

# **PROJECT REPORT**

## **INDEXING WIKI DUMPS ON CLOUD** **USING ELK STACK & SOLR**

## TABLE OF CONTENTS:

<b>PROJECT DESCRIPTION:</b> .....	<b>3</b>
<b>INDEXING</b> .....	<b>3</b>
<b>INDEXING WIKI DUMPS On GOOGLE CLOUD PLATFORM using ELK Stack:</b> .....	<b>3</b>
ELASTICSEARCH .....	3
ADVANTAGES OF ELASTIC SEARCH OVER SOLR .....	4
INSTALLATION OF ELASTICSEARCH .....	4
KIBANA.....	5
INSTALLATION OF KIBANA.....	6
LOGSTASH.....	7
INSTALLATION OF LOGSTASH .....	7
SIMPLE WIKIPEDIA DATASET: .....	8
LOADING WIKI DATAAND INDEXING: .....	8
INDEXING WIKI QUOTE DATA: .....	8
KIBANA VISUALIZATION .....	10
<b>INDEXING WIKI DUMPS On AMAZON WEB SERVICES using ELK Stack:</b> .....	<b>11</b>
Elasticsearch Installation .....	12
Kibana Installation .....	12
Logstash Installation.....	13
KIBANA VISUALIZATIONS THROUGH AWS.....	13
Earthquakes Data.....	13
<b>INDEXING WIKI DUMPS ON MICROSOFT AZURE USING SOLR:</b> .....	<b>14</b>
SOLR:.....	14
ARCHITECTURE OF SOLR.....	14
INSTALLATION STEPS OF SOLR:.....	15
INSTALLATION OF SOLR IN AZURE:.....	16
INDEXED WIKIPEDIA DATA IN SOLR: .....	17
<b>CREDITS</b> .....	<b>17</b>

## PROJECT DESCRIPTION:

The project agenda was loading Wiki Datasets and Indexing them on following Cloud Platforms using Elasticsearch , Logstash and Kibana Stack & SOLR

- Amazon Web Services
- Google Cloud Platform
- Microsoft Azure

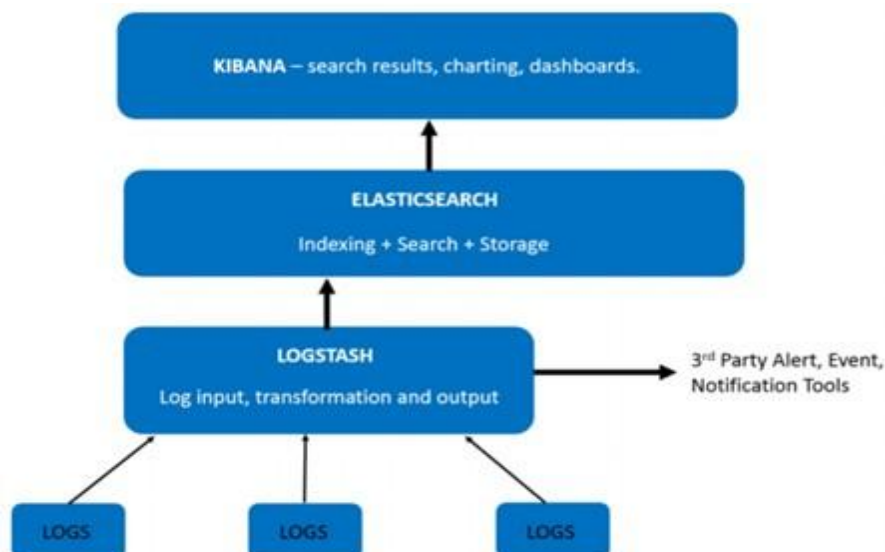
## INDEXING:

Indexing is a way to optimize performance of a database by minimizing the number of disk accesses required when a query is processed. An index or database index is a data structure which is used to quickly locate and access the data in a database table.

## INDEXING WIKI DUMPS On GOOGLE CLOUD PLATFORM using ELK Stack:

### ELASTICSEARCH:

Elastic search is an open source, broadly-distributable, readily-scalable, enterprise-grade search engine based on Lucene and released under the terms of the Apache License. It is Java-based and designed to operate in real time. It can search and index document files in diverse formats. It was designed to be used in distributed environments by providing flexibility and scalability. Now, elastic search is the most popular enterprise search engine followed by Apache Solr, also based on Lucene.



## ADVANTAGES OF ELASTIC SEARCH OVER SOLR

### Build on top of lucene

Elastic search is built on top of Lucene, which is a full-featured information retrieval library, so it provides the most powerful full-text search capabilities of any open source product.

### Document- oriented

Elastic search is document-oriented. It stores real world complex entities as structured JSON documents and indexes all fields by default, with a higher performance result.

### Speed

Elasticsearch is able to execute complex queries extremely fast. It also caches almost all of the structured queries commonly used as a filter for the result set and executes them only once. For every other request which contains a cached filter, it checks the result from the cache. This saves the time parsing and executing the query improving the speed.

### Structured search

Elastic Search is schema free, it accepts JSON documents, as well as tries to detect the data structure, index the data, and make it searchable.

### Data record

Elasticsearch records any changes made in transactions logs on multiple nodes in the cluster to minimize the chance of data loss.

## INSTALLATION OF ELASTICSEARCH:

### Firewall Creation:

Creating Firewall rules for access to ports 9200 and 5601 for Elastic search and Kibana.

[Ingress](#) [Egress](#)

<input type="checkbox"/> Name	Targets	Source filters	Protocols / ports	Action	Priority	Network <a href="#">^</a>
<input type="checkbox"/> elasticsearch	Apply to all	IP ranges: 0.0.0.0/0	tcp:9200	Allow	1000	default
<input type="checkbox"/> kibana	Apply to all	IP ranges: 0.0.0.0/0	tcp:5601	Allow	1000	default
<input type="checkbox"/> default-allow-icmp	Apply to all	IP ranges: 0.0.0.0/0	icmp	Allow	65534	default
<input type="checkbox"/> default-allow-internal	Apply to all	IP ranges: 10.128.0.0/9	tcp:0-65535, udp:0-65535, 1 more <a href="#">▼</a>	Allow	65534	default
<input type="checkbox"/> default-allow-rdp	Apply to all	IP ranges: 0.0.0.0/0	tcp:3389	Allow	65534	default
<input type="checkbox"/> default-allow-ssh	Apply to all	IP ranges: 0.0.0.0/0	tcp:22	Allow	65534	default

To install Java

**\$ sudo apt-get install default-jre**

This will fetch the latest ElasticSearch Version for us

**\$ wget -qO - https://packages.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -**

This will complete the installation

**\$ sudo apt-get install elasticsearch**

Find the line referring to the network.host portion. It will be commented out.

Uncomment the file and make it read network.host "0.0.0.0"

**\$ sudo vi /etc/elasticsearch/elasticsearch.yml**

**ELASTIC SEARCH is installed and running successfully.**

```
venkateshumamaheswaran@instance-1:~$ curl localhost:9200
{
  "name" : "j1H4DEo",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "Oaen330HRfSMhUSo4GYWZw",
  "version" : {
    "number" : "5.6.3",
    "build_hash" : "1a2f265",
    "build_date" : "2017-10-06T20:33:39.012Z",
    "build_snapshot" : false,
    "lucene_version" : "6.6.1"
  },
  "tagline" : "You Know, for Search"
}
```

## KIBANA

Kibana is an open source analytics and visualization platform designed to work with Elasticsearch. You use Kibana to search, view, and interact with data stored in Elasticsearch indices. You can easily perform advanced data analysis and visualize your data in a variety of charts, tables, and maps.

Kibana makes it easy to understand large volumes of data. Its simple, browser-based interface enables you to quickly create and share dynamic dashboards that display changes to Elasticsearch queries in real time.

The different views in Kibana are as follows:

The **discover** view is used to view a list of documents and search for specific documents.

The **visualize** view is used to create visualizations like graphs from the data. We can add those visualization to **dashboards** to have an overview of you data at a glance.

**Timelion** was formerly a plugin and is now build in. It's used to make advanced timeseries analysis.

The **management** tab are the settings of Kibana where we can add index patterns and tune some advanced settings.

The **Dev Tools** currently only contain the so called Console, which was formerly known as the Sense plugin in Elasticsearch. We can use it to send JSON directly to Elasticsearch and more meant for developers or advanced users.

## INSTALLATION OF KIBANA

This will establish the source for Kibana

```
$ echo "deb http://packages.elastic.co/kibana/5.3/debian stable main" | sudo tee -a /etc/apt/sources.list.d/kibana-5.3.x.list
```

Setting up for Kibana installation

```
$ sudo apt-get update
```

```
$ sudo apt-get install kibana
```

```
venkateshumamaheswaran@instance-1:~$ curl -XGET 'http://localhost:5601/'
<script>var hashRoute = '/app/kibana';
var defaultRoute = '/app/kibana';

var hash = window.location.hash;
if (hash.length) {
  window.location = hashRoute + hash;
} else {
  window.location = defaultRoute;
}</script>venkateshumamaheswaran@instance-1:~$
```

```
venkateshumamaheswaran@instance-1:~$ curl -XGET 'http://localhost:5601/'
<script>var hashRoute = '/app/kibana';
var defaultRoute = '/app/kibana';

var hash = window.location.hash;
if (hash.length) {
  window.location = hashRoute + hash;
} else {
  window.location = defaultRoute;
}</script>venkateshumamaheswaran@instance-1:~$ sudo service kibana status
• kibana.service - no description given
   Loaded: loaded (/lib/systemd/system/kibana.service; disabled; vendor preset: enabled)
   Active: active (running) since Tue 2017-10-24 16:31:54 UTC; 10h ago
 Main PID: 32303 (node)
    Tasks: 9 (limit: 4915)
   CGroup: /system.slice/kibana.service
           └─32303 /opt/kibana/bin/../node/bin/node /opt/kibana/bin/../src/cli
```

```
$ sudo service kibana start
```

```

venkateshumamaheswaran@instance-1:~$ sudo service kibana status
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; disabled; vendor preset: enabled)
   Active: active (running) since Sun 2017-12-03 01:47:24 UTC; 39min ago
   Main PID: 1719 (node)
   Tasks: 10 (limit: 4915)
   CGroup: /system.slice/kibana.service
           └─1719 /usr/share/kibana/bin/../node/bin/node --no-warnings /usr/share/kibana/bin/../src/cli -c /etc/kibana/kibana.yml

Dec 03 01:47:39 instance-1 kibana[1719]: {"type":"log","@timestamp":"2017-12-03T01:47:39Z","tags":["status","plugin:console@5.6.3","info"],"pid":1719,"state":"green","message":"Status changed from uninitialized to green - Ready","prevState":"uninitialized","prevMsg":"uninitialized"}
Dec 03 01:47:39 instance-1 kibana[1719]: {"type":"log","@timestamp":"2017-12-03T01:47:39Z","tags":["status","plugin:elasticsearch@5.6.3","error"],"pid":1719,"state":"red","message":"Status changed from yellow to red - Unable to connect to Elasticsearch at http://localhost:9200.","prevState":"yellow","prevMsg":"Waiting for Elasticsearch"}
Dec 03 01:47:39 instance-1 kibana[1719]: {"type":"log","@timestamp":"2017-12-03T01:47:39Z","tags":["status","plugin:metrics@5.6.3","info"],"pid":1719,"state":"green","message":"Status changed from uninitialized to green - Ready","prevState":"uninitialized","prevMsg":"uninitialized"}
Dec 03 01:47:39 instance-1 kibana[1719]: {"type":"log","@timestamp":"2017-12-03T01:47:39Z","tags":["status","plugin:timelion@5.6.3","info"],"pid":1719,"state":"green","message":"Status changed from uninitialized to green - Ready","prevState":"uninitialized","prevMsg":"uninitialized"}
Dec 03 01:47:39 instance-1 kibana[1719]: {"type":"log","@timestamp":"2017-12-03T01:47:39Z","tags":["listening","info"],"pid":1719,"message":"Server running at http://0.0.0.0:601"}
Dec 03 01:47:39 instance-1 kibana[1719]: {"type":"log","@timestamp":"2017-12-03T01:47:39Z","tags":["status","ui settings","error"],"pid":1719,"state":"red","message":"Status changed from uninitialized to red - Elasticsearch plugin is red","prevState":"uninitialized","prevMsg":"uninitialized"}
Dec 03 01:47:42 instance-1 kibana[1719]: {"type":"log","@timestamp":"2017-12-03T01:47:42Z","tags":["status","plugin:elasticsearch@5.6.3","error"],"pid":1719,"state":"red","message":"Status changed from red to red - Service Unavailable","prevState":"red","prevMsg":"Unable to connect to Elasticsearch at http://localhost:9200."}
Dec 03 01:47:44 instance-1 kibana[1719]: {"type":"log","@timestamp":"2017-12-03T01:47:44Z","tags":["status","plugin:elasticsearch@5.6.3","error"],"pid":1719,"state":"red","message":"Status changed from red to red - Elasticsearch is still initializing the kibana index.","prevState":"red","prevMsg":"Service Unavailable"}
Dec 03 01:47:50 instance-1 kibana[1719]: {"type":"log","@timestamp":"2017-12-03T01:47:50Z","tags":["status","plugin:elasticsearch@5.6.3","info"],"pid":1719,"state":"green","message":"Status changed from red to green - Kibana index ready","prevState":"red","prevMsg":"Elasticsearch is still initializing the kibana index."}
Dec 03 01:47:50 instance-1 kibana[1719]: {"type":"log","@timestamp":"2017-12-03T01:47:50Z","tags":["status","ui settings","info"],"pid":1719,"state":"green","message":"Status changed from red to green - Ready","prevState":"red","prevMsg":"Elasticsearch plugin is red"}

```

## LOGSTASH:

Logstash is a tool for managing events and logs. The purpose of Logstash is to get events from any number of inputs (could be from a file, a queue, another Logstash instance, etc), apply filters (parse, modify, or perform any number of processing tasks), and finally output to any number of destinations.

## INSTALLATION OF LOGSTASH:

This setups installs for logstash in your system

```
$ sudo apt-get install apt-transport-https
```

This will establish the source for Logstash

```
$ echo "deb https://artifacts.elastic.co/packages/5.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-5.x.list
```

Setting up for Logstash installation

```
$ sudo apt-get update
```

```
$ sudo apt-get install logstash
```

Start the logstash service so we can start shipping logs

```
$ sudo service logstash start
```

```

venkateshumamaheswaran@instance-1:~$ sudo service logstash status
● logstash.service - LSB: Starts Logstash as a daemon.
   Loaded: loaded (/etc/init.d/logstash; generated; vendor preset: enabled)
   Active: active (exited) since Wed 2017-10-04 06:31:39 UTC; 2 weeks 6 days ago
     Docs: man:systemd-sysv-generator(8)
   Tasks: 0 (limit: 4915)
   CGroup: /system.slice/logstash.service

```



## SIMPLE WIKIPEDIA DATASET:

Elasticsearch only supports JSON documents. We have chosen simplewiki document to index into our Elasticsearch instance.

This is the link for Simple Wikipedia data we are using:

<https://dumps.wikimedia.org/other/cirrussearch/20171106/enwikiquote-20171106-cirrussearch-general.json.gz>

<a href="#">enwikibooks-20171106-cirrussearch-general.json.gz</a>	07-Nov-2017 17:40	100300041
<a href="#">enwikinews-20171106-cirrussearch-content.json.gz</a>	07-Nov-2017 17:41	51697701
<a href="#">enwikinews-20171106-cirrussearch-general.json.gz</a>	07-Nov-2017 17:56	366382264
<a href="#">enwikiquote-20171106-cirrussearch-content.json.gz</a>	07-Nov-2017 17:57	195231449
<a href="#">enwikiquote-20171106-cirrussearch-general.json.gz</a>	07-Nov-2017 17:58	63055360
<a href="#">enwikisource-20171106-cirrussearch-content.json.gz</a>	07-Nov-2017 18:28	4274507543
<a href="#">enwikisource-20171106-cirrussearch-general.json.gz</a>	07-Nov-2017 18:30	144400036
<a href="#">enwikiiversity-20171106-cirrussearch-content.json.gz</a>	07-Nov-2017 18:31	122816349
<a href="#">enwikiiversity-20171106-cirrussearch-general.json.gz</a>	07-Nov-2017 18:32	151778222
<a href="#">enwikivoyage-20171106-cirrussearch-content.json.gz</a>	07-Nov-2017 18:33	150830703

## LOADING WIKI DATA AND INDEXING:

Step 1: Download a wiki dump

Step 2: Get the index ready

Step 3: Prepare the wiki for loading

Step 4: Load the wiki

## INDEXING WIKI QUOTE DATA:

We need analysis-icu plugin for Elasticsearch to handle it the index.

bin/plugin install analysis-icu

Then we need jq for some of the json-foo we do next.

sudo apt-get install jq

Then we have to create 3 vim files createindex.sh, chunker.sh, uploader.sh

## CREATEINDEX.SH

```
export es=localhost:9200
export site=en.wikiquote.org
export index=enwikiquote

curl -XDELETE $es/$index?pretty

curl -s 'https://'$site'/w/api.php?action=cirrus-settings-dump&format=json&formatversion=2' |
jq '{
  analysis: .content.page.index.analysis,
  number_of_shards: 1,
  number_of_replicas: 0
}' |
curl -XPUT $es/$index?pretty -d @-

curl -s 'https://'$site'/w/api.php?action=cirrus-mapping-dump&format=json&formatversion=2' |
jq .content |
sed 's/"index_analyzer"/"analyzer"/' |
sed 's/"position_offset_gap"/"position_increment_gap"/' |
curl -XPUT $es/$index/_mapping/page?pretty -d @-
```



## CODE EXPLANATION FOR CREATEINDEX.SH:

export es=localhost:9200 sets up \$es to be Elasticsearch's address.  
export site=en.wikiquote.org sets up \$site to be the hostname of the MediaWiki instance that you want to use.  
export index=enwikiquote just sets \$index to the name of the index you'll be loading.  
curl -XDELETE \$es/\$index?pretty deletes the index if it already exists.

## CHUNKER.SH

```
export dump=enwikiquote-20171106-cirrussearch-general.json.gz
export index=enwikiquote

mkdir chunks
cd chunks
zcat ../$dump | split -a 10 -l 500 - $index
```

## CODE EXPLANATION FOR CHUNKER.SH:

The first export line just names the file that you downloaded.  
The mkdir and cd lines make a directory to hold the files.  
The last line cuts the file into 500 line chunks. 250 of those lines are metadata lines for the \_bulk api. 250 lines are the actual documents.

## UPLOADER.SH:

```
export es=localhost:9200
export index=enwikiquote
cd chunks

for file in *; do
    echo -n "${file}: "
    took=$(curl -s -XPOST $es/$index/_bulk?pretty --data-binary @$file |
        grep took | cut -d':' -f 2 | cut -d',' -f 1)
    printf '%7s\n' $took
    [ "x$took" = "x" ] || rm $file
done
```

## CODE EXPLANATION FOR UPLOADER.SH:

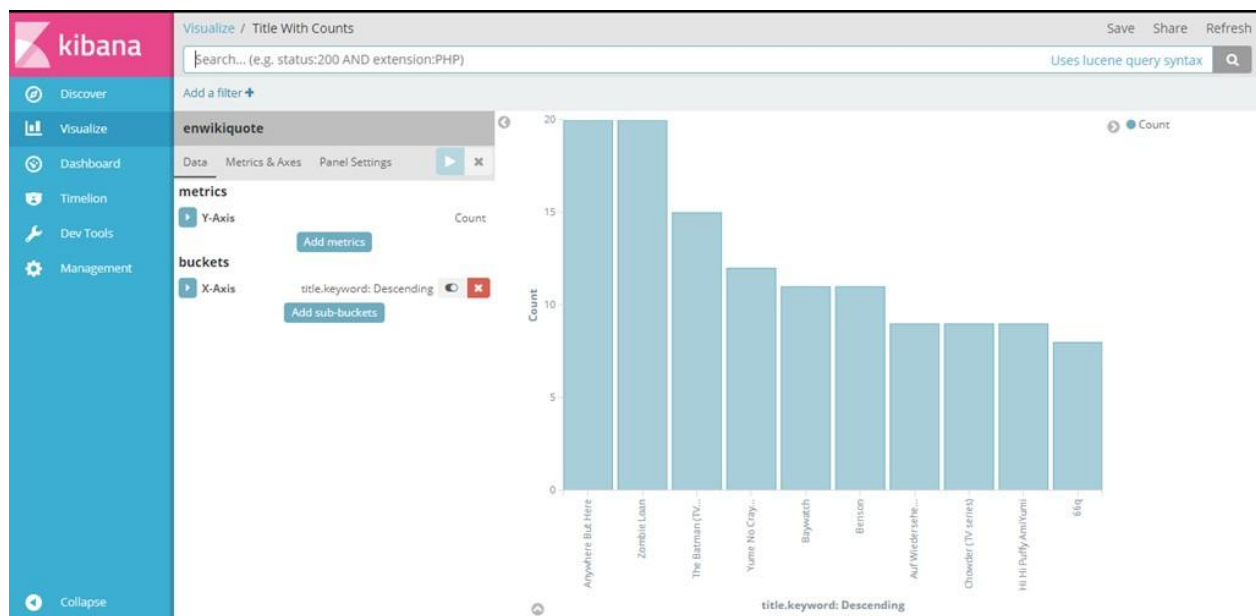
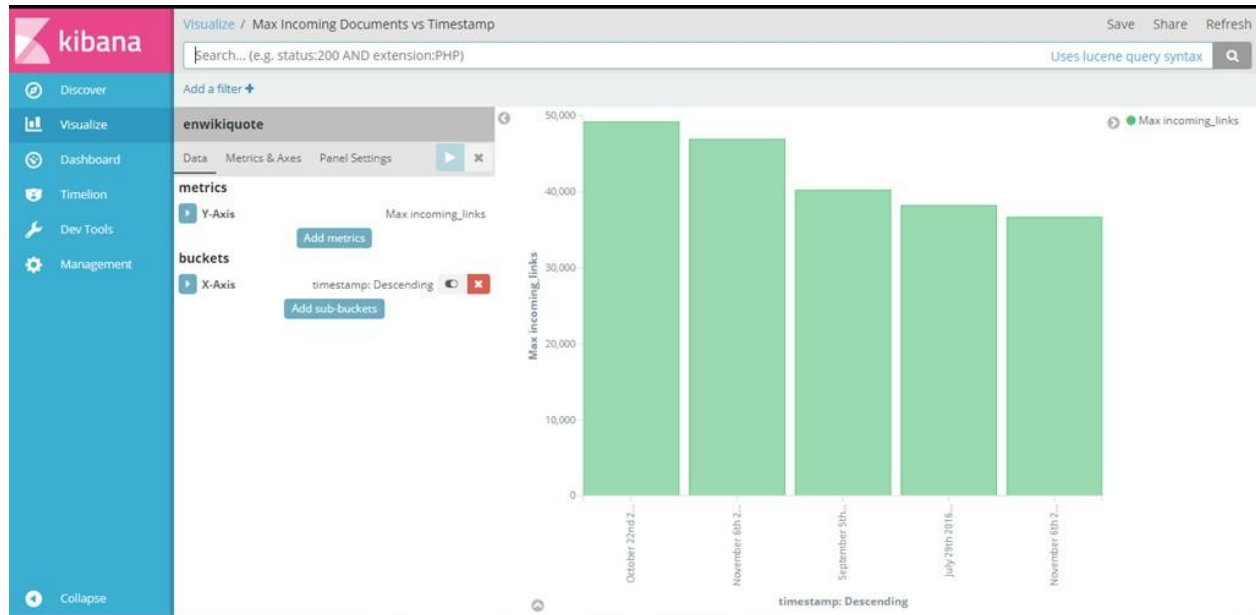
The first three lines should be familiar from above. The loop loads each file and deletes it after it's loaded.  
If the file fails to load it isn't deleted and the loop moves on to the next file.

## KIBANA VISUALIZATION:

Here are some of the example kibana visualizations we have come up with

The screenshot shows the Kibana Discover interface. The left sidebar contains navigation links: Discover, Visualize, Dashboard, Timelion, Dev Tools, and Management. The main panel displays search results for the query 'text: Rain'. The top bar shows '166 hits' and a search bar with the query 'text: Rain'. The 'Selected Fields' section lists fields like '\_id', '\_score', '\_index', '\_type', 'auxiliary\_text', 'category', 'content\_model', 'defaultsort', 'external\_link', and 'heading'. The 'Available Fields' section lists fields like '\_id', '\_score', '\_index', '\_type', 'auxiliary\_text', 'category', 'content\_model', 'defaultsort', 'external\_link', and 'heading'. The search results are displayed in a table format, showing the '\_source' field for each hit. The first hit is a quote about rain: 'Beauty for some provides escape, who gain a happiness in eyeing the gorgeous buttocks of the ape or Autumn sunsets exquisitely dying. Let the rain kiss you. Let the rain beat upon your head with silver liquid drops. Let the rain sing you a lullaby. Like a welcome summer rain, humor may suddenly cleanse and cool the earth, the air and you. Negroes - Sweet and docile, Meek, humble, and kind: Beware the day - They change their mind. We Negro writers, just by being black, have been on the blacklist all our lives ... Censorship for us begins at the color line. When peoples care for you and cry for you, they care for you.' The second hit is a quote about a rainbow: 'I caught a rainbow A beautiful rainbow And made it mine But when the rain stopped I lost my rainbow My pretty rainbow Amid the sunshine [specific citation needed] version: 2,230,511 Wiki: enwikiquote namespace: 1 namespace\_text: Talk title: Thuraya AlArrayed timestamp: March 6th 2017, 13:23:55.000 category: Pages with inadequate citations external\_link: outgoing\_link: Wikiquote:Citing\_sources template: Template:Fix cite source\_text: == Unsourced == == Dreams == :I caught a rainbow :A beautiful rainbow :And made it mine :But when the rain stopped :I lost my rainbow :My pretty rainbow :Amid the sunshine'. The third hit is a quote about rain: 'Roger Smith: "We have choices. Some people like to stand in the rain without an umbrella. That's what it means to live free." namespace: 2 namespace\_text: User title: Peter s timestamp: April 30th 2006, 15:08:56.000 category: external\_link: heading: The Big O outgoing\_link: template: text\_bytes: 143 incoming\_links: 0 redirect: auxiliary\_text: source\_text: == The Big O == Roger Smith: "We have choices. Some people like to stand in the rain without an umbrella. That's what it means to live free." opening\_text: - version\_type: external version: 212,573 defaultsort: false language: en'. The fourth hit is a quote about rain: 'I once had a leather jacket that got ruined in the rain. Why does moisture ruin leather? Aren't cows outside a lot of the time? When it's raining, do cows go up to the farmhouse, "Let us in! We're all wearing leather! Open the door! We're going to ruin the whole outfit here!" Jerry Seinfeld namespace: 2 namespace\_text: User title: Changetheworld timestamp: May 17'.

The screenshot shows the Kibana Discover interface. The left sidebar contains navigation links: Discover, Visualize, Dashboard, Timelion, Dev Tools, and Management. The main panel displays search results for the query 'namespace: 3 namespace\_text: User talk title: 68.196.37.135 timestamp: March 9th 2008, 19:43:15.000 category:'. The top bar shows '113,398 hits' and a search bar with the query 'Search... (e.g. status:200 AND extension:PHP)'. The 'Selected Fields' section lists fields like '\_id', '\_score', '\_index', '\_type', 'auxiliary\_text', 'category', 'content\_model', 'defaultsort', 'external\_link', and 'heading'. The 'Available Fields' section lists fields like '\_id', '\_score', '\_index', '\_type', 'auxiliary\_text', 'category', 'content\_model', 'defaultsort', 'external\_link', and 'heading'. The search results are displayed in a table format, showing the '\_source' field for each hit. The first hit is a quote about Wikiquote: 'namespace: 3 namespace\_text: User talk title: 68.196.37.135 timestamp: March 9th 2008, 19:43:15.000 category: external\_link: heading: outgoing\_link: User:Kalki, Wikiquote:Sandbox, Wikiquote:Vandalism, Wikiquote:What\_Wikiquote\_is\_not, Wikiquote:Wikiquote template: Template:Text2 text: Please stop adding nonsense to Wikiquote. It is considered vandalism. If you want to experiment, please use the sandbox. Wikiquote exists for the collecting of notable quotations of famous people and famous works. For a quick overview of what Wikiquote is, read Wikiquote:Wikiquote, and also what Wikiquote is not for'. The second hit is a quote about Wikiquote: 'namespace: 3 namespace\_text: User talk title: Elessar Alkna timestamp: February 24th 2007, 10:31:43.000 category: external\_link: heading: outgoing\_link: User:Cbrown1023, User\_talk:Cbrown1023, Wikiquote:About, Wikiquote:Browse, Wikiquote:Guide\_to\_layout, Wikiquote:How\_to\_edit\_a\_page, Wikiquote:Sandbox, Wikiquote:Sign\_your\_posts\_on\_talk\_pages, Wikiquote:Village\_pump, Wikiquote:Welcome\_newcomers, Wikiquote:What\_Wikiquote\_is\_not, Wikiquote:Wikiquote, Help:Edit\_summary template: text: Hi Elessar Alkna. Welcome to English Wikiquote. For a quick overview of what Wikiquote is, read Wikiquote:Wikiquote. See also what Wikiquote is not for common'. The third hit is a quote about Wikiquote: 'namespace: 3 namespace\_text: User talk title: Brilliant 420 timestamp: July 21st 2008, 11:24:30.000 category: external\_link: heading: outgoing\_link: Wikiquote:About, Wikiquote:Browse, Wikiquote:Guide\_to\_layout, Wikiquote:How\_to\_edit\_a\_page, Wikiquote:Sandbox, Wikiquote:Sign\_your\_posts\_on\_talk\_pages, Wikiquote:Village\_pump, Wikiquote:Welcome\_newcomers, Wikiquote:What\_Wikiquote\_is\_not, Wikiquote:Wikiquote, Help:Edit\_summary template: text: Hi, Brilliant 420. Welcome to English Wikiquote. For a quick overview of what Wikiquote is, read Wikiquote:Wikiquote. See also what Wikiquote is not for common'. The fourth hit is a quote about Wikiquote: 'namespace: 3 namespace\_text: User talk title: Bynick timestamp: August 12th 2008, 20:19:28.000 category: external\_link: heading: outgoing\_link: User:WelcomeBot, User\_talk:Cbrown1023, Wikiquote:About, Wikiquote:Browse, Wikiquote:Guide\_to\_layout, Wikiquote:How\_to\_edit\_a\_page, Wikiquote:Sandbox, Wikiquote:Sign\_your\_posts\_on\_talk\_pages, Wikiquote:Village\_pump, Wikiquote:Welcome\_newcomers, Wikiquote:What\_Wikiquote\_is\_not, Wikiquote:Wikiquote, Help:Edit\_summary template: text: Hi, Bynick'.



## INDEXING WIKI DUMPS On AMAZON WEB SERVICES using ELK Stack:

The same commands are to be executed for installation of Elasticsearch, Logstash and Kibana as used for Google Cloud Platform. Since the commands are the very same, you can see the screenshots of successful installation of ELK stack on AMAZON WEB SERVICES.

## Elasticsearch Installation:

```
root@ip-172-31-10-175:~# apt-get install elasticsearch
apt-get install elasticsearch
Reading package lists... Done
Building dependency tree
Reading state information... Done
elasticsearch-1.7.2-amd64.deb is already installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ip-172-31-10-175:~# dpkg-query -f='${Package} ${Version} ${Architecture} ${Source} ${Description}\n' -W elasticsearch
Package: elasticsearch
Version: 1.7.2-1
Architecture: amd64
Source: elasticsearch-1.7.2
Description: Elasticsearch is a distributed, RESTful search and analytics engine capable of solving a growing number of use cases. It is built on top of Apache Lucene and is designed to be highly available, scalable, and easy to use.
root@ip-172-31-10-175:~# dpkg-query -f='${Package} ${Version} ${Architecture} ${Source} ${Description}\n' -W elasticsearch-head
Package: elasticsearch-head
Version: 1.7.2-1
Architecture: amd64
Source: elasticsearch-1.7.2
Description: Elasticsearch head is a web-based interface for Elasticsearch. It provides a rich, interactive user interface for managing and querying Elasticsearch clusters.
root@ip-172-31-10-175:~# dpkg-query -f='${Package} ${Version} ${Architecture} ${Source} ${Description}\n' -W elasticsearch-kibana
Package: elasticsearch-kibana
Version: 1.7.2-1
Architecture: amd64
Source: elasticsearch-1.7.2
Description: Elasticsearch Kibana is a web-based interface for Elasticsearch. It provides a rich, interactive user interface for managing and querying Elasticsearch clusters.
root@ip-172-31-10-175:~# dpkg-query -f='${Package} ${Version} ${Architecture} ${Source} ${Description}\n' -W elasticsearch-plugin
Package: elasticsearch-plugin
Version: 1.7.2-1
Architecture: amd64
Source: elasticsearch-1.7.2
Description: Elasticsearch plugin is a web-based interface for Elasticsearch. It provides a rich, interactive user interface for managing and querying Elasticsearch clusters.
root@ip-172-31-10-175:~# dpkg-query -f='${Package} ${Version} ${Architecture} ${Source} ${Description}\n' -W elasticsearch-head
Package: elasticsearch-head
Version: 1.7.2-1
Architecture: amd64
Source: elasticsearch-1.7.2
Description: Elasticsearch head is a web-based interface for Elasticsearch. It provides a rich, interactive user interface for managing and querying Elasticsearch clusters.
root@ip-172-31-10-175:~# dpkg-query -f='${Package} ${Version} ${Architecture} ${Source} ${Description}\n' -W elasticsearch-kibana
Package: elasticsearch-kibana
Version: 1.7.2-1
Architecture: amd64
Source: elasticsearch-1.7.2
Description: Elasticsearch Kibana is a web-based interface for Elasticsearch. It provides a rich, interactive user interface for managing and querying Elasticsearch clusters.
root@ip-172-31-10-175:~# dpkg-query -f='${Package} ${Version} ${Architecture} ${Source} ${Description}\n' -W elasticsearch-plugin
Package: elasticsearch-plugin
Version: 1.7.2-1
Architecture: amd64
Source: elasticsearch-1.7.2
Description: Elasticsearch plugin is a web-based interface for Elasticsearch. It provides a rich, interactive user interface for managing and querying Elasticsearch clusters.
```

## Kibana Installation:

```
root@ip-172-31-10-175:~# apt-get install elasticsearch-head
apt-get install elasticsearch-head
Reading package lists... Done
Building dependency tree
Reading state information... Done
elasticsearch-head-1.7.2-amd64.deb is already installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ip-172-31-10-175:~# dpkg-query -f='${Package} ${Version} ${Architecture} ${Source} ${Description}\n' -W elasticsearch-head
Package: elasticsearch-head
Version: 1.7.2-1
Architecture: amd64
Source: elasticsearch-1.7.2
Description: Elasticsearch head is a web-based interface for Elasticsearch. It provides a rich, interactive user interface for managing and querying Elasticsearch clusters.
root@ip-172-31-10-175:~# dpkg-query -f='${Package} ${Version} ${Architecture} ${Source} ${Description}\n' -W elasticsearch-kibana
Package: elasticsearch-kibana
Version: 1.7.2-1
Architecture: amd64
Source: elasticsearch-1.7.2
Description: Elasticsearch Kibana is a web-based interface for Elasticsearch. It provides a rich, interactive user interface for managing and querying Elasticsearch clusters.
root@ip-172-31-10-175:~# dpkg-query -f='${Package} ${Version} ${Architecture} ${Source} ${Description}\n' -W elasticsearch-plugin
Package: elasticsearch-plugin
Version: 1.7.2-1
Architecture: amd64
Source: elasticsearch-1.7.2
Description: Elasticsearch plugin is a web-based interface for Elasticsearch. It provides a rich, interactive user interface for managing and querying Elasticsearch clusters.
```

```

root@ip-172-31-10-175:~#
No packages marked for update
root@ip-172-31-10-175:~# elasticsearch-1.0.0 /root
root@ip-172-31-10-175:~# wget https://download.elastic.co/logstash/packages/ubuntu/logstash-1.5.4-1.noarch.rpm
--2017-10-04 05:27:30-- https://download.elastic.co/logstash/packages/ubuntu/logstash-1.5.4-1.noarch.rpm
Resolving download.elastic.co (download.elastic.co)... 184.72.218.74, 184.79.136.41, 72.21.119.41, ...
Connecting to download.elastic.co (download.elastic.co|184.72.218.74|:443)... connected.
HTTP request sent, awaiting response... 200 OK
Length: 82738404 (79M) [application/octet-stream]
Saving to: 'logstash-1.5.4-1.noarch.rpm'

logstash-1.5.4-1.noarch.rpm                               100%[=====] 80.40M  44.90kB/s   in 2.0s

2017-10-04 05:27:32 (46.9 MB/s) - 'logstash-1.5.4-1.noarch.rpm' saved [82738404/82738404]

root@ip-172-31-10-175:~# yum install logstash-1.5.4-1.noarch.rpm -y
Loaded plugins: priorities, update-helper, upgrade-helper
Examining logstash-1.5.4-1.noarch.rpm: i:logstash-1.5.4-1.noarch
Running transaction check
Dependencies resolved.
Package logstash-1.5.4-1.noarch.rpm to be installed
Running transaction check
--> Package logstash.noarch 1:1.5.4-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====================================================================================================================================
 Package Arch Version Repository Size
=====================================================================================================================================
Installing:
 logstash noarch 1:1.5.4-1 /logstash-1.5.4-1.noarch 136 M
=====================================================================================================================================

Transaction Summary
=====================================================================================================================================
Install 1 Package

Total size: 136 M
Installed size: 336 M
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : 1:logstash-1.5.4-1.noarch
Verifying : 1:logstash-1.5.4-1.noarch

Installed:
 logstash.noarch 1:1.5.4-1

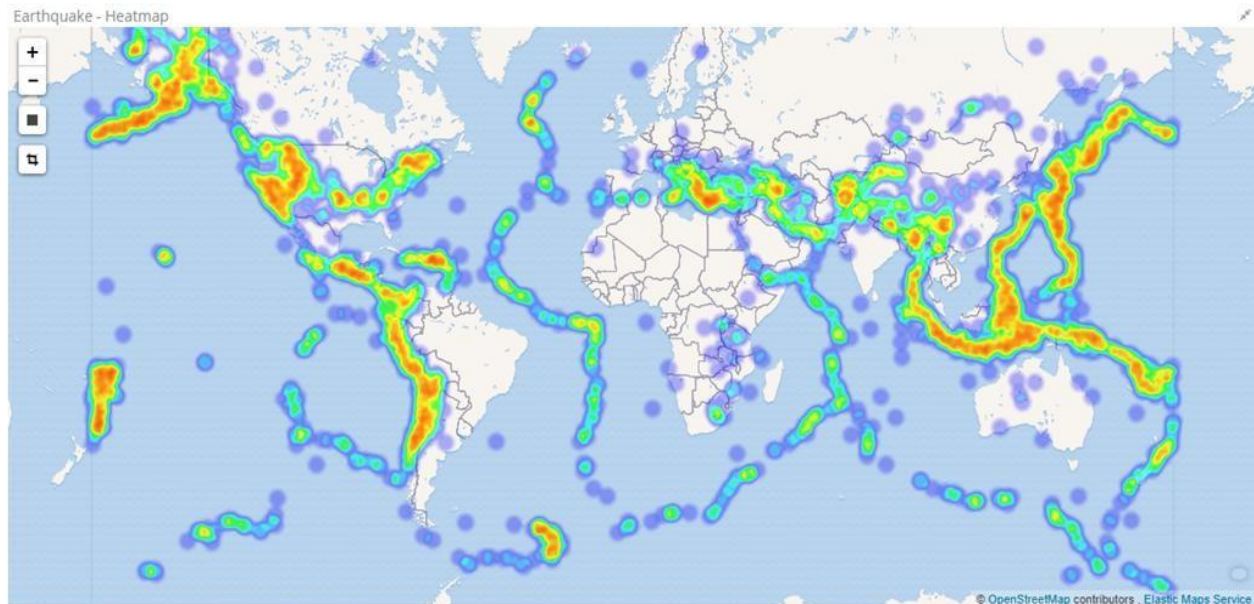
Complete!
root@ip-172-31-10-175:~#

```

The screenshot shows the Kibana search results page. The top navigation bar includes the Kibana logo and a search bar with the query `35.196.79.239:5601/app/kibana/#/discover?_g=0&a=(columns:(L_source).index:AV-Z-01b35HcaJuf0KmlInterval:auto.query:(match_all:{}).sort:(_score.desc))`. The left sidebar contains navigation links for Discover, Visualize, Dashboard, Timeline, Dev Tools, and Management. The main content area displays search results for the index `enwikiquote`. The results are sorted by `_score` in descending order. The first result is for the document `68.196.37.135`, which is a vandalism notice. The second result is for the document `Elessar Alkna`, which is a welcome message. The third result is for the document `Brilliant 420`, which is a welcome message. The fourth result is for the document `Bynick`, which is a welcome message. The interface includes a 'Discover' tab, a 'Visualize' tab, and a 'Dashboard' tab. The 'Discover' tab is active, showing a table of search results with columns for `_source`, `_type`, `_score`, `external_link`, `heading`, `outgoing_link`, `text`, `template`, and `incoming_links`. The search results are displayed in a list format, with each result showing the document's metadata and the full text of the quote.

For more sample visualizations and better hands- on experience on Elasticsearch and Kibana, we tried this data available on github  
<https://github.com/elastic/examples/tree/master/Exploring%20Public%20Datasets/earthquakes>





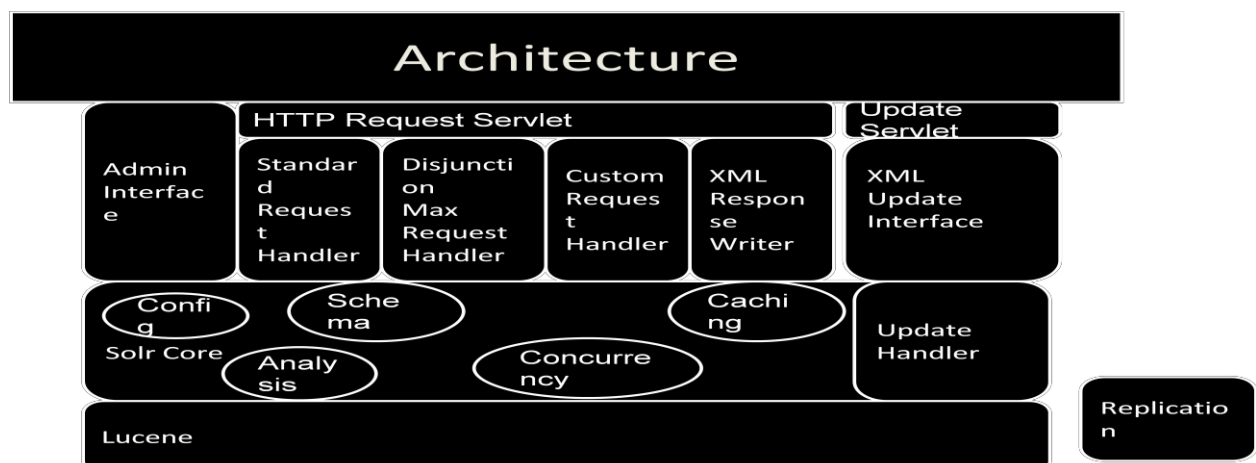
## INDEXING WIKI DUMPS ON MICROSOFT AZURE USING SOLR:

### SOLR:

Solr is powered by Lucene, a powerful open-source full-text search library, under the hood. Solr is designed for scalability and fault tolerance. Solr is widely used for enterprise search and analytics use cases.

- XML/HTTP Interfaces
- Loose Schema to define types and fields
- Web Administration Interface
- Extensive Caching
- Index Replication
- Extensible Open Architecture

### ARCHITECTURE OF SOLR:





## INSTALLATION STEPS OF SOLR:

Below are the steps for the installation steps of SOLR clearly explained step by step:

1. Download solr-3.4
2. Download wikipedia dump
3. data-config.xml was used to index Wikipedia dump.

```
<dataConfig>
<dataSource type="FileDataSource" encoding="UTF-8" />
<document>
<entity name="page"
processor="XPathEntityProcessor"
stream="true"
forEach="/mediawiki/page/"
```

4. The relevant portion of schema.xml is below:

```
<field name="id" type="string" indexed="true" stored="true" required="true"/>
<field name="title" type="string" indexed="true" stored="false"/>
<field name="revision" type="sint" indexed="true" stored="true"/>
<field name="user" type="string" indexed="true" stored="true"/>
<field name="userId" type="int" indexed="true" stored="true"/>
<field name="text" type="text" indexed="true" stored="false"/>
<uniqueKey>id</uniqueKey>
<copyField source="title" dest="titleText"/>
```

5. Add Dih request handler in solrconfig.xml file

```
<requestHandler name="/update/dih" startup="lazy">
<lst name="defaults">
<str name="config">dih-config.xml</str>
</lst>
```

6. Restart solr

7. Index some documents using below command\_

<http://localhost:8983/solr/update/dih?command=full-import>

## Schema: Analyzers:

```
<fieldtype name="nametext" class="solr.TextField">
<analyzer class="org.apache.lucene.analysis.WhitespaceAnalyzer"/>
</fieldtype>
<fieldtype name="text" class="solr.TextField">
<analyzer>
<tokenizer class="solr.StandardTokenizerFactory"/>
<filter class="solr.StandardFilterFactory"/>
<filter class="solr.LowerCaseFilterFactory"/>
<filter class="solr.StopFilterFactory"/>
<filter class="solr.PorterStemFilterFactory"/>
</analyzer>
```

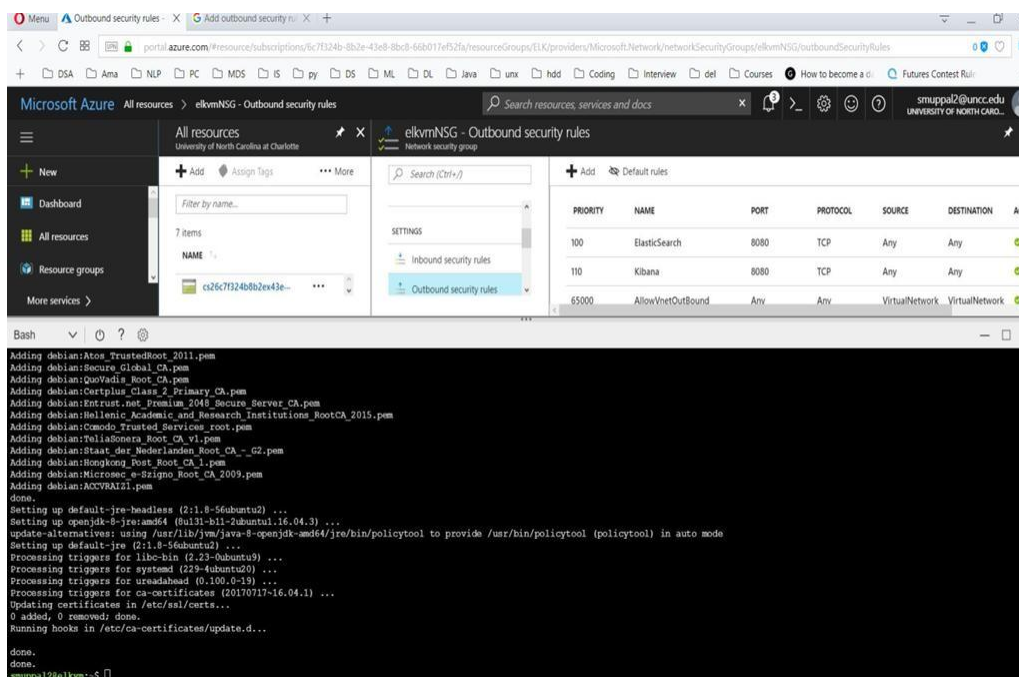
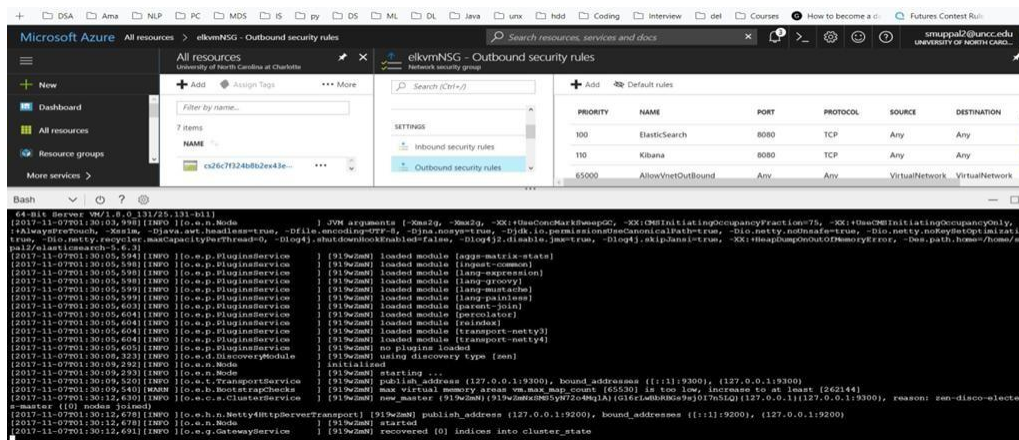
```

</fieldtype>
<fieldtype name="myfieldtype" class="solr.TextField">
  <analyzer>
    <tokenizer class="solr.WhitespaceTokenizerFactory"/>
    <filter class="solr.SnowballPorterFilterFactory" language="German" />
  </analyzer>
</fieldtype>

```

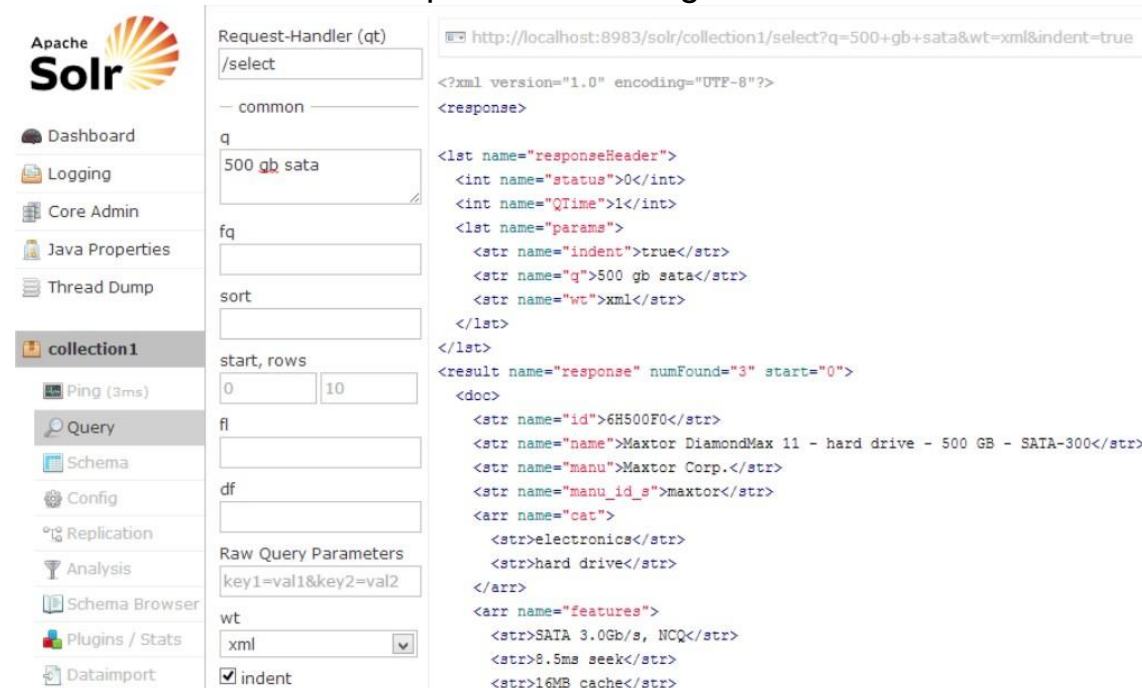
## INSTALLATION OF SOLR IN AZURE:

Below are the screenshots for installation of SOLR using AZURE



## INDEXED WIKIPEDIA DATA IN SOLR:

Below is the indexed wikipedia data using SOLR



The screenshot displays the Apache Solr Admin interface. On the left is a sidebar with navigation links: Dashboard, Logging, Core Admin, Java Properties, Thread Dump, collection1 (selected), Ping (3ms), Query, Schema, Config, Replication, Analysis, Schema Browser, Plugins / Stats, and Dataimport. The main area is titled 'Request-Handler (qt)' and shows the '/select' handler. The 'q' (query) field contains '500 gb sata'. The 'wt' (output format) is set to 'xml' and the 'indent' checkbox is checked. The 'Response' field displays the XML output of the query, which includes a status of 0, a QTime of 1, and a list of 3 results. The first result is a document with fields: id (6H500F0), name (Maxtor DiamondMax 11 - hard drive - 500 GB - SATA-300), manu (Maxtor Corp.), manu\_id\_s (maxtor), cat (electronics, hard drive), and features (SATA 3.0Gb/s, NCQ, 8.5ms seek, 16MB cache).

```
http://localhost:8983/solr/collection1/select?q=500+gb+sata&wt=xml&indent=true

<?xml version="1.0" encoding="UTF-8"?>
<response>

  <lst name="responseHeader">
    <int name="status">0</int>
    <int name="QTime">1</int>
    <lst name="params">
      <str name="indent">true</str>
      <str name="q">500 gb sata</str>
      <str name="wt">xml</str>
    </lst>
  </lst>

  <result name="response" numFound="3" start="0">
    <doc>
      <str name="id">6H500F0</str>
      <str name="name">Maxtor DiamondMax 11 - hard drive - 500 GB - SATA-300</str>
      <str name="manu">Maxtor Corp.</str>
      <str name="manu_id_s">maxtor</str>
      <arr name="cat">
        <str>electronics</str>
        <str>hard drive</str>
      </arr>
      <arr name="features">
        <str>SATA 3.0Gb/s, NCQ</str>
        <str>8.5ms seek</str>
        <str>16MB cache</str>
      </arr>
    </doc>
  </result>
</response>
```

## CREDITS:

<https://logz.io/blog/elk-stack-google-cloud/>

<https://dzone.com/articles/how-to-install-the-elk-stack-on-google-cloud-platf-1>

<https://www.elastic.co/blog/loading-wikipedia>

<https://dumps.wikimedia.org/other/cirrussearch/current/>

<https://www.digitaiocean.com/community/tutorials/how-to-use-kibana-dashboards-and-visualizations>

<https://github.com/elastic/examples/tree/master/Exploring%20Public%20Datasets/earthquakes>

<https://stackoverflow.com/questions/3846793/running-solr-on-azure>