# Procédure d’installation et d’indexation Elasticsearch :

## Prérequis :

### Téléchargements à faire :

Spark : <http://apache.crihan.fr/dist/spark/spark-2.3.0/spark-2.3.0-bin-hadoop2.7.tgz>

Winutils : pour simuler Hadoop sur Windows : <https://github.com/steveloughran/winutils/raw/master/hadoop-2.7.1/bin/winutils.exe>

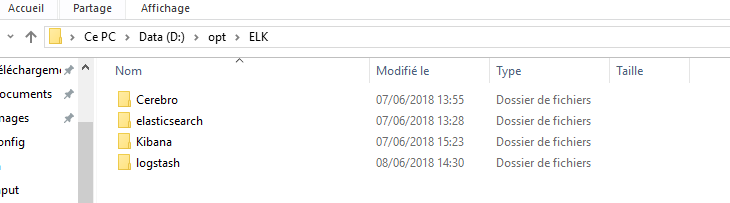
Anaconda : <https://repo.anaconda.com/archive/Anaconda3-5.2.0-Windows-x86_64.exe>

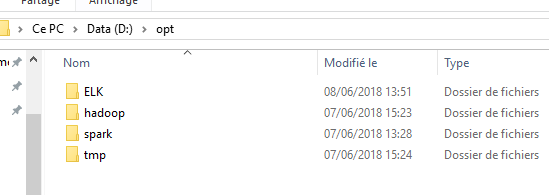
Stack ELK :

* Elasticsearch : <https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.2.4.zip>
* Logstash : <https://artifacts.elastic.co/downloads/logstash/logstash-6.2.4.zip>
* Kibana : <https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.2.4.zip>
* Cerebro plugin pour visualiser Elasticseach index : <https://github-production-release-asset-2e65be.s3.amazonaws.com/54560347/ce10c7a4-3fd6-11e8-966c-a92b06620bba?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20180607%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20180607T090802Z&X-Amz-Expires=300&X-Amz-Signature=806009c59fbcee374b01d4e4f48b74f8d4906d71c67a12bc89ba4549e0fbcc64&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3Dcerebro-0.7.3.zip&response-content-type=application%2Foctet-stream>

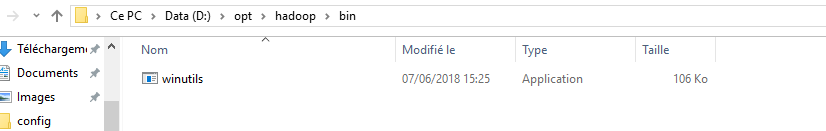
## Installation:

Décompresser les fichiers zip télécharger sur un dossier comme indiquer sur l’imprimé écran suivant:



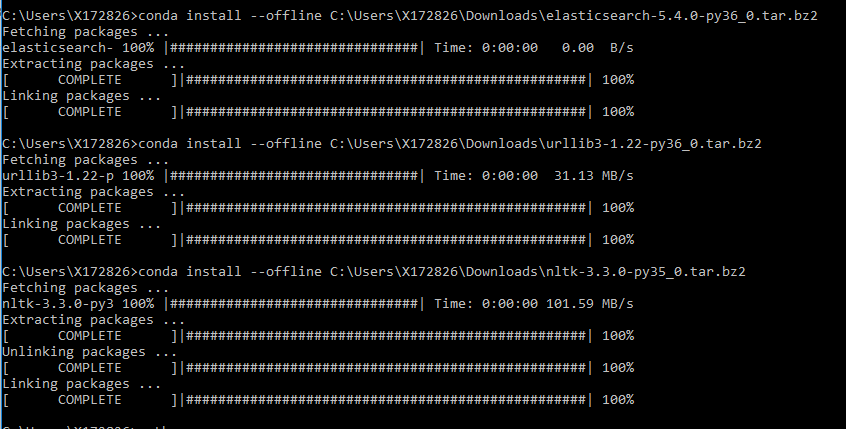


Créer un dossier « hadoop » 🡺 et dans le dossier « hadoop  » un autre dossier bin 🡺et placer le Winutils.exe dans le dossier bin



Téléchargement des Libraires et les installer en offline vue que le proxy SoG bloque les command pip et Conda :

Command d’installation offline : “conda install --offline C:\Users\X172826\Downloads\nltk-3.3.0-py35\_0.tar.bz2”



Changement des variables Windows sur le profile :

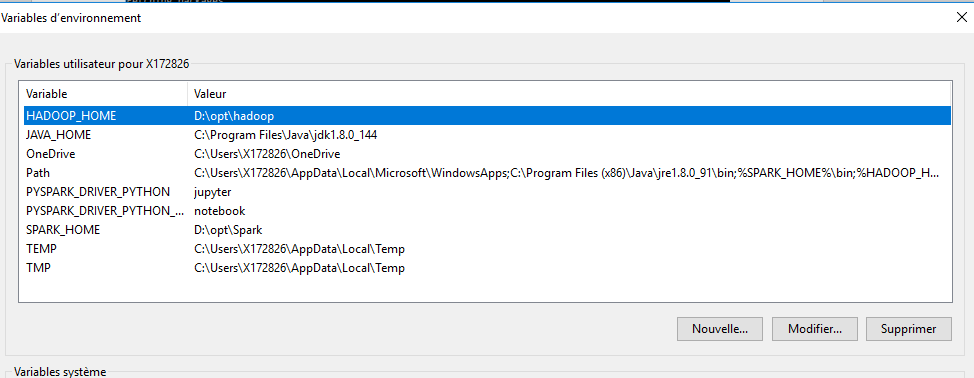
Commandes des variables via CMD:

setx SPARK\_HOME D:\opt\spark

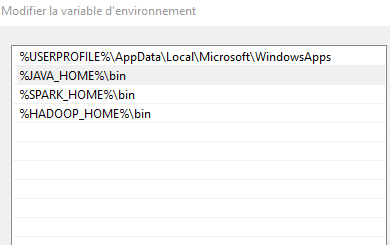
setx HADOOP\_HOME D:\opt\hadoop

setx PYSPARK\_DRIVER\_PYTHON jupyter

setx PYSPARK\_DRIVER\_PYTHON\_OPTS notebook



Et dans Path rajouter :

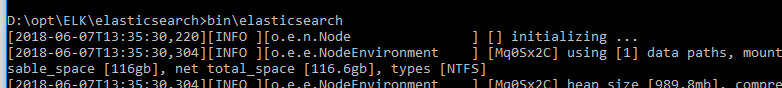


## Exécution :

Elasticsearch :

Pour lancer Elasticsearch aller sur le chemin du dossier d’ES via CMD et tapez la commande suivante :

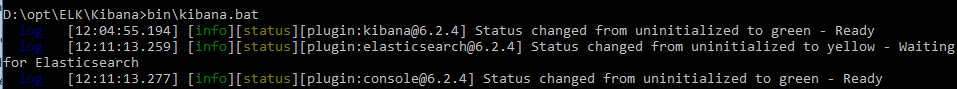
D:\opt\ELK\elasticsearch> bin\elasticsearch



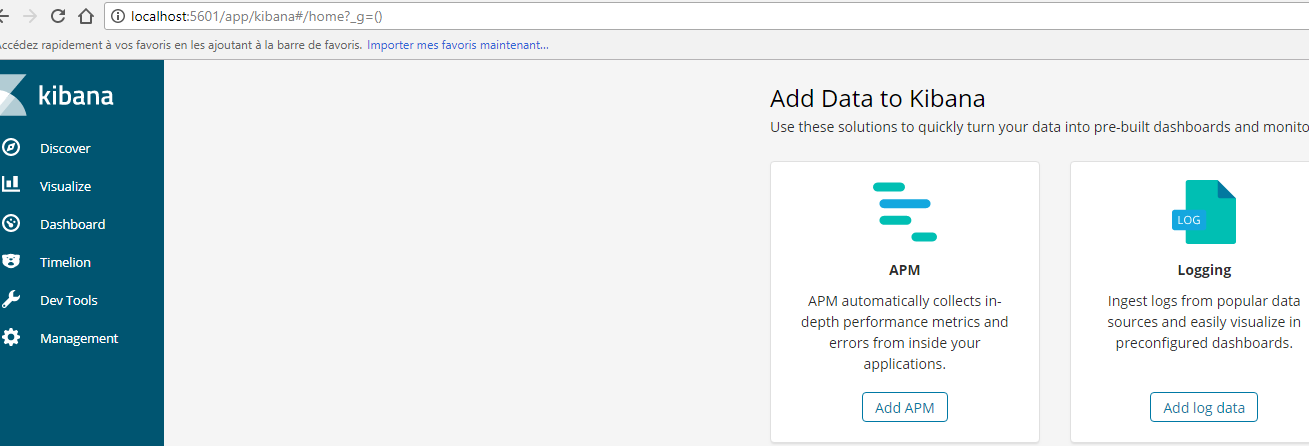
Kibana :

Pour lancer kibana aller sur le chemin du dossier d’kibana via CMD et tapez la commande suivante :

D:\opt\ELK\Kibana >bin\kibana.bat



Allez vérifier sur l’url suivant : localhost :5601



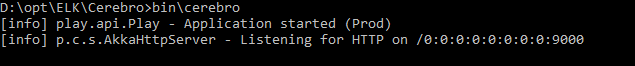
Logstash :

Allez dans le dossier d’elasticseach et taper la commande suivante : D:\opt\ELK\logstash> « bin\logstash -f conf\_csv.conf »



Cerebro :

Command pour lancer Cerebro : D:\opt\ELK\Cerebro>bin\cerebro



## Indexation :

### Méthode 1 :

Code python pour indexation des donner sur Elasticsearch:

From pyspark import SparkContext

from pyspark.sql.types import \*

from elasticsearch import Elasticsearch

from datetime import datetime

import os

os.environ['PYSAPRK\_SUBMIT\_ARGS']= '--jars D:\opt\spark\jars\elasticsearch-spark-20\_2.11-6.2.4.jar pyspark-shell'

filelocation=spark.sparkContext.textFile("file:///C:/Users/X172826/Desktop/Input/PGS\_IN.csv")

dfSpark=spark.read.option("header","true").option("sep", ";").option("inferSchema","true").csv(filelocation)

es=Elasticsearch()

dfSpark.write.format("org.elasticsearch.spark.sql").option("es.mapping.id"," ID Incident").mode("append").save("index/type")

### Méthode 2 :

Fichier de configuration logstash pour indexer le fichier input csv « ce fichier à enregistrer sur le dossier de logstash »:

input {

file {

path => "C:\Users\X172826\Desktop\Input\in\in\PGS\_IN\_dos\_2.csv"

type => "csv"

start\_position => "beginning"

codec => plain {charset => "ISO-8859-1"}

}

}

filter {

csv {

columns => ["ID Incident","Incident Type","Critical User","001 - Contact Entity","Status","Service","Category","Sub-Category","Characterisation","Title","Incident Description","Solution","Open Time","Close Date","Assignment Group","EUS - Follow Group;Open\_By\_Group","Restored\_By\_Group","Previous Assignment Group","Clock\_Group","Activity Description","Activity Date"]

separator => ";"

}

date {

locale => "fr"

match => ["Close Date" , "dd/MM/yyyy HH:mm:ss"]

target => "Close Date"

#tag\_on\_failure => ["error Close Date"]

}

date {

locale => "fr"

match => ["Open Time" , "dd/MM/yyyy HH:mm:ss"]

target => ["Open Time"]

#tag\_on\_failure => ["error Open Time"]

}

}

output {

elasticsearch {

hosts => "localhost:9200"

index=>"in\_ticket"

}

stdout{ codec => rubydebug

}

}

Annexe :

