Cloudera Runtime 7.2.18

Cloudera Search Tutorial

Date published: 2019-11-19
Date modified:

https://docs.cloudera.com/

Legal Notice

Contents

Cloudera Search tutorial

This tutorial introduces you to Cloudera Search and demonstrates some of its basic capabilities to help you become familiar with the concepts and components involved in Search. It demonstrates creating a simple test collection to validate your Cloudera Search installation, and then continues on to more advanced use cases of batch indexing.

The topics in this tutorial assume you have deployed Cloudera Search. The examples in this tutorial use two shards, so make sure that your deployment includes at least two Solr servers.

This tutorial uses modified schema.xml and solrconfig.xml configuration files. In the versions of these files included with the tutorial, unused fields have been removed for simplicity. Original versions of these files include many additional options. For information on all available options, see the Apache Solr wiki:

- SchemaXml
- SolrConfigXml

Validating the Cloudera Search deployment

After installing and deploying Cloudera Search, validate the deployment by indexing and querying sample documents.

Important: This tutorial does not account for Apache Ranger authorization. If you are using Ranger authorization, you must create policies to allow the users and actions described in the tutorial.

You can think of this as a type of "Hello, World!" for Cloudera Search to make sure that everything is installed and working properly.

Before beginning this process, make sure you have access to the Apache Solr admin web console. If your cluster is Kerberos-enabled, make sure you have access to the solr@EXAMPLE.COM Kerberos principal (where EXAMPLE.COM is your Kerberos realm name).

Create a test collection

Generate configuration files and upload the generated configuration to ZooKeeper, so that you can create a collection where you can index sample data.

Procedure

1. Make sure that the SOLR_ZK_ENSEMBLE environment variable is set in /etc/solr/conf/solr-env.sh.
   For example:

   ```
   cat /etc/solr/conf/solr-env.sh
   ```

   ```
   export SOLR_ZK_ENSEMBLE=zk01.example.com:2181,zk02.example.com:2181,zk03
   .example.com:2181/solr
   ```

   This is automatically set on hosts with a Solr Server or Gateway role in Cloudera Manager.
2. If you are using Kerberos, kinit as the user that has privileges to create the collection.
   For example:

   ```
   kinit solr@EXAMPLE.COM
   ```

   Replace *solr@EXAMPLE.COM* with your user name and Kerberos realm name respectively.

3. Generate configuration files for the collection:

```
solrctl instancedir --generate $HOME/test_collection_config
```

4. Upload the configuration to ZooKeeper:

```
solrctl config --upload test_collection_config $HOME/test_collection_con
fig
```

5. Create a new collection with two shards (specified by the -s parameter) using the named configuration (specified by the -c parameter).

```
solrctl collection --create test_collection -s 2 -c test_collection_config
```

Index sample data

Cloudera Search includes sample data for testing and validation. Run the relevant command to index this data for searching.

About this task

Replace *search01.example.com* in the example below with the name of any host running the Solr Server process.

Note: The default static port 8985 in the examples is only valid when you access the host from within the Data Hub cluster. When you want to access the Solr server from outside the cluster, you need to address it via the cdp-proxy-api endpoint. To find out the URL of the Solr server, navigate to the Data Hub Clusters service, or to Management Console Data Hub Clusters , and click on the tile representing your cluster. This brings you to the cluster details page where the URLs to cluster UIs and endpoints are listed. Select the **Endpoints** tab to find the service endpoint URLs.

For more information, see Accessing Cloudera Manager, cluster UIs, and endpoints.

Procedure

1. SSH to a Solr host.
2. Run the following command:

```
cd /opt/cloudera/parcels/CDH/share/doc/solr-doc*/example/exampledocs
                    find *.xml -exec curl -i -k --negotiate -u: https:
//search01.example.com:8985/solr/test_collection/update -H "Content-Type:
text/xml" --data-binary @{} \;
```

Related Information

Accessing Data Hub cluster via SSH

Query sample data

Run a query to verify that the sample data is successfully indexed and that you are able to search it.

About this task

Note: The default static port 8985 in the examples is only valid when you access the host from within the Data Hub cluster. When you want to access the Solr server from outside the cluster, you need to address it via the cdp-proxy-api endpoint. To find out the URL of the Solr server, navigate to the Data Hub Clusters service, or to Management Console Data Hub Clusters , and click on the tile representing your cluster. This brings you to the cluster details page where the URLs to cluster UIs and endpoints are listed. Select the **Endpoints** tab to find the service endpoint URLs.

For more information, see Accessing Cloudera Manager, cluster UIs, and endpoints.

Procedure

1. SSH into a cluster node.
2. Open the Solr admin web interface in a browser.

   https://*search01.example.com*:8985/solr

   Replace *search01.example.com* with the name of any host running the Solr Server process. If you have security enabled on your cluster, enter the credentials for the Kerberos principal when prompted.
3. Select Cloud from the left panel.
4. From the Collection Selector drop-down menu in the left panel, select the test_collection collection.
5. Select Query from the left panel and click Execute Query. If you see results such as the following, indexing was successful:


```
"response":{"numFound":32,"start":0,"maxScore":1.0,"docs":[
      {
        "id":"SP2514N",
        "name":["Samsung SpinPoint P120 SP2514N - hard drive - 250 GB -
ATA-133"],
        "manu":["Samsung Electronics Co. Ltd."],
        "manu_id_s":"samsung",
        "cat":["electronics",
          "hard drive"],
        "features":["7200RPM, 8MB cache, IDE Ultra ATA-133",
          "NoiseGuard, SilentSeek technology, Fluid Dynamic Bearing (FDB)
 motor"],
        "price":[92.0],
        "popularity":[6],
        "inStock":[true],
        "manufacturedate_dt":"2006-02-13T15:26:37Z",
        "store":["35.0752,-97.032"],
        "manu_str":["Samsung Electronics Co. Ltd."],
        "_version_":1650678864820568064,
        "cat_str":["electronics",
          "hard drive"],
        "name_str":["Samsung SpinPoint P120 SP2514N - hard drive - 250 GB
 - ATA-133"],
        "features_str":["7200RPM, 8MB cache, IDE Ultra ATA-133",
          "NoiseGuard, SilentSeek technology, Fluid Dynamic Bearing (FDB)
 motor"],
        "store_str":["35.0752,-97.032"]},
```


Indexing sample tweets with Cloudera Search

After you have verified that Cloudera Search is installed and running properly, you can experiment with other methods of ingesting and indexing data. This tutorial uses tweets to demonstrate batch indexing.

Related Information

To learn more about Solr, see the Apache Solr Tutorial

Create a collection for tweets

In this part of the Cloudera Search tutorial, you create a collection for tweets.

About this task

The remaining examples in the tutorial use the same collection, so make sure that you follow these instructions carefully.

Procedure

1. On a host with Solr Server installed, make sure that the SOLR_ZK_ENSEMBLE environment variable is set in /etc/solr/conf/solr-env.sh.

   For example:

   ```
   cat /etc/solr/conf/solr-env.sh
   ```

   ```
   export SOLR_ZK_ENSEMBLE=zk01.example.com:2181,zk02.example.com:2181,zk03
   .example.com:2181/solr
   ```

   This is automatically set on hosts with a Solr Server or Gateway role in Cloudera Manager.
2. If you are using Kerberos, kinit as the user that has privileges to create the collection.

   For example:

   ```
   kinit solr@EXAMPLE.COM
   ```

   Replace *solr@EXAMPLE.COM* with your user name and Kerberos realm name respectively.
3. Generate the configuration files for the collection, including the tweet-specific managed-schema:

   ```
   solrctl instancedir --generate $HOME/cloudera_tutorial_tweets_config
                     cp /opt/cloudera/parcels/CDH/share/doc/search*/search-
   crunch/solr/collection1/conf/schema.xml $HOME/cloudera_tutorial_tweets_c
   onfig/conf/managed_schema
   ```

   To the overwrite confirmation prompt:

   ```
   cp: overwrite 'cloudera_tutorial_tweets_config/conf/managed-schema'?
   ```

   type 'y'.
4. Upload the configuration to ZooKeeper:

   ```
   solrctl config --upload cloudera_tutorial_tweets_config $HOME/cloudera_t
   utorial_tweets_config
   ```
5. Create a new collection with two shards (specified by the -s parameter) using the named configuration (specified by the -c parameter):

   ```
   solrctl collection --create cloudera_tutorial_tweets -s 2 -c cloudera_tu
   torial_tweets_config
   ```

6. Verify that the collection is live. Open the Solr admin web interface in a browser by accessing the relevant URL:
   - https://*search01.example.com*:8985/solr/#/~cloud

   If you have Kerberos authentication enabled on your cluster, enter the credentials for the solr@EXAMPLE.COM principal when prompted. Replace *search01.example.com* with the name of any host running the Solr Server process. Look for the cloudera_tutorial_tweets collection to verify that it exists.
7. Prepare the configuration for use with MapReduce:

```
cp -r $HOME/cloudera_tutorial_tweets_config $HOME/cloudera_tutorial_twee
ts_mr_config
```

Copy sample tweets to HDFS

Copy the provided sample tweets to HDFS. These tweets are used to demonstrate the batch indexing capabilities of Cloudera Search.

Procedure

1. Copy the provided sample tweets to HDFS:

   Security Enabled:

   a. `kinit [***hdfs@EXAMPLE.COM***]`

   b. `hdfs dfs -mkdir -p /user/[***USER***]`

   c. `hdfs dfs -chown [***USER***]:[***GROUP***] /user/[***USER***]`

   d. `kinit [***USER@EXAMPLE.COM***]`

   e. `hdfs dfs -mkdir -p /user/[***USER***]/indir`

   f.
```
hdfs dfs -put /opt/cloudera/parcels/CDH/share/doc/search*/examples/test-
documents/sample-statuses-*.avro /user/[***USER***]/indir/
```

   g. `hdfs dfs -ls /user/[***USER***]/indir`

   Security Disabled: Run the following commands as *[***USER***]*:

```
sudo -u hdfs hdfs dfs -mkdir -p /user/[***USER***]
```

```
sudo -u hdfs hdfs dfs -chown [***USER***]:[***GROUP***] /user/[***USER***]
```

```
hdfs dfs -mkdir -p /user/[***USER***]/indir
```

```
hdfs dfs -put /opt/cloudera/parcels/CDH/share/doc/search*/examples/test-
documents/sample-statuses-*.avro /user/[***USER***]/indir/
```

```
hdfs dfs -ls /user/[***USER***]/indir
```

2. Ensure that outdir is empty and exists in HDFS:

```
hdfs dfs -rm -r -skipTrash /user/[***USER***]/outdir
```

```
hdfs dfs -mkdir /user/[***USER***]/outdir
```

```
hdfs dfs -ls /user/[***USER***]/outdir
```

What to do next

The sample tweets are now in HDFS and ready to be indexed. Continue to the next section to index the sample tweets.

Using MapReduce batch indexing to index sample Tweets

Batch indexing is useful for periodically indexing large amounts of data, or for indexing a dataset for the first time.

Before continuing, make sure that you have completed the procedures earlier in the tutorial.

Batch indexing into offline Solr shards

Batch indexing into offline Solr shards is mainly intended for offline use-cases by advanced users. Use cases requiring read-only indexes for searching can be handled by using batch indexing without the --go-live option. By not using GoLive, you can avoid copying datasets between segments, thereby reducing resource utilization.

About this task

Running the MapReduce job without GoLive causes the job to create a set of Solr index shards from a set of input files and write the indexes to HDFS. You can then explicitly point each Solr server to one of the HDFS output shard directories.

Procedure

1. If you are working with a secured cluster, configure your client JAAS file ($HOME/jaas.conf) as follows:

```
Client {
 com.sun.security.auth.module.Krb5LoginModule required
 useKeyTab=false
 useTicketCache=true
 principal="solr@EXAMPLE.COM";
 };
```

2. If you are using Kerberos, kinit as the user that has privileges to update the collection:

```
kinit jdoe@EXAMPLE.COM
```

   Replace EXAMPLE.COM with your Kerberos realm name.

3. Delete any existing documents in the cloudera_tutorial_tweets collection. If your cluster does not have security enabled, run the following commands as the solr user by adding sudo -u solr before the command:

```
solrctl collection --deletedocs cloudera_tutorial_tweets
```

4. Delete the contents of the outdir directory:

```
hdfs dfs -rm -r -skipTrash /user/jdoe/outdir/*
```

5. Run the MapReduce job as follows, replacing *nn01.example.com* in the command with your NameNode hostname.

   - Security enabled:

   ```
   YARN_OPTS="-Djava.security.auth.login.config=/path/to/jaas.conf" yarn
    jar /opt/cloudera/parcels/CDH/lib/solr/contrib/mr/search-mr-*-job.jar
    org.apache.solr.hadoop.MapReduceIndexerTool -D 'mapred.child.java.opt
   s=-Xmx500m' -D 'mapreduce.job.user.classpath.first=true' --log4j /opt/
   cloudera/parcels/CDH/share/doc/search*/examples/solr-nrt/log4j.propert
   ies --morphline-file /opt/cloudera/parcels/CDH/share/doc/search*/exa
   mples/solr-nrt/test-morphlines/tutorialReadAvroContainer.conf --outp
   ut-dir hdfs://nn01.example.com:8020/user/jdoe/outdir --verbose --zk-h
   ost zk01.example.com:2181/solr --collection cloudera_tutorial_tweets --s
   hards 2 hdfs://nn01.example.com:8020/user/jdoe/indir
   ```

6. Check the job status at:

   ```
   http://rm01.example.com:8088/ui2/#/yarn-apps/apps
   ```

   For secure clusters, replace http with https and port 8088 with 8090.

7. After the job is completed, check the generated index files. Individual shards are written to the results directory with names of the form part-00000, part-00001, part-00002, and so on. This example has two shards:

   ```
   hdfs dfs -ls /user/jdoe/outdir/results
   ```

   ```
   hdfs dfs -ls /user/jdoe/outdir/results/part-00000/data/index
   ```

8. In the Cloudera Manager web console for the cluster, stop the Solr service (Solr service Actions Stop).

9. Identify the paths to each Solr core:

   ```
   hdfs dfs -ls /solr/cloudera_tutorial_tweets
   ```

   ```
   Found 2 items
   drwxr-xr-x   - solr solr          0 2017-03-13 06:20 /solr/cloudera_tutori
   al_tweets/core_node1
   drwxr-xr-x   - solr solr          0 2017-03-13 06:20 /solr/cloudera_tut
   orial_tweets/core_node2
   ```

10. Move the index shards into place.

    a) (Kerberos only) Switch to the solr user:

    ```
    kinit solr@EXAMPLE.COM
    ```

    b) Remove outdated files. If your cluster does not have security enabled, run the following commands as the solr user by adding sudo -u solr before the command:

    ```
    hdfs dfs -rm -r -skipTrash /solr/cloudera_tutorial_tweets/core_node1/dat
    a/index
    hdfs dfs -rm -r -skipTrash /solr/cloudera_tutorial_tweets/core_node1/dat
    a/tlog
    hdfs dfs -rm -r -skipTrash /solr/cloudera_tutorial_tweets/core_node2/
    data/index
    hdfs dfs -rm -r -skipTrash /solr/cloudera_tutorial_tweets/core_node2/
    data/tlog
    ```

c) Change ownership of the results directory to solr. If your cluster has security enabled, kinit as the HDFS superuser (hdfs by default) before running the following command. If your cluster does not have security enabled, run the command as the HDFS superuser by adding sudo -u hdfs before the command:

```
hdfs dfs -chown -R solr /user/jdoe/outdir/results
```

d) (Kerberos only) Switch to the solr user:

```
kinit solr@EXAMPLE.COM
```

e) Move the two index shards into place. If your cluster does not have security enabled, run the following commands as the solr user by adding sudo -u solr before the command:

```
hdfs dfs -mv /user/jdoe/outdir/results/part-00000/data/index /solr/clou
dera_tutorial_tweets/core_node1/data
```

```
hdfs dfs -mv /user/jdoe/outdir/results/part-00001/data/index /solr/clou
dera_tutorial_tweets/core_node2/data
```

11. In the Cloudera Manager web console for the cluster, start the Solr service (Solr service Actions Start).

12. Run some Solr queries. For example, for a Solr server running on search01.example.com, go to one of the following URLs in a browser, depending on whether you have enabled security on your cluster:

- Security enabled: https://search01.example.com:8985/solr/cloudera_tutorial_tweets/select?q=*:*

If indexing was successful, this page displays the first 10 query results.