**MANUAL CURSO DE AGVD**

**Página Oficial del Curso:**

<https://jdvelasq.github.io/courses/analitica-de-grandes-datos/>

**Laboratorios:**

<https://jdvelasq.github.io/courses/analitica-de-grandes-datos/grades.html>

\* Para poder acceder debe estar ingresado en el listado, si por algún motivo no esta por favor reportarlo en Slack. Actualmente esta en proceso de carga por lo que favor revisar su usuario el día 30 de noviembre de 2019.

**Pagina Slack:**

La página oficial del curso es: analiticagvdunalmed.slack.com

**Github con códigos:**

<https://github.com/amartinUnal/AnaliticaGVD/>

En este se colocaran los códigos que se utilizarán en clase.

**Instalación y configuración de herramientas para la clase:**

1. Crear una cuenta en github:

<https://jdvelasq.github.io/courses/analitica-de-grandes-datos/setup.html>

1. Descargar e instalar VirtualBox + Vagrant específico para Linux (caso de que utilices Windows):

<https://github.com/jdvelasq/vagrant4analytics>

En caso de utilizar iOS / Linux se puede utilizar Docker directamente, para lo cual descargar e instalar Docker.

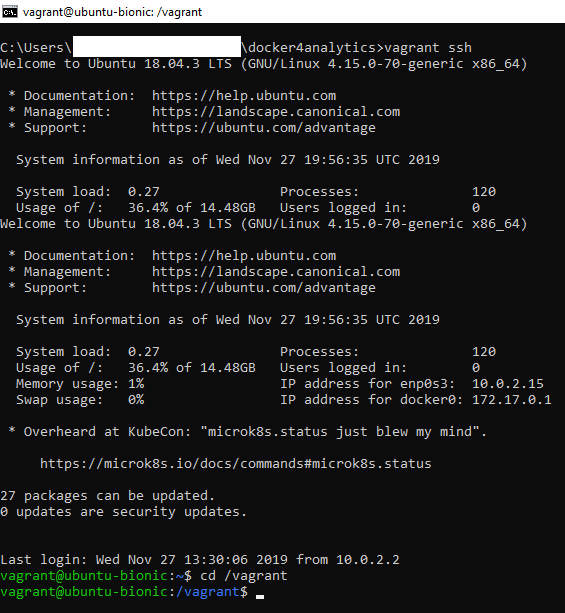
1. Para acceder a Vagrant seguir las instrucciones del repositorio:

<https://jdvelasq.github.io/courses/analitica-de-grandes-datos/setup.html>

\* En caso de tener problemas con el disksize al momento de usar Vagrant, instalar el paquete con la siguiente instrucción en línea de comandos:

vagrant plugin install vagrant-disksize





1. Si tienes Linux / iOS corres el Docker descargando la versión especifica correspondiente al bloque de trabajo del curso.

* Bloque 1: MapReduce
* Bloque 2: Pig
* Bloque 3: Hive
* Bloque 4: Spark

Con la instrucción:

docker run --rm -it -v "$PWD":/datalake --name hive -p 50070:50070 -p 8088:8088 -p 8888:8888 -p 5000:5000 jdvelasq/<nombre del contenedor>

\* Si estas usando el vagrant con SO Ubuntu y te genera error de permisos, recuerda adicionar el sudo antes del docker.

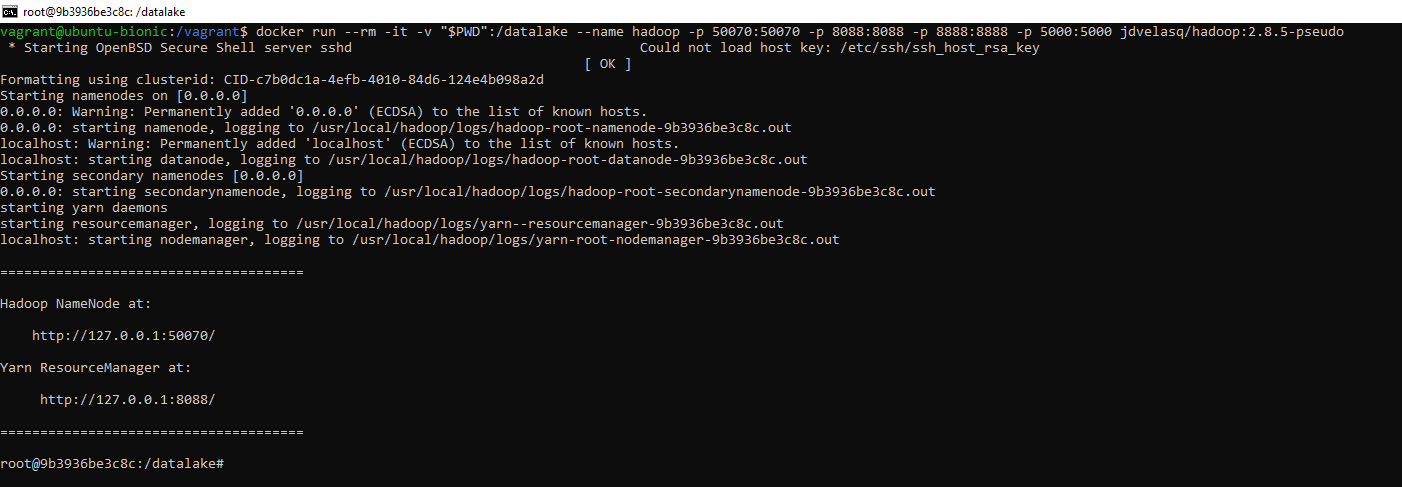
\* La versión pseudo monta hdfs, mientras que la versión standalone trabaja con el sistema de archivos local de la máquina. Por lo anterior la versión standalone puede ser de ejecución más rápida, pero la pseudo aprovecha el sistema de almacenamiento hdfs estándar cuando se trabaja en hadoop.

\* Recuerda correrlo la primera vez en una red distinta a la de la universidad dado que se requiere acceso para descargar ciertos componentes como Docker, las cuales están restringidas al interior del campus. Luego de la primera ejecución los archivos quedan guardados en tu maquina y ya es posible ejecutarlo en la red del campus.

1. Bloque 1: MapReduce

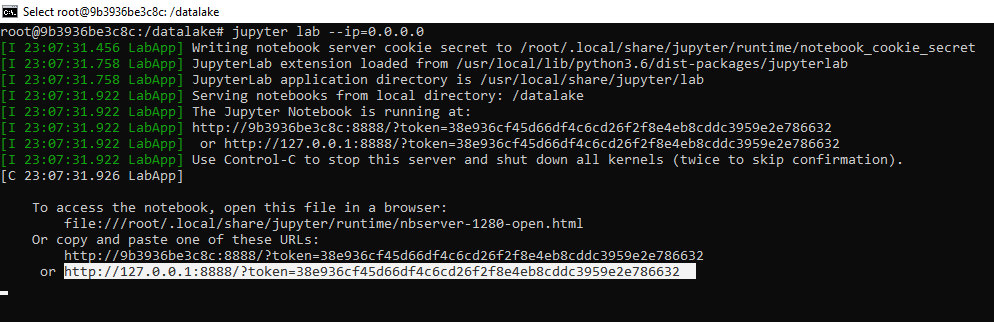
Para la primera clase vamos a utilizar: https://colab.research.google.com

docker run --rm -it -v "$PWD":/datalake --name hadoop -p 50070:50070 -p 8088:8088 -p 8888:8888 -p 5000:5000 jdvelasq/hadoop:2.8.5-pseudo

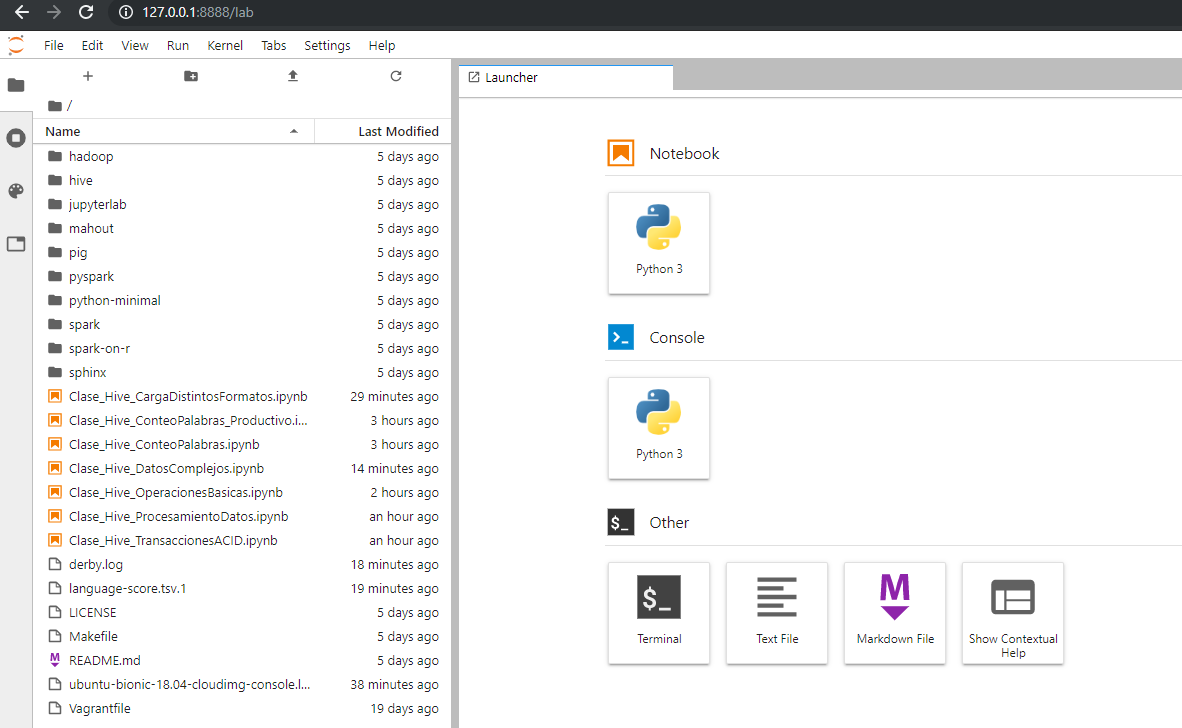


\* Para validar el funcionamiento de los nodos y del Yarn se puede ingresar al navegador la url 127.0.0.1 con el respectivo puerto

En este caso ya esta corriendo el docker para hive, basta con escribir hive o si se desea trabajar en Jupyter basta colocar la instrucción: jupyter lab --ip=0.0.0.0



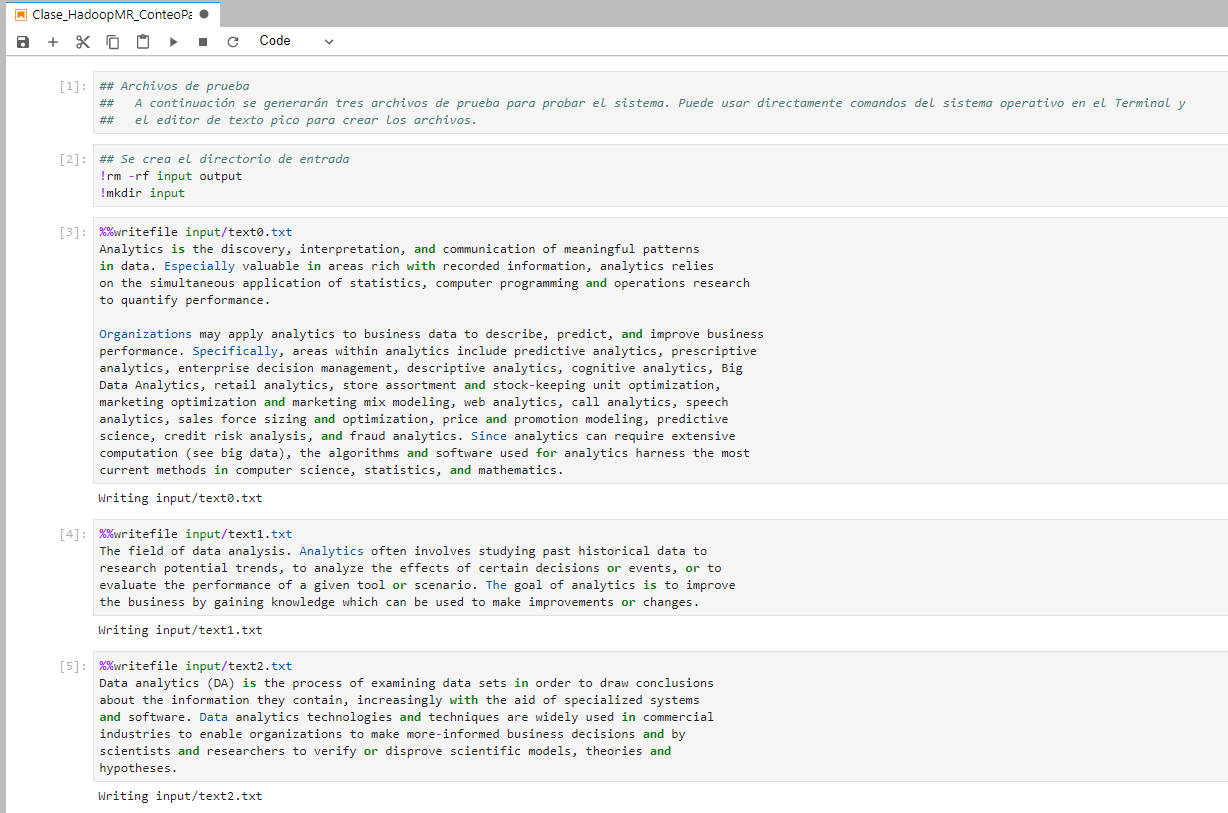
Selecciona la url y la pegas en el explorador:



Los ejemplos de clase corresponderán a:

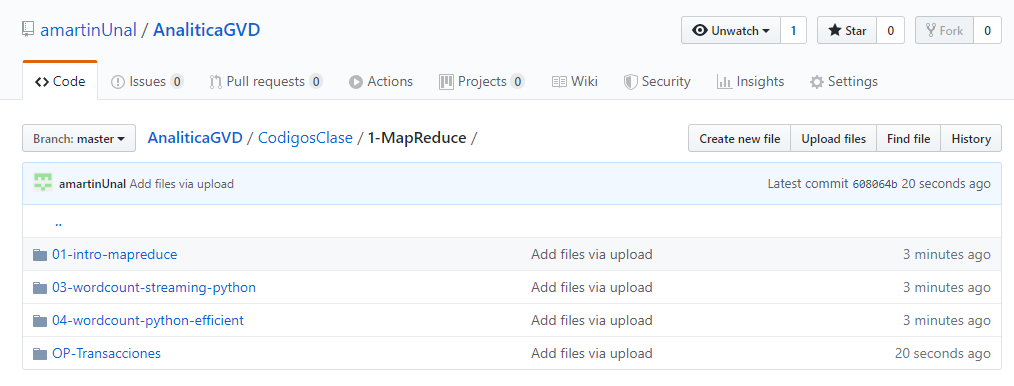
* <https://jdvelasq.github.io/courses/notebooks/hadoop/1-01-intro-mapreduce.html>
* <https://jdvelasq.github.io/courses/notebooks/hadoop/1-03-wordcount-streaming-python.html>
* <https://jdvelasq.github.io/courses/notebooks/hadoop/1-04-wordcount-python-efficient.html>
* <https://jdvelasq.github.io/courses/notebooks/hadoop/1-07-wordcount-standalone.html>
* <https://jdvelasq.github.io/courses/notebooks/hadoop/1-08-(opcional)-wordcount-java-standalone.html>
* <https://jdvelasq.github.io/courses/notebooks/hadoop/1-09-(opcional)-wordcount-java-pseudo.html>

ejemplo:

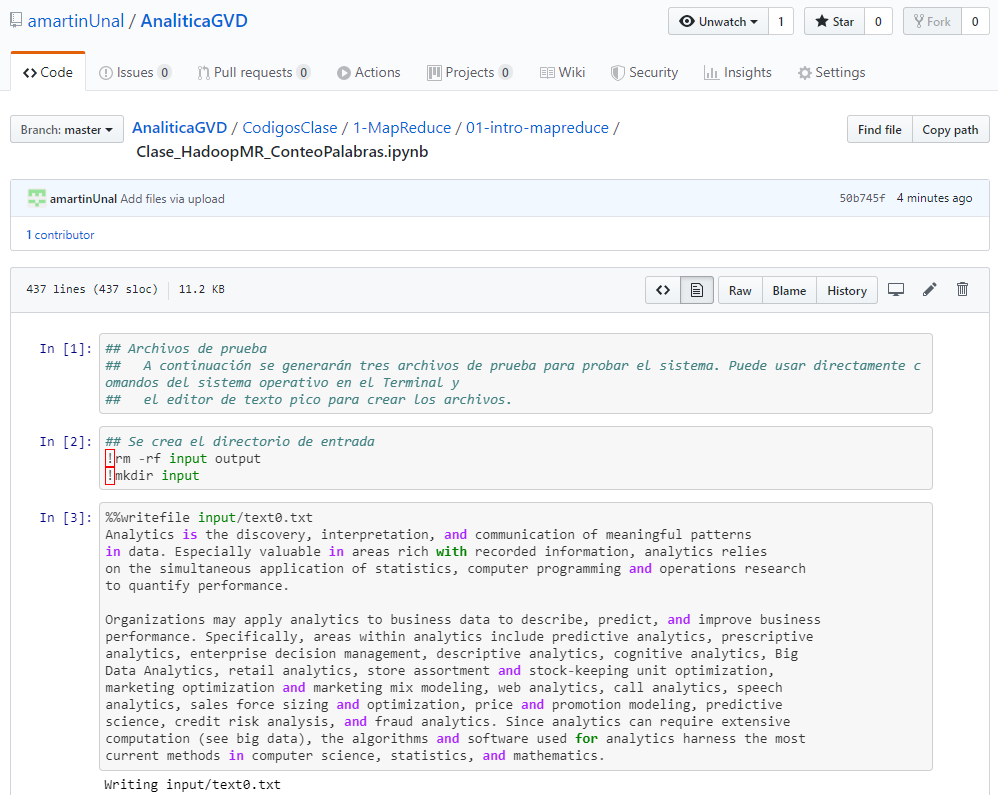


En caso de preferir por consola puedes utilizar el editor pico.

Los códigos están el github/AnaliticaGVD con sus respectivos resultados de ejecución en la estructura de carpetas por nombre:



Archivo de conteo de palabras:

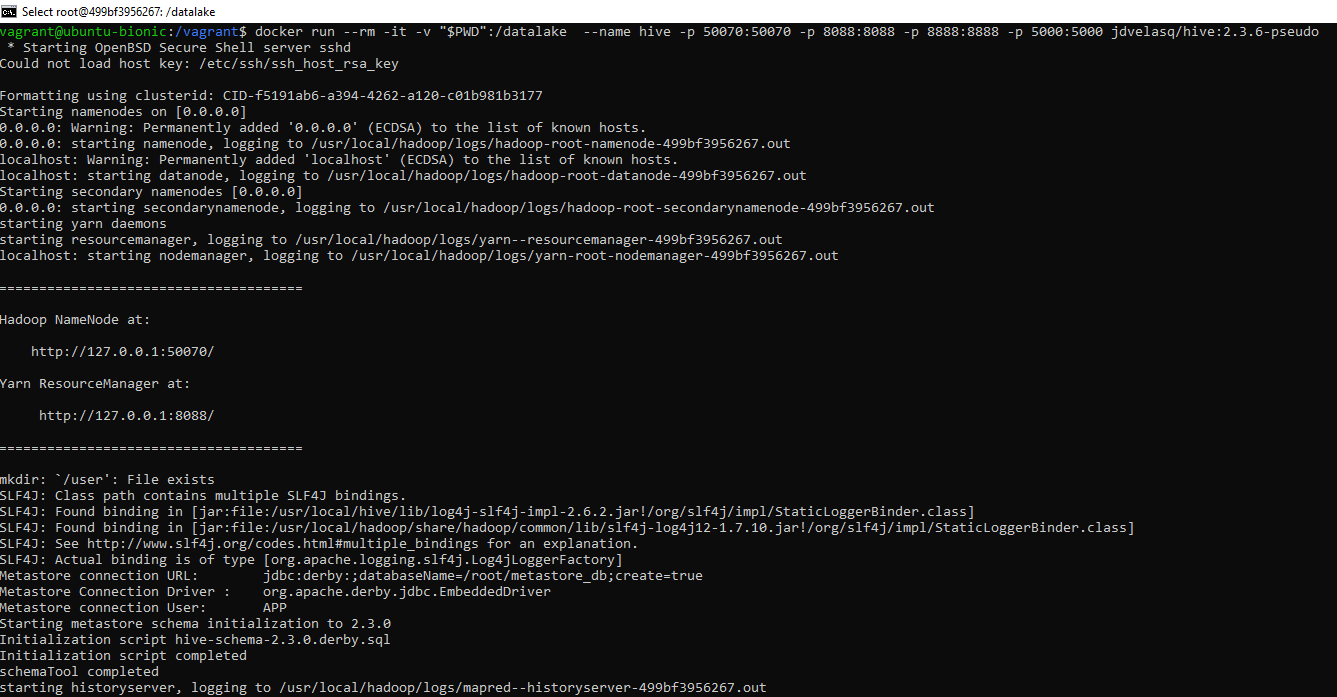


1. Bloque 2: Pig

docker run --rm -it -v "$PWD":/datalake --name pig -p 50070:50070 -p 8088:8088 -p 8888:8888 -p 5000:5000 jdvelasq/pig:0.17.0-pseudo

1. Bloque 3: Hive

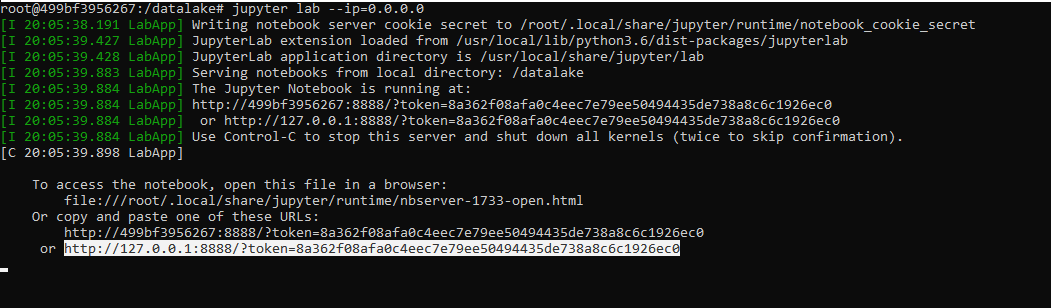
docker run --rm -it -v "$PWD":/datalake --name hive -p 50070:50070 -p 8088:8088 -p 8888:8888 -p 5000:5000 jdvelasq/hive:2.3.6-pseudo



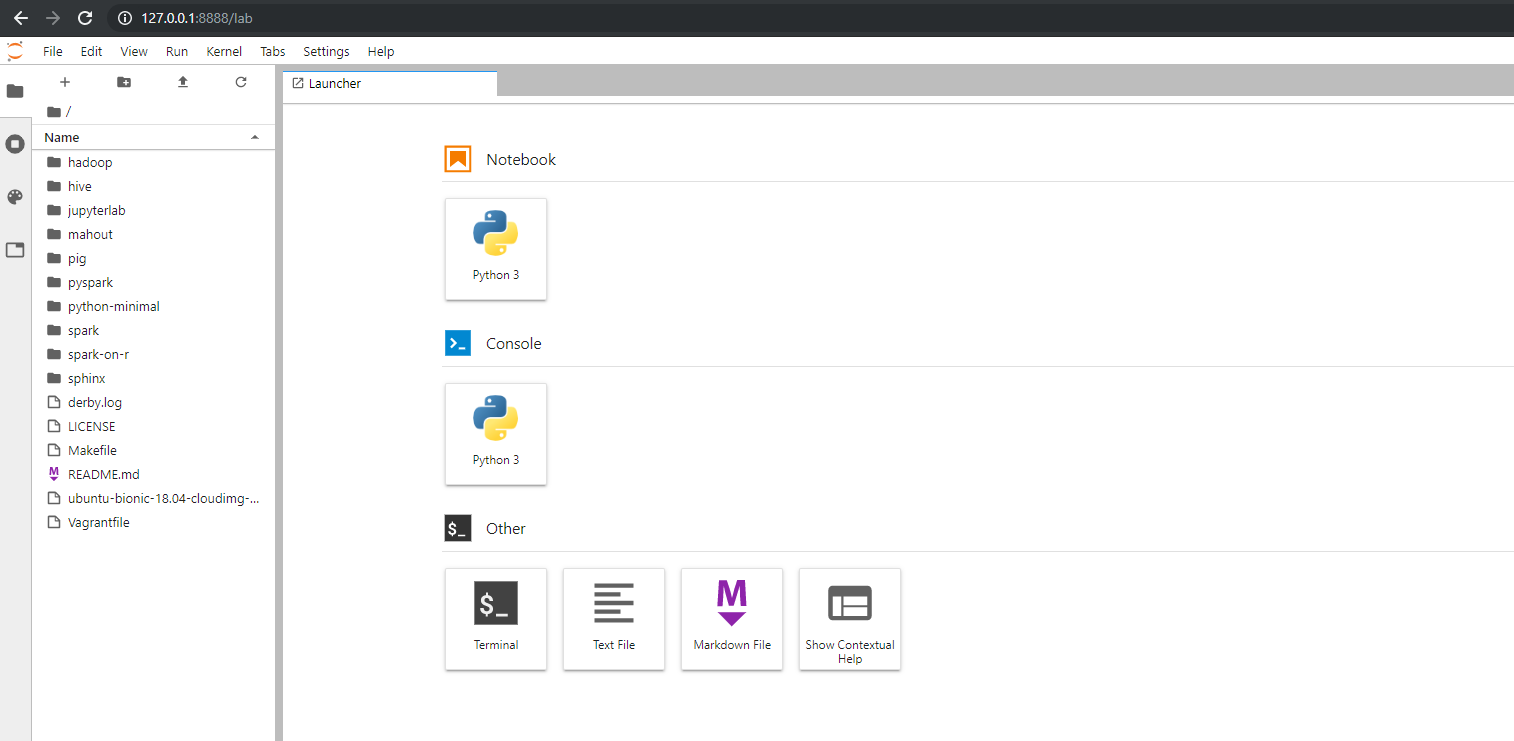
\* Para validar el funcionamiento de los nodos y del Yarn se puede ingresar al navegador la url 127.0.0.1 con el respectivo puerto

En este caso ya esta corriendo el docker para hive, basta con escribir hive o si se desea trabajar en Jupyter basta colocar la instrucción:

jupyter lab –ip=0.0.0.0



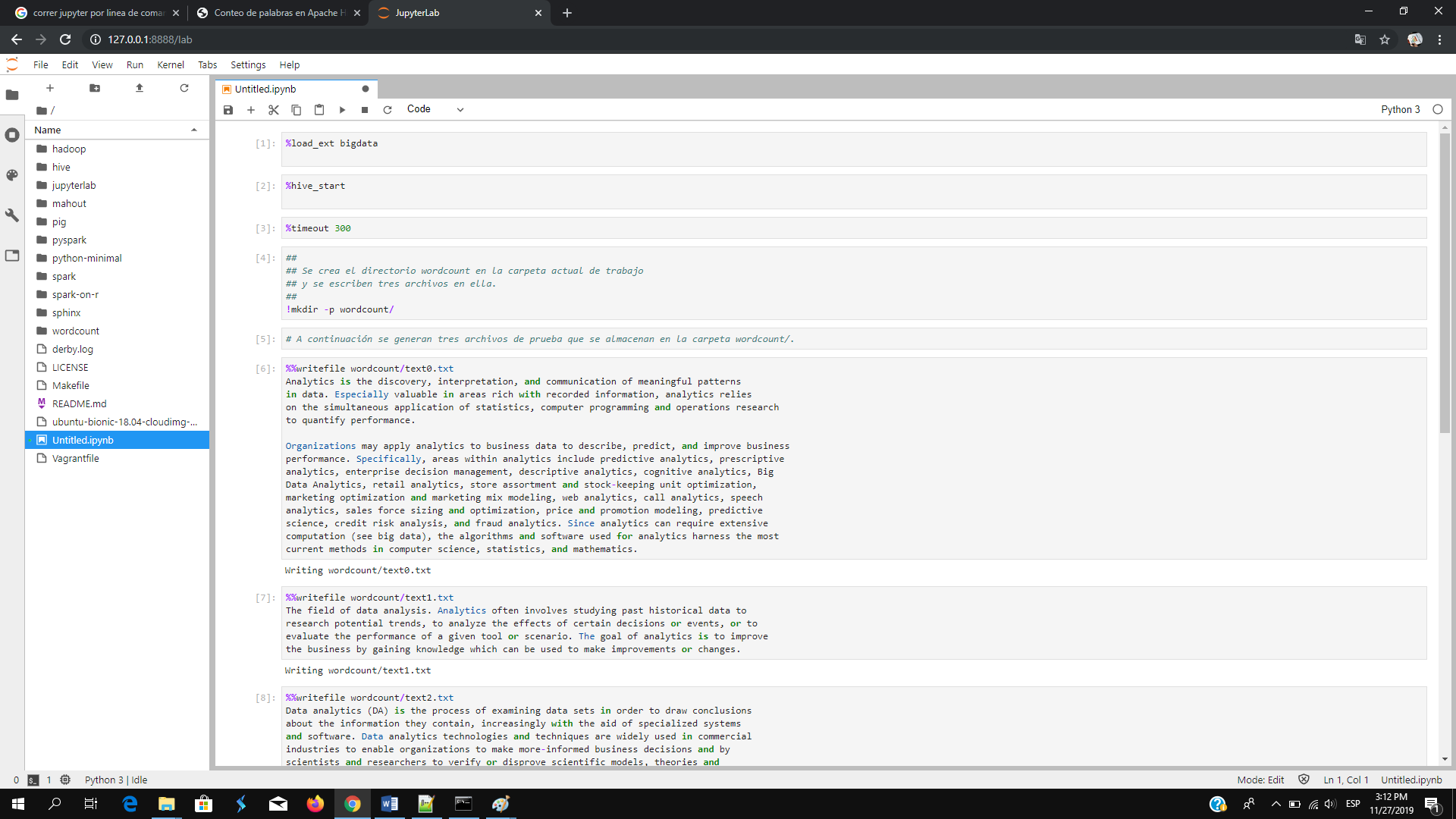
Selecciona la url y la pegas en el explorador:



Los ejemplos de clase corresponderán a:

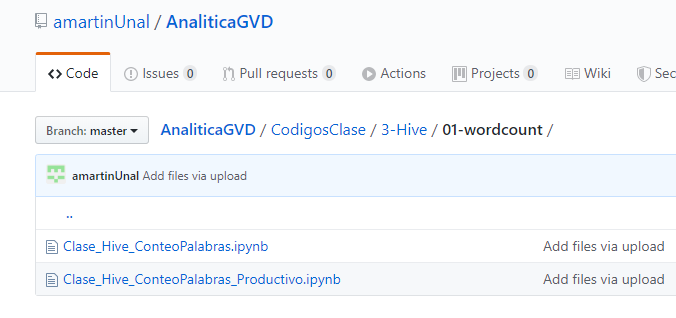
* <https://jdvelasq.github.io/courses/notebooks/hive/1-01-conteo-de-palabras-en-hive.html>
* <https://jdvelasq.github.io/courses/notebooks/hive/1-02-operaciones-basicas-en-hive.html>
* <https://jdvelasq.github.io/courses/notebooks/hive/2-03-procesamiento-de-datos-con-hive.html>
* <https://jdvelasq.github.io/courses/notebooks/hive/2-07-manejo-de-datos-complejos.html>
* <https://jdvelasq.github.io/courses/notebooks/hive/2-08-standalone.html>

ejemplo:

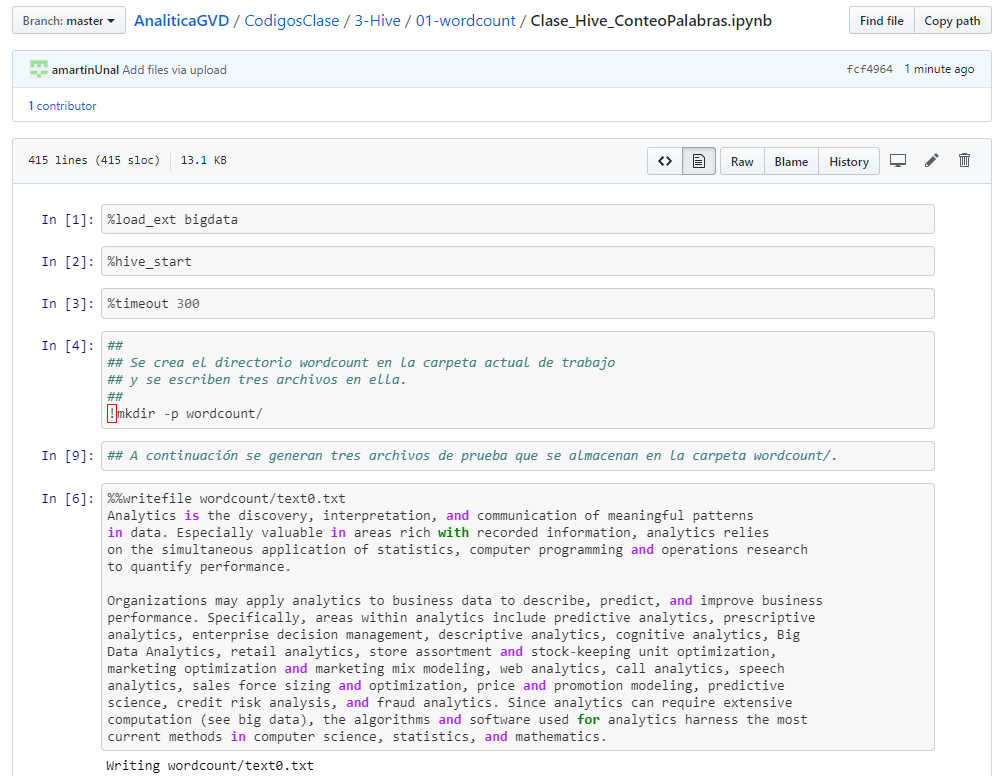


En caso de preferir por consola puedes utilizar el editor pico.

Los códigos están el github/AnaliticaGVD con sus respectivos resultados de ejecución en la estructura de carpetas por nombre:



Archivo de conteo de palabras:



1. Bloque 4: Spark