|  |
| --- |
| Máster en Big Data |
| Tecnologías de Almacenamiento |
| 1. Hands-On: Ejecución Apache Spark |

Presentado por:

José David Angulo y Albert Ripoll

Índice

[1. Introducción 3](#_Toc96241474)

[2. Entorno 3](#_Toc96241475)

[3. Ejecución Spark 3](#_Toc96241476)

[4. SparkUI 4](#_Toc96241477)

# Introducción

El objetivo de este Hands-On es aprender a ejecutar un job de Spark mediante un ejemplo precompilado.

# Entorno

Para la realización de este hands-on, vamos a desplegar un sandbox de MapR en VMWare.

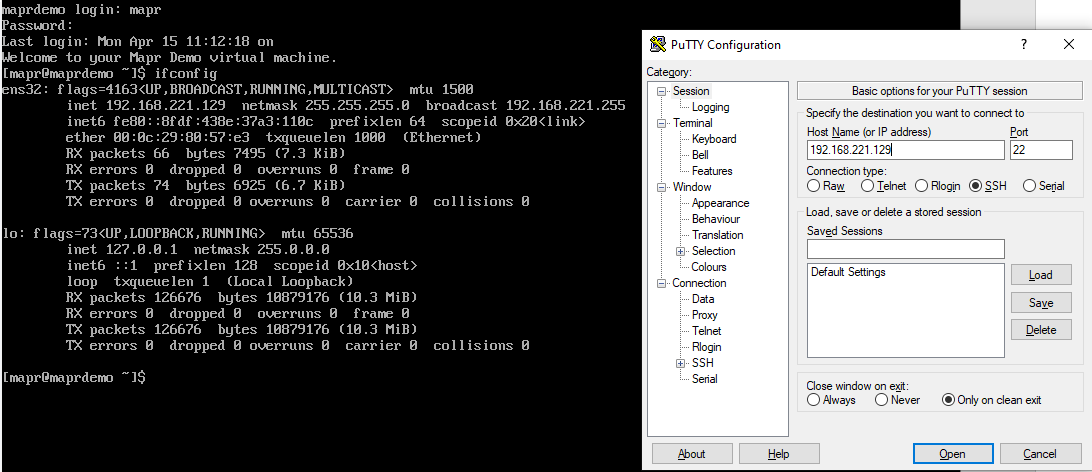
Una vez arrancado el nodo, las credenciales son:

Usuario: *mapr*; Password: *mapr*

Busca la IP de la máquina virtual y accede a HUE http://<ip>:8888/ y abre una conexión ssh para más comodidad (puedes añadir la ip en hosts con el nombre *maprdemo*).

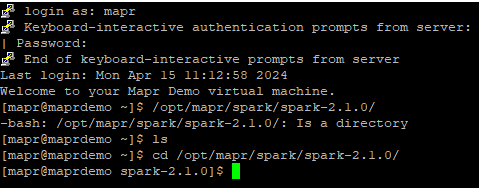
Usuario: *mapr*; Password: *mapr*

ifconfig



Después de ifconfig → la ip de inet la escribimos en host Name or IP address de PuTTY configuration.

192.168.221.131



Volvemos a poner el loging y pass mapr, mapr

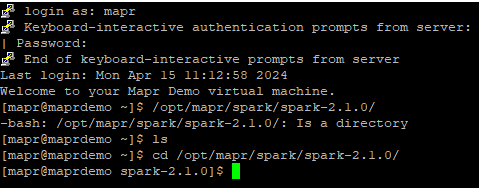
# Ejecución Spark

La ubicación de Spark dentro de la distribución MapR la podemos encontrar en:

/opt/mapr/spark/spark-2.1.0/

Se continua después del loging

cd /opt/mapr/spark/spark-2.1.0/



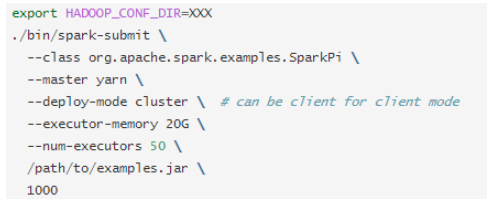
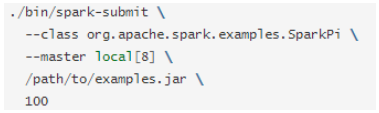
Se pide ejecutar el programa SparkPi, contra YARN en modo cliente, con una memoria por ejecutor de 1g.

El jar ya compilado lo podemos encontrar en: /opt/mapr/spark/spark-2.1.0/examples/jars/spark-examples\_2.11-2.1.0-mapr-1710.jar

Y el package y nombre de la clase es: *org.apache.spark.examples.SparkPi*

El programa acepta por parámetro el número de paralelización que queremos darle, vamos a pasarle, por ejemplo, 4.

Engancha aquí el comando para enviar el Job y una captura de pantalla con la salida.

• En la diapositiva 36 del powerpoint “3. Apache spark” vemos:

• En la diapositiva 37 del powerpoint “3. Apache spark” vemos:

spark-sumit --jars lib/json-simple-1.1.1.jar -- class com.pragsis.ccma.hbbtv.HbbtcProcess

--master yarn-client --executor-memory 15H ccma.jar -i /user/devel/data/pending/ -o /user/devel/data/

• Por lo que para responder a las condiciones de la práctica se elabora el siguiente código para ejecutar.

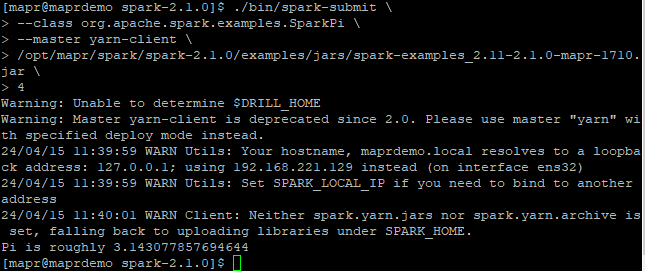
./bin/spark-submit \

--class org.apache.spark.examples.SparkPi \

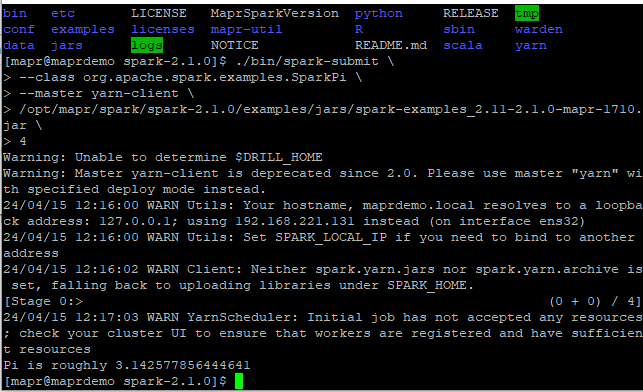
--master yarn-client \

/opt/mapr/spark/spark-2.1.0/examples/jars/spark-examples\_2.11-2.1.0-mapr-1710.jar \

4



Después se cuelga con la ip original que era 192.168.221.129 y se vuelve a repetir todo otra vez con lo que cambia la ip a 192.168.221.131 y sale



Después de colgarse otra vez se repite el proceso y la nueva ip es a 192.168.221.132

Se tiene que repetir para 2, 4 y 8 jobs.

# SparkUI

Consulta la *SparkUI* después de ejecutar el Job y responde a las siguientes preguntas.

Requisito: No accedas directamente a la interficie del job history de spark, hazlo mediante HUE (acceso user:*mapr* / pass: *mapr*)

¿Cuántas tasks se han ejecutado?

¿Cuántos executors se han ejecutado?

Para el job 4: ¿Cuántos RDDs se han creado y que transformaciones los han desencadenado?

Ejecuta el job con 2, 4 y 10 tasks y compara el rendimiento, ¿Que sucede en relación con la performance? ¿Porque se produce este comportamiento?

**DETALLES CON Jobs = 2**

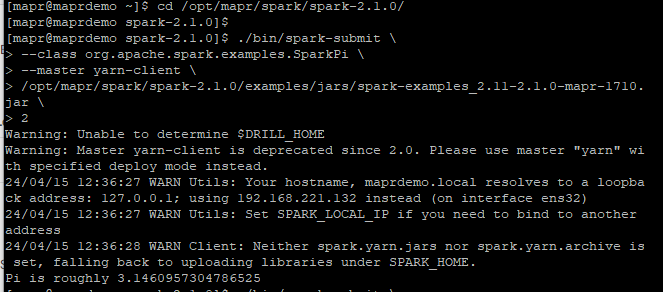
./bin/spark-submit \

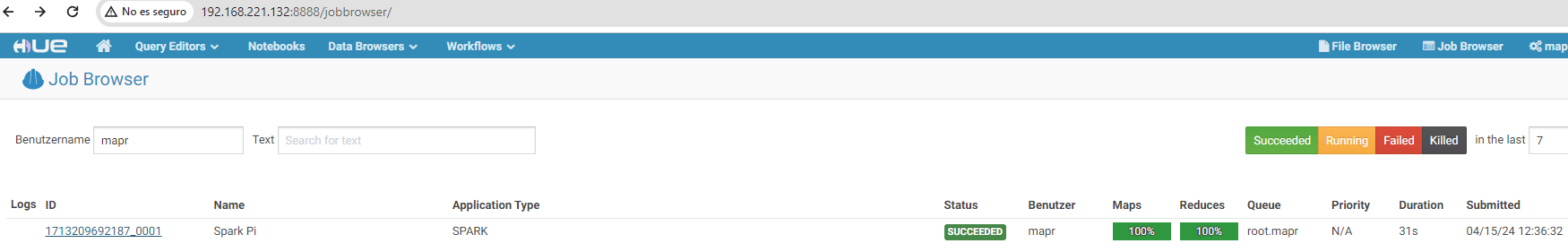
--class org.apache.spark.examples.SparkPi \

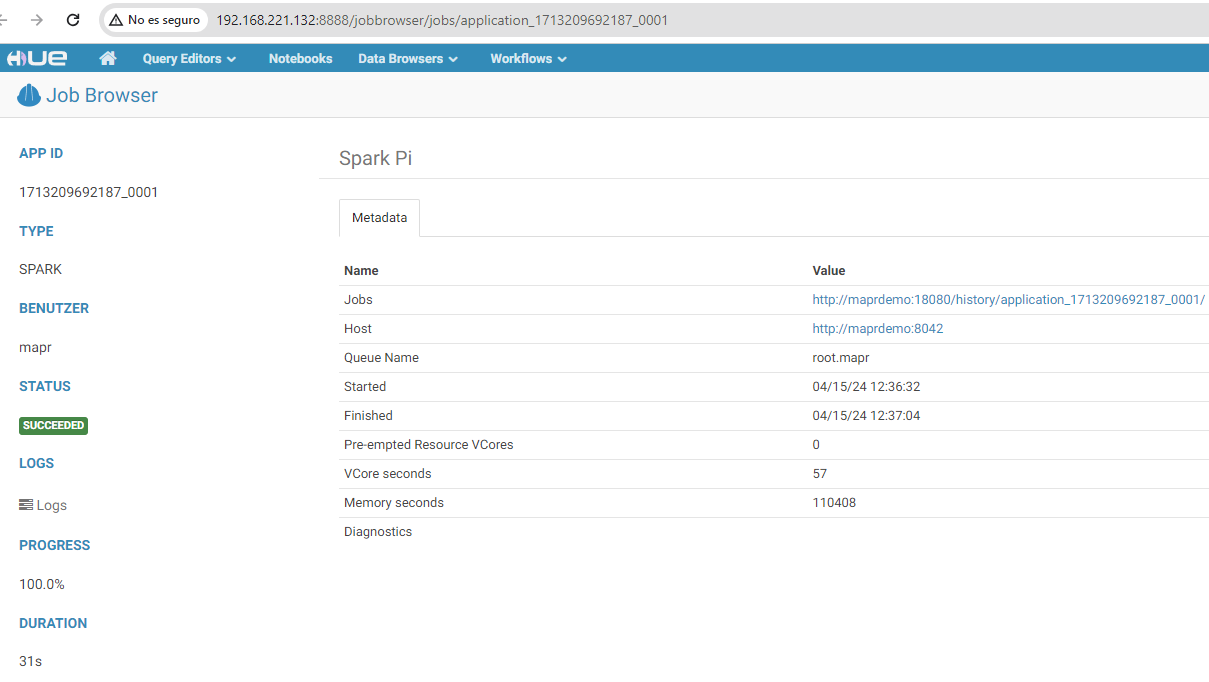
--master yarn-client \

/opt/mapr/spark/spark-2.1.0/examples/jars/spark-examples\_2.11-2.1.0-mapr-1710.jar \

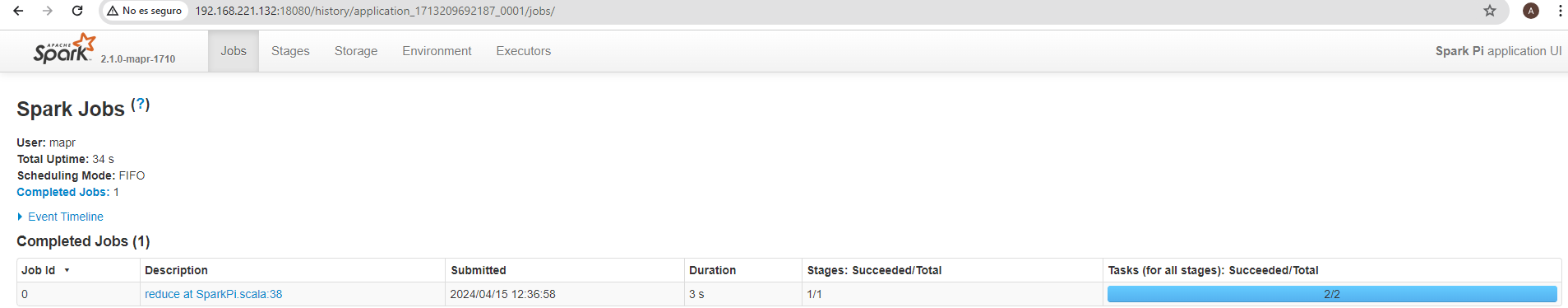
2



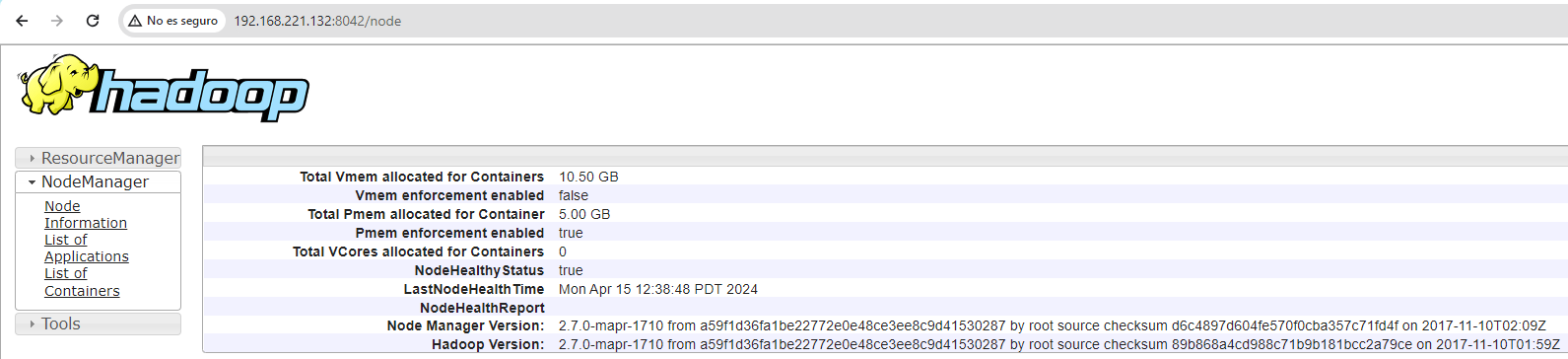
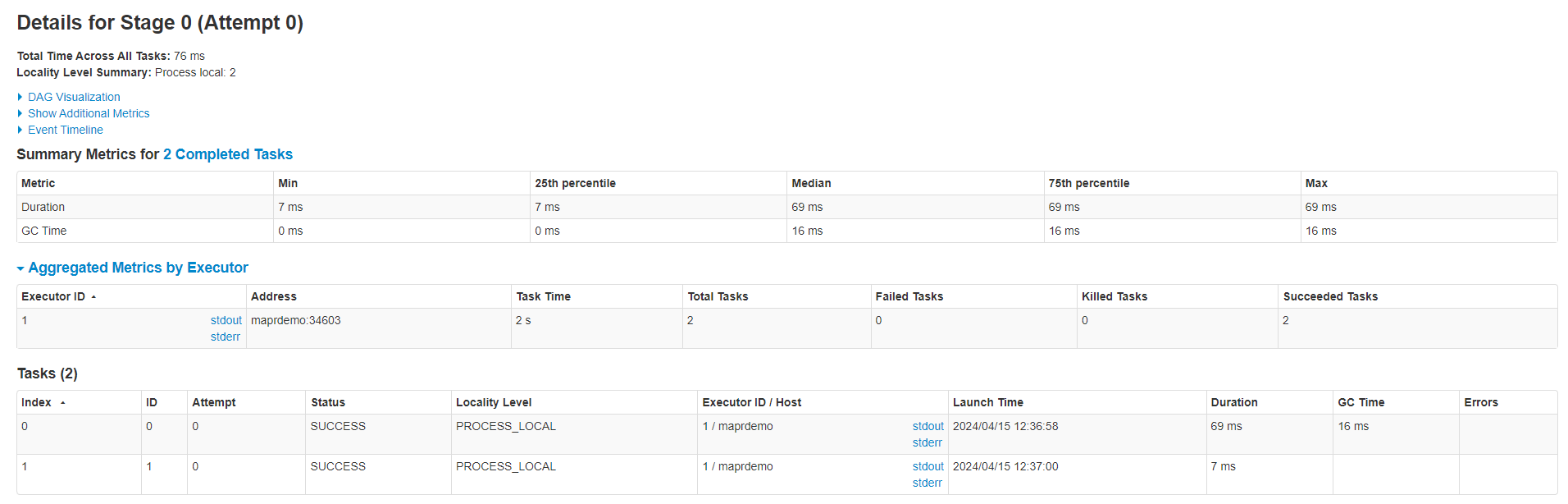




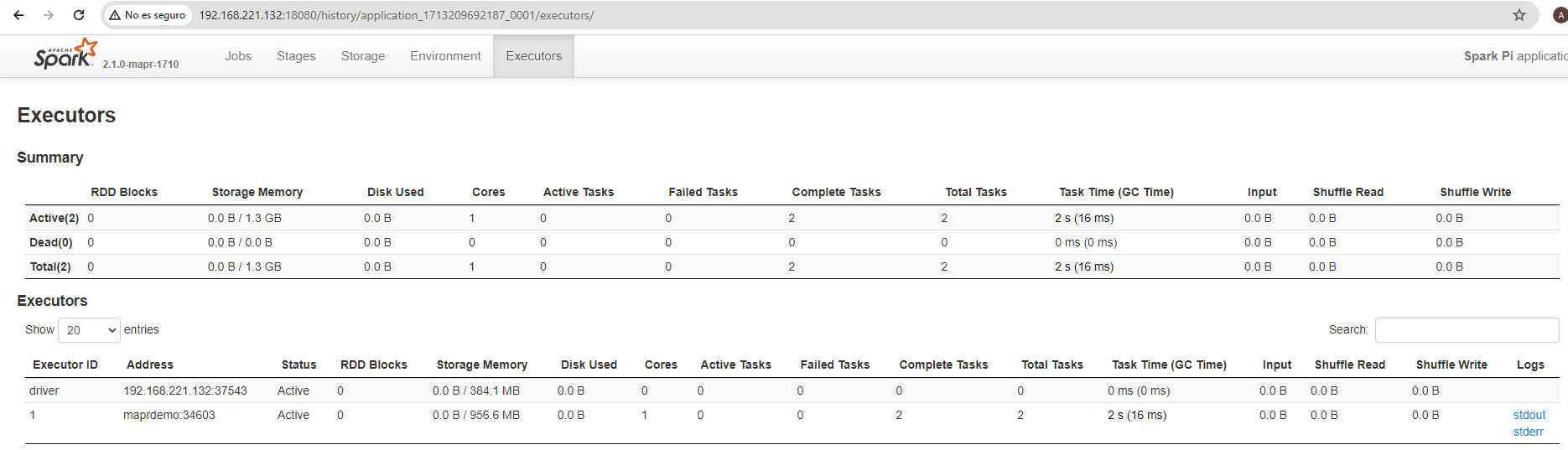
Para el job 2: ¿Cuántas tasks se han ejecutado? Dos

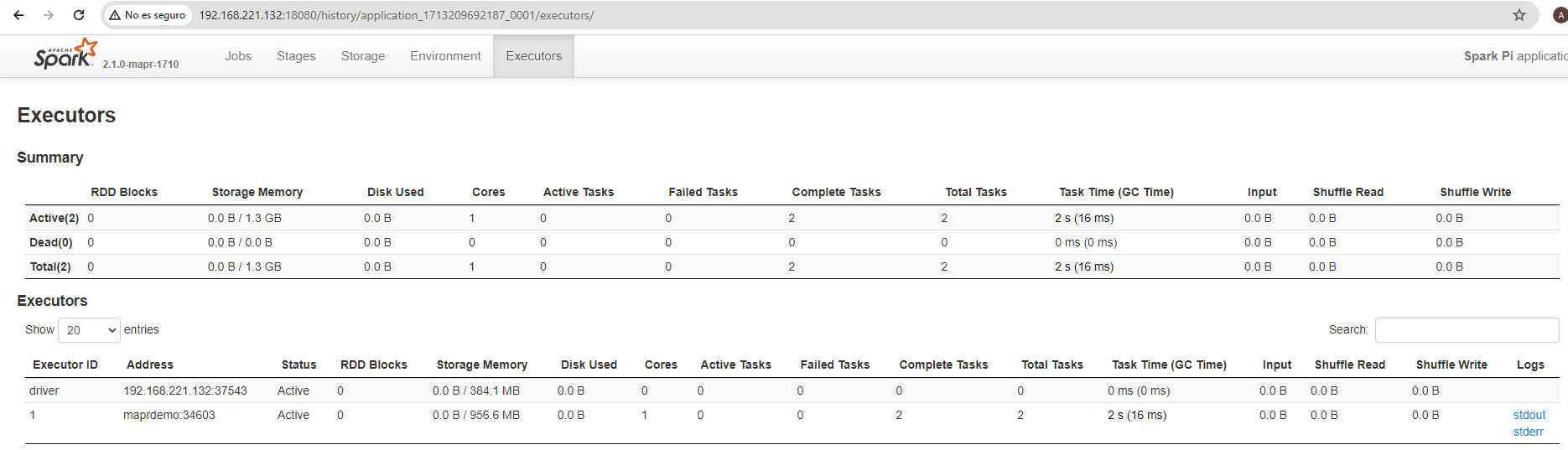


Al darle a reduce at SparkPi.scala:38

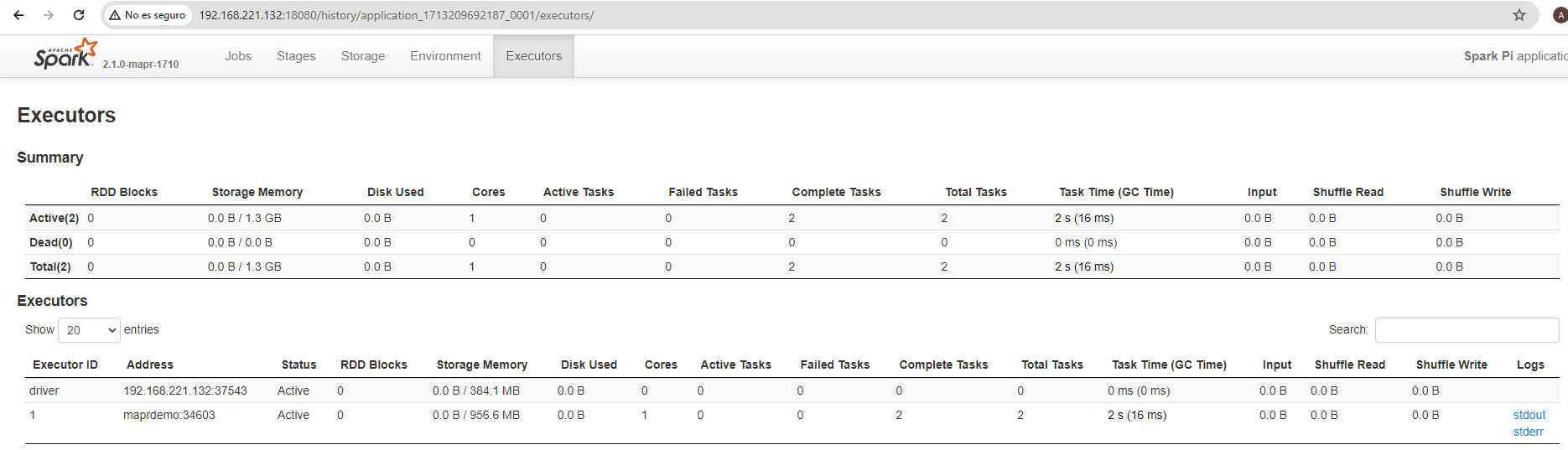


Para el job 2: ¿Cuántos executors se han ejecutado?





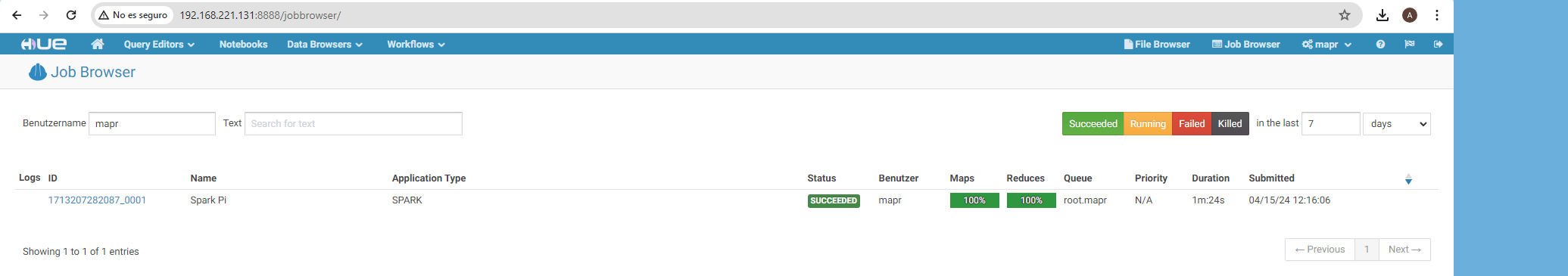
Para el job 2: ¿Cuántos RDDs se han creado y que transformaciones los han desencadenado?



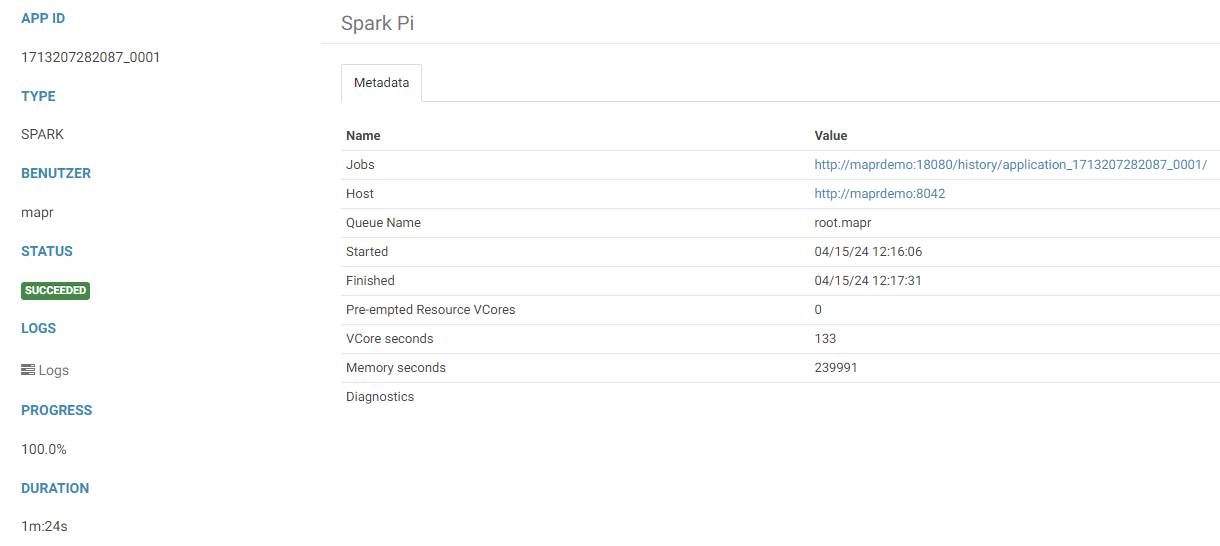
**DETALLES CON Jobs = 4**

Entonces en el Google Chrome se escribe http://192.168.221.131:8888/

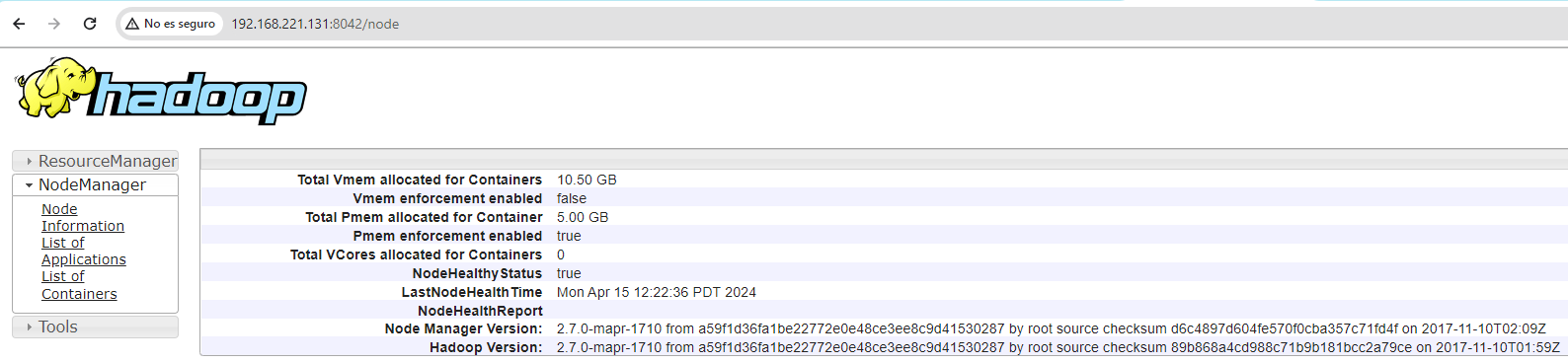
Y en job browser vemos el job.

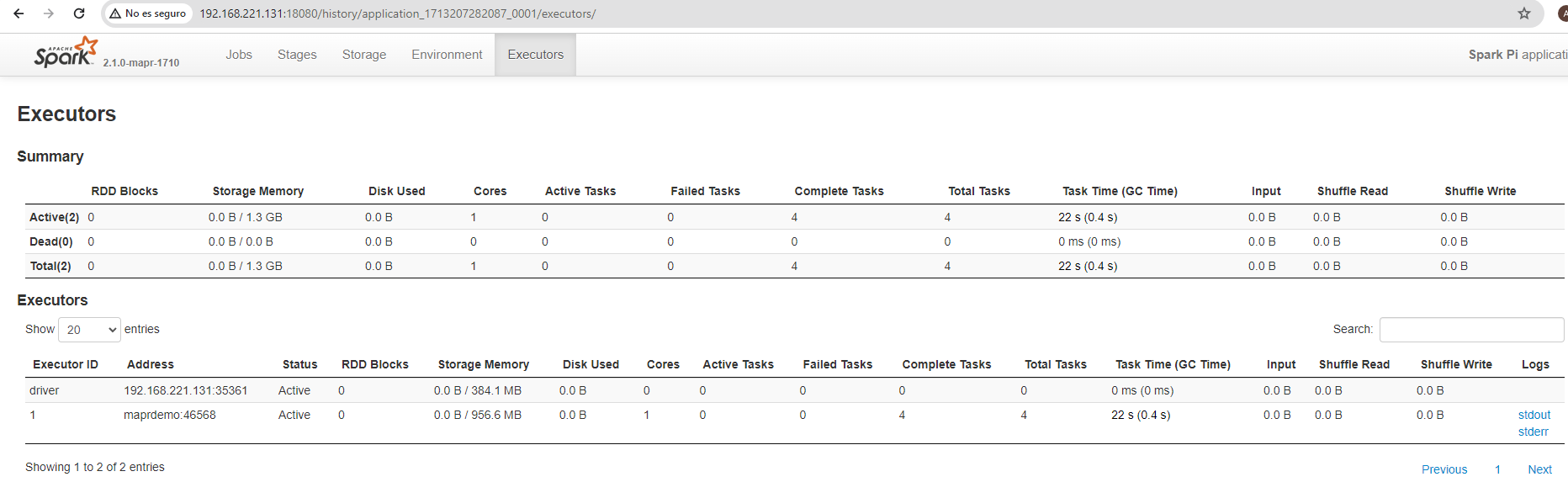


# En view job vemos:

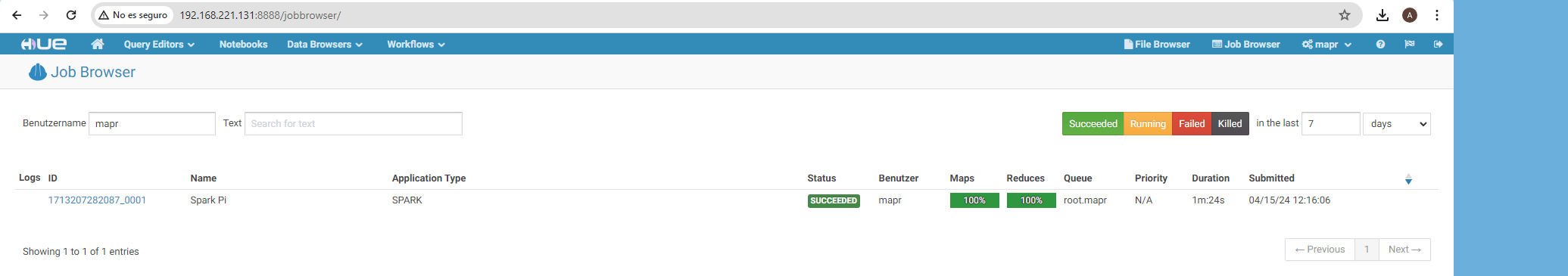


Y su acedemos a los jobs cambiando maprdemo con la ip.

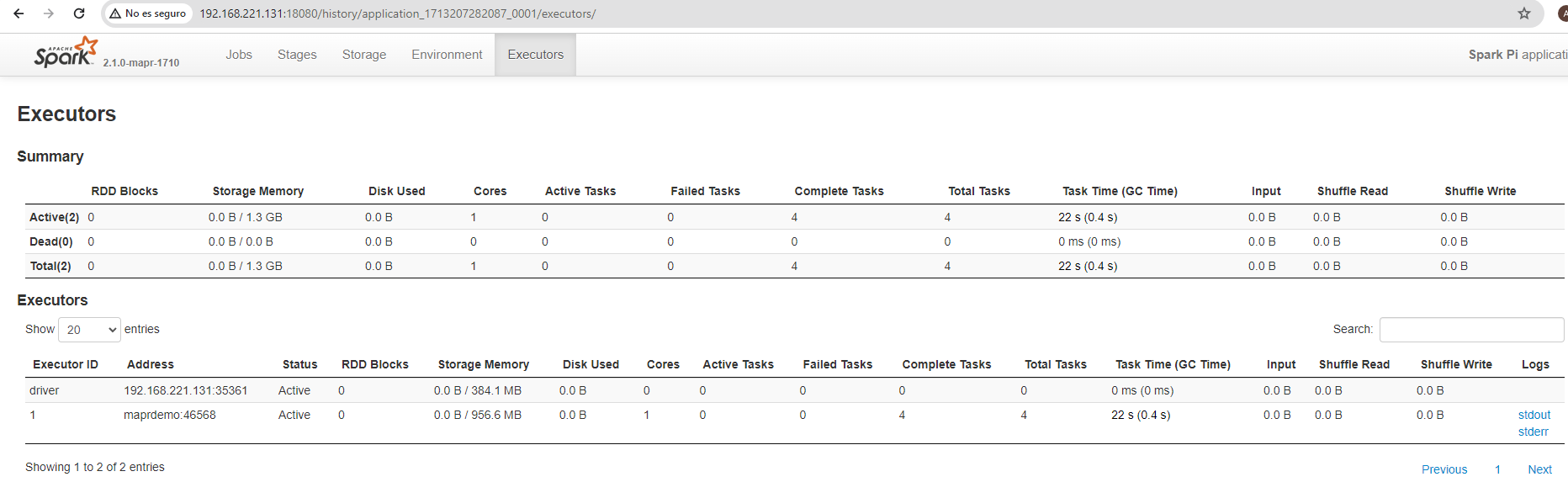




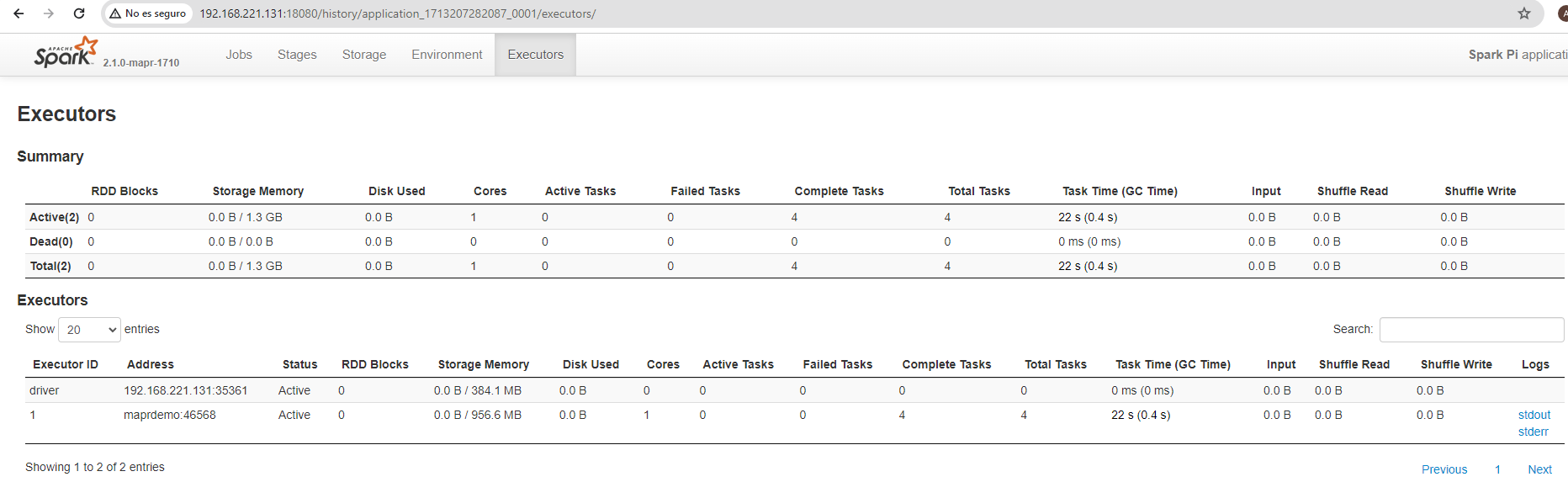
Para el job 4: ¿Cuántas tasks se han ejecutado? Una



Para el job 4: ¿Cuántos executors se han ejecutado? Se han ejecutado dos. Un driver y un maprdemo



Para el job 4: ¿Cuántos RDDs se han creado y que transformaciones los han desencadenado? Se han creado dos que están en activo.



**DETALLES CON Jobs = 10**

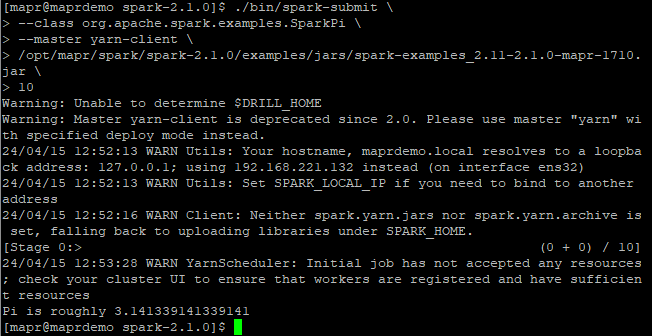
./bin/spark-submit \

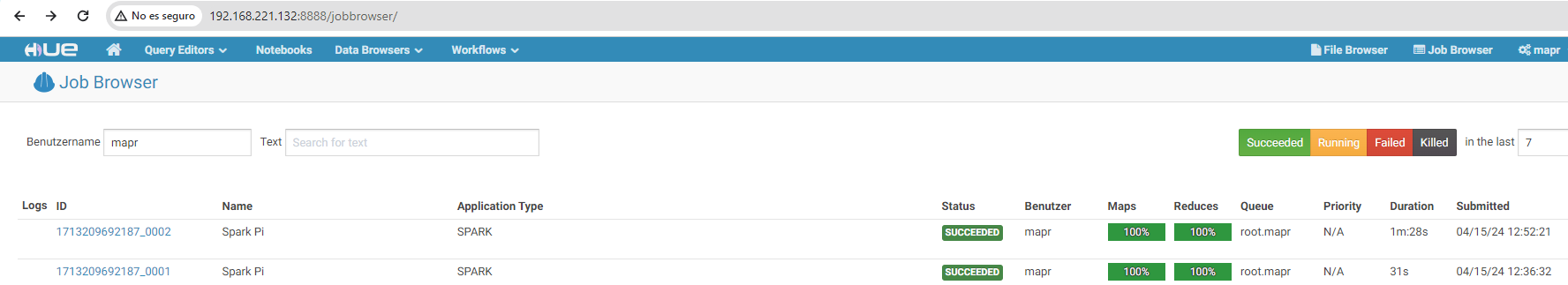
--class org.apache.spark.examples.SparkPi \

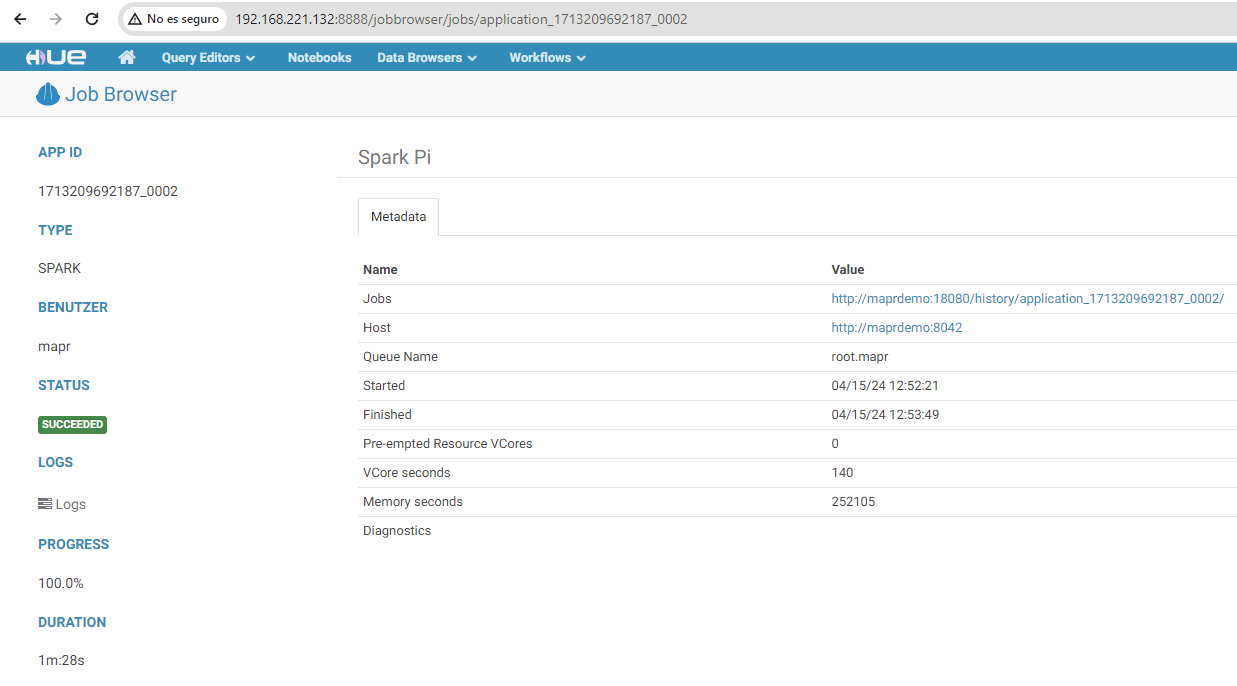
--master yarn-client \

/opt/mapr/spark/spark-2.1.0/examples/jars/spark-examples\_2.11-2.1.0-mapr-1710.jar \

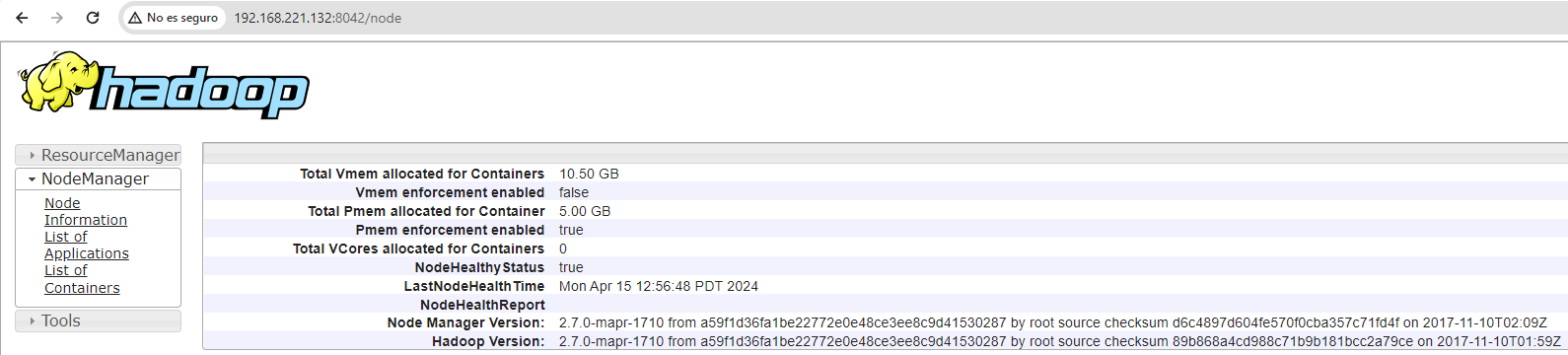
10

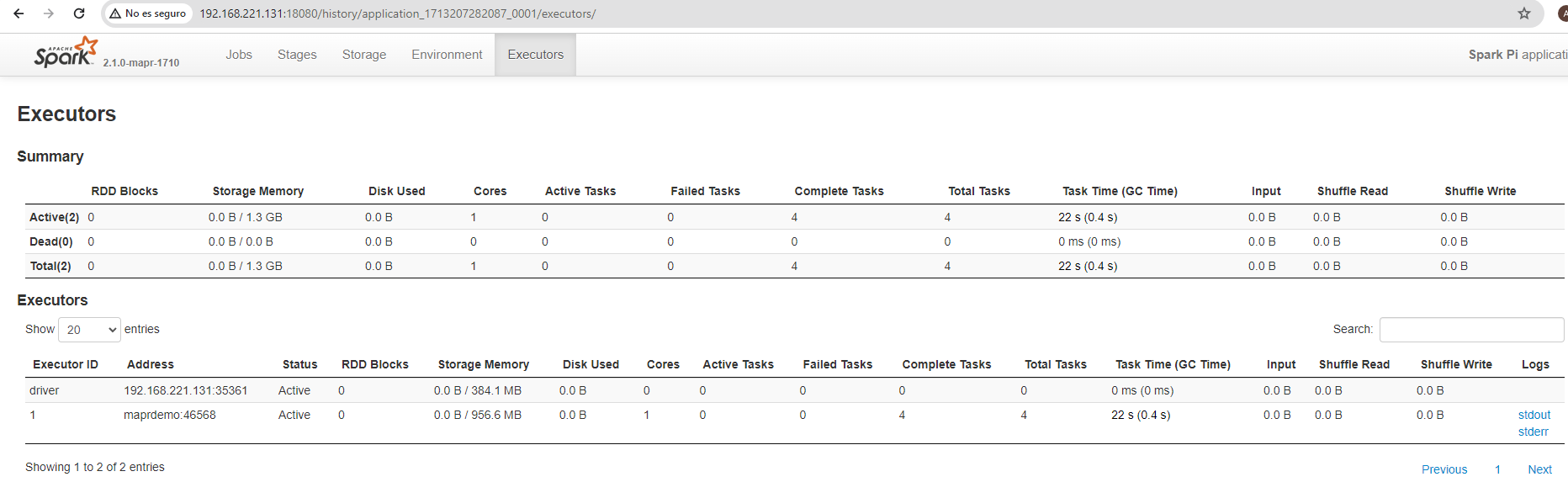




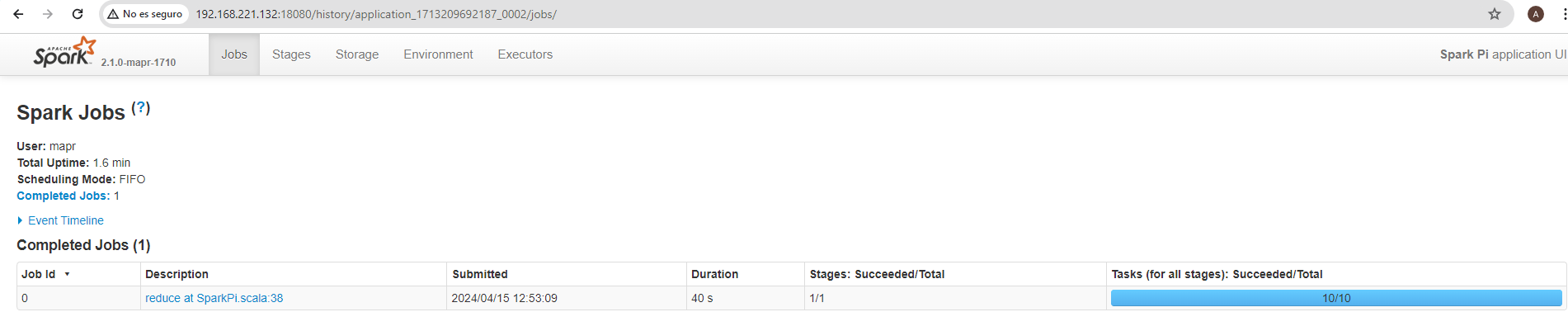


Y su acedemos a los jobs cambiando maprdemo con la ip.

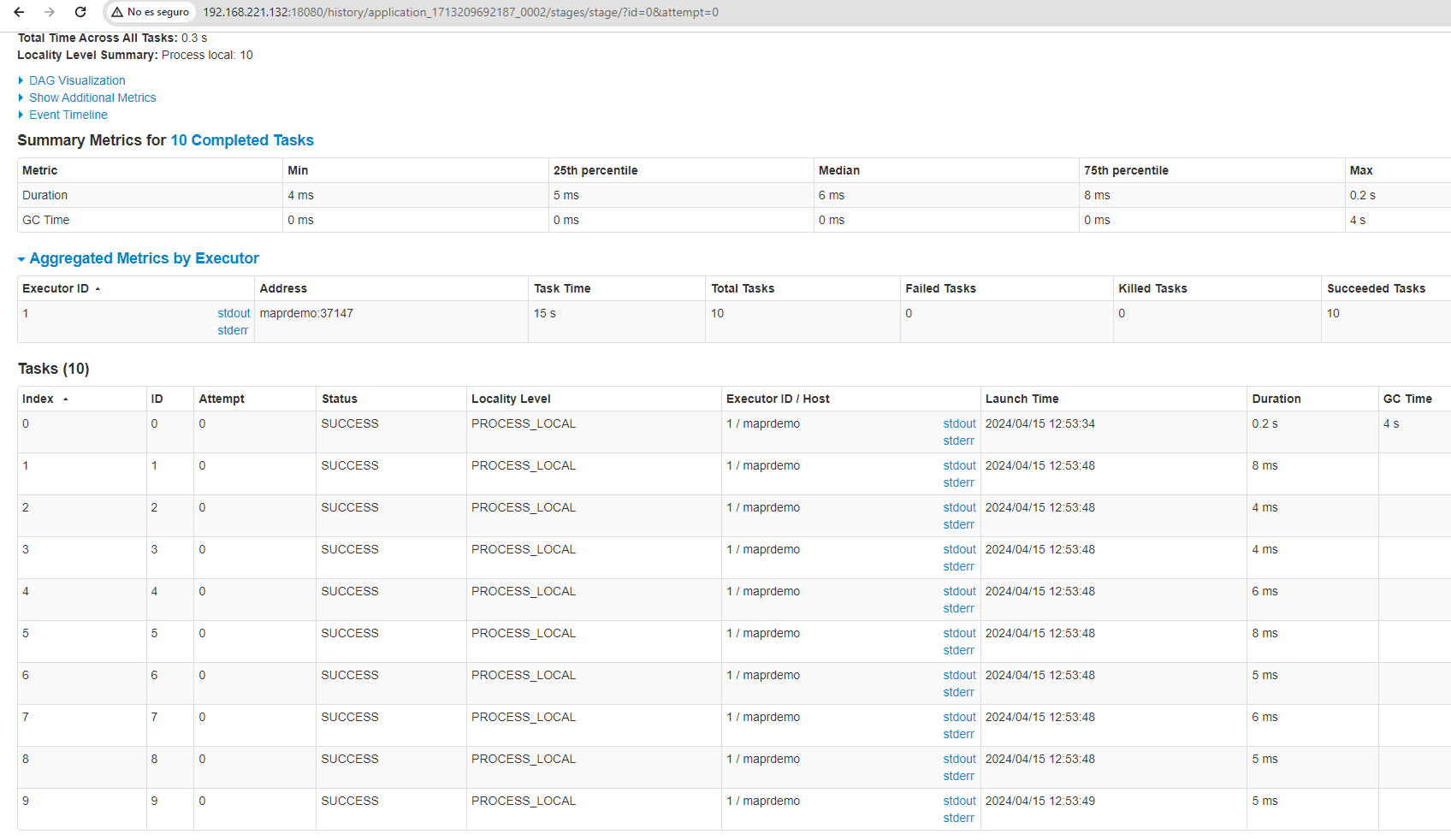




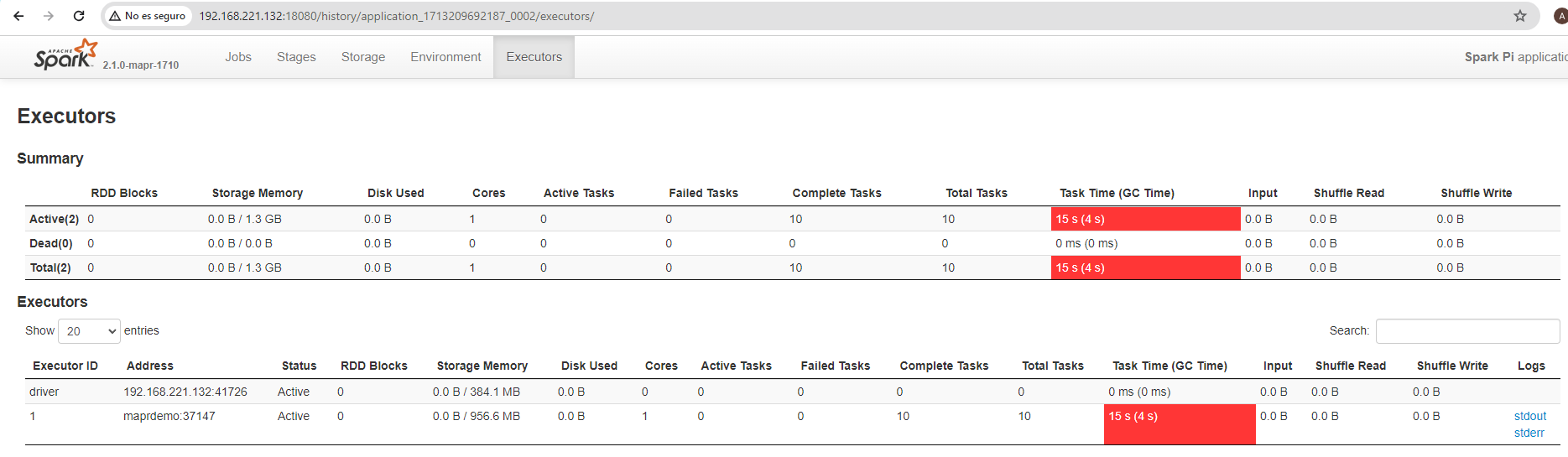
Para el job 10: ¿Cuántas tasks se han ejecutado? Diez

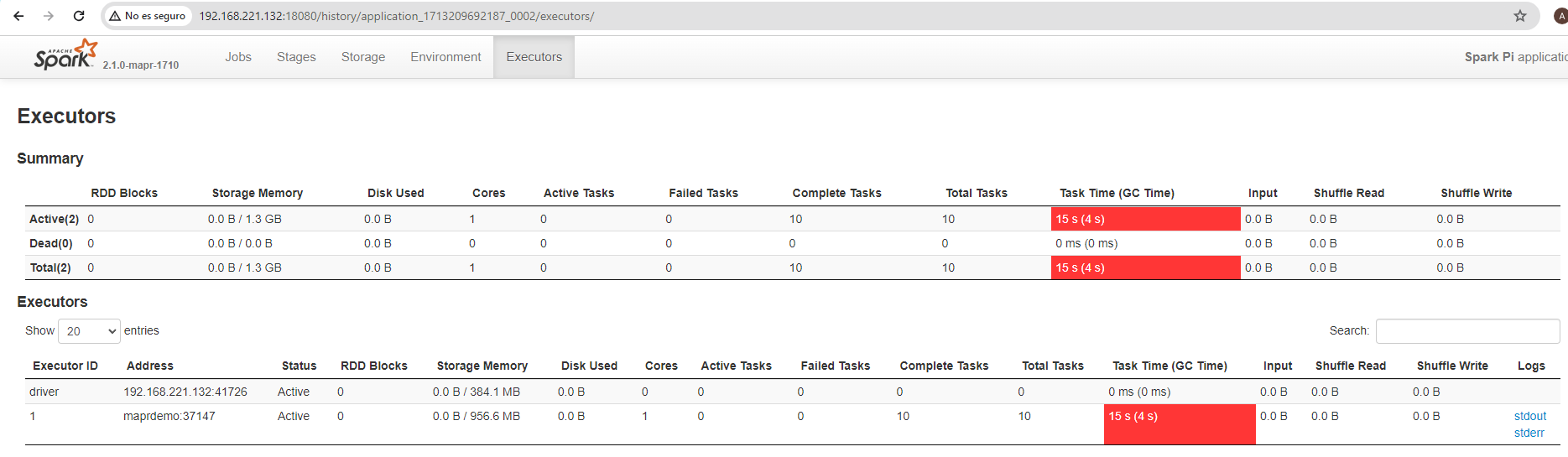


Detalles



Para el job 10: ¿Cuántos executors se han ejecutado?





Para el job 10: ¿Cuántos RDDs se han creado y que transformaciones los han desencadenado?

