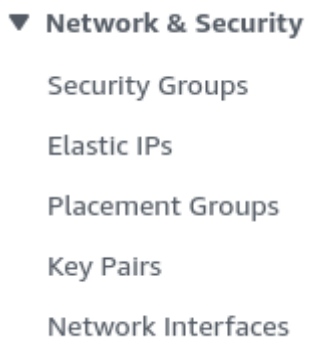
Creación de un cluster hadoop en AWS

Introducción

Con este documento se busca guiar al estudiante para poder crear su propio cluster de hadoop con todas las herramientas como hive, hue y jupyterhub y sqoop utilizando las herramientas que nos brinda el servicio de AWS. Utilizaremos diferentes servicio para la persistencia como lo son S3 y utilizaremos hdfs.

Creación de llaves para el master

Buscamos el servicio EC2 en la consola de AWS y vamos para la sección de Key Pairs.



Le damos en Create Key pair

Donde dice Name ponemos el nombre que queremos utilizar como llave para conectarnos al master del cluster de Hadoop. En nuestro caso lo llamaremos hadoop-cluster. Dejamos por defecto RSA y .pem.

Le damos en create key pair y se descarga la llave.
Una vez descargada corremos el siguiente comando para darle permisos y poder entrar al cluster.

$chmod 400 hadoop-cluster.pem

Creación del bucket S3

Buscamos el servicio S3 en la consola de AWS y le damos en buckets.

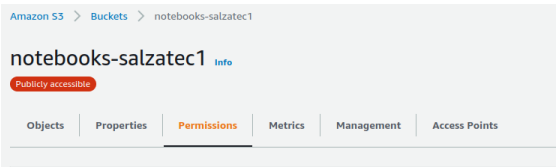
Una vez dentro le damos en Create bucket.

Para el nombre del bucket utilizaremos notebooks-salzatec1. Este nombre debe ser único entre todos los buckets en AWS por lo que recomendamos utilizar nombres que sean únicos a tu aplicación.

Le damos en Create Bucket.

Volvemos el bucket S3 público

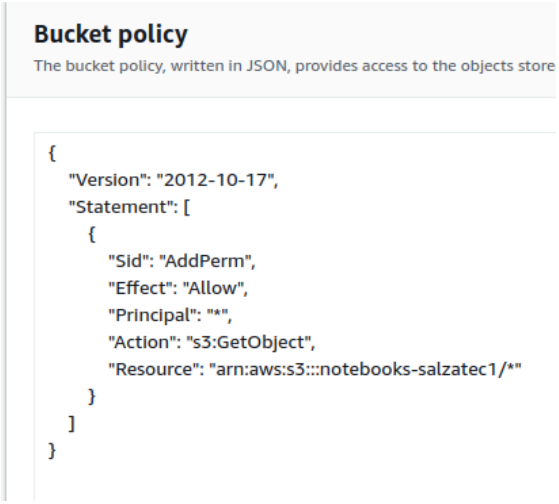
Una vez creado vamos a la consola de S3 en AWS y seleccionamos el bucket luego seleccionamos Permissions.



Donde dice Block public access (bucket settings) le damos en edit.

Deseleccionamos la opción Block all public access y le damos en Save changes.

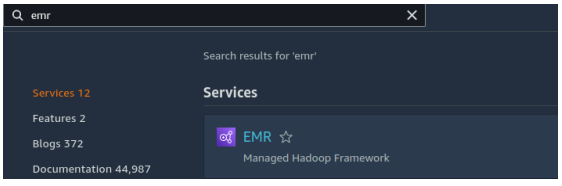
Bajamos y en la seccion Bucket policy le damos edit y pegamos la siguiente política para que todos puedan leer del bucket.



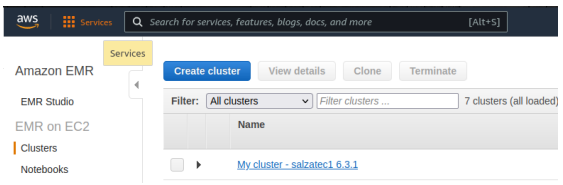
El resource se debe cambiar por el arn de aws correspondiente a bucket que deseas darle permisos.

Creación del cluster

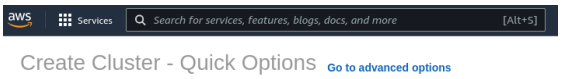
Comenzamos abriendo la cuenta de AWS y buscamos por el servicio de EMR (Managed Hadoop Framework).



Una vez abierta la consola del servicio de EMR le damos en create cluster

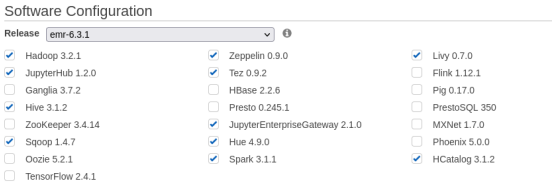


Luego abrimos las opciones avanzadas



Configuración de software

En la configuración de software elegimos la versión emr-6.3.1 que es con la que estaremos trabajando. Los softwares necesarios para correr el cluster y hacer las operaciones para el próximo laboratorio son Hadoop, JupyterHub, Hive, Sqoop, Zeppelin, Tez, JupyterEnterpriseGateway, Hue, Spark, Livy y HCatalog.



En AWS Glue Data Catalog Setting seleccionamos las opciones:

- Use for Hive table metadata
- Use for Spark table metadata

Ahora en la sección de Edit Software Settings seleccionamos enter configuration y pegamos la siguiente política para que podamos conectarnos al bucket de s3 via cluster con jupyter.

[

```
{"classification":"jupyter-s3-conf",
"properties":
{
"s3.persistence.bucket":"notebooks-salzatec1","s3.persistence.enabled":"true"}
}
]
```

Donde dice notebooks-salzatec1 se reemplaza por el nombre del bucket s3 que creaste para la persistencia del cluster.

Le damos en Next.

Hardware

Donde dice Cluster Nodes and Instances podemos ver que los tipos de nodo Master y Core son instancias tipo m5.xlarge, estas instancias no son permitidas por la cuenta academica de AWS por lo que lo que se editaran por instancias m4.xlarge. Para esto clickea en el lápiz al lado del nombre m5.xlarge y selecciona m4.xlarge en la lista desplegada.

Ahora en la sección de Auto-termination seleccionamos Enable auto-termination para que el cluster no siga consumiendo créditos de AWS una vez esté en desuso.
Para esto cambiamos los parámetros y ponemos Terminate cluster when it is idle after 1 hours 0 minutes (En el primer parámetro ponemos 1 y en el segundo 0).

En la sección de WBS Root Volume ponemos que sean 20 gigas de almacenamiento para el cluster

Le damos en Next.

Configuraciones generales del cluster

Cambiamos de cluster name a My cluster - salzatec1 6.3.1. Puedes poner el nombre que desees pero es recomendable que agregues la version de EMR que estas utilizando al final del nombre.

Le damos en Next.

Seguridad

Donde dice EC2 Key pair seleccionamos la key que creamos anteriormente para poder entrar al master del cluster.
En nuestro caso seleccionamos la key hadoop-cluster.

Le damos en Create cluster.

El cluster tardará aproximadamente 20 minutos en crearse.

Cambio de security groups

Como paso final para poder utilizar en cluster en forma sera necesario cambiar los security groups para

abrir los puertos necesarios y las aplicaciones puedan correr en forma.

Buscamos en security groups por el nombre ElasticMapReduce-Master y lo abrimos.

Seleccionamos Edit inbound rules.

Agregamos las siguientes reglas para que sus respectivas aplicaciones puedan funcionar bien y podamos acceder a ellas a través de su interfaz gráfica por el navegador:

SSH
Type: Custom TCP
Port range: 22
Source: Custom (My ip)

HUE
Type: Custom TCP
Port range: 8888
Source: Custom (My ip)

JupyterHub
Type: Custom TCP
Port range: 9443
Source: Custom (My ip)

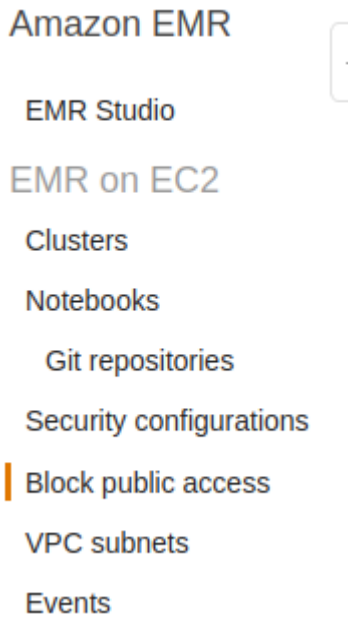
Zeppelin
Type: Custom TCP
Port range: 8890
Source: Custom (My ip)

Le damos en Save rules.

Nota: Es importante que no pongas all ipv4/6 en el source de las reglas porque cuando intentes clonar un cluster con este security groups no te dejará crearlo por tener reglas públicas.

Bloque de acceso público

Uno vez nuestro cluster esta creado buscamos la consola EMR de AWS y le damos en Block public access.



Aqui dejamos Block public access y le en damos en Edit. Agregamos los siguientes port ranges:

Le damos en Save changes.

Ahora podremos acceder por la web a las Hue, JupyterHub y Zeppelin para poder manejar el cluster.

Con esto finaliza la creación de un cluster hadoop en AWS.