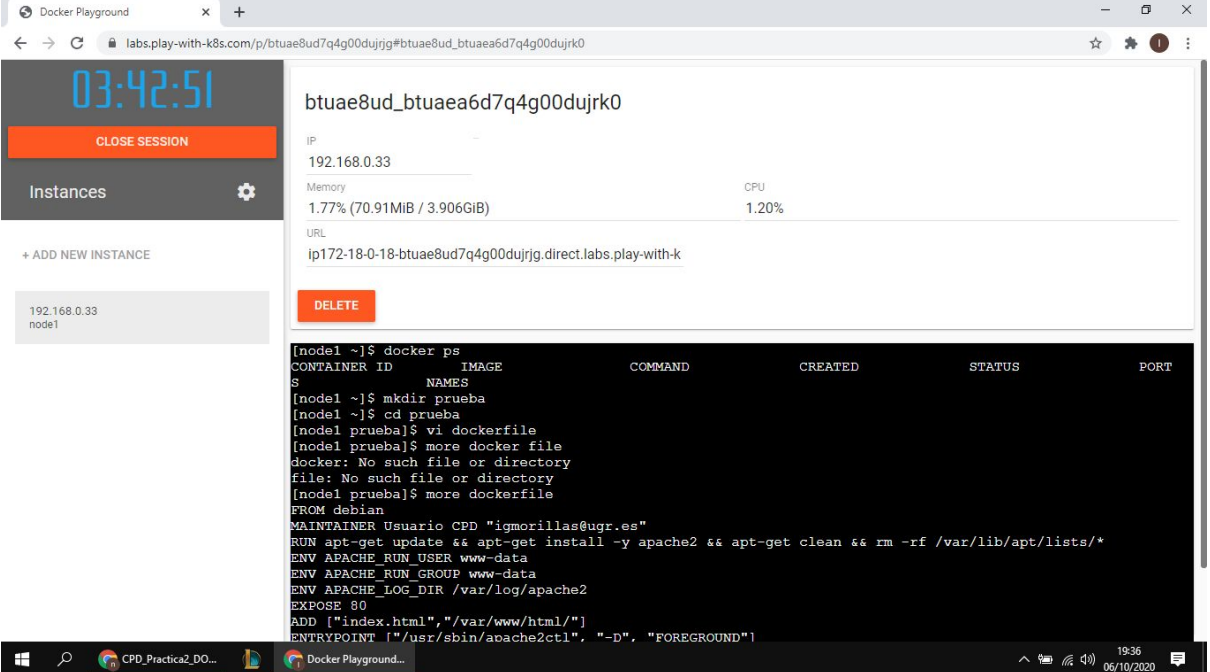


# CPD\_Practica\_2. Docker II

Crear una imagen personalizada de Docker con Apache de forma que cuando se acceda a <http://localhost:8888> aparezca vuestro nombre (editando el index.html) (según apartado I).

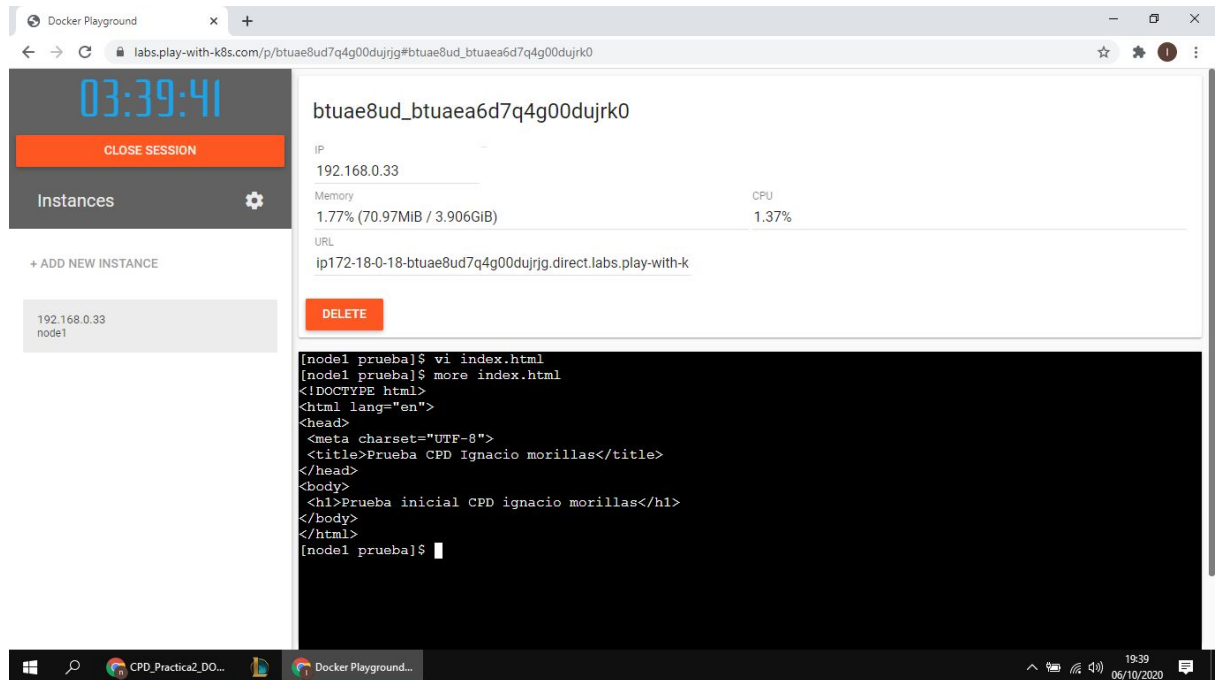
1. Creamos el directorio prueba y con mkdir.
2. creamos el fichero docker file dentro de prueba:



The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:42:51, a 'CLOSE SESSION' button, and an 'Instances' section with a '+ ADD NEW INSTANCE' button and a list of instances including '192.168.0.33 node1'. The main area displays details for a container named 'btuae8ud\_btuae6d7q4g00dujrk0'. The details include IP (192.168.0.33), Memory (1.77% (70.91MiB / 3.906GiB)), CPU (1.20%), and a URL (ip172-18-0-18-btuae8ud7q4g00dujrk.direct.labs.play-with-k). Below the details is a 'DELETE' button. At the bottom, a terminal window shows the following commands and output:

```
[node1 ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORT
S
[node1 ~]$ mkdir prueba
[node1 ~]$ cd prueba
[node1 prueba]$ vi dockerfile
[node1 prueba]$ more dockerfile
docker: No such file or directory
file: No such file or directory
[node1 prueba]$ more dockerfile
FROM debian
MAINTAINER Usuario CPD "igmorillas@ugr.es"
RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -rf /var/lib/apt/lists/*
ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
EXPOSE 80
ADD ["index.html", "/var/www/html/"]
ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

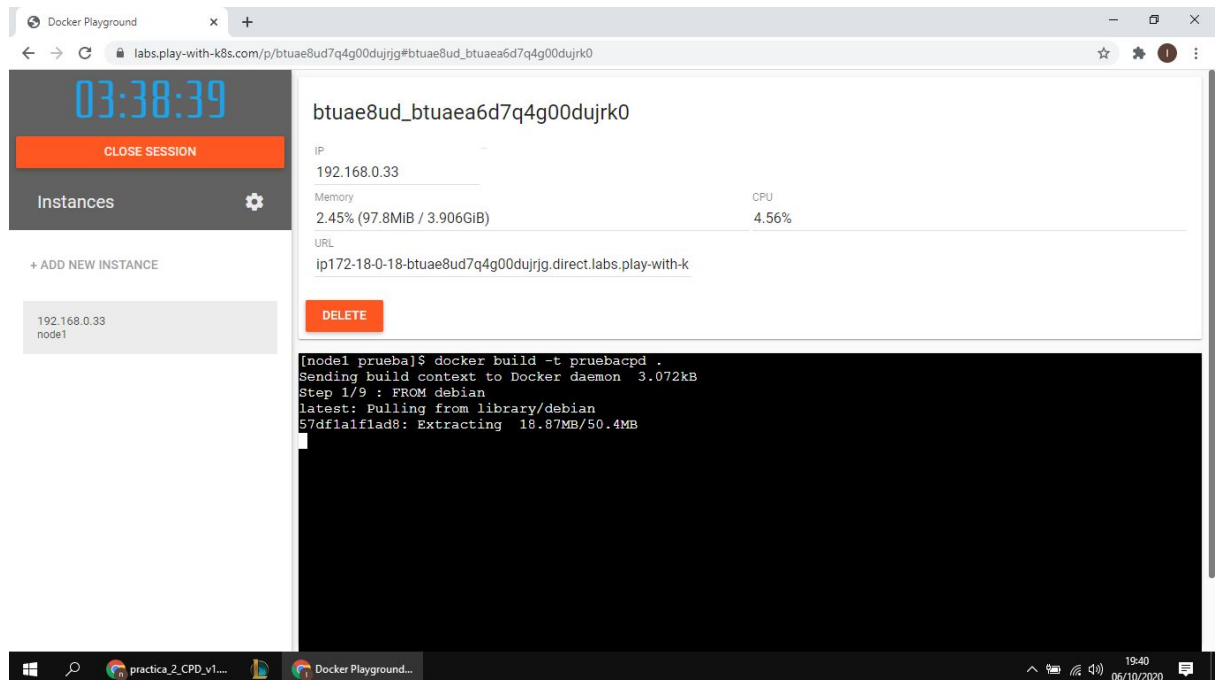
### 3. Creamos un fichero index.html:



Docker Playground interface showing a container named `btuae8ud_btuaea6d7q4g00dujrk0`. The container's IP is `192.168.0.33`, Memory is `1.77% (70.97MiB / 3.906GiB)`, and CPU is `1.37%`. The URL is `ip172-18-0-18-btuae8ud7q4g00dujrk0.direct.labs.play-with-k`. The terminal shows the creation of `index.html` with the following content:

```
[node1 prueba]$ vi index.html
[node1 prueba]$ more index.html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Prueba CPD Ignacio morillas</title>
</head>
<body>
<h1>Prueba inicial CPD ignacio morillas</h1>
</body>
</html>
[node1 prueba]$
```

### 4. Creamos nuestro contenedor:



Docker Playground interface showing a container named `btuae8ud_btuaea6d7q4g00dujrk0`. The container's IP is `192.168.0.33`, Memory is `2.45% (97.8MiB / 3.906GiB)`, and CPU is `4.56%`. The URL is `ip172-18-0-18-btuae8ud7q4g00dujrk0.direct.labs.play-with-k`. The terminal shows the execution of `docker build -t pruebapcd .`, which is pulling the latest image from library/debian.

```
[node1 prueba]$ docker build -t pruebapcd .
Sending build context to Docker daemon 3.072kB
Step 1/9 : FROM debian
latest: Pulling from library/debian
57df1a1flad8: Extracting 18.87MB/50.4MB
```

## Vemos que se ha hecho correctamente

The screenshot shows the Docker Playground web interface. On the left, there's a sidebar with a digital clock showing 03:34:57, a 'CLOSE SESSION' button, an 'Instances' section with a gear icon, and a '+ ADD NEW INSTANCE' button. Below this, a list of instances shows one instance with IP 192.168.0.33 and name 'node1'. The main area displays details for the selected instance: IP 192.168.0.33, Memory usage at 18.82% (752.7MiB / 3.906GiB), CPU usage at 12.50%, and a URL. A 'DELETE' button is present. At the bottom, a terminal window shows the command `docker images` being executed, resulting in a table of installed images:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
pruebacpd	latest	76ec4b9dbaab	2 minutes ago	226MB
debian	latest	f6dcff9b59af	3 weeks ago	114MB

### 5. Ejecutamos el docker:

**`docker run --name x1 -p8080:80 -d pruebacpd`**

Accedemos a 8080 y vemos que se ha modificado correctamente index.html y que se ha creado todo correctamente.

The screenshot shows a web browser with two tabs: 'Docker Playground' and 'Prueba CPD Ignacio morillas'. The active tab shows a warning 'No es seguro' and the URL `ip172-18-0-18-btuaesud7q4g00dujrg-8080.direct.labs.play-with-k8s.com`. The browser interface includes back, forward, and refresh buttons, as well as star, settings, and notification icons.

### Prueba inicial CPD ignacio morillas

The screenshot shows a Windows taskbar with three open applications: 'CPD\_Practica2\_DO...', 'Prueba CPD Ignaci...', and 'Docker Playground...'. The system clock in the bottom right corner shows 19:52 on 06/10/2020.

A partir de los pasos explicados en el apartado II subir la imagen a hub.docker.com e indicar el nombre de la imagen creada.

1. Nos logueamos con docker login
2. Con tag le ponemos un alias al contenedor acorde al repositorio en la web
3. subimos la imagen con docker push "igmorillas/cpd\_practica2\_2020"

The screenshot shows a Docker Playground interface. On the left, there's a sidebar with a timer at 03:48:40, a 'CLOSE SESSION' button, and an 'Instances' section. The main area is a terminal window with the following output:

```
192.168.0.23 8080
Memory 28.85% (1.127GiB / 3.906GiB) CPU 0.97%
URL ip172-18-0-30-btubt1mj2b70009avug0.direct.labs.play-with-
DELETE

[node1 pruebapcd]$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: igmorillas
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

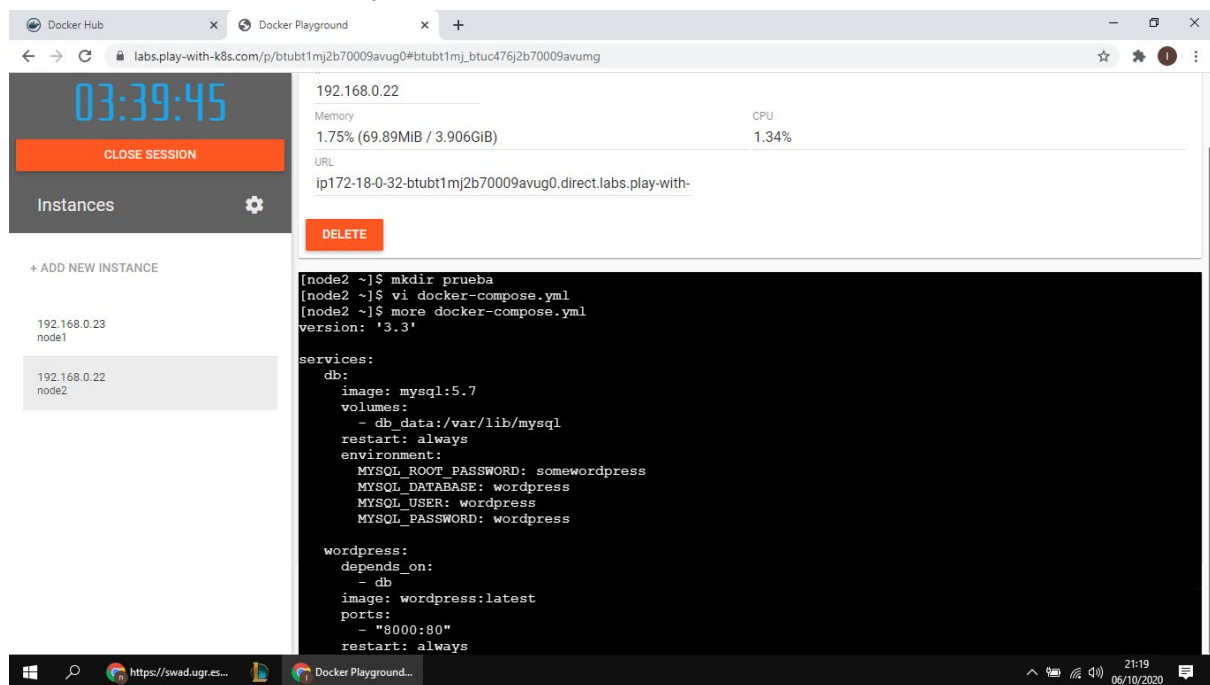
Login Succeeded
[node1 pruebapcd]$ docker tag pruebapcd "igmorillas/cpd_practica2_2020"
[node1 pruebapcd]$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
igmorillas/cpd_practica2_2020  latest     0a68552bcb8   2 minutes ago  226MB
pruebapcd            latest     0a68552bcb8   2 minutes ago  226MB
debian              latest     f6dcff9b59af  3 weeks ago   114MB
[node1 pruebapcd]$ docker push igmorillas/cpd_practica2_2020
The push refers to repository [docker.io/igmorillas/cpd_practica2_2020]
9552b4d16f03: Pushed
21b29b06e25a: Pushed
4ef54afed780: Layer already exists
latest: digest: sha256:e8a5f594ab8a72d8a33b331f6db51c18264cd343d396adfebd23e732a1ff7d03 size: 948
[node1 pruebapcd]$
```

## 4. Vemos que se ha producido una actualización del repositorio

The screenshot shows the Docker Hub repository page for 'igmorillas / cpd\_practica2\_2020'. The page has a blue header with the Docker Hub logo and search bar. Below the header, there's a navigation bar with tabs: General, Tags, Builds, Timeline, Collaborators, Webhooks, and Settings. The 'General' tab is selected. The main content area shows the repository name, a description (which is empty), and the last pushed time (a few seconds ago). On the right, there's a 'Docker commands' section with a 'Public View' button and a command to push a new tag: `docker push igmorillas/cpd_practica2_2020:tagname`. Below the 'General' tab, there's a 'Tags' section showing a table with columns: TAG, OS, and PUSHED. The table has one row: 'latest', 'linux', and 'a few seconds ago'. There's also a 'Recent builds' section with a link to see build results.

Según el apartado III, una vez desplegado el servidor Wordpress, editar la página principal para que aparezca el nombre del usuario y realizar una captura de pantalla.

### 1. Creamos el archivo .yaml



The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:39:45, a 'CLOSE SESSION' button, and a list of instances. The main area displays a terminal window with the following commands and output:

```
[node2 ~]$ mkdir prueba
[node2 ~]$ vi docker-compose.yml
[node2 ~]$ more docker-compose.yml
version: '3.3'

services:
  db:
    image: mysql:5.7
    volumes:
      - db data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: somewordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress

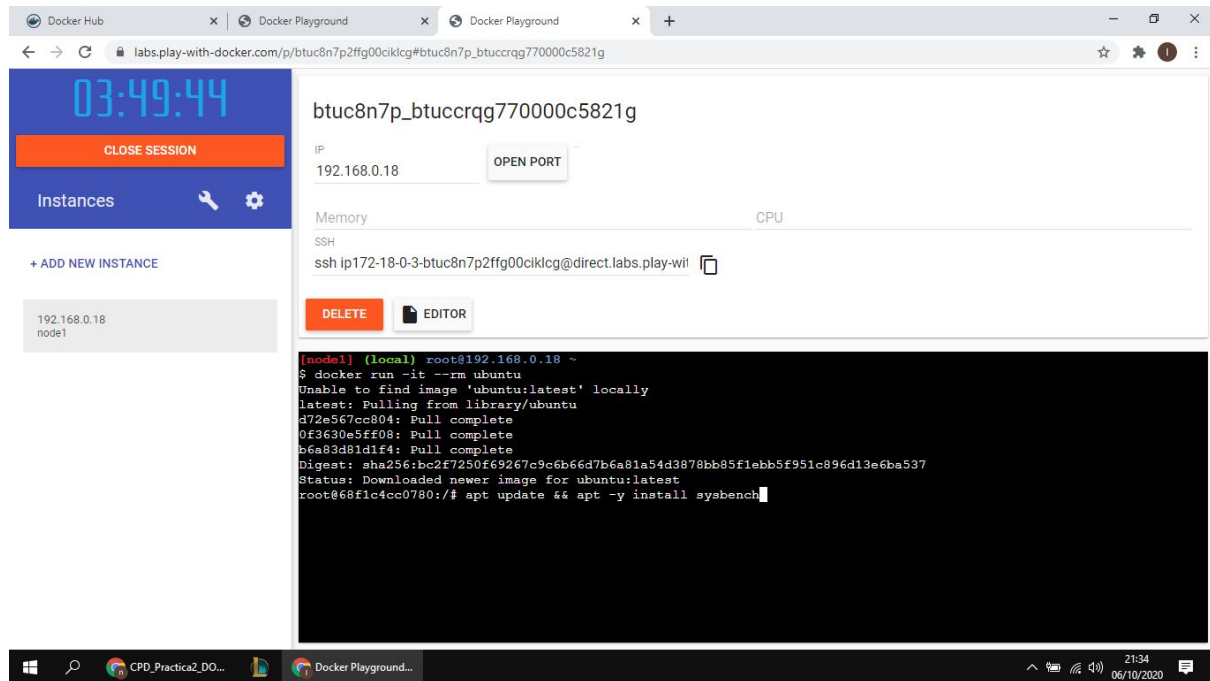
  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "8000:80"
    restart: always
```

### 2. Accediendo a 8080 y tras registrarnos y demás finalmente llegamos a la parte final ya con la alteración hecha



The screenshot shows the WordPress customization interface for the 'CPD Ignacio' theme. The left sidebar contains a list of customization options: 'Tema activo' (Twenty Twenty), 'Identidad del sitio', 'Colores', 'Opciones del tema', 'Plantilla de portada', 'Imagen de fondo', 'Menús', and 'Widgets'. The main area displays the site's header with the title 'CPD IGNACIO 2020' and the author 'Por igmorillas'. The page content is currently empty, showing 'SIN CATEGORÍA' at the top and bottom.

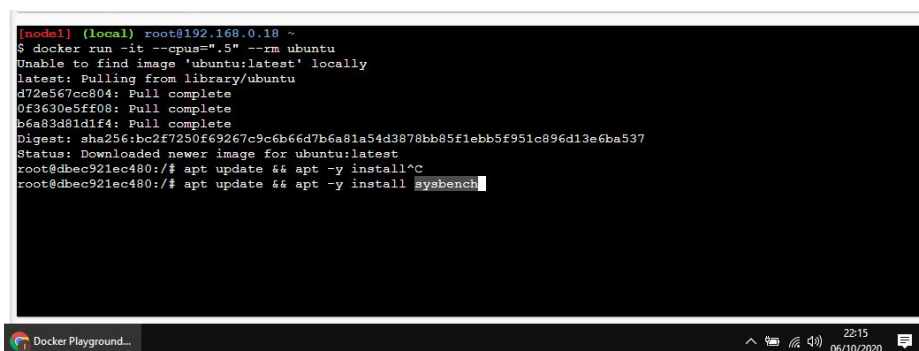
Siguiendo los pasos del apartado IV, ejecute el test que permita evaluar el tiempo de ejecución de un benchmark, y comprobar cómo podemos aumentar o reducir la CPU dedicada y por tanto el tiempo de ejecución. Prepare un contenedor con dicho experimento, súbalo a [hub.docker.com](https://hub.docker.com) e indique en el documento los pasos que realiza para el experimento y los tiempos de ejecución obtenidos.



## PASOS:

1. `docker run -it --cpus=".<numerocpu>" --rm ubuntu` (indicamos el numero de cpu)
2. `apt update && apt -y install sysbench`(actualizamos y instalamos el benchmark)
3. `sysbench --test=cpu --cpu-max-prime=20000 run` (Ejecutamos el benchmark)

## EJEMPLO CON 5



Obtenemos dicho resultado



```

CPU speed:
  events per second:    150.24

General statistics:
  total time:           10.0016s
  total number of events: 1503

Latency (ms):
  min:                  2.59
  avg:                   6.62
  max:                   62.13
  95th percentile:      51.02
  sum:                   9943.54

Threads fairness:
  events (avg/stddev):   1503.0000/0.00
  execution time (avg/stddev): 9.9435/0.00

```

## EJEMPLO CON 3 CPU

```

root@dbec921ec480:/# exit
exit
[node1] (local) root@192.168.0.18 ~
$ docker run -it --cpus="3" --rm ubuntu
root@e84b8ec18a1b:/# apt update && apt -y install sysbench
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [369 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [114 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [75.9 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [1169 B]
Get:8 http://archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:10 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [21.6 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [708 kB]

```

Obtenemos dicho resultado

```

CPU speed:
  events per second:    85.03

General statistics:
  total time:           10.0405s
  total number of events: 854

Latency (ms):
  min:                  2.64
  avg:                   11.75
  max:                   78.83
  95th percentile:      74.46
  sum:                   10036.04

Threads fairness:
  events (avg/stddev):   854.0000/0.00
  execution time (avg/stddev): 10.0360/0.00

```

## Conclusión

Cuanto más cpus contamos más eventos se podrán realizar en un periodo de tiempo