Documentation

Setting Up Apache Solr

Development and Production

Ron Kow

December 2020

Table of Contents

Chapter 1

Introduction

This report documents the setup of an Apache Solr server in development (Chapter 2), and in a production server (Chapter 3).

This documentation is written as a tutorial for developers with no knowledge of Solr. Knowledge of Linux command line is required. Brevity with clarity is the goal. If more details are helpful for better understanding at any step, I will provide the link to the official documentation.

The practices in this document are my own adaptations of recommended practices. I welcome any feedback. You can contact me at `ronkow2020@gmail.com`.

Chapter 2

Development Setup

This chapter describes the setup of a Solr server in **Windows 10 Home**. We will use Solr version 7.7.3.

2.1 Software Requirements

The only requirement for running Solr is Java Runtime Environment (JRE) version 1.8 or later (https://lucene.apache.org/solr/guide/7_7/solr-system-requirements.html).

Download and install JRE from https://www.java.com/en/download/. When installation is complete, check the version in Windows PowerShell:

```
PS C:\> java -version
java version "1.8.0_271"
Java(TM) SE Runtime Environment (build 1.8.0_271-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.271-b09, mixed mode)
```

2.2 Solr Setup

2.2.1 Creating a Solr Node

We will now set up a Solr server, also called a *Solr node*. To store data and create an index, we also need to create a *core* in the node. Setting up a Solr node without any cores is a three-step process. We just need to download the required version to our project directory, decompress the file, and start the server.

Create a new directory for our project. Let's name the project directory `grammar`. Download `solr-7.7.3.zip`:

https://archive.apache.org/dist/lucene/solr/7.7.3/

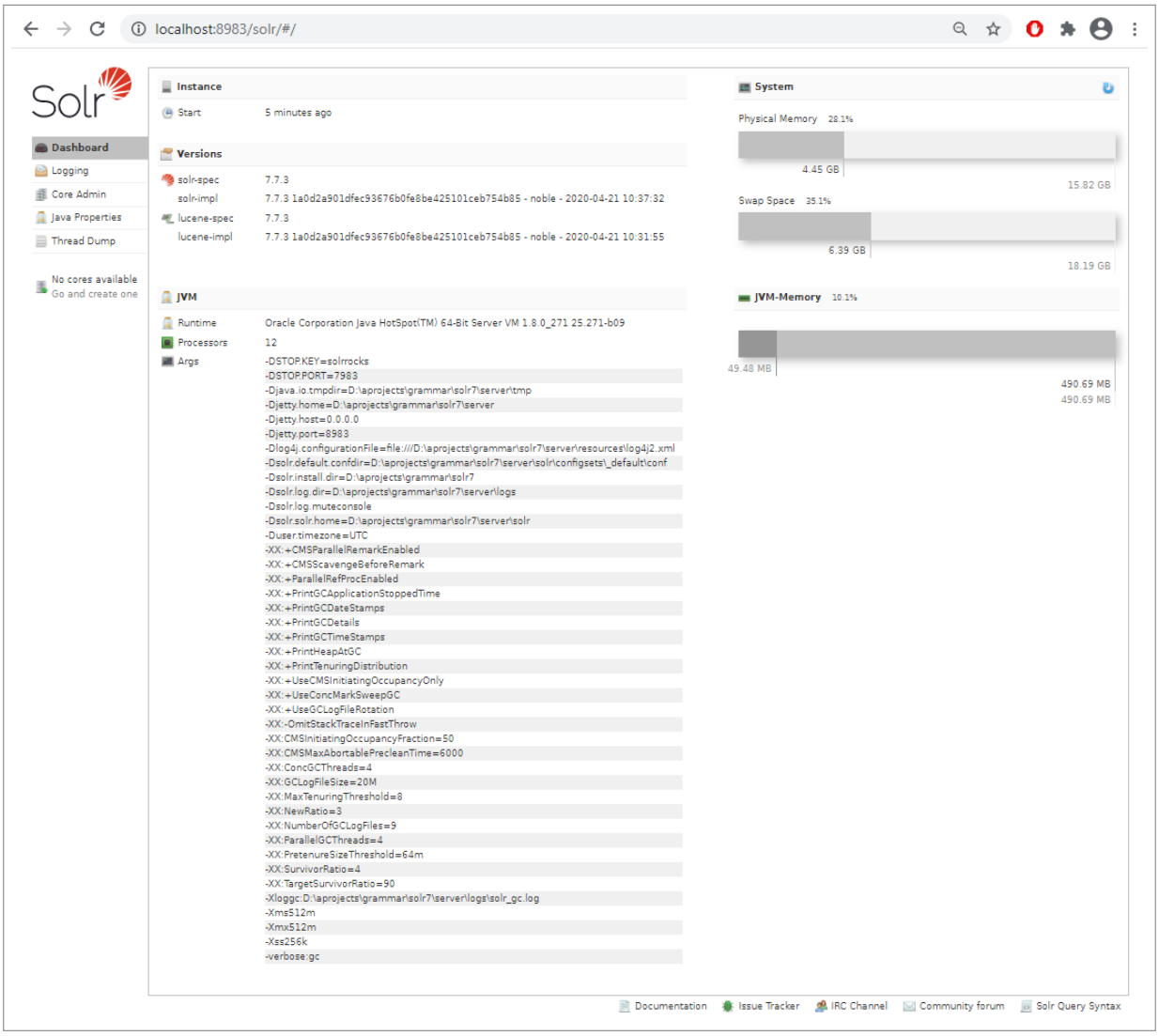
The ZIP file contains a single directory named `solr-7.7.3`. Move this directory to the project directory. Rename it `solr7`. In Windows PowerShell, go to this directory and start the Solr server:

```
PS D:\> cd projects\grammar\solr7
PS D:\projects\grammar\solr7> bin/solr start
Waiting up to 30 to see Solr running on port 8983
Started Solr server on port 8983. Happy searching!
```

Other than `start`, other commands are `stop`, `restart`, and `status`. The following shows the status output:

```
PS D:\projects\grammar\solr7> bin/solr status

Found Solr process 2412 running on port 8983
{
  "solr_home":"D:\\projects\\grammar\\solr7\\server\\solr",
  "version":"7.7.3 1a0d2a901dfec93676b0fe8be425101ceb754b85 - noble - 2020-04-21 10:37:32",
  "startTime":"2020-12-22T17:21:43.818Z",
  "uptime":"0 days, 0 hours, 0 minutes, 18 seconds",
  "memory":"52.5 MB (%10.7) of 490.7 MB"}
```

In the web browser, you will see the Solr Administration User Interface (Solr Admin UI) at `http://localhost:8983`:

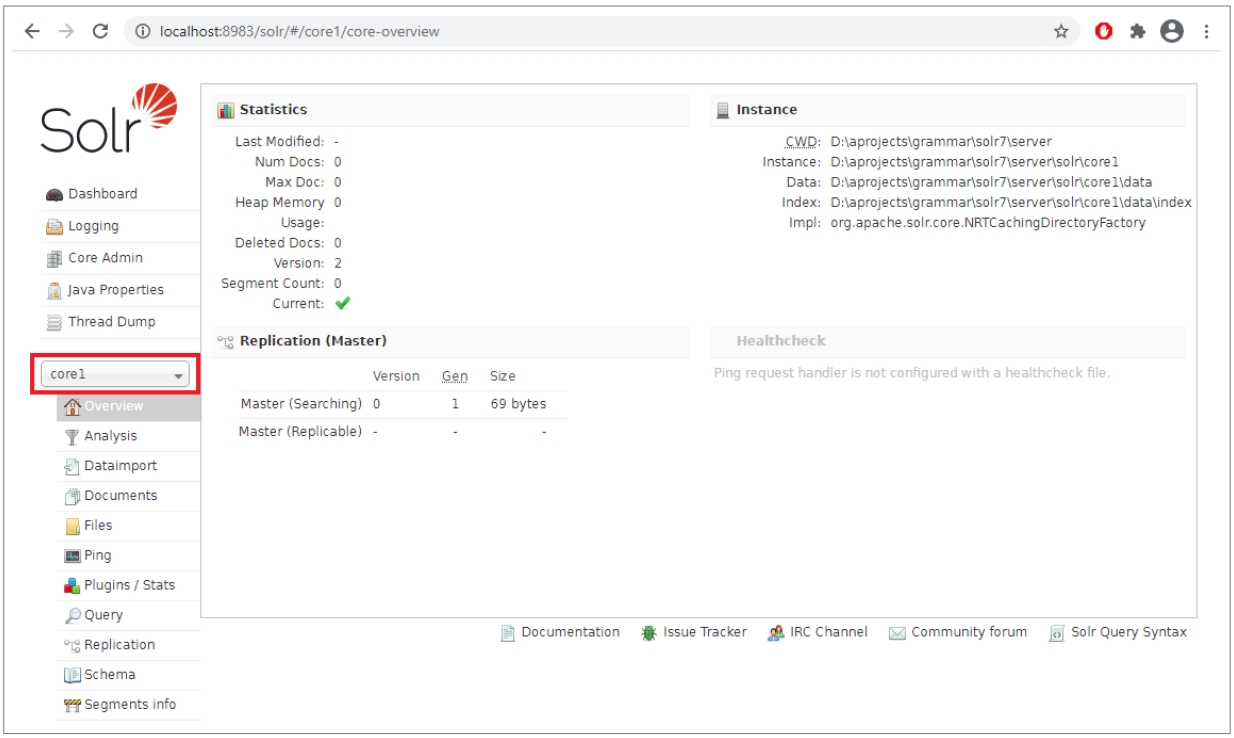


2.2.2 Creating a Core

We have just created a Solr node. A *Solr instance*, representing a logical index, is called a *core*. We can create and run multiple cores in a single Solr node. Let's create a core named `core1`:

```
PS D:\projects\grammar\solr7> bin/solr create -c core1
WARNING: Using _default configset with data driven schema functionality.
NOT RECOMMENDED for production use.
To turn off: bin\solr config -c core1 -p 8983 -action set-user-property
-property update.autoCreateFields -value false

Created new core 'core1'
```

As we are not in production environment, we can ignore the warning. Refresh the Solr Admin UI and we will now be able to select the newly created core from **Core Selector**:



2.2.3 Deleting a Core

To delete a previously created core:

```
PS D:\projects\grammar\solr7> bin/solr delete -c core_name
```

2.2.4 Setting Up Learning To Rank

To enable Learning To Rank (LTR) for `core1`, we need to add the following configuration to the file `solrconfig.xml`, located in the `core1` directory `D:\projects\grammar\solr7\server\solr\core1\conf\`:

- Include the module `solr-ltr-7.7.3.jar`, located at `D:\projects\grammar\solr7\dist\`
- Declare the `ltr` query parser plugin
- Configure the `QUERY_DOC_FV` cache (feature values cache)
- Declare the `features` transformer

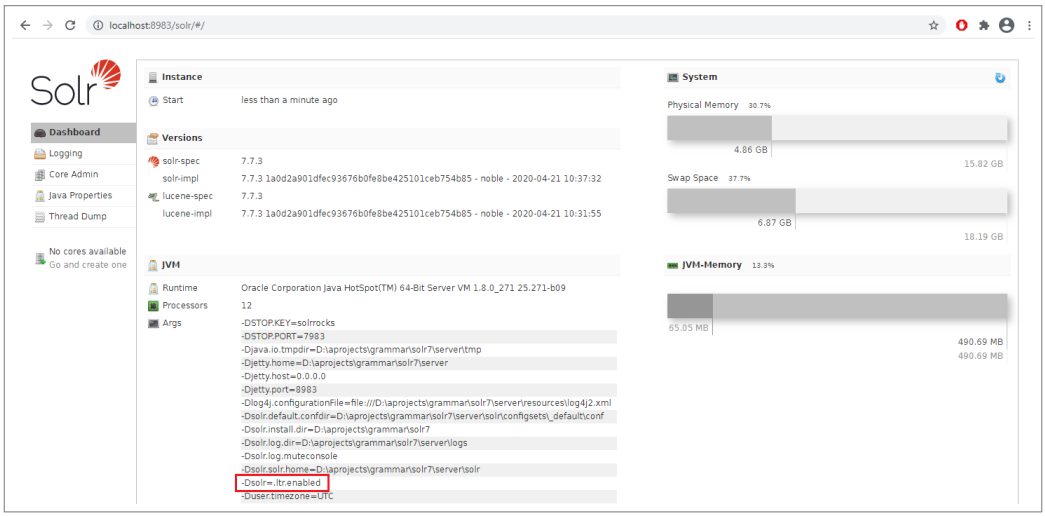
Put the following code before `</config>` at the last line of `solrconfig.xml`:

```
<lib dir="${solr.install.dir:../../../..}/dist/" regex="solr-ltr-\d.*\.jar" />

<queryParser name="ltr" class="org.apache.solr.ltr.search.LTRQParserPlugin"/>

<cache name="QUERY_DOC_FV"
       class="solr.search.LRUCache"
       size="4096"
       initialSize="2048"
       autowarmCount="4096"
       regenerator="solr.search.NoOpRegenerator" />

<transformer name="features"
  class="org.apache.solr.ltr.response.transform.LTRFeatureLoggerTransformerFactory">
    <str name="fvCacheName">QUERY_DOC_FV</str>
</transformer>
```

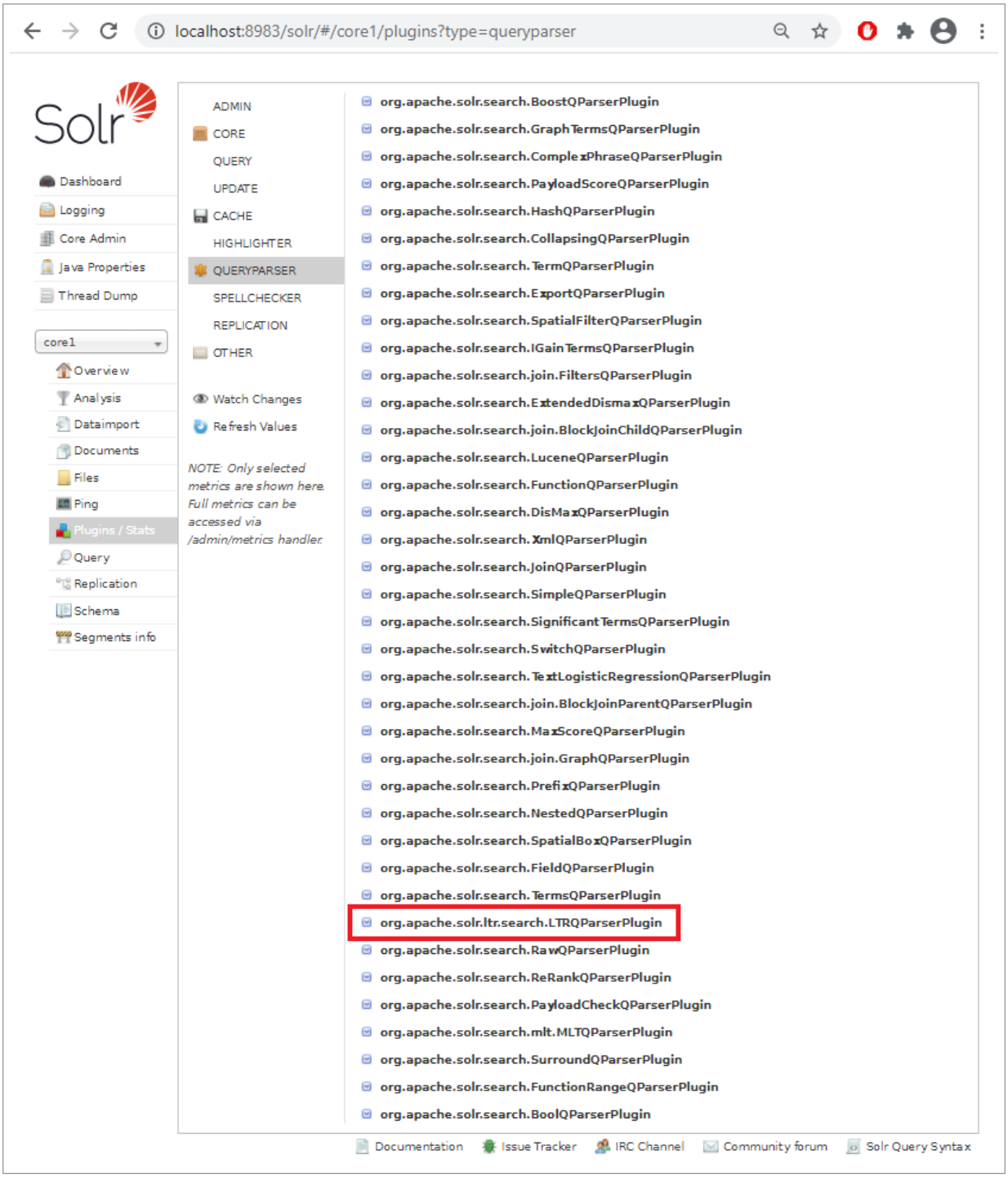
Restart the Solr server with the LTR plugin enabled:

```
PS D:\projects\grammar\solr7> bin/solr restart -Dsolr.ltr.enabled -p 8983
```

Refresh the Solr Admin UI. The Dashboard will show that LTR is now enabled:



In the list of Query Parser plugins for `core1`, you will see the LTR plugin:



Chapter 3

Production Setup

3.1 Virtual Private Server

Setup of Solr server on a production server requires root access. In this document, we describe the setup on an Ubuntu virtual private server (VPS).

3.1.1 Memory Requirements

Start with 1 GB and monitor the memory usage. If you find your Solr server stopping on its own, it is likely that you need more memory.

3.1.2 Login From Command Line Interface

We will now access the server from Windows PowerShell. We assume the following settings:

- Username: `root`
- Server IP address: `11.22.33.44`
- Secure Shell protocol (SSH) port: `7000`
- Server name: `server`

Log in to your account:

```
PS D:\> ssh -p 7000 root@11.22.33.44
root@11.22.33.44's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0 x86_64)
* Documentation:  https://help.ubuntu.com
* Management:     https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage
Last login: Tue Dec 22 12:58:17 2020 from 138.75.182.210
root@server:~#
```

3.2 Software Requirements

As before, we need to install Java Runtime Environment (JRE). We can either install OpenJRE or Oracle JRE. The following is a tutorial:

https://ubuntu.com/tutorials/install-jre.

To install OpenJRE:

```
root@server:~# sudo apt install openjdk-8-jre
```

When installation is complete, check the version is 1.8 or later:

```
root@server:~# java -version
openjdk version "1.8.0_275"
OpenJDK Runtime Environment (build 1.8.0_275-8u275-b01-0ubuntu1~20.04-b01)
OpenJDK 64-Bit Server VM (build 25.275-b01, mixed mode)
```

During setup of Solr, we will need curl to send data, and a text editor to edit text files. We will install `curl` and the text editor `nano`:

```
root@server:~# sudo apt install curl
root@server:~# sudo apt install nano
```

3.3 Solr Setup

The official documentation for production setup is at:

https://lucene.apache.org/solr/guide/7_7/taking-solr-to-production.html

I will now walk through the steps to set up a Solr server on the virtual private server. The process is quite different from the local setup in Windows. In development, all the files remain in one directory after setup. In production, the files are distributed to different locations in the server. It is important to know where these locations are and what files are stored in them so that we can troubleshoot any problems that we encounter.

Our goal is to get the Solr Admin UI running at `http://11.22.33.44:8983/`. After that, we need to set up authentication so that only authorized users can access the Admin UI.

The steps are:

1. Download Solr and extract the *service installation script* `install_solr_service.sh`.
2. Run the service installation script.
3. Increase the file limit and maximum processes limit in the Linux server.
4. Check that the Admin UI loads in the web browser.
5. Create a file named `security.json`. In this file, we configure and enable authentication and authorization.
6. Check that authentication is required to access the Admin UI in the web browser.
7. Add users (and passwords) to `security.json`.
8. Add a new core.
9. Enable Learning To Rank.

3.3.1 Service Installation Script

Download `Solr-7.7.3.tgz` to the Linux server:

```
root@server:~# wget https://archive.apache.org/dist/lucene/solr/7.7.3/solr-7.7.3.tgz
```

Extract the service installation script:

```
root@server:~# tar xzf solr-7.7.3.tgz solr-7.7.3/bin/install_solr_service.sh --strip-components=2
```

At this point, the current directory contains these two files:

```
root@server:~# ls
install_solr_service.sh  solr-7.7.3.tgz
```

Run the service installation script by using the following command provided by Solr. Inspect the output to see what the script does:

```
root@server:~# sudo bash ./install_solr_service.sh solr-7.7.3.tgz
id: ‘solr’: no such user
Creating new user: solr
Adding system user ‘solr’ (UID 109) ...
Adding new group ‘solr’ (GID 114) ...
Adding new user ‘solr’ (UID 109) with group ‘solr’ ...
Creating home directory ‘/var/solr’ ...

Extracting solr-7.7.3.tgz to /opt
Installing symlink /opt/solr -> /opt/solr-7.7.3 ...
Installing /etc/init.d/solr script ...
Installing /etc/default/solr.in.sh ...

Service solr installed.
Customize Solr startup configuration in /etc/default/solr.in.sh
solr.service - LSB: Controls Apache Solr as a Service
    Loaded: loaded (/etc/init.d/solr; generated)
    Active: active (running) since Sun 2020-12-20 05:37:03 UTC; 5s ago
      Docs: man:systemd-sysv-generator(8)
   Process: 3714 ExecStart=/etc/init.d/solr start (code=exited, status=0/SUCCESS)
     Tasks: 40 (limit: 19660)
    Memory: 181.6M
    CGroup: /system.slice/solr.service
            3755 java -server -Xms512m -Xmx512m -XX:NewRatio=3 -XX:SurvivorRatio=4
            -XX:TargetSurvivorRatio=90 -XX:MaxTenuringThreshold=8 -XX:+UseConcMarkSweepGC
            -XX:ConcGCThreads=4 -XX:ParallelGCThreads=4 -XX:+CMSScavengeBeforeRemark
            -XX:PretenureSizeThreshold=64m -X>

Dec 20 05:36:56 server.mydomain solr[3717]: *** [WARN] *** Your open file limit is currently 1024.
Dec 20 05:36:56 server.mydomain solr[3717]:  It should be set to 65000 to avoid operational disruption.
Dec 20 05:36:56 server.mydomain solr[3717]:  If you no longer wish to see this warning, set
                                               SOLR_ULIMIT_CHECKS to false in your profile or solr.in.sh
Dec 20 05:36:56 server.mydomain solr[3717]: *** [WARN] ***  Your Max Processes Limit is currently 62987.
Dec 20 05:36:56 server.mydomain solr[3717]:  It should be set to 65000 to avoid operational disruption.
Dec 20 05:36:56 server.mydomain solr[3717]:  If you no longer wish to see this warning, set
                                               SOLR_ULIMIT_CHECKS to false in your profile or solr.in.sh
Dec 20 05:37:03 server.mydomain solr[3717]: [194B blob data]
Dec 20 05:37:03 server.mydomain solr[3757]: Started Solr server on port 8983 (pid=3755). Happy searching!
Dec 20 05:37:03 server.mydomain solr[3717]: [14B blob data]
Dec 20 05:37:03 server.mydomain systemd[1]: Started LSB: Controls Apache Solr as a Service.
```

The service installation script does the following:

- Creates a new username (`solr`) and group (`solr`).
- Creates the *Solr home directory* `/var/solr/`. This directory stores writable files, that is, files that will be changed while Solr is running.
- Extracts `solr-7.7.3` to the *Solr installation directory* `/opt/solr-7.7.3/`. This directory stores files that will not be changed.
- Creates a link pointing `/opt/solr/` to `/opt/solr-7.7.3/`. This makes future upgrading of Solr easier. If we upgrade to Solr 8.0, the service installation script will just update this link to point to the later version.
- Installs a script named `solr` in `/etc/init.d/`.
- Installs a script named `solr.in.sh` in `/etc/default/`.
- Starts Solr and advises us to set our file limit and max processes limit to 65000.

3.3.2 Solr Installation Directory

The installation directory simply contains the complete installation directory structure:

```
root@server:~# cd /opt/solr
root@server:/opt/solr# ls
CHANGES.txt  LUCENE_CHANGES.txt  README.txt  contrib  docs     licenses
LICENSE.txt  NOTICE.txt          bin         dist     example  server
```

3.3.3 Solr Home Directory

It is important to know what the home directory contains.

```
root@server:~# cd /var/solr
root@server:/var/solr# ls
data  log4j2.xml  logs  solr-8983.pid

root@server:/var/solr# cd data
root@server:/var/solr/data# ls
solr.xml  zoo.cfg

root@server:/var/solr/data# cd ..
root@server:/var/solr# cd logs
root@server:/var/solr/logs# ls
solr-8983-console.log  solr.log.1  solr.log.3  solr.log.5  solr.log.7  solr_gc.log.0.current
solr.log               solr.log.2  solr.log.4  solr.log.6  solr.log.8  solr_slow_requests.log
```

Inspect the error messages in `/var/solr/logs/solr.log` if there are any problems.

3.3.4 Environment Variables

Environment variables are stored in the scripts `/etc/init.d/solr` and `/etc/default/solr.in.sh`. In the script `/etc/default/solr.in.sh`:

```
SOLR_PID_DIR="/var/solr"
SOLR_HOME="/var/solr/data"
LOG4J_PROPS="/var/solr/log4j2.xml"
SOLR_LOGS_DIR="/var/solr/logs"
SOLR_PORT="8983"
```

In the script `/etc/init.d/solr`:

```
SOLR_INSTALL_DIR="/opt/solr"
SOLR_ENV="/etc/default/solr.in.sh"
RUNAS="solr"
```

The script `/etc/init.d/solr` enables us to control Solr using the `service` application. So we can use `service` to start, stop, restart the Solr server, or check its status:

```
root@server:~# service solr start
root@server:~# service solr stop
root@server:~# service solr restart
root@server:~# service solr status
```

3.3.5 File Limit and Maximum Processes Limit

Edit the file `limits.conf` in the location `/etc/security/`:

```
root@server:~# cd /etc/security
root@server:/etc/security# nano limits.conf
```

Add the following four lines before `# End of file` at the last line to increase the file limit and maximum processes limit:

```
solr soft nofile 65000
solr hard nofile 6500
solr soft nproc 65000
solr hard nproc 65000
```

In the web browser, the Admin UI will be running at `http://11.22.33.44:8983`.

3.4 Authentication and Authorization

3.4.1 Creating `security.json`

Authentication and authorization settings are defined in a file named `security.json` in the location `/var/solr/data`, which we need to create as the user `solr` (instead of `root`):

```
root@server:~# su -l solr
solr@server:~# cd /var/solr/data
solr@server:/var/solr/data# touch security.json
solr@server:/var/solr/data# nano security.json
```

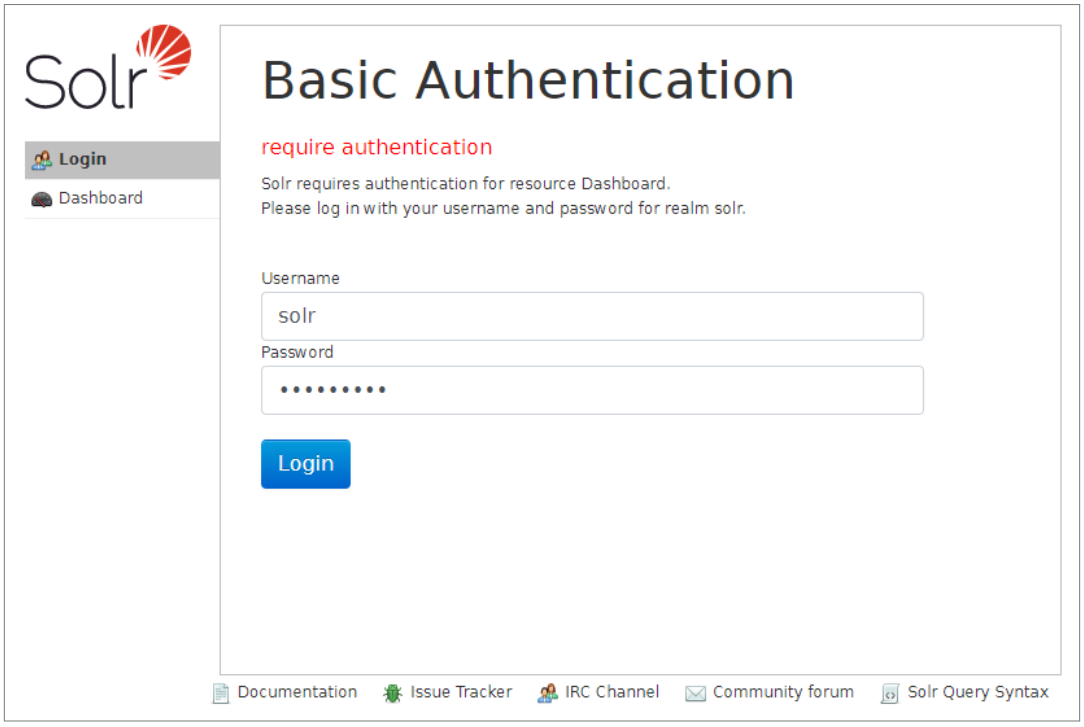
Add the following text to `security.json` (the text is copied from:
https://lucene.apache.org/solr/guide/7_7/basic-authentication-plugin.html):

```
{
  "authentication":{
    "blockUnknown":true,
    "class":"solr.BasicAuthPlugin",
    "credentials":{"solr":"IV0EHq1OnNrj6gvRCwvFwTrZ1+z1oBbnQdiVC3otuq0=
    Ndd7LKvVBAaZIF0QAVi1ekCfAJXr1GGfLtRUXhgrF8c="}
  },
  "authorization":{
    "class":"solr.RuleBasedAuthorizationPlugin",
    "permissions":[{"name":"security-edit",
                    "role":"admin"}],
    "user-role":{"solr":"admin"}
  }
}
```

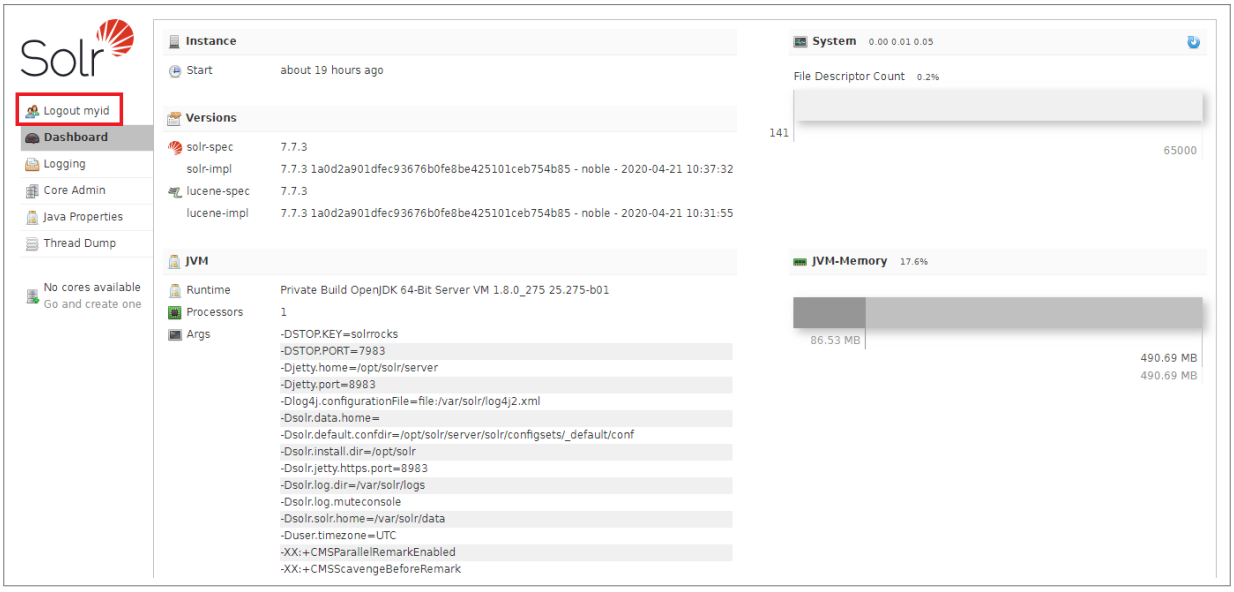
We have defined the following in `security.json`:

- We use the plugins `solr.BasicAuthPlugin` and `solr.RuleBasedAuthorizationPlugin`.
- `"blockUnknown":true` means that unauthenticated requests are not allowed.
- In `"credentials"`, we define a username `solr` with the encrypted password `SolrRocks`.
- In `"permissions"`, we define the `role`, and authorize it to edit `security.json`.
- In `"user-role"`, we assign to the username `solr` the `admin` role.

If we try to access the Admin UI at `http://11.22.33.44:8983`, we will be prompted to enter a username and password. We can log in using the username `solr` and password `SolrRocks`:



On the Admin UI, we will see a logout option:



3.4.2 Setting Environment Variables

In the script `/etc/default/solr.in.sh`, we need to set two environment variables `SOLR_AUTH_TYPE` and `SOLR_AUTHENTICATION_OPTS`. We simply need to un-comment the two lines provided in the script:

```
# Settings for authentication
# Please configure only one of SOLR_AUTHENTICATION_CLIENT_BUILDER or SOLR_AUTH_TYPE parameters
#SOLR_AUTHENTICATION_CLIENT_BUILDER="org.apache.solr.client.solrj.impl.PreemptiveBasicAuthClientBuilderFactory"
SOLR_AUTH_TYPE="basic"
SOLR_AUTHENTICATION_OPTS="-Dbasicauth=solr:SolrRocks"
```

3.4.3 Adding a New User

We will now add a new username and password to `security.json` using `curl`:

```
root@server:~# curl --user solr:SolrRocks http://localhost:8983/solr/admin/authentication
-H 'Content-type:application/json' -d '{''set-user":{''myid":''mypassword"}}'
```

`security.json` now includes the new credentials:

```
  "authentication":{
    "credentials":{"solr":"IV0EHq1OnNrj6gvRCwvFwTrZ1+z1oBbnQdiVC3otuq0=
                           Ndd7LKvVBAaZIF0QAVi1ekCfAJXr1GGfLtRUXhgrF8c="
                   "myid":"LjhBLN8qxCE0FMoAmbGxVsdIbdhkVXzwyZ1M/+2IEbU=
                           w+2WKqSWVxq22/WOnKQY/yt6yC+c1UDcjAqfAIkP2LI="}
  },
```

To learn more about securing Solr:

https://lucene.apache.org/solr/guide/7_7/securing-solr.html

3.5 Setup of Cores and Learning To Rank

3.5.1 Creating a Core

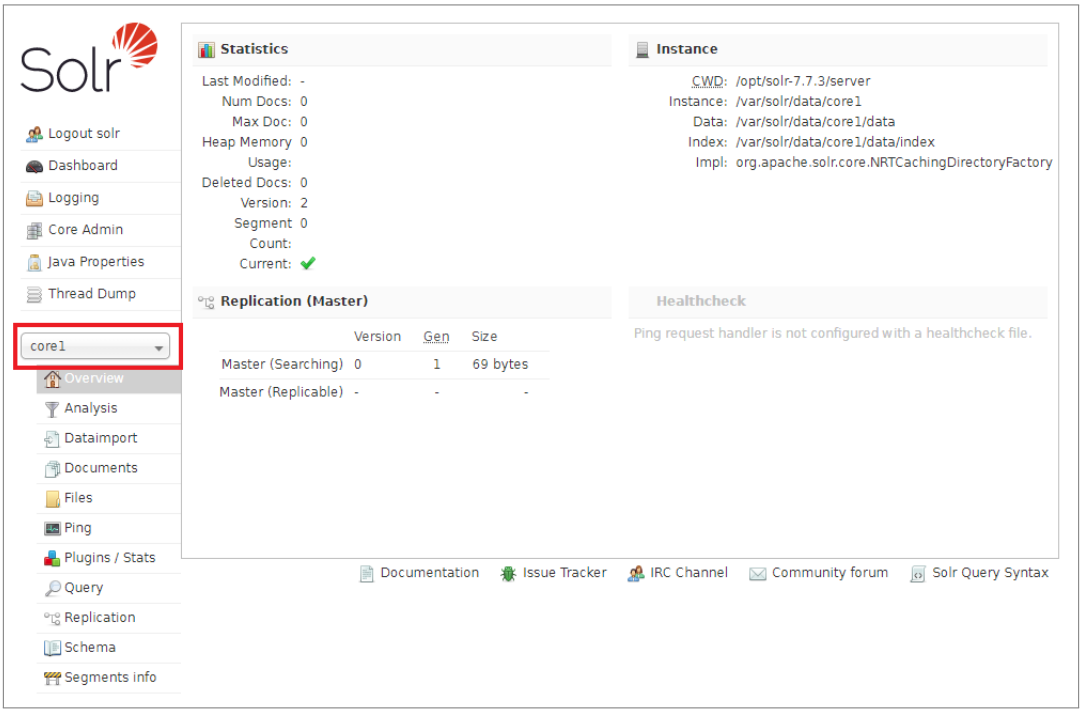
The process of creating a core in production is similar to what we did in development, except that the core directory will be created in the home directory `/var/solr/data/` instead of the installation directory `/opt/solr/`. We run the command in the installation directory as the user `solr`. Let's create a core named `core1`:

```
root@server:~# su -l solr
solr@server:~# cd /opt/solr/
solr@server:/opt/solr# bin/solr create -c core1
WARNING: Using _default configset with data driven schema functionality.
NOT RECOMMENDED for production use.
To turn off: bin/solr config -c core1 -p 8983 -action set-user-property
-property update.autoCreateFields -value false

Created new core 'core1'
```

The warning, which we can ignore, is explained here in the section **Create a New Collection**:

https://lucene.apache.org/solr/guide/7_7/solr-tutorial.html.

The core directory `core1` is created in `/var/solr/data/`. Refresh the Solr Admin UI and we will now be able to select `core1` from **Core Selector**:



3.5.2 Setting Up Learning To Rank

To enable Learning To Rank (LTR), we do what we did in development. Put the following code before `</config>` at the last line of `/var/solr/data/core1/conf/solrconfig.xml`:

```
<lib dir="${solr.install.dir:../../../..}/dist/" regex="solr-ltr-\d.*\.jar" />

<queryParser name="ltr" class="org.apache.solr.ltr.search.LTRQParserPlugin"/>

<cache name="QUERY_DOC_FV"
       class="solr.search.LRUCache"
       size="4096"
       initialSize="2048"
       autowarmCount="4096"
       regenerator="solr.search.NoOpRegenerator" />

<transformer name="features"
             class="org.apache.solr.ltr.response.transform.LTRFeatureLoggerTransformerFactory">
    <str name="fvCacheName">QUERY_DOC_FV</str>
</transformer>
```

Restart Solr:

```
root@server:~# service solr restart
```