

README

1. Disponibilizar servicio de entorno analítico.

2. Integrantes: Nicolas Alarcon
 Sandra Hernández

3. Para realizar proyectos de investigación en los cuales trabajan un gran número de ingenieros es necesario que todos tengan acceso a los datos y avances de sus colegas de esta manera se lograría un avance mucho más eficiente del proyecto además de tener menos probabilidad de errores ya que los datos serán actualizados en “tiempo real”. Por todo lo mencionado anteriormente se deben tener los servicios como la Base de datos, herramientas para procesamiento big data (spark), interfaz de pruebas de código (jupyter notebook) y el soporte para lenguajes de programación python y R en un servidor de AWS la cual brinda seguridad y accesibilidad de los datos a través de todo el mundo.

4. Habilitar un servidor de AWS el cual tenga todas las herramientas mencionadas anteriormente y que puedan ser accedidas por todos los ingenieros de un proyecto dado.

user : ubuntu

ip: 13.59.44.23

ssh -i "kpLab.pem" ubuntu@13.59.44.23

kpLab.pem adjuntado

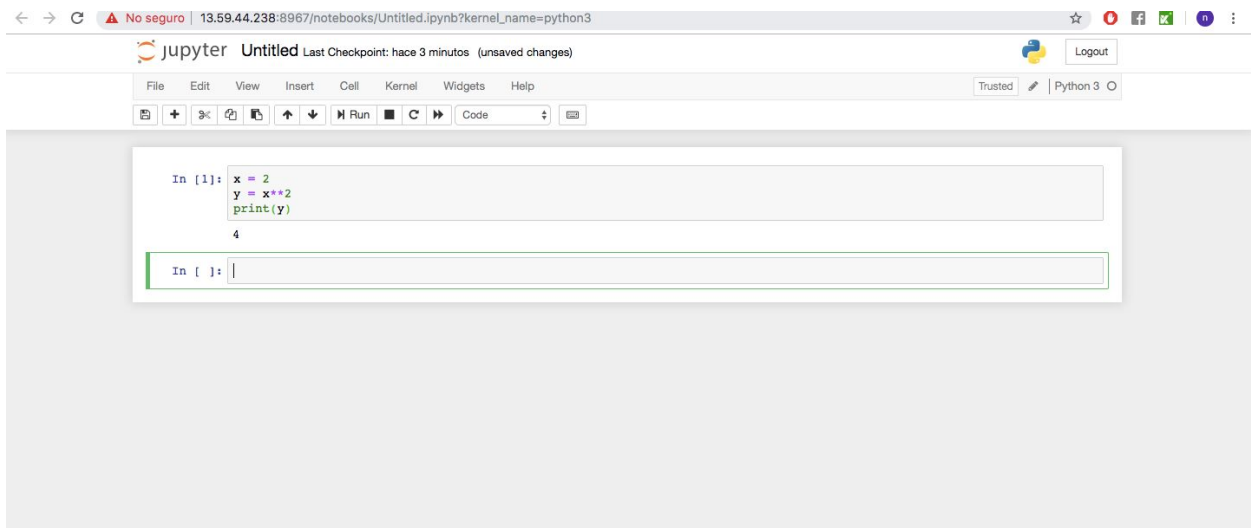
5. a) Las principales barreras que se presentaron a la hora de realizar la actividad fueron la falta de conocimientos sobre las diferentes tecnologías a ocupar así como también el tiempo de creación de una cuenta en AWS, pero sobre todo la poca experiencia en el uso y manejo de las diferentes herramientas.

El tiempo dedicado al trabajo fue en gran medida a la comprensión y lectura de cómo funcionan y que son las diferentes tecnologías así como al tiempo para corregir los diferentes errores en la instalación y configuración de estas.

6.



Jupyter notebook montado en el servidor



Ejecutando un código python en jupyter.

Amazon S3 > labdistribuidos

Información general	Propiedades	Permisos	Administración
---------------------	-------------	----------	----------------

🔍 Escriba un prefijo y pulse Intro para buscar. Pulse ESC para borrar.

[Cargar](#) [+ Crear carpeta](#) [Descargar](#) [Acciones](#) [Versiones](#) [Ocultar](#) [Mostrar](#) EE.UU. Este (Ohio) ↻

Mostrando desde 1 hasta 3			
<input type="checkbox"/> Nombre ▾	Última modificación ▾	Tamaño ▾	Clase de almacenamiento ▾
<input type="checkbox"/> 59650920_665138443927095_3462932375854383104_o.jpg	may. 21, 2019 9:10:37 p. m. GMT-0400	502.8 KB	Estándar
<input type="checkbox"/> License premium.txt	may. 22, 2019 12:53:44 a. m. GMT-0400	1.1 KB	Estándar
<input type="checkbox"/> licencia.txt	may. 22, 2019 1:02:17 a. m. GMT-0400	17.5 KB	Estándar
Mostrando desde 1 hasta 3			

Bucket S3 creado para la implementación

```
ubuntu@ip-172-31-44-174: /usr/local/spark-2.4.3-bin-hadoop2.7/bin
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
19/05/22 07:44:15 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform..
. using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://ip-172-31-44-174.us-east-2.compute.internal:4040
Spark context available as 'sc' (master = local[*], app id = local-1558511069291).
Spark session available as 'spark'.
Welcome to

  ____      __
 / ___ |    /  \
| |  \| |  /    \
| |___| | /  /\
|  ___ | / ____ \
| |  \| | \____/
|_|___|_|

version 2.4.3

Using Scala version 2.11.12 (OpenJDK 64-Bit Server VM, Java 11.0.3)
Type in expressions to have them evaluated.
Type :help for more information.

scala> 
```

Spark ejecutando al el servidor

7. link:

Introduciendo en el servidor el comando

```
jupyter notebook --ip=0.0.0.0 --no-browser --port=8890
```

y percatandose que el puerto 8890 esté abierto en nuestro servidor. este se puede acceder mediante la ip publica del servidor (<http://13.59.44.238:8890>)

8.

- Paso 1: Habilitar una máquina virtual al la nube en AWS.
 - Crear una cuenta en AWS
 - Solicitar el servicio EC2 para crear un MV
 - Generar la key-par para acceso a la máquina virtual: "kpLab.pem"
 - Crear una MV con sistema operativo ubuntu
- Paso 2: Instalar las diferentes tecnologías solicitadas S3, jupyter, spark, python y R.
 - Antes que todo se debe instalar java jdk: "sudo apt-get install default-jdk"
 - Instalar scala: "install scala"
 - Instalar git: "install git"
 - Descargar Spark: "wget https://www-us.apache.org/dist/spark/spark-2.4.3/spark-2.4.3-bin-hadoop2.7.tgz"
 - Instalar Spark: "sudo tar xvf spark-2.4.3-bin-hadoop2.7.tgz -C/usr/local"
 - Instalar Anaconda: esta posee múltiples herramientas entre las que se encuentra Jupyter: paso a paso para su instalación en el siguiente link: <https://linuxize.com/post/how-to-install-anaconda-on-ubuntu-18-04/>
 - Modificar archivo ~/.bashrc agregando al final de este lo siguiente:

```
export JAVA_HOME=/usr/lib/jvm/default-java
export SBT_HOME=/usr/share/sbt-launcher-packaging/bin/sbt-launch.jar
export SPARK_HOME=/usr/lib/spark
export PATH=$PATH:$JAVA_HOME/bin
```

```
export  
PATH=$PATH:$SBT_HOME/bin:$SPARK_HOME/bin:$SPARK_HOME/s  
bin  
export PYSPARK_DRIVER_PYTHON=jupyter  
export PYSPARK_DRIVER_PYTHON_OPTS='notebook'  
export PYSPARK_PYTHON=python2.7  
export PYTHONPATH=$SPARK_HOME/python:$PYTHONPATH
```

- Crear un Buckets de S3 de acceso público
- vincular S3 con spark.