

## Tarea 2, Tema 8 (Dockers)

Student: Tiago de Souza Oliveira

Pasos en la instalación:

1. Descarga de [aquí](#) un fichero comprimido con todo lo necesario para iniciar el cluster
2. Descomprime el fichero y examina su contenido:
  1. Para cada directorio examina el Dockerfile hasta estar seguro de qué es lo que hace. ¿Para qué sirve la imagen hpcnube-base-image?

The purpose of the image hpcnube-base-image is to serve as blueprint for the subsequent images. It saves work by not repeating the same resources for other images, which should run at the same software virtual machine platform, Operational System and environment parameters.

**Base:** This Dockerfile aims to create a container based on a ubuntu:latest version image for having a Hadoop service up-running.

**DataNode-NodeManager:** This Dockerfile aims to create a container to expose services for the Yarn manager and worker nodes for the hadoop map-reduce cluster. Having also defined the HDFS repositories and mounting for the worker nodes being able to access.

**NameNode:** Configure and Instantiate hadoop demons or worker nodes to be added to the cluster.

**ResourceManager:** Configure and Instantiate the resource manager YARN.

2. Comprueba qué comando va a ejecutar cada contenedor y examina los scripts que lanzan ese comando. ¿Por qué crees que esos scripts ejecutan un lazo infinito?

From the container management perspective, the option –detach runs the container in the background, and the terminal is not attached to the container's console.

From the container inside execution, the shell scripts to put services up-running has the following loop instruction, which means that a process keep running continuously and at each 10 seconds verifies if the service/daemon has failed. If so, then log in the terminal and terminate itself.

```
while true
do
    sleep 10
    if ! ps aux | grep ${DAEMON} | grep -q -v grep
    then
        echo "El demonio ${DAEMON} ha fallado"
        exit 1
    fi
done|
```

3. Sigue los pasos en el fichero LEEME para construir las imágenes y levantar los contenedores
  1. Si todo va bien deberías tener levantados 7 contenedores
  2. Ejecuta los pasos 5, 6 y 7 del fichero LEEME y comprueba que todo funciona correctamente en el cluster

## Steps:

1. Constituir una imagen base con el software necesario. Le añadimos la etiqueta latest

```
docker image build -t hpcnube-base-image:1.0.0 ./Base && docker image tag hpcnube-base-image:1.0.0 hpcnube-base-image:latest
```

## MacOS

## 2. Construir imágenes para los diferentes servicios

2.a Construir imagen para el NameNode y ponle la etiqueta latest

```
docker image build -t hpcnube-namenode-image:1.0.0 ./NameNode && docker image tag hpcnube-namenode-image:1.0.0 hpcnube-namenode-image:latest
```

MacOS

## 2 h Construir la imagen para el ResourceManager

```
docker image build -t hpcnube-resourcemanager-image:1.0.0 ./ResourceManager &&
docker image tag hpcnube-resourcemanager-image:1.0.0
hpcnube-resourcemanager-image:latest
```

MacOS					
MacOS					

	hpcube-namenode-image	latest	aaa571804c30	2 minutes ago	4.59GB
MacOS	hpcube-base-image	1.0.0	5cbc9026c59b	6 minutes ago	3.21GB
	hpcube-base-image	latest	5cbc9026c59b	6 minutes ago	3.21GB

```
[base] -> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hpcloud/hpcube-remotemanager	1.0.0	6fd9e020d113	19 hours ago	4.63GB
hpcloud/hpcube-remotemanager-image	latest	6fd9e020d113	19 hours ago	4.63GB
hpcloud/hpcube-nanomode	1.0.0	58fbca82d265	19 hours ago	4.63GB
hpcloud/hpcube-nanomode-image	latest	58fbca82d265	19 hours ago	4.63GB
rocky/8-php	latest	4256803d69b9	30 hours ago	3.66GB
furu	latest	ff7fd8fc512f	5 months ago	1.29GB
opensearchproject/opensearch	latest	efc9b65b275f	12 months ago	5.12GB
		fdad1a7bd24	13 months ago	990MB

## 2.c Construir la imagen para los DataNodes/Nodemangers

```
docker image build -t hpcnube-dnnm-image:1.0.0 ./DataNode-NodeManager && docker image tag hpcnube-dnnm-image:1.0.0 hpcnube-dnnm-image:latest
```

MacOS

# Ubuntu

2.d Ejecuta "docker image ls" para comprobar que se han creado las 4 imágenes

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hpcnube-dnmm-image	1.0.0	6e7e68712841	3 minutes ago	4.59GB
hpcnube-dnmm-image	latest	6e7e68712841	3 minutes ago	4.59GB
hpcnube-resourcemanager-image	1.0.0	f3ea38a9eff0	5 minutes ago	4.59GB
hpcnube-resourcemanager-image	latest	f3ea38a9eff0	5 minutes ago	4.59GB
hpcnube-namenode-image	1.0.0	aaa571804c30	7 minutes ago	4.59GB
hpcnube-namenode-image	latest	aaa571804c30	7 minutes ago	4.59GB
hpcnube-base-image	1.0.0	5cbc9026c59b	11 minutes ago	3.21GB
hpcnube-base-image	latest	5cbc9026c59b	11 minutes ago	3.21GB
myphpserverimage	latest	29e4b5ef1c2d	6 days ago	302MB
myphpserver	latest	82bf238e6d81	6 days ago	244MB
silex/emacs	latest	b7df30055d9e	9 days ago	377MB

MacOS

```
(base) tigao@viveira@whiterock:~/Downloads/HPCHube_cluster$ docker image ls
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
hpcnbase            -image   1.0          8d9f7ec4ab7c  19 minutes ago  4.63GB
hpcnbase            -image   latest       Babf7ec4ab7c  19 minutes ago  4.63GB
hpcnbase-ressourcenamanager-image 1.0.0    6fd09e20dd13  19 hours ago   4.63GB
hpcnbase-ressourcenamanager-image  latest     6fd09e20dd13  19 hours ago   4.63GB
hpcnbase-namenode-image 1.0.0    58fbca82d265  20 hours ago   4.63GB
hpcnbase-namenode-image  latest     58fbca82d265  20 hours ago   4.63GB
hpcnbase-namenode-image 1.0      4256903d99b8  31 hours ago   3.26GB
hpcnbase-base-image  latest     f7fdbfc512fe  5 months ago   1.29GB
rocky-8-mhpc        latest     ef9c9b5b2f7f  12 months ago  5.12GB
ffurf               latest     f0dad1a7bd24  13 months ago  990MB
opensearchproject/opensearch  latest     ...
```

3. Definir la red a la que se conectarán los contenedores  
`docker network create hpcnube-net`

```
tiagoooliveira@whiterock HPCNube_cluster % docker network create hpcnube-net  
8d78e247393378edcef85f2a26da51cf98e10fdf3d0944d575f4d71c21d1c34f8e
```

3.a Para obtener información sobre la red:

```

(tiagooliveira@whiterock HPCNube_cluster) docker network create hpcnube-net
8d78e2a73379adee802e2ed63cf99a5bfdf0d9944d7f7fd212d5c4f8e
(tiagooliveira@whiterock HPCNube_cluster) docker network inspect hpcnube-net
[{"Name": "hpcnube-net", "Id": "8d78e2a73379adee802e2ed63cf99a5bfdf0d9944d7f7fd212d5c4f8e", "Created": "2023-12-22T17:34:49.493442Z", "Scope": "local", "Driver": "bridge", "EnableIPv6": false, "LinkLocalIPv6PrefixLen": 0, "IPAM": {"Driver": "default", "Options": {}, "Config": [{"Subnet": "172.21.0.0/16", "Gateway": "172.21.0.1"}]}, "Internal": false, "Attachable": false, "Ingress": false, "ConfigFrom": {}, "Network": ""}, {"Name": "hpcnube-net", "Id": "dcf159dd16d4d9c1e08aad290f86b62ba3d2dc74c63616ea62918c7f7d75", "Created": "2023-12-24T20:17:06.066490268+01:00", "Scope": "local", "Driver": "bridge", "EnableIPv6": false, "LinkLocalIPv6PrefixLen": 0, "IPAM": {"Driver": "default", "Options": {}, "Config": [{"Subnet": "172.21.0.0/16", "Gateway": "172.21.0.1"}]}, "Internal": false, "Attachable": false, "Ingress": false, "ConfigFrom": {}, "Network": ""}
]

```

MacOS

```

(base) tiagooliveira@whiterock:~/Downloads/HPCNube_cluster$ docker network create hpcnube-net
dcf159dd16d4d9c1e08aad290f86b62ba3d2dc74c63616ea62918c7f7d75
(base) tiagooliveira@whiterock:~/Downloads/HPCNube_cluster$ docker network inspect hpcnube-net
[
  {
    "Name": "hpcnube-net",
    "Id": "dcf159dd16d4d9c1e08aad290f86b62ba3d2dc74c63616ea62918c7f7d75",
    "Created": "2023-12-24T20:17:06.066490268+01:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.20.0.0/16",
          "Gateway": "172.20.0.1"
        }
      ],
      "Internal": false,
      "Attachable": false,
      "Ingress": false,
      "ConfigFrom": {},
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {},
    "Labels": {}
  }
]

```

Ubuntu

#### 4. Iniciar los contenedores en esa red

##### 4.a Un contenedor para el servicio NameNode

```

docker container run --init --detach --name hpcnube-namenode --network=hpcnube-net
--hostname hpcnube-namenode -p 9870:9870 hpcnube-namenode-image

```

```

(tiagooliveira@whiterock HPCNube_Cluster) docker container run --init --detach --name hpcnube-namenode --network=hpcnube-net
@ hpcnube-namenode-image
966effde4811071b25e76bc23dbf710382a18181f77d06da44a8a1e8e98b3
(tiagooliveira@whiterock HPCNube_Cluster) docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
tiagooliveira@whiterock HPCNube_Cluster % docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
hpcnube-dmn-image 1.0.0 6e76e8712841 18 hours ago 4.59GB
hpcnube-dmn-image latest 6e76e8712841 18 hours ago 4.59GB
hpcnube-namenode-image 1.0.0 f3ee38a9ef70 18 hours ago 4.59GB
hpcnube-resourcemanager-image latest f3ee38a9ef70 18 hours ago 4.59GB
hpcnube-namenode-image 1.0.0 aa571884c38 18 hours ago 4.59GB
hpcnube-namenode-image latest aa571884c38 18 hours ago 4.59GB
hpcnube-base-image 1.0.0 86bd982a5e9b 19 hours ago 3.21GB
hpcnube-base-image latest 86bd982a5e9b 19 hours ago 3.21GB
myphaserverimage latest 29e4a5ef1c2d 6 days ago 302MB

```

MacOS

```

(base) tiagooliveira@whiterock:~/Downloads/HPCNube_Cluster$ docker container run --init --detach --name hpcnube-namenode --network=hpcnube-net
--hostname hpcnube-namenode -p 9870:9870 hpcnube-namenode-image
966effde4811071b25e76bc23dbf710382a18181f77d06da44a8a1e8e98b3
(base) tiagooliveira@whiterock:~/Downloads/HPCNube_Cluster$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
9e6effde4811 hpcnube-namenode-image "/opt/bd/start-daemo..." 30 seconds ago Up 22 seconds 8020/tcp, 9820/tcp, 9871/tcp, 0.0.0.
0:9870->9870/tcp, :::9870->9870/tcp hpcnube-namenode
(base) tiagooliveira@whiterock:~/Downloads/HPCNube_Cluster$ 

```

Ubuntu

##### 4.b Un contenedor para el servicio ResourceManager

```

docker container run --init --detach --name hpcnube-resourcemanager
--network=hpcnube-net --hostname hpcnube-resourcemanager -p 8088:8088
hpcnube-resourcemanager-image

```

```

(tiagooliveira@whiterock HPCNube_Cluster) docker container run --init --detach --name hpcnube-resourcemanager --network=hpcnube-net --hostname hpcnube-resourcemanager
--port=8088:8088 hpcnube-resourcemanager-image
966effde4811071b25e76bc23dbf710382a18181f77d06da44a8a1e8e98b3
(tiagooliveira@whiterock HPCNube_Cluster) docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
tiagooliveira@whiterock HPCNube_Cluster % docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
hpcnube-dmn-image 1.0.0 6e76e8712841 18 hours ago 4.59GB
hpcnube-dmn-image latest 6e76e8712841 18 hours ago 4.59GB
hpcnube-resourcemanager-image 1.0.0 f3ee38a9ef70 18 hours ago 4.59GB
hpcnube-resourcemanager-image latest f3ee38a9ef70 18 hours ago 4.59GB
hpcnube-namenode-image 1.0.0 aa571884c38 19 hours ago 4.59GB
hpcnube-namenode-image latest aa571884c38 19 hours ago 4.59GB
hpcnube-base-image 1.0.0 86bd982a5e9b 19 hours ago 3.21GB
hpcnube-base-image latest 86bd982a5e9b 19 hours ago 3.21GB
myphaserverimage latest 29e4a5ef1c2d 6 days ago 302MB
myphaser latest 82ff38bed81 6 days ago 244MB

```

MacOS

```

(base) tiagooliveira@whiterock:~/Downloads/HPCNube_Cluster$ docker container run --init --detach --name hpcnube-resourcemanager --network=hpcnube-net
--hostname hpcnube-resourcemanager -p 8088:8088 hpcnube-resourcemanager-image
ab07be34a87e hpcnube-resourcemanager-image "/opt/bd/start-daemo..." 6 seconds ago Up 3 seconds 8030-8033/tcp, 8098/tcp, 8098:8088->8088/tcp, :::8088->8088/tcp
9e6effde4811 hpcnube-namenode-image "/opt/bd/start-daemo..." About a minute ago Up About a minute 8020/tcp, 9820/tcp, 9871/tcp, 0.0.0.0:9870->9870/tcp, :::9870->9870/tcp hpcnube-namenode
(base) tiagooliveira@whiterock:~/Downloads/HPCNube_Cluster$ 

```

Ubuntu

##### 4.c Cuatro contenedores que actúen como DataNodes y NodeManagers

```
for n in {1..4}; do docker container run --init --detach --name hpcnube-dnnm${n} --network=hpcnube-net --hostname hpcnube-dnnm${n} hpcnube-dnnm-image; done
```

MacOS

```
(base) tlaogooliveira@tletterock:~/Downloads/HPCnube_cluster$ for n in [1..4]; do docker container run --init --detach --name hpcnube-dnms${n} --network=hpcnube-net -h hostname hpcnube-dnms${n} hpcnube-dnmm-image; done
ba9ba8810805fc38c291c192449953dca055bd34b4c4e88a97f67498a001
9d1161284ce25e2ze242b71d9a2f21c0861992a8327f9409360ea
ca3d271560b8d4bddc2e93cd8c7e179b34ce1ae49b8df9f1826335
8aa0b77a168b17f2e6f52ed69797934c322463d892695de42e44e108a4
(base) tlaogooliveira@tletterock:~/Downloads/HPCnube_cluster$ docker ps
CONTAINER ID IMAGE COMMAND CREATED NAMES STATUS PORTS
8aa0b77a168 hpcnube-dnmm-image "/opt/db/start-daemo..." 5 seconds ago Up 5 seconds
044/tcp, 8984/tcp, 8984-9867/tcp, 50800-50859/tcp, 50816-50208/tcp hpcnube-dnmm4
ca3d271560b8 hpcnube-dnmm-image "/opt/db/start-daemo..." 8 seconds ago Up 5 seconds
044/tcp, 8984/tcp, 8984-9867/tcp, 50800-50859/tcp, 50816-50208/tcp hpcnube-dnmm3
9d1161284ce2 hpcnube-dnmm-image "/opt/db/start-daemo..." 9 seconds ago Up 7 seconds
044/tcp, 8984/tcp, 8984-9867/tcp, 50800-50859/tcp, 50816-50208/tcp hpcnube-dnmm2
8aa0b77a16805f hpcnube-dnmm-image "/opt/db/start-daemo..." 9 seconds ago Up 8 seconds
044/tcp, 8984/tcp, 8984-9867/tcp, 50800-50859/tcp, 50816-50208/tcp hpcnube-dnmm1
(base) tlaogooliveira@tletterock:~/Downloads/HPCnube_cluster$ ./hpcnube-resourcemanager -image "/opt/db/start-l-daemo..." -about A minute ago
hpcnube-resourcemanager
cp, 0.0...>8088->8088/tcp, ::1:8088->8088/tcp
9efffe0de4811 hpcnube-namenode-image "/opt/db/start-daemo..." 3 minutes ago Up 3 minutes
871/tcp, 0.0.0.0:9870->870/tcp, 0.0.0.0:9870->870/tcp
(base) tlaogooliveira@tletterock:~/Downloads/HPCnube_cluster$
```

# Ubuntu

4.d Ejecuta "docker container ps" para ver que están ejecutándose los 6 contenedores

[base] #	CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
	Raaab077169	hpcubce-dnnm-image	/opt/bd/start-daemo...	About a minute ago	Up About a minute	8048/tcp, 8042/tcp, 8044/tcp, 8048/tcp, 9864-9867/tcp, 50000-50050/tcp, 50100-50200/tcp hpcubce-dnnm
	cad3725608b8	hpcubce-dnnm-image	/opt/bd/start-daemo...	About a minute ago	Up About a minute	8046/tcp, 8042/tcp, 8044/tcp, 8048/tcp, 9864-9867/tcp, 50000-50050/tcp, 50100-50200/tcp hpcubce-dnnm
	5f1009592020	hpcubce-dnnm-image	/opt/bd/start-daemo...	About a minute ago	Up About a minute	8044/tcp, 8042/tcp, 8044/tcp, 8048/tcp, 9864-9867/tcp, 50000-50050/tcp, 50100-50200/tcp hpcubce-dnnm
	5f1000520020	hpcubce-dnnm-image	/opt/bd/start-daemo...	About a minute ago	Up About a minute	8048/tcp, 8042/tcp, 8044/tcp, 8048/tcp, 9864-9867/tcp, 50000-50050/tcp, 50100-50200/tcp hpcubce-dnnm
	ab07ba100805	hpcubce-dnnm-image	/opt/bd/start-daemo...	About a minute ago	Up About a minute	8046/tcp, 8042/tcp, 8044/tcp, 8048/tcp, 9864-9867/tcp, 50000-50050/tcp, 50100-50200/tcp hpcubce-dnnm
	ab07ba3467e	hpcubce-resourcemanager-image	/opt/bd/start-daemo...	2 minutes ago	Up 2 minutes	8030-8033/tcp, 8080/tcp, 0.0.0.0:8088->8088/tcp, :::8088->8088/tcp
	Peedefed4e811	hpcubce-nanodenome-image	/opt/bd/start-daemo...	4 minutes ago	Up 4 minutes	8026/tcp, 9828/tcp, 9871/tcp, 0.0.0.0:9870->9870/tcp, :::9870->9870/tcp

# Ubuntu

## 5. Comprobar el estado de HDFS

5.a Entrar en el contenedor con el NameNode haciendo

```
docker container exec -ti hpcnube-namenode /bin/bash
```

5.b Una vez dentro, comprueba el estado del HDFS con

```
hdfs dfsadmin -report
```

```
(base) tianyuan@tianyuan-OptiPlex-5090:~/Downloads/hpChube_Cluster$ docker container exec -tl hpchube-namenode /bin/bash
hadoop@hpchube-namenode:~$ hadoop fs -ls /
Configured Capacity: 143148190876 (1.38 TB)
Present Capacity: 143148190876 (1.38 TB)
DFS Remaining: 15180536368 (293.47 GB)
DFS Used: 98304 (96 KB)
DFS Used%: 0.00%
Replicated Blocks:
  Under replicated blocks: 0
  Blocks with corrupt replicas: 0
  Missing blocks: 0
  Missing blocks (with replication factor 1): 0
  Low redundancy blocks with highest priority to recover: 0
  Pending deletion blocks: 0
Erasure Coded Block Groups:
  Low redundancy block groups: 0
  Block groups with corrupt internal blocks: 0
  Missing block groups: 0
  Low redundancy blocks with highest priority to recover: 0
  Pending deletion blocks: 0
```

## Ubuntu

## Live datanodes (4):

```
Name: 172.20.0.4:9866 (hpcnube-dnnm1.hpcnube-net)
Hostname: hpcnube-dnnm1
Decommision Status : Normal
Configured Capacity: 357862047744 (333.29 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 260831107560 (242.92 GB)
DFS Remaining: 78777454592 (73.37 GB)
DFS Used%: 0.00%
DFS Remaining%: 22.01%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xcvers: 0
Last contact: Sun Dec 24 19:24:59 GMT 2023
Last Block Report: Sun Dec 24 19:21:56 GMT 2023
Num of Blocks: 0
```

```
Name: 172.20.0.5:9866 (hpcnube-dnnm2.hpcnube-net)
Hostname: hpcnube-dnnm2
Decommision Status : Normal
Configured Capacity: 357862047744 (333.29 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 260831710560 (242.92 GB)
DFS Remaining: 78777454592 (73.37 GB)
DFS UsedK: 0.00%
DFS Remaining%: 22.01%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache UsedK: 100.00%
Cache Remaining%: 0.00%
Xreceivers: 0
Last contact: Sun Dec 24 19:24:59 GMT 2023
Last Block Report: Sun Dec 24 19:21:56 GMT 2023
Number of Blocks: 0
```

```
Name: 172.20.0.6:9866 (hpcnube-dnnm3.hpcnube-net)
Hostname: hpcnube-dnnm3
Decommission Status : Normal
Configured Capacity: 357862047744 (333.29 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 260831170560 (242.92 GB)
DFS Remaining: 78777454592 (73.37 GB)
DFS Used%: 0.00%
DFS Remaining%: 22.01%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xcelsvers: 0
Last contact: Sun Dec 24 19:25:00 GMT 2023
Last Block Report: Sun Dec 24 19:22:00 GMT 2023
Num of Blocks: 0

Name: 172.20.0.7:9866 (hpcnube-dnnm4.hpcnube-net)
Hostname: hpcnube-dnnm4
Decommission Status : Normal
Configured Capacity: 357862047744 (333.29 GB)
DFS Used: 24576 (24 KB)
Non DFS Used: 260831170560 (242.92 GB)
DFS Remaining: 78777454592 (73.37 GB)
DFS Used%: 0.00%
DFS Remaining%: 22.01%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xcelsvers: 0
Last contact: Sun Dec 24 19:24:58 GMT 2023
Last Block Report: Sun Dec 24 19:22:01 GMT 2023
Num of Blocks: 0
```

## 6. Comprobar el estado de YARN

6.a Entrar en el contenedor con el ResourceManager haciendo

```
docker container exec -ti hpcnube-resourcemanager /bin/bash
```

6.b Una vez dentro, comprueba el estado del YARN con estos comandos

```
yarn node -list -showDetails
```

```
[base] hdp@hpcnode-resourcemanager:~$ docker container exec -t hpcnode-resourcemanager /bin/bash
hadoop@hpcnode-resourcemanager:~$ hadoop -list shareddata
[2023-12-24 20:39:01,786 INFO Configuration] Configuration resource 'shareddata' not found
[2023-12-24 20:39:01,786 INFO ResourceUtilization] ResourceUtilization: Unable to find 'resource-types.xml'.
[2023-12-24 20:39:01,786 INFO ResourceUtilization] ResourceUtilization: Unable to find 'shareddata'.
[2023-12-24 20:39:01,786 INFO ResourceUtilization] ResourceUtilization: Unable to find 'shareddata'.
Detailed Node Information:
  Node-Id : StateNode-HTTP-Address Number-of-Running-Containers
  hpcnode-dn01:34129          RUNNING hpcnode-dn01:8842           0
Detailed Node Information:
  Configured Resources : vcores=14293, vcores=<
  Allocated Resources : vcores=14293, vcores=<
  Resource Utilization by Node : Pmem:12800 MB, Wmem:12802 MB, Vcores:0.45318225
  Resource Utilization by Containers : Pmem:0 MB, Wmem:0 MB, Vcores:0.0
  Node-Label(s) : 
hpcnode-dn01:39915          RUNNING hpcnode-dn01:8842           0
Detailed Node Information:
  Configured Resources : vcores=14293, vcores=<
  Allocated Resources : vcores=19, vcores=<
  Resource Utilization by Node : Pmem:12121 MB, Wmem:12121 MB, Vcores:0.47967345
  Resource Utilization by Containers : Pmem:0 MB, Wmem:0 MB, Vcores:0.0
  Node-Label(s) : 
hpcnode-dn01:39917          RUNNING hpcnode-dn01:8842           0
Detailed Node Information:
  Configured Resources : vcores=14293, vcores=<
  Allocated Resources : vcores=19, vcores=<
  Resource Utilization by Node : Pmem:12081 MB, Wmem:12081 MB, Vcores:0.44318563
  Resource Utilization by Containers : Pmem:0 MB, Wmem:0 MB, Vcores:0.0
  Node-Label(s) : 
hpcnode-dn01:41395          RUNNING hpcnode-dn01:8842           0
Detailed Node Information:
  Configured Resources : vcores=14293, vcores=<
  Allocated Resources : vcores=19, vcores=<
  Resource Utilization by Node : Pmem:11990 MB, Wmem:11990 MB, Vcores:0.5368023
  Resource Utilization by Containers : Pmem:0 MB, Wmem:0 MB, Vcores:0.0
  Node-Label(s) : 
```

## Ubuntu

yarn top

```
YARN Top: 20:40:58, Up 0d, 119, 0 active users, queueSize: root  
Nodes: 100 available, 0 active, 0 standby, 0 transitioning, 0 decommissioned, 0 lost, 0 rebooted, 0 shutdown  
Queues: GBalloons: 0 running, 0 submitted, 0 pending, 0 completed, 0 killed, 0 failed  
Queue(s) MemGB(s): 55 available, 0 allocated, 0 pending, 0 reserved  
Queue(s) Vcores: 16 available, 0 allocated, 0 pending, 0 reserved  
Queue(s) Containers: 0 allocated, 0 pending, 0 reserved
```

111

## 7. Comprobar la información proporcionada por los interfaces web de los servicios:

- Namenode: <http://localhost:9870>

- ResourceManager: <http://localhost:8088>

The screenshot shows the Apache Hadoop YARN ResourceManager UI at [localhost:8088/cluster](http://localhost:8088/cluster). The top navigation bar includes links for Getting Started, Installing Scala on Ubuntu, How to Install and Co..., Install Apache Spark o..., Why every software e..., CS 329S | Syllabus - st..., Use spark-redis comp..., Effective Programmin..., Compile and build Sca... .

**All Applications**

**Cluster Metrics**

	Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources	Total Resources
0	0	0	0	0	<memory:0 B, Vcores:0>	<memory:55.48 GB, vCores:16>	

**Cluster Nodes Metrics**

	Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes
4	0	0	0	0

**Scheduler Metrics**

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[memory-mb (unit=M), vcores]	<memory:128, vCores:1>	<memory:2560, vCores:1>

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU Vcores	Allocated Memory MB
No data available in table														

Showing 0 to 0 of 0 entries

## Ubuntu

En Linux, dado el modo bridge de la red, es posible acceder a esos interfaces usando directamente la IP de los contenedores. Para comprobar la IP, ejecuta:

```
docker container inspect nombre_del_contenedor | grep IPAddress
```

# Ubuntu

#### 8. Parar, reiniciar y borrar los contenedores:

#### 8.a Para detener los contenedores sin borrarlos:

`docker container stop nombre_del_contenedor`

## Ubuntu

#### 8.b Para reiniciar un contenedor parado

`docker container start nombre del contenedor`

#### 8.c Para borrar un contenedor parado

`docker container rm nombre del contenedor`

ar y borrar las imágenes

## 9. Listar y borrar las imágenes

## docker images

```
docker image rm nombre de la imagen:1.0.0 nombre de la imagen:latest
```

```
(base) tiagoooliveira@whiterock:~/Downloads/HPCnube_cluster$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
hpcnube-resourcemanager-image   1.0.0   0fd09e2d113  38 hours ago  4.63GB
hpcnube-resourcemanager-image   latest   0fd09e2d113  38 hours ago  4.63GB
rocky-8-rhmc        latest   f77df8c512fe  5 months ago  1.29GB
ffurf               latest   efc3b05b27f7  12 months ago  5.12GB
opensearchproject/opensearch    latest   fd6d1a7bd24  13 months ago  990kB
opensearchproject/opensearch-dashboards  latest   9cae0e7311ce  13 months ago  1.29GB
internal/ffurfray      latest   fbba20e4b84  13 months ago  5.12GB
internal/ffurf         latest   7a048bc1e64f  13 months ago  5.12GB
internal/hdfs-dev      latest   71731d194792  13 months ago  2.59GB
internal/pulsar-dev     latest   fc9ed98cc1c3  13 months ago  2.59GB
gcr.io/kds-minikube/kicbase  v0.0.36  866cf1e4e3f2  14 months ago  1.11GB
ubuntu              v20.04   080e5dfb52c7  14 months ago  72.8MB
rayproject/ray        1.13.0-py3-cpu  ba528ee0e52d  19 months ago  2.8GB
helidon-world        latest   feb5d9fead65  2 years ago  19.3kB
(base) tiagoooliveira@whiterock:~/Downloads/HPCnube_cluster$ docker image rm --force hpcnube-resourcemanager-image:latest
Untagged: hpcnube-resourcemanager-image:1.0.0
Untagged: hpcnube-resourcemanager-image:latest
Deleted: sha256:0fd09e2d11391b20326fb6059bc3b15799cbddfb338124e9e15ca12280b
(base) tiagoooliveira@whiterock:~/Downloads/HPCnube_cluster$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
rocky-8-rhmc        latest   f77df8c512fe  5 months ago  1.29GB
ffurf               latest   efc3b05b27f7  12 months ago  5.12GB
opensearchproject/opensearch    latest   fd6d1a7bd24  13 months ago  990kB
opensearchproject/opensearch-dashboards  latest   9cae0e7311ce  13 months ago  1.29GB
internal/ffurfray      latest   fbba20e4b84  13 months ago  5.12GB
internal/ffurf         latest   7a048bc1e64f  13 months ago  5.12GB
internal/hdfs-dev      latest   71731d194792  13 months ago  2.59GB
internal/pulsar-dev     latest   fc9ed98cc1c3  13 months ago  2.59GB
gcr.io/kds-minikube/kicbase  v0.0.36  866cf1e4e3f2  14 months ago  1.11GB
ubuntu              v20.04   080e5dfb52c7  14 months ago  72.8MB
rayproject/ray        1.13.0-py3-cpu  ba528ee0e52d  19 months ago  2.8GB
helidon-world        latest   feb5d9fead65  2 years ago  19.3kB
(base) tiagoooliveira@whiterock:~/Downloads/HPCnube_cluster$
```

Ubuntu

## 10. Borrar la red

docker network rm hpcnube-net

```
(base) tiagoooliveira@whiterock:~/Downloads/HPCnube_cluster$ docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
6c314ecfce96  bridge    bridge      local
54758d8e0920  docker_opensearch-net  bridge      local
d0c2159abf31  host      host      local
dcf159d1d816  hpcnube-net    bridge      local
7e969eb25406  minikube   bridge      local
f0ec70a4e322  none      null      local
e89c0160cb2d  opendistro_opensearch-net  bridge      local
(base) tiagoooliveira@whiterock:~/Downloads/HPCnube_cluster$ docker network rm hpcnube-net
hpcnube-net
(base) tiagoooliveira@whiterock:~/Downloads/HPCnube_cluster$ docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
6c314ecfce96  bridge    bridge      local
54758d8e0920  docker_opensearch-net  bridge      local
d0c2159abf31  host      host      local
7e969eb25406  minikube   bridge      local
f0ec70a4e322  none      null      local
e89c0160cb2d  opendistro_opensearch-net  bridge      local
(base) tiagoooliveira@whiterock:~/Downloads/HPCnube_cluster$
```

Ubuntu

---

## TAREAS

1. Crea un Dockerfile para un nuevo contenedor denominado FrontEnd, que servira de punto de acceso al cluster. Características:
  1. Debe partir de la misma imagen base, hpcnube-base-image, que el resto de contenedores
  2. Debe instalar un entorno de desarrollo Java, incluyendo openjdk-8-jdk, maven y algún editor (p.e. vim)
  3. Debe ejecutar un servidor de SSH (ese es el único servicio que debe ejecutar)
  4. Debe crear un usuario de nombre luser y UID 2000, que debe poder acceder al contenedor via SSH
    1. El usuario debe tener un directorio HOME con un fichero .bashrc en el que se definan las variables JAVA\_HOME y HADOOP\_HOME. Además, debe añadir a la variable PATH la ruta \${HADOOP\_HOME}/bin.
    2. Para ponerle la contraseña, puedes usar el comando chpasswd

### Ubuntu: Dockerfile for frontend-image

```
# Use the "hpcnube-base-image" as the base image
FROM hpcnube-base-image

# Switch to root user. To ensure it is able to run ssh service
USER root

# Install the required packages (OpenJDK 8, vim, openssh-server)
RUN apt-get update && \
    apt-get install -y openjdk-8-jdk vim openssh-server

# Env variables
ENV HADOOP_VERSION 3.3.6
ENV LOG_TAG "[DNMM Hadoop_${HADOOP_VERSION}]:"
ENV BASE_DIR /opt/bd
ENV HADOOP_HOME ${BASE_DIR}/hadoop/
ENV HADOOP_CONF_DIR ${HADOOP_HOME}/etc/hadoop/
ENV DATA_DIR /var/data/hadoop/hdfs

# Create repositories for data
RUN echo "$LOG_TAG Crea directorios para los datos de HDFS del DataNode" && \
    mkdir -p ${DATA_DIR}/dn && chown -R hdadmin:hadoop ${DATA_DIR}

# Create repositories for log
RUN echo "$LOG_TAG Crea directorio para los ficheros de log" && \
    mkdir ${HADOOP_HOME}/logs

# Copy config files
RUN echo "$LOG_TAG Copia los ficheros de configuracion y el script de inicio"
COPY Config-files/core-site.xml ${HADOOP_CONF_DIR}/core-site.xml
COPY Config-files/hdfs-site.xml ${HADOOP_CONF_DIR}/hdfs-site.xml
COPY Config-files/yarn-site.xml ${HADOOP_CONF_DIR}/yarn-site.xml
COPY Config-files/mapred-site.xml ${HADOOP_CONF_DIR}/mapred-site.xml

# Add a user named "luser" with UID 2000
RUN useradd -u 2000 -m luser
```

```
# Set the password for the "luser" user (change it as needed)
RUN echo 'luser:HPC2024!' | chpasswd

# Create the .bashrc file for the "luser" with Java and Hadoop environment variables
RUN echo 'export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64' >> /home/luser/.bashrc
RUN echo 'export HADOOP_HOME=/path/to/hadoop' >> /home/luser/.bashrc

# Add Hadoop binaries to the PATH
RUN echo 'export PATH=${HADOOP_HOME}/bin:$PATH' >> /home/luser/.bashrc

# Expose SSH port
EXPOSE 22

# Start the SSH server
CMD ["/usr/sbin/sshd", "-D"]
```

## To build the image

```
docker build -t frontend-image .
```

**To run the image** (to start a new container and maps port 2222 on the host to port 22 on the container)

```
docker container run --init --detach --name frontend2 -p 2222:22 frontend-image
```

```
[base] $ curl -v http://localhost:22222/frontend/5 docker container run --init --detach --name frontend-p 2222:22 frontend:Image
docker: Error response from daemon: Conflict: The name "frontend" is already in use by container "dib19e8008eeaaec2bde5f7f2da80fb2c1d1ac8de25e26ed63fa15fb89cb04". You have to remove (or rename) that container to be able to reuse that name.
[base] $ curl -v http://localhost:22222/frontend/5 docker container run --init --detach --name Frontend2-p 2222:22 Frontend:Image
[base] $ curl -v http://localhost:22222/frontend/5 docker container run --init --detach --name Frontend2-p 2222:22 Frontend:Image
[base] $ curl -v http://localhost:22222/frontend/5 docker container run --init --detach --name Frontend2-p 2222:22 Frontend:Image
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5760516224bc Frontend:Image "/usr/sbin/sshd -D" 3 seconds ago Up 1 second 0.0.0.0:22-->22/tcp, ::2222:22>22/tcp Frontend2
[base] $ curl -v http://localhost:22222/frontend/5 docker container run --init --detach --name Frontend2-p 2222:22 Frontend:Image
[base] $ curl -v http://localhost:22222/frontend/5 docker container run --init --detach --name Frontend2-p 2222:22 Frontend:Image
```

## Ubuntu

Passphrase: hpc\_cloud

```
(base) tigagoollveira@whiterock:~/Downloads/HPChub/_cluster/Config-Files-Frontend$ ssh-keygen -t rsa 4096 -b 4096 -f ./ssh/Frontend-key
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter passphrase again:
Your identification has been saved in /home/tigagoollveira/.ssh/frontend-key
Your public key has been saved in /home/tigagoollveira/.ssh/frontend-key.pub
The key fingerprint is:
SHA256:KwDkWzLwZmHg5FeLn6kwZYJvFKZlTuWRgMh tigagoollveira@whiterock
The key's randomart image is:
++-[RSA 4096]--+
|+*+..|
|+*+o+ |
|+*+o+ + |
|Eoo+ * B o .|
|..= + = .|
| .o+ .|
| .x .|
|oo. .|
++-[SHA256]--+
```

## Ubuntu

5. Descarga los ficheros de configuración de Hadoop para este FrontEnd desde [aqui](#)
  6. Comprueba que puedes conectarte al FrontEnd por ssh como usuario luser y que puedes ejecutar el comando `hadoop version`

**To run a container where the host public-key is mounted host->container**

```
docker container run --init --name frontend -p 2222:22 -v  
/home/tiagooooliveira/.ssh/frontend-key.pub:/home/luser/.ssh/authorized_keys frontend-image
```

```
[base] tiago@liverpool:~/autotester$ ./loadtest /HPChubr_cluster/Config-files-frontends $ sudo docker container run --init --name frontend2 -p 2222:22 -v /home/tiago@liverpool:/ssh:/frontend-key-pub:/home/usr_ssh/authorized_keys frontend-test-image
[tiago@liverpool ~]$ ls
[base] tiago@liverpool:~/autotester$ ./loadtest /HPChubr_cluster/Config-files-frontends
```

## To see if the ssh port is open

```
sudo ss -tulpn | grep ':2222'
```

## To connect to container SSH service with luser

```
ssh -i ~/.ssh/frontend-key -p 2222 luser@localhost
```

Ubuntu

2. Crea un docker-compose.yml que automatice el despliegue de todos los contenedores
  1. Todas las imágenes, excepto la imagen base que la construís antes con docker build, se deben construir usando docker compose
  2. Los comandos que se ejecutan en los contenedores deben especificarse en el docker-compose.yml, no en los Dockerfiles (para asegurar que los diferentes servicios se ejecuten en orden usad, en el fichero yml, la orden depends\_on)
  3. Recordad publicar los puertos necesarios para cada servicio
  4. Adicionalmente, para el contenedor con el FrontEnd debéis indicar que comparta dos directorios, uno para los códigos y otro para los datos, del PC anfitrión al directorio /home/luser del contenedor (por ejemplo, a los directorios /home/luser/codigos y /home/luser/datos). Nota: no es necesario crear esas carpetas de forma explícita, se crean al compartir.
  5. Para comprobar que el fichero yml está bien construido podéis ejecutar: docker compose config
  6. Para iniciar el cluster indicando que le lancen varias instancias del servicio Datanode/Nodemanager utilizad la opción --scale del comando docker-compose: docker compose up --scale dnmm=4 -d

### docker-compose.yml

```
version: '3'

services:
  hpcnube-namenode-image:
    build:
      context: ./NameNode
      dockerfile: Dockerfile
    image: hpcnube-namenode-image:latest

  hpcnube-resourcemanager-image:
    build:
      context: ./ResourceManager
      dockerfile: Dockerfile
    image: hpcnube-resourcemanager-image:latest

  hpcnube-dnnm-image:
    build:
      context: ./DataNode-NodeManager
      dockerfile: Dockerfile
    image: hpcnube-dnnm-image:latest

  frontend-image:
    build:
      context: ./FrontEnd
      dockerfile: Dockerfile
```

```

image: frontend-image:latest
volumes:
- /home/tiagooooliveira/.ssh/frontend-key.pub:/home/luser/.ssh/authorized_keys
- /home/tiagooooliveira/HPCNube_cluster/luser:/home/luser
- /home/tiagooooliveira/HPCNube_cluster/luser/codigos:/home/luser/codigos
- /home/tiagooooliveira/HPCNube_cluster/luser/datos:/home/luser/datos

ports:
- "2222:22"

hpcnube-namenode:
  container_name: hpcnube-namenode
  image: hpcnube-namenode-image:latest
  networks:
  - hpcnube-net
  hostname: hpcnube-namenode
  ports:
  - "9870:9870"
  #init: true

hpcnube-resourcemanager:
  container_name: hpcnube-resourcemanager
  image: hpcnube-resourcemanager-image:latest
  networks:
  - hpcnube-net
  hostname: hpcnube-resourcemanager
  ports:
  - "8088:8088"
  #init: true

hpcnube-dnnm:
  build:
    context: ./DataNode-NodeManager
    dockerfile: Dockerfile
  image: hpcnube-dnnm-image:latest
  networks:
  - hpcnube-net
  hostname: hpcnube-dnnm
  #init: true
  depends_on:
  - hpcnube-namenode
  - hpcnube-resourcemanager

networks:
  hpcnube-net:

```

## To run the docker-compose

**Before docker-compose up** (to ensure the ./Base image is available for others usage)

docker image build -t hpcnube-base-image:1.0.0 ./Base && docker image tag

hpcnube-base-image:1.0.0 hpcnube-base-image:latest

docker-compose up --scale hpcnube-dnnm=4 -d

```
[User] ~ % docker compose -r > ./downLoad/HPCnube_Cluster$ docker-compose up --scale hpcnube-dnn=4 -d
Building hpcnube-nanode-1image
Step 1/2 : FROM hpcnube-base:ImageLatest
Step 2/2 : MAINTAINER Toma <t.f.penarust@es>
--> Running In 5a4def44dd09
Removing intermediate container 5a4def44dd09
--> d59eacf95c
Step 3/2 : ENV HADOOP_HOME /hadoop
--> Running In 5d5f3ba2a24c5f
Removing Intermediate container 5d5f3ba2a24c5f
Step 4/2 : ENV HADOOP_VERSION 3.3.6
--> Running In b883fbfb330
Removing Intermediate container b883fbfb330
--> 5f2b043d219
Step 5/2 : ENV LOG_TAG "[Hadoop_S][HADOOP_VERSION]"
--> Running In 590852542962
Removing Intermediate container 590852542962
--> 11dc472ea21
Step 6/2 : ENV SEASIDE_DIR /opt/hd
--> Running In 590852542962
Removing Intermediate container 590852542962
--> e824918a4d20
Step 7/2 : ENV HADOOP_HOME $SEASIDE_DIR/hadoop/
--> Running In 590852542962
Removing Intermediate container 9e5ec58a7a0
--> 3d68d9ad0f9f
Step 8/2 : ENV HADOOP_CONF_DIR $HADOOP_HOME/etc/hadoop/
--> Running In 55ed72ce25e
Removing Intermediate container b55d72ce25e
--> 89ed674a0b85
Step 9/2 : ENV DATA_DIR /var/lib/hadoop/hdfs
--> Running In 55ed72ce25e
Removing Intermediate container 095671b4bc27
--> 2ae7b18d141
Step 10/2 : RUN hdfs dfs -mkdir -p $LOG_DIR Crea directorios para los datos de HDFS del NameNode y haz que sean propiedad del usuario hddmin
--> Running In 0a317e546046
Removing Intermediate container 0a317e546046
Step 11/2 : RUN hdfs dfs -mkdir -p $DATA_DIR Crea directorios para los datos de HDFS del NameNode y haz que sean propiedad del usuario hddmin
--> Running In 0a317e546046
Removing Intermediate container 0a317e546046
Step 12/2 : RUN hdfs namenode -format Crea directorio para los ficheros de losod
--> Running In 5f28013f779a
Removing Intermediate container 5f28013f779a
mkitdr -p $(DATA_DIR)/nn RA C
```

```
>>> c4db5280eefb
Step 5/9 : RUN echo 'export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64' >> /home/luser/.bashrc
--> Running in 36d7ebcb9b247
Removing intermediate container 36d7ebcb9b247
--> f174ecc0e60a
Step 6/9 : RUN echo 'export HADOOP_HOME=/path/to/hadoop' >> /home/luser/.bashrc
--> Running in 4bd250a18ffb
Removing intermediate container 4bd250a18ffb
--> beddf201288e
Step 7/9 : RUN echo 'export PATH=$[HADOOP_HOME]/bin:$PATH' >> /home/luser/.bashrc
--> Running in 5048ae6c728c
Removing intermediate container 5048ae6c728c
--> 725acee86524f
Step 8/9 : EXPOSE 22
--> Running in e3894eb8f329
Removing intermediate container e3894eb8f329
--> 46734ade6328
Step 9/9 : CMD ["/usr/sbin/sshd", "-D"]
--> Running in 9be2c4ee8f93
Removing intermediate container 9be2c4ee8f93
--> 97ef95476981
Successfully built 97ef95476981
Successfully tagged frontend-image:latest
WARNING: Image for service frontend-image was built because it did not already exist. To rebuild this
--build'.
Creating hpcnube-resourcemanager ... error
Creating hpcnube_cluster_hpcnube-resourcemanager-image_1 ...
Creating hpcnube_cluster_hpcnube-namenode-image_1 ...
Creating hpcnube_cluster_frontend-image_1 ...
Creating hpcnube-namenode ...
Creating hpcnube_cluster_hpcnube-dnmm-image_1 ...

Creating hpcnube_cluster_hpcnube-resourcemanager-image_1 ... done
Creating hpcnube_cluster_hpcnube-namenode-image_1 ... done
Creating hpcnube_cluster_frontend-image_1 ... done
ERROR: for hpcnube-namenode Cannot create container for service hpcnube-namenode: Conflict. The container name is already in use.
Creating hpcnube_cluster_hpcnube-dnmm-image_1 ... done

ERROR: for hpcnube-resourcemanager Cannot create container for service hpcnube-resourcemanager: Conflict. The container name is already in use by container "ab0b7be34a87e92ee9e17ea7df9d4ee6361cb3909427bfaea522141e5127a829". You have to remove or rename that container.

ERROR: for hpcnube-namenode Cannot create container for service hpcnube-namenode: Conflict. The container name is already in use by container "ab0b7be34a87e92ee9e17ea7df9d4ee6361cb3909427bfaea522141e5127a829". You have to remove or rename that container.

ERROR: Encountered errors while bringing up the project.
(base) tlaigoo@livera@whiteRock:~/Downloads/HPCNube_Cluster$
```

```
(base) tlagoo@oliveira:~/Downloads/HPCNube_Cluster$ docker images
```

REPOSITORY	TAG	IMAGE ID
frontend-image	latest	97ef95476981
hpcnube-dnnm-image	latest	85588df53fce
hpcnube-resourcemanager-image	latest	12124376c8c8
hpcnube-namenode-image	latest	1229f94e68be
hpcnube-base-image	1.0.0	42568030d69b
hpcnube-base-image	latest	42568030d69b
rocky-8-mhpc	latest	f7fd8fc512fe
ffurf	latest	efc9b65b27f7
opensearchproject/opensearch	latest	f6dad1a7bd24

## docker compose config

```
(base) tiagoooliveira@whiterock:~/Downloads/HPCNube_cluster$ docker compose config
name: hpcnube_cluster
services:
  frontend-image:
    build:
      context: /home/tiagoooliveira/Downloads/HPCNube_cluster/FrontEnd
      dockerfile: Dockerfile
      image: hpcnube前端-image:latest
    networks:
      - default: null
    ports:
      - mode: ingress
        target: 22
        published: "2222"
        protocol: tcp
    volumes:
      - type: bind
        source: /home/tiagoooliveira/.ssh/frontend-key.pub
        target: /home/luser/.ssh/authorized_keys
        bind:
          create_host_path: true
  hpcnube-dnnn:
    build:
      context: /home/tiagoooliveira/Downloads/HPCNube_cluster/DataNode-NodeManager
      dockerfile: Dockerfile
    depends_on:
      hpcnube-namenode:
        condition: service_started
        required: true
      hpcnube-resourcemanager:
        condition: service_started
        required: true
    hostname: hpcnube-dnnn
    image: hpcnube-dnnn-image:latest
    networks:
      - hpcnube-net: null
  hpcnube-dnnn-image:
    build:
      context: /home/tiagoooliveira/Downloads/HPCNube_cluster/DataNode-NodeManager
      dockerfile: Dockerfile
      image: hpcnube-dnnn-image:latest
    networks:
      - default: null
  hpcnube-namenode:
    container_name: hpcnube-namenode
    hostname: hpcnube-namenode
    image: hpcnube-namenode-image:latest
```

```
hpcnube-namenode:
  container_name: hpcnube-namenode
  hostname: hpcnube-namenode
  image: hpcnube-namenode-image:latest
  networks:
    - hpcnube-net: null
  ports:
    - mode: ingress
      target: 9870
      published: "9870"
      protocol: tcp
  hpcnube-namenode-image:
    build:
      context: /home/tiagoooliveira/Downloads/HPCNube_cluster/NameNode
      dockerfile: Dockerfile
      image: hpcnube-namenode-image:latest
    networks:
      - default: null
  hpcnube-resourcemanager:
    container_name: hpcnube-resourcemanager
    hostname: hpcnube-resourcemanager
    image: hpcnube-resourcemanager-image:latest
    networks:
      - hpcnube-net: null
    ports:
      - mode: ingress
        target: 8088
        published: "8088"
        protocol: tcp
  hpcnube-resourcemanager-image:
    build:
      context: /home/tiagoooliveira/Downloads/HPCNube_cluster/ResourceManager
      dockerfile: Dockerfile
      image: hpcnube-resourcemanager-image:latest
    networks:
      - default: null
  networks:
    default:
      name: hpcnube_cluster_default
    hpcnube-net:
      name: hpcnube_cluster_hpcnube-net
(base) tiagoooliveira@whiterock:~/Downloads/HPCNube_cluster$
```

## Notes:

1. I tried to run this task first on my MacOS, but some configurations for Docker network were not working well. So I switched to my Ubuntu Linux laptop.
2. I have to comment on the “init: true” instruction in docker-compose due to some compatibility issue: “Unsupported config option for...”. I have to double check the reason for that because this options ensures that the container's main process is properly monitored and reaps zombie processes.
3. I could not manage to make the ssh service up and running due to the following issue - that I could not solve on time: “Missing privilege separation directory: /run/sshd”. Maybe I

have to change the approach of sharing/mapping the host public key with “luser” inside the container via the mounting repository. Also added “USER root” in the Dockerfile to ensure the right power to run ssh service.

- a. I will try one of the solutions proposed here  
<https://stackoverflow.com/questions/22886470/start-sshd-automatically-with-docker-container>
4. The docker-compose finishes with some errors besides the images are created. It seems I have to perform deeper clean-up (--force) from previous tests, since it complains mostly about conflicts when creating those images: hpcnube-resourcemanager, hpcnube-namenode and frontend-image.