `dismax`, `edismax`, and `hl`.

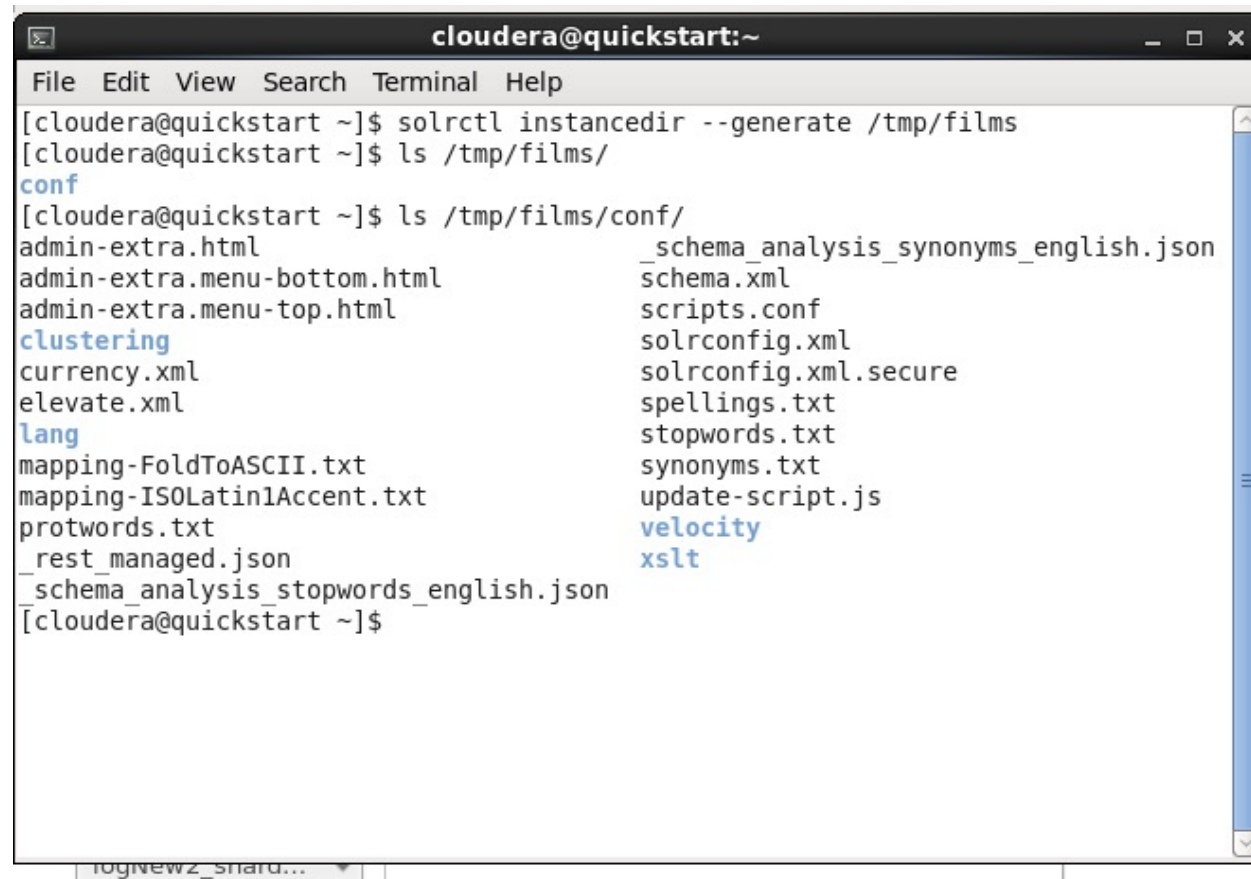
The right panel displays the JSON response for the query: `http://quickstart.cloudera:8983/solr/#/logsNew_shard1_replica1/select?q=*&wt=json&indent=true`. The response is a JSON object with the following structure:

```
{
  "responseHeader": {
    "status": 0,
    "QTime": 14,
    "params": {
      "indent": "true",
      "q": "/*",
      "_": "1529968927625",
      "wt": "json"
    }
  },
  "response": {
    "numFound": 12,
    "start": 0,
    "docs": [
      {
        "id": "book1",
        "cat_s": "fantasy",
        "pubyear_i": 2010,
        "title_t": "The Way of Kings",
        "author_s": "Brandon Sanderson",
        "series_s": "The Stormlight Archive",
        "sequence_i": 1,
        "publisher_s": "Tor",
        "_version_": 1604274044759179300
      },
      {
        "id": "book2",
        "cat_s": "fantasy",
        "pubyear_i": 1996,
        "title_t": "A Game of Thrones",
        "author_s": "George R.R. Martin",
        "series_s": "A Song of Ice and Fire"
      }
    ]
  }
}
```

# Creating Schema Config

- `solrctl instancedir --generate /tmp/films`

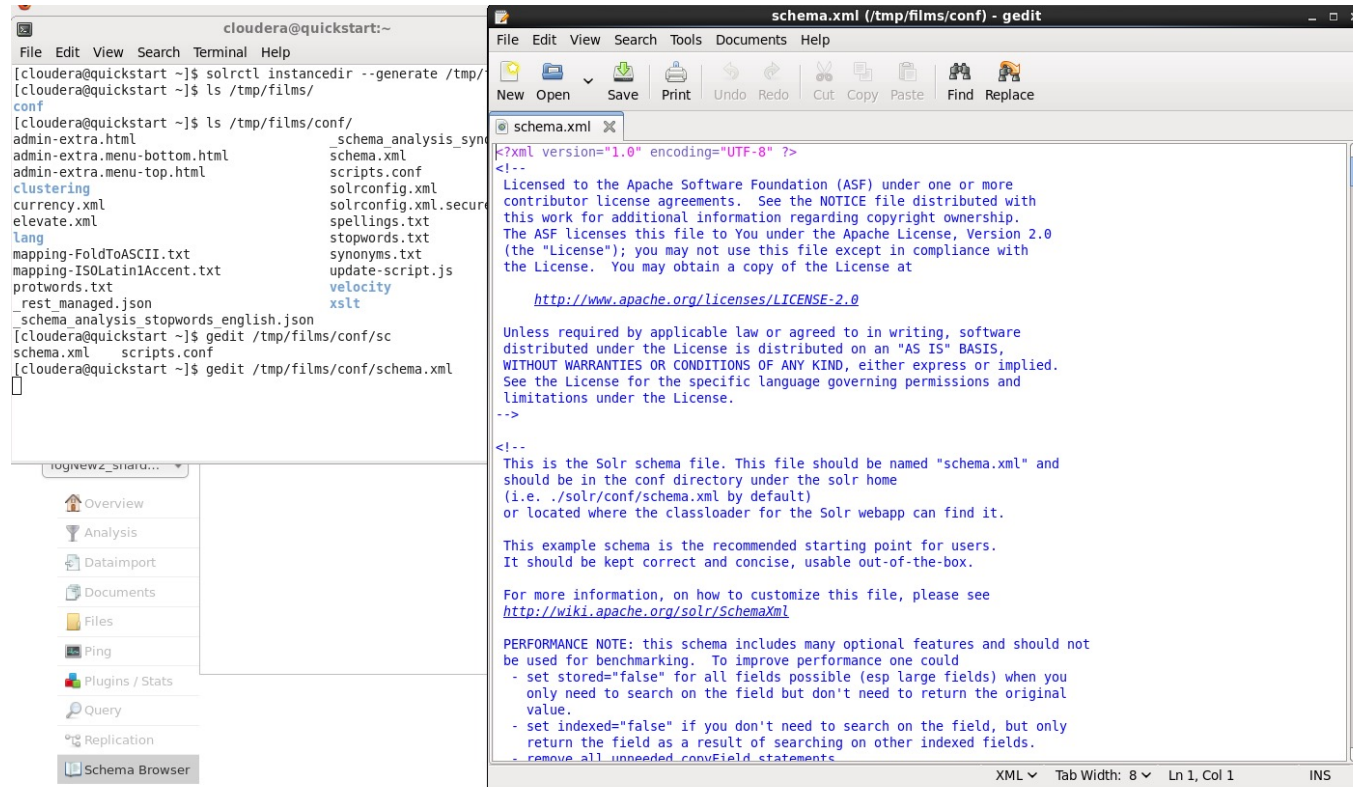
# Editing Schema Config



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ solrctl instancedir --generate /tmp/films  
[cloudera@quickstart ~]$ ls /tmp/films/  
conf  
[cloudera@quickstart ~]$ ls /tmp/films/conf/  
admin-extra.html                                _schema_analysis_synonyms_english.json  
admin-extra.menu-bottom.html                    schema.xml  
admin-extra.menu-top.html                       scripts.conf  
clustering                                       solrconfig.xml  
currency.xml                                    solrconfig.xml.secure  
elevate.xml                                     spellings.txt  
lang                                             stopwords.txt  
mapping-FoldToASCII.txt                         synonyms.txt  
mapping-ISOLatin1Accent.txt                     update-script.js  
protwords.txt                                   velocity  
_rest_managed.json                             xslt  
_schema_analysis_stopwords_english.json  
[cloudera@quickstart ~]$
```

# Editing Schema Config :

## gedit /tmp/films/conf/schema.xml



The image shows a terminal window on the left and a gedit editor window on the right. The terminal window, titled 'cloudera@quickstart:~', shows the following commands and output:

```
[cloudera@quickstart ~]$ solrctl instancedir --generate /tmp/
[cloudera@quickstart ~]$ ls /tmp/films/
conf
[cloudera@quickstart ~]$ ls /tmp/films/conf/
admin-extra.html          schema.analysis_synon
admin-extra.menu-bottom.html schema.xml
admin-extra.menu-top.html scripts.conf
clustering                solrconfig.xml
currency.xml              solrconfig.xml.secure
elevate.xml               spellings.txt
lang                      stopwords.txt
mapping-FoldToASCII.txt   synonyms.txt
mapping-ISOLatin1Accent.txt update-script.js
protwords.txt             velocity
_rest_managed.json        xslt
schema.analysis_stopwords_english.json
schema.xml                scripts.conf
[cloudera@quickstart ~]$ gedit /tmp/films/conf/sc
[cloudera@quickstart ~]$ gedit /tmp/films/conf/schema.xml
```

The gedit editor window, titled 'schema.xml (/tmp/films/conf) - gedit', shows the content of the schema.xml file. The content is as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements.  See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License.  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->

<!--
This is the Solr schema file. This file should be named "schema.xml" and
should be in the conf directory under the solr home
(i.e. ./solr/conf/schema.xml by default)
or located where the classloader for the Solr webapp can find it.

This example schema is the recommended starting point for users.
It should be kept correct and concise, usable out-of-the-box.

For more information, on how to customize this file, please see
http://wiki.apache.org/solr/SchemaXml

PERFORMANCE NOTE: this schema includes many optional features and should not
be used for benchmarking.  To improve performance one could
- set stored="false" for all fields possible (esp large fields) when you
  only need to search on the field but don't need to return the original
  value.
- set indexed="false" if you don't need to search on the field, but only
  return the field as a result of searching on other indexed fields.
- remove all unneeded <copyField> statements
```

# Creating new fieldType (Use this to create your own directed\_by field)

- <http://www.solrtutorial.com/schema-xml.html>
- `<fields>`
  - `<field name="id" type="string" indexed="true" stored="true" required="true" />`
  - `<field name="name" type="textgen" indexed="true" stored="true"/>`
  - ...`</fields>`

# Film Dataset commands

## **Instancedir and collection**

- `solrctl instancedir --create films /tmp/films`
- `solrctl collection --create films`

## **Edit Schema**

- `ls /tmp/films/conf/`
- `gedit /tmp/films/conf/schema.xml`

# References

- <http://www.lucenetutorial.com/lucene-vs-solr.html>
- <https://lucene.apache.org/solr/>
- [https://lucene.apache.org/solr/guide/6\\_6/the-standard-query-parser.html](https://lucene.apache.org/solr/guide/6_6/the-standard-query-parser.html)
- [https://lucene.apache.org/solr/guide/8\\_5/solr-tutorial.html](https://lucene.apache.org/solr/guide/8_5/solr-tutorial.html)

# Commands Details

- `solrctl instancedir -- generate` - Use this command to generate new instance.
- `solrctl instancedir -- create <collection_name>` - To upload the contents of instance directory to Zookeeper.
- `solrctl collection -- create <collection_name>` - Used to create new collection.



# Commands

- `solrctl config --create logs_config predefinedTemplate -p immutable=false`
- `solrctl instancedir --generate $HOME/logs_config`
- `solrctl collection --create logNew2 -c logs_config`
- `solrctl instancedir --generate /tmp/films`
- `ls /tmp/films/conf`