Engenharia de Computação

Especialização Lato Sensu em Ciência de Dados e Analytics

Soluções em Processamento para Big Data

{ Prática Google Data Proc }

Prof. Jairson Rodrigues
jairson.rodrigues@univasf.edu.br

{ google data proc }

AGENDA

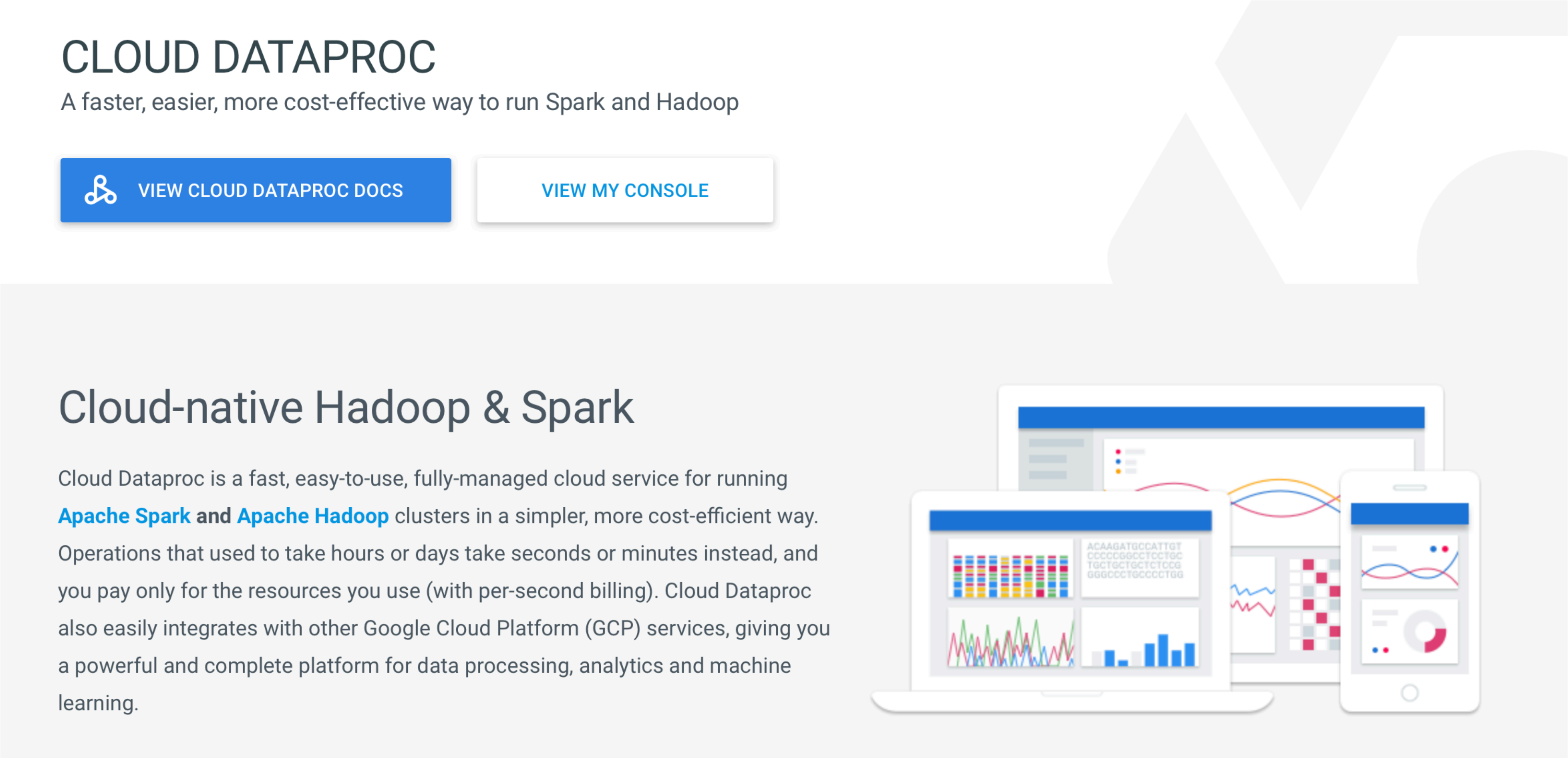
Cluster Google

{ roteiro }

- Cadastro no Google DataProc
- Construir um cluster de quatro máquinas
- Acessar o Name Node
- Executar algoritmos de ML
    - ETL
    - Regressão Logística
    - Árvores de Decisão
    - Kmeans

{ google data proc }

https://cloud.google.com/dataproc/



CLOUD DATAPROC

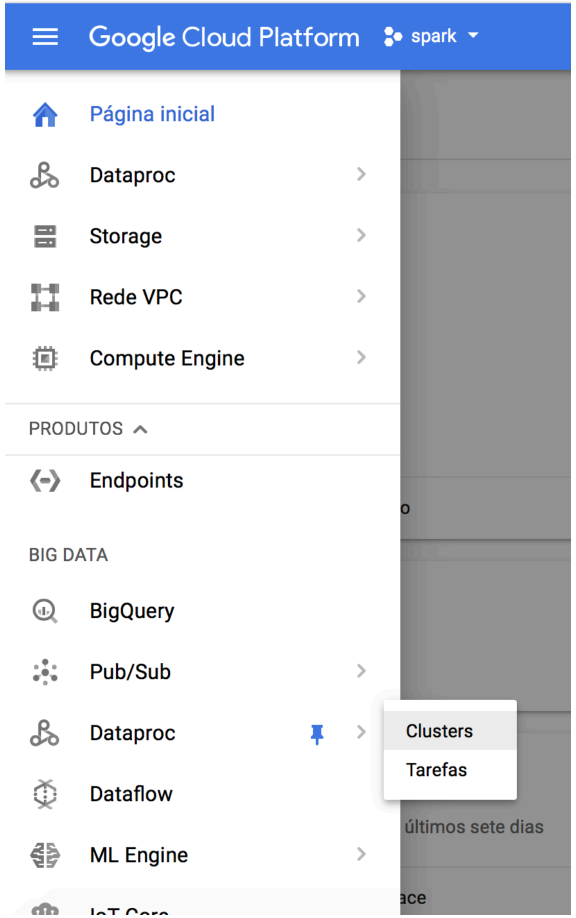
A faster, easier, more cost-effective way to run Spark and Hadoop

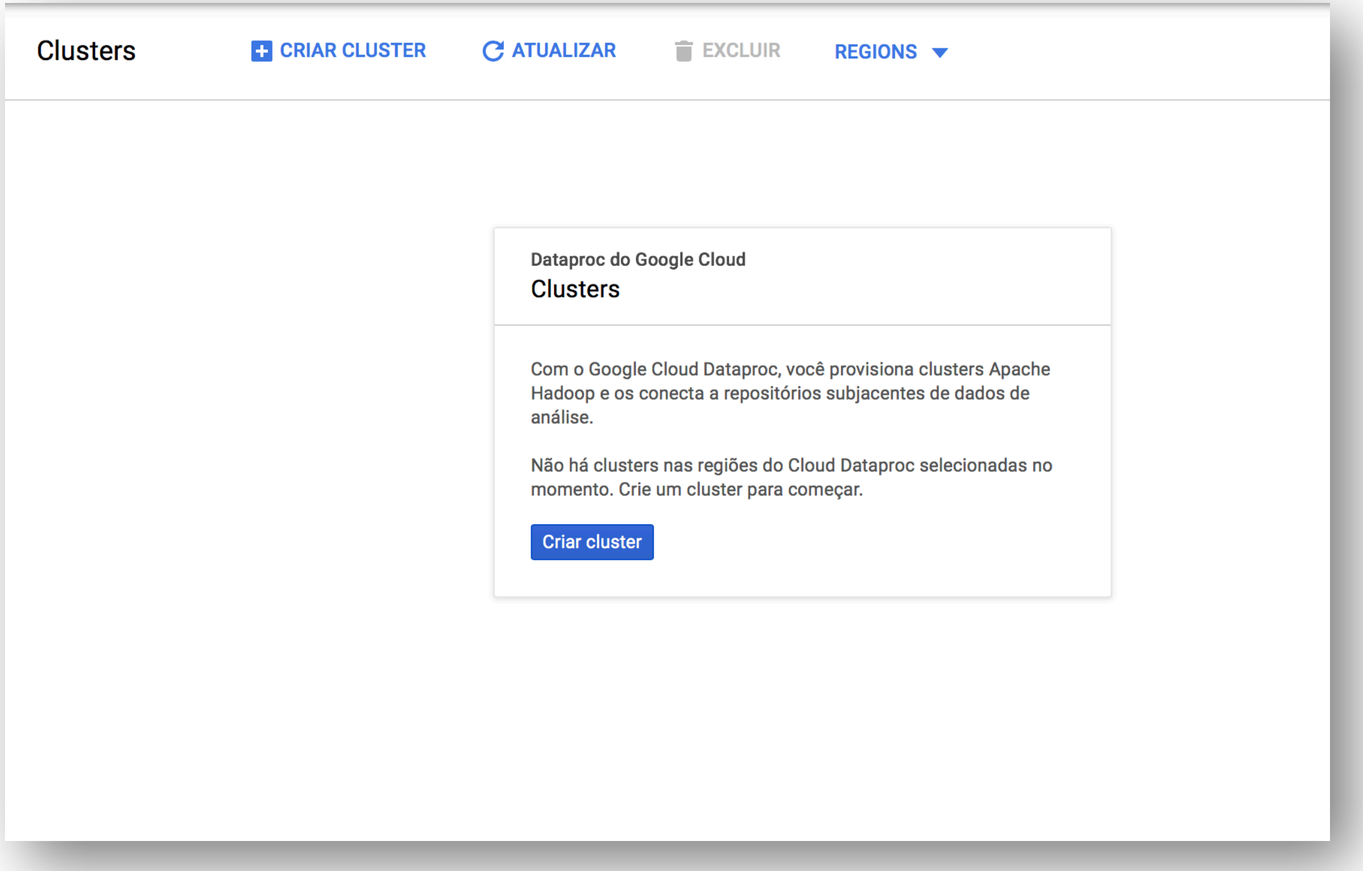
Cloud-native Hadoop & Spark

Cloud Dataproc is a fast, easy-to-use, fully-managed cloud service for running **Apache Spark and Apache Hadoop** clusters in a simpler, more cost-efficient way. Operations that used to take hours or days take seconds or minutes instead, and you pay only for the resources you use (with per-second billing). Cloud Dataproc also easily integrates with other Google Cloud Platform (GCP) services, giving you a powerful and complete platform for data processing, analytics and machine learning.

{ google data proc }

https://console.cloud.google.com/dataproc/





Dataproc do Google Cloud

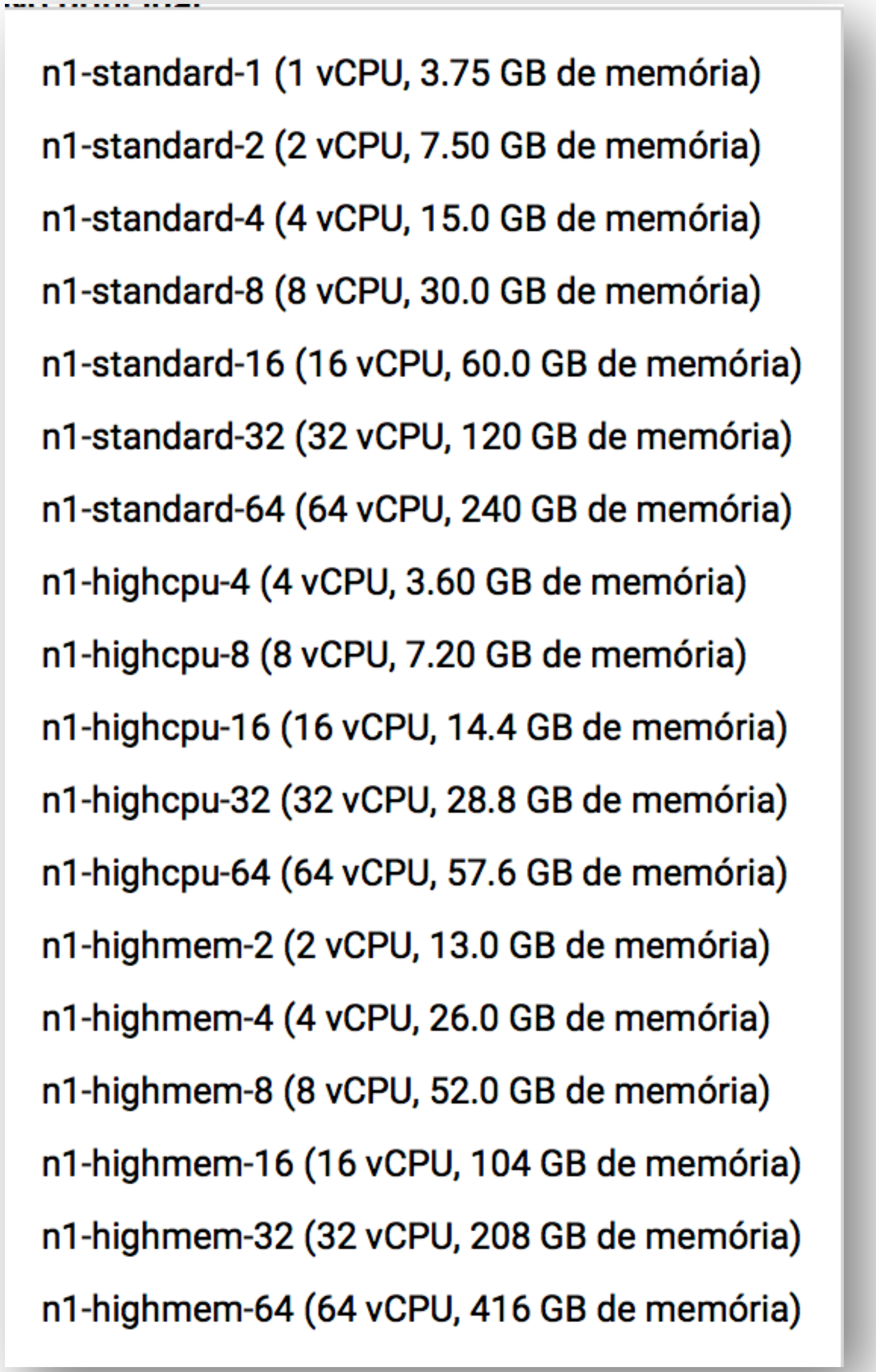
Clusters

Com o Google Cloud Dataproc, você provisiona clusters Apache Hadoop e os conecta a repositórios subjacentes de dados de análise.

Não há clusters nas regiões do Cloud Dataproc selecionadas no momento. Crie um cluster para começar.

{ google data proc }

Tipos de Máquina Virtual



n1-standard-1 (1 vCPU, 3.75 GB de memória)

n1-standard-2 (2 vCPU, 7.50 GB de memória)

n1-standard-4 (4 vCPU, 15.0 GB de memória)

n1-standard-8 (8 vCPU, 30.0 GB de memória)

n1-standard-16 (16 vCPU, 60.0 GB de memória)

n1-standard-32 (32 vCPU, 120 GB de memória)

n1-standard-64 (64 vCPU, 240 GB de memória)

n1-highcpu-4 (4 vCPU, 3.60 GB de memória)

n1-highcpu-8 (8 vCPU, 7.20 GB de memória)

n1-highcpu-16 (16 vCPU, 14.4 GB de memória)

n1-highcpu-32 (32 vCPU, 28.8 GB de memória)

n1-highcpu-64 (64 vCPU, 57.6 GB de memória)

n1-highmem-2 (2 vCPU, 13.0 GB de memória)

n1-highmem-4 (4 vCPU, 26.0 GB de memória)

n1-highmem-8 (8 vCPU, 52.0 GB de memória)

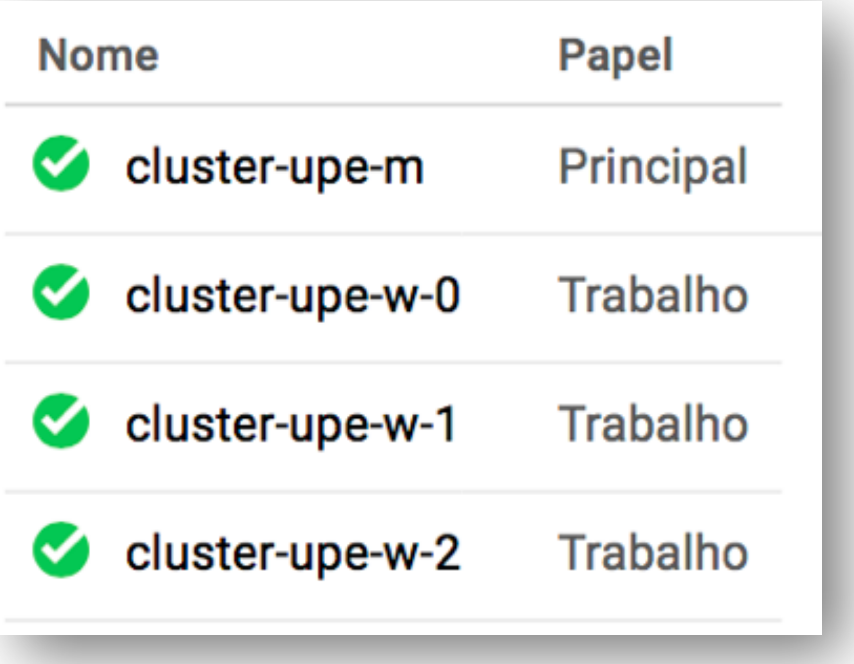
n1-highmem-16 (16 vCPU, 104 GB de memória)

n1-highmem-32 (32 vCPU, 208 GB de memória)

n1-highmem-64 (64 vCPU, 416 GB de memória)

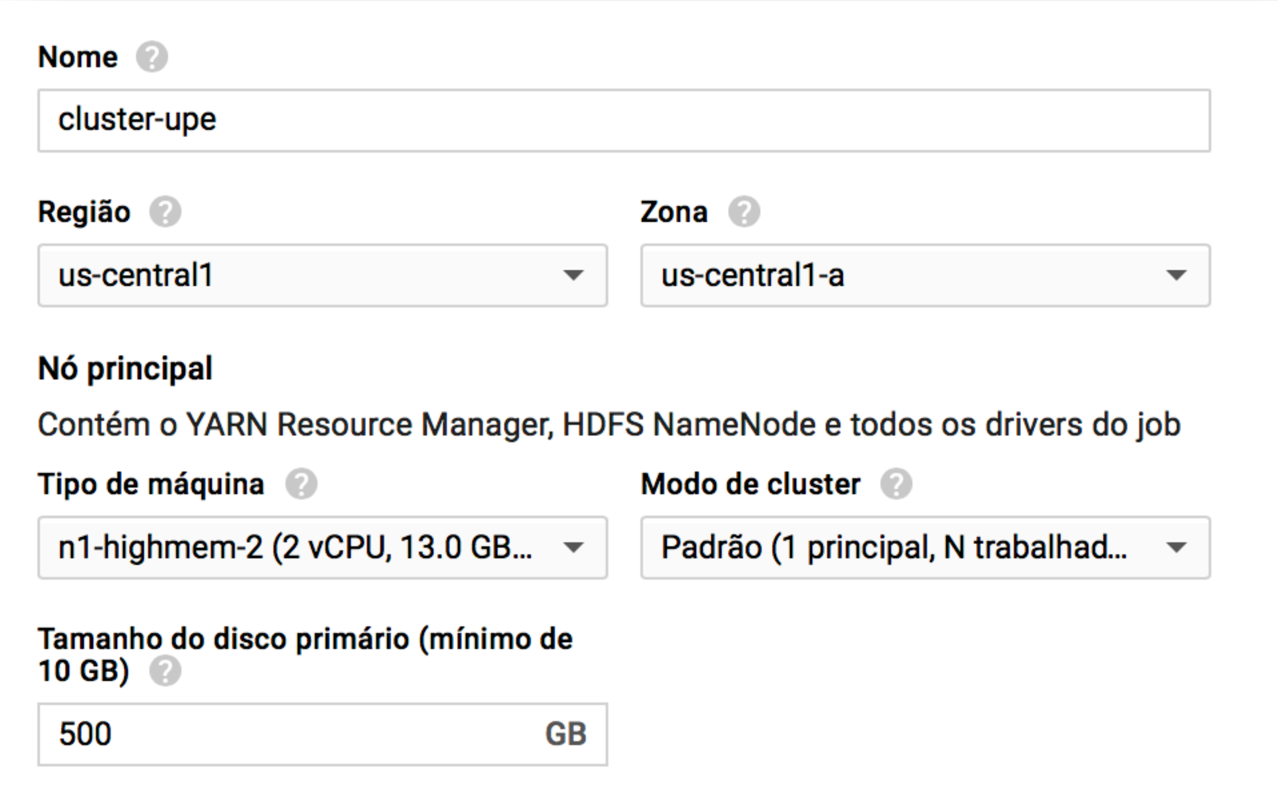
{ configuração de nós }

- Master: Google VM Machine
  - n1-highmem-2
  - CPU: 2
  - RAM: 13 Gb RAM
  - Disco local: 500 Gb
  - # instâncias: 1
- Slave: Google VM Machine
  - n1-highmem-2
  - CPU: 2
  - RAM: 13 Gb RAM
  - Disco local: 500 Gb
  - Disco SSD 375 Gb
  - # instâncias: 3



	Nome	Papel
✓	cluster-upe-m	Principal
✓	cluster-upe-w-0	Trabalho
✓	cluster-upe-w-1	Trabalho
✓	cluster-upe-w-2	Trabalho

{ google data proc – master }



Nome

cluster-upe

Região

us-central1

Zona

us-central1-a

Nó principal

Contém o YARN Resource Manager, HDFS NameNode e todos os drivers do job

Tipo de máquina

n1-highmem-2 (2 vCPU, 13.0 GB...

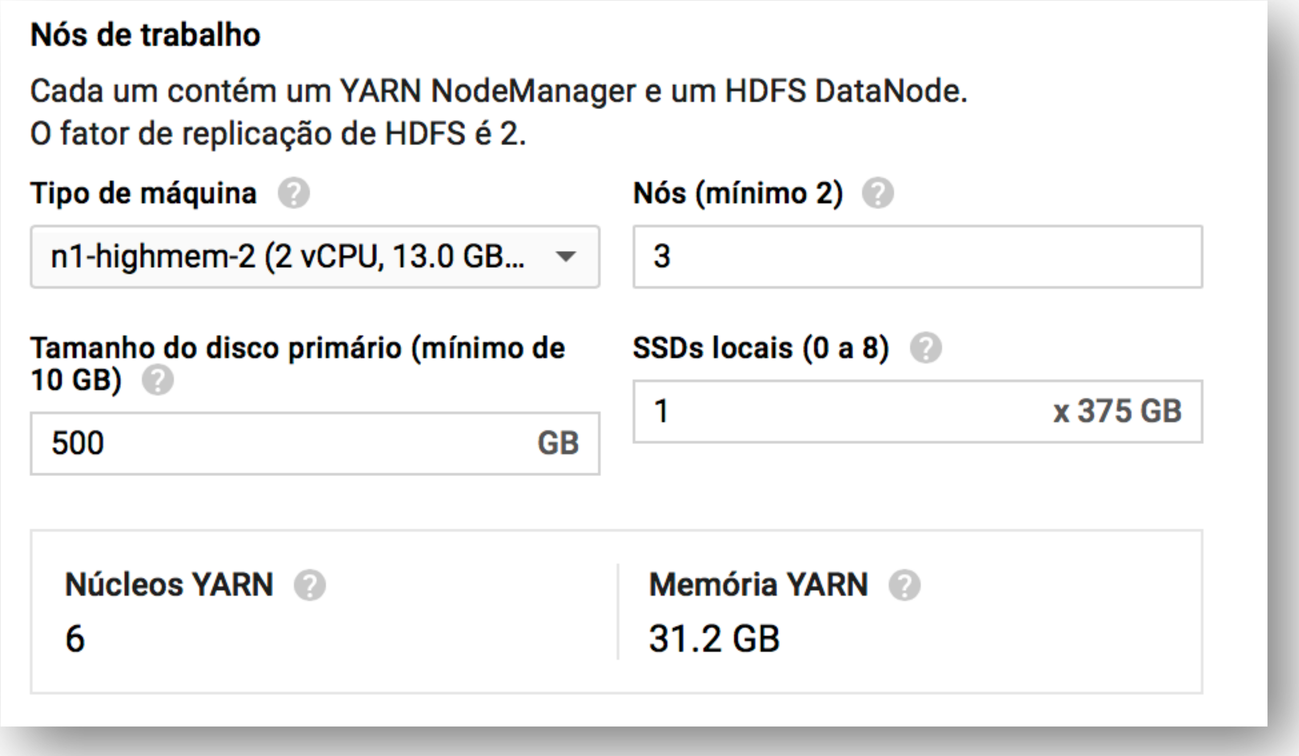
Modo de cluster

Padrão (1 principal, N trabalhad...

Tamanho do disco primário (mínimo de 10 GB)

500 GB

{ google data proc – slaves }



Nós de trabalho

Cada um contém um YARN NodeManager e um HDFS DataNode.
O fator de replicação de HDFS é 2.

Tipo de máquina: n1-highmem-2 (2 vCPU, 13.0 GB...

Nós (mínimo 2): 3

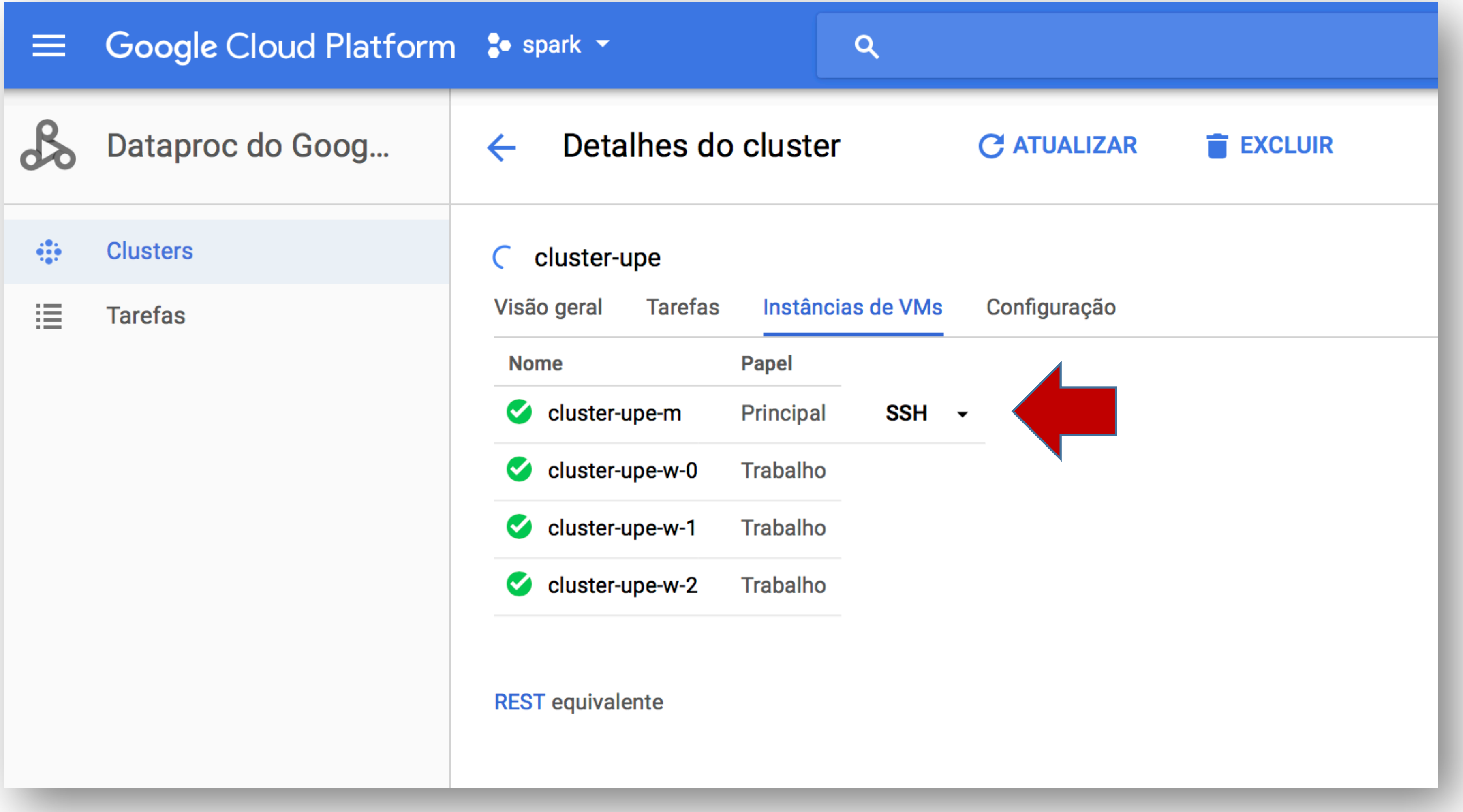
Tamanho do disco primário (mínimo de 10 GB): 500 GB

SSDs locais (0 a 8): 1 x 375 GB

Núcleos YARN: 6

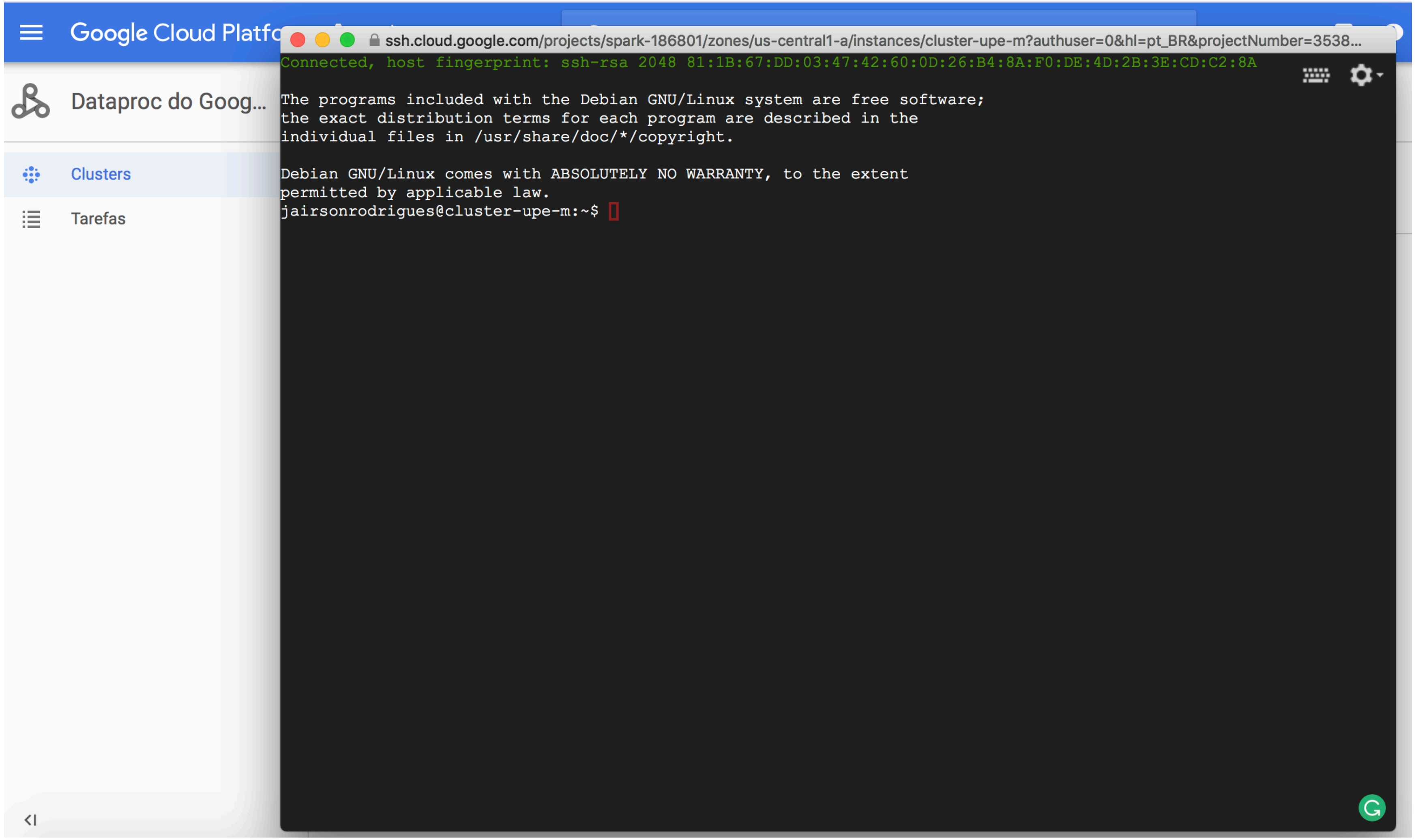
Memória YARN: 31.2 GB

{ detalhes do cluster }



Nome	Papel
✓ cluster-upe-m	Principal
✓ cluster-upe-w-0	Trabalho
✓ cluster-upe-w-1	Trabalho
✓ cluster-upe-w-2	Trabalho

{ conexão SSH }


Google Cloud Platfo
Dataproc do Goog...
Clusters
Tarefas
ssh.cloud.google.com/projects/spark-186801/zones/us-central1-a/instances/cluster-upe-m?authuser=0&hl=pt_BR&projectNumber=3538...

```
Connected, host fingerprint: ssh-rsa 2048 81:1B:67:DD:03:47:42:60:0D:26:B4:8A:F0:DE:4D:2B:3E:CD:C2:8A

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
jairsonrodrigues@cluster-upe-m:~$
```

{ configurações iniciais }

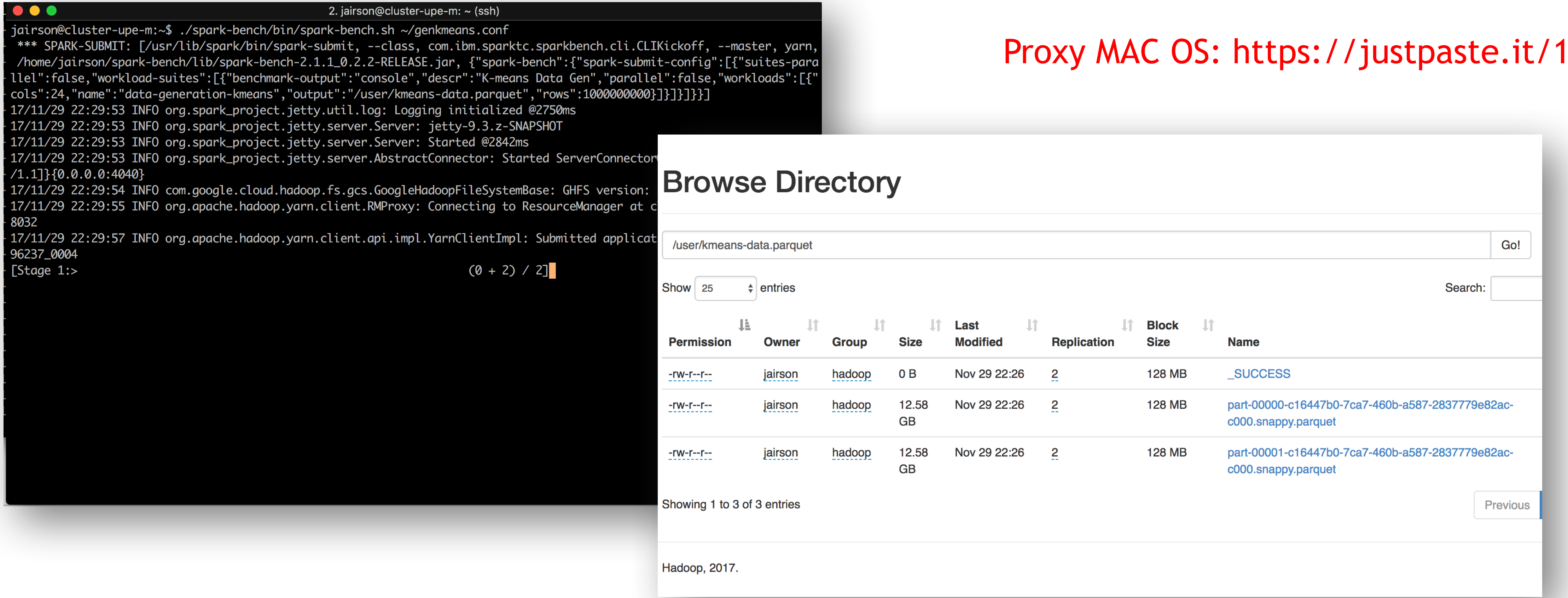
- wget https://www.dropbox.com/s/r5xg2hi28g4s51f/kddcup_2.11-1.0.jar?dl=0
- wget https://github.com/SparkTC/spark-bench/releases/download/v55/spark-bench_2.1.1_0.2.2-RELEASE_55.tgz
- wget https://www.dropbox.com/s/0hi816m22uia2cw/spark-bench-env.sh?dl=0
- wget https://www.dropbox.com/s/98udl9vmi1fayzq/genkmeans.conf?dl=0
- wget https://www.dropbox.com/s/mwnad29s3zdtutx/kmeans.conf?dl=0
- wget http://kdd.ics.uci.edu/databases/kddcup99/kddcup.data.gz
- mv ~/kddcup_2.11-1.0.jar\?dl\=0 kddcup_2.11-1.0.jar
- mv ~/kmeans.conf?dl=0 ~/kmeans.conf
- mv ~/genkmeans.conf?dl=0 ~/genkmeans.conf
- gunzip kddcup.data.gz
- tar -xvf ~/spark-bench_2.1.1_0.2.2-RELEASE_55.tgz
- mv ~/spark-bench_2.1.1_0.2.2-RELEASE ~/spark-bench
- mv ~/spark-bench-env.sh\?dl\=0 ~/spark-bench/bin/spark-bench-env.sh
- mv kddcup.data kddcup.data.10
- hadoop fs -mkdir -p /kddcup/input/
- hadoop fs -copyFromLocal kddcup.data.10 /kddcup/input/

```
wget https://www.dropbox.com/s/hnfuinu3oose116/init.sh?dl=0
mv init.sh\?dl\=0 init.sh
chmod a+x init.sh; ./init.sh
```

{ geração de dados - kmeans -> 25 gb }

- ./spark-bench/bin/spark-bench.sh ~/genkmeans.conf

Proxy MAC OS: https://justpaste.it/1e30d



2. jairson@cluster-upe-m: ~ (ssh)

```
jairson@cluster-upe-m:~$ ./spark-bench/bin/spark-bench.sh ~/genkmeans.conf
 *** SPARK-SUBMIT: [/usr/lib/spark/bin/spark-submit, --class, com.ibm.sparktc.sparkbench.cli.CLIKickoff, --master, yarn,
 /home/jairson/spark-bench/lib/spark-bench-2.1.1_0.2.2-RELEASE.jar, {"spark-bench":{"spark-submit-config":[{"suites-para
llel":false,"workload-suites":[{"benchmark-output":"console","descr":"K-means Data Gen","parallel":false,"workloads":[{"
cols":24,"name":"data-generation-kmeans","output":"/user/kmeans-data.parquet","rows":1000000000}]}]}]}}]
17/11/29 22:29:53 INFO org.spark_project.jetty.util.log: Logging initialized @2750ms
17/11/29 22:29:53 INFO org.spark_project.jetty.server.Server: jetty-9.3.z-SNAPSHOT
17/11/29 22:29:53 INFO org.spark_project.jetty.server.Server: Started @2842ms
17/11/29 22:29:53 INFO org.spark_project.jetty.server.AbstractConnector: Started ServerConnector
/1.1]}{0.0.0.0:4040}
17/11/29 22:29:54 INFO com.google.cloud.hadoop.fs.gcs.GoogleHadoopFileSystemBase: GHFS version:
17/11/29 22:29:55 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at c
8032
17/11/29 22:29:57 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted applicat
96237_0004
[Stage 1:>                                                  (0 + 2) / 2]
```

Browse Directory

/user/kmeans-data.parquet

Show 25 entries

Search:

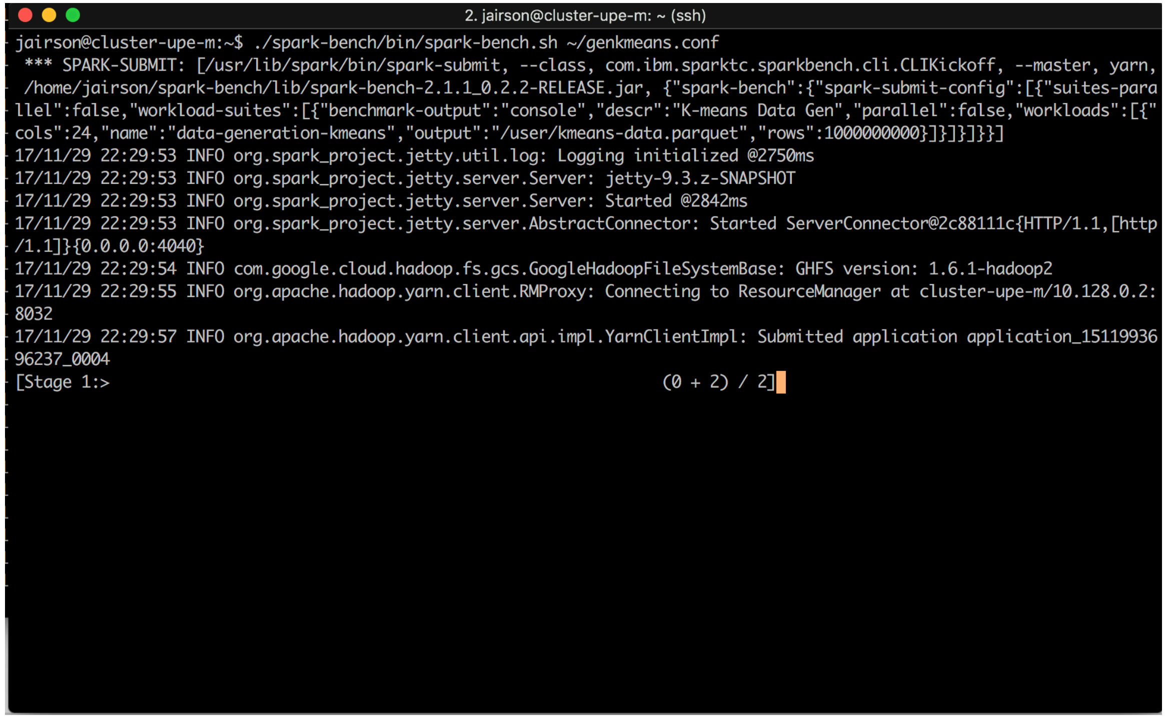
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	jairson	hadoop	0 B	Nov 29 22:26	2	128 MB	_SUCCESS
-rw-r--r--	jairson	hadoop	12.58 GB	Nov 29 22:26	2	128 MB	part-00000-c16447b0-7ca7-460b-a587-2837779e82ac-c000.snappy.parquet
-rw-r--r--	jairson	hadoop	12.58 GB	Nov 29 22:26	2	128 MB	part-00001-c16447b0-7ca7-460b-a587-2837779e82ac-c000.snappy.parquet

Showing 1 to 3 of 3 entries

Hadoop, 2017.

{ execução kmeans -> 8 gb }

- ./spark-bench/bin/spark-bench.sh ~/kmeans.conf



```
jairson@cluster-upe-m:~$ ./spark-bench/bin/spark-bench.sh ~/genkmeans.conf
 *** SPARK-SUBMIT: [/usr/lib/spark/bin/spark-submit, --class, com.ibm.sparktc.sparkbench.cli.CLIKickoff, --master, yarn,
 /home/jairson/spark-bench/lib/spark-bench-2.1.1_0.2.2-RELEASE.jar, {"spark-bench":{"spark-submit-config":[{"suites-para
llel":false,"workload-suites":[{"benchmark-output":"console","descr":"K-means Data Gen","parallel":false,"workloads":[{"
cols":24,"name":"data-generation-kmeans","output":"/user/kmeans-data.parquet","rows":1000000000}]}]}]}}]
17/11/29 22:29:53 INFO org.spark_project.jetty.util.log: Logging initialized @2750ms
17/11/29 22:29:53 INFO org.spark_project.jetty.server.Server: jetty-9.3.z-SNAPSHOT
17/11/29 22:29:53 INFO org.spark_project.jetty.server.Server: Started @2842ms
17/11/29 22:29:53 INFO org.spark_project.jetty.server.AbstractConnector: Started ServerConnector@2c88111c{HTTP/1.1,[http
/1.1]}{0.0.0.0:4040}
17/11/29 22:29:54 INFO com.google.cloud.hadoop.fs.gcs.GoogleHadoopFileSystemBase: GHFS version: 1.6.1-hadoop2
17/11/29 22:29:55 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at cluster-upe-m/10.128.0.2:
8032
17/11/29 22:29:57 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_15119936
96237_0004
[Stage 1:>                                                        (0 + 2) / 2]
```

- 40 milhões de amostras
- 24 dimensões R^{24}
- # clusters: 5

{ envio de jobs / ETL / LogReg / Naïve Bayes }

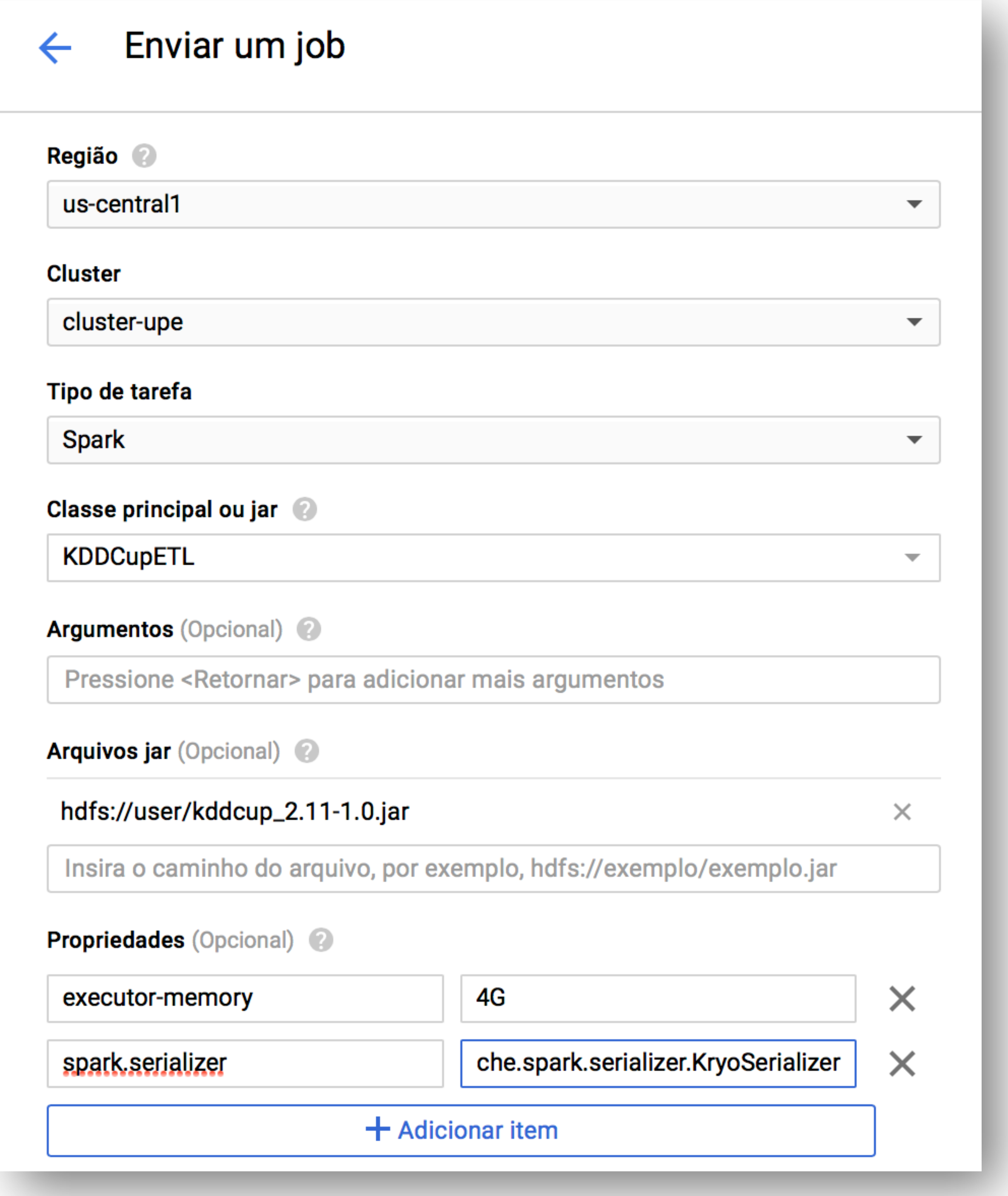
-
  ```
  spark-submit --class "KDDCupETL" --master yarn --conf spark.serializer=org.apache.spark.serializer.KryoSerializer kddcup_2.11-1.0.jar
  ```

-
  ```
  spark-submit --class "KDDCupRL" --master yarn --conf spark.serializer=org.apache.spark.serializer.KryoSerializer kddcup_2.11-1.0.jar
  ```

-
  ```
  spark-submit --class "KDDCupNaiveBayes" --master yarn --conf spark.serializer=org.apache.spark.serializer.KryoSerializer kdkddcup_2.11-1.0.jar
  ```

{ envio de jobs - interface web }

- KDDCupETL
- KDDCupRL
- KDDCupNaiveBayes



Enviar um job

Região: us-central1

Cluster: cluster-upe

Tipo de tarefa: Spark

Classe principal ou jar: KDDCupETL

Argumentos (Opcional): Pressione <Retornar> para adicionar mais argumentos

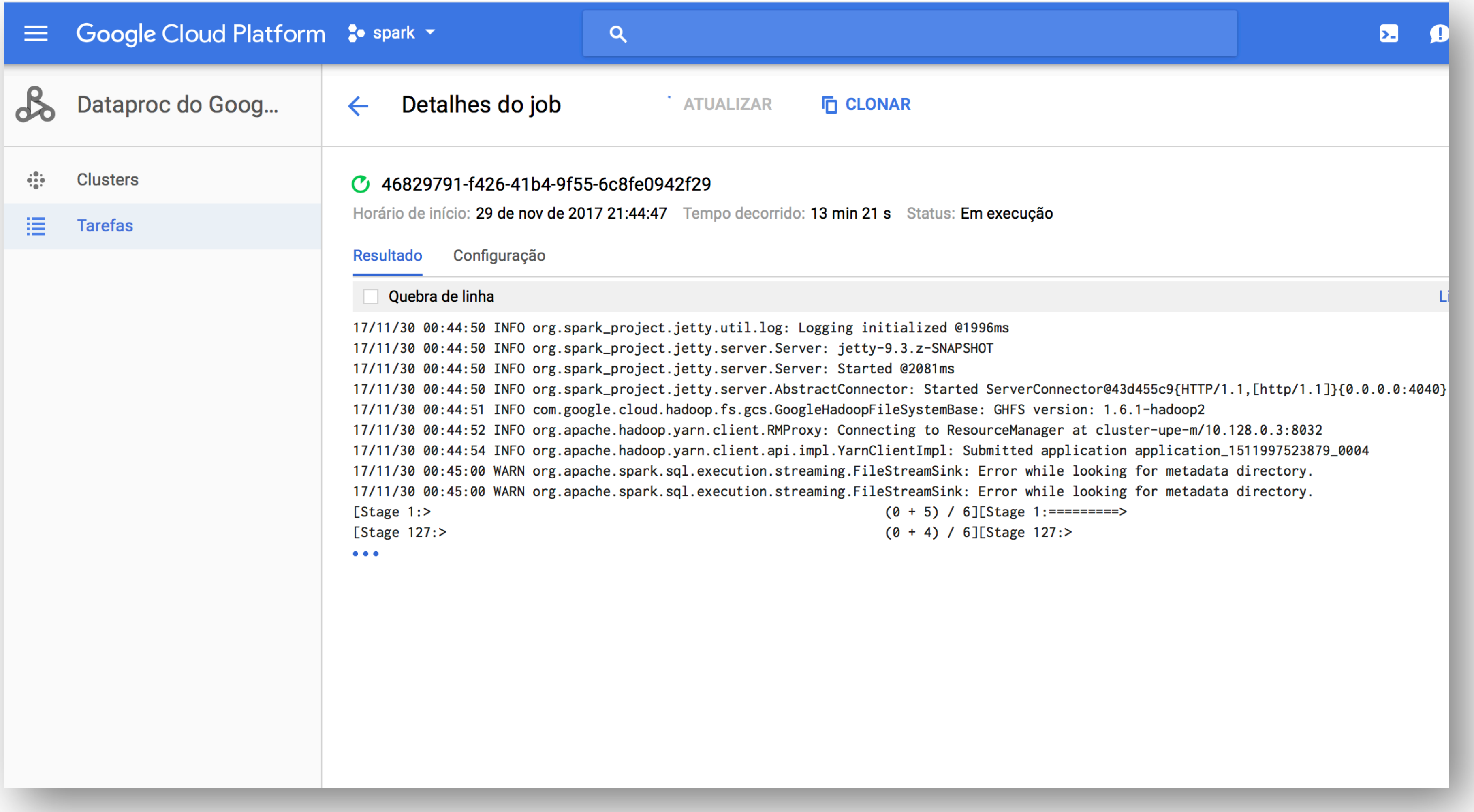
Arquivos jar (Opcional): hdfs://user/kddcup_2.11-1.0.jar

Insira o caminho do arquivo, por exemplo, hdfs://exemplo/exemplo.jar

Propriedades (Opcional)

executor-memory	4G
spark.serializer	che.spark.serializer.KryoSerializer

{ envio de jobs - interface web }



46829791-f426-41b4-9f55-6c8fe0942f29

Horário de início: 29 de nov de 2017 21:44:47 Tempo decorrido: 13 min 21 s Status: Em execução

□ Quebra de linha

```
17/11/30 00:44:50 INFO org.spark_project.jetty.util.log: Logging initialized @1996ms
17/11/30 00:44:50 INFO org.spark_project.jetty.server.Server: jetty-9.3.z-SNAPSHOT
17/11/30 00:44:50 INFO org.spark_project.jetty.server.Server: Started @2081ms
17/11/30 00:44:50 INFO org.spark_project.jetty.server.AbstractConnector: Started ServerConnector@43d455c9{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
17/11/30 00:44:51 INFO com.google.cloud.hadoop.fs.gcs.GoogleHadoopFileSystemBase: GHFS version: 1.6.1-hadoop2
17/11/30 00:44:52 INFO org.apache.hadoop.yarn.client.RMProxy: Connecting to ResourceManager at cluster-upe-m/10.128.0.3:8032
17/11/30 00:44:54 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1511997523879_0004
17/11/30 00:45:00 WARN org.apache.spark.sql.execution.streaming.FileStreamSink: Error while looking for metadata directory.
17/11/30 00:45:00 WARN org.apache.spark.sql.execution.streaming.FileStreamSink: Error while looking for metadata directory.
[Stage 1:>                                                          (0 + 5) / 6][Stage 1:=========>
[Stage 127:>                                                        (0 + 4) / 6][Stage 127:>
•••
```