



链滴

Docker 部署 Solr 极简教程

作者: [Gakkiyomi2019](#)

原文链接: <https://ld246.com/article/1608172447194>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



solr是apache旗下的一款开源，高效的搜索引擎，底层使用的apache的lucene。这篇文章将以极简方式搭建单机版的solr。本教程旨在试玩与学习，生产环境搭建还请移步官方文档与相关社区。

我作为一个solr新手来搭建环境时，首先肯定是从官方文档的quick start开始啦,但是官方文档里面居没有docker安装的教程，这对新时代的我来说是无法接受的。但是网上的各种docker安装教程不知是因为solr镜像版本的原因还是其他原因，在我使用data-import的功能时，各种目录对不上，以及各种sdk和api介绍搞得我眼花缭乱，无用的信息太多只会让自己的脑子发热以及打击自信心。

所以在探索出我要的东西之后，决定写下这个教程只服务于docker安装以及data-import功能介绍和用。如若与您所需要的东西不符，那也没必要看下去了。

在安装前请确保服务器能连接互联网与存在docker服务。

提示

本文只介绍solr的docker安装与data-import功能对于postgresql的使用。如需了解solr的api或者sdk的使用请移步官方文档与相关社区。

下载镜像

直接在docker里面下载官方镜像，以最新版为例。

`docker pull solr`

```
[root@nap-146 ~]# docker images | grep solr
solr          latest          6a8d720e47a0   4 days ago    520MB
[root@nap-146 ~]#
```

准备工作

- 需要创建工作目录，方便我们挂载配置文件，用来测试solr的data-import功能

```
[root@nap-222 solr]# pwd
/srv/solr
[root@nap-222 solr]# ls
data lib
[root@nap-222 solr]#
```

创建data目录用来挂载配置文件 空的就行启动后docker会把容器内的配置文件复制出来

创建lib目录放置postgres-jdbc-driver的jar包

- 准备postgres-jdbc-driver的jar包

下载好jar包放入lib目录

```
[root@nap-222 solr]# cd lib
[root@nap-222 lib]# ls
postgresql-42.2.2.jar
[root@nap-222 lib]#
```

启动容器

```
docker run -d -p 8983:8983 -v /srv/solr/lib:/data-lib -v /srv/solr/data:/var/solr/data -t --privileged=true --name solr solr
```

启动完成后已root用户进入docker容器

```
docker exec -it -u root solr bash
```

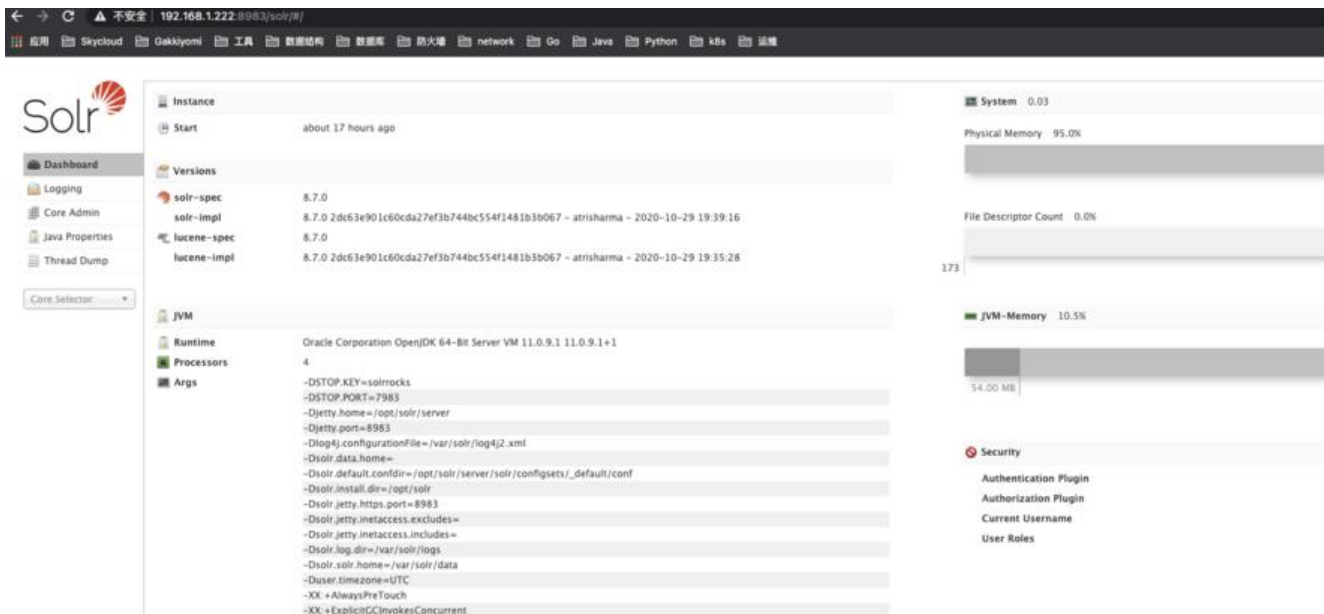
在容器里将/data-lib目录下的jdbc-driver包cp到/opt/solr-8.7.0/server/solr-webapp/webapp/WEB-INF/lib

在容器里将/opt/solr-8.7.0/dist/solr-dataimporthandler-8.7.0.jarcp到/opt/solr-8.7.0/server/solr-webapp/webapp/WEB-INF/lib

```
root@ccc493b3f4fd:/opt/solr-8.7.0/server/solr-webapp/webapp/WEB-INF/lib# ls
antlr4-runtime-4.5.1-1.jar          hadoop-auth-3.2.0.jar             kerb-core-1.0.1.jar              netty-common-4.1.50.Final.jar
asm-8.0.1.jar                     hadoop-common-3.2.0.jar           kerb-util-1.0.1.jar             netty-handler-4.1.50.Final.jar
asm-commons-8.0.1.jar             hadoop-hdfs-client-3.2.0.jar     kerby-asn1-1.0.1.jar            netty-resolver-4.1.50.Final.jar
avatica-core-1.13.0.jar           hppc-0.8.1.jar                   kerby-pkix-1.0.1.jar            netty-transport-4.1.50.Final.jar
caffeine-2.8.4.jar                htrace-core4-4.1.0-incubating.jar lucene-analyzers-common-8.7.0.jar netty-transport-native-epoll-4.1.50.Final.jar
calcite-core-1.18.0.jar           http2-client-9.4.27.v20200227.jar lucene-analyzers-kuromoji-8.7.0.jar netty-transport-native-unix-common-4.1.50.Final.jar
calcite-linq4j-1.18.0.jar         http2-common-9.4.27.v20200227.jar lucene-analyzers-nori-8.7.0.jar  opentracing-api-0.33.0.jar
commons-cli-1.4.jar               http2-hpack-9.4.27.v20200227.jar  lucene-analyzers-phonetic-8.7.0.jar opentracing-noop-0.33.0.jar
commons-codec-1.13.jar           http2-http-client-transport-9.4.27.v20200227.jar lucene-backward-codecs-8.7.0.jar opentracing-util-0.33.0.jar
commons-collections-3.2.2.jar     httpclient-4.5.12.jar            lucene-classification-8.7.0.jar  postgresql-42.2.2.jar
commons-collections4-4.4.jar      httpcore-4.4.13.jar              lucene-codecs-8.7.0.jar         protobuf-java-3.11.0.jar
commons-compiler-3.0.9.jar        httpmime-4.5.12.jar              lucene-core-8.7.0.jar           re2j-1.2.jar
commons-configuration2-2.11.1.jar jackson-annotations-2.11.2.jar    lucene-expressions-8.7.0.jar    rrd4j-3.5.jar
commons-exec-1.3.jar              jackson-core-2.11.2.jar          lucene-grouping-8.7.0.jar       s2-geometry-library-java-1.0.0.jar
commons-io-2.8.0.jar              jackson-databind-2.11.2.jar      lucene-highlighter-8.7.0.jar    snappy-java-1.1.7.6.jar
commons-lang-2.6.jar              jackson-dataformat-smile-2.11.2.jar lucene-join-8.7.0.jar           solr-core-8.7.0.jar
commons-lang3-3.10.jar            janino-3.0.9.jar                 lucene-memory-8.7.0.jar         solr-dataimporthandler-8.7.0.jar
commons-math3-3.6.1.jar           java-sever-0.9.0.jar              lucene-misc-8.7.0.jar           solr-solrj-8.7.0.jar
commons-text-1.6.jar              jetty-alpn-client-9.4.27.v20200227.jar lucene-queries-8.7.0.jar        spatial4j-0.7.jar
curator-client-2.13.0.jar         jetty-alpn-java-client-9.4.27.v20200227.jar lucene-queryparser-8.7.0.jar    stax2-api-3.1.4.jar
curator-framework-2.13.0.jar     jetty-client-9.4.27.v20200227.jar lucene-sandbox-8.7.0.jar        t-digest-3.1.jar
curator-recipes-2.13.0.jar       jetty-http-9.4.27.v20200227.jar  lucene-spatial-extras-8.7.0.jar woodstox-core-asl-4.4.1.jar
disruptor-3.4.2.jar              jetty-io-9.4.27.v20200227.jar    lucene-spatial3d-8.7.0.jar     zookeeper-3.6.2.jar
eigenbase-properties-1.1.5.jar   jetty-util-9.4.27.v20200227.jar  lucene-suggest-8.7.0.jar        zookeeper-jute-3.6.2.jar
guava-25.1-jre.jar               jose4j-0.6.5.jar                  lucene-suggest-8.7.0.jar        netty-buffer-4.1.50.Final.jar
hadoop-annotations-3.2.0.jar     json-path-2.4.0.jar               netty-4.1.50.Final.jar          netty-codec-4.1.50.Final.jar
hadoop-annotations-3.2.0.jar     jzlib-1.0.0.jar                   netty-4.1.50.Final.jar          netty-codec-4.1.50.Final.jar
```

到这里就可以打开solr的admin控制页面了

访问目标ip的8983端口 192.168.1.222:8983



创建一个新的索引库

```
docker exec -it --user=solr solr bin/solr create_core -c nap
```

由于我的配置文件挂载到宿主机上了，所以我们只要修改宿主机上的文件即可。

进入宿主主机上新建立的数据目录，这时候我们创建的索引库已经在文件夹里生成了

```
[root@nap-222 data]# ls
filestore nap solr.xml userfiles zoo.cfg
[root@nap-222 data]#
```

写入solrconfig.xml

进入目标索引库的conf目录

```
[root@nap-222 conf]# pwd
/srv/solr/data/nap/conf
[root@nap-222 conf]# ls
data-config.xml dataimport.properties lang managed-schema protwords.txt solrconfig.xml stopwords.txt synonyms.txt
[root@nap-222 conf]#
```

在这个<selcet>所在的标签上面新添加一个标签

```
http://wiki.apache.org/solr/SearchHandler

For processing Search Queries, the primary Request Handler
provided with Solr is "SearchHandler" It delegates to a sequent
of SearchComponents (see below) and supports distributed
queries across multiple shards

-->
<requestHandler name="/dataimport"
  class="org.apache.solr.handler.dataimport.DataImportHandler">
  <lst name="defaults">
    <str name="config">data-config.xml</str>
  </lst>
</requestHandler>
<requestHandler name="/select" class="solr.SearchHandler">
  <!-- default values for query parameters can be specified, these
  will be overridden by parameters in the request
  -->
  <lst name="defaults">
    <str name="echoParams">explicit</str>
    <int name="rows">10</int>
  <!-- Default search field
```

```
<requestHandler name="/dataimport"
  class="org.apache.solr.handler.dataimport.DataImportHandler">
  <lst name="defaults">
    <str name="config">data-config.xml</str>
  </lst>
</requestHandler>
```

写入data-config.xml

在conf目录添加data-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<dataConfig>
  <dataSource type="JdbcDataSource" driver="org.postgresql.Driver" url="jdbc:postgresql:/
192.168.1.54:5432/nap2" user="postgres" password="r00tme"/>
  <document name="test">
    <entity name="test" pk="id"
query="select id,created_at,deleted from test">
      <field column="id" name="id"/>
      <field column="created_at" name="created_at"/>
      <field column="deleted" name="deleted"/>
    </entity>
  </document>
</dataConfig>
```

[root@nap-222 conf]#

里面填上自己的数据源，支持多个和多种数据源，按规则配置即可

写入managed-schema

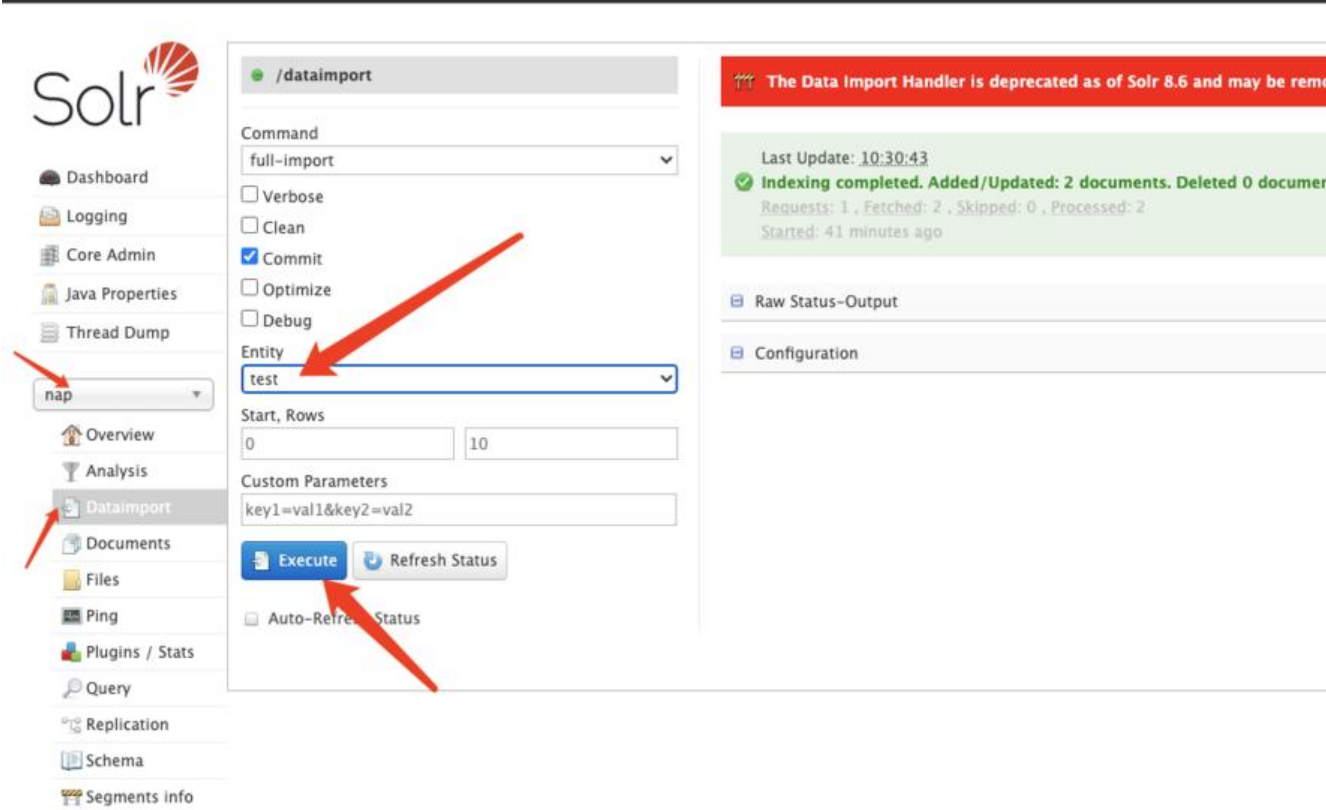
```
<fieldType name="text_ws" class="solr.TextField" positionIncrementGap="100">
  <analyzer>
    <tokenizer class="solr.WhitespaceTokenizerFactory"/>
  </analyzer>
</fieldType>
<field name="_version_" type="long" indexed="true" stored="true"/>
<field name="created_at" type="pdate" indexed="true" stored="true"/>
<field name="deleted" type="boolean" indexed="true" stored="true"/>
<field name="id" type="string" multiValued="false" indexed="true" required="true" stored="true"/>
<dynamicField name="*_txt_en_split_tight" type="text_en_splitting_tight" indexed="true" stored="true"/>
<dynamicField name="*_descendent_path" type="descendent_path" indexed="true" stored="true"/>
<dynamicField name="*_ancestor_path" type="ancestor_path" indexed="true" stored="true"/>
<dynamicField name="*_txt_en_split" type="text_en_splitting" indexed="true" stored="true"/>
<dynamicField name="*_txt_sort" type="text_gen_sort" indexed="true" stored="true"/>
```

添加相应的field，id默认不能删除，_version_也不能删除，删除solr启动会报错。

重启容器

docker restart solr

在admin控制台手动同步数据



查询数据

The screenshot displays the Solr Admin interface. On the left is a navigation menu with options like Dashboard, Logging, Core Admin, Java Properties, Thread Dump, and a 'Query' section which is highlighted with a red arrow. The main area is titled 'Request-Handler (qt)' and contains a search box with '/select'. Below this are various configuration fields: 'common', 'q' (with '*:*' entered), 'fq', 'sort', 'start, rows' (with '0' and '10' entered), 'fl', 'df', 'Raw Query Parameters' (with 'key1=val1&key2=val2'), and 'wt' (with '-----' selected). A list of checkboxes for query options is shown, including 'indent off', 'debugQuery', 'dismax', 'edismax', 'hl', 'facet', 'spatial', and 'spellcheck'. A blue 'Execute Query' button is at the bottom, with a red arrow pointing to it. The right pane shows the URL 'http://192.168.1.222:8983/solr/nap/select?q=%3A*' and a JSON response:

```
{
  "responseHeader": {
    "status": 0,
    "QTime": 1,
    "params": {
      "q": "*:*",
      "_: "1608168778015"}},
  "response": {
    "numFound": 2,
    "start": 0,
    "numFoundExact": true,
    "docs": [
      {
        "deleted": false,
        "created_at": "2020-12-09T14:45:12.398Z",
        "id": "5cccc344-4ad2-4b35-85c0-c535c205c575",
        "_version_": "1686288250535149568"},
      {
        "deleted": true,
        "created_at": "2020-12-09T14:47:29.669Z",
        "id": "a89e887a-8c2c-4a25-beeb-3aaba9cf5e28",
        "_version_": "1686288250539343872"}]
  }
}
```

修改postgres源数据solr是否能触发同步?

修改数据源里的数据solr不会察觉同步，所以需要自己维护，例如api通知，或者使用solr的插件可以置定时同步与轮询。

总结

本文作为一个新手试玩教程应该还算是比较简单明了，如果按照此教程安装有不了解的也可以在评论区言，欢迎交流，斧正。