



Apache

Solr

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

Solr is a scalable, ready to deploy, search/storage engine optimized to search large volumes of text-centric data. Solr is enterprise-ready, fast and highly scalable. In this tutorial, we are going to learn the basics of Solr and how you can use it in practice.

Audience

This tutorial will be helpful for all those developers who would like to understand the basic functionalities of Apache Solr in order to develop sophisticated and high-performing applications.

Prerequisites

Before proceeding with this tutorial, we expect that the reader has good Java programming skills (although it is not mandatory) and some prior exposure to Lucene and Hadoop environment.

Copyright & Disclaimer

© Copyright 2016 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial	i
Audience.....	i
Prerequisites.....	i
Copyright & Disclaimer	i
Table of Contents	ii
1. Solr – Overview.....	1
Features of Apache Solr.....	1
Lucene in Search Applications	2
2. Solr – Search Engine Basics	3
Search Engine Components.....	3
How do Search Engines Work?.....	4
3. Solr – Set Up Solr on Windows.....	6
Setting Java Environment	7
4. Solr – Set Up Solr on Hadoop	8
Downloading Hadoop.....	8
Installing Hadoop.....	10
Verifying Hadoop Installation	12
Installing Solr on Hadoop	14
5. Solr – Architecture	17
Solr Architecture – Building Blocks.....	17
6. Solr – Terminology	19
General Terminology	19
SolrCloud Terminology	19
Configuration Files.....	20
7. Solr – Basic Commands	21
Starting Solr	21
Stopping Solr	22
Restarting Solr	22
Solr – help Command	23
Solr – status Command.....	23
Solr Admin	24
8. Solr – Core	25
Creating a Core	25
Deleting a Core	28
9. Solr – Indexing Data	30
Indexing in Apache Solr	30
Adding Documents using Post Command	30
Adding Documents using the Solr Web Interface	34
Adding Documents using Java Client API.....	37

10. Solr – Adding Documents (XML)	39
Adding Documents Using XML	39
11. Solr – Updating Data	43
Updating the Document Using XML	43
Updating the Document Using Java (Client API)	44
12. Solr – Deleting Documents	46
Deleting the Document	46
Deleting a Field	47
Deleting All Documents	49
13. Solr – Retrieving Data	52
14. Solr – Querying Data	54
Retrieving the Records	56
Restricting the Number of Records	59
Response Writer Type	60
List of the Fields.....	61
15. Solr – Faceting	62
Faceting Query Example	62
Faceting Using Java Client API	65

1. Solr – Overview

Solr is an open-source search platform which is used to build **search applications**. It was built on top of **Lucene** (full text search engine). Solr is enterprise-ready, fast and highly scalable. The applications built using Solr are sophisticated and deliver high performance.

It was **Yonik Seely** who created Solr in 2004 in order to add search capabilities to the company website of CNET Networks. In Jan 2006, it was made an open-source project under Apache Software Foundation. Its latest version, Solr 6.0, was released in 2016 with support for execution of parallel SQL queries.

Solr can be used along with Hadoop. As Hadoop handles a large amount of data, Solr helps us in finding the required information from such a large source. Not only search, Solr can also be used for storage purpose. Like other NoSQL databases, it is a **non-relational data storage and processing technology**.

In short, Solr is a scalable, ready to deploy, search/storage engine optimized to search large volumes of text-centric data.

Features of Apache Solr

Solr is a wrap around Lucene's Java API. Therefore, using Solr, you can leverage all the features of Lucene. Let us take a look at some of most prominent features of Solr:

- **Restful APIs:** To communicate with Solr, it is not mandatory to have Java programming skills. Instead you can use restful services to communicate with it. We enter documents in Solr in file formats like XML, JSON and .CSV and get results in the same file formats.
- **Full text search:** Solr provides all the capabilities needed for a full text search such as tokens, phrases, spell check, wildcard, and auto-complete.
- **Enterprise ready:** According to the need of the organization, Solr can be deployed in any kind of systems (big or small) such as standalone, distributed, cloud, etc.
- **Flexible and Extensible:** By extending the Java classes and configuring accordingly, we can customize the components of Solr easily.
- **NoSQL database:** Solr can also be used as big data scale NOSQL database where we can distribute the search tasks along a cluster.
- **Admin Interface:** Solr provides an easy-to-use, user friendly, feature powered, user interface, using which we can perform all the possible tasks such as manage logs, add, delete, update and search documents.

- **Highly Scalable:** While using Solr with Hadoop, we can scale its capacity by adding replicas.
- **Text-Centric and Sorted by Relevance:** Solr is mostly used to search text documents and the results are delivered according to the relevance with the user's query in order.

Unlike Lucene, you don't need to have Java programming skills while working with Apache Solr. It provides a wonderful ready-to-deploy service to build a search box featuring auto-complete, which Lucene doesn't provide. Using Solr, we can scale, distribute, and manage index, for large scale (Big Data) applications.

Lucene in Search Applications

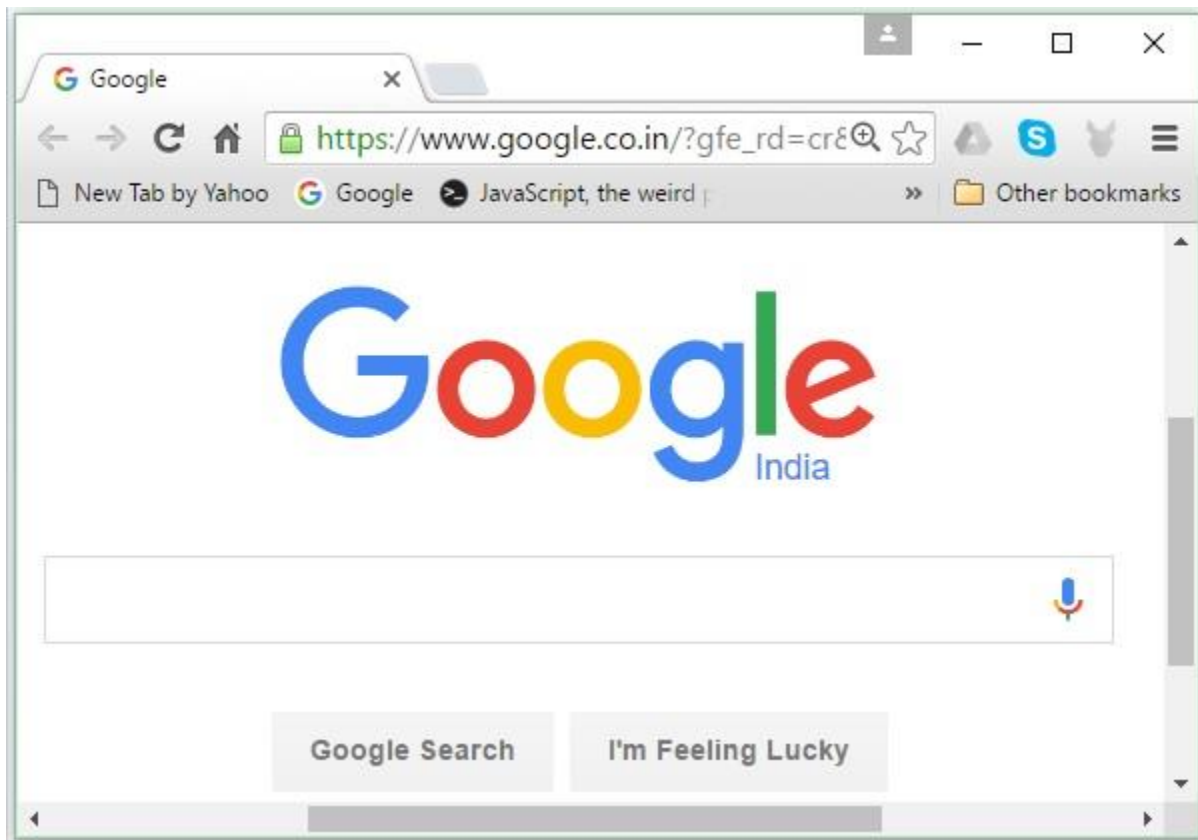
Lucene is simple yet powerful Java-based search library. It can be used in any application to add search capability. Lucene is a scalable and high-performance library used to index and search virtually any kind of text. Lucene library provides the core operations which are required by any search application, such as **Indexing** and **Searching**.

If we have a web portal with a huge volume of data, then we will most probably require a search engine in our portal to extract relevant information from the huge pool of data. Lucene works as the heart of any search application and provides the vital operations pertaining to indexing and searching.

2. Solr – Search Engine Basics

A Search Engine refers to a huge database of Internet resources such as webpages, newsgroups, programs, images, etc. It helps to locate information on the World Wide Web.

Users can search for information by passing queries into the Search Engine in the form of keywords or phrases. The Search Engine then searches in its database and returns relevant links to the user.



Search Engine Components

Generally, there are three basic components of a search engine as listed below:

- **Web Crawler** – Web crawlers are also known as **spiders** or **bots**. It is a software component that traverses the web to gather information.
- **Database** – All the information on the Web is stored in databases. They contain a huge volume of web resources.

- **Search Interfaces** – This component is an interface between the user and the database. It helps the user to search through the database.

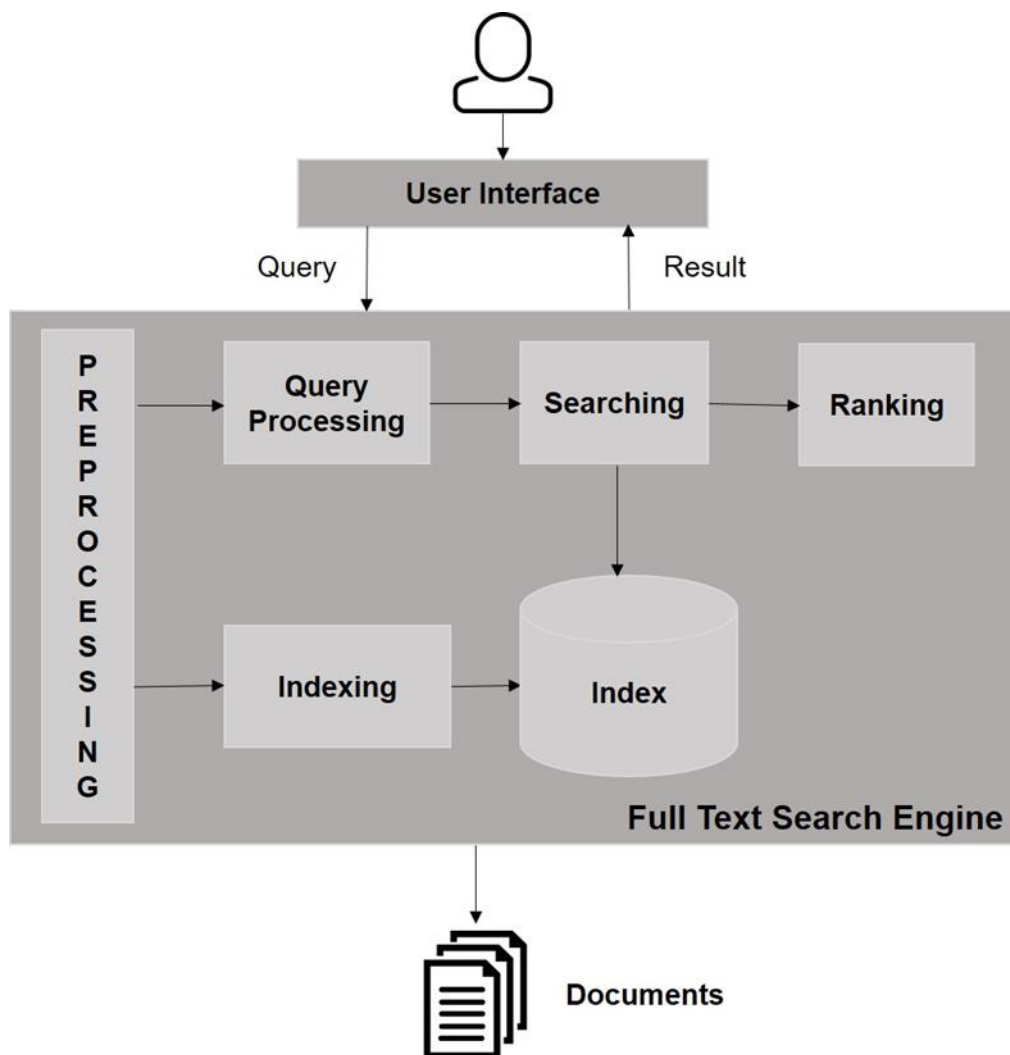
How do Search Engines Work?

Any search application is required to perform some or all of the following operations.

Step	Title	Description
1	Acquire Raw Content	The very first step of any search application is to collect the target contents on which search is to be conducted.
2	Build the document	The next step is to build the document(s) from the raw contents which the search application can understand and interpret easily.
3	Analyze the document	Before indexing can start, the document is to be analyzed.
4	Indexing the document	Once the documents are built and analyzed, the next step is to index them so that this document can be retrieved based on certain keys, instead of the whole contents of the document. Indexing is similar to the indexes that we have at the end of a book where common words are shown with their page numbers so that these words can be tracked quickly, instead of searching the complete book.
5	User Interface for Search	Once a database of indexes is ready, then the application can perform search operations. To help the user make a search, the application must provide a user interface where the user can enter text and initiate the search process.
6	Build Query	Once the user makes a request to search a text, the application should prepare a query object using that text, which can then be used to inquire the index database to get relevant details.

7	Search Query	Using the query object, the index database is checked to get the relevant details and the content documents.
8	Render Results	Once the required result is received, the application should decide how to display the results to the user using its User Interface.

Take a look at the following illustration. It shows an overall view of how Search Engines function.



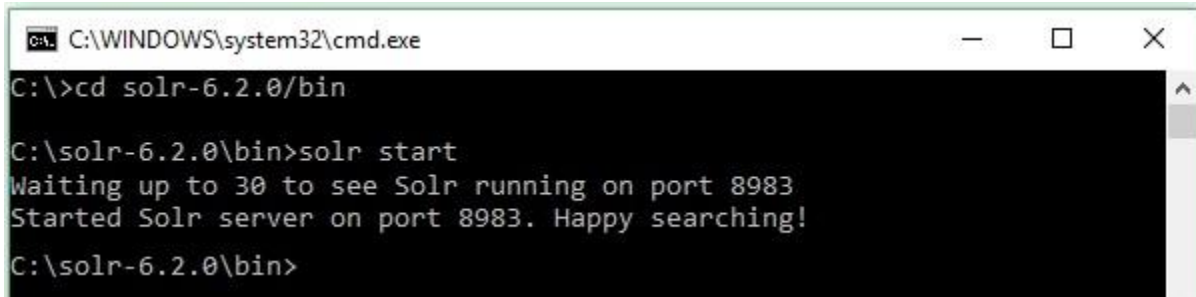
Apart from these basic operations, search applications can also provide administration-user interface to help the administrators control the level of search based on the user profiles. Analytics of search result is another important and advanced aspect of any search application.

3. Solr – Set Up Solr on Windows

In this chapter, we will discuss how to set up Solr in Windows environment. To install Solr on your Windows system, you need to follow the steps given below:

- Visit the homepage of Apache Solr and click the download button.
- Select one of the mirrors to get an index of Apache Solr. From there download the file named **Solr-6.2.0.zip**.
- Move the file from the **downloads folder** to the required directory and unzip it.

Suppose you downloaded the Solr file and extracted it in onto the C drive. In such case, you can start Solr as shown in the following screenshot.

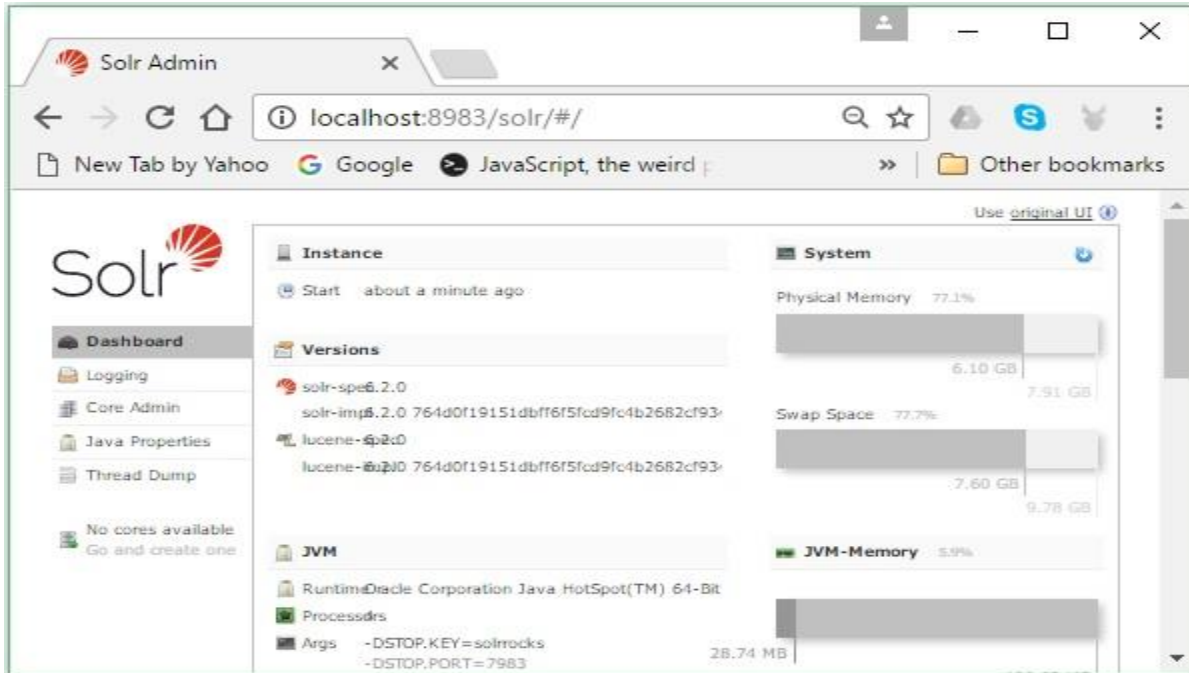


```
C:\WINDOWS\system32\cmd.exe
C:\>cd solr-6.2.0\bin
C:\solr-6.2.0\bin>solr start
Waiting up to 30 to see Solr running on port 8983
Started Solr server on port 8983. Happy searching!
C:\solr-6.2.0\bin>
```

To verify the installation, use the following URL in your browser.

<http://localhost:8983/>

If the installation process is successful, then you will get to see the dashboard of the Apache Solr user interface as shown below.



Setting Java Environment

We can also communicate with Apache Solr using Java libraries; but before accessing Solr using Java API, you need to set the classpath for those libraries.

Setting the Classpath

Set the **classpath** to Solr libraries in the **.bashrc** file. Open **.bashrc** in any of the editors as shown below.

```
$ gedit ~/.bashrc
```

Set classpath for Solr libraries (**lib** folder in HBase) as shown below.

```
export CLASSPATH = $CLASSPATH://home/hadoop/Solr/lib/*
```

This is to prevent the "class not found" exception while accessing the HBase using Java API.

4. Solr – Set Up Solr on Hadoop

Solr can be used along with Hadoop. As Hadoop handles a large amount of data, Solr helps us in finding the required information from such a large source. In this section, let us understand how you can install Hadoop on your system.

Downloading Hadoop

Given below are the steps to be followed to download Hadoop onto your system.

Step 1: Go to the homepage of Hadoop. You can use the link: <http://hadoop.apache.org/>. Click the link **Releases**, as highlighted in the following screenshot.



It will redirect you to the **Apache Hadoop Releases** page which contains links for mirrors of source and binary files of various versions of Hadoop as follows-

Version	Release Date	Tarball	GPG	SHA-256
3.0.0-alpha1	03 September, 2016	source binary	signature signature	checksum file checksum file
2.7.3	25 August, 2016	source binary	signature signature	227785DC 6E3E6EF8.. D489DF38 08244890..
2.6.4	11 February, 2016	source binary	signature signature	F755D961 18316335.. C58F08D2 E0B13035..
2.5.2	19 Nov, 2014	source binary	signature signature	139EF872 09C5637E.. 0BDB4850 A3825208..

Step 2: Select the latest version of Hadoop (in our tutorial, it is 2.6.4) and click its **binary link**. It will take you to a page where mirrors for Hadoop binary are available. Click one of these mirrors to download Hadoop.

Download Hadoop from Command Prompt

Open Linux terminal and login as super-user.

```
$ su
password:
```

Go to the directory where you need to install Hadoop, and save the file there using the link copied earlier, as shown in the following code block.

```
# cd /usr/local
# wget http://redrockdigimark.com/apachemirror/hadoop/common/hadoop-2.6.4/hadoop-2.6.4.tar.gz
```

After downloading Hadoop, extract it using the following commands.

```
# tar zxvf hadoop-2.6.4.tar.gz
# mkdir hadoop
# mv hadoop-2.6.4/* to hadoop/
# exit
```

Installing Hadoop

Follow the steps given below to install **Hadoop** in pseudo-distributed mode.

Step 1: Setting Up Hadoop

You can set the Hadoop environment variables by appending the following commands to `~/.bashrc` file.

```
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_INSTALL=$HADOOP_HOME
```

Next, apply all the changes into the current running system.

```
$ source ~/.bashrc
```

Step 2: Hadoop Configuration

You can find all the Hadoop configuration files in the location "`$HADOOP_HOME/etc/hadoop`". It is required to make changes in those configuration files according to your Hadoop infrastructure.

```
$ cd $HADOOP_HOME/etc/hadoop
```

In order to develop Hadoop programs in Java, you have to reset the Java environment variables in `hadoop-env.sh` file by replacing **JAVA_HOME** value with the location of Java in your system.

```
export JAVA_HOME=/usr/local/jdk1.7.0_71
```

The following are the list of files that you have to edit to configure Hadoop:

- core-site.xml
- hdfs-site.xml
- yarn-site.xml
- mapred-site.xml

core-site.xml

The **core-site.xml** file contains information such as the port number used for Hadoop instance, memory allocated for the file system, memory limit for storing the data, and size of Read/Write buffers.

Open the core-site.xml and add the following properties inside the <configuration>, </configuration> tags.

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

hdfs-site.xml

The **hdfs-site.xml** file contains information such as the value of replication data, **namenode** path, and **datanode** paths of your local file systems. It means the place where you want to store the Hadoop infrastructure.

Let us assume the following data.

```
dfs.replication (data replication value) = 1
```

(In the below given path /hadoop/ is the user name.

hadoopinfra/hdfs/namenode is the directory created by hdfs file system.)

```
namenode path = //home/hadoop/hadoopinfra/hdfs/namenode
```

(hadoopinfra/hdfs/datanode is the directory created by hdfs file system.)

```
datanode path = //home/hadoop/hadoopinfra/hdfs/datanode
```

Open this file and add the following properties inside the <configuration>, </configuration> tags.

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>

  <property>
    <name>dfs.name.dir</name>
```



```

    <value>file:///home/hadoop/hadoopinfra/hdfs/namenode</value>
  </property>

  <property>
    <name>dfs.data.dir</name>
    <value>file:///home/hadoop/hadoopinfra/hdfs/datanode</value>
  </property>
</configuration>

```

Note: In the above file, all the property values are user-defined and you can make changes according to your Hadoop infrastructure.

yarn-site.xml

This file is used to configure yarn into Hadoop. Open the yarn-site.xml file and add the following properties in between the <configuration>, </configuration> tags in this file.

```

<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>

```

mapred-site.xml

This file is used to specify which MapReduce framework we are using. By default, Hadoop contains a template of yarn-site.xml. First of all, it is required to copy the file from **mapred-site.xml.template** to **mapred-site.xml** file using the following command.

```
$ cp mapred-site.xml.template mapred-site.xml
```

Open **mapred-site.xml** file and add the following properties inside the <configuration>, </configuration> tags.

```

<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>

```

Verifying Hadoop Installation

The following steps are used to verify the Hadoop installation.

Step 1: Name Node Setup

Set up the namenode using the command "hdfs namenode -format" as follows.

```
$ cd ~
$ hdfs namenode -format
```

The expected result is as follows.

```
10/24/14 21:30:55 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:  host = localhost/192.168.1.11
STARTUP_MSG:  args = [-format]
STARTUP_MSG:  version = 2.6.4

...
...
10/24/14 21:30:56 INFO common.Storage: Storage directory
/home/hadoop/hadoopinfra/hdfs/namenode has been successfully formatted.
10/24/14 21:30:56 INFO namenode.NNStorageRetentionManager: Going to retain 1
images with txid >= 0
10/24/14 21:30:56 INFO util.ExitUtil: Exiting with status 0
10/24/14 21:30:56 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at localhost/192.168.1.11
*****/
```

Step 2: Verifying the Hadoop dfs

The following command is used to start the Hadoop dfs. Executing this command will start your Hadoop file system.

```
$ start-dfs.sh
```

The expected output is as follows:

```
10/24/14 21:37:56
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/hadoop/hadoop-2.6.4/logs/hadoop-
hadoop-namenode-localhost.out
localhost: starting datanode, logging to /home/hadoop/hadoop-2.6.4/logs/hadoop-
```

```
hadoop-datanode-localhost.out  
Starting secondary namenodes [0.0.0.0]
```

Step 3: Verifying the Yarn Script

The following command is used to start the Yarn script. Executing this command will start your Yarn demons.

```
$ start-yarn.sh
```

The expected output as follows:

```
starting yarn daemons  
starting resourcemanager, logging to /home/hadoop/hadoop-2.6.4/logs/yarn-hadoop-  
resourcemanager-localhost.out  
localhost: starting nodemanager, logging to /home/hadoop/hadoop-2.6.4/logs/yarn-  
hadoop-nodemanager-localhost.out
```

Step 4: Accessing Hadoop on Browser

The default port number to access Hadoop is 50070. Use the following URL to get Hadoop services on browser.

```
http://localhost:50070/
```

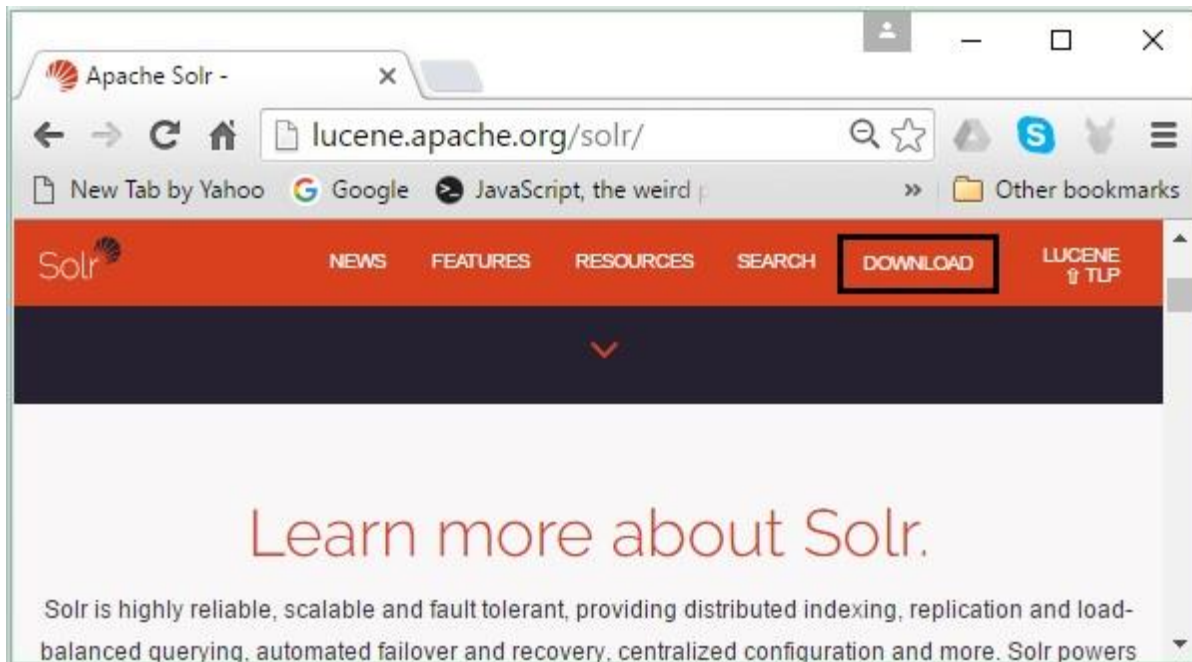
Started:	Mon Sep 19 10:54:50 IST 2016
Version:	2.6.4, r5082c73637530b0b7e115f9625ed7fac69f937e6
Compiled:	2016-02-12T09:45Z by jenkins from (detached from 5082c73)
Cluster ID:	CID-2e1fc039-ebd3-45d0-81bb-96ccc4ec0d3c
Block Pool ID:	BP-1803105006-127.0.0.1-1420541953245

Installing Solr on Hadoop

Follow the steps given below to download and install Solr.

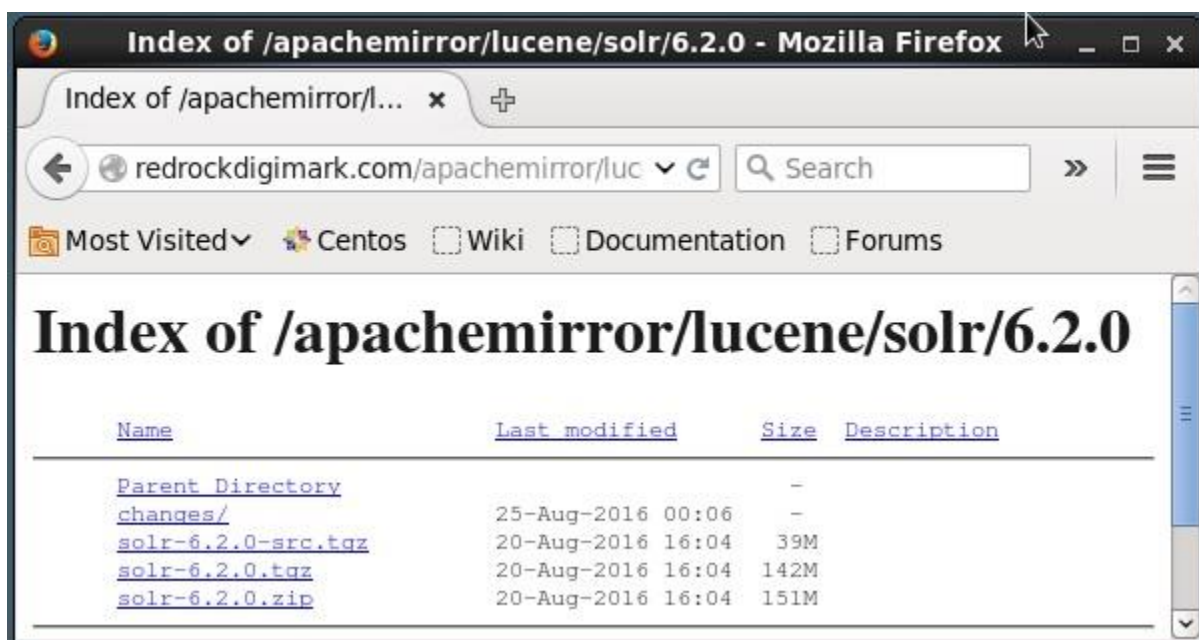
Step 1

Open the homepage of Apache Solr by clicking the following link - <http://lucene.apache.org/Solr/>



Step 2

Click the **download button** (highlighted in the above screenshot). On clicking, you will be redirected to the page where you have various mirrors of Apache Solr. Select a mirror and click on it, which will redirect you to a page where you can download the source and binary files of Apache Solr, as shown in the following screenshot.



Step 3

On clicking, a folder named **Solr-6.2.0.tqz** will be downloaded in the downloads folder of your system. Extract the contents of the downloaded folder.

Step 4

Create a folder named Solr in the Hadoop home directory and move the contents of the extracted folder to it, as shown below.

```
$ mkdir Solr
$ cd Downloads
$ mv Solr-6.2.0 /home/Hadoop/
```

Verification

Browse through the **bin** folder of Solr Home directory and verify the installation using the **version** option, as shown in the following code block.

```
$ cd bin/
$ ./Solr version
6.2.0
```

Setting home and path

Open the **.bashrc** file using the following command:

```
[Hadoop@localhost ~]$ source ~/.bashrc
```

Now set the home and path directories for Apache Solr as follows:

```
export SOLR_HOME=/home/Hadoop/Solr
export PATH=$PATH:/$SOLR_HOME/bin/
```

Open the terminal and execute the following command:

```
[Hadoop@localhost Solr]$ source ~/.bashrc
```

Now, you can execute the commands of Solr from any directory.

End of ebook preview
If you liked what you saw...
Buy it from our store @ <https://store.tutorialspoint.com>