BIG DATA

PROCESAMIENTO DE DATOS EMR

EDUARD LARA

1. APACHE HADOOP

- Apache Hadoop es un framework, es decir, es una librería de software
- Se utiliza para el procesamiento distribuido de grandes conjuntos de datos en un conjunto de ordenadores o cluster, utilizando para ello modelos de programación simple.
- Se puede escalar desde un ordenador hasta miles de ordenadores
- Cada ordenador ofrece computación local y almacenamiento y está diseñado para detectar y manejar errores en la capa de aplicación.

1. APACHE HADOOP

Módulos de la Arquitectura Hadoop

- Hadoop Common contiene las librerías y utilidades necesarias para el resto de los módulos de Hadoop. Es el núcleo de hadoop y permite la abstracción sobre el S.O. y el sistema de archivos. Contiene todos los ficheros JAR y los scripts necesarios para arrancar.
- Hadoop distributed File System HDFS es un sistema de ficheros distribuido. Es altamente tolerante a fallos, permite la replicación de datos entre sus nodos y tiene una alta capacidad de transmisión de datos.
- Hadoop Yarn permite gestionar los recursos del clúster y permite la planificación de trabajos,

1. APACHE HADOOP

Módulos de la Arquitectura Hadoop

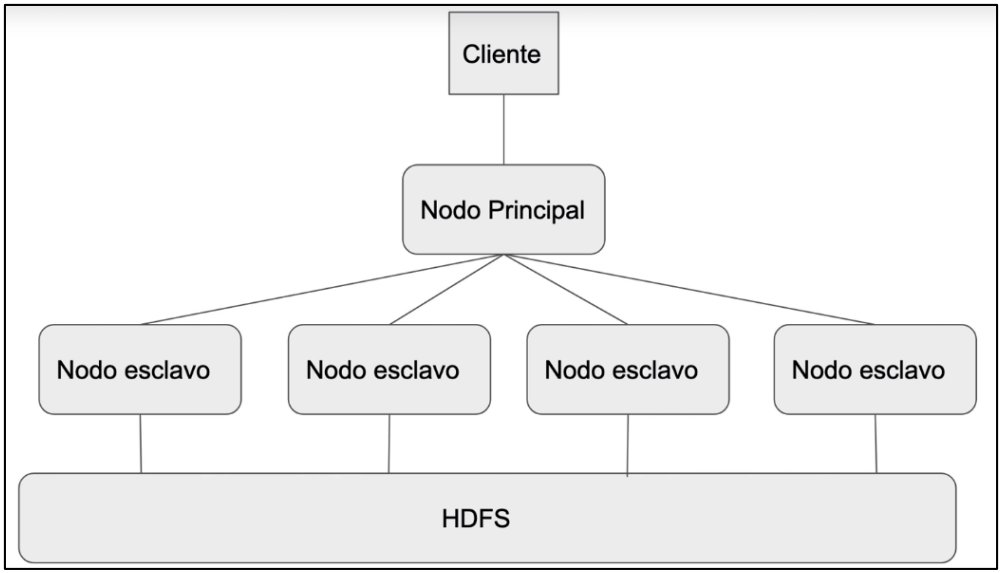
Hadoop Map Reduce

- Permite el procesamiento de grandes volúmenes de datos en paralelo mediante algoritmos distribuidos en un clúster de ordenadores.
- Los datos pueden estar en sistema de archivos HDFS (datos no estructurados), en una base de datos (datos estructurados) o pueden ser datos locales (Linux).
- MapReduce se divide en 2 tareas:
  - **Map** consiste en dividir los datos en pequeños conjuntos de datos y hacer un procesamiento en paralelo de ellos
  - **Reduce** sería recoger los datos de la tarea anterior Map y se juntan nuevamente para crear un fichero de salida.

1. APACHE HADOOP

Arquitectura Apache Hadoop

- En la parte superior de la arquitectura esta el cliente
- Un nodo principal gestiona los recursos de Hadoop
- Los nodos esclavos hacen el procesamiento de los datos
- En parte inferior tenemos el sistema de archivos hdfs donde se almacenan los datos de Apache.



2. EMR - ELASTIC MAP REDUCE

- Es una plataforma en clúster en grupo de servidores completamente administrada de AWS
- Sirve para procesar y analizar grandes cantidades de datos.
- Reduce el tiempo y coste ya que un cluster EMR se puede eliminar después de realizar el trabajo, con lo cual ahorramos costes.
- Utiliza frameworks de Big Data de código abierto:
  - Apache Hadoop
  - Apache Spark,
  - Presto
  - Apache HBase.

2. EMR - ELASTIC MAP REDUCE

Casos de uso que podemos utilizar en EMR

- El análisis y procesamiento de logs
- Realizar trabajos de ETL de extracción, transformación y carga de datos.
- El análisis de clics en sitios web.
- Machine learning.

3. ARQUITECTURA EMR

- EMR puede contener hasta 50 grupos de instancias (un grupo instancias es un conjunto de instancias de EC2)

- ¿Cómo se reparten estos 50 grupos de instancias?
  - 1 único grupo de instancias Master
  - 1 o mas grupos de instancias Core
  - Hasta 48 grupos de instancias Task

3. ARQUITECTURA EMR

Nodo Master o Principal

- Existe un único nodo Master que gestiona los recursos clúster.
- Coordina la distribución y la ejecución en paralelo de los trabajos Map Reduce.
- Gestiona el acceso al sistema de ficheros HDFS.
- Verifica el estado de los otros nodos como el nodo core y el nodo Task.

3. ARQUITECTURA EMR

Nodo Core o Central

- Es un nodo esclavo,
- Ejecuta las tareas que le pide el nodo Master.
- Almacena datos de HDFS o de EMRFS que es el sistema de ficheros de EMR
- Ejecuta también el programa DataNode, de gestión de datos, el programa NodeManager de Gestión de recursos y el programa ApplicationMaster para la ejecución y monitorización de los contenedores que contienen los programas y tareas a realizar.

3. ARQUITECTURA EMR

Nodo Task

- Es un nodo esclavo
- Es opcional, no almacena datos de HDFS
- Pueden ser añadidos/eliminados de un cluster EMR en ejecución
- Sirve para proporcionar un extra de capacidad de procesamiento.

3. ARQUITECTURA EMR

HDFS

- Es un sistema de ficheros distribuidos
- Permite el acceso simultáneo a datos desde diferentes clientes a un conjunto de servidores distribuidos por medio de una red.
- Permite compartir ficheros entre diferentes clientes y servidores.
- Utiliza la replicación de datos entre servidores para evitar posibles fallos en alguno de servidores de la red.
- Tiene un espacio de nombres global, es decir, una ruta común para todos los ficheros almacenados en HDFS
- Mantiene copia de los ficheros en varios servidores.

3. ARQUITECTURA EMR

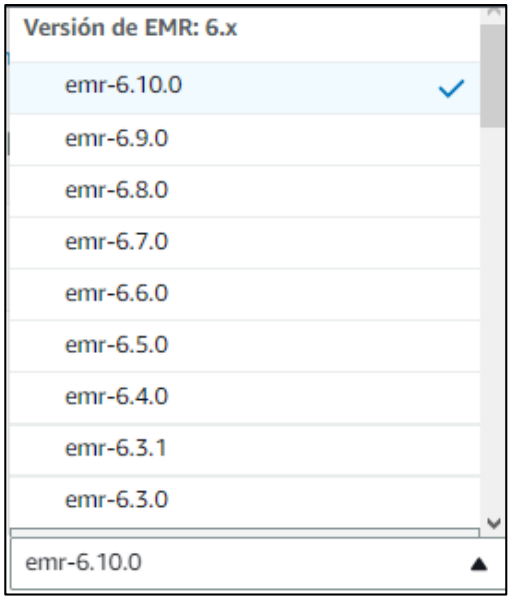
HDFS

- El factor de replicación es por defecto 3, es decir se hacen 3 copias de un mismo fichero en otros servidores de la red, pero este valor se puede aumentar o disminuir mediante la configuración de EMR.
- Esta información de metadatos de los ficheros, se almacena también en el nogo Master
- HDFS está diseñado para almacenar grandes ficheros de tamaño de Terabytes o de Petabytes.
- El tamaño del bloque de datos que se almacena en HDFS es por defecto de 64 MB, aunque también podemos hacerlos de 128 o de hasta 256 MB

4. OPERACIONES EMR

Releases EMR

- Es un conjunto de aplicaciones de software y componentes que son publicados en tu clúster EMR
- Cada release o cada versión de tu cluster EMR tendrá sus propias versiones de Hadoop, HBase, Hive, etc..



4. OPERACIONES EMR

Lanzamiento de un clúster EMR

- Se puede realizar mediante:
  - un script ya pre-construido,
  - automáticamente desde S3
  - el modo interactivo desde el nodo master y ejecutando tu script

4. OPERACIONES EMR

Tipos de clúster EMR

- Clúster de larga duración,
  - Los trabajos se ejecutan de forma frecuente
  - Los datos son tan grandes que resulta ineficiente cargarlos cada vez en un nuevo clúster.
  - La propiedad de autodeterminación está deshabilitada, porque se quiere mantener en ejecución el clúster, una vez el trabajo ha acabado..
  - Por último, la protección de borrado de clúster accidental también está habilitada para evitarlo.

4. OPERACIONES EMR

Tipos de clúster EMR

- Clúster temporal
  - Ejecuta el procesamiento y luego finaliza.
  - Sirve para realizar trabajos de procesamiento por lotes de tipo batch.
  - Permite reducir costes, ya que sólo se paga cuando el clúster está en ejecución. Cuando termina el trabajo el clúster finaliza y por tanto ya no se paga
  - Reducción de costes
  - Las entradas y salidas se pueden guardar en S3 para facilitar el lanzamiento del clúster y hay una recuperación sencilla en caso de fallos.

4. OPERACIONES EMR

Tipos de instancias para el clúster EMR

- MapReduce
  - Para trabajos orientados a batch proceso por lotes,
  - utilizaría distancia de tipo M3 o M4
  - escala de forma horizontal.
- Machine Learning. Utiliza instancias tipo P2 o C3 o C4
- Spark. Utiliza instancias de tipo R3 o R4
- Grandes instancias basadas en HDFS
  - Se utiliza en trabajos MapReduce que requiere un alto rendimiento de entrada/salida.
  - Utiliza estancias de tipo I2 o D2.

4. OPERACIONES EMR

Número de instancias para clúster EMR

- El numero de instancias y su tipo ayudan a ajustar el clúster EMR para manejar la cantidad de datos a procesar. EMR da la posibilidad de añadir más o menos a posteriori y volver a lanzar el clúster. Si se añaden más instancias, se incrementará el coste de tu clúster.
- Tamaño del nodo Master. Necesita pocos requerimientos de computación, ya que solo realiza tareas coordinación entre nodos.
  - Utilizar instancias m3.xlarge o m4.xlarge para clústers con 50 nodos o menos.
  - Utilizar instancias m3.2xlarge o m4.2xlarge para clusters con mas de 50 nodos,

4. OPERACIONES EMR

Número de instancias para clúster EMR

- Tamaño del nodo Core. Nodo que realiza tareas de procesamiento. Puede almacenar datos HDFS. No se puede disminuir el número de nodos core por debajo del factor de replicación (nº de veces que se duplican los datos dentro del clúster)
  - Clúster con 10 nodos → el factor de replicación será 3 y el nº de nodos core mínimo también será 3
  - Clúster entre 4 y 9 nodos → factor replicación=2
  - Clúster de 3 nodos o menos → factor replicación=1
- AWS recomienda clústers pequeños, con nodos grandes:
  - Con menos nudos se reduce la posibilidad de fallos y se reduce la cantidad de mantenimiento a realizar.

4. OPERACIONES EMR

Monitorización

- CloudWatch. Tipos de monitorización:
  - Por eventos: Responde a cambios de estado en los recursos
  - Por métricas. Las métricas de las ejecuciones de nuestro cluster se actualizan cada 5 minutos y no tienen cargo, ya que forman parte de EMR.
    - Permiten almacenar los datos durante dos semanas.
    - CloudWatch permite ver las métricas de las ejecuciones en tu clúster, muy útil para ver los consumos de tus trabajos EMR.

4. OPERACIONES EMR

Monitorización

- Interfaces web del propio Hadoop y Yarn
  - Aplicaciones utilizadas en EMR, como pueden ser Hadoop o HDFS
  - Estas interfaces web se configuran creando un túnel SSH con el nodo Master
- Ganglia
  - Herramienta de monitorización OpenSource
  - Permite sacar informes gráficos y monitorizar el rendimiento de tu cluster EMR

4. OPERACIONES EN EMR

Tamaño de los clústers EMR

El ajuste del tamaño de los clúster EMR se puede realizar de dos formas:

- Mediante un ajuste manual, es decir, pulsando el botón resize cuando estamos ejecutando tu cluster DMR, que permite cambiar el número de nodos de tu cluster.
  - Hay que tener en cuenta no bajar el número de nodos core por debajo del nº del factor de replicación.
- Mediante autoescalado, donde puedes indicar un número máximo y mínimo de instancias que se autogestionaran según condiciones de uso especificadas, por ejemplo uso de HDFS, etc

5. APLICACIONES EN EMR

HUE

- Es una interfaz web de código abierto para Apache Hadoop, y también para aplicaciones no Hadoop
- Se ejecuta sobre un navegador web y facilita la gestión del clúster EMR.
- Permite ver buckets S3, explorar ficheros de HDFS, consultar datos con HIVE y PIG

5. APLICACIONES EN EMR

HIVE

- Infraestructura DataWarehouse por encima de Hadoop
- Sirve para analizar grandes conjuntos de datos
- Permite consultar bases de datos mediante la interfaz HiveSQL, muy similar a SQL
- Los casos de uso en el que podemos utilizar Hive sería:
  - El procesamiento y análisis de logs,
  - Unir grandes tablas de datos
  - Trabajos por lotes
  - Consultas interactivas sobre HFDS o S3, etc..
- Permite la integración con S3 y DynamoDB. Copiar datos entre DynamoDB y HDFS, entre DynamoDB y S3

5. APLICACIONES EN EMR

HBASE

- Es un almacén de datos distribuido para Big Data en Hadoop
- Es una base de datos no relacional, se ejecuta sobre S3 o HDFS
- Permite acceder en tiempo real a tablas con billones de filas y millones de columnas.
- Casos de uso
  - Análisis de datos para compañías de publicidad online,
  - Empresas con grandes volúmenes de datos (para datos financieros)

5. APLICACIONES EMR

HBASE

- Cuando utilizar HBASE
  - Cuando tenemos grandes cantidades datos >100 GB
  - Cuando requieren buenos tiempos de respuesta, a pesar de tener grandes volúmenes de datos
  - Es No SQL
  - Cuando necesitamos un acceso rápido en tiempo real
  - cuando necesitamos tolerancia fallos en un entorno no relacional.
- Cuando no es conveniente utilizar HBASE
  - En aplicaciones transaccionales,
  - En base de datos relacionales con poco volumen datos

5. APLICACIONES EN EMR

PRESTO

- Es un motor de consultas SQL Open Source, capaz de ejecutar consultas de datos tanto relacionales como no relacionales y con tamaños desde GBbytes hasta PBytes
- Es más rápido que Hive
- Usar Presto, cuando se necesita alta concurrencia.
  - Capaz de ejecutar o lanzar miles de consultas por día.
  - Procesamiento en memoria, evitando problemas E/S
- Cuando no es conveniente utilizar Presto
  - Para OLTP el sistema transaccional.
  - Cuando hay que realizar union de tablas muy grandes
  - Cuando necesitamos hacer procesamiento por lotes.

6. SPARK EN EMR

- Framework de computación en clúster de código abierto.
- Procesamiento rápido de grandes cantidades de datos.
- Ejecución en memoria (es más rápido) o desde disco.
- Tiene una gran popularidad en entornos de Big Data.
- Tiene APIs para programación en Java, Scala, Python y R
- Contiene herramientas de alto nivel como puede ser
  - Spark SQL para el procesamiento de datos estructurados basados en SQL
  - MLib para Machine Learning
  - GraphX para procesamiento de grafos o gráficos
  - Spark streaming para lectura de datos en tiempo real.

6. SPARK EN EMR

Cuando usar Spark

- Para análisis interactivo, siendo más rápido que Hive.
- Para streaming de datos. Es capaz de leer datos simultáneos y realizar consultas en tiempo real.
- En Machine learning como máquinas de recomendación, detección de fraude, segmentación de clientes, etc
- Para la integración de datos, haciendo trabajos ETL

Cuando no utilizar Spark

- No se trata de una base de datos
- No está diseñada para OLTP
- En procesamiento por lotes donde es mejor usar Hive
- En entornos de reporting multiusuario con alta frecuencia. En este caso es mejor ETL en Spark y

6. COMPONENTES SPARK EN EMR

Spark Core

- Es el componente principal de Spark,
- Es el motor de ejecución general,
- Distribuye, planifica las tareas,
- Se ocupa de la gestión de la memoria
- Soporta APIs de lenguajes como Sscala Python, Java y SQL
- Soporta APIs para acceder a datos RDD, dataset y Dataframe.

GraphX

- Permite la computación paralela de gráficos.
- Construye y transforma gráficos de datos estructurados y soporta algoritmos gráficos.

6. COMPONENTES SPARK EN EMR

Spark SQL

- Permite ejecutar consultas SQL contra datos estructurados.
- Permite diferentes formatos: Avro, Parquet, ORC y JSON
- Soporta consultas en tablas Hive utilizando HiveSQL
- Tiene conectores JDBC/ODBC para acceder a cualquier tipo de base de datos existente.

Spark Streaming

- Permite la escalabilidad, alto rendimiento y tolerancia a fallos en el procesamiento de los datos.

6. COMPONENTES SPARK EN EMR

MLib

- Es una librería para machine learning.
- Permite ejecutar algoritmos de machine learning de forma distribuida.
- Permite además leer datos de HDFS, HBASE o cualquier fuente de datos de Hadoop
- Permite escribir aplicaciones utilizando el lenguaje de programación Scala Java, Python o SparkR

6. COMPONENTES SPARK EN EMR

Planificadores de tareas

- Hadoop Yarn, que permite la autentificación con Kerberos y también la reubicación dinámica de los atributos de tareas.
- Apache Mesos,
  - permite la integración entre Hadoop y Spark
  - permite la integración entre Kafka y Elastix search
  - Gestión de recursos de datacenter completos y entornos cloud.

Integración con otros servicios de AWS

- DynamoDB, RDS, ElasticSearch RedShift, S3, Kinesis, Kafka,

7. ALMACENAMIENTO DE FICHEROS EN EMR

- Hadoop divide los ficheros de datos HDFS, en trozos de forma automática
- Hadoop también divide los datos en S3 leyendo los ficheros en un rango de múltiples peticiones http.
- Utiliza algoritmos de compresión de datos: GZIP, BZIP2, LZO, SNAPPY (de más a menos comprensión)
- EMR permite los siguientes tipos de ficheros de datos:
  - TEXT (como csv)
  - Formato Parquet, formato de ficheros orientados a columna.
  - ORC
  - Sequence que es un fichero plano que contiene pares clave valor binarios
  - Avro, que es un framework de señalización de datos.

7. ALMACENAMIENTO DE FICHEROS EN EMR

Tamaño de los ficheros

- Los ficheros gzip no se pueden dividir y hay que mantenerse en el rango de 1 a 2 GBytes.
- Intentar evitar los ficheros pequeños de menos de 100 MB
- La herramienta S3DistCp que se puede utilizar para unir o juntar ficheros pequeños para crear un fichero más grande y se puede utilizar también para copiar ficheros de S3 a HDFS y viceversa de HDFS a S3.

8. AWS LAMBDA

- Es un servicio que funciona como FaaS es decir, function as a service - función como servicio, sólo se paga por el uso, por el número de milisegundos que se está ejecutando esta función Lambda
- Es un servicio sin estado, es decir, que no guarda información de una ejecución a otra.
- Es persistente.
- Es un servicio conducido por eventos, es decir, cuando se produce un cambio en una tabla de dynamoDB podemos lanzar una función lambda o cuando llega un fichero a S3 también podemos llamar a una función lambda que realice alguna acción.

8. AWS LAMBDA

- Se integra por tanto con otros muchos servicios de AWS y se invocan vía API Gateway
- Lambda recibe un evento y produce un nuevo evento o una nueva respuesta.
- Esta respuesta puede ser una invocación a otros servicios de AWS como httprequest, SNS, SQS o una llamada a otra función lambda.
- Puede tener como entradas un S3, un DynamoDB, un kinesis stream, un redshift.
- Luego estaría el proceso lambda, que ejecuta un tipo de tarea y como salida puede mandar información a IoT, Kinesis Firehose, Elastic Search, data pipeline, otras bases de datos, etc..

8. AWS LAMBDA

- Podemos crear un entorno sin servidores, con funciones lambda, para ejecutar un trabajo de tipo MapReduce
- Cuando una funcion Lambda es invocada, recibe:
  - la información del contexto, que es información sobre el objeto desde donde se ha invocado a la función Lambda
  - la información del evento que ha ocurrido.
- Límites técnicos del servicio de Lambda
  - El tamaño máximo que puede ocupar una función Lambda es de 512 MB
  - El número máximo de procesos o hilos en ejecución que puede tener una función AWS lambda es de 1024 procesos
  - La duración máxima es de 900 segundos. Es decir, es el tiempo límite que tiene la función Lambda para ejecutarse.

9. HCATALOG Y GLUE

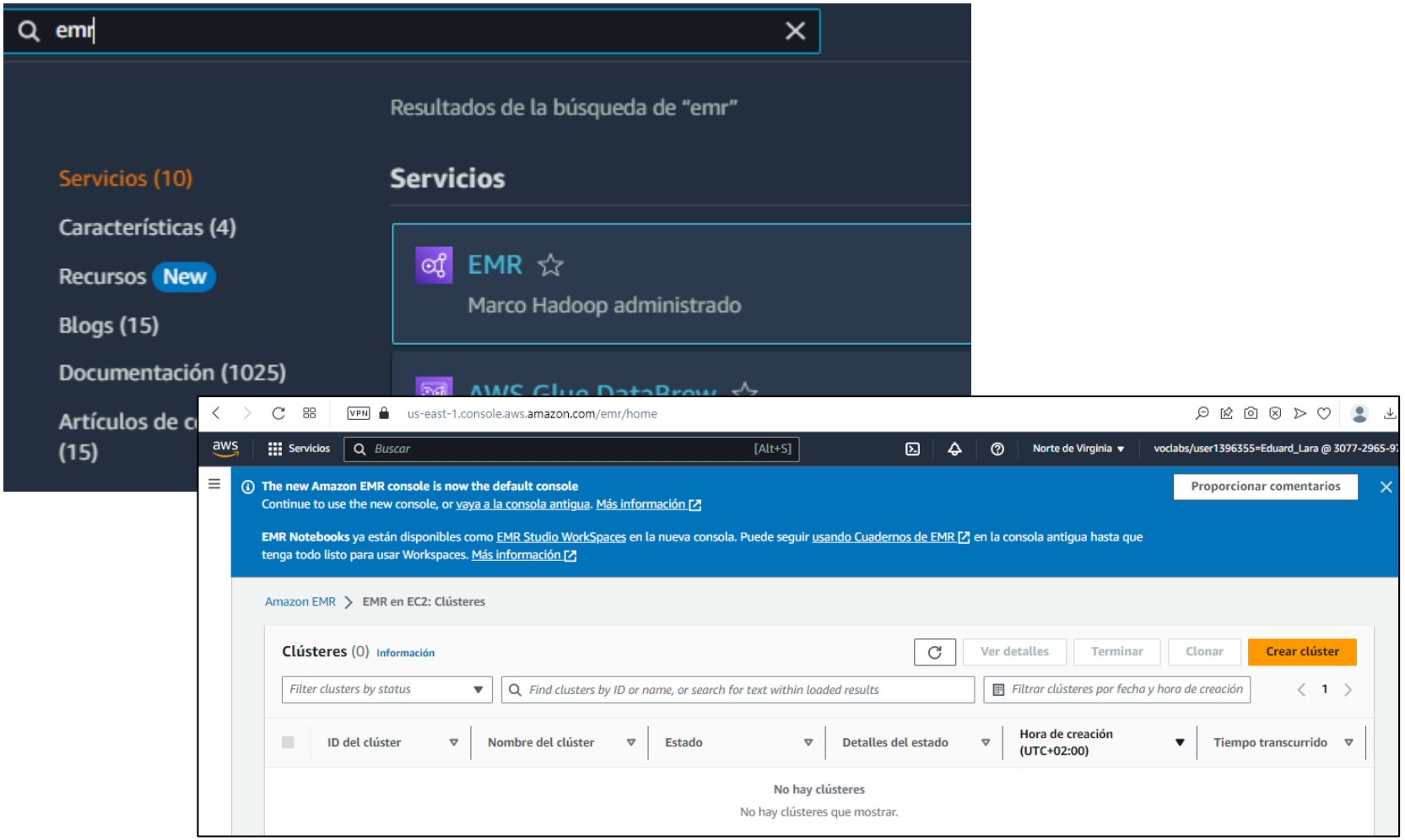
- HCatalog
  - Es una herramienta para acceder a las tablas de metadatos de Hive como otras herramientas como Pig, SparkSQL o aplicaciones MapReduce
- Servicios AWS Glue
  - Es un servicio ETL: extracción, transformación y carga de datos completamente administrado.
  - Es un servicio server, es decir, no necesita configurar servidores. Funcionan sin servidores.
  - Sirve para categorizar, limpiar y enriquecer los datos.
  - Se utiliza también para mover datos entre distintos almacenes de datos
  - También se utiliza como catálogo de datos, es decir, el descubrimiento de esquemas, etc..

9. HCATALOG Y GLUE

- Casos de uso AWS Glue
  - Consulta de datos en buckets de S3
  - Normalización y enriquecimiento de datos en un data warehouse
  - Catálogo de datos, es decir, podemos leer datos de RedShift, S3, RDS. Hacer un seguimiento de esquemas y cómo están construidas las tablas
  - También como procesos ETL conducidos por eventos. Un S3 llama a la función Lambda y a su vez llama a un Glue ETL que a su vez los resultados los pasa a un redshift o a un S3, es decir, procesos de extracción, transformación y carga de datos en diferentes sistemas.
  - Por último, estos scripts de ETL pueden ser construidos automáticamente por Glue en función de nuestras especificaciones o podemos personalizarlos con los lenguajes de programación Python o Scala.

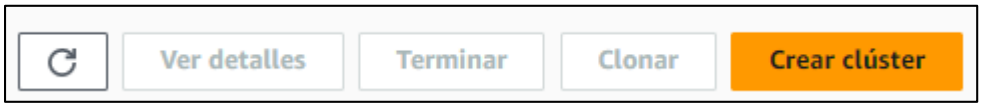
10. LABORATORIO. CREAR CLUSTER EMR DE FORMA RAPIDA

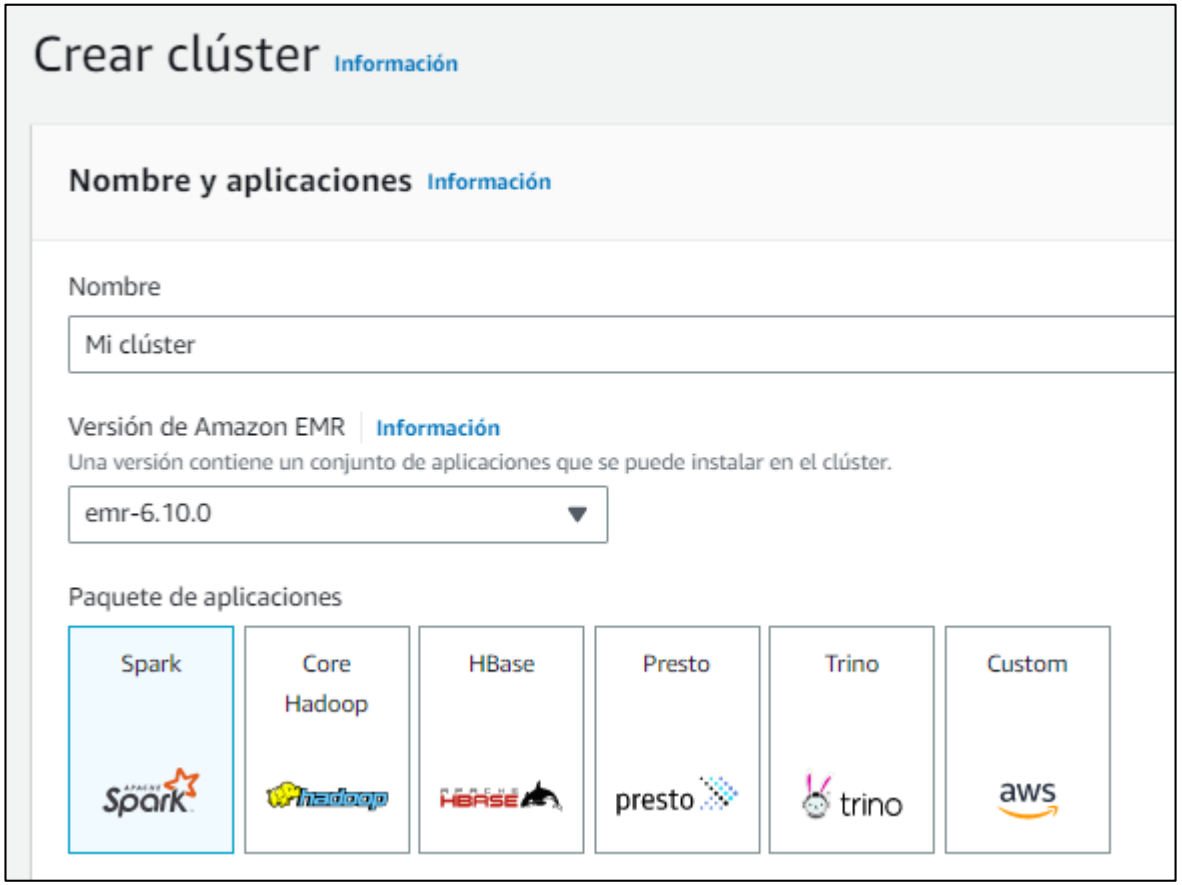
Paso 1. Vamos a crear un clúster EMR. Para ello buscamos el servicio EMR en el buscador y lo seleccionamos



10. LABORATORIO. CREAR CLUSTER EMR DE FORMA RAPIDA

Paso 2. Vamos a la parte de clusters y le damos a crear un clúster nuevo.





10. LABORATORIO. CREAR CLUSTER EMR

Paso 3. Primero para completar y configurar nuestro clúster EMR, ponemos su nombre, pej, mi_primer_cluster.

10. LABORATORIO. CREAR CLUSTER EMR

Paso 3. Pasos opcional Luego tenemos el modo de lanzamiento que sería el modo clúster o también podemos hacer en modo ejecución de pasos y si quisiéramos lanzar este clúster EMR para hacer por ejemplo ejecutar un programa en Spark o hacer una consulta en hive, pues entonces utilizaremos la ejecución por pasos. Podríamos la opción de aplicación Spark y aquí podríamos configurar nuestra aplicación Spark que vamos a ejecutar

Lo mismo si quisiéramos en lugar de eso ejecutar una consulta hive pues los mismos.Tenemos de configurar aquí las opciones

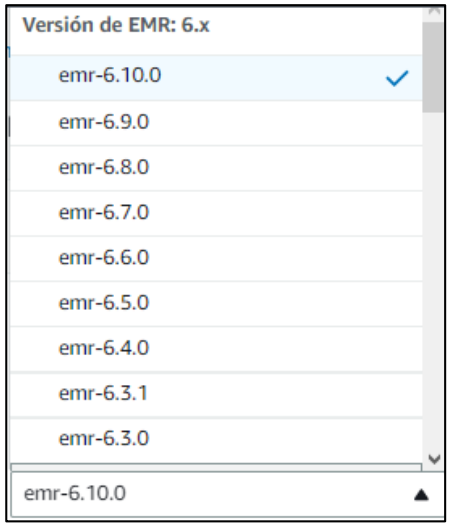
por defecto. Ahora lo vamos a dejar en modo clúster.

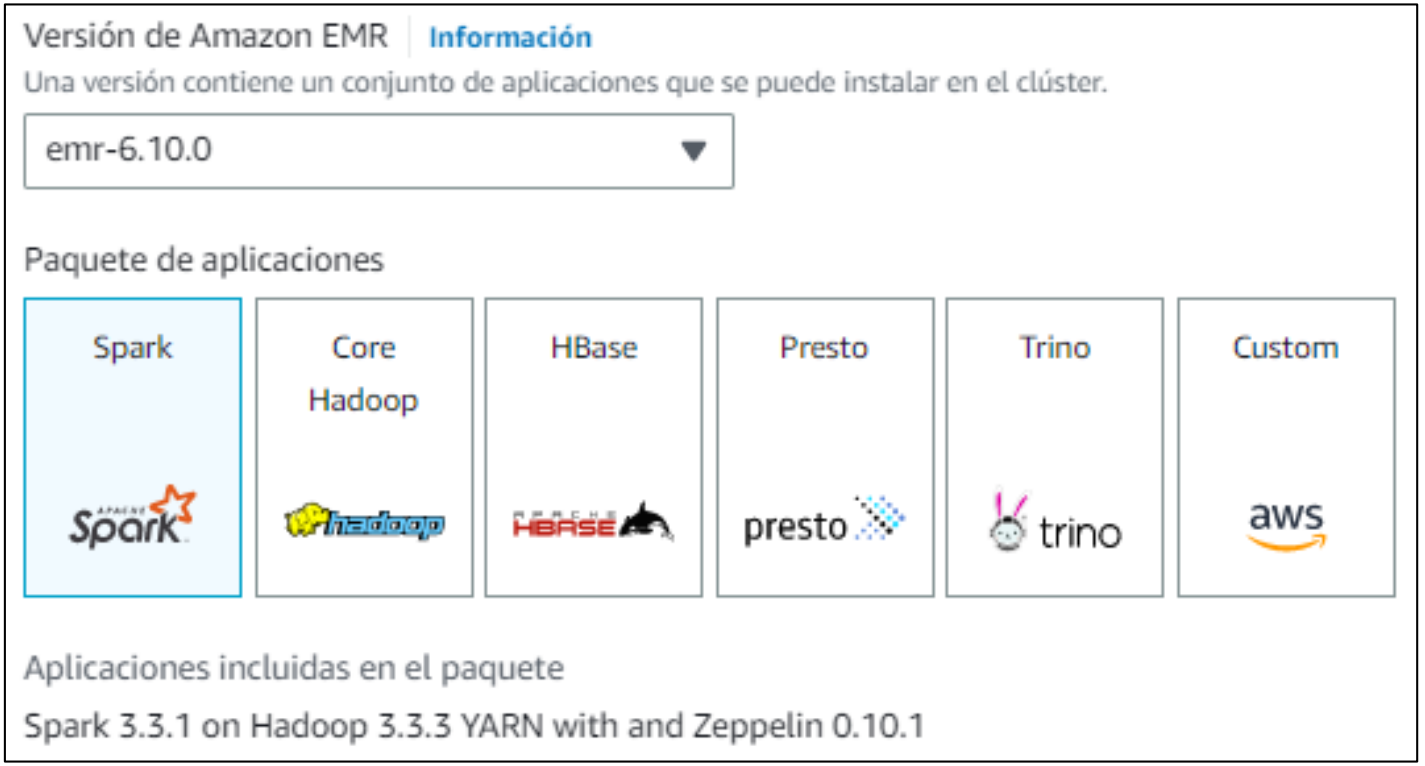
Simplemente lo que hace es arrancar cluster con la configuración que vamos a hacer ahora.

10. LABORATORIO. CREAR CLUSTER EMR DE FORMA RAPIDA

Paso 4. Según la versión o release de Amazon EMR que seleccionemos para nuestro clúster, habrá preconfigurados unos paquetes u otros de aplicaciones que se podrán instalar con este versión en el clúster.

La versión por defecto EMR 6.10.0, contempla la instalación de los siguientes paquetes de aplicaciones:





10. LABORATORIO. CREAR CLUSTER EMR DE FORMA RAPIDA

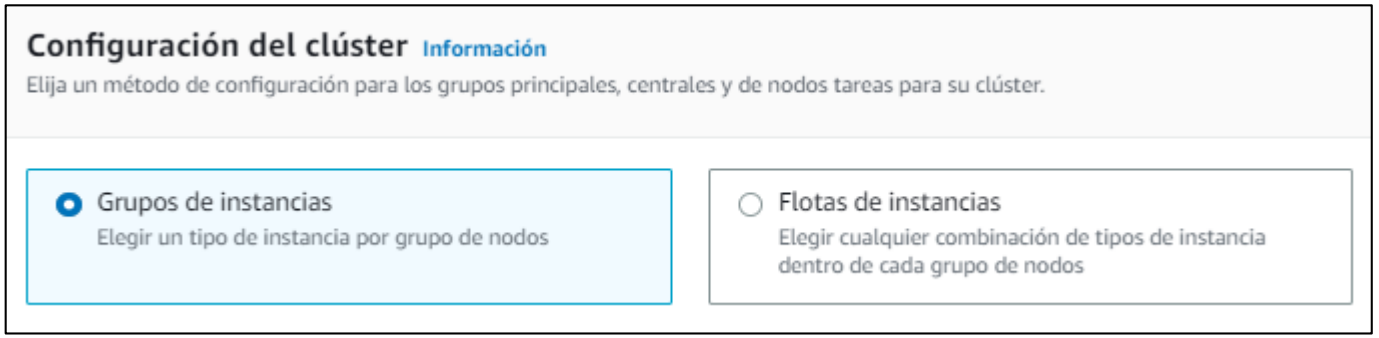
Paso 5. Dentro de la release 6.10 EMR, vemos que cada paquete incluye una serie de aplicaciones:

- Paquete Spark → Spark 3.3.1 on Hadoop 3.3.3 YARN with and Zeppelin 0.10.1
- Paquete Core Hadoop incluye: Hadoop 3.3.3 with Hive 3.1.3, Hue 4.10.0, Pig 0.17.0 and Tez 0.10.2.

Según las aplicaciones que necesitemos que tengan nuestro clúster, elegiremos un paquete u otro dentro de esta versión. Elegimos la opción Hadoop Core

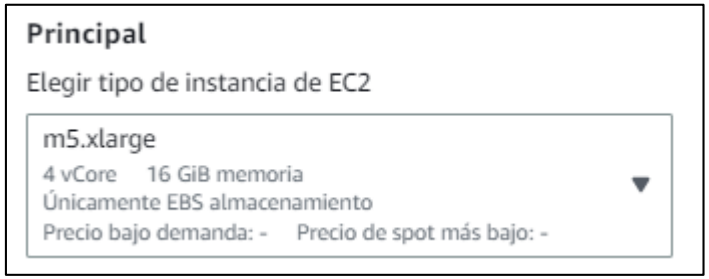
10. LABORATORIO. CREAR CLUSTER EMR DE FORMA RAPIDA

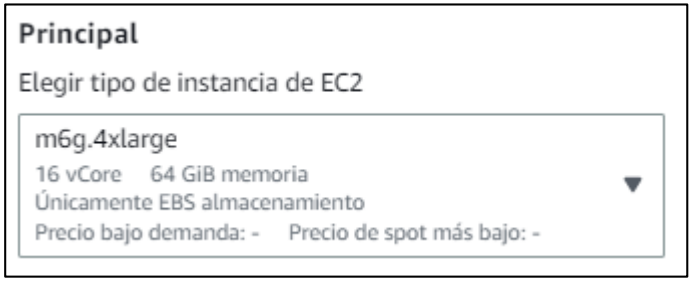
Paso 6. En la Configuración del clúster, elegiremos la opción Grupo de instancias, que permite elegir un tipo de instancia por grupo de nodos



10. LABORATORIO. CREAR CLUSTER EMR DE FORMA RAPIDA

Paso 7. A continuación elegimos el tipo de instancia EC2 (tipo de procesador y memoria) que van a tener nuestros nodos Principal y Central. Según el tipo de instancia elegida tendrá unas determinadas características.





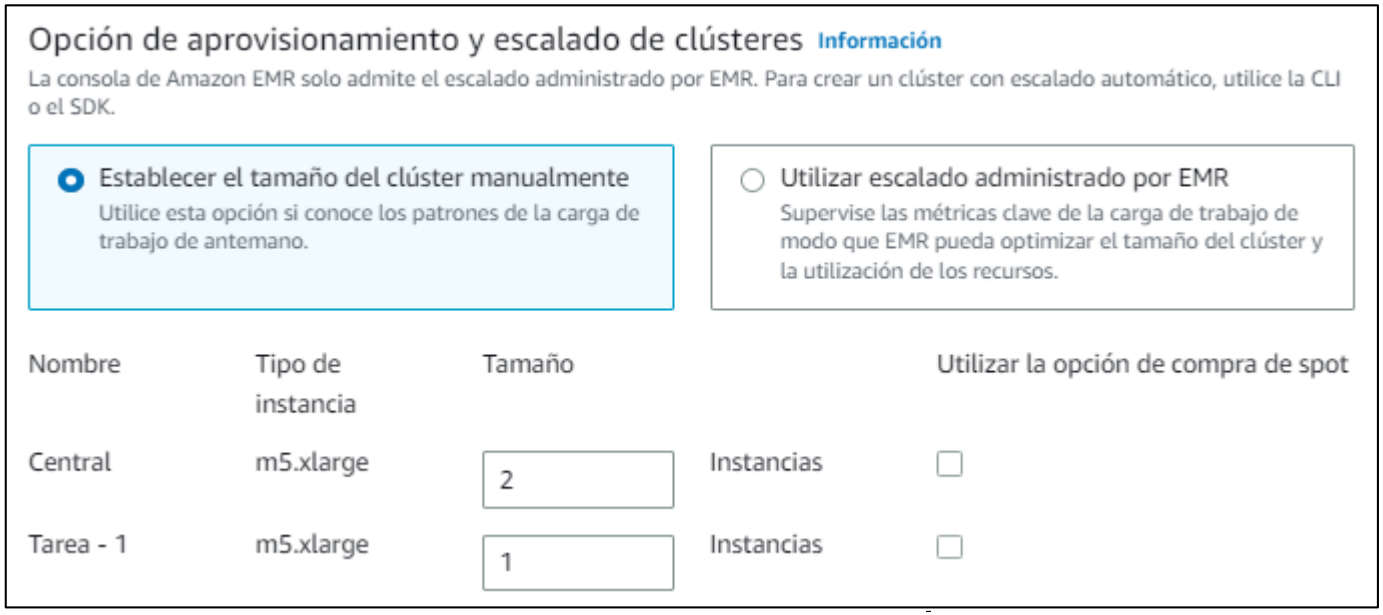
Si elegimos m6g.4xlarge viene con 16 cores y 64 GB. Si elegimos m4.large viene con 2 cores y 8 GB.

El tipo de instancia por defecto m5.xlarge, que tiene 4 Cores y 16GB de RAM, es el más adecuado para temas de Big Data y procesamiento de clúster EMR

10. LABORATORIO. CREAR CLUSTER EMR DE FORMA RAPIDA

Paso 8. La elección del tamaño del clúster se puede realizar de 2 formas:

- Establecer el tamaño del clúster manualmente, si conocemos los patrones de la carga de trabajo a priori
- Utilizar el escalado gestionado por EMR



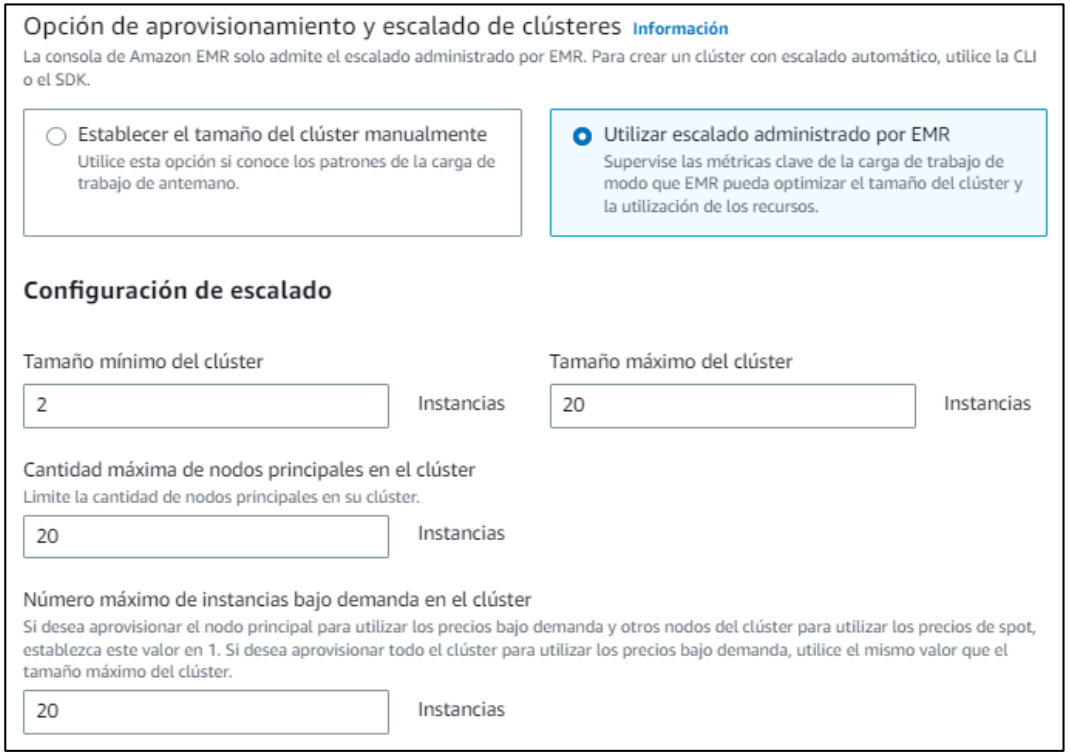
Un clúster formado por 1 nodo master, 2 nodos centrales y un nodo task opcional

Elegimos esta opción

10. LABORATORIO. CREAR CLUSTER EMR DE FORMA RAPIDA

Paso 9. Con la opción Escalado administrado por EMR nuestro clúster EMR puede optimizar el tamaño del clúster y la utilización de los recursos, en función de las métricas clave de la carga de trabajo

Puede variar el nº de instancias automáticamente, entre un valor mínimo 2 y un valor máximo de instancias de 20



Según el nº de instancias que pongamos será un precio u otro

10. LABORATORIO. CREAR CLUSTER EMR DE FORMA RAPIDA

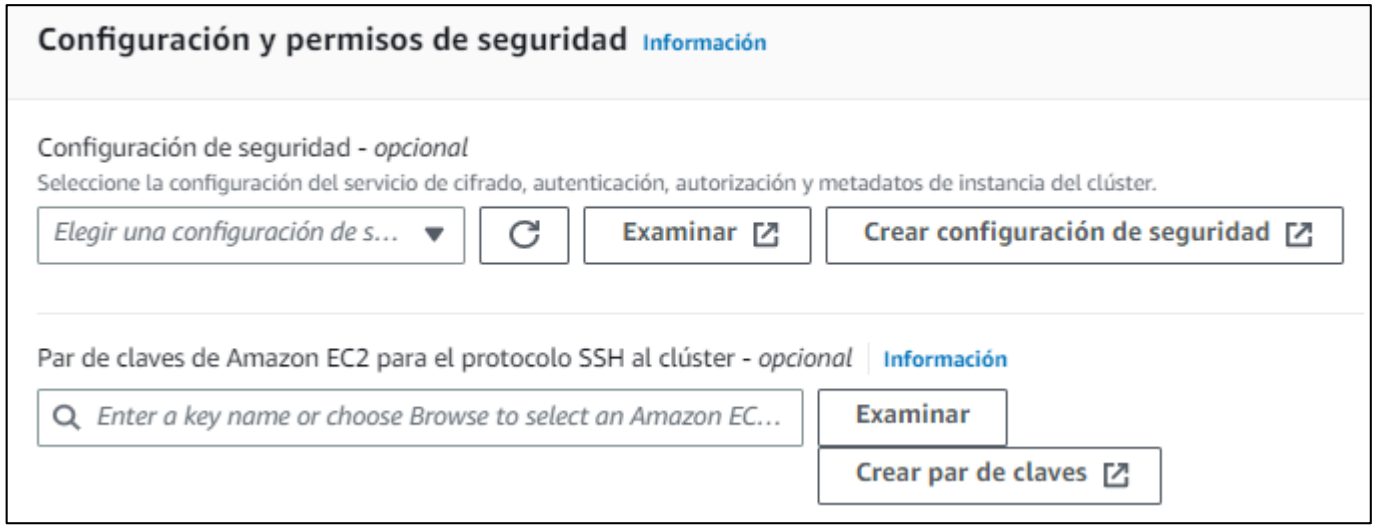
Paso 10. En la opción de terminación del clúster, si se habilita la terminación automática, y el clúster permanece inactivo un cierto tiempo (una hora, dos horas, etc), se termina de automáticamente. Si lo creamos, lo utilizamos, pero no lo terminamos nos siguen cobrando. Si el clúster EMR esta activo, sigue facturando.

Ponemos que después de una hora, se termine de forma automática

10. LABORATORIO. CREAR CLUSTER EMR DE FORMA RAPIDA

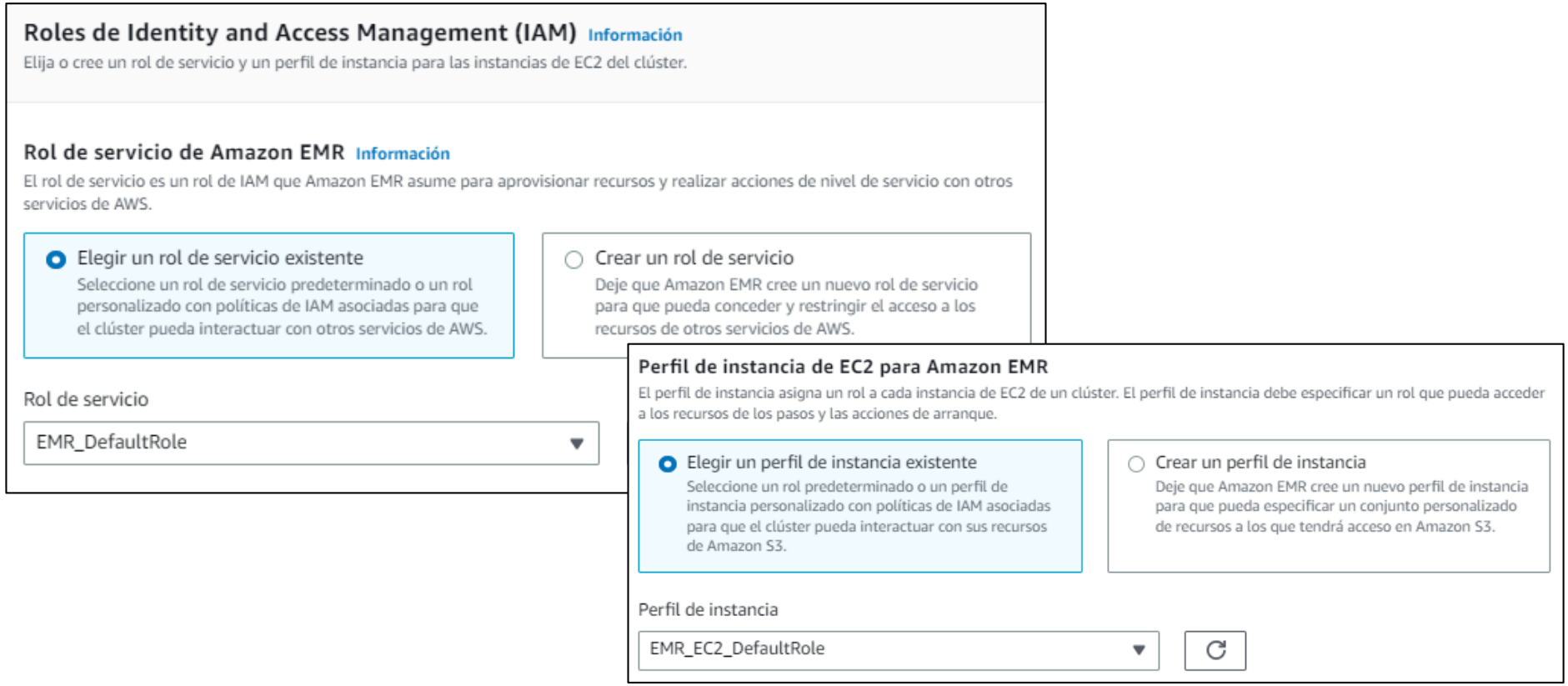
Paso 11. En la opción de Configuración y permisos de seguridad podemos elegir seleccionar un par de claves de acceso que ya tengamos si queremos conectarnos al nodo maestro para ejecutar algún tipo de script mediante el terminal SSH (como si nos conectáramos a un terminal de una instancia de EC2)

Le damos a continuar sin un par de claves, ya que solo queremos ejecutar este cluster EMR para hacer una consulta o para hacer un programa de Spark. No queremos conectarnos a los nodos



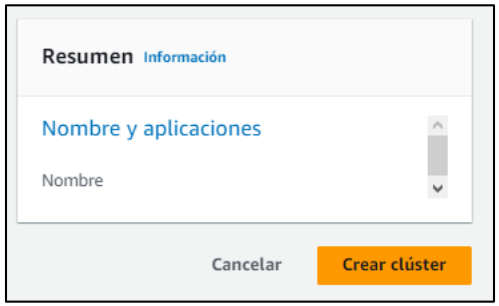
10. LABORATORIO. CREAR CLUSTER EMR DE FORMA RAPIDA

Paso 12. En cuanto a los permisos y roles de servicio, tenemos que indicar un rol de servicio y un perfil de instancia de EC2 para Amazon EMR. Podemos crearlos de nuevo, pero también podemos utilizar los perfiles predeterminados como EMR_DefaultRole y EMR_EC2_DefaultRole

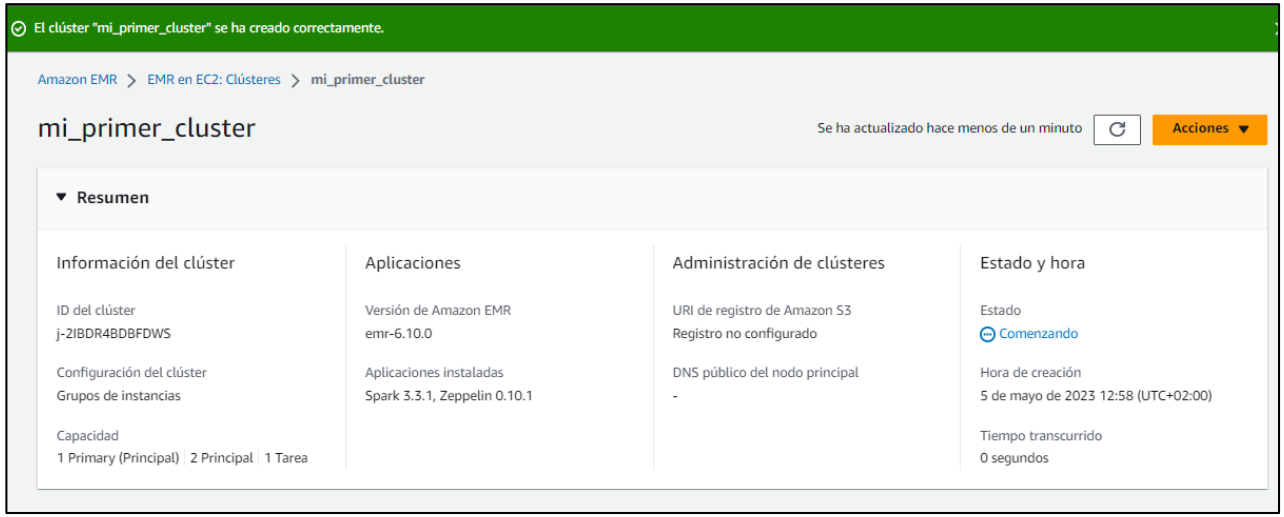


10. LABORATORIO. CREAR CLUSTER EMR

Paso 13. Una vez hemos configurado todas estas opciones rápidas para crear un clúster EMR y simplemente nos quedaría darle a crear el clúster.

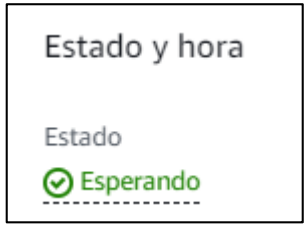


Comienza a crear todos los nodos o instancias EC2 Entonces esto tardará un poco

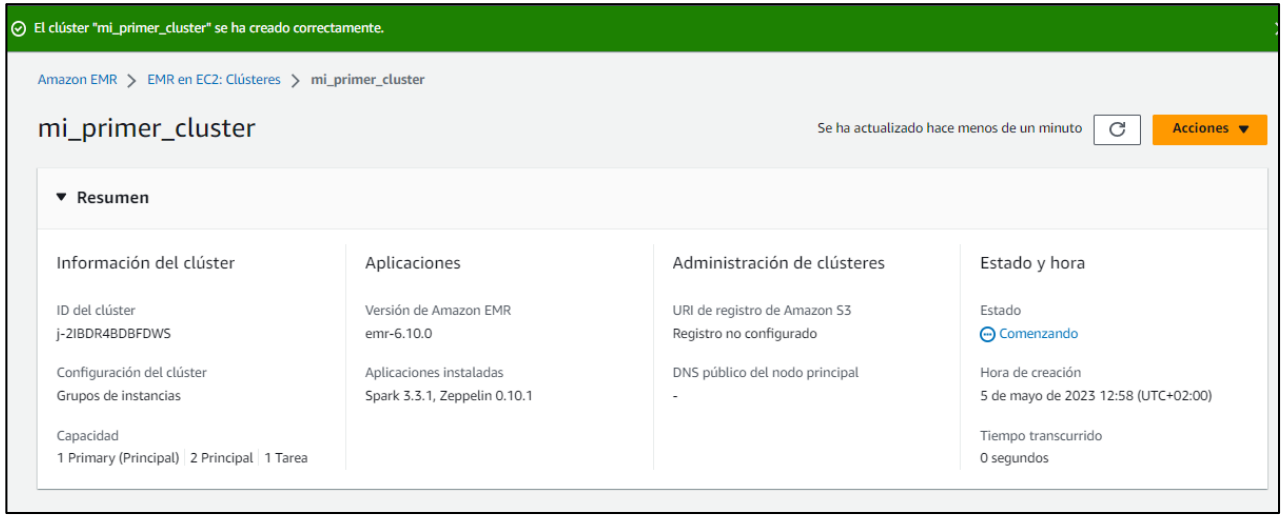


10. LABORATORIO. CREAR CLUSTER EMR

Paso 14. Una vez finalizado la creación del cluster, vemos que esta en estado Esperando, es un estado de reposo, mientras no tiene que hacer ningún tipo de trabajo.

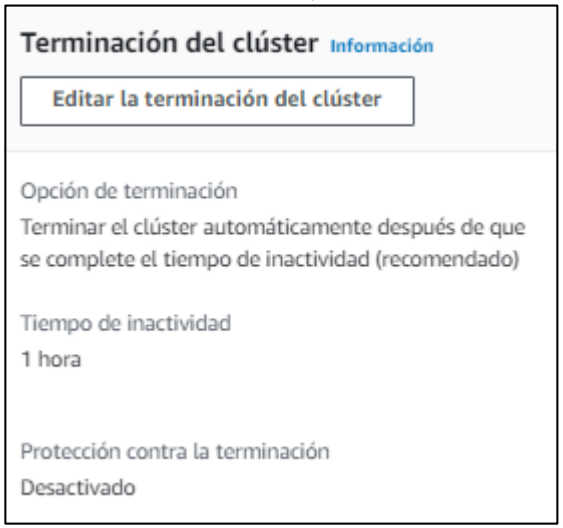


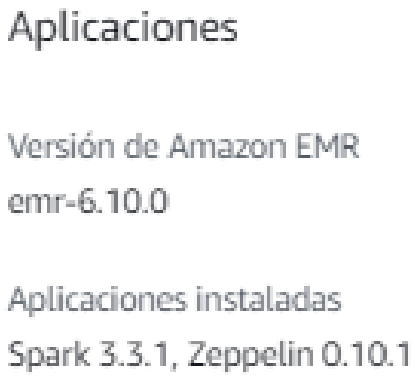
Paso 15. En resumen, pues tenéis aquí la información que hemos configurado.



10. LABORATORIO. CREAR CLUSTER EMR

Paso 16. En propiedades vemos la terminación del clúster. Se termina automáticamente después de una hora sin actividad

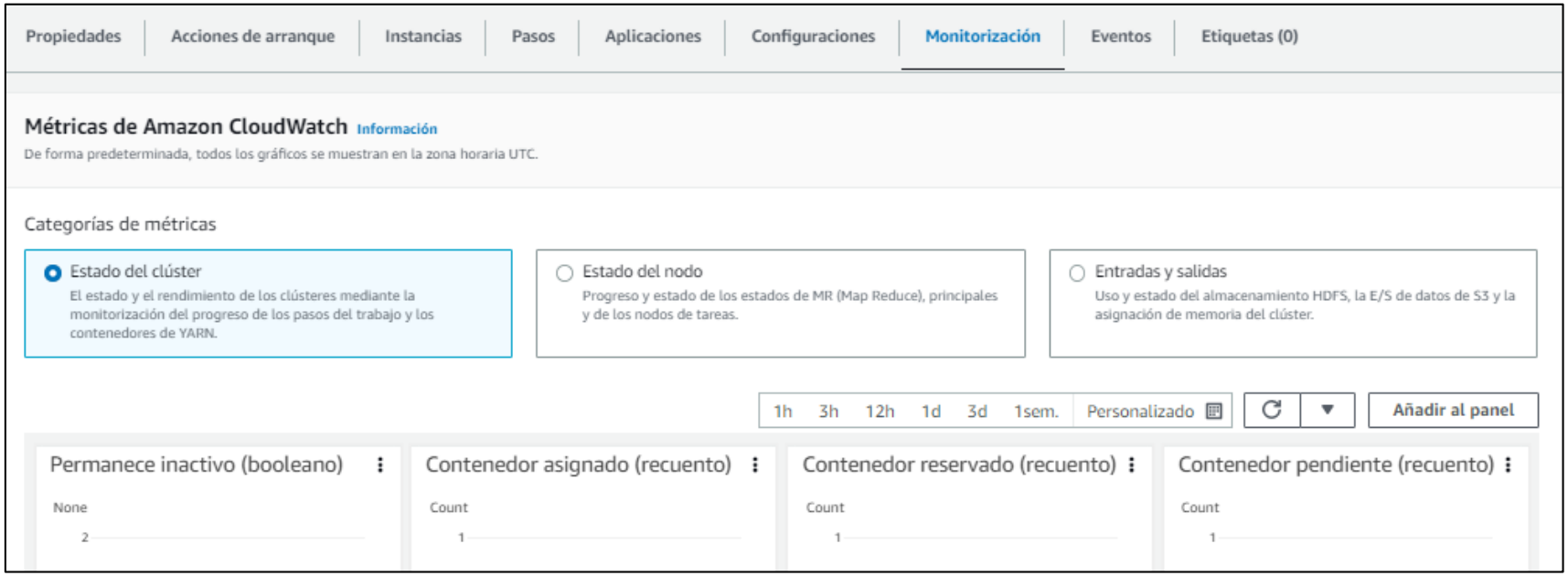




Paso 17. Vemos la versión usada de EMR, y las aplicaciones instaladas, Spark, Zepelin. Encontramos la información que hemos ido configurando por defecto de forma rápida, y otras cosas que hemos ido poniendo nosotros

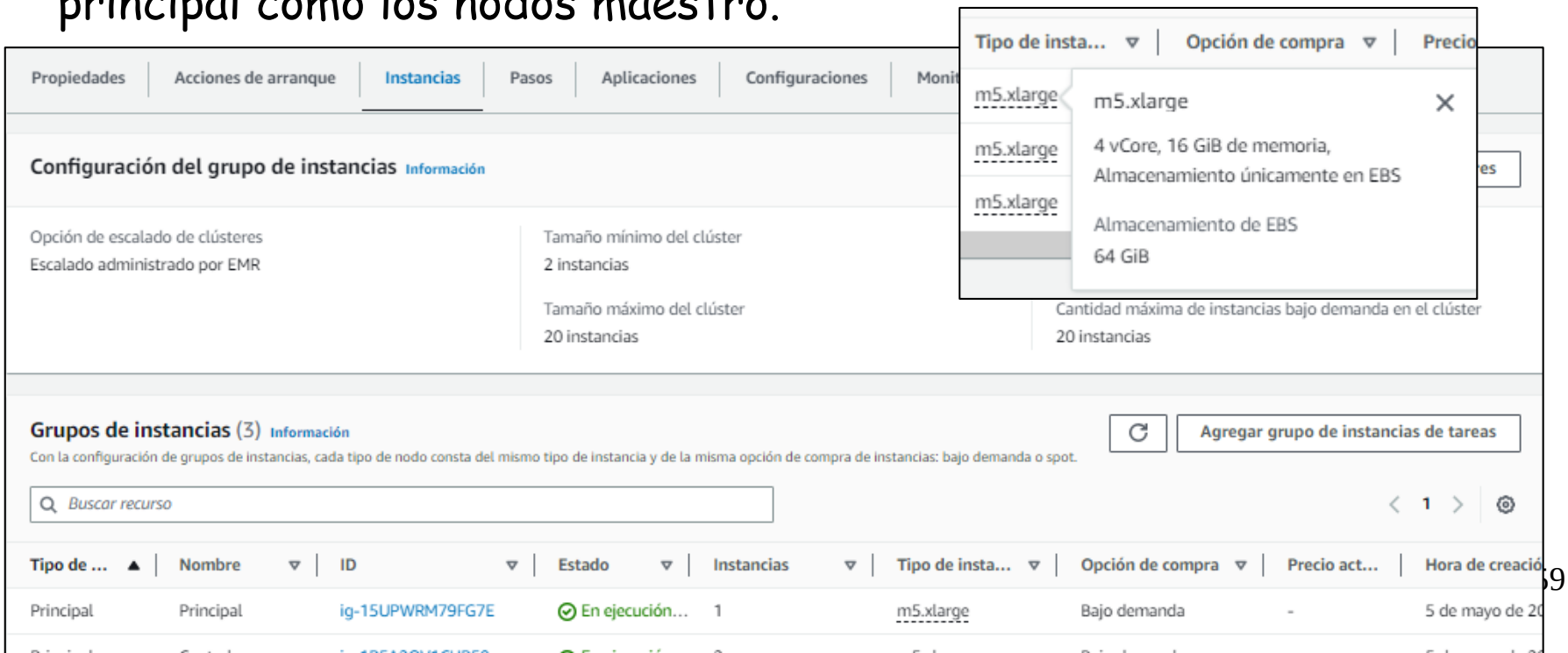
10. LABORATORIO. CREAR CLUSTER EMR

Paso 18. En la pestaña Monitorizacion podemos ver la monitorización tanto de estado de clúster, como del estado del nodo como entrada salida.



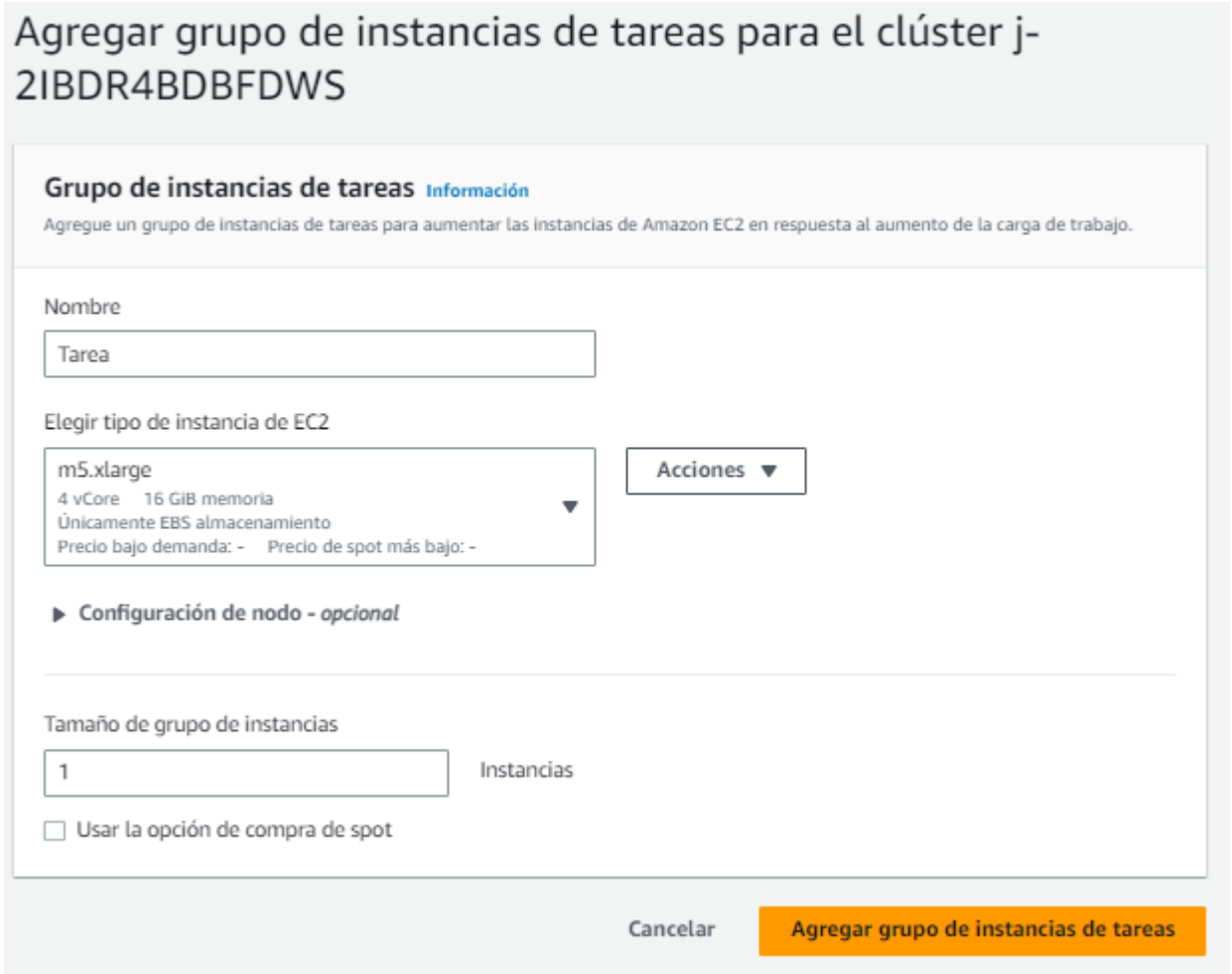
10. LABORATORIO. CREAR CLUSTER EMR

Paso 19. En la pestaña Instancias, vemos que hemos puesto un nodo Principal que está en ejecución en verde, que tiene un procesador m5.xlarge y con 16 GB de memoria. Hemos puesto 2 nodos core, también con un procesador m5.xlarge con 16 gigas. Vale, tienen el mismo procesador los dos.e acabé de configurar la misma en el mismo procesador, el mismo tipo de instancias para todos los nodos, tanto el principal como los nodos maestro.



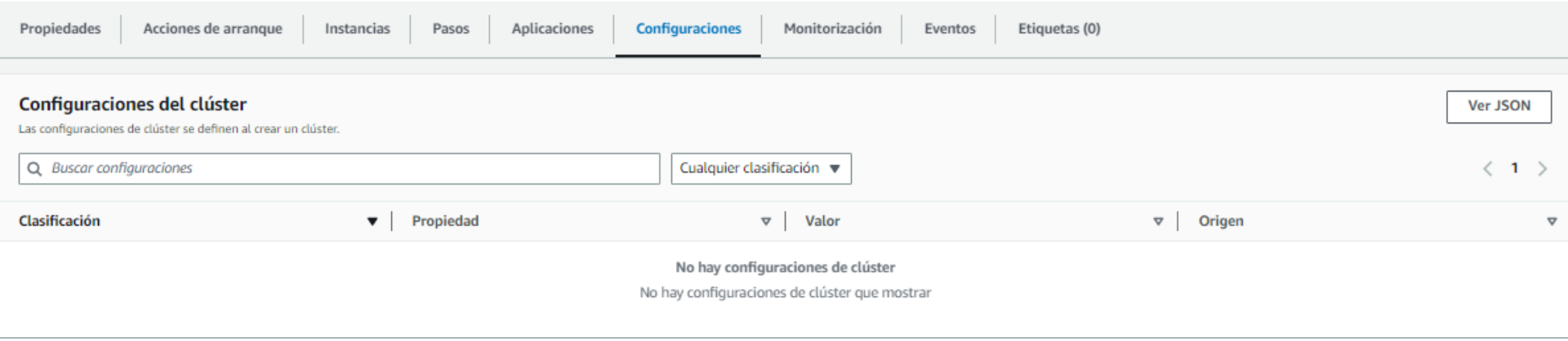
10. LABORATORIO. CREAR CLUSTER EMR

Paso 20. Una vez que está arrancado,el cluster podemos cambiar el tamaño y configurar otro número de instancias principales diferentes.



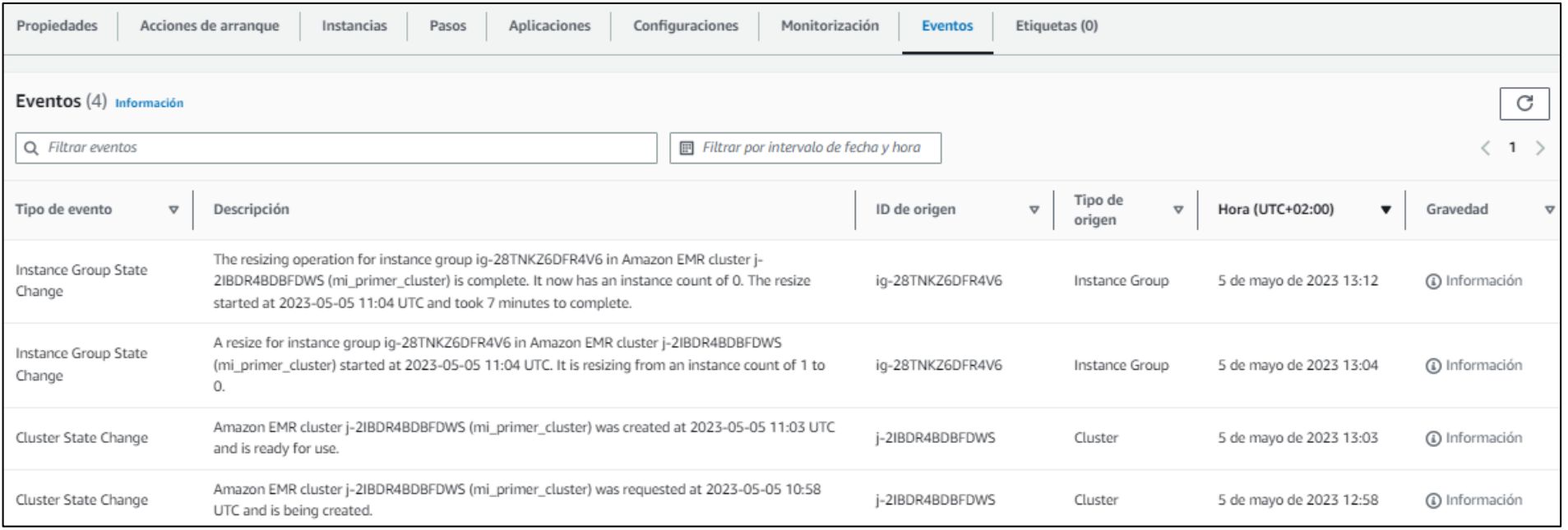
10. LABORATORIO. CREAR CLUSTER EMR

Paso 21. En cuanto a las configuraciones, aquí no tenemos nada más



10. LABORATORIO. CREAR CLUSTER EMR

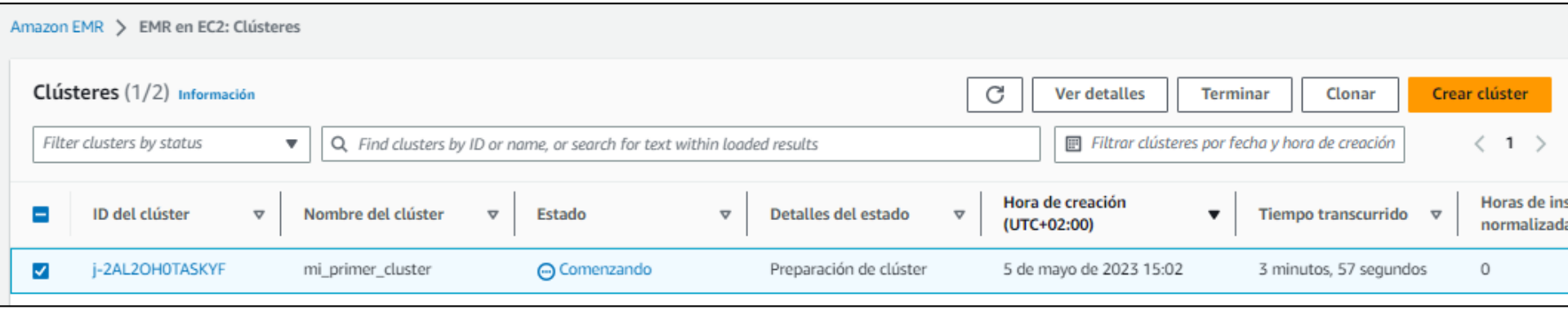
Paso 22. En la pestaña Eventos, tenemos los eventos que ha ido haciendo para arrancar este clase de mente, los pasos que está completado y ninguno pendiente, con lo cual ahora mismo nuestro claster está esperando a ejecutar algún tipo de tarea. Lo hemos creado de forma rápida, hemos instalado todas las aplicaciones que podemos utilizar y eso sería todo para la creación rápida del clúster.



10. LABORATORIO. CREAR CLUSTER EMR

Paso 23. Una vez creado el clúster y vistas sus configuraciones, veremos como se finaliza el clúster, ya que no queremos utilizarlo más,

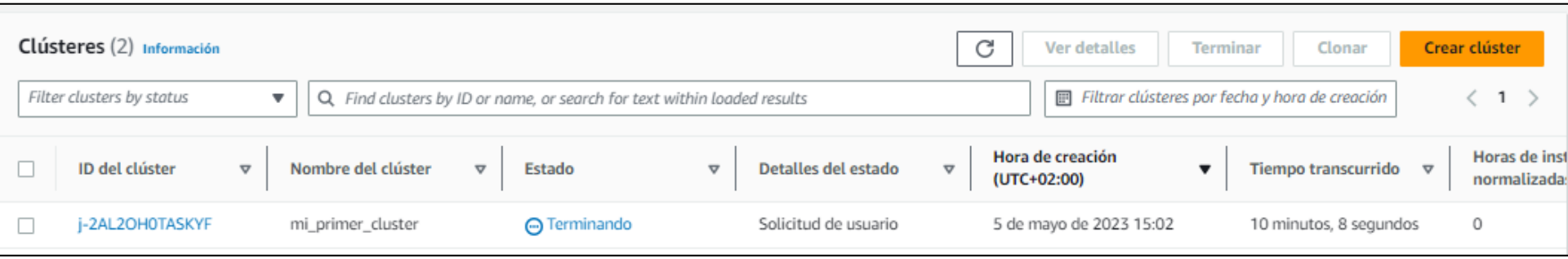
Vamos a Clústeres EMR y seleccionamos nuestro clúster



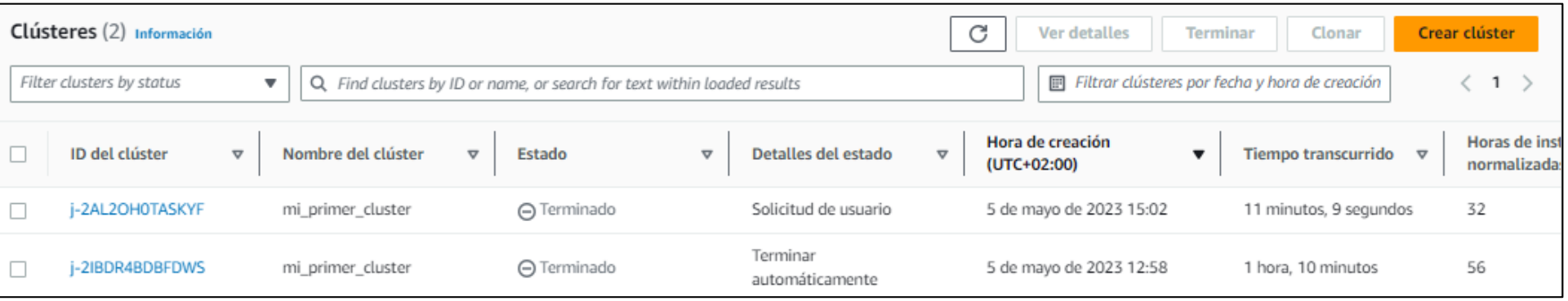
Paso 24. Le damos al botón Terminar. Nos pregunta si estamos seguro y le damos al botón Terminar.

10. LABORATORIO. CREAR CLUSTER EMR

Paso 25. Automáticamente comienza al estado de Terminando dónde empieza a cerrar las instancias de EC2 creadas, etc.



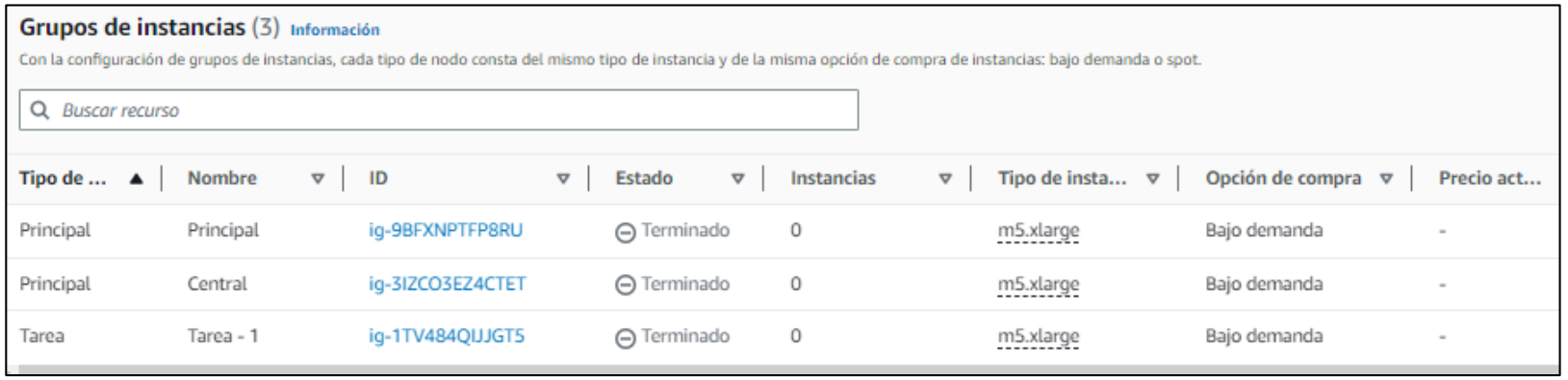
Tarda un cierto tiempo. Al final vemos que mi_primer_cluster aparece en estado Terminado, ha realizado las tareas necesarias para finalizar.



10. LABORATORIO. CREAR CLUSTER EMR

Paso 26. Una vez ha pasado a estado Terminado, estará ahí bastante tiempo, puede estar 20 o 30 días. No tiene ningún coste, no esta consumiendo recursos, simplemente aparece como terminado y esta guardada la información durante un tiempo, por si quisiéramos volver a clonarlo y ejecutarlo.

Paso 27. Si volvemos a entrar en mi_primer_cluster, vemos que va terminando todas las solicitudes. Al final las instancias del maestro y principal también se quedan estado Terminado, finalizando todas las tareas.



Grupos de instancias (3) Información

Con la configuración de grupos de instancias, cada tipo de nodo consta del mismo tipo de instancia y de la misma opción de compra de instancias: bajo demanda o spot.

Buscar recurso

Tipo de ...	Nombre	ID	Estado	Instancias	Tipo de insta...	Opción de compra	Precio act...
Principal	Principal	ig-9BFXNPTFP8RU	Terminado	0	m5.xlarge	Bajo demanda	-
Principal	Central	ig-3IZCO3EZ4CTET	Terminado	0	m5.xlarge	Bajo demanda	-
Tarea	Tarea - 1	ig-1TV484QIJJGT5	Terminado	0	m5.xlarge	Bajo demanda	-

10. LABORATORIO. CREAR CLUSTER EMR

Paso 28. Lo importante es que hemos finalizado nuestro clúster. Si entramos en el cluster, la función de Finalizar ya está deshabilitada, indicando que ya está parado y está terminado