



RAPEX Searcher – User setup

Quick start setup guide to work with RAPEX Searcher system

27/11/2019

TABLE OF CONTENTS

1. INSTALLING REQUIRED SOFTWARE	4
1.1. OPENVPN	4
1.2. MOBAXTERM.....	6
1.3. WINDOWS REMOTE DESKTOP	9
1.4. PGADMIN	11
2. RAPEX SEARCHER EXECUTION AND DATA FLOW	14
2.1. ETL.....	15
2.2. RAPEX SEARCHER.....	16
2.2.1. <i>Searcher</i>	17
2.2.2. <i>Scraper</i>	17
2.2.3. <i>Text mining</i>	18
2.3. ELASTICSEARCH INDEXING	18
3. ACCESSING KIBANA	21

Document characteristics

Property	Value
Release date	27/11/2019
Status:	Initial version
Version:	1.0
Authors:	Everis
Reviewed by:	
Approved by:	

Document history

Version	Description	Date
1.0	Initial version	27/11/2019

1. Installing required software

This guide is made on a Windows 10 environment. In other versions some steps may change.

The necessary software to run the RAPEX Searcher are:

- OpenVPN
- MobaXterm
- Windows Remote Desktop
- pgAdmin

These technologies and their purpose will be explained in the following parts of this document.

1.1. OpenVPN

RAPEX Searcher is deployed in a private network in the AWS Amazon using a Bastion as a Jump machine, so the first requirement to connect to this network is to use a VPN.

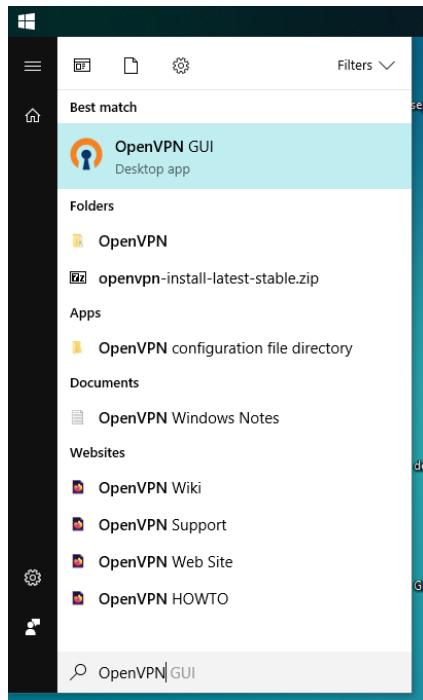
There are multiple software to make VPN connections, in this manual we will use OpenVPN software.


To install it, go to <https://openvpn.net/community-downloads/> and download OpenVPN software version fits better with your computer. Download the .ovpn file that is needed to connect to the Rapex Searcher network and put it in an easy accessible location. This file is the following:

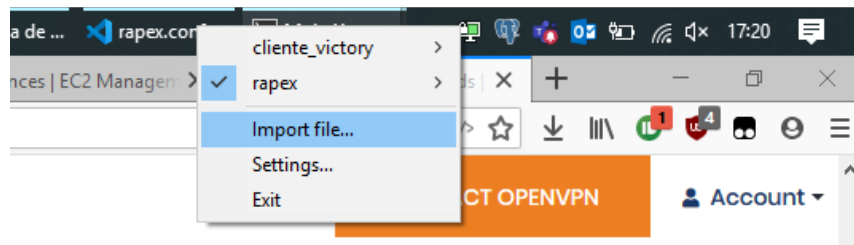


Follow the steps as any common installation. Once it is ready, it should appear in task bar. If not, search it and launch it.

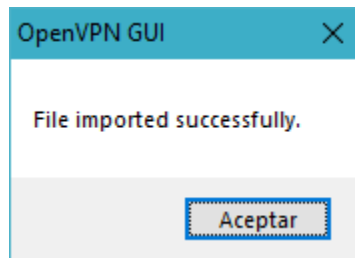
RAPEX Searcher – User setup



Now, check again if it appears in your task bar with the  icon. Right-click on it and select “Import file...”:

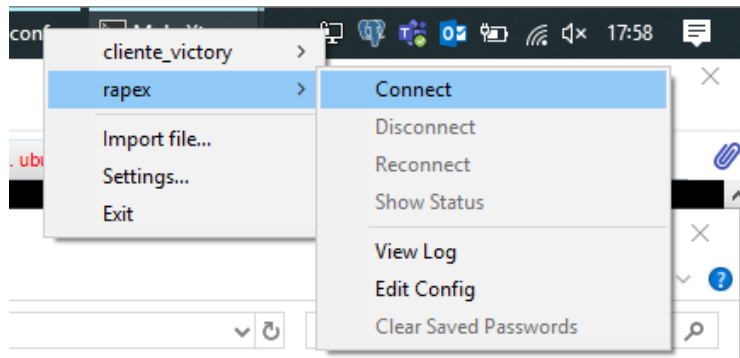


Now you will see an open dialog. Search here the file where you saved it before and open it. The program should show you the following info message:

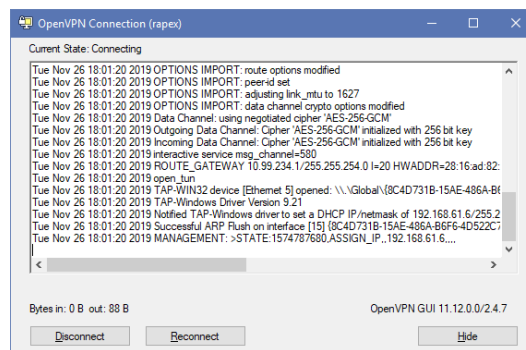



Now right-click again in OpenVPN icon and select the entry that has been added named “rapex” (or like your .ovpn filename):

RAPEX Searcher – User setup



A window showing OpenVPN log connection like the following appears.



If everything went right, the OpenVPN icon on the task bar will be green. Now, check again if it appears in your task bar with the  appearance.

1.2. MobaXterm

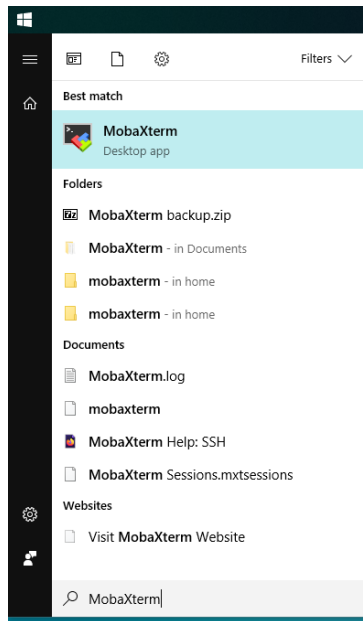
MobaXterm is a software used to stabilise SSH connections to the RAPEX Ubuntu server that contains the searcher and scrapper components and the Logstash component.

This server is deployed in the AWS EC2 service with the name: **RPXPUBBUBU01**

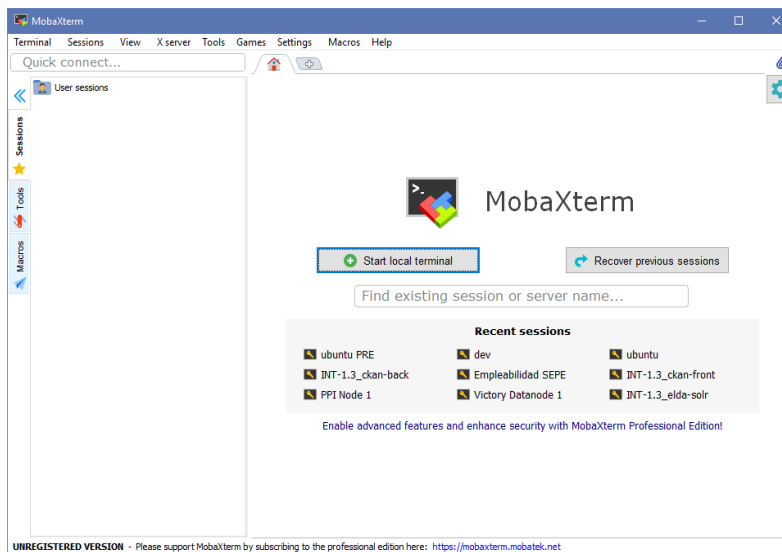
To install it, go to <https://mobaxterm.mobatek.net/download-home-edition.html> and download the version you want (portable or installable). If it is not compatible with your system you can use other SSH clients as PuTTY. This configuration can be used as reference anyway.

Once it is installed in your computer, open it.

RAPEX Searcher – User setup



You should see a screen like the following:



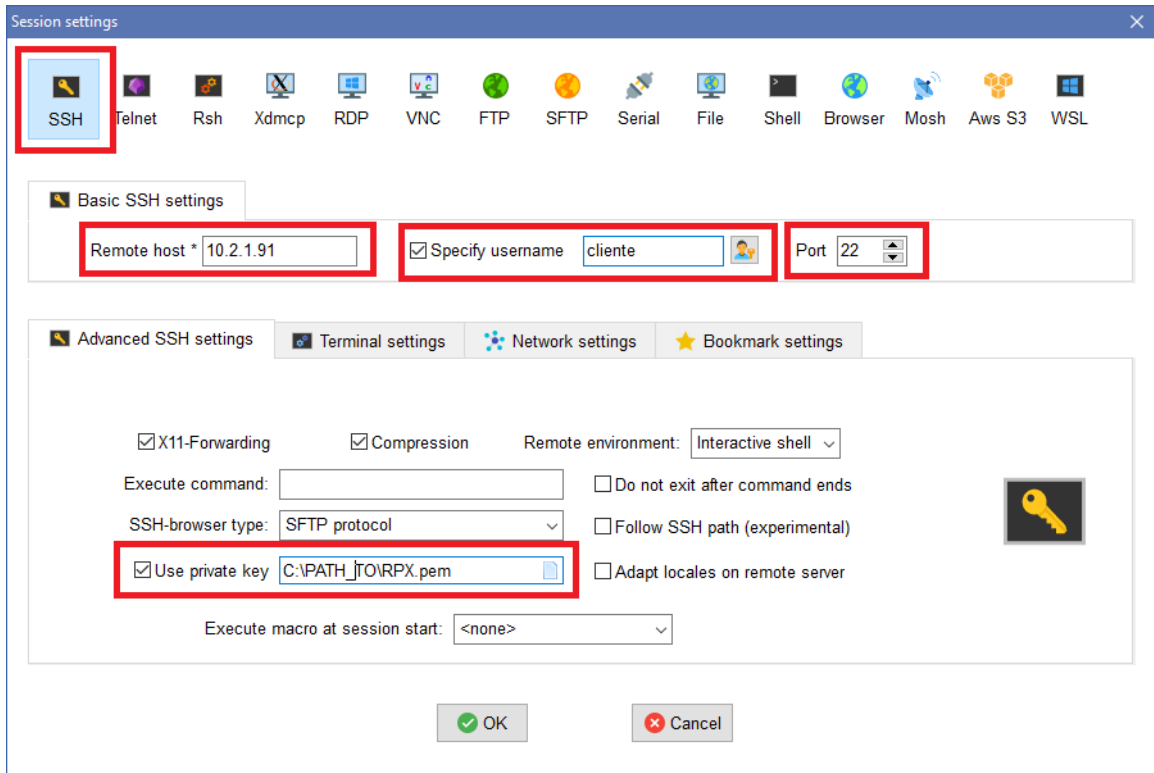
Right-click on “User sessions” and then “New session”. Select SSH option in the top left part of the window. Now we are going to configure the login settings for the Ubuntu machine that launches the RAPEX Searcher process. The information you need is the following:

- Remote Host: 10.2.1.91
- Port: 22
- Username: cliente
- Private key: You have to put here the path to the RPX.pem key in order to authenticate in the system. You can download it here:

RAPEX Searcher – User setup

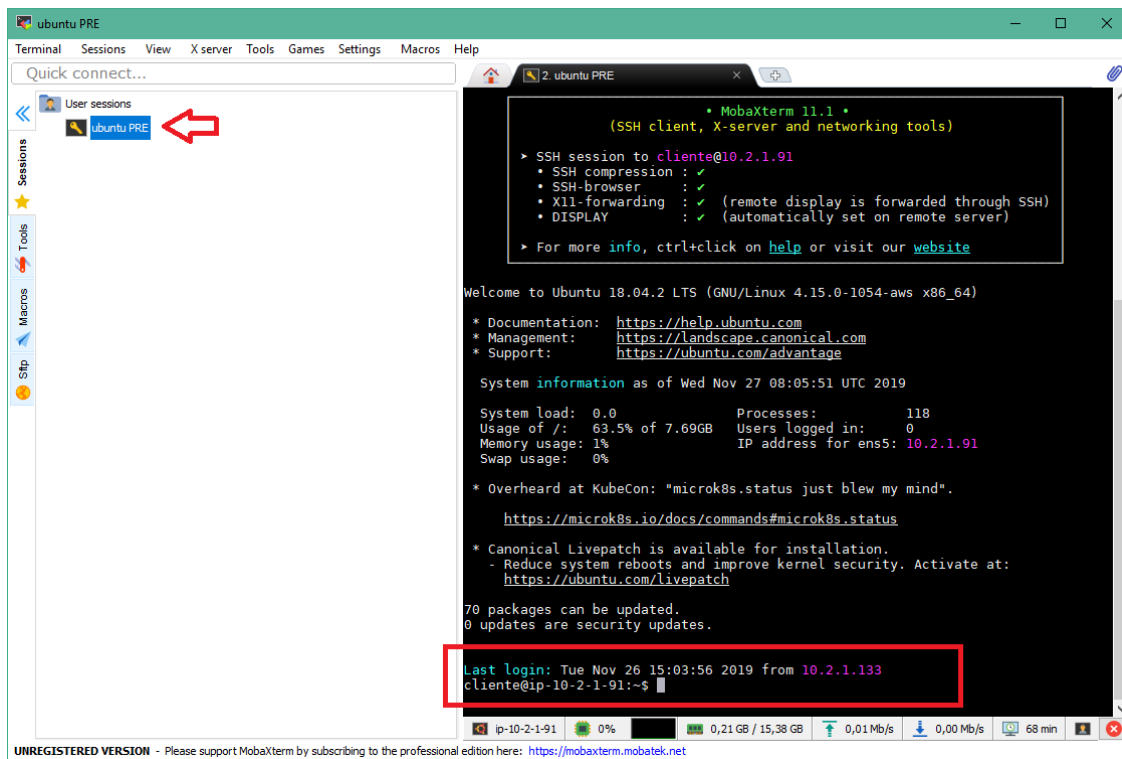


Take a look to the next picture to know where to put this parameters.



The setup will ask you for a name for this new session. Enter any valid text (“Ubuntu PRE” for example) and click ok to save. Now you can enter the into the Ubuntu terminal double-clicking session’s name. If everything is ok, you should have available the shell:

RAPEX Searcher – User setup



If you click in the “Sftp” tab at the left you will see the system directories where you can see easily what files and directories are available in the machine. This eases file manipulation tasks (upload/download, open, removal, etc.).

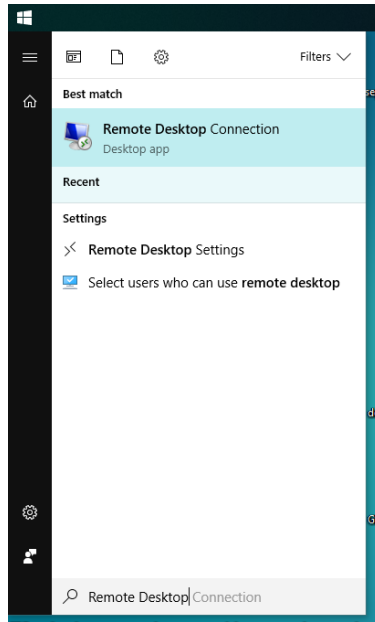
Every interaction you need to do with RAPEX Searcher module should be managed from here.

1.3. Windows Remote Desktop

The ETL component is deployed in the AWS EC2 Windows machine called: **RPXPRESWBETL01**

You do not need any additional software in Windows 10 to connect to the Windows server (ETL machine). Press “Windows” key and search “Remote desktop”.

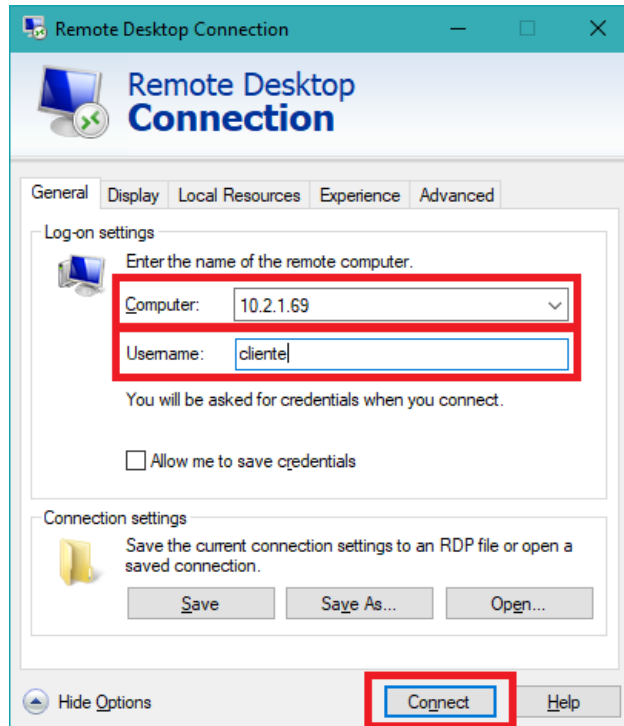
RAPEX Searcher – User setup



Now you need to introduce the host, user and password for the remote connection. These are:

- Computer: 10.2.1.69
- Username: cliente
- Password: TFWRjzE98EGcxXWU

The password shall be supplied after clicking on “Connect” button:



RAPEX Searcher – User setup

Accept the certificate prompt that appears after introducing the password. You know can interact with the Windows machine. You can logout easily clicking on the close button at the top (by default).



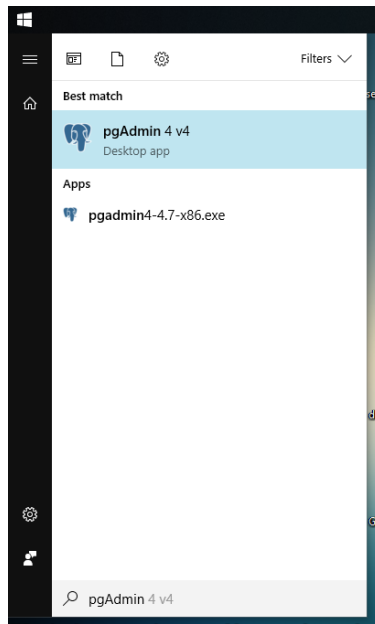
1.4. pgAdmin

In order to access to the PostgreSQL database you need an administration tool or development environment like Oracle SQL Developer or Microsoft SQL Server. In this case, for PostgreSQL we recommend pgAdmin.

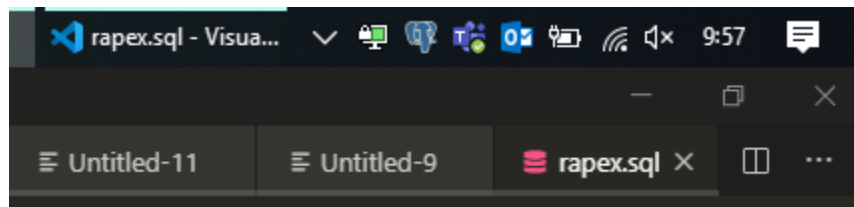
Download pgAdmin 4 from its official site: <https://www.pgadmin.org/download/>. Choose your version (Windows for this guide <https://www.pgadmin.org/download/pgadmin-4-windows/>) and install it as usual.

Once it is installed, you should have available the executable:

RAPEX Searcher – User setup



Launch it. Now the elephant from PostgreSQL should appear in the task bar. Double-click on it to open a new tab in your browser with pgAdmin service.

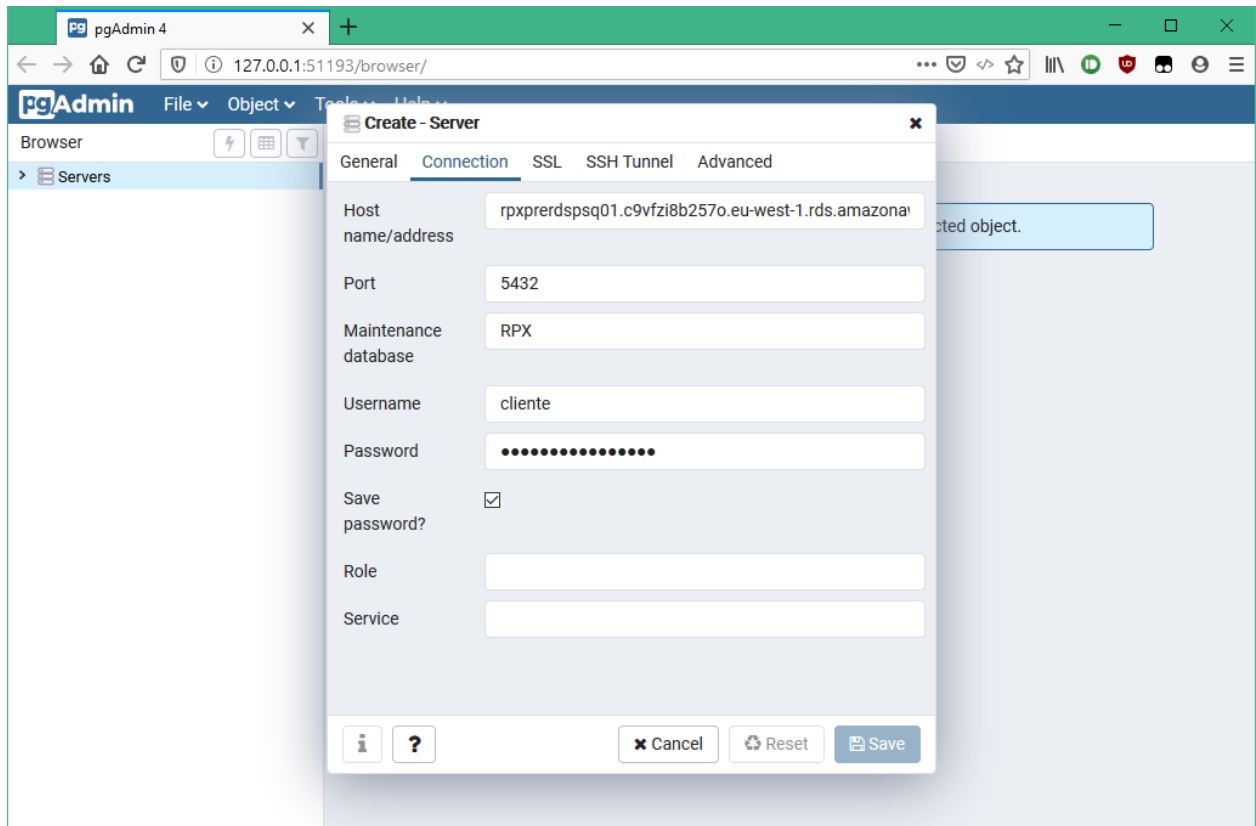


In this new tab, double-click on “Servers”, then “Create” and “Server...” last. You need to enter the credentials for the database. In this window, enter the name of the connection as you want (“rapex-pre” for example). Then, for “Connection”, you need the following information:

- Host: rpxprerdspsq01.c9vfzi8b257o.eu-west-1.rds.amazonaws.com
- Port: 5432
- Database: RPX
- User: cliente
- Password: Z69yyckK6M8mMWRReg

Remember checking “Save password?” option. At last, click “Save”.

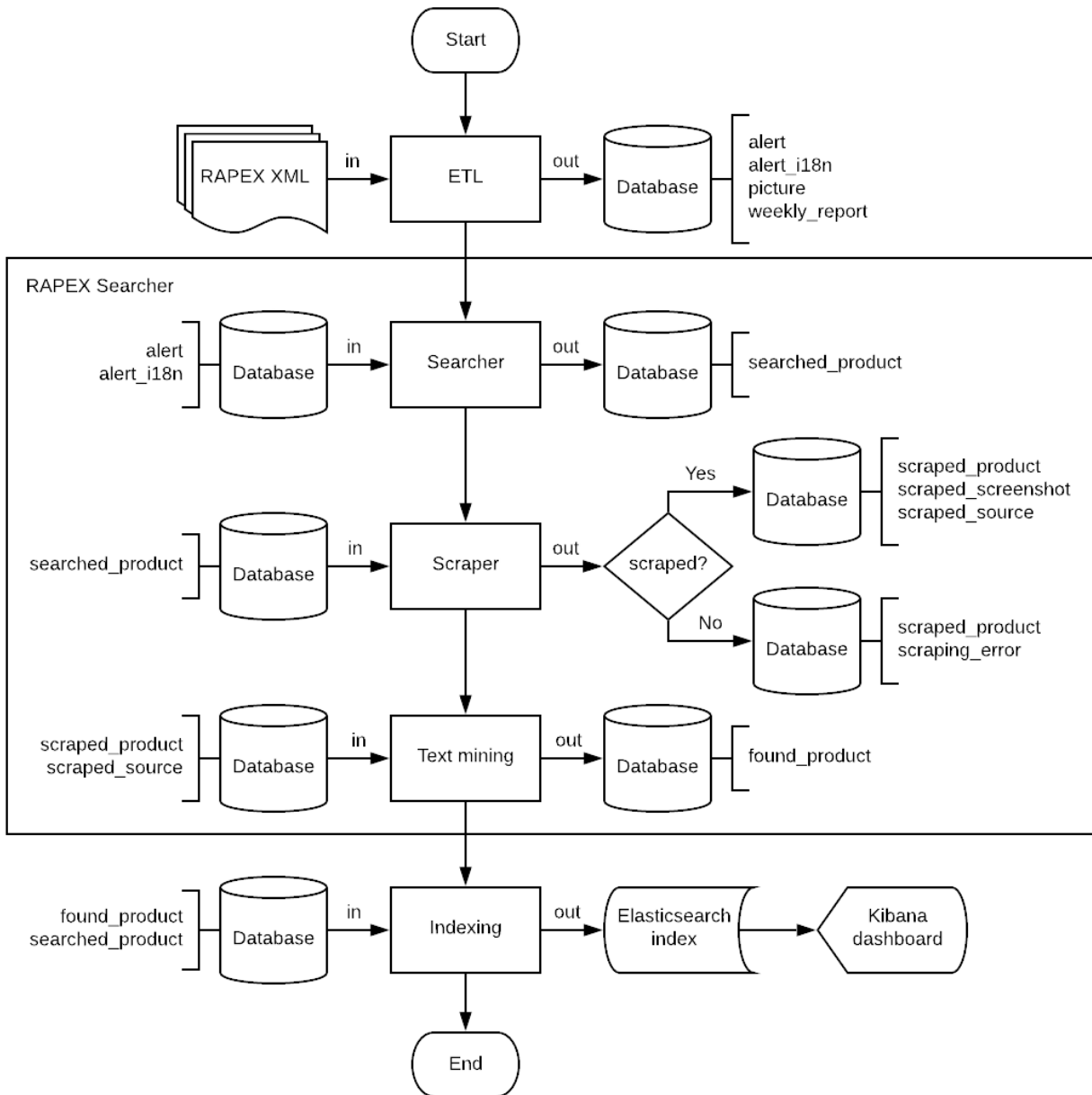
RAPEX Searcher – User setup



You can see the tables opening the recently created connection. Go to RPX database, then “Schemas”, “public” and “Tables”.

2. RAPEX Searcher execution and data flow

The following diagram shows the flow of the program and how it interacts with the database and its tables.



Next there is an explanation by process.

2.1. ETL

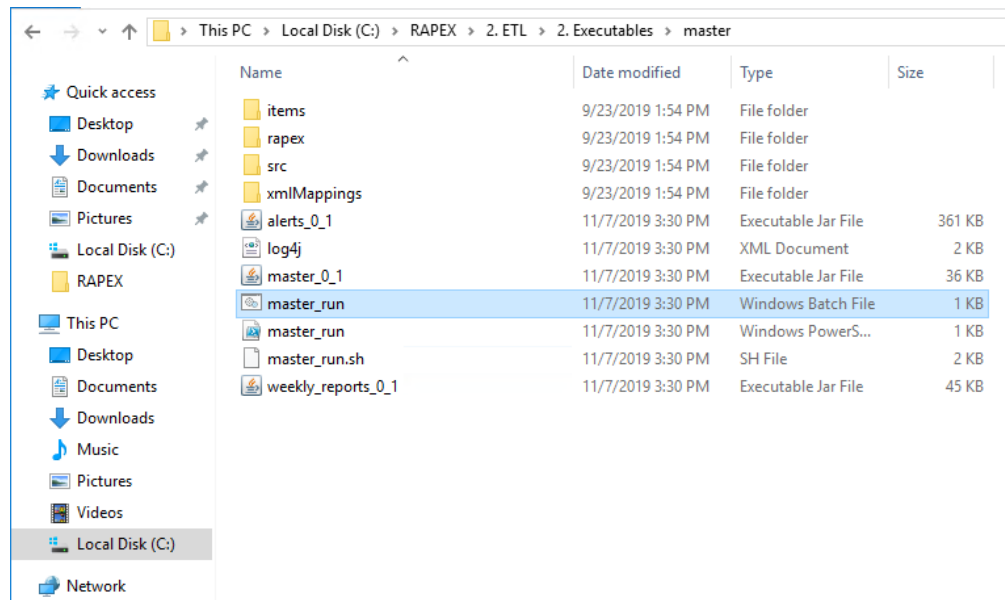
The weekly file is obtained through the following URL:

https://ec.europa.eu/consumers/consumers_safety/safety_products/rapex/alerts/?event=main.weeklyReports.XML

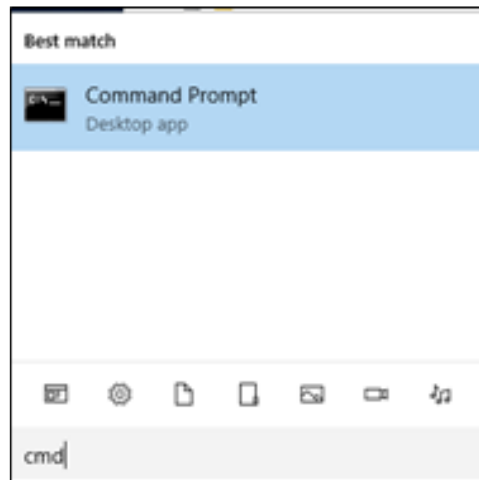
To execute this component, go to Windows ETL machine (explained in Windows Remote Desktop chapter). There are two ways to refresh data, automatically or manual:

1. The automatic way is a schedule task, it's happen every night at 12:00 p.m.

The manual way; when we are in windows remote desktop, we go to RAPEX folder -> 2.ETL -> 2. Executables -> master and we found a batch file “master run”, if we click twice it will be start the data refresh.

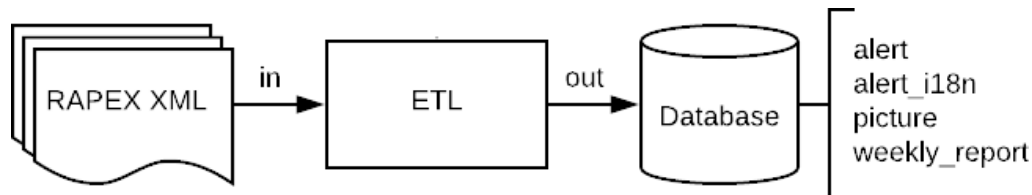


Other way to run by is by console. Launch “cmd” console open Windows dialog and type “cmd”:



And then write this sentence

```
C:\RAPEX\2. ETL\2. Executables\master>master_run.bat
```



The ETL takes RAPEX XML and processes the data present there storing it in alert, alert_i18n, picture and weekly_report tables.

2.2. RAPEX Searcher

This component (divided between three “minor” components) shall be executed from the Ubuntu machine (connection explained in MobaXterm chapter). Despite being three different parts with three different inputs and outputs, the whole component is launched as a single one.

Launch it with:

```
sudo systemctl start rapex
```

Replace `start` by `stop` to stop the service and replace it by `restart` to restart it.

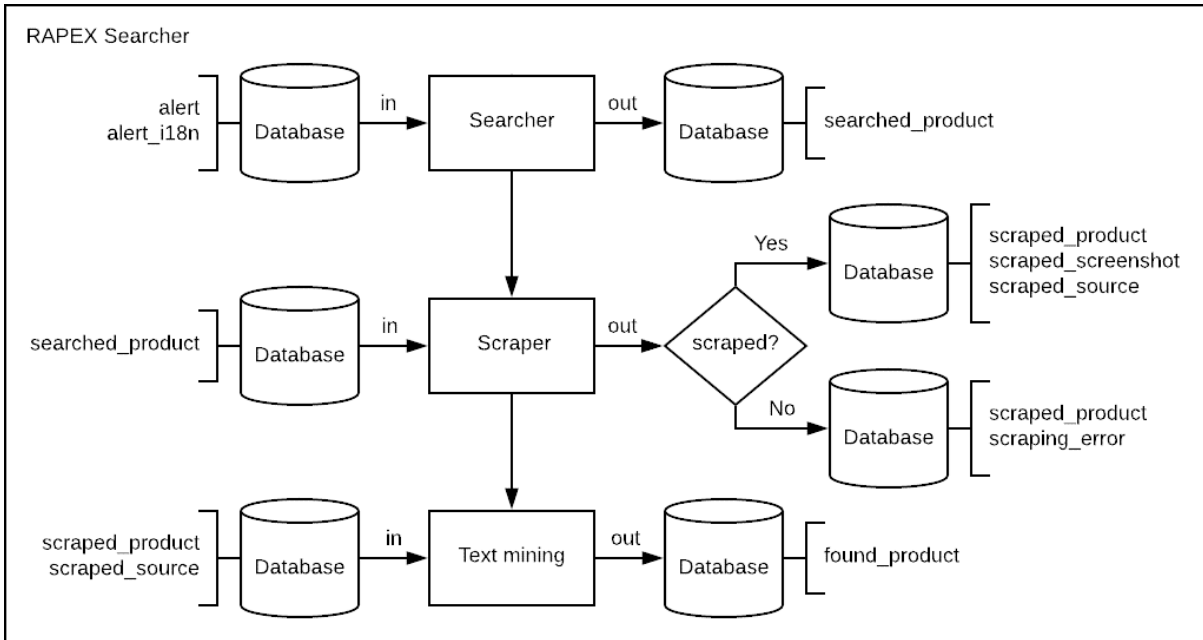
You can see the output of the service executing the following command:

```
sudo journalctl -u rapex -f
```

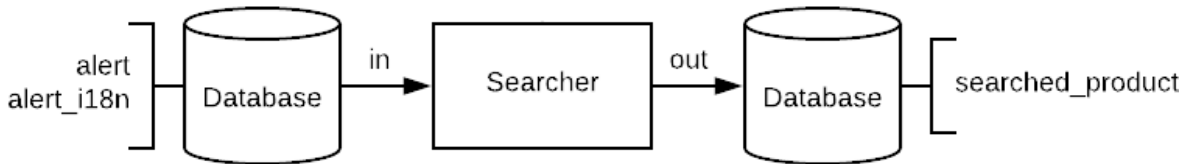
With this command the shell will print every log line as it is working. You cannot write any command in this mode. To exit, press Ctrl+C.

Every new registry added in “out” tables contains a “timestamp” for each registry processed.

RAPEX Searcher – User setup

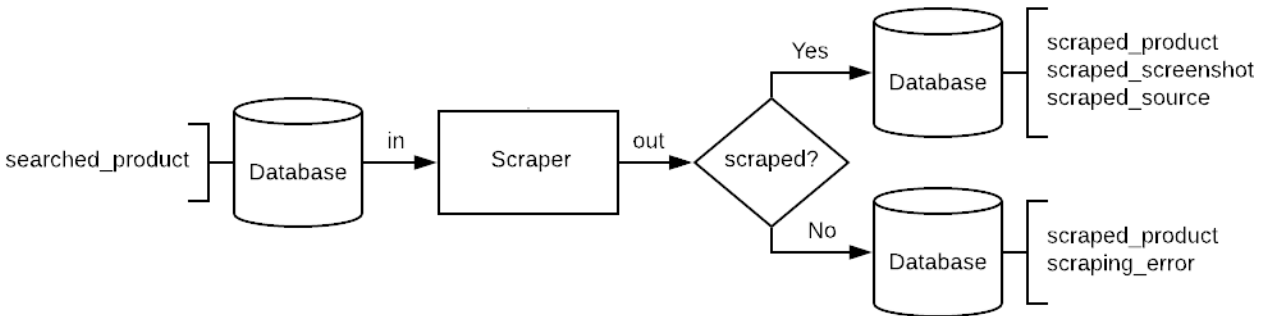


2.2.1. Searcher



The component that makes the query to Google API takes the information from the alerts imported by the ETL. Every entry in alert table is an alert from RAPEX. Every alert will have as many entries in alert_i18n table as languages configured. This table contains alert information that has multi-language values (as brand, name or description for example).

2.2.2. Scraper

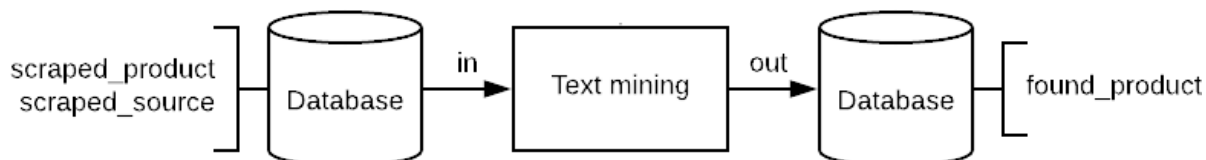


The scraper component takes the results from Google (URLs mainly). It tries to access these URLs to extract their HTML code. This scraping could not be done by some reasons that appear in scraping_error table.

RAPEX Searcher – User setup

- In case the scraping can be done, the process adds a row in scraped_product table by URL. Each URL should have a scraped_source row and a scraped_screenshot associated.
- In case the scraping cannot be done, the process adds a row in scraped_product showing that the scraping was not done and why (associating an error from scraping_error table).

2.2.3. Text mining



This process takes from every scraped_product that could be scraped its HTML code (scraped_source table). Based on the analysis done, it will add or not a new row in found_product table. This means that every row in found_product is a positively identified URL.

2.3. Elasticsearch indexing

The indexing is managed by Logstash. This program is launched by a command with RAPEX config file location as parameter.

Execute the following command to launch Logstash and index in Elasticsearch the data from the output of RAPEX Searcher.

```
sudo /usr/share/logstash/bin/logstash -f /etc/logstash/conf.d/rapex.conf
```

It will execute two queries to the PostgreSQL database to make the indices used later in Kibana.

This SQL query for rapex_searched_products_last index:

```
SELECT sp.id AS searched_product_id,
       sp.alert_id,
       sp.searched_date,
       sp.url,
       sp.domain,
       wr.publication_date,
       al.value AS alert_level,
       at.value AS alert_type,
       c.value AS category,
       rt.value AS risk_type,
       ct.iso_value
FROM searched_product sp
     JOIN alert a ON sp.alert_id = a.id
     JOIN weekly_report wr ON a.weekly_report_id = wr.id
     JOIN alert_level al ON a.alert_level_id = al.id
```

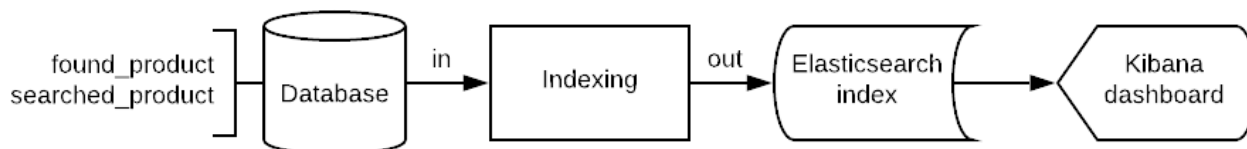
RAPEX Searcher – User setup

```
JOIN alert_type at ON a.alert_type_id = at.id
JOIN category c ON a.category_id = c.id
JOIN alert_risk_type ar ON ar.alert_id = a.id
JOIN risk_type rt ON ar.risk_type_id = rt.id
JOIN country ct ON ct.id = sp.country_id
```

And this one for rapex_found_products_last index:

```
SELECT fp.id AS found_product_id,
shp.alert_id,
fp.discovery_date,
shp.url,
shp.domain,
rt.value AS risk_type,
al.value AS alert_level,
at.value AS alert_type,
c.value AS category,
ct.iso_value,
shp.searched_date,
shp.id as searched_product_id,
twp.total_found_wepages
FROM found_product fp
JOIN scraped_product sp ON fp.scraped_product_id = sp.id
JOIN searched_product shp ON shp.id = sp.searched_product_id
JOIN alert a ON shp.alert_id = a.id
JOIN alert_risk_type ar ON ar.alert_id = a.id
JOIN risk_type rt ON ar.risk_type_id = rt.id
JOIN alert_level al ON a.alert_level_id = al.id
JOIN alert_type at ON a.alert_type_id = at.id
JOIN category c ON a.category_id = c.id
JOIN country ct ON ct.id = shp.country_id
JOIN (SELECT a.id as alert_id, count(fp.id) as
total_found_wepages
FROM found_product fp
JOIN scraped_product sp ON fp.scraped_product_id
= sp.id
JOIN searched_product shp ON shp.id =
sp.searched_product_id
JOIN alert a ON shp.alert_id = a.id
GROUP BY a.id) twp ON shp.alert_id = twp.alert_id
```

During the execution, the data will follow the next flow.



As said before, Kibana needs Elasticsearch indices for its visualisations. To do so, the indexing process takes the information resultant from found_product and searched_product. These

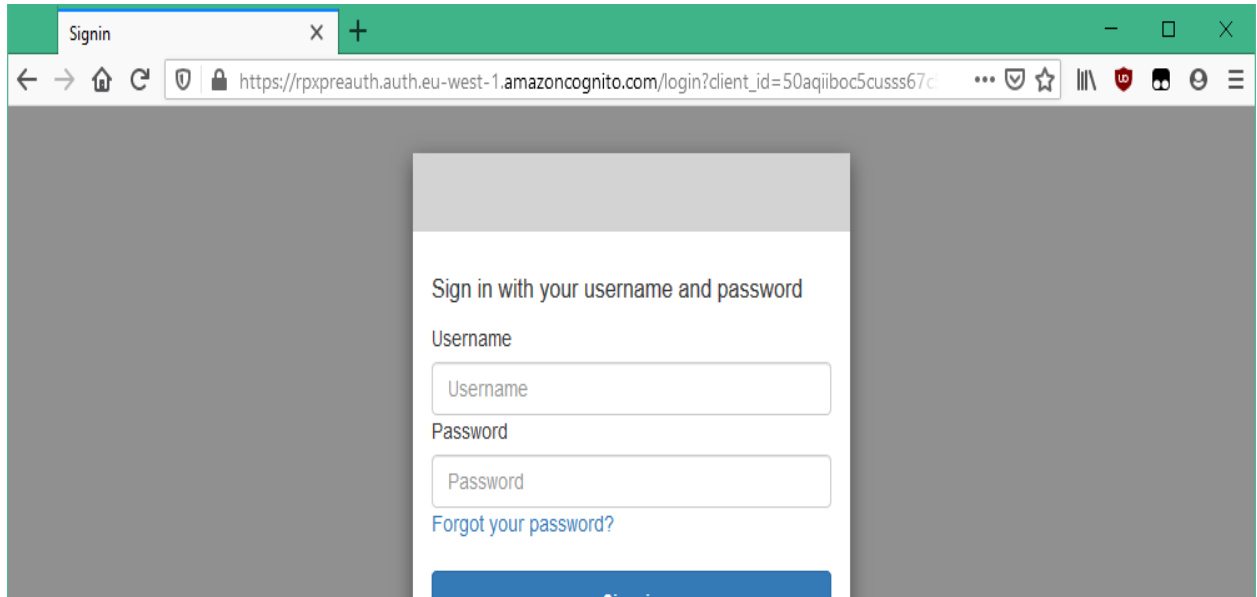
RAPEX Searcher – User setup

tables are the necessary to build the dashboard showed before. Some other information are taken too (like alert ID from alert table) but these two are the most relevant.

3. Accessing Kibana

Kibana URL is: https://prerpxesaut.everincloud.com/_plugin/kibana/

You need a user and password to access Kibana. This user will be created based on the username and email provided to the team. There could be as many users as desired. These credentials are used in the following window:



Once you are in, click on the dashboard button on the left panel to see the dashboard configured.

