Steps to setup HDP search:

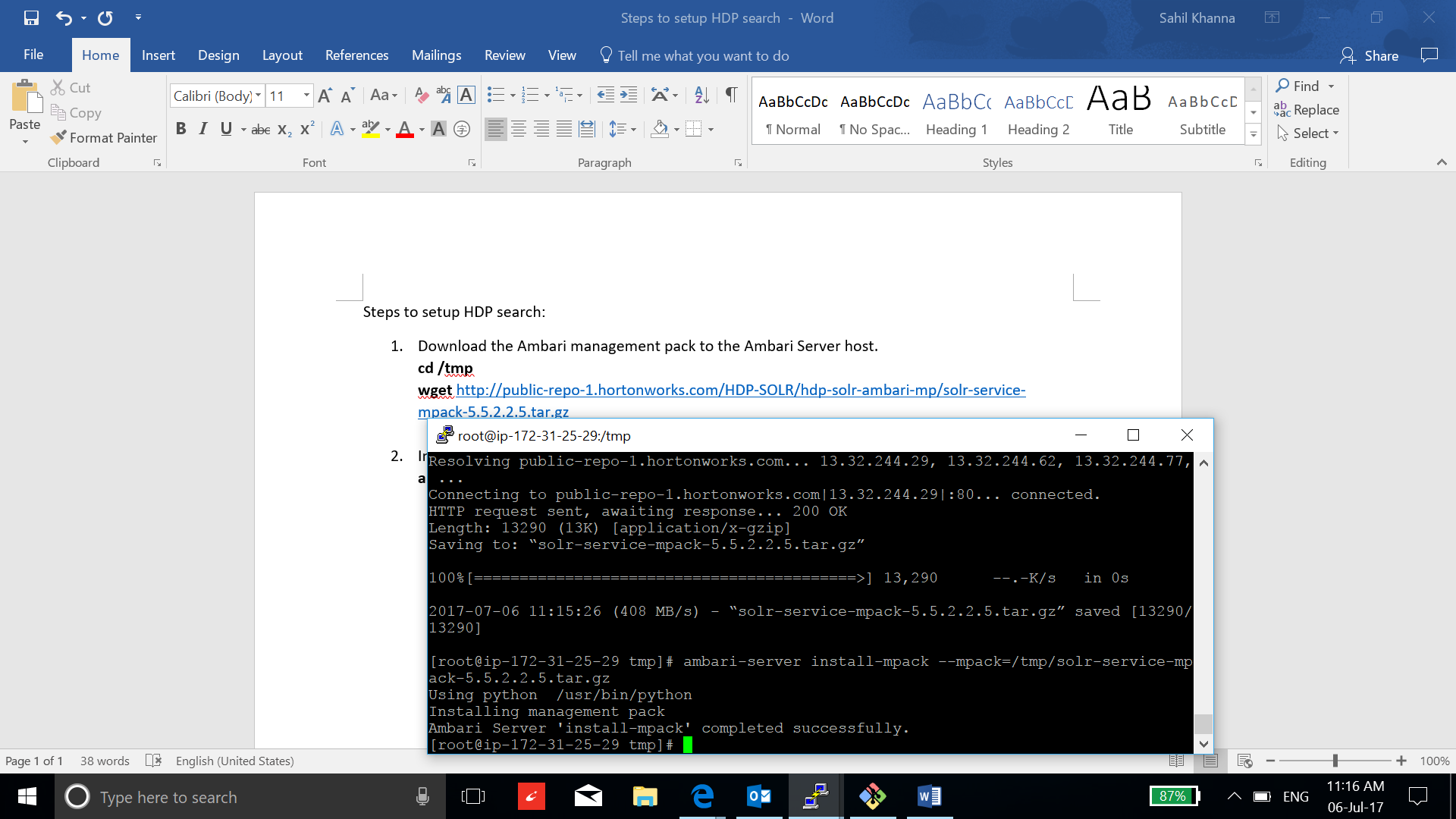
1. Download the Ambari management pack to the Ambari Server host.

**cd /tmp**

**wget** <http://public-repo-1.hortonworks.com/HDP-SOLR/hdp-solr-ambari-mp/solr-service-mpack-5.5.2.2.5.tar.gz>

1. Install the management pack on the Ambari Server host, using the following command:

**ambari-server install-mpack --mpack=/tmp/solr-service-mpack-5.5.2.2.5.tar.gz**



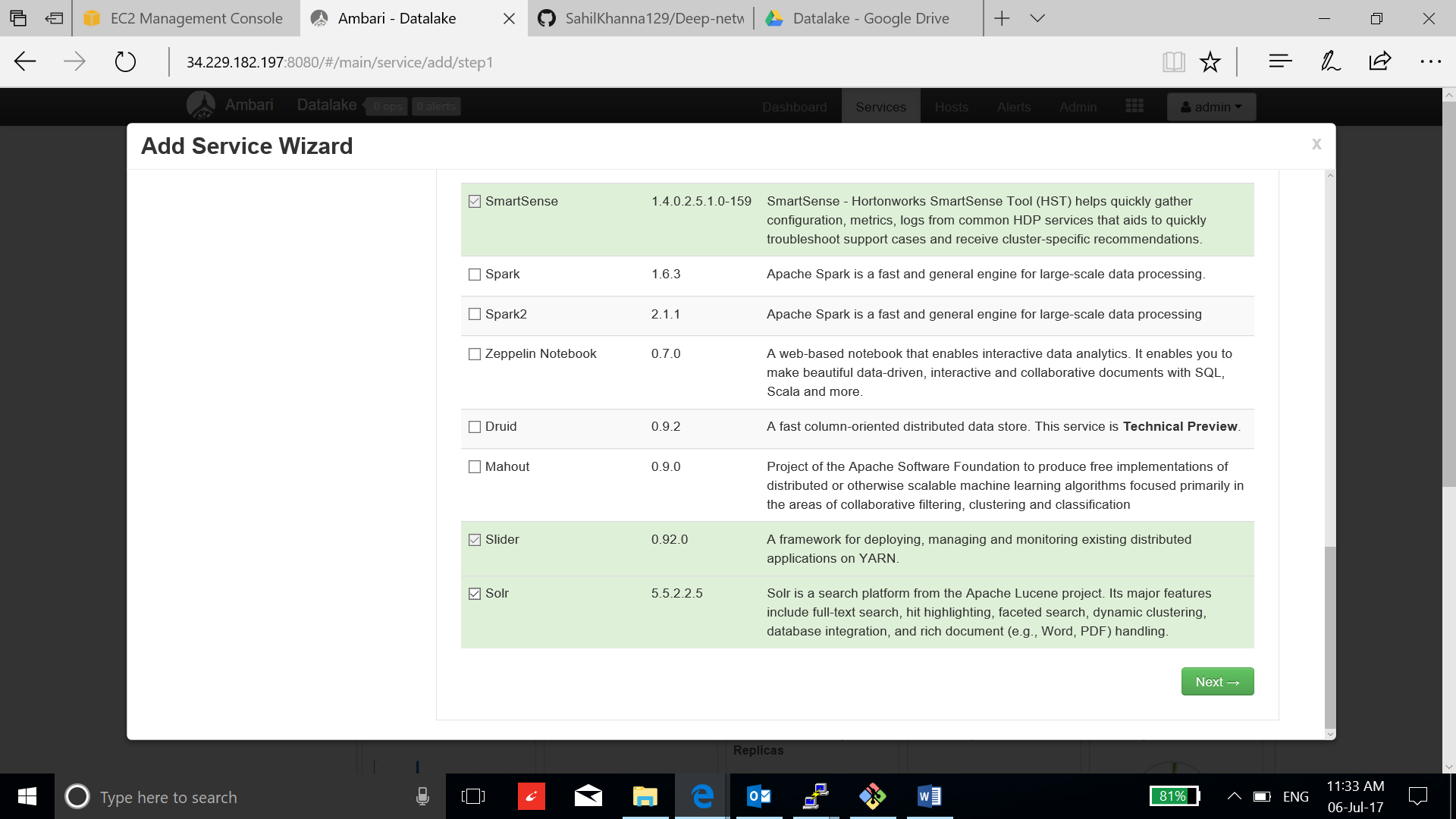
1. Edit the file: /var/lib/ambari-server/resources/stacks/HDP/2.6/repos/repoinfo.xml, add the HDP-SOLR-2.5-100: <http://public-repo-1.hortonworks.com/HDP-SOLR-2.5-100/repos/centos6>



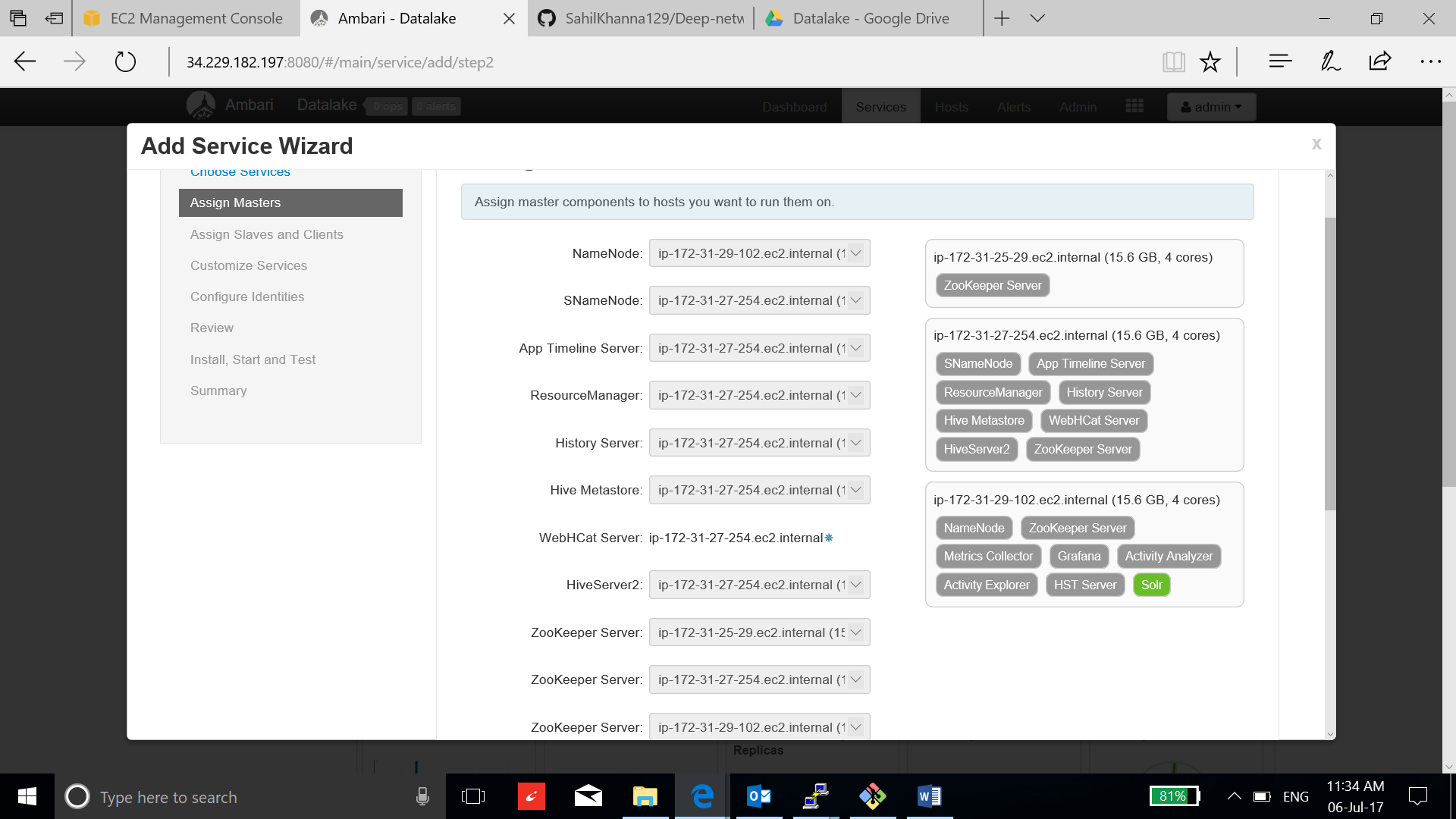
1. Restart the Ambari Server, to recognize the new repository and the management pack:

**ambari-server restart**

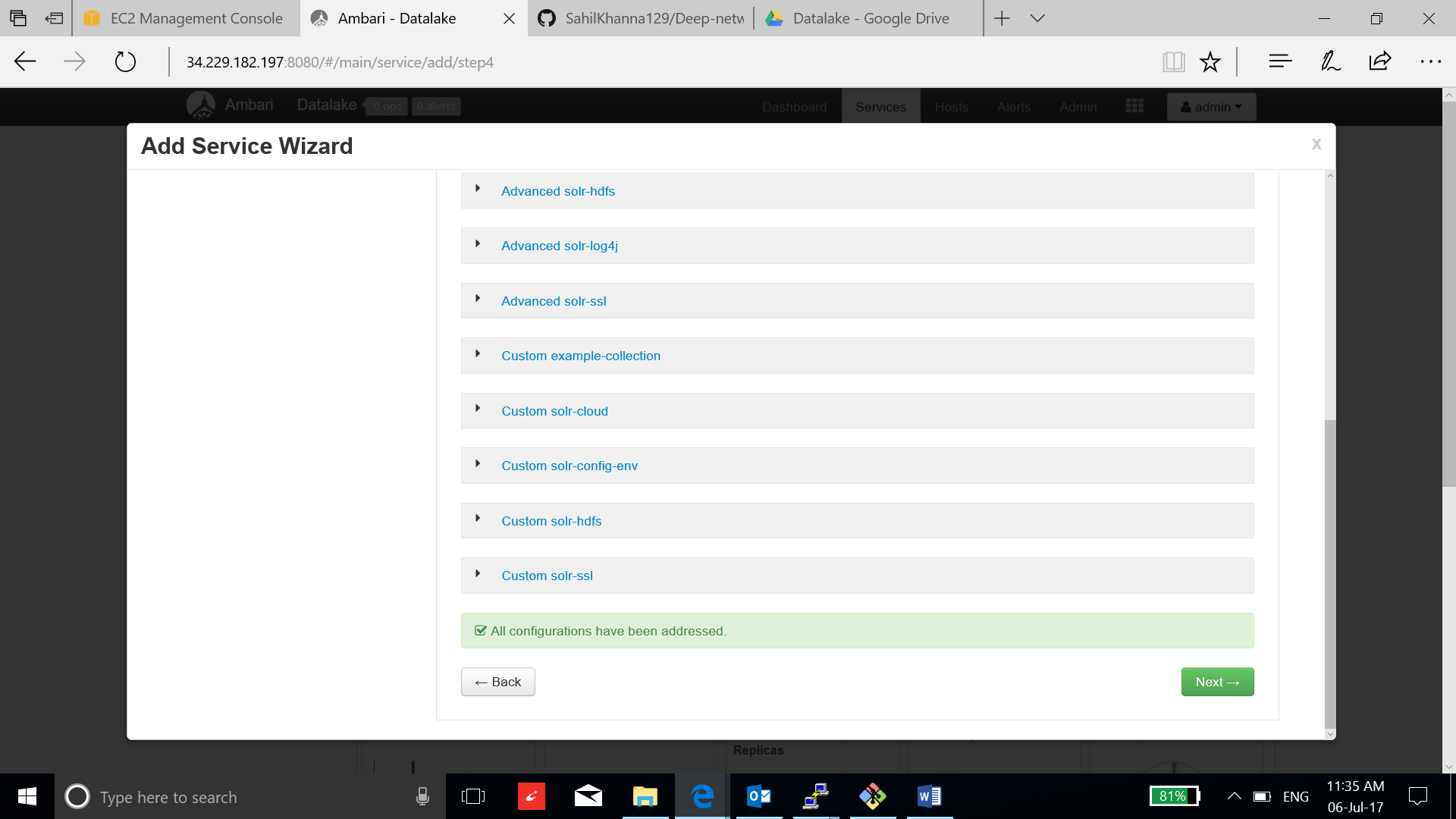
1. Now go to the Ambari and select add service to add solr:



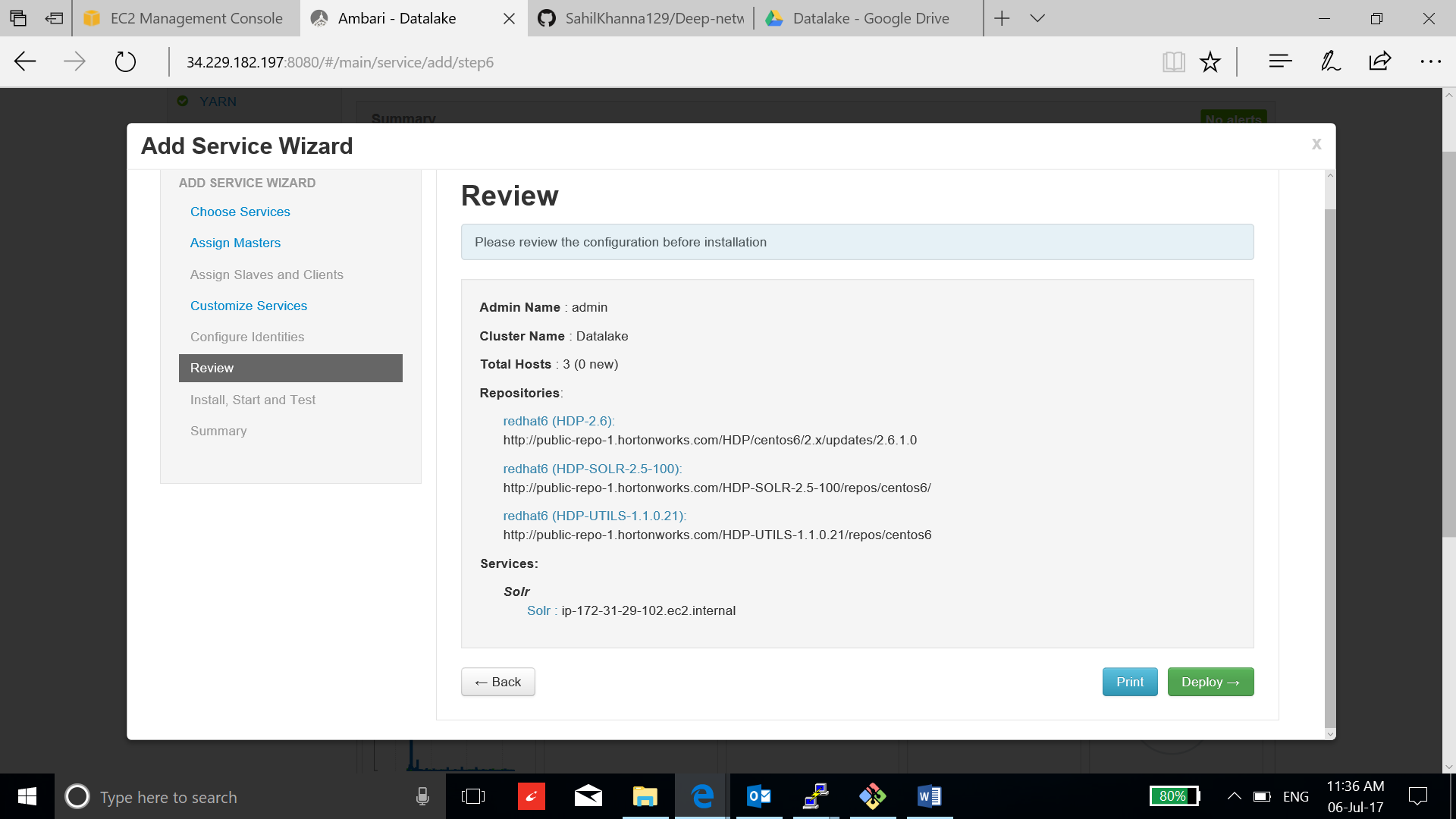
1. Appoint Solr master to the instance and click next:



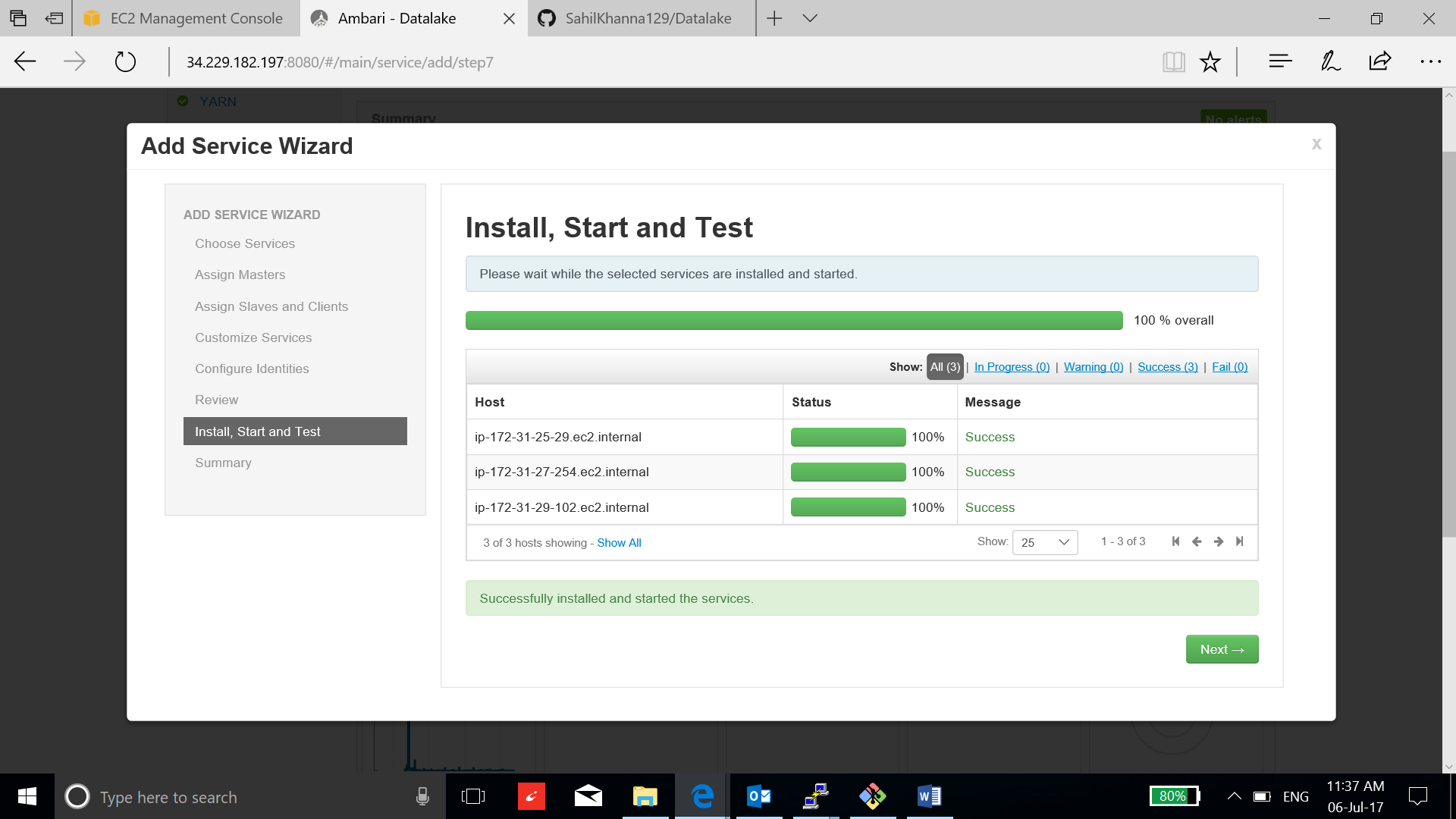
1. Keep the default configuration and click next:



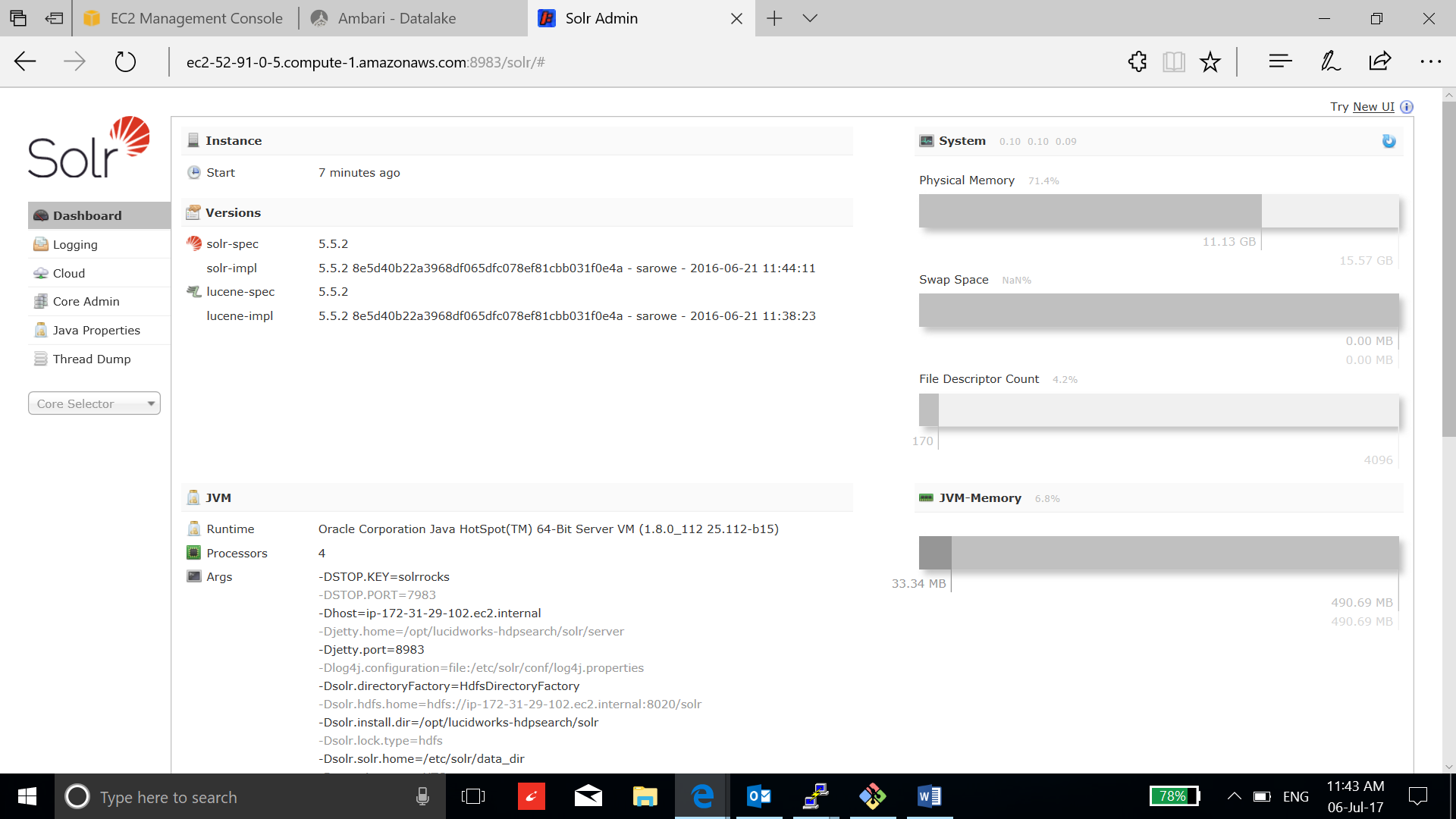
1. Review and click deploy:



1. After the installation is complete, click next and complete the process:



1. Go to Solr dashboard to check the installation:

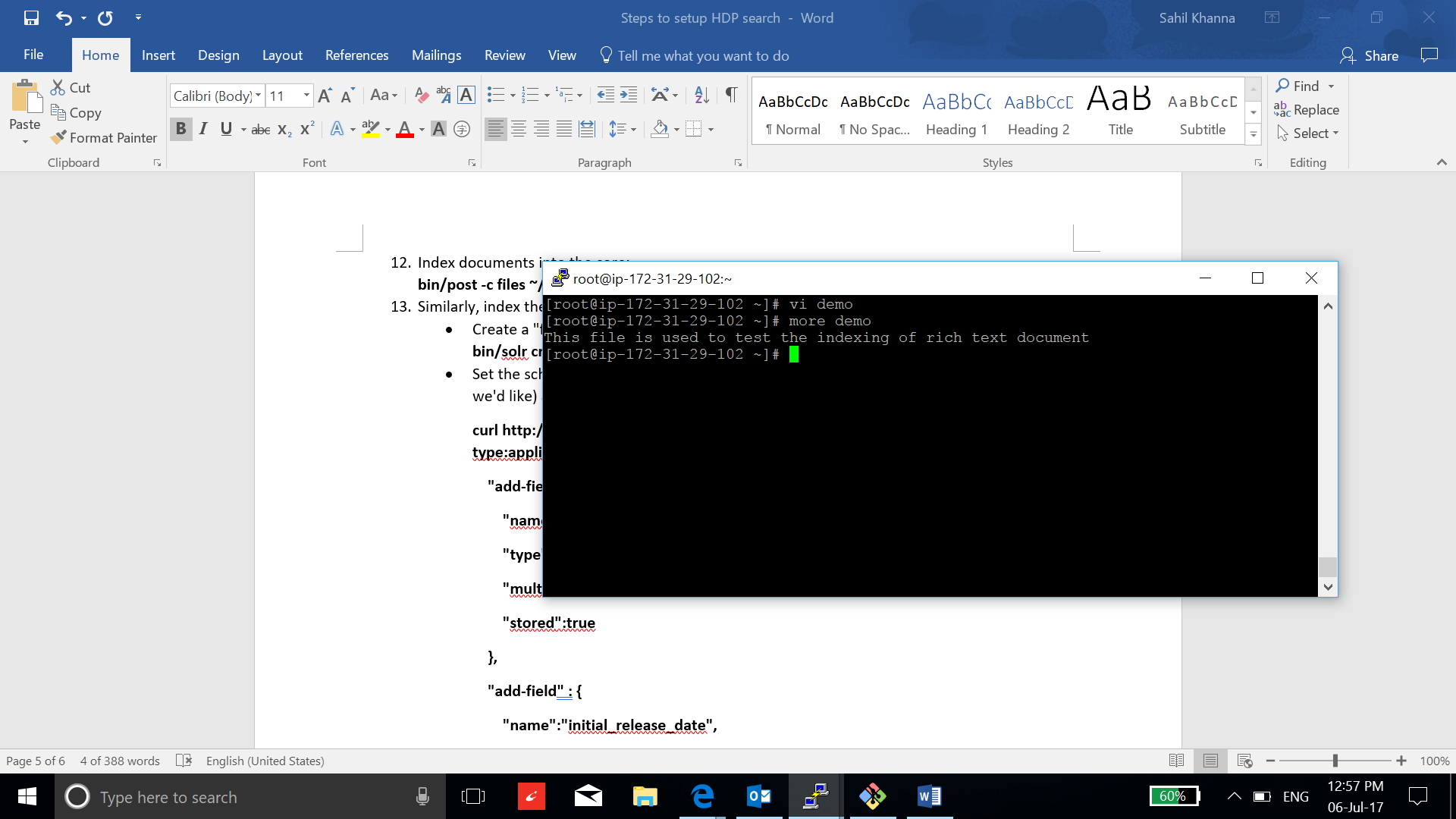


1. Use Solr on the sample data provided with the installation to get started, go to the /opt/lucidworks-hdpsearch/solr/ directory in the Solr host and create a core “files”:

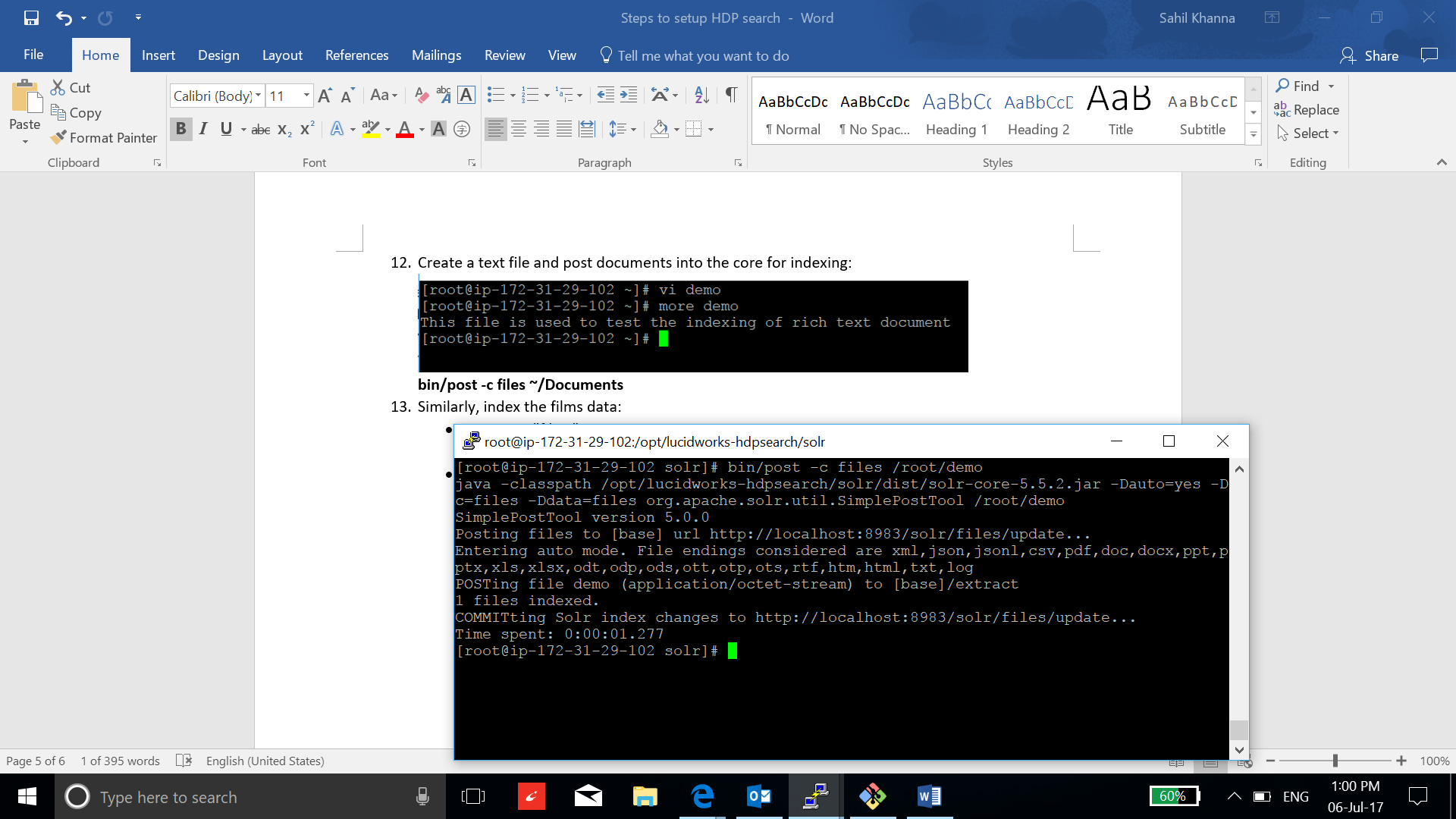
**bin/solr create -c files -d example/files/conf**



1. Create a text file and post documents into the core for indexing:

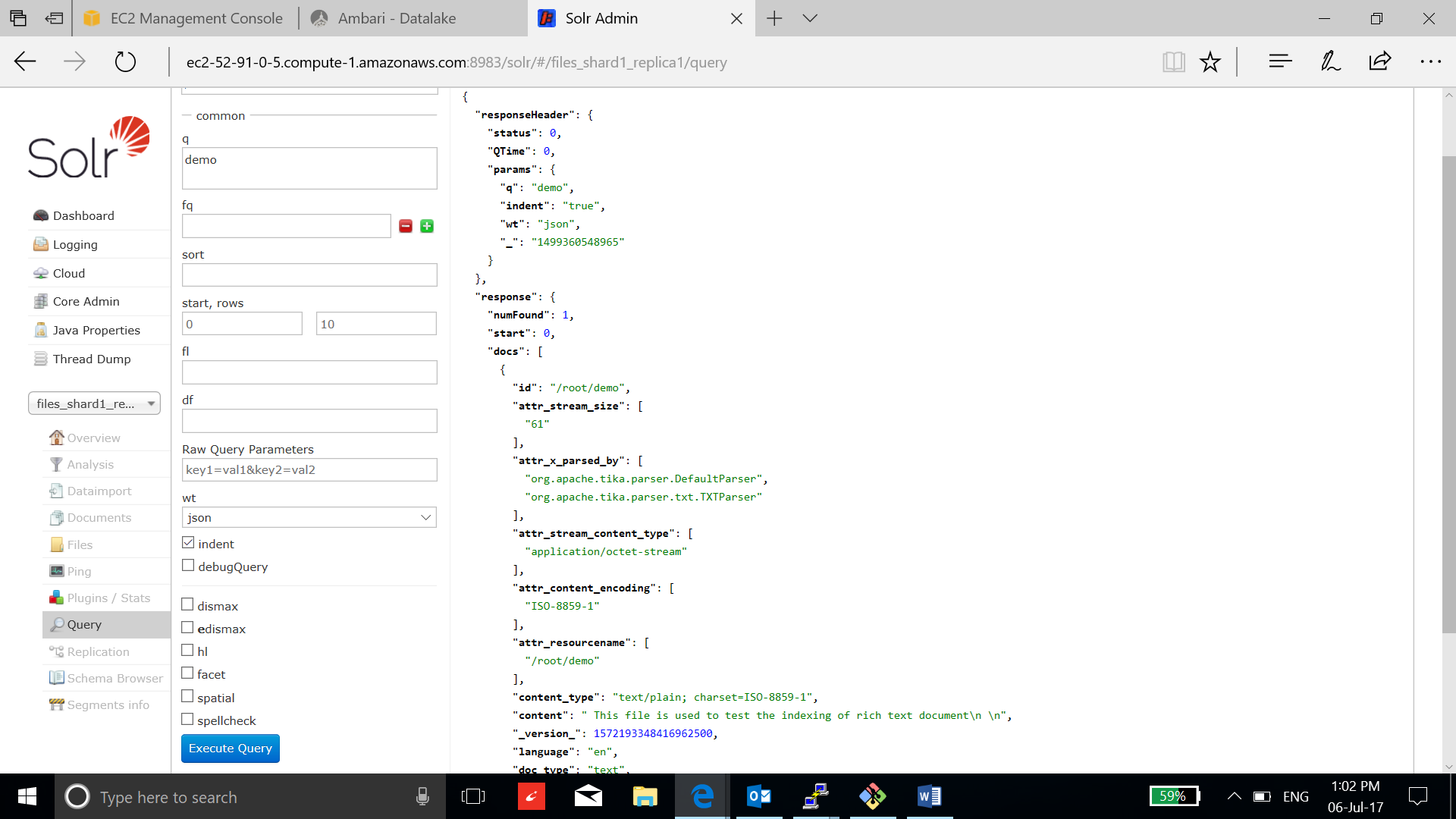


**bin/post -c files /root/demo**



1. Verify the indexing:

* Go to Solr dashboard.
* Select the collection.
* Go to query
* Search any key



1. Similarly, index the films data:

* Create a "films" core:

**bin/solr create -c films**



* Set the schema on a couple of fields that Solr would otherwise guess differently:

**curl http://localhost:8983/solr/films/schema -X POST -H 'Content-type:application/json' --data-binary '{**

**"add-field": {**

**"name":"name",**

**"type":"text\_general",**

**"multiValued":false,**

**"stored":true**

**},**

**"add-field": {**

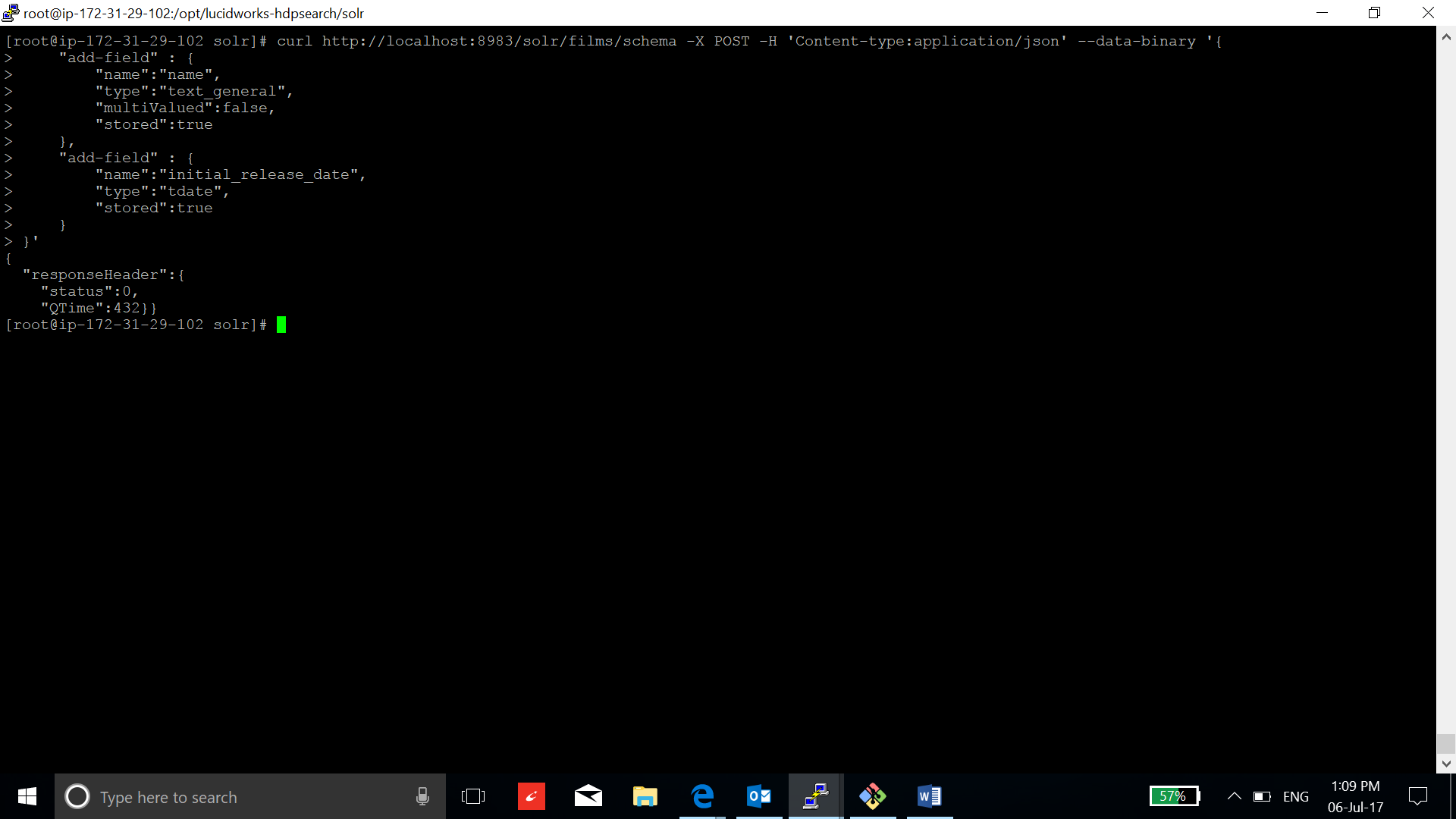
**"name":"initial\_release\_date",**

**"type":"tdate",**

**"stored":true**

**}**

**}'**



* Now let's index the data, using one of these three commands:

**- JSON: bin/post -c films example/films/films.json**

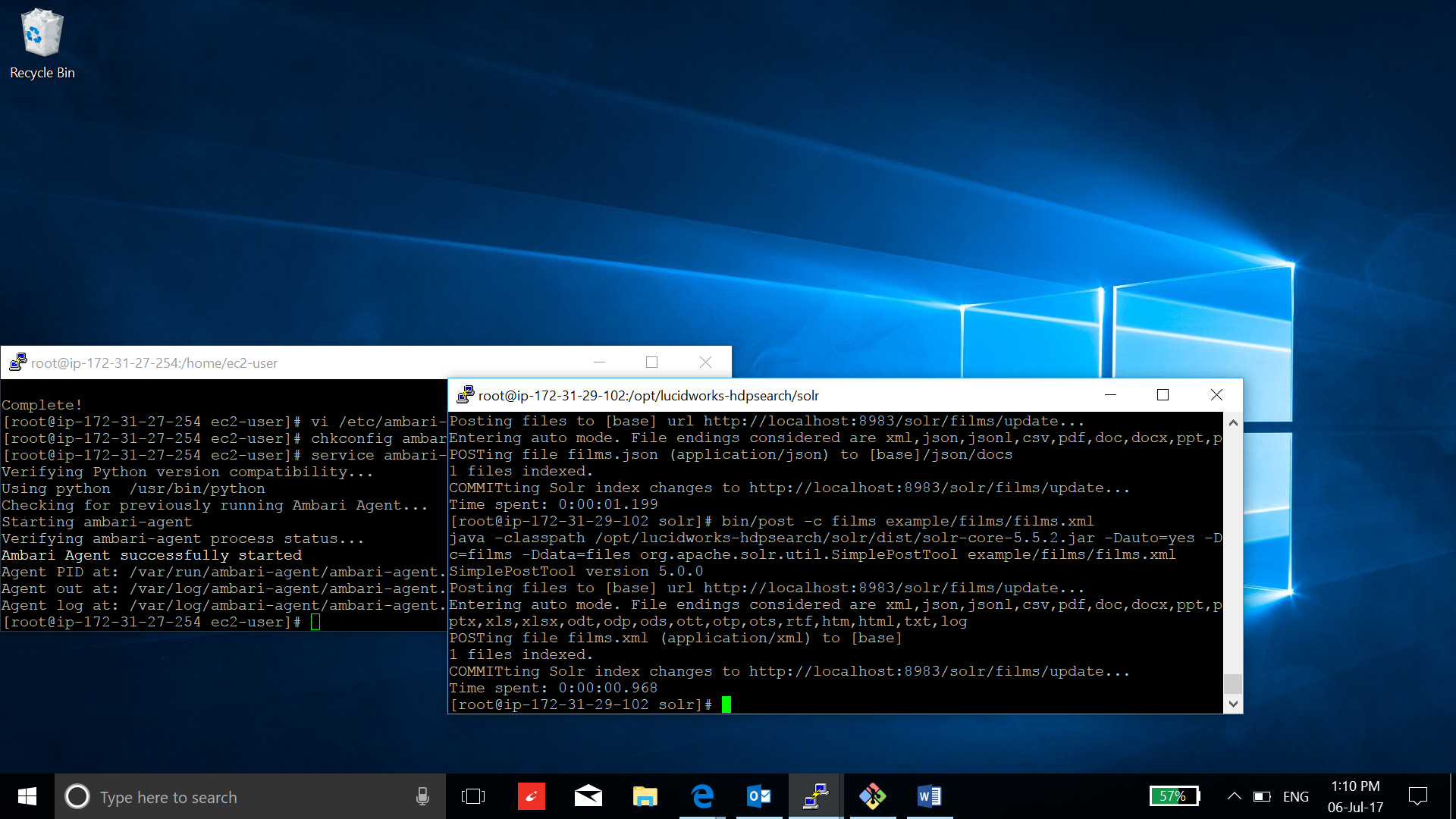
**- XML: bin/post -c films example/films/films.xml**

**- CSV: bin/post \**

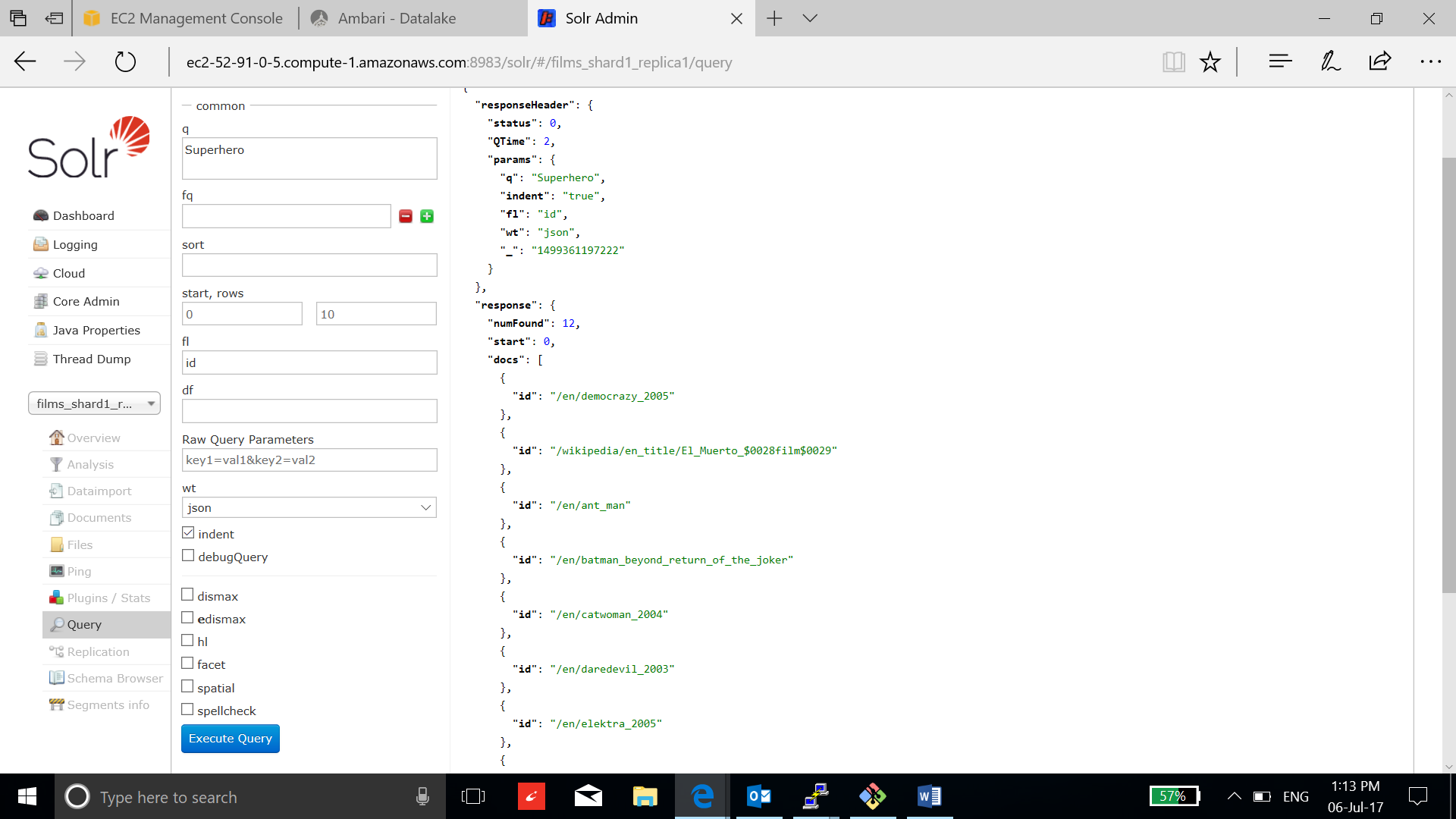
**-c films \**

**example/films/films.csv \**

**-params "f.genre.split=true&f.directed\_by.split=true&f.genre.separator=|&f.directed\_by.separator=|"**



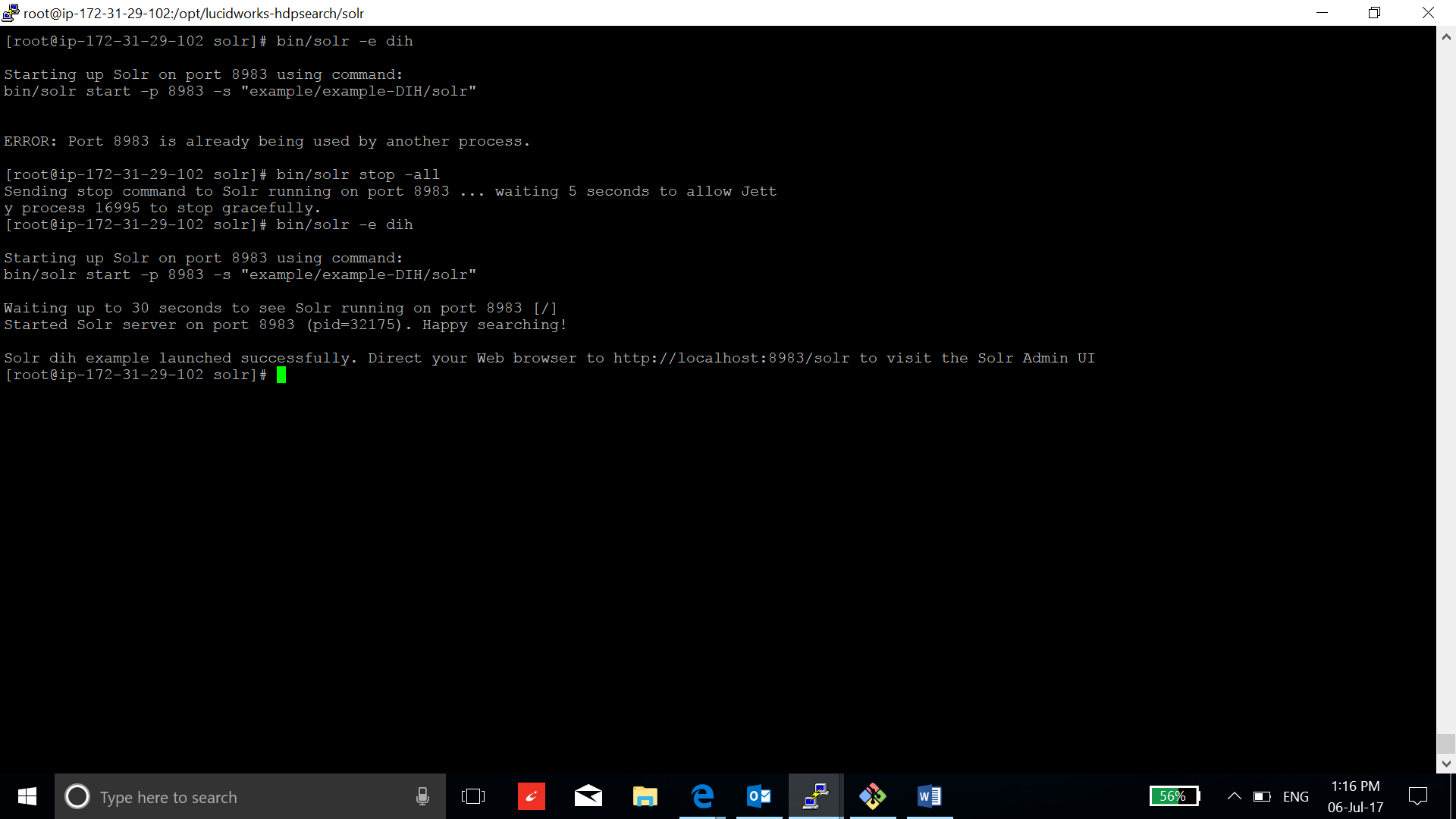
1. Run a query to verify:



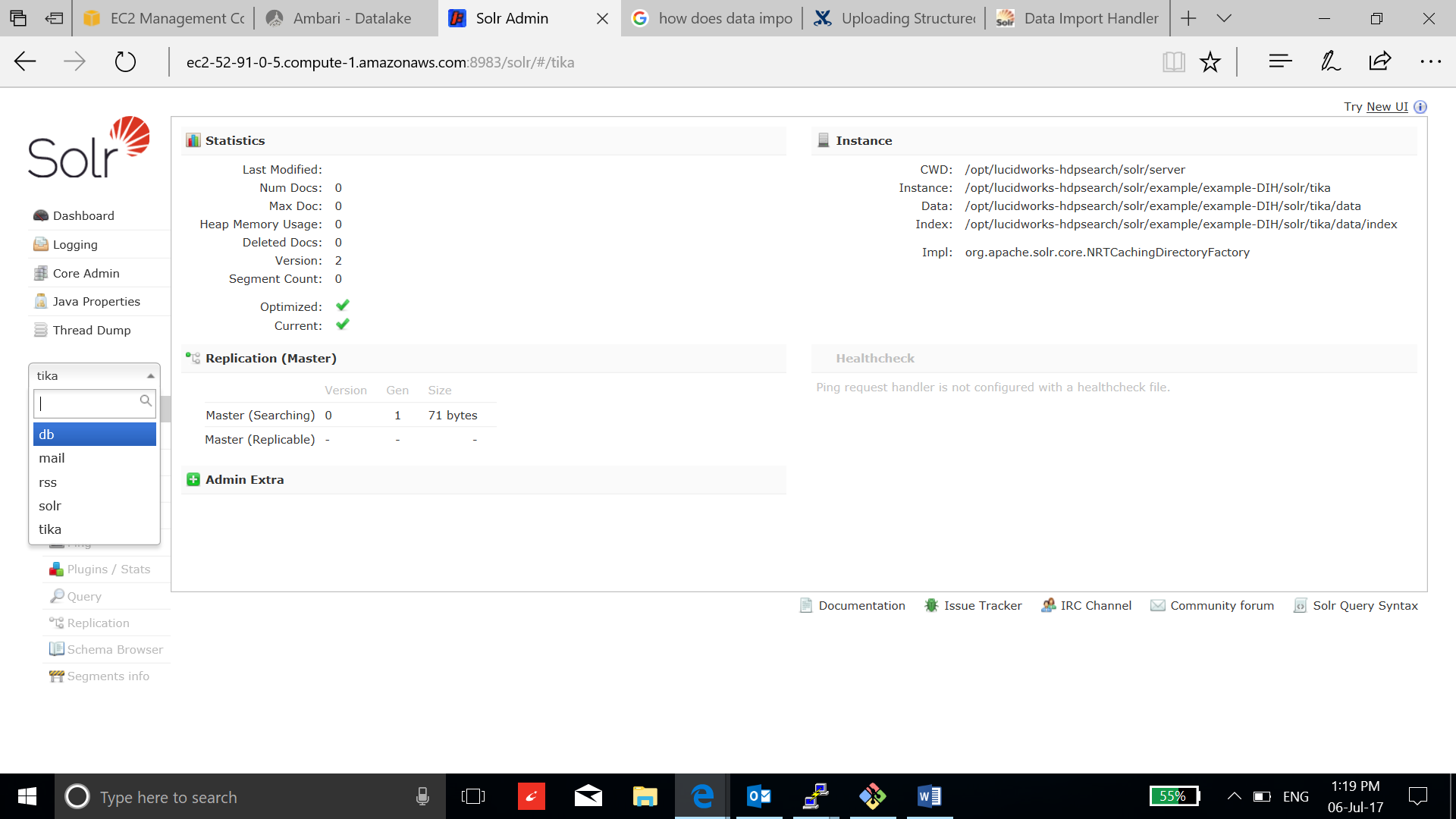
1. Run the Solr Data Import Handler example, do:

**bin/solr stop -all**

**bin/solr -e dih**

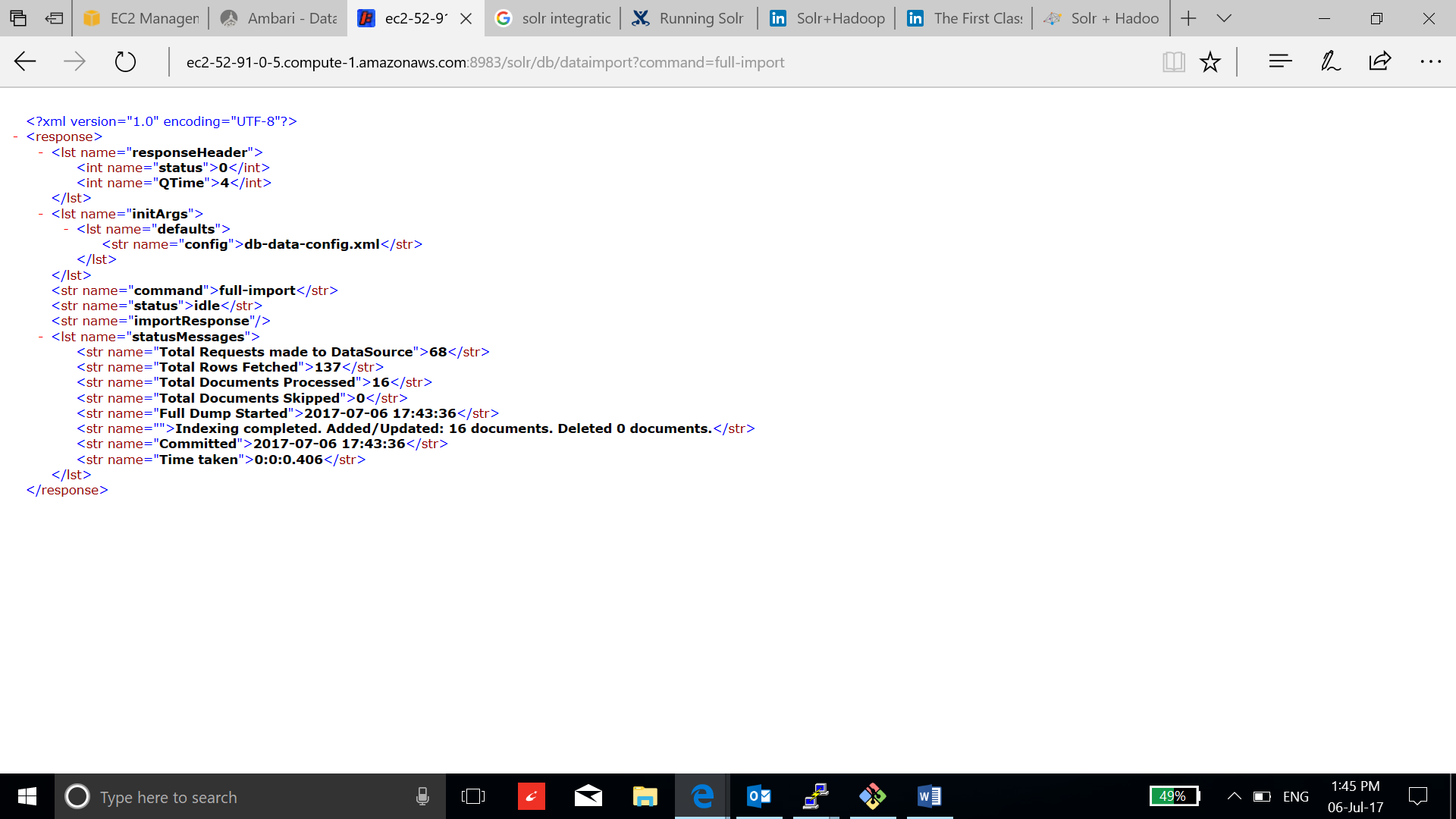


1. Go to Solr dashboard:



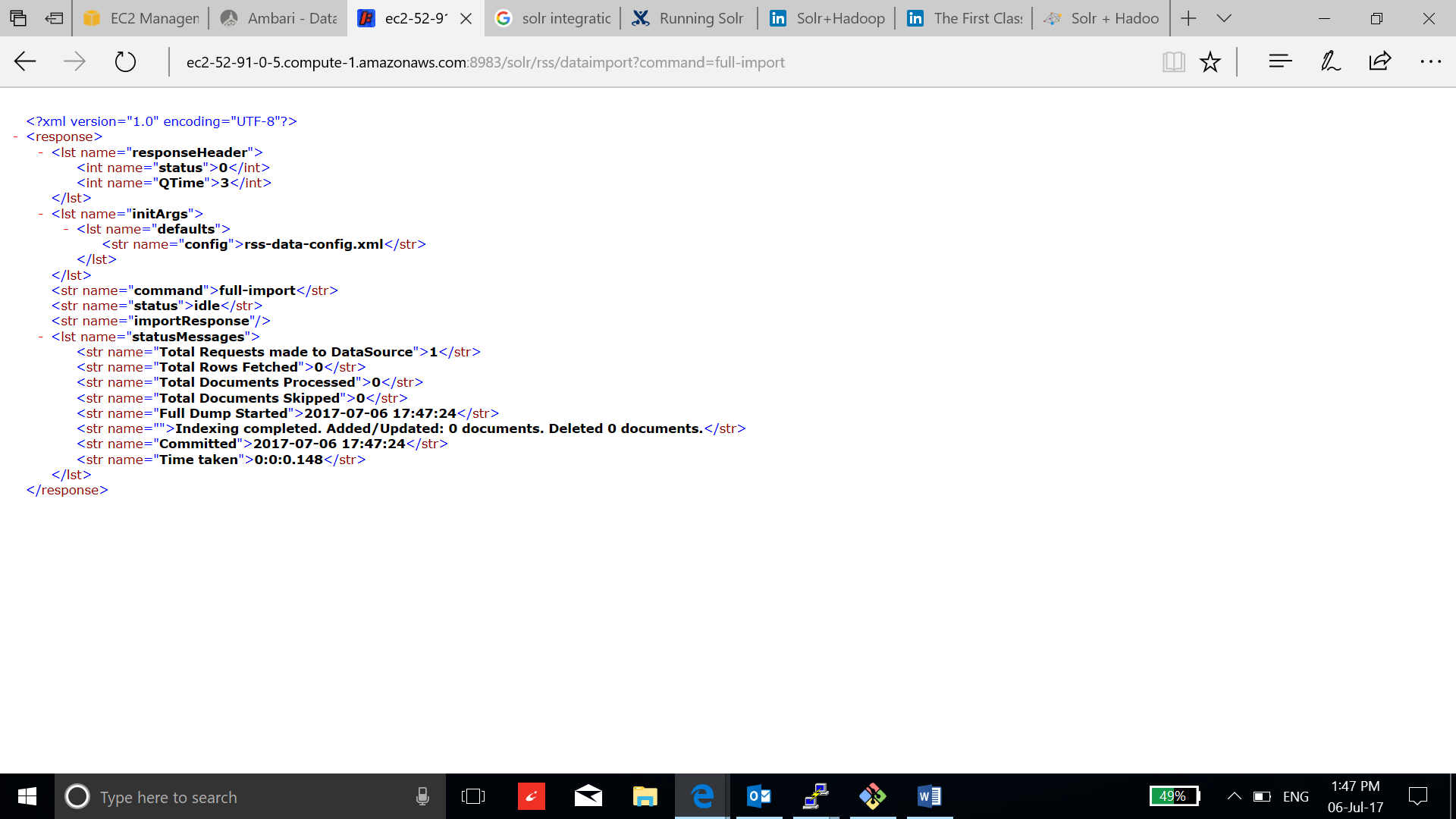
* To import data from the hsqldb database, connect to:

http://<public-DNS>:8983/solr/db/dataimport?command=full-import



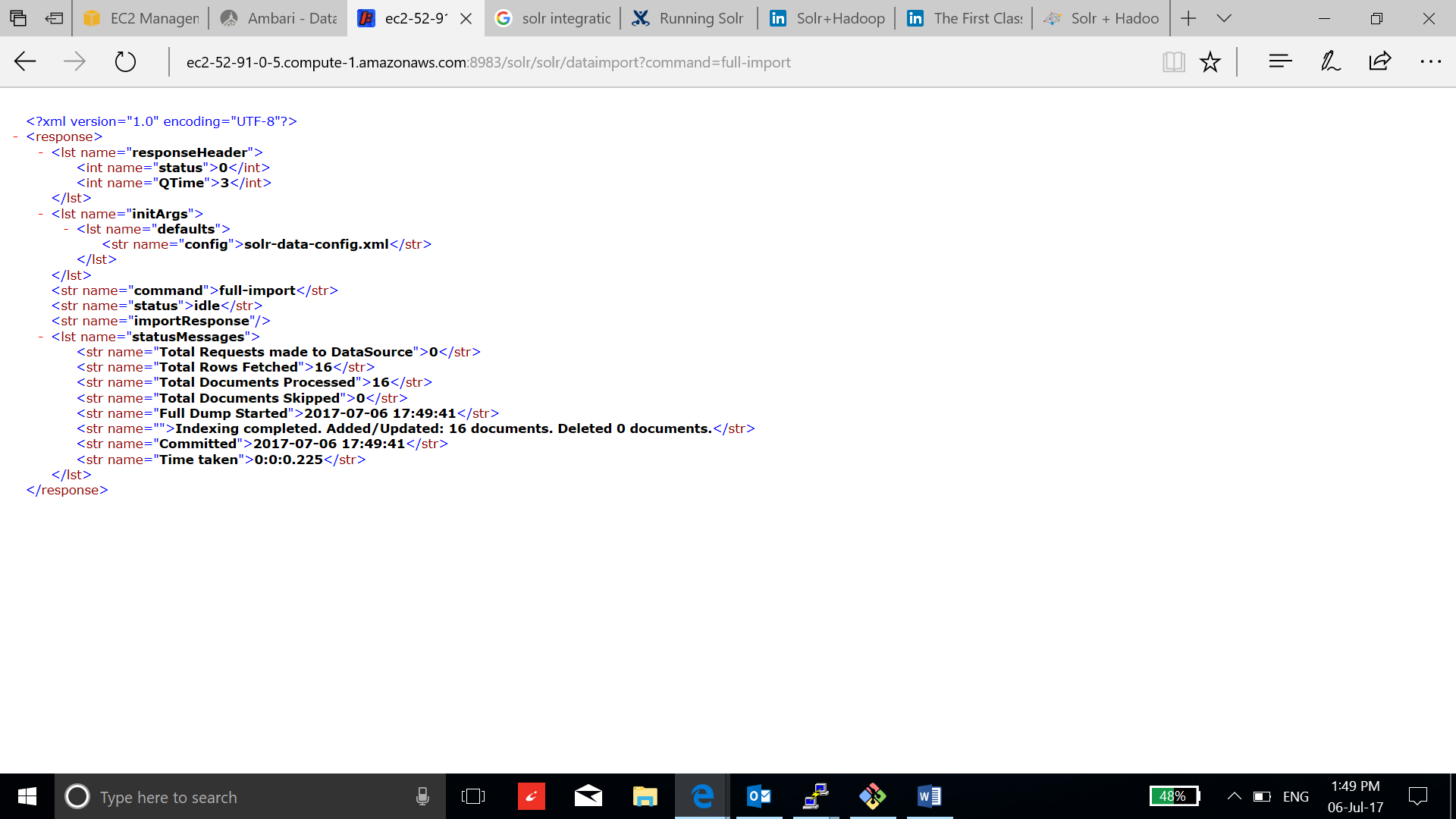
* To import data from an RSS feed, connect to:

http://<public-DNS>:8983/solr/rss/dataimport?command=full-import



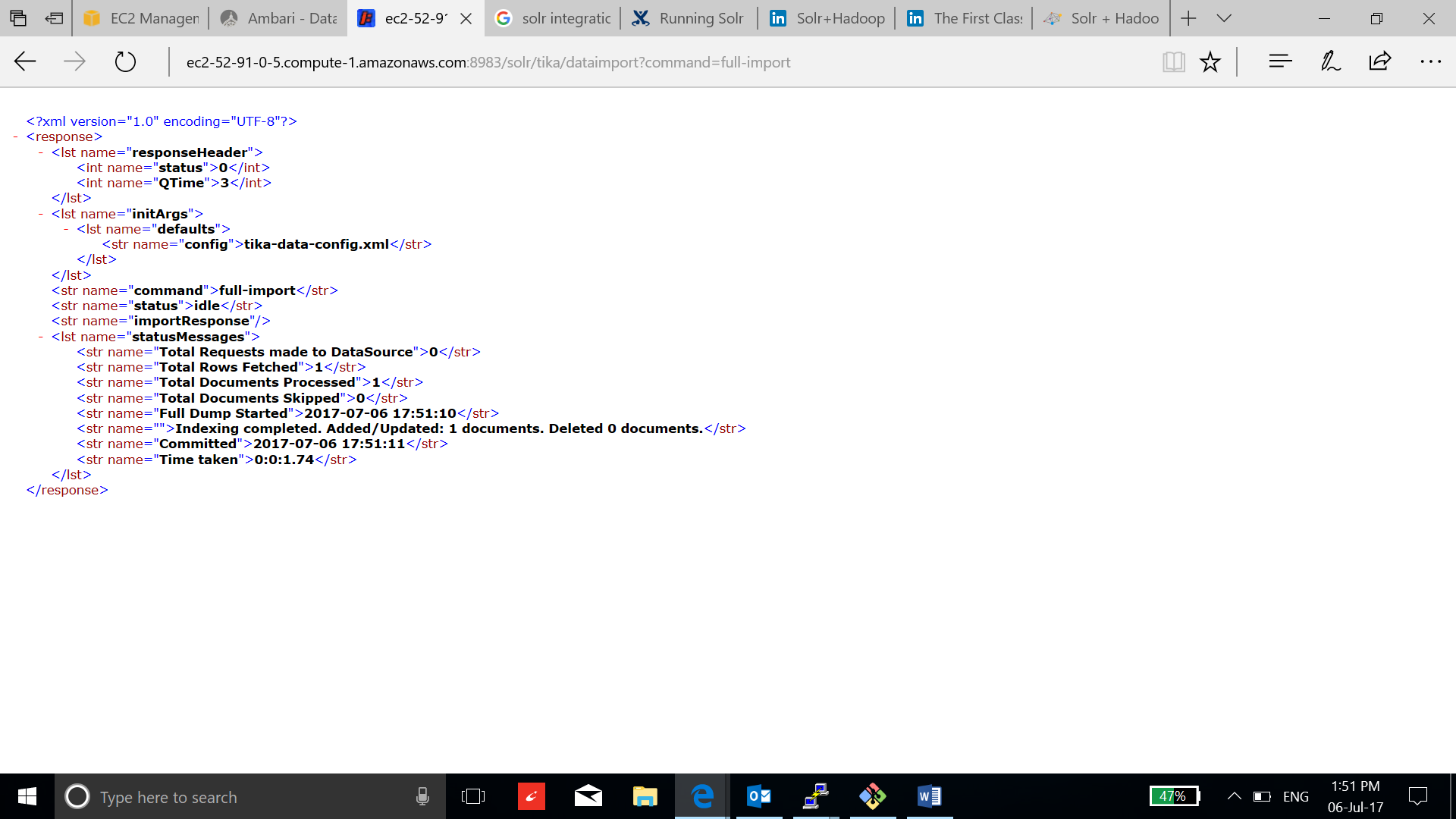
* To copy data from db Solr core, connect to:

http://<public-DNS>:8983/solr/solr/dataimport?command=full-import

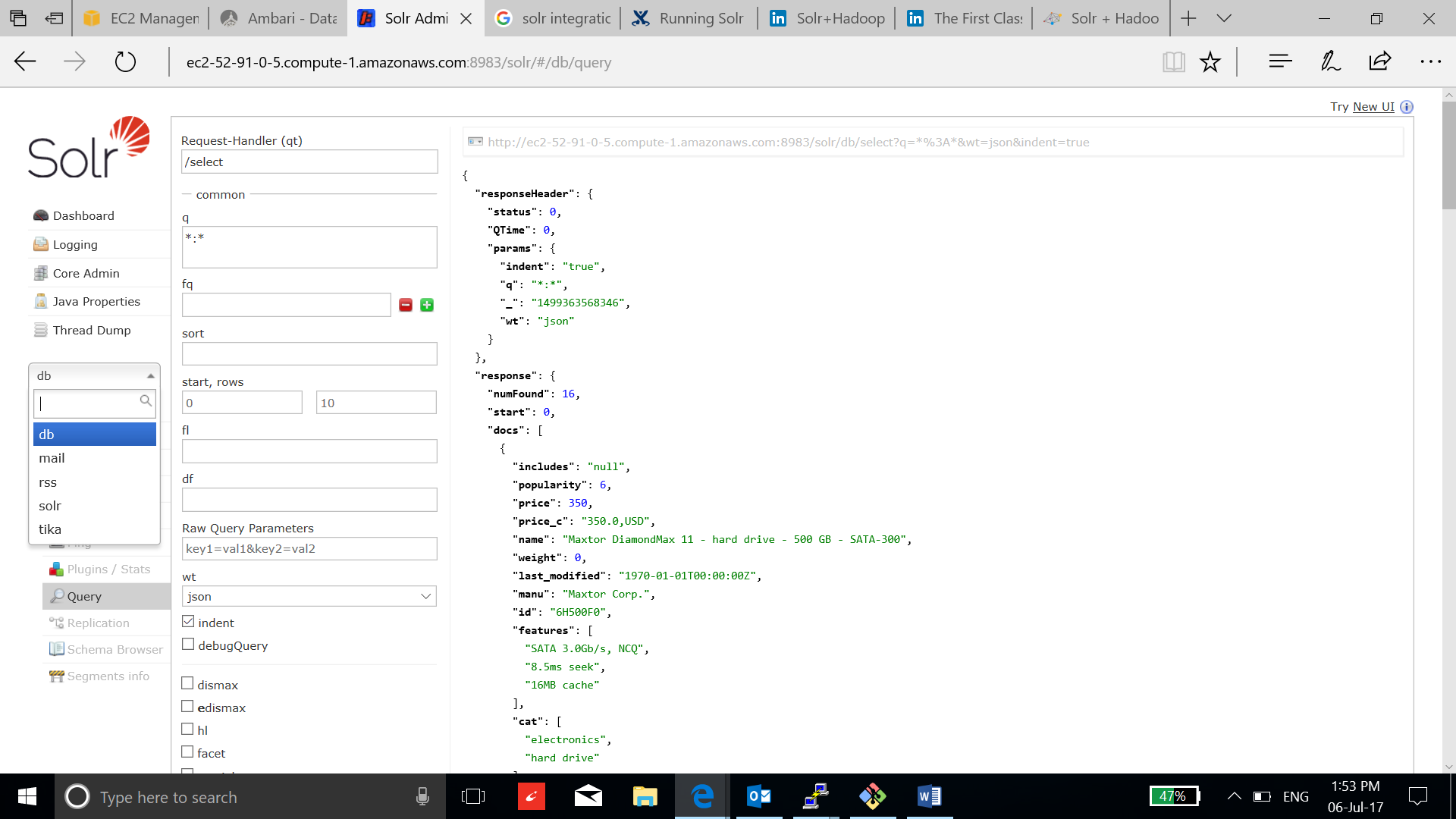


* To index a full text document using Tika integration:

http://<public-DNS>:8983/solr/tika/dataimport?command=full-import



1. You can verify the import file on the Solr Dashboard:



1. CSV file indexing in HDFS.

hadoop jar /opt/lucidworks-hdpsearch/job/solr-hadoop-job-2.2.5.jar com.lucidworks.hadoop.ingest.IngestJob -DcsvFieldMapping=0=flight\_date,1=airplane\_id,2=carrier, -DcsvFirstLineComment -DidField=id -DcsvDelimiter="," -Dlww.commit.on.close=true -cls com.lucidworks.hadoop.ingest.CSVIngestMapper -c demo -i processed\_data/flight\_data/\* -of com.lucidworks.hadoop.io.LWMapRedOutputFormat -s http://34.229.45.91:8983/solr