DIGIT.D1 – Big Data PoC

DORIS

D03.01 Application Deployment

everis Spain S.L.U

Table of Contents

Figures

Tables

No se encuentran elementos de tabla de ilustraciones.

1 PURPOSE

The purpose of this document is to define the actions to be performed as well as the commands that need to be thrown to carry out the deployment of the DORIS System and all the components needed in the Ubuntu operating system.

For this purpose, this guide will define the step-by-step installation of all the applications and frameworks necessary for the correct operation of the system.

2 PREREQUISITES

For the correct deployment of the application, following the steps in this guide, the following are required:

- A version of Ubuntu 14.04 or compatible
- Administrator rights for Ubuntu to be able to execute "sudo" commands
- An EU Login user account
- Access to the commission network: network 1
- A username and password for EUSurvey access
- Make sure that the Tomcat user, once installed, has “sudo” rights to be able to execute Solr restart commands

3 DORIS DEPLOYMENT

To carry out the DORIS deployment, we will need to install the following components:

- MongoDB 3.2
- JDK7/JRE8
- Solr 6.0.0
- Apache Tomcat 7
- FTP Server

3.1 Install Mongo DB

3.1.1 **Importing the public key**

In this step, we will import the MongoDB GPG public key.

MongoDB is already included in Ubuntu package repositories, but the official MongoDB repository provides the most up-to-date version and is the recommended way of installing the software. Ubuntu ensures the authenticity of software packages by verifying that they are signed with GPG keys, so we first have to import the key for the official MongoDB repository.

```
➢ sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80
  --recv EA312927
```

After successfully importing the key you will see:

```
➢ gpg: Total number processed: 1
➢ gpg: imported: 1  (RSA: 1)
```

3.1.2 **Creating a list file**

Next, we have to add the MongoDB repository details so the APT (Advanced Packaging Tool) will know where to download the packages from.
Issue the following command to create a list file for MongoDB.

```
➢ echo "deb http://repo.mongodb.org/apt/ubuntu
  precise/mongodb-org/3.2 multiverse" | sudo tee
  /etc/apt/sources.list.d/mongodb-org-3.2.list
```

After adding the repository details, we need to update the packages list.

```
➢ sudo apt-get update
```

3.1.3 Installing and verifying MongoDB

Now we can install the Mongo DB package itself with the following command:

```
➢ sudo apt-get install -y mongodb-org
```

This command will install several packages containing the latest stable version of MongoDB along with helpful management tools for the MongoDB server.

After package installation, MongoDB will be started automatically. You can check this by running the following command:

```
➢ sudo service mongod status
```

If MongoDB is running, you will see an output like this (with a different process ID):

```
➢ mongod start/running, process 2245
```

You can also stop, start, and restart MongoDB using the service command (e.g. service mongod stop, service mongod start).

3.1.4 Creating consultation database

To create the necessary mongo consultation database, you need to follow the next few steps:

Access the mongo interface using the following command:

```
➢ mongo
```

Check if a consultation database has already been created using the following command:

```
➢ show dbs
```

If you do not see the consultation database in the list, you need to create it.

Rather than explicitly creating a database, MongoDB will create a database when it first has data saved to it. In order to save data, you need to connect to tour new database, even though it doesn't exist yet. It sounds odd, but it is really easy. Simply enter this command into the Mongo shell:

```
➢ use consultation
```

The next step is to insert a record into this database to create it, for example:

```
➢ db.countries.save({country:"England",countryCode:"EN"})
```

Check again to see if a consultation database has already been created using the

aforementioned command.

3.2 Install JRE/JDK 8

3.2.1 Checking and installing JRE7/JDK8

Before you begin, make sure the server is fully updated with the following command:

```
➢ sudo apt-get update
```

After your server has been fully updated, you will need to know what version of Java is currently installed or, if it is not installed, with the following command:

```
➢ java -version
```

Once you have verified if Java is installed or not, choose the type of Java installation that you want, in our case JRE8 and JDK8 with the following commands:

```
➢ sudo apt-get install python-software-properties
➢ sudo add-apt-repository ppa:webupd8team/java
➢ sudo apt-get update
➢ sudo apt-get install oracle-java8-installer
```

3.2.2 Verifying Java installation

To verify if Java is installed or not use the following command:

```
➢ java -version
```

After successful installation, you will see the following input:

```
➢ java version "1.8.0_131"
➢ Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
➢ Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed
  mode)
```

If you have multiple versions of Java installed on your server, then you can select a default version. Check your alternatives using the following command:

```
➢ sudo update-alternatives --config java
```

Once you have viewed your alternatives, choose the version that you want by selecting the assigned number, and then hit enter. In our case, Java version 8.

3.3 Install Solr

To be able to install Solr in our system, we need to have Java installed (See 4.2 Install JRE/JDK 8).

3.3.1 Downloading Solr 6.0.0

To download and unzip Solr version 6.0.0, use the following commands:

```
➢ wget https://archive.apache.org/dist/lucene/solr/6.0.0/solr-6.0.0.tgz
➢ tar xzf solr-6.0.0.tgz solr-6.0.0/bin/install_solr_service.sh --strip-components=2
```

3.3.2 Installing Solr as a service

Once you have downloaded and unzipped the Solr service installer file, run the shell script file to install Solr as a service using the following command:

```
➢ ./install_solr_service.sh solr-5.2.1.tgz
```

You will see the following message:

```
➢ Waiting to see Solr listening on port 8983 [/]
➢ Started Solr server on port 8983 (pid=XXXXX). Happy
  searching!
```

Solr will be started automatically. To confirm it has started, run:

```
➢ sudo service solr status
```

It is very important that the folder in which Solr cores are allowed has the correct rights, so as to allow Tomcat users to write in new Solr cores.

The default path for the Solr cores in this document is: /var/solr/data

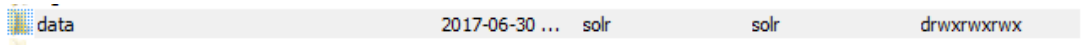

Figure 1 - Solr cores rights

You need to deploy the following two cores in the default path:

default_core

default_core.zip

default_core_attachment

default_core_attachment.zip

3.4 Install Apache Tomcat 7

3.4.1 Installing Tomcat 7

Before you begin, make sure that the server is fully updated with the following command:

```
➢ sudo apt-get update
```

Now you are ready to install Tomcat. Run the following command to start the installation:

```
➢ sudo apt-get install tomcat7
```

Answer yes to the prompt to install tomcat. This will install Tomcat and its dependencies, such as Java, and it will also create the Tomcat 7 user. It also starts Tomcat with its default settings.

3.4.2 Changing Tomcat 7 configuration

Once installed, you need to make a quick change to the Java options that Tomcat uses when it starts. Open the Tomcat 7 parameters file:

```
➢ sudo nano /etc/default/tomcat7
```

Find the JAVA_OPTS line and replace it with the following. Feel free to change the Xmx and MaxPermSize values – these settings affect how much memory Tomcat will use:

```
➢ JAVA_OPTS="-Djava.awt.headless=true -Xmx512m -
  XX:MaxPermSize=256m -XX:+UseConcMarkSweepGC"
```

Save and exit.

3.4.3 Restarting Tomcat

Restart Tomcat using this command

```
➢ sudo service tomcat7 restart
```

Tomcat will be started at its default port: 8080.

To verify Tomcat is running, you can access the default splash page by going to your domain or IP address followed by :8080 in a web browser

```
➢ http://server_IP_address:8080
```

Then, the Tomcat welcome page will be shown:

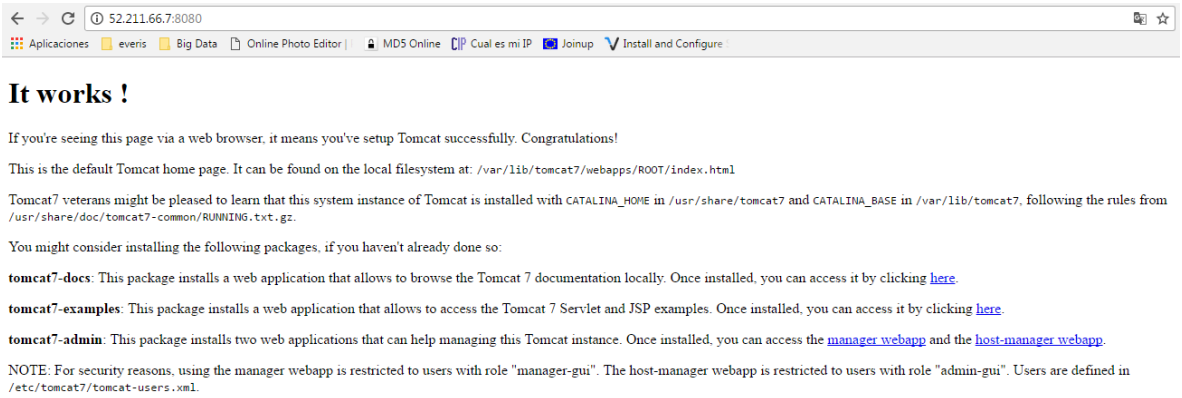

Figure 2 - Tomcat welcome page

3.4.4 Configure Tomcat web management interface

With the following command, you can install the Tomcat web interface (Manager webapp)

```
➢ sudo apt-get install tomcat7-admin
```

In order to use the Manager webapp once installed, you must add a login to your Tomcat server. You can do this by editing the **tomcat-users.xml** file:

```
➢ sudo nano /etc/tomcat7/tomcat-users.xml
```

This file is filled with comments which describe how to configure the file. You need to insert the following lines between <tomcat-users>…</tomcat-users>, the result will be:

```
<tomcat-users>
   <user username="your_user" password="your_password"
   roles="manager-gui,admin-gui"/>
</tomcat-users>
```

Save and quit the tomcat-users.xml file. To put your changes into effect, restart the Tomcat service (See 4.4.3 Restarting tomcat).

Take a look at the Web Application Manager, accessible via the link or http://server_IP_address:8080/manager/html and type your username and password to log in. The following page will be appear:

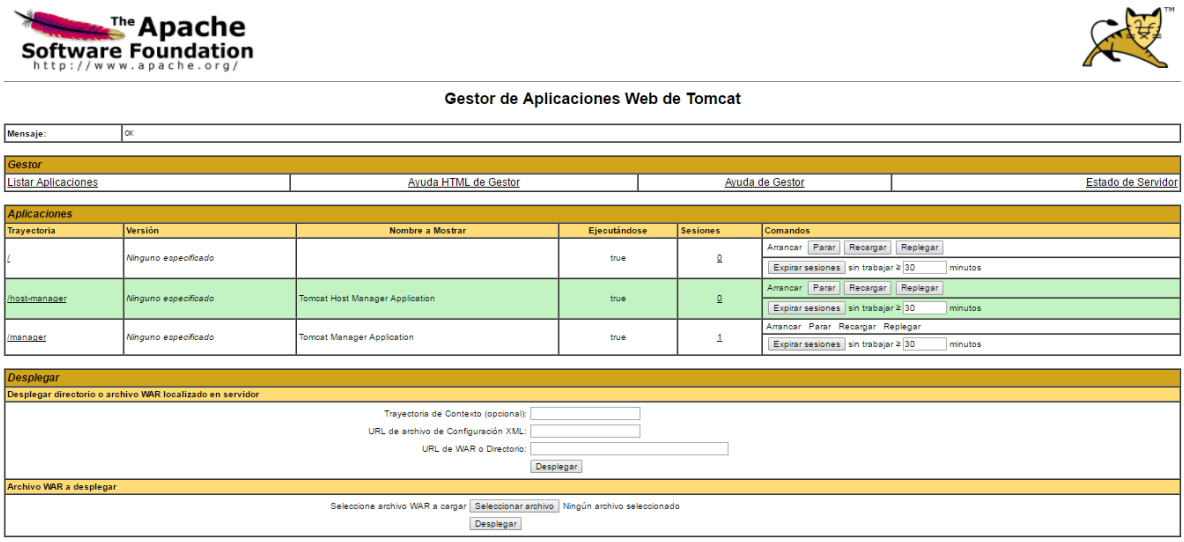

Figure 3 - Tomcat Manager web page

3.5 Deploy DORIS Importer and DORIS Board in Tomcat

To be able to install Solr in your system, you need to have Tomcat installed (See 4.4 Install Apache Tomcat 7).

3.5.1 Deploying war files

There are three war files you want to install in the Tomcat server:

- doris-importer-api.war
- doris-importer-ui.war
- dorisBoard.war

To install these war files in Tomcat, you must copy these war files into /var/lib/tomcat7/webapps/ directory. To do so, use the following commands:

```
➢ sudo cp origin_doris-importer-ui.war_file
  /var/lib/tomcat7/webapps/doris-importer-ui.war
➢ sudo cp origin_doris-importer-api.war_file
  /var/lib/tomcat7/webapps/doris-importer-api.war
➢ sudo cp origin_dorisBoard.war_file
  /var/lib/tomcat7/webapps/dorisBoard.war
```

Then, restart Tomcat to deploy the applications (See 4.4.3 Restarting tomcat).

3.5.2 Configuring environment

You must configure several elements to adapt the application to your work environment. This configurations are:

- For DORIS importer:
    - To create an EUSurvey web service user with the necessary permissions and configure it in DORIS importer
    - To modify DORIS configuration files with the correct URL and paths from our system
    - To create necessary folders in our file system and provide it with permissions to write/modify
    - To create of an FTP Server which is needed to make it possible to translate the position papers
- For DORIS board:
    - Configure Tomcat with the ECAS module

3.5.2.1 *Modifying DORIS configuration files*

There are two files that you need to configure with your paths and URL. These files are:

In DORIS importer API project:

- doris-importer.properties

In DORIS importer UI project:

- config.js

3.5.2.1.1 **Modifying Doris-importer.properties**

Once the DORIS system is deployed in the Tomcat server, you need to make several changes to adapt the default configuration to your environment.

The default deployment path is: ${CATALINA_HOME}/webapps/

In this folder, there will be three folders corresponding to the three projects that correspond to the three applications that make up the DORIS system.

The folder that contains the file you need to modify is: doris-importer-api.

This folder contains a file named: doris-importer.properties.

By editing this file, you will see the following information which explains which properties need to be modified:

```
################################
# The next values of this variables need to be modified in the properties bellow
# ${host}                          --> The IP in which Doris System is deployed
# ${importer.api.path}             --> Root path to access to doris-importer-api application, for example: http://${host}:${tomcat_port}/doris-importer-api
# ${tomcat_port}                   --> Listening tomcat port, for example: 8080
# ${solr.root.directory}           --> The solr instalation root folder
# ${solr.restart.sentence}         --> Sentence to restart solr service, for example: sudo service solr restart
# ${cores.path}                    --> Solr cores folder, for example: /var/solr/data/
# ${excel.source.path}             --> Default folder to store the excel importation sources, for example: /home/ubuntu/doristmp/ManualExcelData/
# ${postion.papers.folder.path}    --> Default folder to store the position papers once translated, for example: /home/ubuntu/doristmp/positionPapers/translated/
# ${postion.papers.original.folder.path} --> Default folder to store the original position papers, for example: /home/ubuntu/doristmp/positionPapers/original/
#
# ${ftp.url}                       --> FTP url access, for example: ftp://username:password@${host}/  Replacing username, password and ${host}
#
# ${mongodb.hostname}              --> Mongo DB host name --> Usually localhost
# ${mongodb.port}                  --> Mongo DB connection port. Default port is 27017
# ${mongodb.user}                  --> User to connect to mongo db
# ${mongodb.pass}                  --> Password to connect to mongo db
#
# ${eusurvey.result.zipped}        --> Folder to obtain cosultation data zip file, for example: /home/ubuntu/doristmp/EUSurveyData/dataZipped
# ${eusurvey.result.unzipped}      --> Folder in which consultation data is unzipped, for example: /home/ubuntu/doristmp/EUSurveyData/dataUnzipped
# ${eusurvey.user.account}         --> Username to connect to EUSurvey webservice
# ${eusurvey.password.account}     --> Password to connect to EUSurvey webservice
#
# ${proxy.required}                --> true/false when the system uses proxy.
# ${proxy.host}                    --> proxy IP when required is true
#
# ${board.path}                    --> Root path to access to dorisBoard application, for example http://${host}:${tomcat_port}/dorisBoard
#
# ${similarity.endpoint}           --> Root path from similarity webservice, for example: http://localhost:5000/consultationsSimilarity
#
# ${analytics.endpoint}            --> Root path from analytics webservice, for example: http://localhost:5000/consultationsEnricher
################################
```

Figure 4 - Properties configuration

3.5.2.1.2 **Modifying config.js**

This file is located in the following path:

${CATALINA_HOME}/webapps/doris-importer-ui/src/main/webapp/app/

In this file there are three variables which need to be changed:

```
➢ var DORIS_API_URL = "http://<app_host>:<app_port>/doris-importer-api/";
➢ var DORIS_UI_URL = "http://<app_host>:<app_port>/doris-importer-ui/";
➢ var DORIS_BOARD_URL = "http://<localhost>:<app_port>/dorisBoard/";
```

In these three variables, you need to change the application host to your application host, and the application port to your application port.

3.5.2.2 *Creating necessary folders*

The DORIS system requires several previously created folders to download position papers and files from the EUSurvey web service. The path for these folders will be configured into the configuration files explained in the previous section of this document. The following commands should be used to create the folders and the rights needed to allow Tomcat users to access these folders.

For position papers:

```
➢ sudo mkdir -p /home/ubuntu/doristmp/positionPapers/original
➢ sudo mkdir -p /home/ubuntu/doristmp/positionPapers/translated
```

For manual Excel data extraction mode:

```
➢ sudo mkdir -p /home/ubuntu/doristmp/ManualExcelData
```

For EUSurvey data extraction mode:

```
➢ sudo mkdir -p /home/ubuntu/doristmp/EUSurveyData
```

Make permissions for these directories (use –R to set permissions recursively):

```
➢ sudo chmod 777 -R /home/ubuntu/doristmp/
```

Also, the DORIS system needs to create new Solr cores, so, you need to set the right permissions for the folder containing the cores. In this guide, the default Solr cores folder is /var/solr. With the following command you can set all permissions for this folder recursively.

```
➢ sudo chmod 777 -R /var/solr
```

Another alternative is to set the Tomcat default user and the Solr default user as the same user, to allow cores created in Tomcat to be accessed from Solr.

3.5.2.3 *Creating FTP server*

This project uses an FTP server to provide file transfer between DORIS importer and Translation API. To create an FTP server you need to take the following steps.

3.5.2.3.1 **Installing vsftpd**

Start by updating your package list and installing the vsftpd daemon in your ubuntu

```
➢ sudo apt-get update
➢ sudo apt-get install vsftpd
```

When the installation is complete, copy the configuration file so you can start with a blank configuration, saving the original as a backup.

```
➢ sudo cp/etc/vsftpd.conf/etc/vsftpd.conf.orig
```

3.5.2.3.2 **Creating a user to connect to FTP**

With vsftpd installed, in the instructions that follow, create user access to the required data.

First, add a new user:

```
➢ sudo adduser ftpdoris
```

Assign a password when prompted and feel free to press "ENTER" through the other prompts.

3.5.2.3.3 **Configuring FTP access**

We are planning to allow a single user with a local shell account to connect to FTP. For this, you need to change configuration file vsftpd.conf. Start by opening the config file to verify that the settings in your configuration match those below:

- `sudo nano /etc/vsftpd.conf`

In this file you need to make several changes.

Firstly, you need to verify that anonymous access is disabled and local access is enabled, the results will be as follows:

```
...
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
...
```

Next you will need to change some values in the file. In order to allow the user to upload files, uncomment the write_enable setting so that you have:

```
write_enable=YES
```

Uncomment the chroot to prevent the FTP-connected user from accessing any files or commands outside the directory tree.

```
chroot_local_user=YES
```

Configure the local folder as root folder. This folder will be the folder than contains the position papers (original and translated).

```
local_root=<position_papers_folder>
```

Then, you have to allow web access to the service to be able to open communication with the web service response. For that, you need to add the following lines:

```
listen_address=0.0.0.0
allow_writeable_chroot=YES
pasv_address=<your_ubuntu_ip>
pasv_enable=YES
port_enable=YES
```

Since we only want to allow FTP access on a case-by-case basis, set up the configuration so that access is only given to a user when they are explicitly added to a list, rather than by default, by adding the following lines:

```
userlist_enable=YES
userlist_file=/etc/vsftpd.userlist
userlist_deny=NO
```

`userlist_deny` toggles the logic. When it is set to "YES", users on the list are denied

FTP access. When it is set to "NO", only users on the list are allowed access. When you are done making the change, save and exit the file.

Finally, create and add your user to the file. Use the -a flag to append elements to the file:

```
echo "ftpdoris" | sudo tee -a /etc/vsftpd.userlist
```

Now you are ready for testing.

```
ftp -p <your_ubuntu_ip>
```

Enter your username and password, in this case ftpdoris

```
Output
Connected to <your_ubuntu_ip>.
220 (vsFTPd 3.0.2)
Name (34.248.45.19:ubuntu): ftpdoris
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

3.6 Deploy Analytics Module

To deploy the analytics module you need to install several applications explained below.

3.6.1 **Installing python-pip**

Before you begin, make sure that the server is fully updated with the following command:

- `sudo apt-get update`

Now you are ready to install python-pip. Run the following command to start the installation:

- sudo apt-get install python-pip

Check the version of Pip that is installed:

- Pip -V

The output should look like this:

```
Output
```
- pip 9.0.1 from /usr/local/lib/python2.7/dist-packages (python 2.7)

This module will only run properly when the DORIS System is deployed on a network with access to the European Commission network: Network1.

Now you are ready to install the required Python packages.

3.6.2 **Installing Python packages**

3.6.2.1 *Installing flask-restful*

- sudo pip install flask-restful

3.6.2.2 *Installing numpy*

- sudo pip install numpy

3.6.2.3 *Installing requests*

- sudo pip install requests

3.6.2.4 *Installing unirest*

- sudo pip install unirest

3.6.2.5 *Installing unidecode*

- sudo pip install unidecode

3.6.2.6 *Installing pymongo*

- sudo pip install pymongo

3.6.2.7 *Intalling scikit-learn*

- sudo pip install scikit-learn

3.6.2.8 *Installing scipy*

- sudo pip install scipy

3.6.3 **Installing analytics module**

To install analytics module in your system, you need to insert the following two files into your file system.


public_consultations_enricher_prod.py


fingerprint.py

There are several changes to be made in public_consultations_enricher_prod.py to configure it to your URL.

Edit the file and modify the following lines:

```
dorisHost=<your_doris_ip>
dorisPort=<your_doris_port>
hdlr=logging.FileHandler('<local_logging_path>/enricher.log'
, mode='w')
```

You need to create the local folder that is referred to as "<**local_logging_path**>"

To start the analytics service, run the following command:

- FLASK_APP=<**analytics_path**>/public_consultations_enricher_prod.py flask run

The output will look like:

Output

- * Serving Flask app "public_consultations_enricher_prod"
- INFO:werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)