Amazon Elastic MapReduce

Amazon Elastic MapReduce (Amazon EMR) es un servicio web para la configuración y depliegue de un cluster basado en instancias de máquinas en el servicio Amazon Elastic Compute Cloud (Amazon EC2) y que es gestionado mediante Hadoop. También se puede ejecutar en Amazon EMR otros marcos de trabajo distribuídos como Spark, e interactuar con los datos en otros almacenes de datos como Amazon S3.

Creación de un cluster con EMR

Un cluster EMR suele tener un ciclo de vida totalmente automatizado y que se establece en el momento de su creación. El proceso general sería:

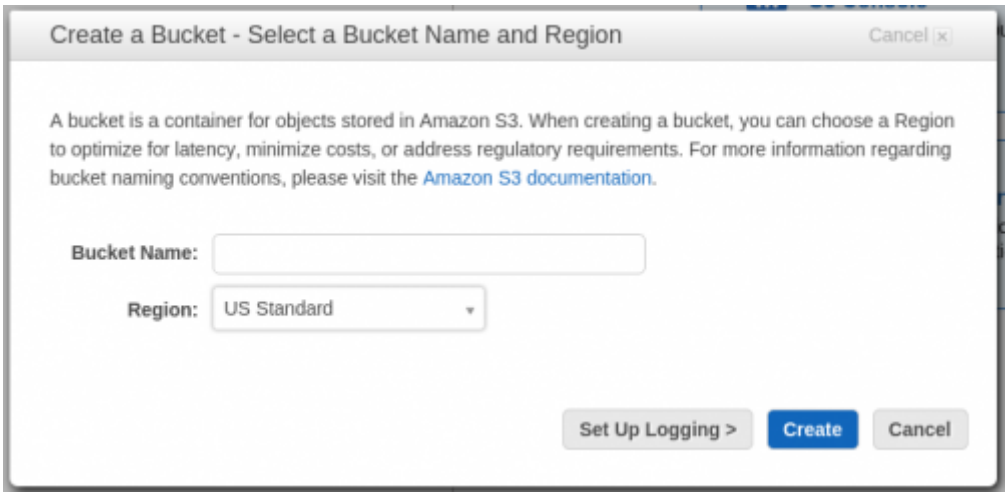
- Lanzamiento de las instancias EC2 de las que se compone el cluster
- Ejecución de los scripts de instalación, tanto automáticos de amazon (como las imagenes preconfiguradas AMI) como los añadidos por el usuario en las acciones de inicialización (Bootstrap actions).
- Trabajos a realizar (Steps) normalmente consistentes en carga de datos de entrada, procesamiento de los mismos, y almacenado de los resultados.
- Apagado automático del cluster una vez se han terminado todos los steps.

En las siguientes subsecciones se explican todos lo básico para poder lanzar un cluster EMR y analizar los resultados de las ejecuciones.

Almacenamento con S3

Amazon EMR puede hacer uso de Amazon S3 como almacenamiento de los datos de entrada, los ficheros de log y los datos de salida. Para más información sobre este tipo de sistema de almacenamiento visita la wiki de amazon.

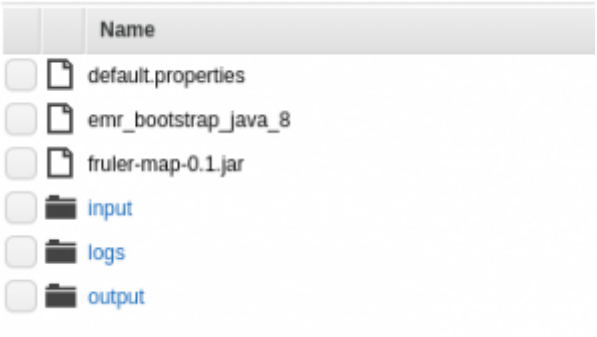
Para crear un nuevo contenedor de datos S3 (bucket), solamente es necesario entrar en el servicio S3 y pulsar “Create Bucket” rellenando el nombre del nuevo contenedor y la región donde estará el mismo (es importante que esta sea la misma región utilizada para desplegar el cluster EMR).



Una vez creado el contenedor, suele ser una buena práctica organizarlo de la siguiente manera:

- Crear una carpeta Log donde guardar los logs de los despliegues de máquinas EC2, así como de las ejecuciones de los diferentes trabajos.
- Crear una carpeta `input` para tener almacenados todos los datos de entrada.
- Crear una carpeta `output` que servirá para guardar los resultados de las ejecuciones.

Además, será necesario tener en este contenedor todo lo necesario para el trabajo que se vaya a ejecutar en el cluster, así como los diferentes scripts de configuración (tal como se comenta en la siguiente sección).



Configuración do cluster

Logs

Spark sobre EMR

Instalar Spark

```
s3://support.elasticmapreduce/spark/install-spark
```

Executar un traballo

- **Step type**: Custom JAR
- **JAR Location**:

```
s3://<CLUSTER_REGION>.elasticmapreduce/libs/script-runner/script-runner.jar
```

- **Arguments**:

```
/home/hadoop/spark/bin/spark-submit --deploy-mode cluster --master yarn-cluster --class <MAIN_CLASS> s3://<BUCKET>/<FILE_JAR> <JAR_OPTIONS>
```

Java 8 en EMR

```
# Check java version
JAVA_VER=$(java -version 2>&1 | sed 's/java version
"\(.*\)\.\(.*\)\..*"/\1\2/; 1q')

if [ "$JAVA_VER" -lt 18 ]
then
    # Download jdk 8
    echo "Downloading and installing jdk 8"
    wget --no-cookies --no-check-certificate --header "Cookie:
gpw_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-
cookie"
"http://download.oracle.com/otn-pub/java/jdk/8-b132/jdk-8-linux-x64.rpm"

    # Silent install
    sudo yum -y install jdk-8-linux-x64.rpm

    # Figure out how many versions of Java we currently have
    NR_OF_OPTIONS=$(echo 0 | alternatives --config java 2>/dev/null | grep
'There ' | awk '{print $3}' | tail -1)

    echo "Found $NR_OF_OPTIONS existing versions of java. Adding new
version."

    # Make the new java version available via /etc/alternatives
    sudo alternatives --install /usr/bin/java java
/usr/java/default/bin/java 1

    # Make java 8 the default
    echo $(($NR_OF_OPTIONS + 1)) | sudo alternatives --config java

    # Set some variables
    export JAVA_HOME=/usr/java/default/bin/java
    export JRE_HOME=/usr/java/default/jre
    export PATH=$PATH:/usr/java/default/bin
fi

# Check java version again
JAVA_VER=$(java -version 2>&1 | sed 's/java version
"\(.*\)\.\(.*\)\..*"/\1\2/; 1q')

echo "Java version is $JAVA_VER!"
echo "JAVA_HOME: $JAVA_HOME"
echo "JRE_HOME: $JRE_HOME"
echo "PATH: $PATH"
```