Docker

Jesús Morán y Cristian Augusto

Grupo de Investigación en Ingeniería del Software
http://giis.uniovi.es
Universidad de Oviedo

Problemas de “desarrollo”

No sé qué pudo ocurrir… en mi PC funcionaba

Instala la nueva versión de java

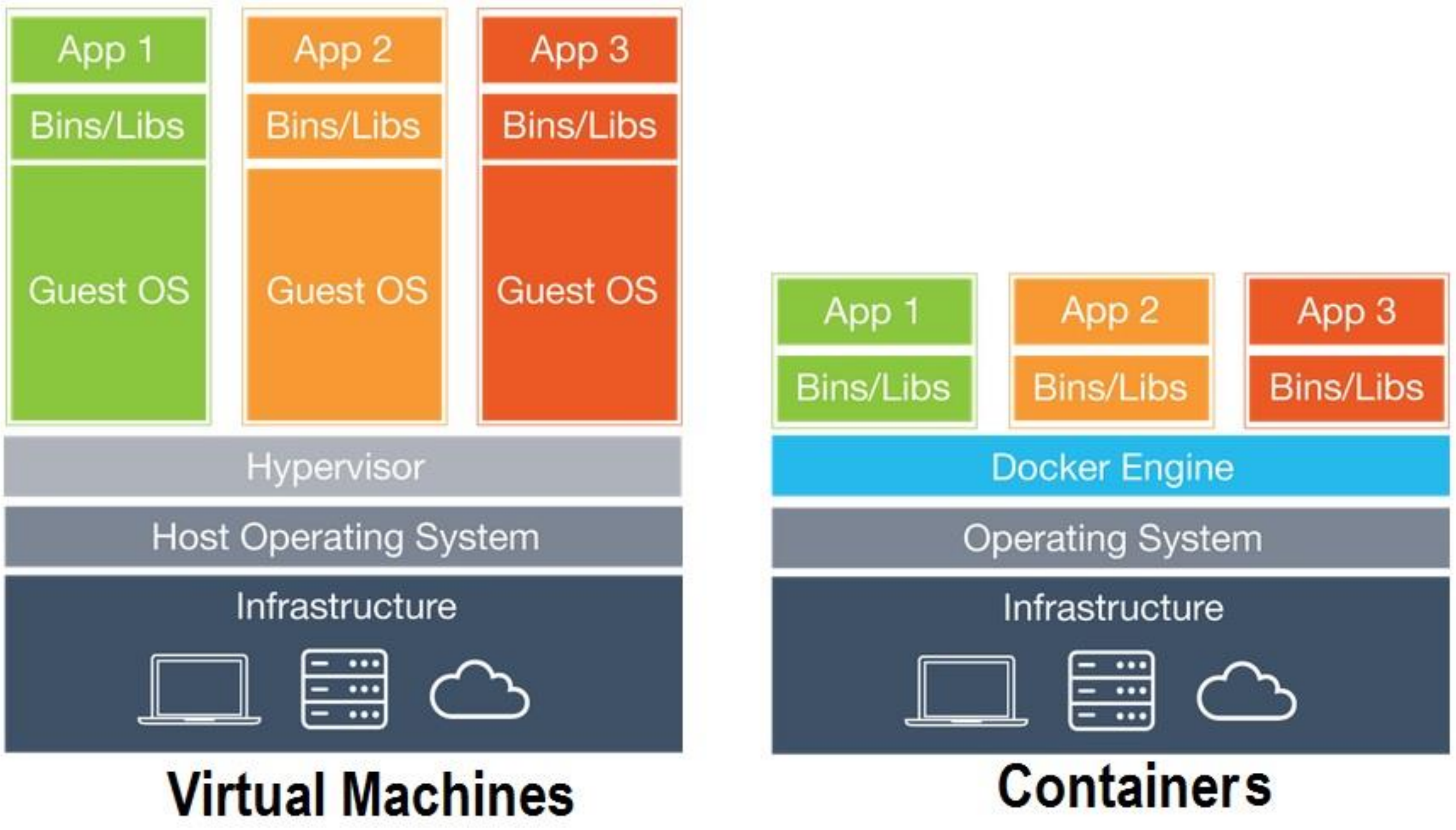
Falta una librería

Tienes que abrir el puerto 8000

Prueba a instalar tomcat

…

Máquina virtual vs Docker



Crédito de la imagen: docker.com

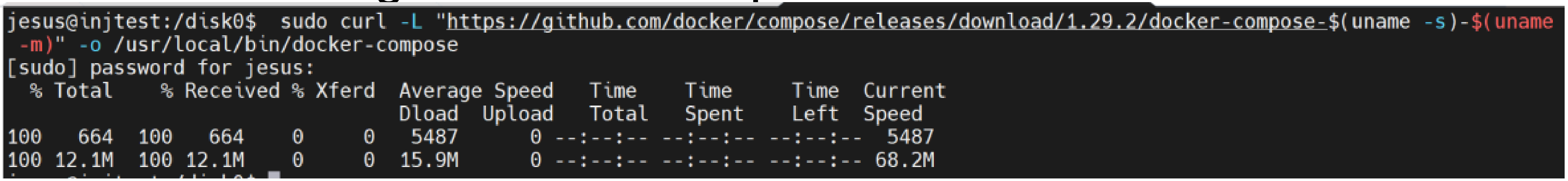
Terminología docker

- **Imagen**: "*An executable package that includes everything needed to run an application -the code, a runtime, libraries, environment variables, and configuration files.*" Docker.com

- **Contenedor**: "A runtime instance of an image--what the image becomes in memory when executed (that is, an image with state, or a user process" Docker.com

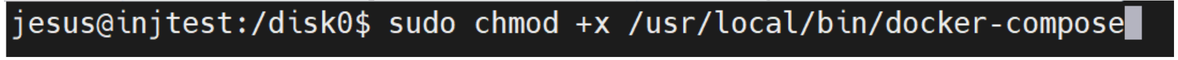
Docker Compose

- Instalación:
  - □ Descargar docker-compose



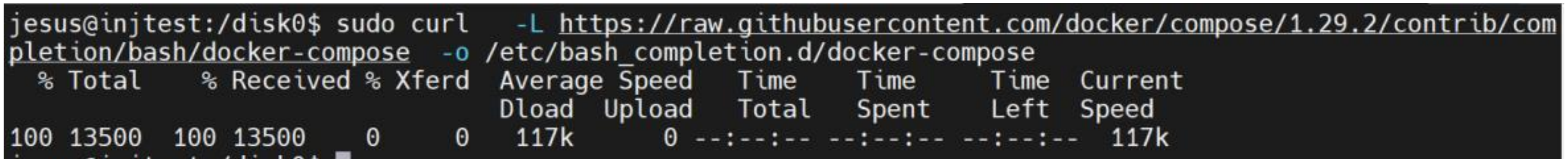
```
jesus@injtest:/disk0$  sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname
 -m)" -o /usr/local/bin/docker-compose
[sudo] password for jesus:
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   664  100   664    0     0   5487      0 --:--:-- --:--:-- --:--:--  5487
100 12.1M  100 12.1M    0     0  15.9M      0 --:--:-- --:--:-- --:--:-- 68.2M
```

  - □ Dar permisos de ejecución



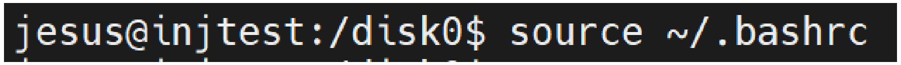
```
jesus@injtest:/disk0$ sudo chmod +x /usr/local/bin/docker-compose
```

  - □ Instalar command line completion



```
jesus@injtest:/disk0$ sudo curl   -L https://raw.githubusercontent.com/docker/compose/1.29.2/contrib/com
pletion/bash/docker-compose  -o /etc/bash_completion.d/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 13500  100 13500    0     0   117k      0 --:--:-- --:--:-- --:--:--  117k
```

  - □ Actualizar la configuración de la terminal



```
jesus@injtest:/disk0$ source ~/.bashrc
```

Docker Compose

```
docker-compose --version
```

- Desde Docker Desktop 3.4.0 hay dos versiones
  - La versión docker-compose: implementada en python
  - La versión docker compose (dentro de docker-cli):
    - Se conoce como Compose V2
    - Implementada en Go
    - Todavía no tiene toda la funcionalidad de docker-compose
    - Más amigable con la nube:
      - Backends para guardar imágenes: ECS de AWS, ACI de Azure,…

```
docker compose --version
```

Docker Compose

```
version: '3.3’

services:
    db:
        image: mysql:5.7
        volumes:
             - dbdata:/var/lib/mysql
        restart: always
        environment:
             MYSQL_ROOT_PASSWORD: somewordpress
             MYSQL_DATABASE: wordpress
             MYSQL_USER: wordpress
             MYSQL_PASSWORD: wordpress
    wordpress:
        depends_on:
             - db
        image: wordpress:latest
        ports:
             - "8000:80"
        restart: always
        environment:
             WORDPRESS_DB_HOST: db:3306
             WORDPRESS_DB_USER: wordpress
              WORDPRESS_DB_PASSWORD: wordpress
volumes:
         dbdata:
```

Docker Compose

- Desplegar/arrancar los contenedores

```
docker-compose up
```

- Parar los contenedores

```
docker-compose stop
```

- Eliminar contenedores + red

```
docker-compose down
```

- Eliminar contenedores + red + volúmenes

```
docker-compose down --volumes
```

Docker Compose

- “Construir” o “Reconstruir” las imágenes

```
docker-compose build
```

- Reiniciar los contenedores

```
docker-compose restart
```

- Descargar las imágenes necesarias

```
docker-compose pull
```

- Modo “detached” (no se muestran los logs)

```
docker-compose --detach
```

Docker Compose: cluster de pruebas

- Virtualizar un cluster con Docker compose:
  - Instalar Docker (en Almalinux):

```
sudo yum install -y yum-utils device-mapper-persistent-data lvm2
```

```
sudo yum-config-manager --add-repo
```

```
sudo yum insall docker-ce
```

  - Inicializar Docker:

```
sudo systemctl start docker
```

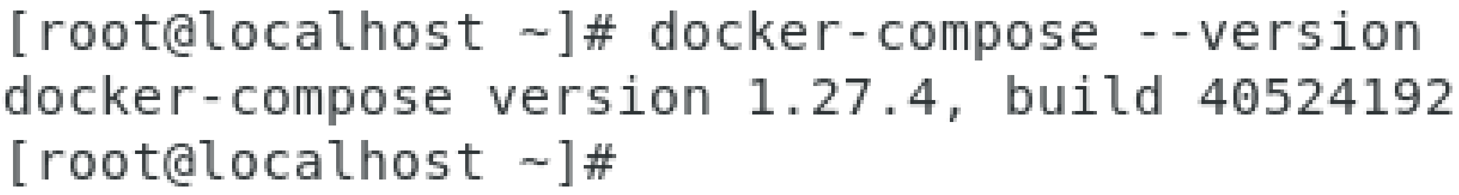
Docker Compose: cluster de pruebas

- Virtualizar un cluster con Docker compose:
  - Instalar Docker Compose (en Almalinux):
    - Descargar de Github:

```
sudo curl -L
"https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

    - Le assignamos permisos de ejecución:

```
sudo chmod +x /usr/local/bin/docker-compose
```

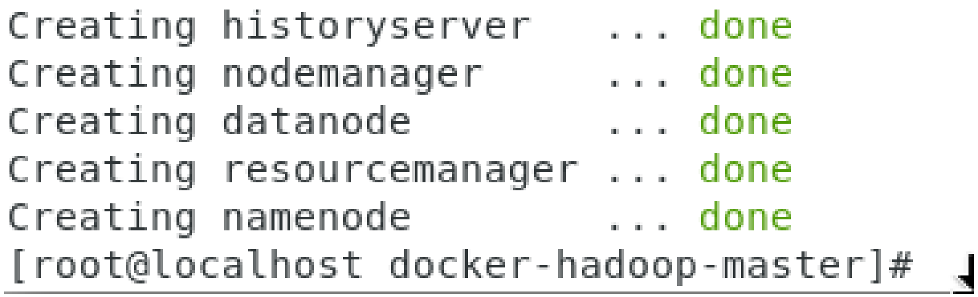


```
[root@localhost ~]# docker-compose --version
docker-compose version 1.27.4, build 40524192
[root@localhost ~]#
```

Docker Compose: cluster de pruebas

- Virtualizar un cluster con Docker compose:
  - □ Instalar el cluster Big Data:
    - Descargar las imágenes de Hadoop de Big Data Europe: https://github.com/big-data-europe/docker-hadoop
    - Descomprimir las imagenes: `unzip docker-hadoop-master.zip`
    - Entrar en la carpeta: `cd docker-hadoop-master`
    - Desplegar el cluster de 3 nodos:

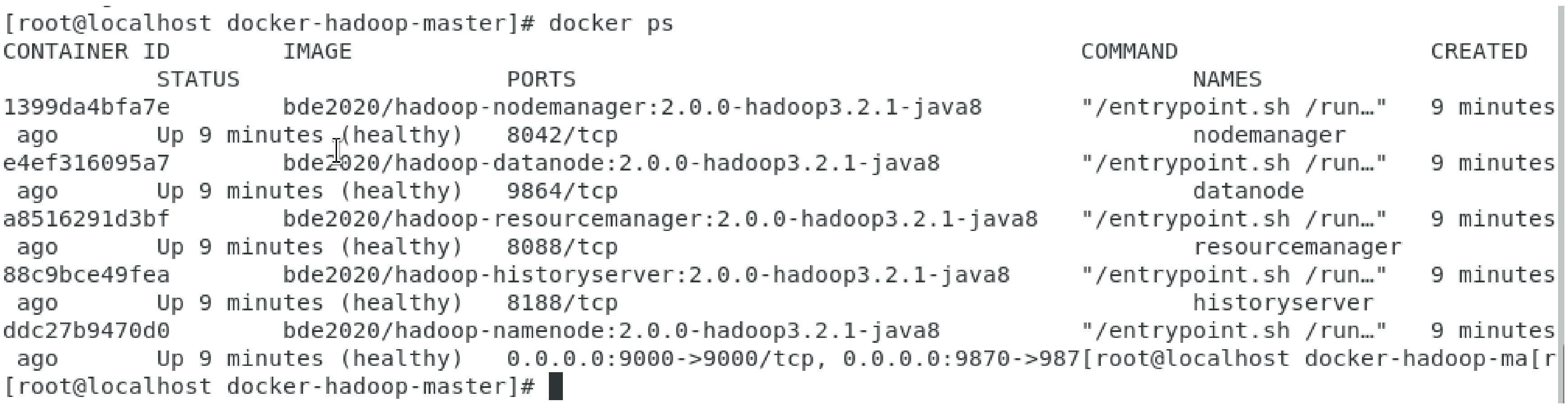
```
docker-compose up -d
```



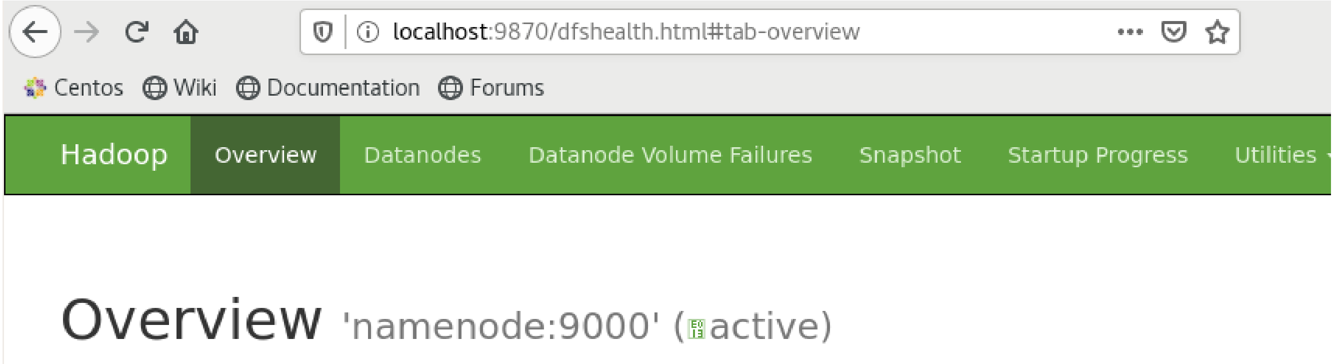
```
Creating historyserver    ... done
Creating nodemanager      ... done
Creating datanode         ... done
Creating resourcemanager ... done
Creating namenode         ... done
[root@localhost docker-hadoop-master]#
```

Docker Compose: cluster de pruebas

- Virtualizar un cluster con Docker compose:
  - Comprobar que se desplegaron los contenendores



```
[root@localhost docker-hadoop-master]# docker ps
CONTAINER ID        IMAGE                                                    COMMAND                  CREATED
          STATUS                       PORTS                                                     NAMES
1399da4bfa7e        bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8       "/entrypoint.sh /run…"   9 minutes
 ago        Up 9 minutes (healthy)    8042/tcp                                                    nodemanager
e4ef316095a7        bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8          "/entrypoint.sh /run…"   9 minutes
 ago        Up 9 minutes (healthy)    9864/tcp                                                    datanode
a8516291d3bf        bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8   "/entrypoint.sh /run…"   9 minutes
 ago        Up 9 minutes (healthy)    8088/tcp                                                    resourcemanager
88c9bce49fea        bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8     "/entrypoint.sh /run…"   9 minutes
 ago        Up 9 minutes (healthy)    8188/tcp                                                    historyserver
ddc27b9470d0        bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8          "/entrypoint.sh /run…"   9 minutes
 ago        Up 9 minutes (healthy)    0.0.0.0:9000->9000/tcp, 0.0.0.0:9870->987[root@localhost docker-hadoop-ma[r
[root@localhost docker-hadoop-master]#
```



localhost:9870/dfshealth.html#tab-overview

Overview 'namenode:9000' (active)

Docker Compose: cluster de pruebas

- Virtualizar un cluster con Docker compose:
  - Ejecutar programas en el cluster:
    - Conectarse a un nodo: `sudo docker exec -it namenode bash`
    - Crear datos de entrada:

```
mkdir libros
cd libros
curl http://www.gutenberg.org/cache/epub/2000/pg2000.txt> quijote.txt
curl http://www.gutenberg.org/cache/epub/345/pg345.txt > dracula.txt
```

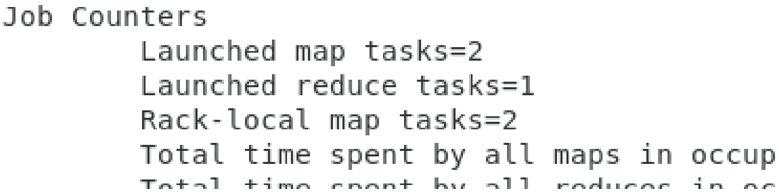
    - Cargar los datos en el Sistema de archivos distribuidos HDFS:

```
hdfs dfs -mkdir -p librosEnHDFS
hdfs dfs -put ./librosEnHDFS
```

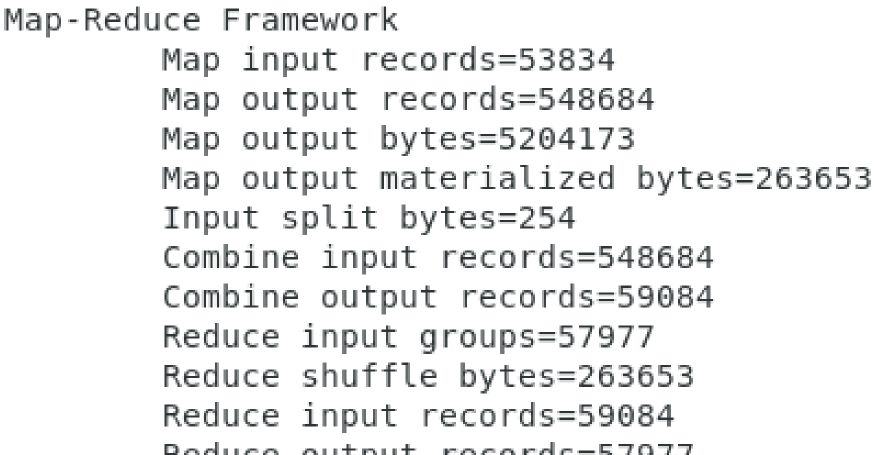
Docker Compose: cluster de pruebas

- Virtualizar un cluster con Docker compose:
  - Ejecutar programas en el cluster:
    - Ejecutar programa:

```
hadoop jar
$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-
examples-${HADOOP_VERSION}.jar wordcount
/user/root/librosEnHDFS/libros/
/user/root/salidaEnHDFS
```



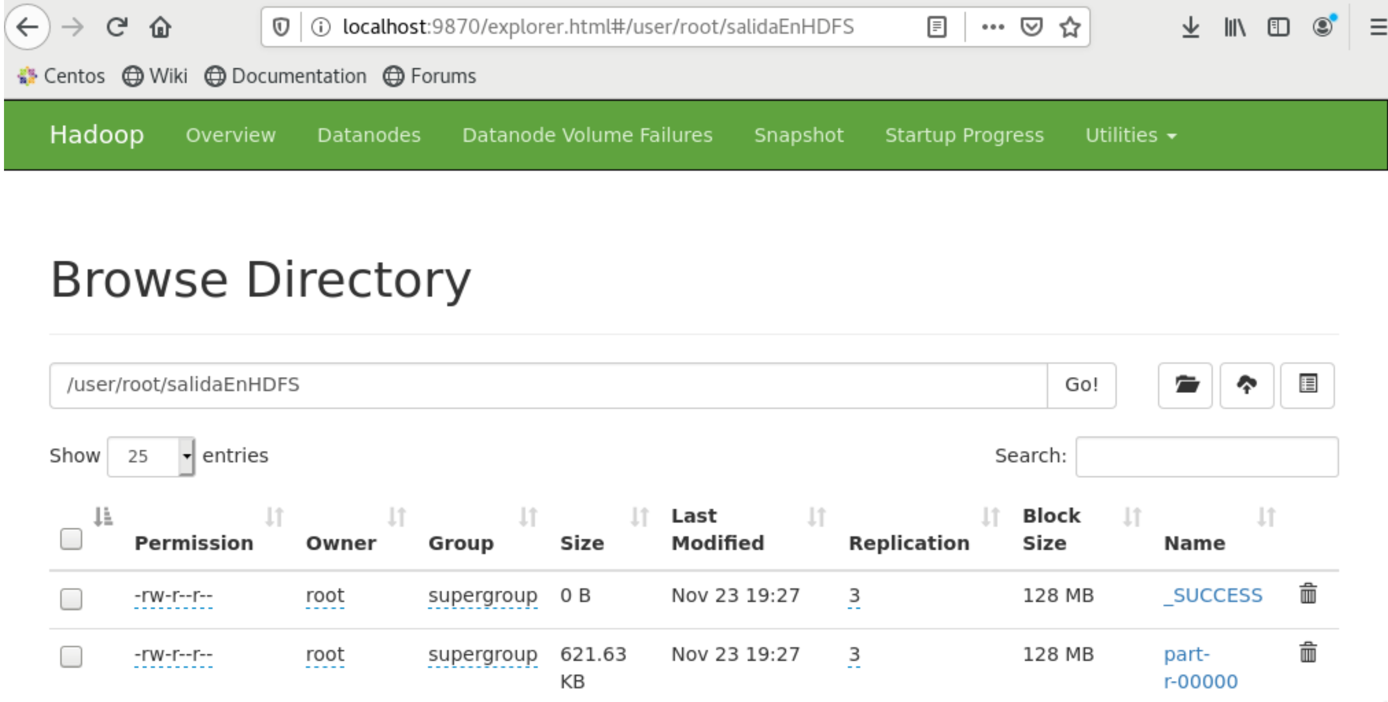
```
Job Counters
        Launched map tasks=2
        Launched reduce tasks=1
        Rack-local map tasks=2
        Total time spent by all maps in occup
        Total time spent by all reduces in oc
```



```
Map-Reduce Framework
        Map input records=53834
        Map output records=548684
        Map output bytes=5204173
        Map output materialized bytes=263653
        Input split bytes=254
        Combine input records=548684
        Combine output records=59084
        Reduce input groups=57977
        Reduce shuffle bytes=263653
        Reduce input records=59084
        Reduce output records=57977
```

Docker Compose: cluster de pruebas

- Virtualizar un cluster con Docker compose:
  - Ejecutar programas en el cluster:
    - Comprobar la salida:
      - Desde servidor web:



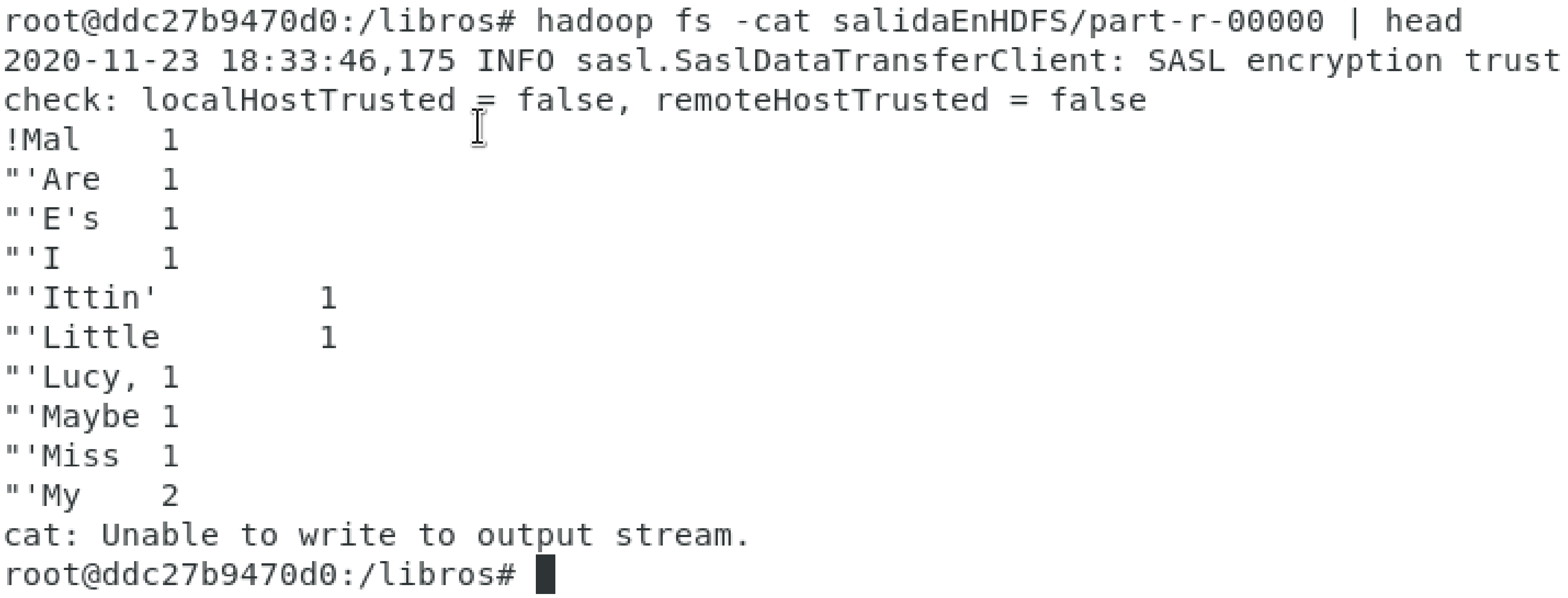
Browse Directory

/user/root/salidaEnHDFS

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	root	supergroup	0 B	Nov 23 19:27	3	128 MB	_SUCCESS
□	-rw-r--r--	root	supergroup	621.63 KB	Nov 23 19:27	3	128 MB	part-r-00000

Docker Compose: cluster de pruebas

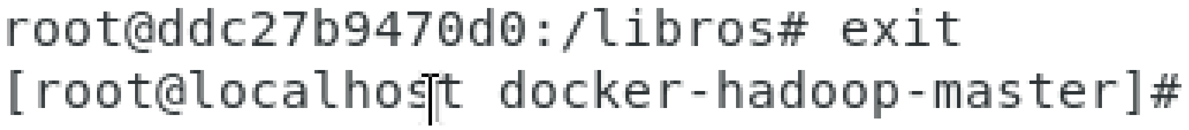
- Virtualizar un cluster con Docker compose:
  - Ejecutar programas en el cluster:
    - Comprobar la salida:
      - Desde la CLI



```
root@ddc27b9470d0:/libros# hadoop fs -cat salidaEnHDFS/part-r-00000 | head
2020-11-23 18:33:46,175 INFO sasl.SaslDataTransferClient: SASL encryption trust
check: localHostTrusted = false, remoteHostTrusted = false
!Mal    1
"'Are   1
"'E's   1
"'I     1
"'Ittin'        1
"'Little        1
"'Lucy, 1
"'Maybe 1
"'Miss  1
"'My    2
cat: Unable to write to output stream.
root@ddc27b9470d0:/libros# 
```

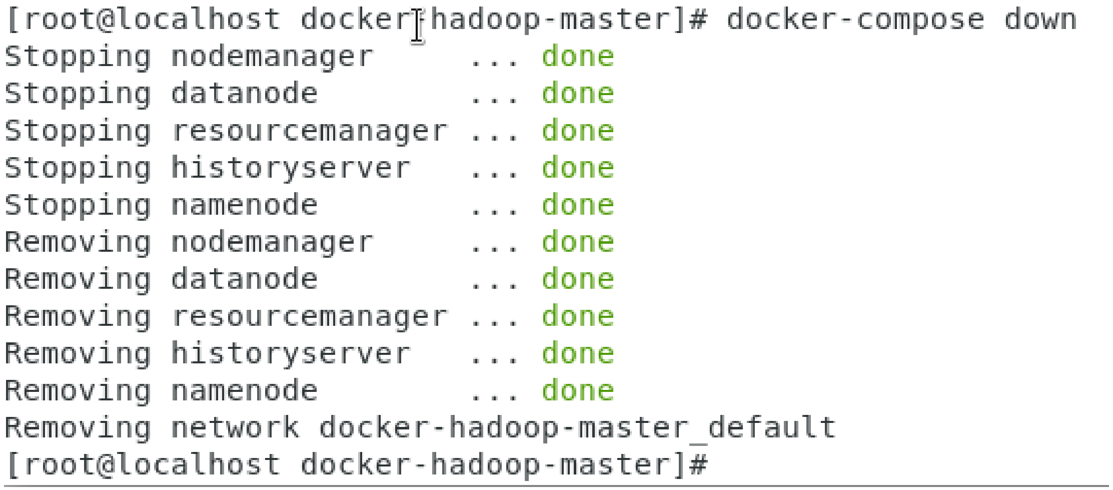
Docker Compose: cluster de pruebas

- Virtualizar un cluster con Docker compose:
  - □ Apagar cluster:
    - Salir del contenedor: Control + D



```
root@ddc27b9470d0:/libros# exit
[root@localhost docker-hadoop-master]#
```

    - Apagar los contenedores: `docker-compose down`



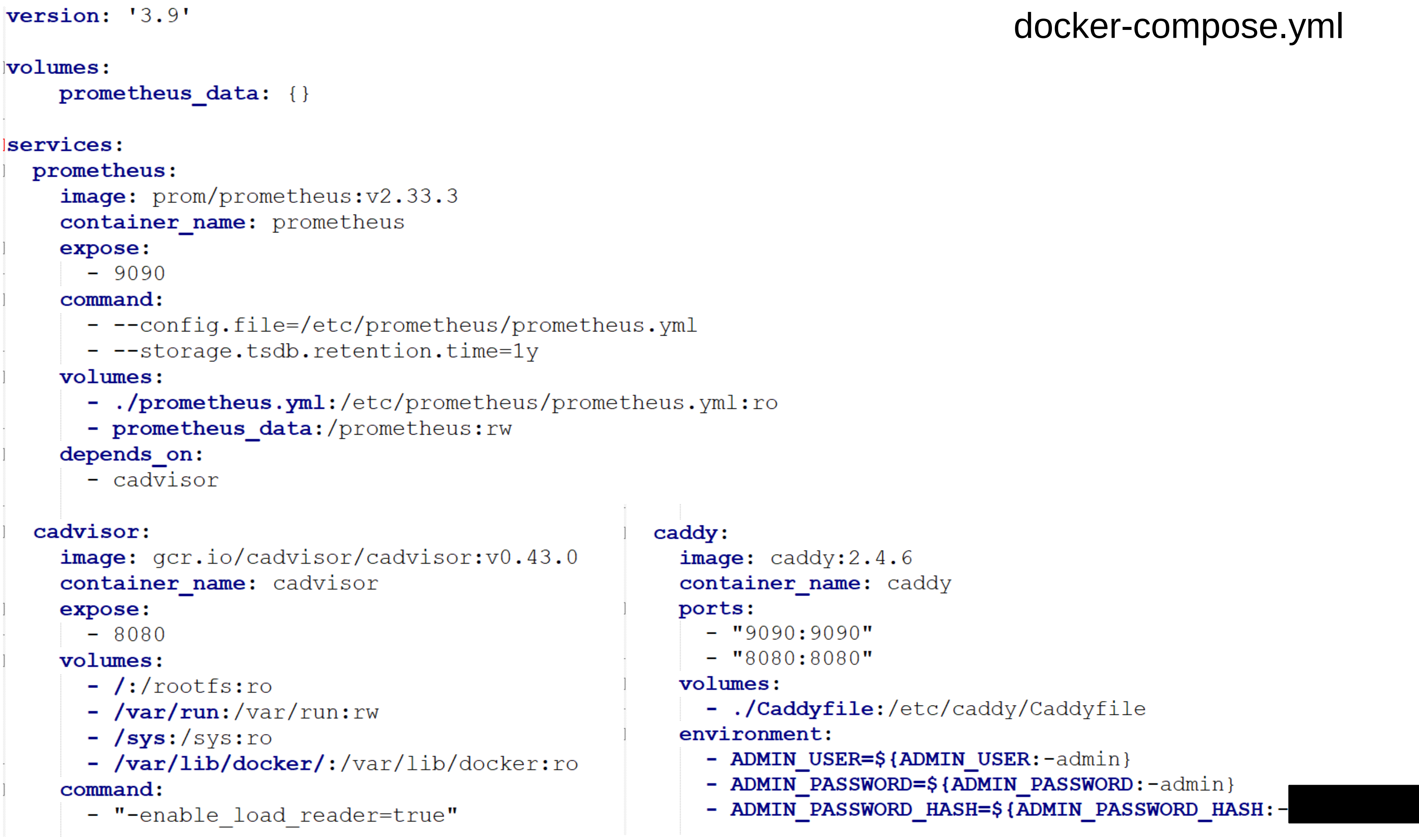
```
[root@localhost docker-hadoop-master]# docker-compose down
Stopping nodemanager      ... done
Stopping datanode         ... done
Stopping resourcemanager ... done
Stopping historyserver    ... done
Stopping namenode         ... done
Removing nodemanager      ... done
Removing datanode         ... done
Removing resourcemanager ... done
Removing historyserver    ... done
Removing namenode         ... done
Removing network docker-hadoop-master_default
[root@localhost docker-hadoop-master]#
```

Monitorizar Docker

- Crear docker-compose con servicios:
  - cAdvisor: obtiene métrica del uso de contenedores
  - Prometheus: base de datos que almacena las métricas
  - caddy: servidor web y proxy reverso

Monitorizar Docker

docker-compose.yml



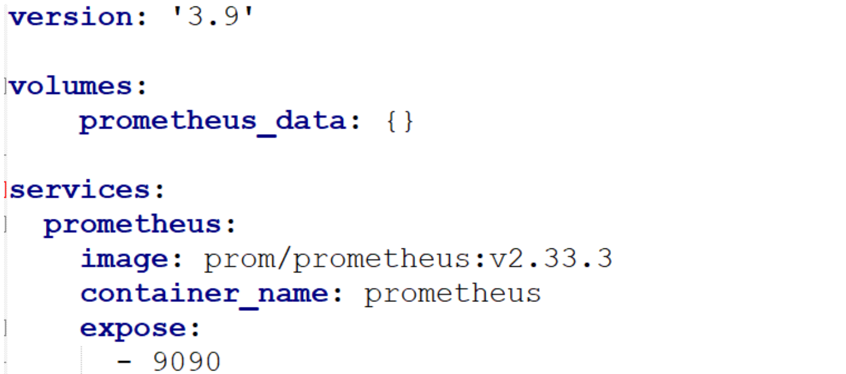
```
version: '3.9'

volumes:
    prometheus_data: {}

services:
  prometheus:
    image: prom/prometheus:v2.33.3
    container_name: prometheus
    expose:
      - 9090
    command:
      - --config.file=/etc/prometheus/prometheus.yml
      - --storage.tsdb.retention.time=1y
    volumes:
      - ./prometheus.yml:/etc/prometheus/prometheus.yml:ro
      - prometheus_data:/prometheus:rw
    depends_on:
      - cadvisor

  cadvisor:
    image: gcr.io/cadvisor/cadvisor:v0.43.0
    container_name: cadvisor
    expose:
      - 8080
    volumes:
      - /:/rootfs:ro
      - /var/run:/var/run:rw
      - /sys:/sys:ro
      - /var/lib/docker/:/var/lib/docker:ro
    command:
      - "-enable_load_reader=true"
```

```
  caddy:
    image: caddy:2.4.6
    container_name: caddy
    ports:
      - "9090:9090"
      - "8080:8080"
    volumes:
      - ./Caddyfile:/etc/caddy/Caddyfile
    environment:
      - ADMIN_USER=${ADMIN_USER:-admin}
      - ADMIN_PASSWORD=${ADMIN_PASSWORD:-admin}
      - ADMIN_PASSWORD_HASH=${ADMIN_PASSWORD_HASH:-[redacted]
```

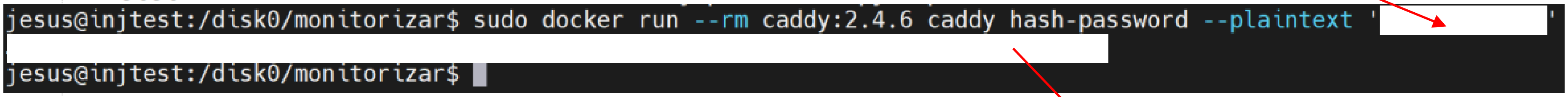
Monitorizar Docker

docker-compose.yml

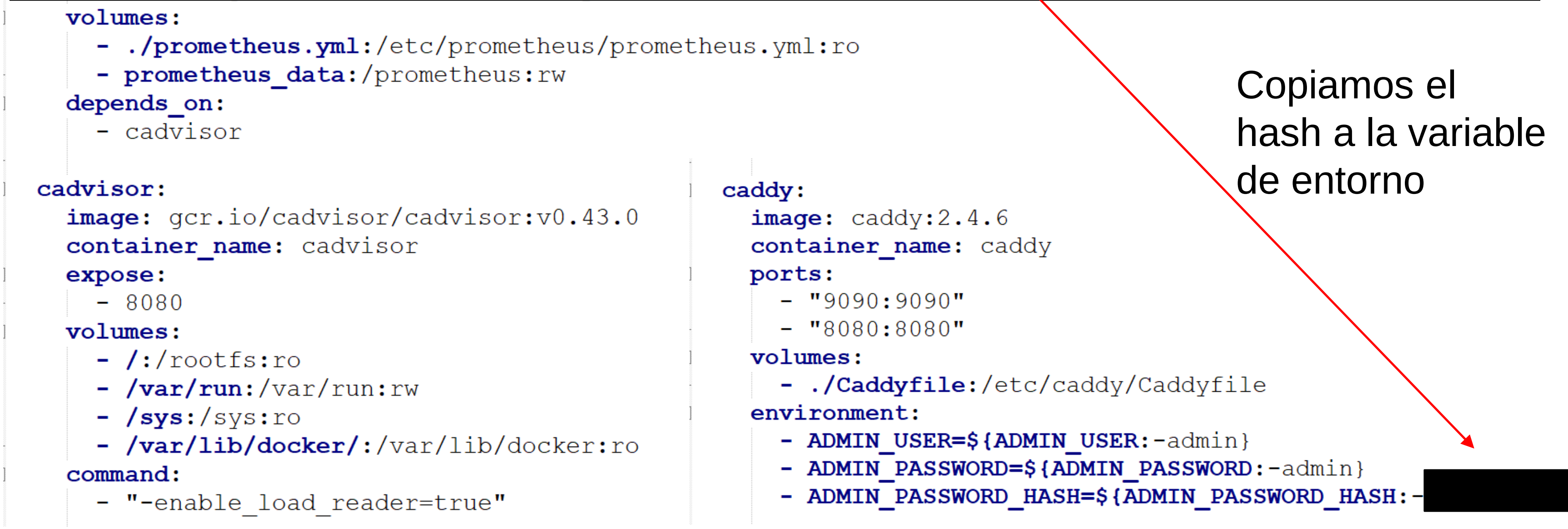
Para saber cuál es el hash de la contraseña que queremos asignar: escribimos la contraseña



```
version: '3.9'

volumes:
    prometheus_data: {}

services:
  prometheus:
    image: prom/prometheus:v2.33.3
    container_name: prometheus
    expose:
      - 9090
```



```
jesus@injtest:/disk0/monitorizar$ sudo docker run --rm caddy:2.4.6 caddy hash-password --plaintext '[redacted]'
[redacted]
jesus@injtest:/disk0/monitorizar$
```

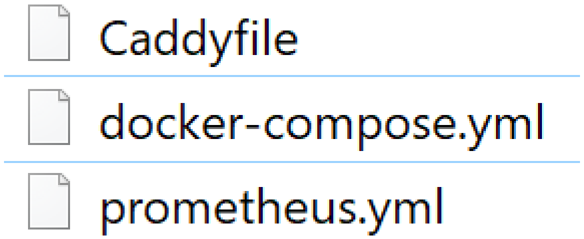


```
    volumes:
      - ./prometheus.yml:/etc/prometheus/prometheus.yml:ro
      - prometheus_data:/prometheus:rw
    depends_on:
      - cadvisor

  cadvisor:
    image: gcr.io/cadvisor/cadvisor:v0.43.0
    container_name: cadvisor
    expose:
      - 8080
    volumes:
      - /:/rootfs:ro
      - /var/run:/var/run:rw
      - /sys:/sys:ro
      - /var/lib/docker/:/var/lib/docker:ro
    command:
      - "-enable_load_reader=true"
```

```
  caddy:
    image: caddy:2.4.6
    container_name: caddy
    ports:
      - "9090:9090"
      - "8080:8080"
    volumes:
      - ./Caddyfile:/etc/caddy/Caddyfile
    environment:
      - ADMIN_USER=${ADMIN_USER:-admin}
      - ADMIN_PASSWORD=${ADMIN_PASSWORD:-admin}
      - ADMIN_PASSWORD_HASH=${ADMIN_PASSWORD_HASH:-[redacted]
```

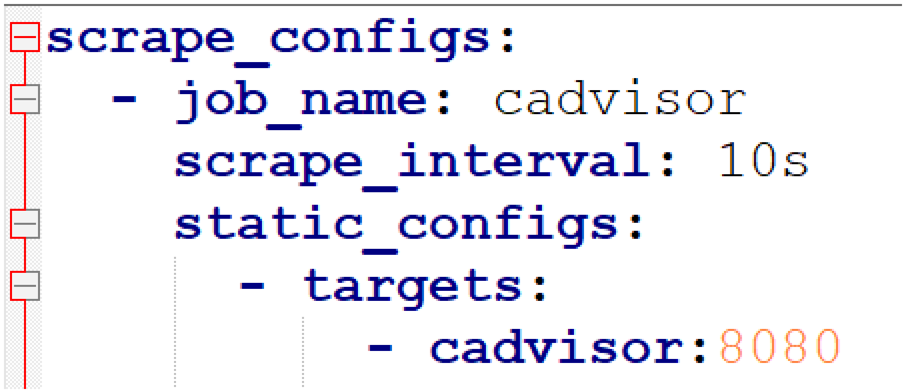
Copiamos el hash a la variable de entorno

Monitorizar Docker



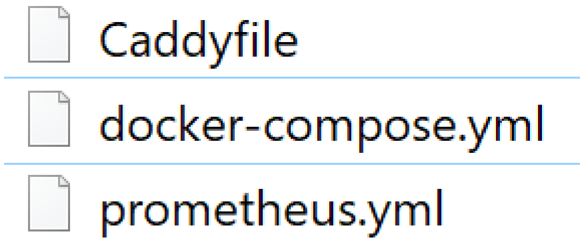
Caddyfile
docker-compose.yml
prometheus.yml

- Archivos:
  - docker-compose-yml: contiene los contenedores
  - prometheus.yml:
    - Cada 10s guarda las métricas de cadvisor



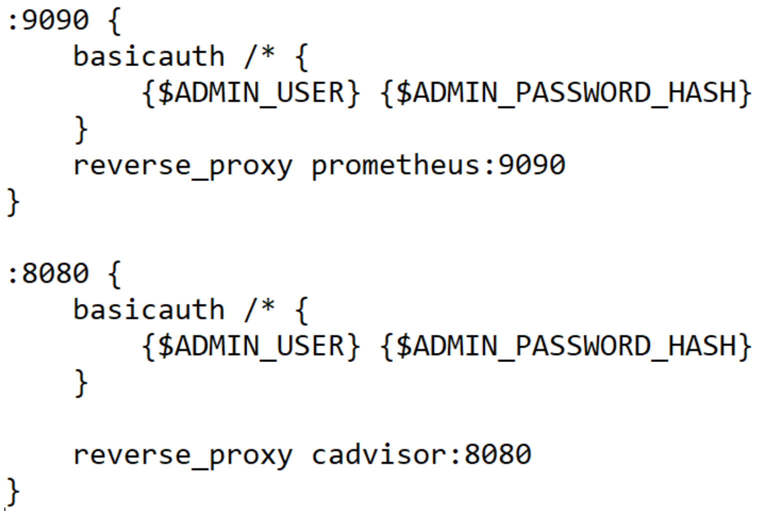
```
scrape_configs:
  - job_name: cadvisor
    scrape_interval: 10s
    static_configs:
      - targets:
        - cadvisor:8080
```

Monitorizar Docker



Caddyfile
docker-compose.yml
prometheus.yml

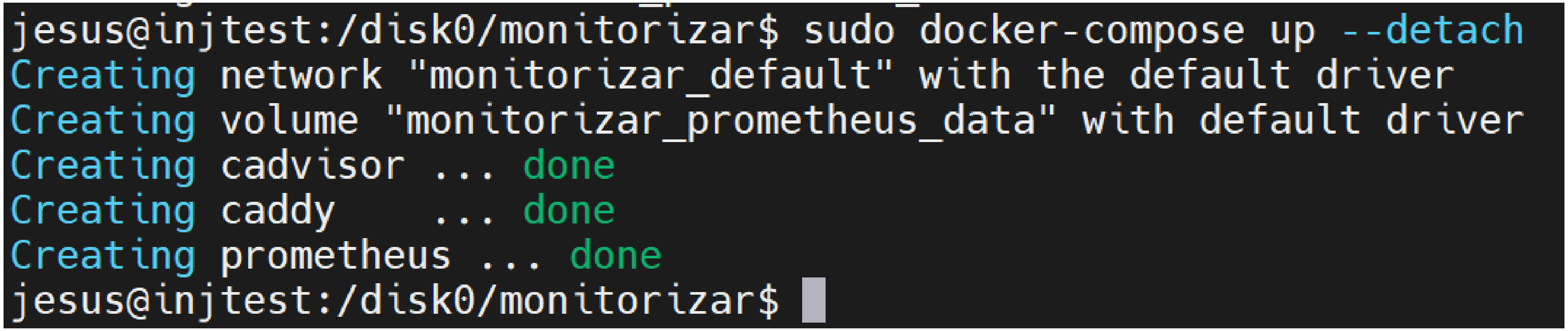
- Archivos:
  - docker-compose-yml: contiene los contenedores
  - prometheus.yml:
  - Caddyfile: autenticación básica (usuario y contraseña) y proxy reverso a cadvisor y prometheus



```
:9090 {
    basicauth /* {
        {$ADMIN_USER} {$ADMIN_PASSWORD_HASH}
    }
    reverse_proxy prometheus:9090
}

:8080 {
    basicauth /* {
        {$ADMIN_USER} {$ADMIN_PASSWORD_HASH}
    }

    reverse_proxy cadvisor:8080
}
```

Monitorizar Docker

- Desplegar servicio de monitorización:

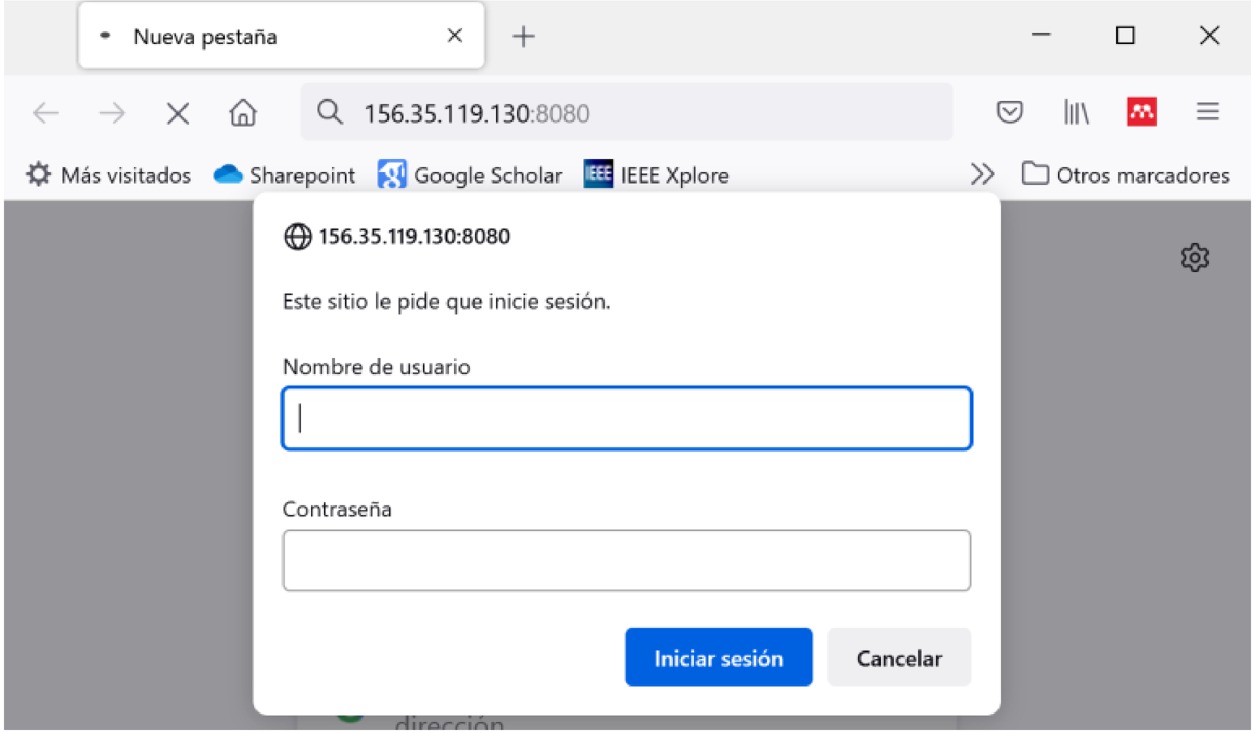


```
jesus@injtest:/disk0/monitorizar$ sudo docker-compose up --detach
Creating network "monitorizar_default" with the default driver
Creating volume "monitorizar_prometheus_data" with default driver
Creating cadvisor ... done
Creating caddy      ... done
Creating prometheus ... done
jesus@injtest:/disk0/monitorizar$
```

  - □ Crea red
  - □ Crea volumen para guardar los datos de la bd
  - □ Crea contenedores

Monitorizar Docker

- cAdvisor



156.35.119.130:8080

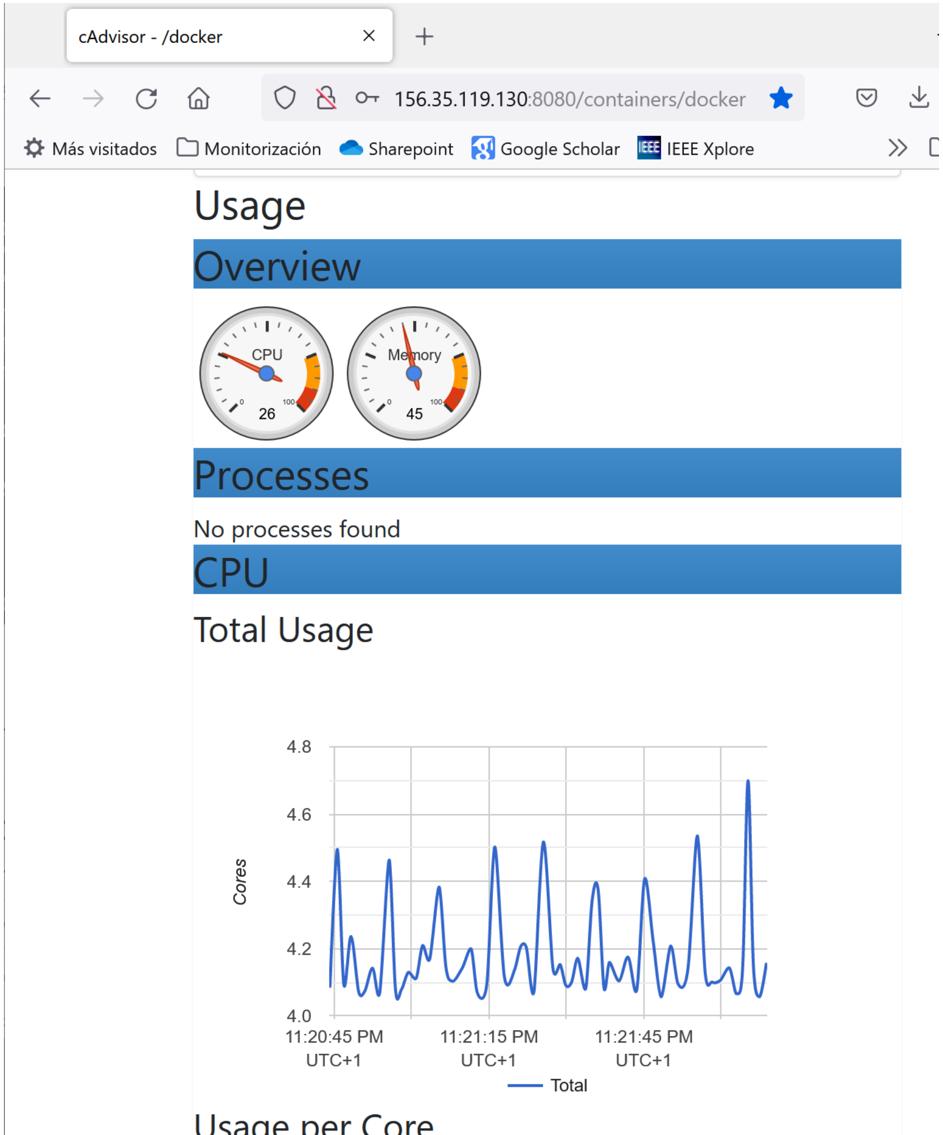
Este sitio le pide que inicie sesión.

Nombre de usuario

Contraseña

Monitorizar Docker

- cAdvisor



Usage

Overview

Processes

No processes found

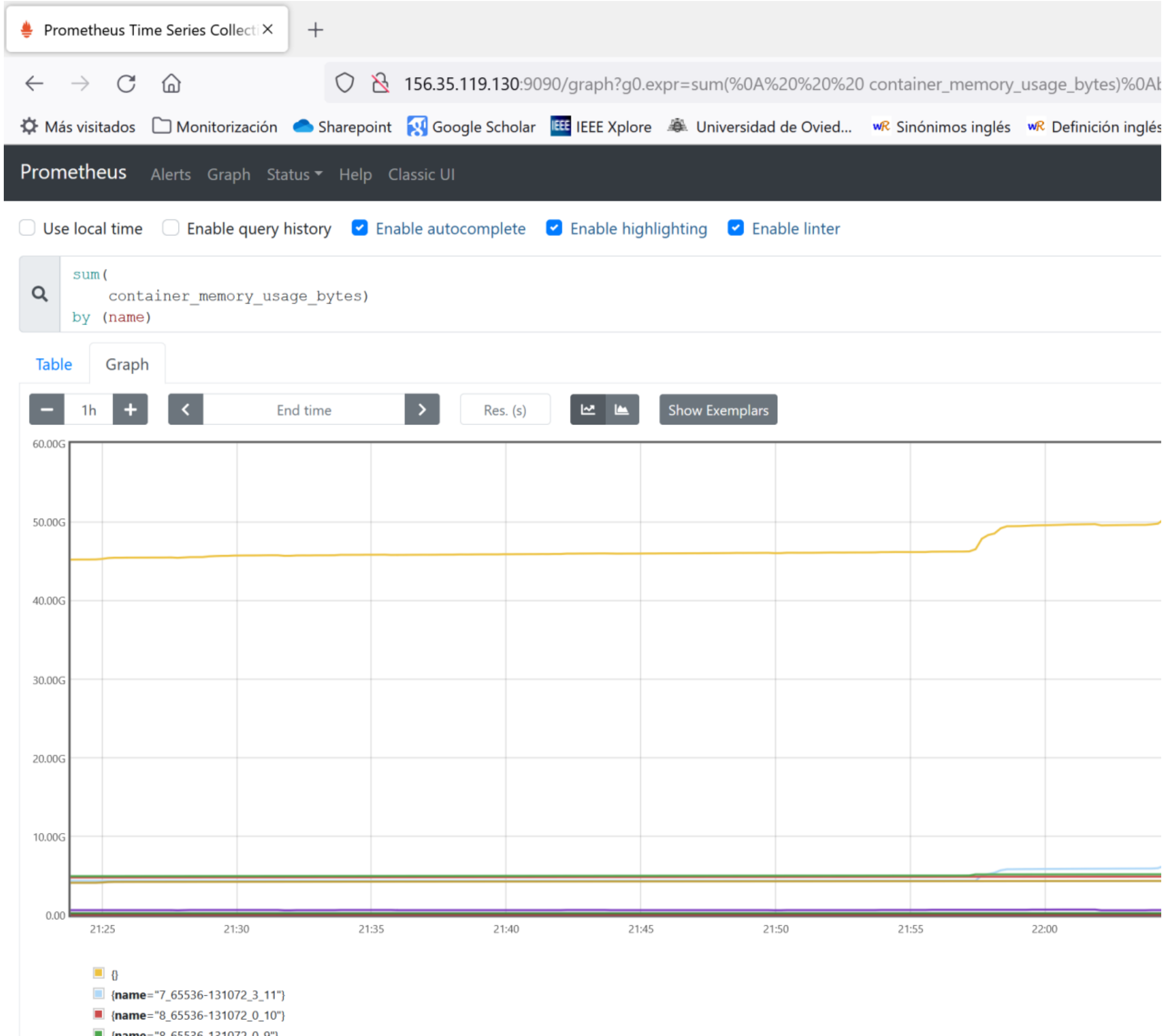
CPU

Total Usage

Usage per Core

Monitorizar Docker

- Prometheus:



□ Use local time □ Enable query history ☑ Enable autocomplete ☑ Enable highlighting ☑ Enable linter

```
sum(
    container_memory_usage_bytes)
by (name)
```

🟨 {}
🟦 {name="7_65536-131072_3_11"}
🟥 {name="8_65536-131072_0_10"}
🟩 {name="8_65536-131072_0_9"}

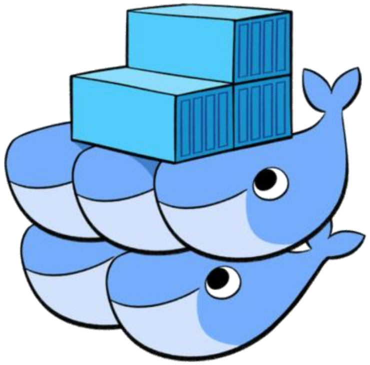
Data + Dev + Op

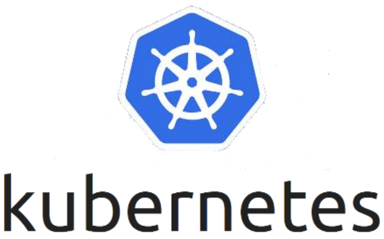
- Devops:
  - Desarrollo + Operaciones
  - Integración continua
  - Despliegue continuo
  - Entrega continua

- DataOps
  - Desarrollo + Operaciones + científicos/analistas de datos
  - Analítica continua

Orquestación de contenedores

- Organizar y dirigir cómo se despliegan los contenedores. Alternativas:





Jesús Morán y Cristian Augusto

Grupo de Investigación en Ingeniería del Software
http://giis.uniovi.es
Universidad de Oviedo