**Documentación general del entorno completo de los contenedores del nodo V1**

**Introducción**

Este documento supone que el sistema ya tiene correctamente instalado y configurado docker (aunque incluye las instrucciones para instalarlo), además de tener las 2 carpetas con los archivos necesarios (la del laboratorio y la del ecosistema hadoop).

Los 9 contenedores que se crearán son:

* Laboratorio
  + JupyterHub
* Ecosistema Hadoop
  + Namenode
  + Datanode1
  + Datanode2
  + Datanode3
  + hiveServer
* Imágenes externas
  + MYSQL
    - Metastore
    - Mysql
  + gethue/hue
    - hue

Los cuales provienen de 2 imágenes principales que se tendrán que construir, una para el laboratorio y otra para el ecosistema Hadoop. Las otras imágenes utilizadas son públicas y se podrán descargar de DockerHub (este proceso se realiza automáticamente por docker-compose)

**Instalación de docker:**

yum update -y

yum install -y yum-utils

yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

yum install docker-ce

yum install docker-ce –allowerasing

systemctl enable --now docker

**Copiar archivos**

scp -P PUERTO -i "LLAVE.pem" -r \ruta\a\EntornoCompletoV1\\* user@10.51.225.75:ruta/destino/

**Construir imágenes**

Posicionarse en la carpeta donde están los archivos de la imagen del laboratorio y ejecutar:

docker build –t nombre\_imagen\_laboratorio .

Donde se deberá sustituir **nombre\_imagen\_laboratorio** por el nombre que desees ponerle a la imagen del laboratorio.

Lo mismo aplica para la imagen de hadoop

Posicionarse en la carpeta donde están los archivos de la imagen de hadoop y ejecutar:

docker build –t nombre\_imagen\_hadoop .

Donde se deberá sustituir **nombre\_imagen\_hadoop** por el nombre que desees ponerle a la imagen de hadoop.

**Ejecutar docker-compose**

Asegurarse que docker este corriendo en modo swarm, esto con el comando

docker swarm init

Posicionarte en donde esta el docker-compose.yml, hacer las modificaciones que desees (puertos, nombres, etc) y arrancar los servicios.

docker stack deploy –c docker-compose.yml nombre\_del\_sistema

En donde **nombre\_del\_sistema** será reemplazado por el nombre que desees que tengan el conjunto de servicios que estás por desplegar

**\*\*\* NOTA IMPORTANTE. Si ya has construido este ecosistema y estás apuntando a volúmenes en el host del anterior ecosistema NO realices los siguientes pasos. Únicamente los realizarás cuando es la primera vez que ejecutarás este sistema y cuando no estés apuntando a volumen compartido con el host que ya tenga los datos necesarios para arrancar.**

**Formateo de namenode**

Accede al namenode para ejecutar comandos en él. Primero localiza cual es el CONTAINER ID o CONTAINER NAME de tu contenedor namenode:

docker ps

Una vez localizado:

docker exec –it CONTAINER\_NAME bash

Dentro del contenedor cambia al usuario de hadoop con la sentencia **su hadoop** y posteriormente ejecuta:

stop-all.sh

schematool -dbType mysql -initSchema

hadoop namenode -format

start-dfs.sh

start-yarn.sh

hdfs dfs -mkdir -p /user/hive/warehouse

hdfs dfs -mkdir -p /tmp

hdfs dfsadmin -safemode leave

**Reinicia el contenedor de hue y hiveServer**

Es probable que la primera vez que crees el ecosistema hadoop, hue tenga un error en la intefaz web. Esto es porque no encuentra tablas en la base de datos que necesita. Lo mismo ocurre con el servidor de hive Esto se soluciona fácilmente reiniciando el contenedor de hue y hive.

Localiza cual es el CONTAINER ID o CONTAINER NAME de tu contenedor de hue y hive mediante

docker ps

Una vez localizado:

docker restart CONTAINER\_NAME

en donde reemplazarás CONTAINER\_NAME por el nombre del contenedor de hue y luego por el de hive.

**Anexos**

**Dockerfile (laboratorio)**

Imagen basada en Ubuntu con:

* JupyterHub
  + Certificado HTTPS autofirmado
  + Modificaciones visuales a la interfaz web
* JupyterLab
  + Python
    - Python-Hive
    - Python spark a través de livy
    - Pandas, numpy, scikit-learn
  + R
    - Dplyr, tidyverse, RJDBC, devtools
  + Julia
  + Monitoreo de procesos (medidor de cpu y memoria)
* Limitador de recursos por usuario (RAM y CPU)

**Dockerfile (hadoop)**

Imagen basada en centos 7 con:

* SSH y utilerías relacionadas
* Python3
* Java8
* Hadoop
  + Hive
  + Spark
  + Livy

A pesar de que esta es una imagen base, cada contenedor juega un papel y toma un rol diferente:

* Namenode: Aquí residen los servicios de spark y livy. Es el coordinador del entorno hadoop
* Datanodes: Los workers, aquí es donde se almacenan los datos del hdfs y se realizan los procesos mapreduce.
* HiveServer: Servidor de hive y hive2.