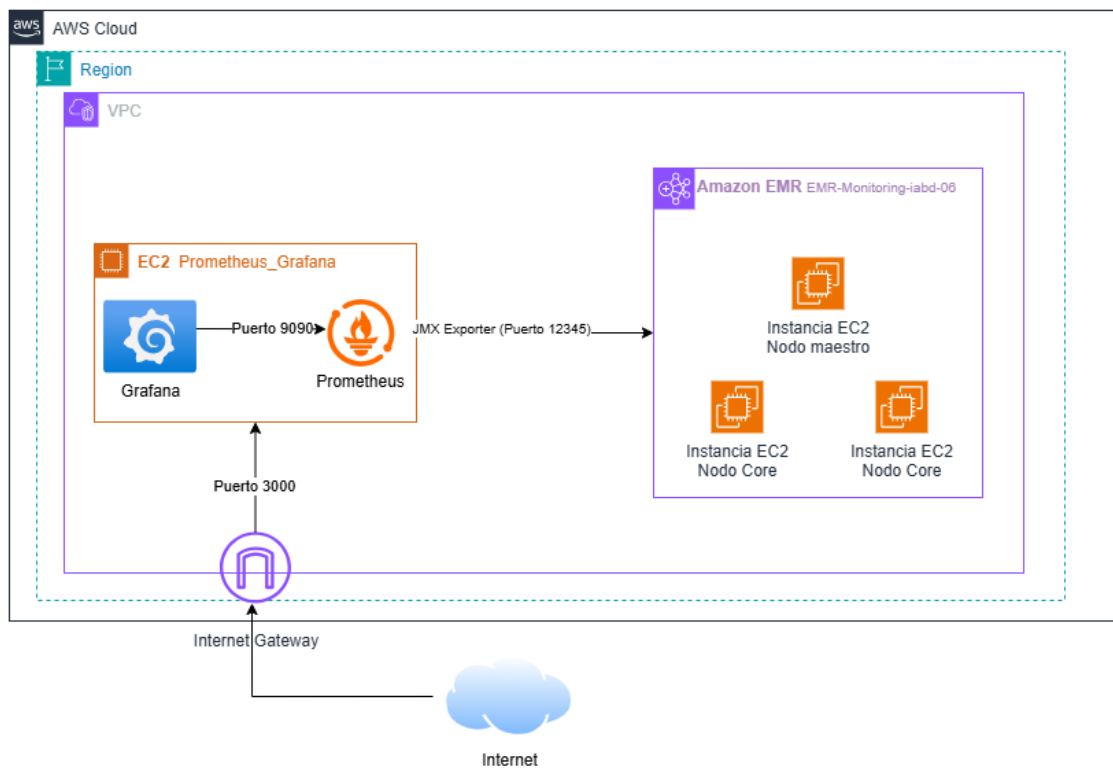


16-4-2025

Monitoreo de un clúster AWS EMR

Prometheus y Grafana



Álex González Puente
IES ATAULFO ARGENTA

ÍNDICE

Objetivo	2
Instrucciones	2
1. Configuración del clúster AWS EMR.....	2
1.1. Crear un clúster EMR	2
1.2. Conectar al nodo maestro	7
2. Configuración de JMX Exporter	9
2.1. Instalar JMX Exporter	9
2.2. Crear el archivo de configuración.....	9
2.3. Configurar el NameNode para usar JMX Exporter.....	10
2.4. Reiniciar el NameNode	11
3. Despliegue de Prometheus y Grafana	12
3.1. Crear una instancia EC2 para Prometheus y Grafana.....	12
3.2. Instalar Prometheus	14
3.3. Configurar Prometheus	15
3.4. Instalar Grafana.....	16
3.5. Configurar Grafana	18
4. Visualización de métricas en Grafana	20
4.1. Crear un dashboard en Grafana	20
4.2. Explorar métricas.....	23
Reflexión	23
1. Métricas más importantes para monitorear	23
2. Posibles mejoras en la configuración de JMX Exporter	23
3. Ventajas de utilizar Prometheus y Grafana	23
Descargar archivos	24

Objetivo

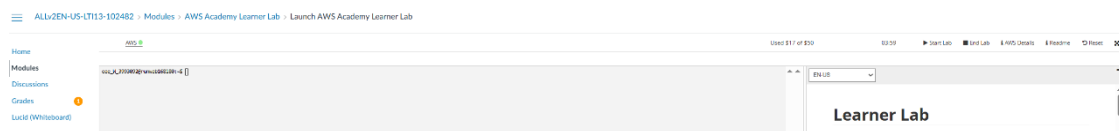
El objetivo de esta práctica es implementar un sistema de monitoreo para un clúster de procesamiento en AWS EMR, que permita supervisar su estado y rendimiento de forma centralizada. Para ello, se utilizarán herramientas de código abierto como JMX Exporter, Prometheus y Grafana, las cuales facilitarán la recopilación, exposición y visualización de métricas relevantes del clúster.

Instrucciones

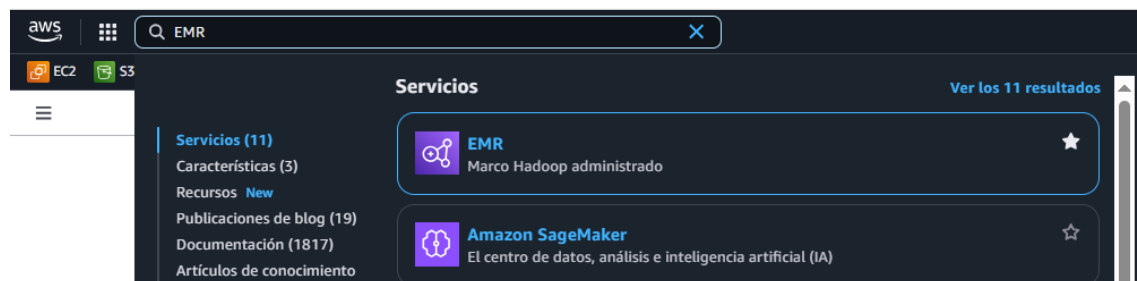
1. Configuración del clúster AWS EMR

1.1. Crear un clúster EMR

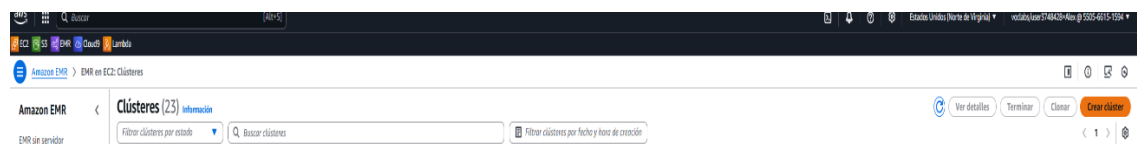
Para iniciar la creación de nuestro clúster, primero activaremos el laboratorio y esperaremos hasta que el sistema se inicie. Una vez se ilumine en verde, haremos clic en el ícono para acceder a la consola de gestión.



Se abrirá una nueva pestaña en la que deberemos utilizar la barra de búsqueda ubicada en la parte superior izquierda para escribir "EMR" y así acceder al servicio correspondiente de Amazon EMR.



En esta sección se mostrarán todos nuestros clústeres existentes. Para crear uno nuevo, debemos hacer clic en la opción “Crear clúster”.



Ahora procederemos a configurar los parámetros para crear nuestro clúster Hadoop con la siguiente estructura:

- 1 Nodo Maestro (Master).
- 2 Nodos Núcleo (Core).

Para ello, comenzaremos asignando un nombre a nuestro clúster, que en este caso será "**EMR-Monitoring-iabdXX**". A continuación, seleccionaremos la versión "**emr-6.6.0**", con las aplicaciones **Hadoop**, **Spark** y **Hive**.

Crear clúster [Información](#)

▼ Nombre y aplicaciones - *obligatorio* [Información](#)

Asigne un nombre a su clúster y elija las aplicaciones que desea instalar en él.

Nombre

Versión de Amazon EMR [Información](#)

Una versión contiene un conjunto de aplicaciones que se puede instalar en el clúster.

Paquete de aplicaciones

Spark 	Core Hadoop 	HBase 	Presto 	Trino 	Custom 
--	--	--	---	--	---

- | | | |
|---|---|--|
| <input type="checkbox"/> Flink 1.14.2 | <input type="checkbox"/> Ganglia 3.7.2 | <input type="checkbox"/> HBase 2.4.4 |
| <input type="checkbox"/> HCatalog 3.1.2 | <input checked="" type="checkbox"/> Hadoop 3.2.1 | <input checked="" type="checkbox"/> Hive 3.1.2 |
| <input type="checkbox"/> Hue 4.10.0 | <input type="checkbox"/> JupyterEnterpriseGateway 2.1.0 | <input type="checkbox"/> JupyterHub 1.4.1 |
| <input type="checkbox"/> Livy 0.7.1 | <input type="checkbox"/> MXNet 1.8.0 | <input type="checkbox"/> Oozie 5.2.1 |
| <input type="checkbox"/> Phoenix 5.1.2 | <input type="checkbox"/> Pig 0.17.0 | <input type="checkbox"/> Presto 0.267 |
| <input checked="" type="checkbox"/> Spark 3.2.0 | <input type="checkbox"/> Sqoop 1.4.7 | <input type="checkbox"/> TensorFlow 2.4.1 |
| <input type="checkbox"/> Tez 0.9.2 | <input type="checkbox"/> Trino 367 | <input type="checkbox"/> Zeppelin 0.10.0 |
| <input type="checkbox"/> ZooKeeper 3.5.7 | | |

Configuración del Catálogo de datos de AWS Glue

Utilice el Catálogo de datos de AWS Glue para proporcionar un meta-almacén externo a la aplicación.

- ☐ Usar para metadatos de la tabla de Hive
☐ Usar para metadatos de la tabla de Spark

Opciones del sistema operativo [Información](#)

- ☒ Versión de Amazon Linux
☐ Imagen de máquina de Amazon (AMI) personalizada
☒ Aplicar automáticamente las actualizaciones más recientes de Amazon Linux

Posteriormente, eliminaremos el grupo de instancias "Tarea 1".

Tarea 1 de 1

[Eliminar grupo de instancias](#)

Nombre

Elegir tipo de instancia de EC2

[Acciones ▼](#)

► Configuración de nodo - *opcional*

Seleccionamos "**m4.large**" como tipo de instancia tanto para el nodo Master como para los nodos Core. Debido a que necesitamos 2 nodos Core, ajustamos la configuración para establecer el **tamaño de instancias** en **2**.

▼ **Configuración del clúster - obligatorio** [Información](#)
Elija un método de configuración para los grupos principales, centrales y de nodos tareas para su clúster.

☒ **Grupos de instancias uniformes**
Elija el mismo tipo de instancia de EC2 y la misma opción de compra (bajo demanda o de spot) para todos los nodos de su grupo de nodos. [Más información](#)

☐ **Flotas de instancias flexibles**
Elija entre la más amplia variedad de opciones de aprovisionamiento para las instancias de EC2 de su clúster. Diversifique los tipos de instancias y las opciones de compra, y utilice una estrategia de asignación. [Más información](#)

Grupos de instancias uniformes

Principal

Elegir tipo de instancia de EC2

m4.large
2 vCore 8 GiB memoria
Únicamente EBS almacenamiento
Precio bajo demanda: -
Precio de spot más bajo: -

Acciones ▼

☐ **Utilice la alta disponibilidad**
Lance un clúster más resiliente y de alta disponibilidad con tres nodos principales en instancias bajo demanda. Esta configuración se aplica durante toda la vida útil del clúster. [Más información](#)

► **Configuración de nodo - opcional**

Central Eliminar grupo de instancias

Elegir tipo de instancia de EC2

m4.large
2 vCore 8 GiB memoria
Únicamente EBS almacenamiento
Precio bajo demanda: -
Precio de spot más bajo: -

Acciones ▼

► **Configuración de nodo - opcional**

Agregar grupo de instancias de tareas
Puede agregar hasta 48 grupos más de instancias de tareas.

Volumen raíz de EBS
El volumen raíz de EBS se aplica a los sistemas operativos y las aplicaciones que instale en el clúster.

Tamaño (GiB)

10

10- 100 GiB por volumen SSD de uso general (gp2)

▼ **Aprovisionamiento y escalado de clústeres - obligatorio** [Información](#)
Elija cómo Amazon EMR debe dimensionar su clúster.

Elija una opción

☒ **Establecer el tamaño del clúster manualmente**
Utilice esta opción si conoce los patrones de la carga de trabajo de antemano.

☐ **Utilizar escalado administrado por EMR**
Supervise las métricas clave de la carga de trabajo de modo que EMR pueda optimizar el tamaño del clúster y la utilización de los recursos.

☐ **Utilizar el escalamiento automático personalizado**
Para escalar mediante programación los nodos principales y los nodos de tarea, cree políticas de escalamiento automático personalizadas.

Configuración de aprovisionamiento
Establezca el tamaño del principal grupo de instancias. Amazon EMR intenta aprovisionar esta capacidad al lanzar el clúster.

Nombre	Tipo de instancia	Tamaño de instancia(s)	Utilizar la opción de compra de spot
Central	m4.large	2	<input type="checkbox"/>

Pondremos **4 horas** para que el cluster termine en caso de inactividad.

▼ **Terminación del clúster y reemplazo de nodos** [Información](#)

Elija la configuración de terminación y proteja su clúster contra un apagado accidental.

Opción de terminación

☐ Terminar manualmente el clúster

☐ Terminar automáticamente el clúster después de que finalice el último paso

☒ Terminar el clúster después del tiempo de inactividad (recomendado)

Tiempo de inactividad

Ingrese el tiempo hasta que el clúster termine.

0 días ▼

04:00:00

Elija una hora mayor a 1 minuto (00:01:00) y menor a 7 días. La hora está en formato hh:mm:ss (24 horas).

☐ Use la protección contra la terminación

Protege al clúster para evitar una terminación accidental. Si está activada, deberá primero desactivar la protección para terminar el clúster. Recomendamos activar la protección frente a terminaciones para los clústeres de larga duración.

Reemplazo de nodos en mal estado - novedad | [Información](#)

☒ Activar

Amazon EMR detiene correctamente los procesos en los nodos en mal estado para minimizar la pérdida de datos y las interrupciones del trabajo. Reemplaza rápidamente los nodos en mal estado por nuevas instancias de EC2 para que sus trabajos funcionen sin problemas.

En la sección de redes, se conservará la **VPC predeterminada**. Para permitir el acceso posterior al clúster mediante SSH, se generará un par de claves denominado “**vockey**”.

▼ **Configuración de seguridad y par de claves de EC2** [Información](#)

Elija una configuración de seguridad o cree una nueva que pueda reutilizar con otros clústeres.

Configuración de seguridad

Seleccione la configuración del servicio de cifrado, autenticación, autorización y metadatos de instancia del clúster.

[Examinar](#)

[Crear configuración de seguridad](#)

Par de claves de Amazon EC2 para el protocolo SSH al clúster | [Información](#)

[Examinar](#)

[Crear par de claves](#)

Por último, configuramos los roles, seleccionando la opción de “existente” para ambos. Los roles que asignaremos son los siguientes:

- **Rol de servicio:** LabRole.
- **Perfil de instancia:** EMR_EC2_DefaultRole.

Roles de Identity and Access Management (IAM) - obligatorio [Información](#)

Elija o cree un rol de servicio y un perfil de instancia para las instancias de EC2 del clúster.

Rol de servicio de Amazon EMR [Información](#)

El rol de servicio es un rol de IAM que Amazon EMR asume para aprovisionar recursos y realizar acciones de nivel de servicio con otros servicios de AWS.

☒ **Elegir un rol de servicio existente**

Seleccione un rol de servicio predeterminado o un rol personalizado con políticas de IAM asociadas para que el clúster pueda interactuar con otros servicios de AWS.

☐ **Crear un rol de servicio**

Deje que Amazon EMR cree un nuevo rol de servicio para que pueda conceder y restringir el acceso a los recursos de otros servicios de AWS.

Rol de servicio

LabRole

Perfil de instancia de EC2 para Amazon EMR

El perfil de instancia asigna un rol a cada instancia de EC2 de un clúster. El perfil de instancia debe especificar un rol que pueda acceder a los recursos de los pasos y las acciones de arranque.

☒ **Elegir un perfil de instancia existente**

Seleccione un rol predeterminado o un perfil de instancia personalizado con políticas de IAM asociadas para que el clúster pueda interactuar con sus recursos de Amazon S3.

☐ **Crear un perfil de instancia**

Deje que Amazon EMR cree un nuevo perfil de instancia para que pueda especificar un conjunto personalizado de recursos a los que tendrá acceso en Amazon S3.

Perfil de instancia

EMR_EC2_DefaultRole

Rol de escalamiento automático personalizado - opcional

Cuando se activa una regla de escalamiento automático personalizada, Amazon EMR asume esta función para agregar y finalizar instancias de EC2. [Más información](#)

Rol de escalamiento automático personalizado

Elegir rol de IAM

Crear rol de IAM

Si hemos configurado todo correctamente, el resultado final debería ser el siguiente:

Resumen [Información](#)

Nombre y aplicaciones - obligatorio

Nombre
EMR-Monitoring-iabd06

Versión de Amazon EMR
emr-6.6.0

Paquete de aplicaciones
Custom (Hadoop 3.2.1, Hive 3.1.2, Spark 3.2.0)

Configuración del clúster - obligatorio

Grupos de instancias uniformes
Principal (m4.large), Central (m4.large)

Aprovisionamiento y escalado de clústeres - obligatorio

Configuración de aprovisionamiento
Tamaño del núcleo: 2 instancias

Redes - obligatorio

Resumen [Información](#)

Redes - obligatorio

VPC
[vpc-0611d8f6c...](#)

Subred
[subnet-000163...](#)

Grupos de seguridad de nodos principales
[sg-00be73a24b...](#)

Grupos de seguridad de nodos básicos
[sg-06838c1a61...](#)

Terminación del clúster

Terminación del clúster
Terminar el clúster después del tiempo de inactividad
Tiempo de inactividad: 4 horas

Registros de clúster

Resumen [Información](#)

Terminación del clúster
Terminar el clúster después del tiempo de inactividad
Tiempo de inactividad: 4 horas

Registros de clúster

Ubicación de Amazon S3
[s3://aws-logs...](#)

Configuración de seguridad y par de claves de EC2

Par de claves de Amazon EC2
[vockey](#)

Roles de Identity and Access Management (IAM) - obligatorio

Rol de servicio
[LabRole](#)

Perfil de instancia
[EMR_EC2_DefaultRole](#)

Cancelar

Crear clúster

Finalmente, para completar la creación del clúster, hacemos clic en “Crear Clúster”. Una vez que las instancias se hayan iniciado, nuestro clúster debería aparecer en estado “Esperando”.

The screenshot shows the AWS EMR console for a cluster named 'EMR-Monitoring-lab06'. The cluster is in the 'Waiting' state. The console displays various details about the cluster, including its ID, ARN, configuration, and the instances it contains. The 'Instances (hardware)' tab is selected, showing a list of instances with their IDs, names, and states.

Tipo y nombre	ID	Estado	Instancias	Opción de compra y precio	Tamaño de EBS (GiB)	ID de instancia EC...	Nombre
Principal	ig-2631IC0GBW6XX	En ejecución	1	Bajo demanda	-	-	-
Principal (Central)	ig-15ML8C55V6CC9	En ejecución	2	Bajo demanda	-	-	-

1.2. Conectar al nodo maestro

Tras la configuración previa, procederemos a conectarnos al nodo maestro. Como previamente creamos las claves vockey para autenticarnos, necesitamos acceder nuevamente a la pestaña donde encendimos nuestro laboratorio. Allí, seleccionamos "AWS Details" y descargamos el archivo PEM, que contiene nuestra clave privada para acceder de manera segura.

The screenshot shows the AWS Cloud Access page. The 'AWS CLI' section is expanded, showing the 'Download PEM' button. The page also displays the 'Session started at' and 'Session to end at' times, as well as the 'Accumulated lab time'.

Historial de descargas recientes

labsuser.pem
1.678 B • Hecho

Ahora, abrimos CMD, nos dirigimos a la ubicación donde se encuentra el archivo que acabamos de descargar (labuser.pem) y ejecutamos el siguiente comando:

```
ssh -i /ruta/a/tu/archivo.pem hadoop@<ip-nodo-maestro>
```

Conectarse al nodo principal mediante SSH

Puede conectarse al nodo principal de Amazon EMR mediante SSH para realizar acciones como ejecutar consultas interactivas, examinar archivos de registro, enviar comandos de Linux y ver interfaces Web alojadas en clústeres de Amazon EMR. [Más información](#)

Windows | Mac/Linux

1. Abra una ventana de terminal. En Mac OS X, elija Applications (Aplicaciones) > Utilities (Utilidades) > Terminal. En otras distribuciones de Linux, el terminal suele encontrarse en Aplicaciones (Aplicaciones) > Accessories (Accesorios) > Terminal.

2. Para establecer una conexión con el nodo principal, escriba el siguiente comando. Sustituya `~/voockey.pem` por la ubicación y el nombre de archivo del archivo de clave privada (.pem) que utilizó para lanzar el clúster.

```
ssh -i ~/vockey.pem hadoop@ec2-18-234-106-153.compute-1.amazonaws.com
```

3. Escriba Yes (Sí) para descartar la advertencia de seguridad.

[Ver interfaces Web alojadas en clústeres de Amazon EMR](#)

Cerrar

Para encontrar este comando, debemos dirigirnos al resumen de nuestro EMR y verificar nuestro DNS público del nodo principal.

Administración de clústeres

Destino del registro en Amazon S3

[aws-logs-550566151594-us-east-1/elasticmapreduce](#)

IU de aplicación persistente

Servidor de historial de Spark [↗](#)

[Servidor de línea de tiempo de YARN](#)

[UI de Tez](#)

DNS público del nodo principal

ec2-18-234-106-153.compute-1.amazonaws.com


Conectarse al nodo principal mediante SSH

[Conectarse al nodo principal mediante SSM](#)

Al establecer la conexión nos deberá aparecer lo siguiente:

```
C:\Users\alex>cd Downloads

C:\Users\alex>Downloads>ssh -i labsuser.pem hadoop@ec2-18-234-106-153.compute-1.amazonaws.com
The authenticity of host 'ec2-18-234-106-153.compute-1.amazonaws.com (18.234.106.153)' can't be established.
ED25519 key fingerprint is SHA256:7KXidwmQcuu9ScfJVteHlONV83ubKRwDVGK3jpjDX0Po.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-234-106-153.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
```



```
#
##### Amazon Linux 2
##### \
##### \ AL2 End of Life is 2026-06-30.
# / V
^ ^ ^ ^ ^
^ ^ ^ ^ ^ A newer version of Amazon Linux is available!
^ ^ ^ ^ ^
^ ^ ^ ^ ^ Amazon Linux 2023, GA and supported until 2028-03-15.
m / ^ ^ ^ ^ ^ https://aws.amazon.com/linux/amazon-linux-2023/
```

```
EEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRR
E:::EEEEEEEEEEEEEE M:::~M M:::~M R:::~::~~R
EE:::EEEEEEEEEEEEEE M:::~M M:::~M R:::~RRRRRR::~R
E:::E EEEEE M:::~M M:::~M RR::~R R:::~R
E:::E M:::~M M:::~M M:::~M R:::~R R:::~R
E:::EEEEEEEEEE M:::~M M:::~M M:::~M R:::~RRRRRR::~R
E:::EEEEEEEEEE M:::~M M:::~M M:::~M R:::~RRRRRR::~R
E:::EEEEEEEEEE M:::~M M:::~M M:::~M R:::~RRRRRR::~R
E:::E M:::~M M:::~M M:::~M R:::~R R:::~R
E:::E EEEEE M:::~M MMM M:::~M R:::~R R:::~R
EE:::EEEEEEEEEEEEEE M:::~M M:::~M R:::~R R:::~R
E:::EEEEEEEEEEEEEE M:::~M M:::~M RR::~R R:::~R
EEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRR RRRRRR
```

```
[hadoop@ip-172-31-51-126 opt]$
```

2. Configuración de JMX Exporter

2.1. Instalar JMX Exporter

JMX Exporter es una herramienta que permite extraer métricas de aplicaciones Java, como Hadoop o Spark, a través de JMX y exponerlas en un formato compatible con Prometheus. En el contexto de un clúster AWS EMR, se utiliza para monitorear el estado y rendimiento de los servicios del clúster, facilitando su visualización en herramientas como Grafana sin necesidad de modificar el código de las aplicaciones.

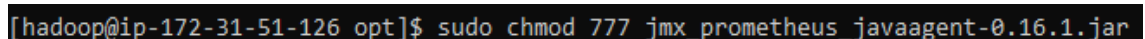
Para llevar a cabo la instalación de JMX Exporter, debemos acceder al directorio /opt en el nodo maestro y ejecutar el siguiente comando para descargar el agente:

```
cd /opt
wget
https://repo1.maven.org/maven2/io/prometheus/jmx/jmx_prometheus_javaagent/0.16.1/jmx_prometheus_javaagent-0.16.1.jar
```



Una vez descargado el archivo, es necesario otorgarle los permisos adecuados para su ejecución. Para ello, se utiliza el siguiente comando:

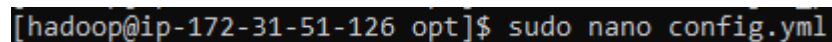
```
chmod 777 jmx_prometheus_javaagent-0.16.1.jar
```



2.2. Crear el archivo de configuración

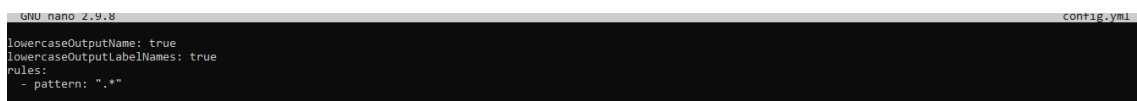
A continuación, crearemos el archivo de configuración que utilizaremos con JMX Exporter para definir las métricas que deseamos recolectar. Para ello, ejecutamos el siguiente comando:

```
nano config.yml
```



Le agregaremos el siguiente contenido:

```
lowercaseOutputName: true
lowercaseOutputLabelNames: true
rules:
- pattern: ".*"
```



Este bloque de configuración tiene el siguiente propósito:

- **lowercaseOutputName:** true: convierte los nombres de las métricas a minúsculas para mantener un formato uniforme y compatible con las convenciones de Prometheus.
- **lowercaseOutputLabelNames:** true: aplica el mismo criterio a los nombres de las etiquetas (labels), asegurando consistencia en los datos recolectados.
- **rules:** define las reglas de extracción de métricas desde JMX. En este caso, el patrón "*" indica que se expondrán todas las métricas disponibles, sin aplicar filtros específicos.

Finalmente, le damos permisos de igual forma que al archivo anterior.

```
[hadoop@ip-172-31-51-126 opt]$ sudo chmod 777 config.yml
```

2.3. Configurar el NameNode para usar JMX Exporter

Editamos el archivo de configuración del **NameNode** ejecutando el siguiente comando:

```
sudo nano /etc/hadoop/conf/hadoop-env.sh
```

```
[hadoop@ip-172-31-51-126 opt]$ sudo nano /etc/hadoop/conf/hadoop-env.sh
```

A continuación, agregamos la siguiente línea al archivo de configuración:

```
export HADOOP_NAMENODE_OPTS="-
javaagent:/home/hadoop/jmx_prometheus_javaagent-
0.16.1.jar=12345:/home/hadoop/config.yml $HADOOP_NAMENODE_OPTS"
```

```

# where log files are stored. $HADOOP_HOME/logs by default.
#export HADOOP_LOG_DIR=

# File naming remote slave hosts. $HADOOP_HOME/conf/slaves by default.
#export HADOOP_WORKERS=

# host:path where hadoop code should be rsync'd from. Unset by default.
#export HADOOP_MASTER=

# Seconds to sleep between slave commands. Unset by default. This
# can be useful in large clusters, where, e.g., slave rsyncs can
# otherwise arrive faster than the master can service them.
#export HADOOP_WORKER_SLEEP=

# The directory where pid files are stored. /tmp by default.
#export HADOOP_PID_DIR=

# A string representing this instance of hadoop. $USER by default.
#export HADOOP_IDENT_STRING=

# The scheduling priority for daemon processes. See 'man nice'.
#export HADOOP_NICENESS=

# tez environment, needed to enable tez
#export TEZ_CONF_DIR=/etc/tez/conf
#export TEZ_JARS=/usr/lib/tez
# Add tez into HADOOP_CLASSPATH
#export HADOOP_CLASSPATH:$HADOOP_CLASSPATH:$(TEZ_CONF_DIR):$(TEZ_JARS):/*:$(TEZ_JARS)/lib/*
#export HADOOP_CLASSPATH:"$HADOOP_CLASSPATH:/usr/lib/hadoop-lzo/lib/*"
#export JAVA_LIBRARY_PATH="$JAVA_LIBRARY_PATH:/usr/lib/hadoop-lzo/lib/native"

#export HADOOP_CLASSPATH:$HADOOP_CLASSPATH:/usr/share/aws/aws-java-sdk/*
#export HADOOP_CLASSPATH:"$HADOOP_CLASSPATH:/usr/share/aws/emr/emrfs/conf:/usr/share/aws/emr/emrfs/lib/*:/usr/share/aws/emr/emrfs/auxlib/*"
#export HADOOP_CLASSPATH:"$HADOOP_CLASSPATH:/usr/share/aws/emr/ddb/lib/emr-ddb-hadoop.jar"
#export HADOOP_CLASSPATH:"$HADOOP_CLASSPATH:/usr/share/aws/emr/goodies/lib/emr-hadoop-goodies.jar"
#export HADOOP_CLASSPATH:"$HADOOP_CLASSPATH:/usr/share/aws/emr/kinesis/lib/emr-kinesis-hadoop.jar"

# Add CloudWatch sink jar to classpath
#export HADOOP_CLASSPATH:"$HADOOP_CLASSPATH:/usr/share/aws/emr/cloudwatch-sink/lib/*"

# Add security artifacts to classpath
#export HADOOP_CLASSPATH:"$HADOOP_CLASSPATH:/usr/share/aws/emr/security/conf:/usr/share/aws/emr/security/lib/*"

#export HADOOP_OPTS="$HADOOP_OPTS -server -XX:+ExitOnOutOfMemoryError"
#export HADOOP_NAMENODE_HEAPSIZE=1024
#export HADOOP_DATANODE_HEAPSIZE=614
#export HADOOP_JOB_HISTORYSERVER_HEAPSIZE=2252
#export HADOOP_NAMENODE_OPTS="-javaagent:/home/hadoop/jmx_prometheus_javaagent-0.16.1.jar=12345:/home/hadoop/config.yml $HADOOP_NAMENODE_OPTS"
```

Esta línea configura el agente de **JMX Exporter** para que se ejecute en el **NameNode**, especificando la ubicación del archivo `jmx_prometheus_javaagent-0.16.1.jar` y el archivo de configuración `config.yml`. Además, define el puerto 12345 para la exposición de las métricas a Prometheus.

2.4. Reiniciar el NameNode

Finalmente, reiniciamos el servicio del NameNode para que los cambios realizados en la configuración surtan efecto. Para hacerlo, ejecutamos el siguiente comando:

```
sudo systemctl restart hadoop-hdfs-namenode
```

```
[hadoop@ip-172-31-62-212 opt]$ [hadoop@ip-172-31-62-212 opt]$ sudo systemctl restart hadoop-hdfs-namenode
```

Tras esperar a que el servicio se reinicie, podremos verificar si el proceso se ha realizado correctamente. Para ello, debemos acceder a las aplicaciones de nuestro EMR, ingresar en Hadoop, y dentro de este, desplegar la opción "Utilities" y entrar en "Metrics".

The screenshot shows the Hadoop Overview page for the NameNode instance 'ip-172-31-63-31.ec2.internal:8020' (active). The page includes a table with system information and a detailed view of JMX metrics.

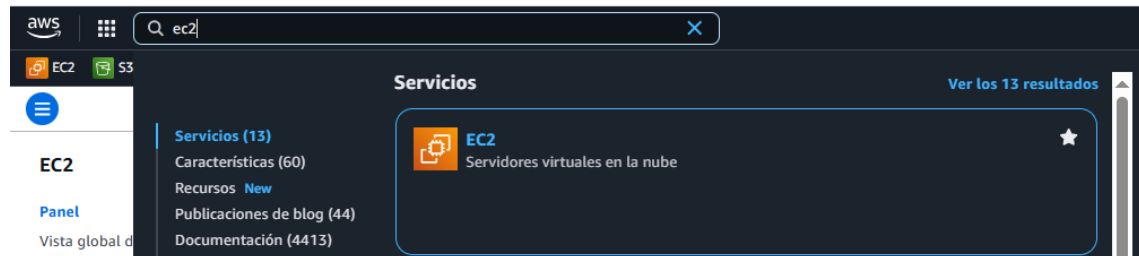
Field	Value
Started:	Mon Apr 07 18:23:22 +0000 2025
Version:	3.2.1-amzn2-4, RUNKNOWN
Compiled:	Tue Apr 05 19:08:00 +0000 2022 by release from Unknown
Cluster ID:	CID-af47480a-ec04-43d0-b0c1-90c4895801a7
Block Pool ID:	BP-1775342470-172-31-63-31-1744042375214

The JMX metrics section displays various system and JVM statistics, including memory usage, buffer pool statistics, and garbage collection metrics. The metrics are organized into sections for different JVM components and memory areas.

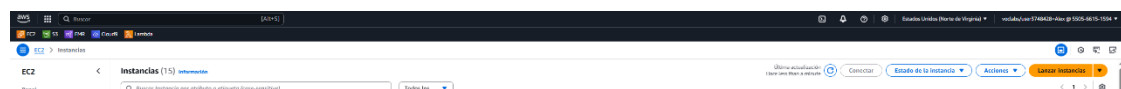
3. Despliegue de Prometheus y Grafana

3.1. Crear una instancia EC2 para Prometheus y Grafana

El siguiente paso será crear una instancia EC2 con Ubuntu en la misma VPC que el clúster EMR. Para ello, debemos volver a utilizar la barra de búsqueda ubicada en la parte superior izquierda, pero esta vez escribiendo “EC2”.



En esta sección se mostrarán todas nuestras instancias existentes. Para crear una nueva, debemos hacer clic en la opción “Lanzar instancias”.



Ahora procederemos a configurar los parámetros para crear nuestra nueva instancia de la siguiente forma:

- **Nombre:** Prometheus_Grafana.
- **Imágenes de máquina de Amazon (AMI):** Ubuntu Server 22.04 LTS (HVM), SSD Volume Type.
- **Tipo de instancia:** t2.micro.

Lanzar una instancia Información

Amazon EC2 le permite crear máquinas virtuales, o instancias, que se ejecutan en la nube de AWS. Comience rápidamente siguiendo los sencillos pasos que se indican a continuación.

Nombre y etiquetas Información

Nombre: [Agregar etiquetas adicionales](#)

▼ Imágenes de aplicaciones y sistemas operativos (Imagen de máquina de Amazon) Información

Una AMI es una plantilla que contiene la configuración de software (sistema operativo, servidor de aplicaciones y aplicaciones) necesaria para lanzar la instancia. Busque o examine las AMI si no ve lo que busca a continuación.

Recientes Mis AMI **Inicio rápido**

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Debian

Imágenes de máquina de Amazon (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
ami-0f9d6c2d2f067fca (64 bits, x86_64) / ami-0f9d6c2d2f067fca (64 bits, ARMv8)
Virtualización: hvm Activado para HVM: true Tipo de dispositivo raíz: ebs

Descripción

Canonical, Ubuntu, 22.04, amd64 jammy image

Arquitectura	ID de AMI	Fecha de publicación	Nombre de usuario
64 bits (x86)	ami-0f9d6c2d2f067fca	2023-05-05	ubuntu

[Promotor verificado](#)

▼ Tipo de instancia Información | Obtener asesoramiento

Tipo de instancia

t2.micro

Familia: t2 1 vCPU 1 GB Memoria Generación actual: true

☐ Todas las generaciones

[Comparar tipos de instancias](#)

Se aplican costos adicionales a las AMI con software preinstalado

Asignamos el par de claves "vockey" para permitir la conexión SSH y poder realizar los pasos siguientes. En cuanto a la **VPC**, utilizamos la predeterminada, tal como se configuró en el clúster EMR, asegurando que la comunicación. Además, seleccionamos el mismo grupo de seguridad que se usa en el clúster EMR para garantizar que ambas instancias puedan interactuar.

▼ Par de claves (inicio de sesión) Información

Puede utilizar un par de claves para conectarse de forma segura a la instancia. Asegúrese de que tiene acceso al par de claves seleccionado antes de lanzar la instancia.

Nombre del par de claves - obligatorio

vockey

🔄 Crear un nuevo par de claves

▼ Configuraciones de red Información Editar

Red | Información
vpc-0611d8f6c989743f

Subred | Información
Sin preferencias (subred predeterminada en cualquier zona de disponibilidad)

Asignar automáticamente la IP pública | Información
Habilitar

Se aplican cargos adicionales cuando no se cumplen los límites del nivel gratuito

Firewall (grupos de seguridad) | Información
Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agregue reglas para permitir que un tráfico específico llegue a la instancia.

☐ Crear grupo de seguridad
☒ Seleccionar un grupo de seguridad existente

Grupos de seguridad comunes | Información

Seleccionar grupos de seguridad

ElasticMapReduce-master sg-00be73a24bf96c95e X
VPC: vpc-0611d8f6c989743f

🔄 Compare reglas de grupo de seguridad

Los grupos de seguridad que agrega o elimine aquí se agregarán a todas las interfaces de red o se eliminarán de ellas.

▼ Configurar almacenamiento Información Avanzado

1x

8

 GiB

gp2

 Volumen raíz, No cifrado

☒ Los clientes que cumplan los requisitos de la capa gratuita pueden obtener hasta 30 GB de almacenamiento magnético o de uso general (SSD) de EBS

Agregar un nuevo volumen

La AMI seleccionada contiene más volúmenes de almacén de instancias de los que permite la instancia. Solo se podrá obtener acceso desde la instancia a los primeros 0 volúmenes de almacén de instancias de la AMI

☒ Haga clic en actualizar para ver la información de la copia de seguridad
Las etiquetas que asigna determinan si alguna política de Data Lifecycle Manager realizará una copia de seguridad de la instancia.

0 x sistemas de archivos

Edit

El resumen deberá quedar de la siguiente manera:

▼ Resumen

Número de instancias | Información

1

Imagen de software (AMI)
Canonical, Ubuntu, 22.04, amd64...más información
ami-0f9de6e2d2f067fca

Tipo de servidor virtual (tipo de instancia)
t2.micro

Firewall (grupo de seguridad)
ElasticMapReduce-master

Almacenamiento (volúmenes)
Volúmenes: 1 (8 GiB)

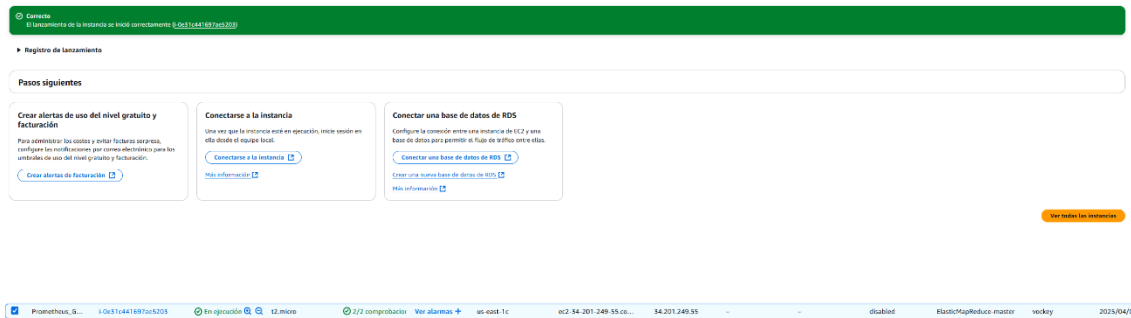
☒ **Nivel gratuito:** Durante el primer año que abre una cuenta de AWS, obtiene 750 horas al mes de uso de instancias t2.micro (o t3.micro cuando t2.micro no esté disponible) si se utiliza con AMI de nivel gratuito, 750 horas al mes de uso de direcciones IPv4 públicas, 30 GiB de almacenamiento de EBS, 2 millones de E/S, 1 GB de instantáneas y 100 GB de ancho de banda para Internet.

Cancelar

Lanzar instancia

📄 Código de versión preliminar

Después de hacer clic en "Lanzar instancia", se creará la nueva instancia y deberemos esperar a que se inicie correctamente.



3.2. Instalar Prometheus

Una vez que la instancia esté en funcionamiento, podremos conectarnos de la misma manera que lo hicimos con el clúster EMR, pero en este caso, utilizaremos el siguiente comando:

```
ssh -i /ruta/a/tu/archivo.pem ubuntu@<dirección pública de la instancia EC2>
```

```
ubuntu@ip-172-31-87-236: ~
System information as of Mon Apr  7 16:54:17 UTC 2025

System load:  0.0          Processes:            107
Usage of /:   21.8% of 7.57GB Users logged in:             0
Memory usage: 21%         IPv4 address for eth0: 172.31.87.236
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-87-236:~$
```


Una vez dentro de la instancia, procederemos a descargar y ejecutar **Prometheus** con los siguientes comandos:

```
wget
https://github.com/prometheus/prometheus/releases/download/v2.30.3/prometheus-
2.30.3.linux-amd64.tar.gz
tar -xzf prometheus-2.30.3.linux-amd64.tar.gz
cd prometheus-2.30.3.linux-amd64
./prometheus --config.file=prometheus.yml
```

[illegible]

3.3. Configurar Prometheus

A continuación, editamos el archivo “prometheus.yml” para agregar el clúster EMR como objetivo de monitoreo. Para ello, volvemos a hacer usos del comando “nano” y agregamos el siguiente bloque de configuración:

```
scrape_configs:
- job_name: 'emr-namenode'
  static_configs:
    - targets: ['18.234.106.153:12345']
scrape_configs:
- job_name: 'emr-namenode'
  static_configs:
    - targets: ['<ip-nodo-maestro>:12345']
```

```
ubuntu@ip-172-31-87-236:~/prometheus-2.30.3.linux-amd64$ nano prometheus.yml
```

```
# GNU nano 2.7.1
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
            # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]

  - job_name: 'emr-namenode'
    static_configs:
      - targets: ['18.234.106.153:12345']
```


Este bloque le indica a Prometheus que recolecte métricas del NameNode del clúster EMR, accediendo a la dirección IP del nodo maestro en el puerto 12345, que es donde configuramos previamente **JMX Exporter**.

3.4. Instalar Grafana

A continuación, instalamos **Grafana** en la misma instancia EC2 donde configuramos Prometheus. Para ello, ejecutamos los siguientes comandos:

Permite que APT use repositorios HTTPS

```
sudo apt-get install -y apt-transport-https
```

```
ubuntu@ip-172-31-87-236:~$ sudo apt-get install -y apt-transport-https
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apt-transport-https is already the newest version (2.4.13).
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
```

Instala herramientas para manejar repositorios y descargar archivos

```
sudo apt-get install -y software-properties-common wget
```

```
ubuntu@ip-172-31-87-236:~$ sudo apt-get install -y software-properties-common wget
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.22.9).
software-properties-common set to manually installed.
wget is already the newest version (1.21.2-2ubuntu1.1).
wget set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.
```

Agrega la clave del repositorio de Grafana

```
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
```

```
ubuntu@ip-172-31-87-236:~$ wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

Añade el repositorio oficial de Grafana

```
echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
```

```
ubuntu@ip-172-31-87-236:~$ echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
deb https://packages.grafana.com/oss/deb stable main
```

Actualiza la lista de paquetes

sudo apt-get update

```
ubuntu@ip-172-31-87-236:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://packages.grafana.com/oss/deb stable InRelease
Reading package lists... Done
W: https://packages.grafana.com/oss/deb/dists/stable/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
W: Target Packages (main/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list.d/grafana.list:1 and /etc/apt/sources.list.d/grafana.list:2
W: Target Packages (main/binary-all/Packages) is configured multiple times in /etc/apt/sources.list.d/grafana.list:1 and /etc/apt/sources.list.d/grafana.list:2
W: Target Translations (main/i18n/Translation-en) is configured multiple times in /etc/apt/sources.list.d/grafana.list:1 and /etc/apt/sources.list.d/grafana.list:2
W: Target CNF (main/cnf/Commands-amd64) is configured multiple times in /etc/apt/sources.list.d/grafana.list:1 and /etc/apt/sources.list.d/grafana.list:2
W: Target CNF (main/cnf/Commands-all) is configured multiple times in /etc/apt/sources.list.d/grafana.list:1 and /etc/apt/sources.list.d/grafana.list:2
W: Target Packages (main/binary-amd64/Packages) is configured multiple times in /etc/apt/sources.list.d/grafana.list:1 and /etc/apt/sources.list.d/grafana.list:2
W: Target Packages (main/binary-all/Packages) is configured multiple times in /etc/apt/sources.list.d/grafana.list:1 and /etc/apt/sources.list.d/grafana.list:2
W: Target Translations (main/i18n/Translation-en) is configured multiple times in /etc/apt/sources.list.d/grafana.list:1 and /etc/apt/sources.list.d/grafana.list:2
W: Target CNF (main/cnf/Commands-amd64) is configured multiple times in /etc/apt/sources.list.d/grafana.list:1 and /etc/apt/sources.list.d/grafana.list:2
W: Target CNF (main/cnf/Commands-all) is configured multiple times in /etc/apt/sources.list.d/grafana.list:1 and /etc/apt/sources.list.d/grafana.list:2
```

Instala Grafana

sudo apt-get install grafana

```
ubuntu@ip-172-31-87-236:~$ sudo apt-get install grafana
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  musl
The following NEW packages will be installed:
  grafana musl
0 upgraded, 2 newly installed, 0 to remove and 37 not upgraded.
Need to get 160 MB of archives.
After this operation, 631 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 musl amd64 1.2.2-4 [407 kB]
Get:2 https://packages.grafana.com/oss/deb stable/main amd64 grafana amd64 11.6.0 [169 MB]
Fetched 169 MB in 3s (58.4 MB/s)
Selecting previously unselected package musl:amd64.
(Reading database ... 65861 files and directories currently installed.)
Preparing to unpack .../musl_1.2.2-4_amd64.deb ...
Unpacking musl:amd64 (1.2.2-4) ...
Selecting previously unselected package grafana.
Preparing to unpack .../grafana_11.6.0_amd64.deb ...
Unpacking grafana (11.6.0) ...
Setting up musl:amd64 (1.2.2-4) ...
Setting up grafana (11.6.0) ...
Adding system user `grafana' (UID 115) ...
Adding new user `grafana' (UID 115) with group `grafana' ...
Not creating home directory `/usr/share/grafana'.
### NOT starting on installation, please execute the following statements to configure grafana to start automatically using systemd
  sudo /bin/systemctl daemon-reload
  sudo /bin/systemctl enable grafana-server
### You can start grafana-server by executing
  sudo /bin/systemctl start grafana-server
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Inicia Grafana y lo deja activo tras reiniciar

sudo systemctl start grafana-server

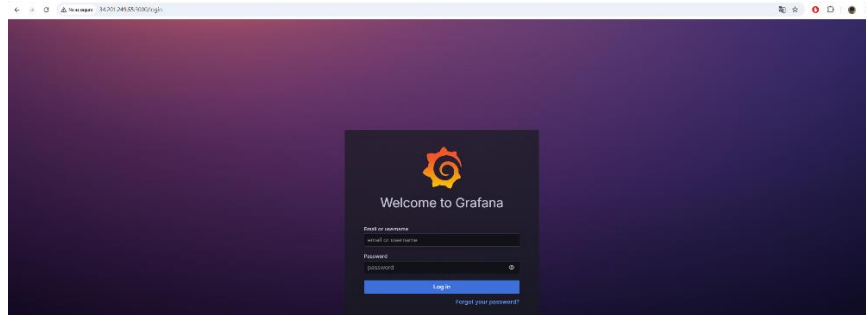
sudo systemctl enable grafana-server

```
ubuntu@ip-172-31-87-236:~$ sudo systemctl start grafana-server
ubuntu@ip-172-31-87-236:~$ sudo systemctl enable grafana-server
Synchronizing state of grafana-server.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable grafana-server
Created symlink /etc/systemd/system/multi-user.target.wants/grafana-server.service → /lib/systemd/system/grafana-server.service.
```

3.5. Configurar Grafana

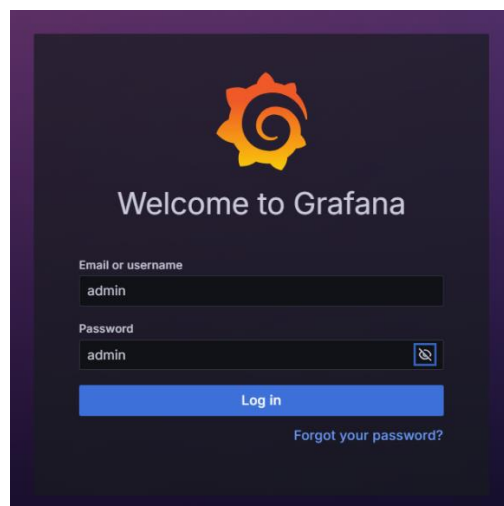
Accedemos a Grafana a través del navegador ingresando la siguiente dirección:

http://<ip-instancia-ec2>:3000



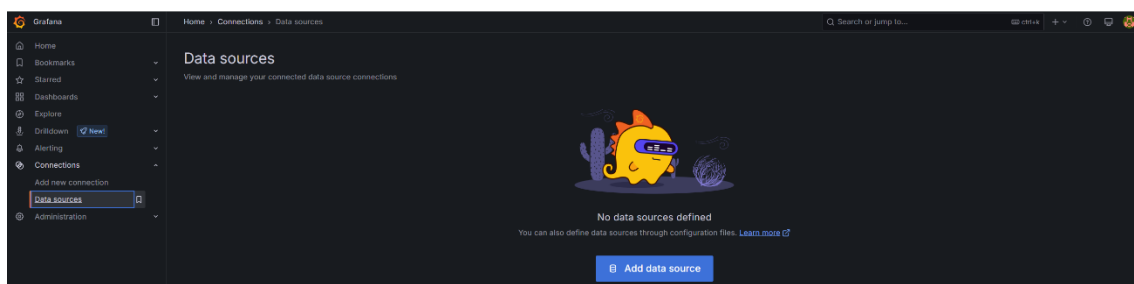
Para iniciar sesión deberemos hacerlo con las siguientes credenciales:

- **Email or username:** admin.
- **Password:** admin.

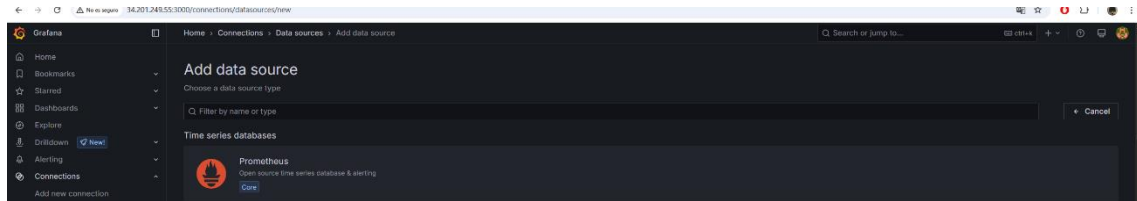


Una vez dentro de **Grafana**, deberemos agregar **Prometheus** como fuente de datos para que Grafana pueda comenzar a visualizar las métricas que Prometheus ha recolectado. Para hacerlo, seguimos estos pasos:

1. En el menú izquierdo deberemos desplegar el apartado “Connections” y entrar en “Data sources”.

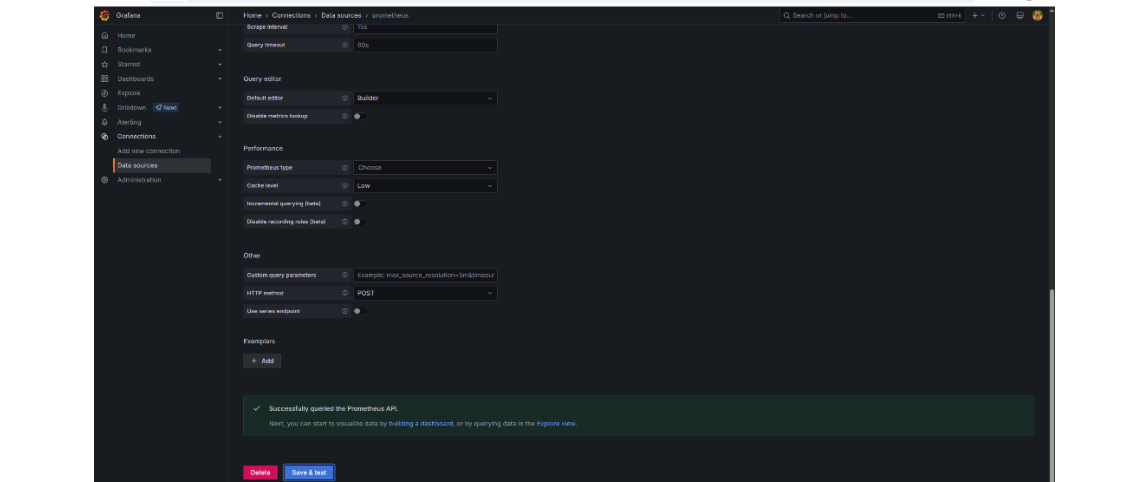
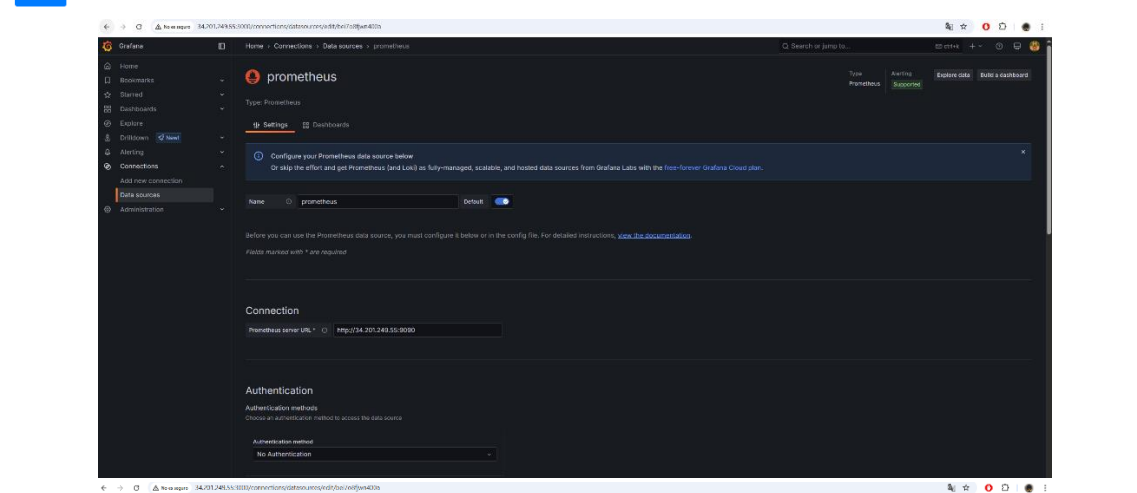
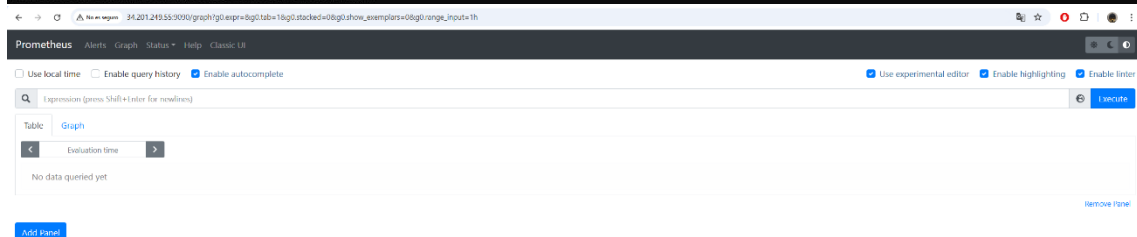
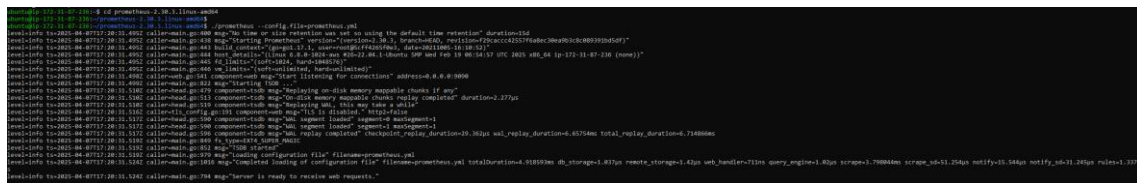


2. Hacemos clic en “Add Data Source” y seleccionamos **Prometheus**.



3. Con el servidor de Prometheus en funcionamiento, procedemos a asignarle un nombre y configuramos la siguiente URL como fuente de datos en **Grafana**:

http://< ip-instancia-ec2>:9090

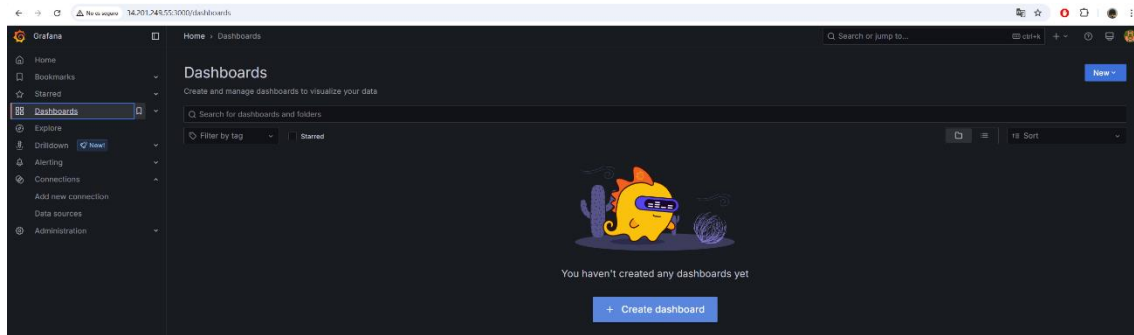


4. Visualización de métricas en Grafana

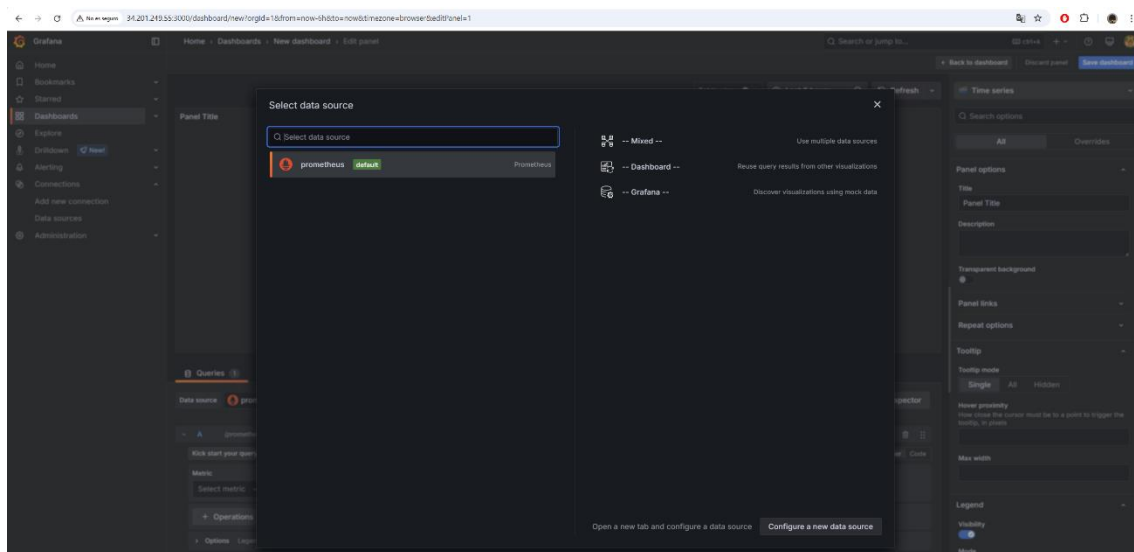
4.1. Crear un dashboard en Grafana

En este apartado, crearemos un dashboard en **Grafana** para visualizar las métricas del clúster. Los pasos son los siguientes:

1. En el menú izquierdo debemos entrar en “Dashboards” y hacer clic en “Create dashboard”.



2. Seleccionamos el data source que hemos creado previamente.

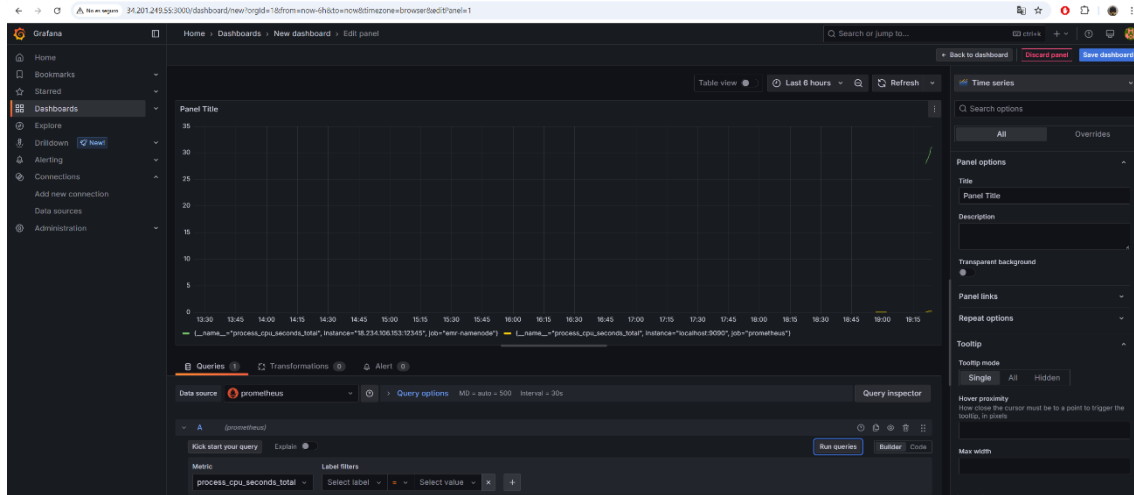


Ya tendríamos el dashboard creado, pero ahora falta indicar las métricas específicas que queremos analizar en los paneles.

CPU

Proporciona el número total de segundos que el proceso ha utilizado la CPU.

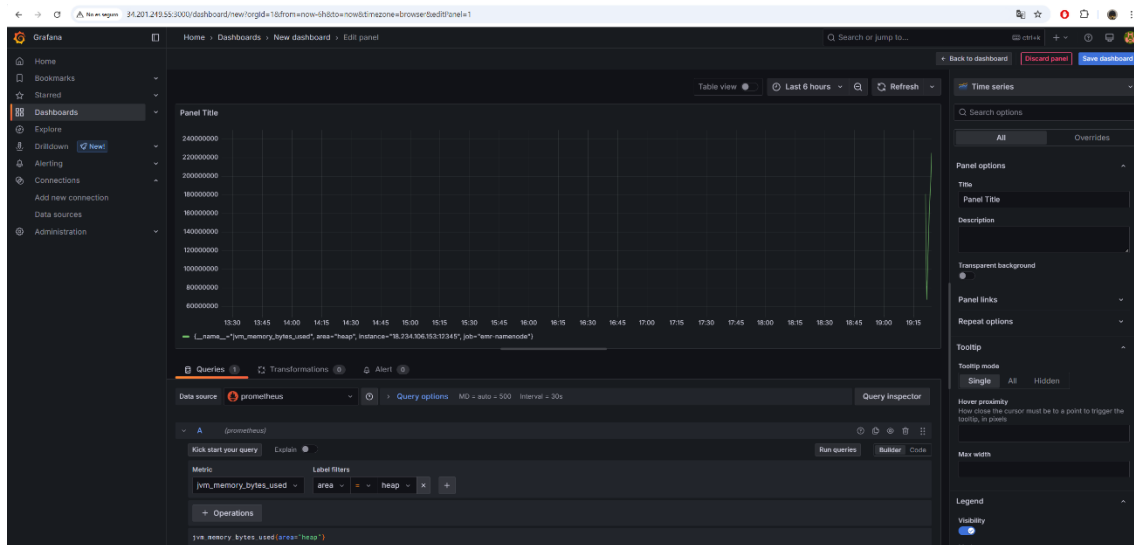
process_cpu_seconds_total



RAM

Muestra la cantidad de memoria utilizada por la JVM en el área de heap.

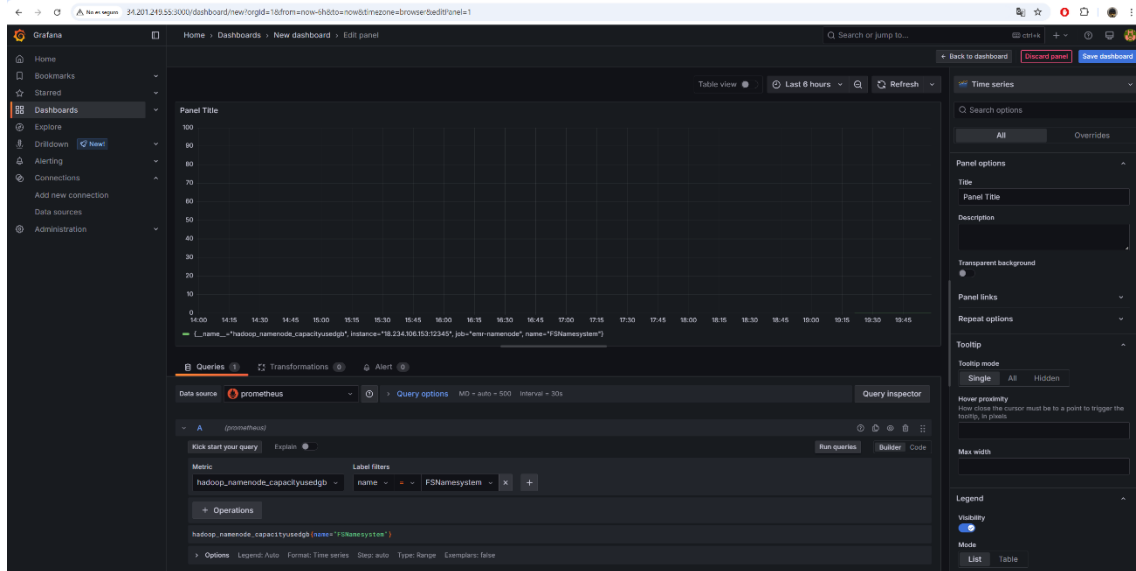
jvm_memory_bytes_used{area="heap"}



Espacio utilizado en HDFS

Proporciona el espacio utilizado en HDFS, en unidades de GB.

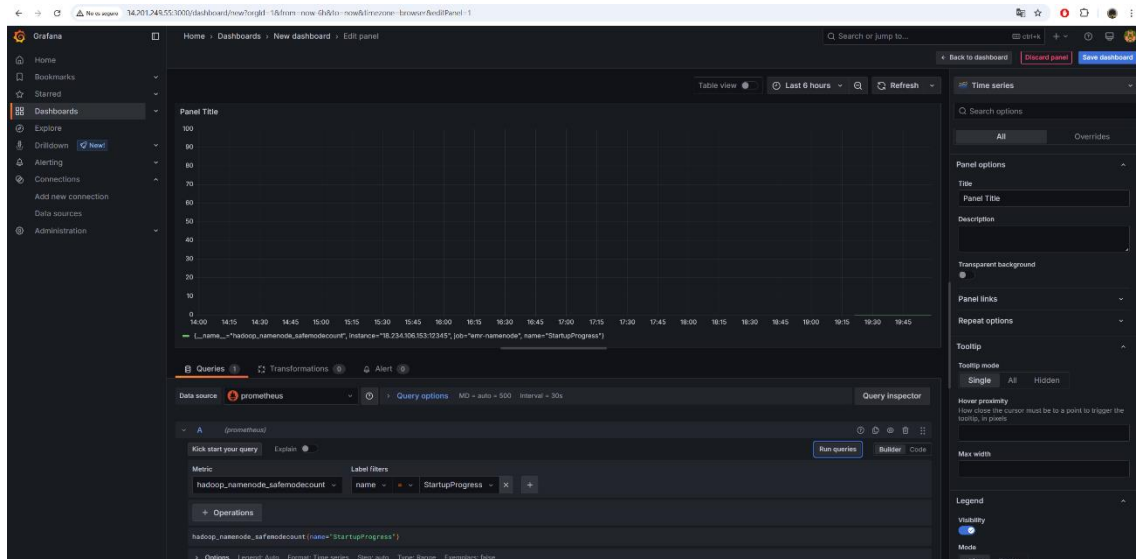
`hadoop_namenode_capacityusedgb{name="FSNamesystem"},}`



Estado del NameNode

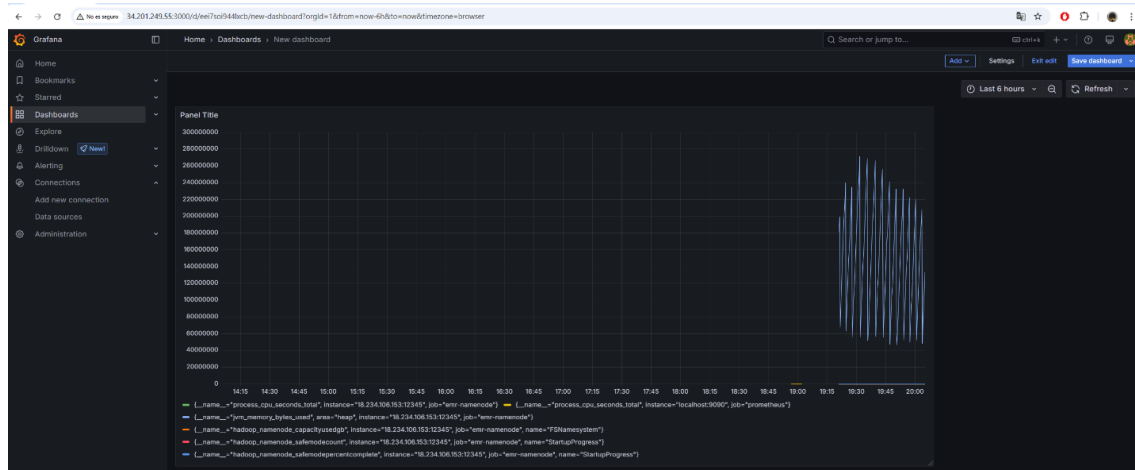
Muestra si el Namenode está en modo seguro. Si el valor es 0, significa que no está en modo seguro.

`hadoop_namenode_safemodecount{name="StartupProgress"},}`



4.2. Explorar métricas

Por último, guardamos el dashboard y observamos las gráficas generadas con el conjunto de métricas configuradas.



Reflexión

1. Métricas más importantes para monitorear

Las métricas que considero más importantes para monitorear en un clúster EMR son:

- **Uso de CPU y memoria:** porque detectan sobrecarga en los nodos y ayudan a escalar recursos.
- **Espacio en HDFS:** controla el almacenamiento y previene fallos por disco lleno.
- **Estado del NameNode y DataNodes:** es vital saber si están activos o en fallo.
- **Tiempos de ejecución de jobs y número de tareas fallidas:** ayudan a evaluar el rendimiento de Spark/Hadoop.

2. Posibles mejoras en la configuración de JMX Exporter

La configuración de JMX Exporter se puede mejorar personalizando el archivo config.yml, aplicando filtros más específicos con la opción pattern para extraer solo las métricas realmente necesarias. También se pueden añadir etiquetas (labels) que ayuden a identificar el origen de cada métrica, así como nombres personalizados (name) para facilitar su interpretación en Grafana.

3. Ventajas de utilizar Prometheus y Grafana

Entre las principales ventajas del uso conjunto de Prometheus y Grafana, destaca la manera en que ambas herramientas se integran y se complementan funcionalmente.

Por un lado, Prometheus permite la recopilación eficiente de métricas mediante su lenguaje de consultas propio, PromQL, y una arquitectura basada en el modelo pull. Este enfoque resulta especialmente adecuado para entornos dinámicos y distribuidos, ya que

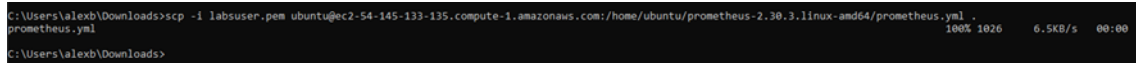
posibilita la extracción directa de información desde los servicios, sin necesidad de que estos la transmitan de forma activa.

Por otro lado, Grafana se orienta a la visualización de dichas métricas, proporcionando dashboards interactivos y opciones avanzadas para la configuración de alertas personalizadas, lo que facilita un monitoreo continuo y favorece una toma de decisiones informada basada en datos en tiempo real.

Descargar archivos

Como extra, si en algún momento necesitamos descargar archivos desde alguna de nuestras instancias EC2 hacia nuestra máquina local, podemos hacerlo utilizando el siguiente comando:

```
scp -i /ruta/a/tu/archivo.pem ubuntu@<ip-de-la-instancia>:/ruta/remota/al/archivo  
/ruta/local/de/destino
```



```
C:\Users\alexb\Downloads>scp -i labsuser.pem ubuntu@ec2-54-145-133-135.compute-1.amazonaws.com:/home/ubuntu/prometheus-2.30.3.linux-amd64/prometheus.yml .  
prometheus.yml 100% 1026 6.5KB/s 00:00  
C:\Users\alexb\Downloads>
```