Tarefas UD05
Bloque 03

Administración de sistemas operativos

Unidade Didáctica 05:
Integración de sistemas operativos en redes libres e privativas

Nome: Rubén Apelidos: Rey Feal

Data:

Índice

Tarefa 1. Lista comandos impresión

Completa as seguintes táboas. Engade as liñas que precises.

1.1. GNU/Linux (incluídos os de CUPS)

Comando	Función	Exemplo

1.2. Windows

Comando	Función	Exemplo

Tarefa 2. CUPS

Bótalle unha ollada ao seguinte documento web:

https://weblinus.com/como-instalar-un-servidor-de-impresion-cups-en-debian-y-derivados/

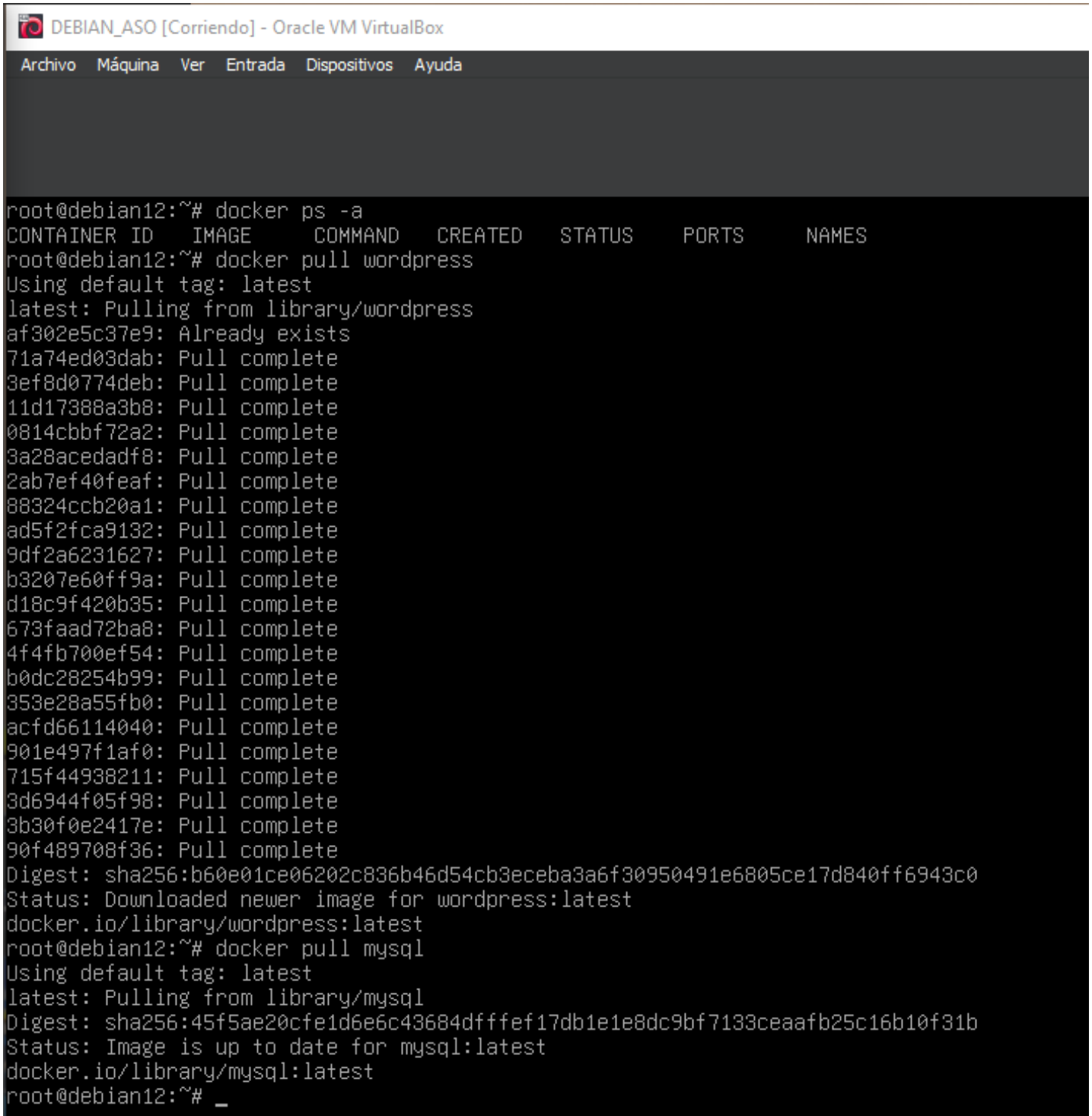
Empregando CUPS, instala unha impresora virtual PDF. Unha vez configurada, facer unha demostración e o resultado de impresión

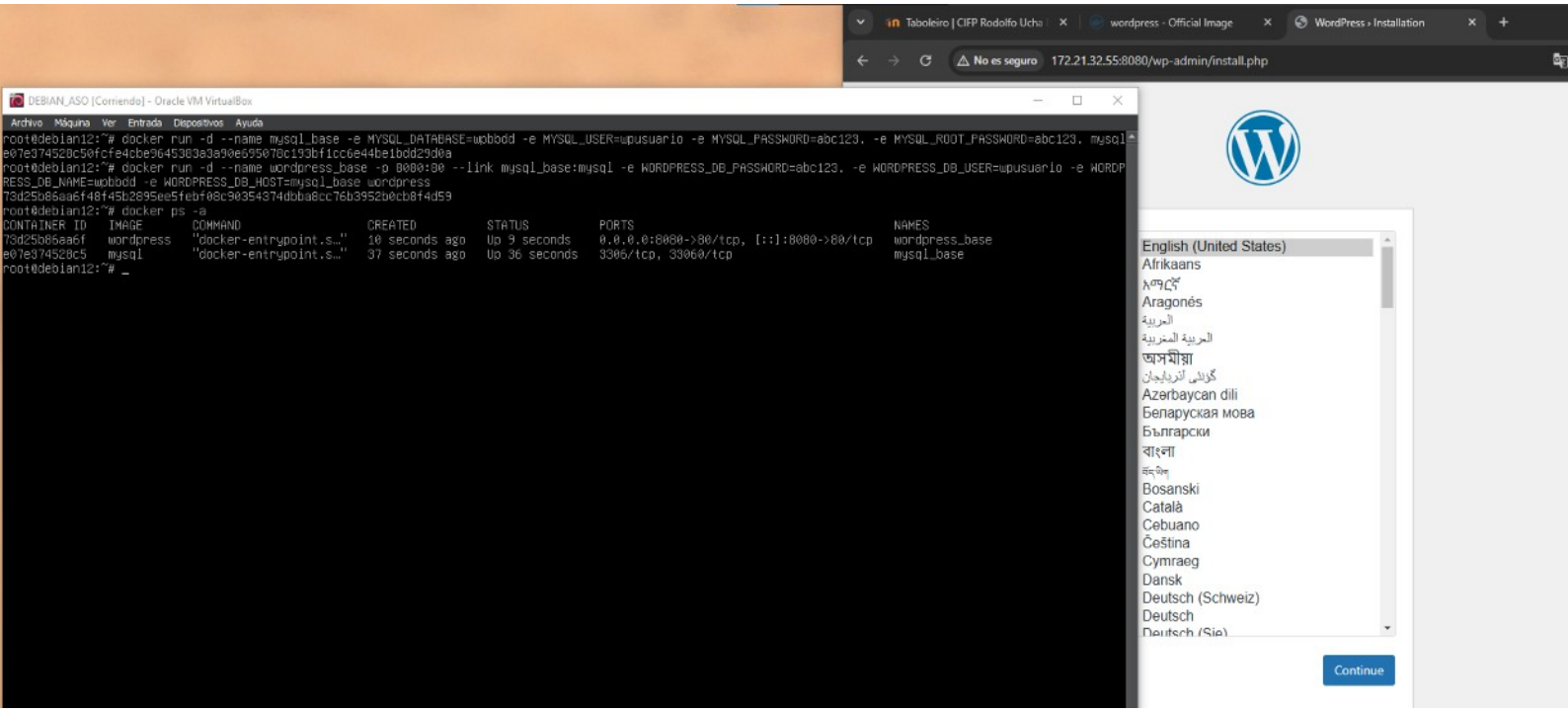
Tarefa 3. Repaso Docker

3.1. Manexo de imaxes, contedores e orquestación

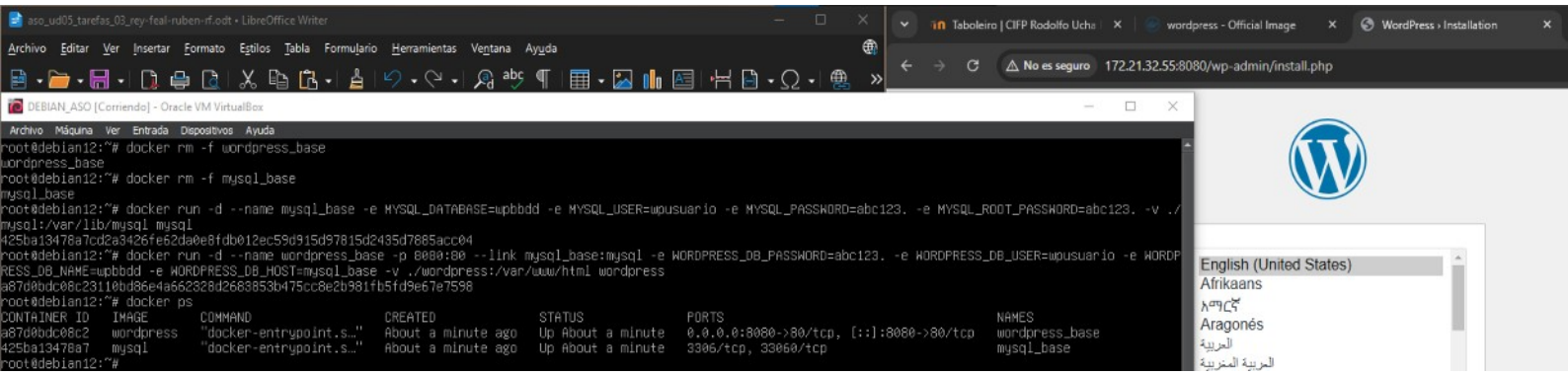
Completa os seguintes apartados facendo as capturas de imaxe necesarias que demostren o correcto funcionamento dos contedores.

1. Descarga as imaxes de Wordpress e MySQL. Revisa a documentación oficial de Wordpress en Docker Hub. Executa os dous contedores definindo as variables de entorno de forma que o nome da base dataso sexa «`wpbbdd`», o usuario «`wpusuario`» con contrasinal «`abc123.`». Recorda mapear o porto 80, por exemplo ao 8080. Comprobar que é accesible. Para conectar os dous contedores, usar a opción `–link`.

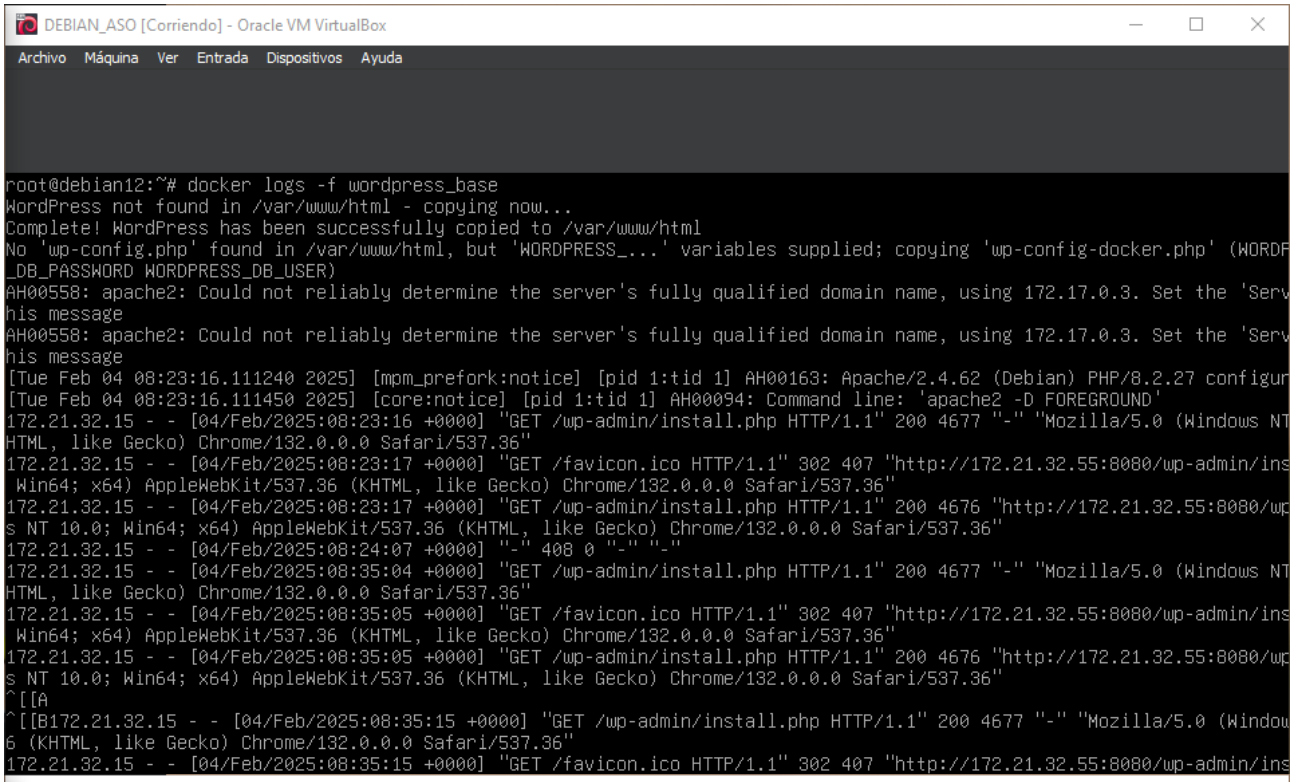


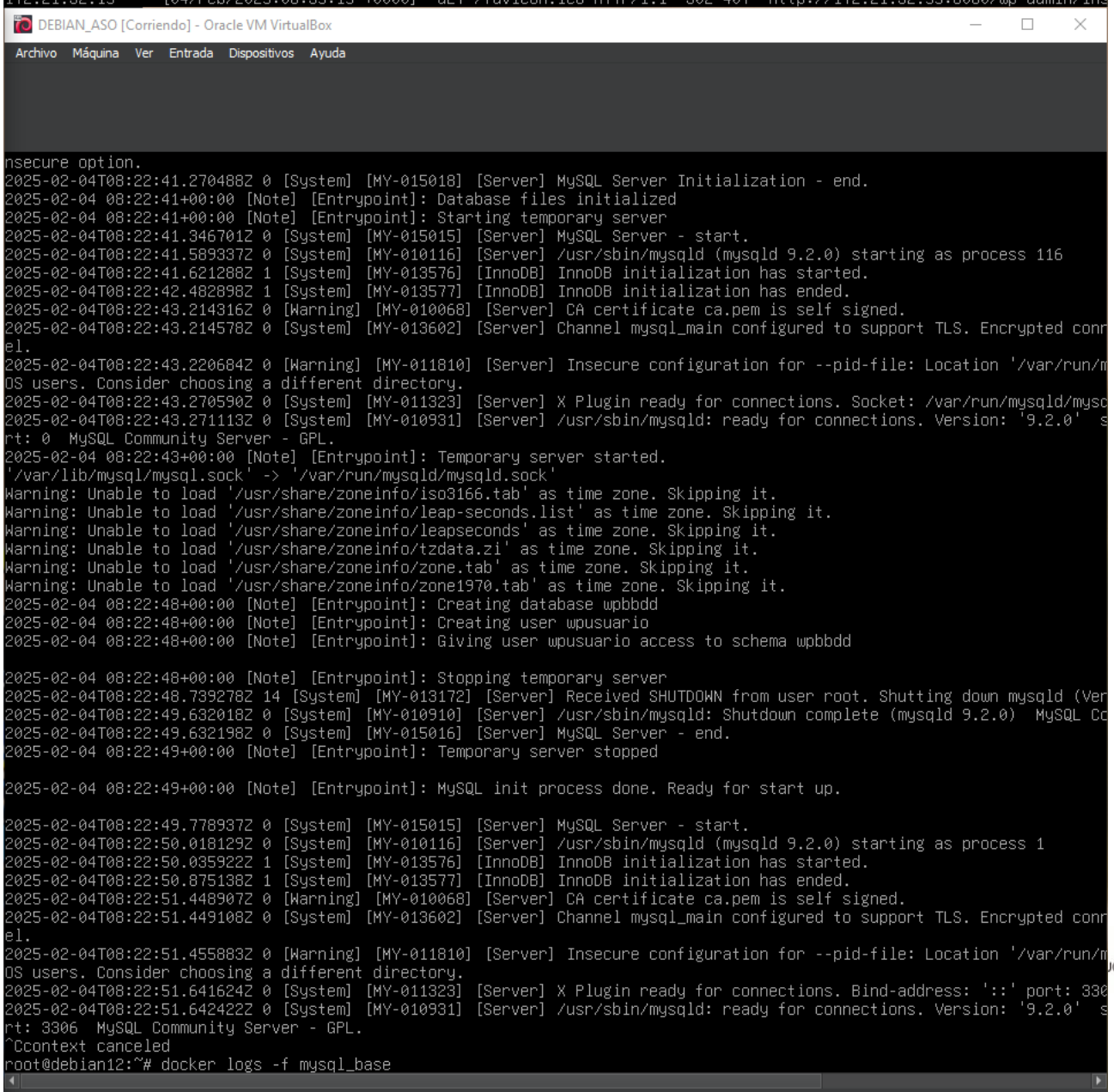


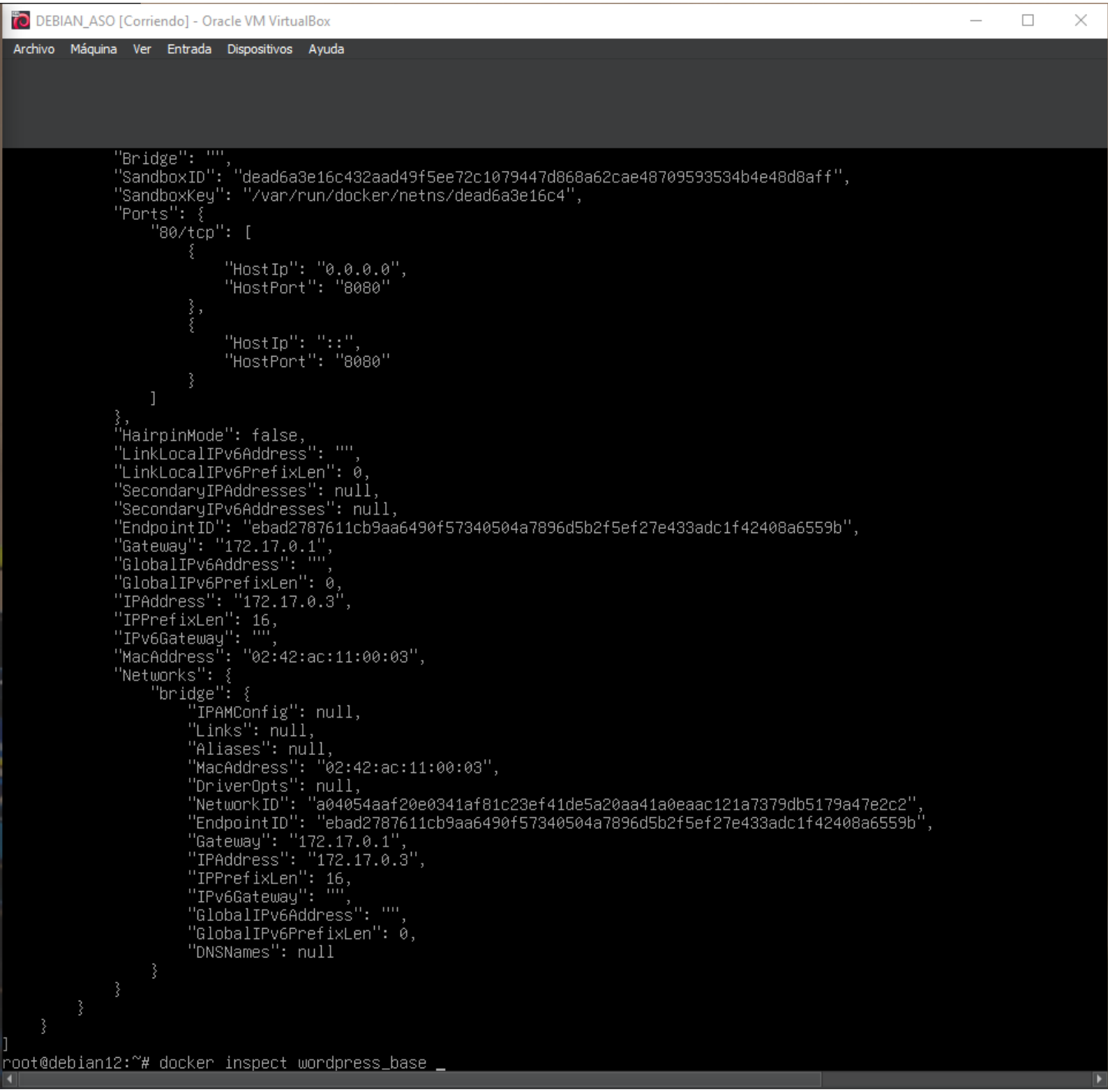
2. Agora borra os dous contedores e volve a executalos con persistencia de datos, tanto da base de datos (`/var/lib/mysql`) como dos ficheiros web de Wordpress (`/var/www/html`).

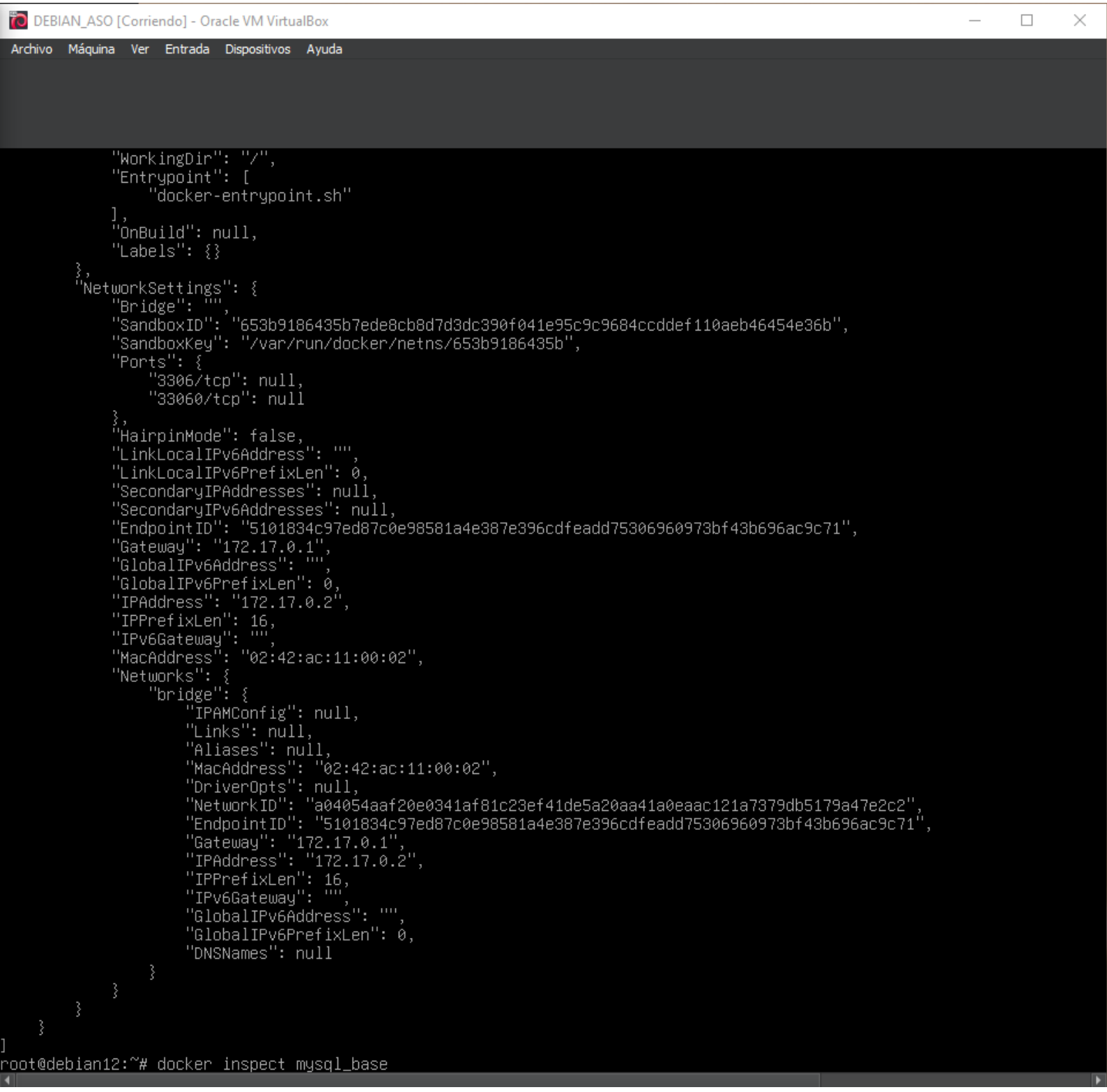


3. Comproba os logs de cada contedor e inspecciona a súa configuración.






DEBIAN_ASO [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
"Bridge": "",
"SandboxID": "dead6a3e16c432aad49f5ee72c1079447d868a62cae48709593534b4e48d8aff",
"SandboxKey": "/var/run/docker/netns/dead6a3e16c4",
"Ports": {
"80/tcp": [
{
"HostIp": "0.0.0.0",
"HostPort": "8080"
},
{
"HostIp": "::",
"HostPort": "8080"
}
]
},
"HairpinMode": false,
"LinkLocalIPv6Address": "",
"LinkLocalIPv6PrefixLen": 0,
"SecondaryIPAddresses": null,
"SecondaryIPv6Addresses": null,
"EndpointID": "ebad2787611cb9aa6490f57340504a7896d5b2f5ef27e433adc1f42408a6559b",
"Gateway": "172.17.0.1",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"IPAddress": "172.17.0.3",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"MacAddress": "02:42:ac:11:00:03",
"Networks": {
"bridge": {
"IPAMConfig": null,
"Links": null,
"Aliases": null,
"MacAddress": "02:42:ac:11:00:03",
"DriverOpts": null,
"NetworkID": "a04054aaf20e0341af81c23ef41de5a20aa41a0eaac121a7379db5179a47e2c2",
"EndpointID": "ebad2787611cb9aa6490f57340504a7896d5b2f5ef27e433adc1f42408a6559b",
"Gateway": "172.17.0.1",
"IPAddress": "172.17.0.3",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"DNSNames": null
}
}
}
}
]
root@debian12:~# docker inspect wordpress_base _

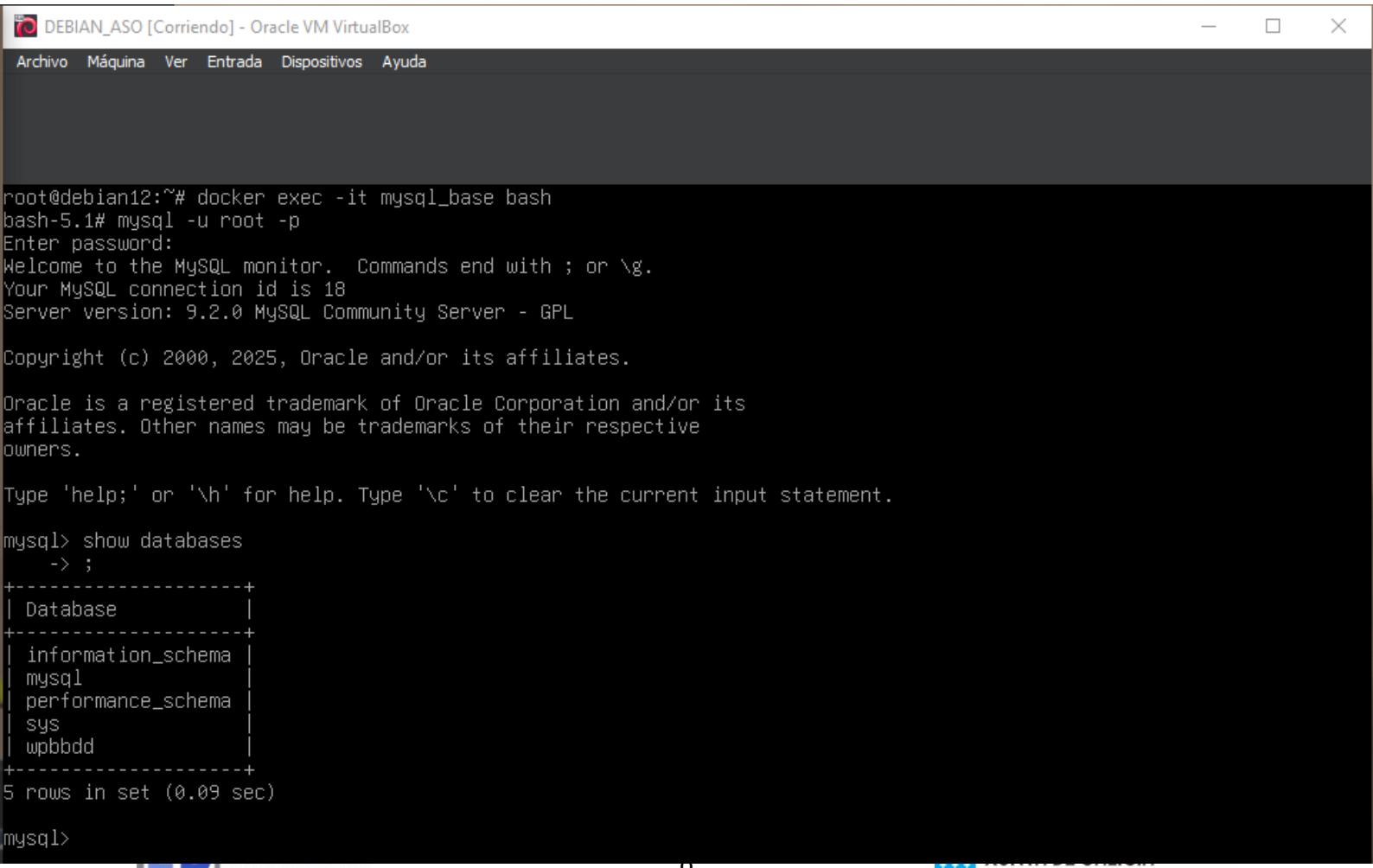

DEBIAN_ASO [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
"WorkingDir": "/",
"Entrypoint": [
"docker-entrypoint.sh"
],
"OnBuild": null,
"Labels": {}
},
"NetworkSettings": {
"Bridge": "",
"SandboxID": "653b9186435b7ede8cb8d7d3dc390f041e95c9c9684ccddef110aeb46454e36b",
"SandboxKey": "/var/run/docker/netns/653b9186435b",
"Ports": {
"3306/tcp": null,
"33060/tcp": null
},
"HairpinMode": false,
"LinkLocalIPv6Address": "",
"LinkLocalIPv6PrefixLen": 0,
"SecondaryIPAddresses": null,
"SecondaryIPv6Addresses": null,
"EndpointID": "5101834c97ed87c0e98581a4e387e396cdfeadd75306960973bf43b696ac9c71",
"Gateway": "172.17.0.1",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"IPAddress": "172.17.0.2",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"MacAddress": "02:42:ac:11:00:02",
"Networks": {
"bridge": {
"IPAMConfig": null,
"Links": null,
"Aliases": null,
"MacAddress": "02:42:ac:11:00:02",
"DriverOpts": null,
"NetworkID": "a04054aaf20e0341af81c23ef41de5a20aa41a0eaac121a7379db5179a47e2c2",
"EndpointID": "5101834c97ed87c0e98581a4e387e396cdfeadd75306960973bf43b696ac9c71",
"Gateway": "172.17.0.1",
"IPAddress": "172.17.0.2",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"DNSNames": null
}
}
}
}
]
root@debian12:~# docker inspect mysql_base

4. Mostra o contido do ficheiro `/etc/hosts` do contedor de Wordpress.

```
root@debian12:~# docker exec -it wordpress_base /bin/bash
root@a87d0bdc08c2:/var/www/html# cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      mysql 425ba13478a7 mysql_base
172.17.0.3      a87d0bdc08c2
root@a87d0bdc08c2:/var/www/html#
```

5. Accede ao contedor de MySQL. Conéctate ao xestor de base de datos a través de terminal e completa os seguintes comandos SQL:

    1. Mostra sa bases de datos dispoñibles



2. Mostra os permisos do usuario creado para Worrpress.

```
mysql> SHOW GRANTS FOR wpusuario;
+---------------------------------------------------------+
| Grants for wpusuario@%                                  |
+---------------------------------------------------------+
| GRANT USAGE ON *.* TO `wpusuario`@`%`                   |
| GRANT ALL PRIVILEGES ON `wpbbdd`.* TO `wpusuario`@`%`   |
+---------------------------------------------------------+
2 rows in set (0.00 sec)

mysql>
```

3. Usando a base de datos de Wordpress, mostra as súas táboas.

```
mysql> use wpbbdd
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----------------------+
| Tables_in_wpbbdd      |
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
12 rows in set (0.00 sec)

mysql>
```

4. Mostra a descrición da táboa `wp_users`.

```
mysql> DESCRIBE wp_users;
+---------------------+---------------------+------+-----+---------------------+----------------+
| Field               | Type                | Null | Key | Default             | Extra          |
+---------------------+---------------------+------+-----+---------------------+----------------+
| ID                  | bigint unsigned     | NO   | PRI | NULL                | auto_increment |
| user_login          | varchar(60)         | NO   | MUL |                     |                |
| user_pass           | varchar(255)        | NO   |     |                     |                |
| user_nicename       | varchar(50)         | NO   | MUL |                     |                |
| user_email          | varchar(100)        | NO   | MUL |                     |                |
| user_url            | varchar(100)        | NO   |     |                     |                |
| user_registered     | datetime            | NO   |     | 0000-00-00 00:00:00 |                |
| user_activation_key | varchar(255)        | NO   |     |                     |                |
| user_status         | int                 | NO   |     | 0                   |                |
| display_name        | varchar(250)        | NO   |     |                     |                |
+---------------------+---------------------+------+-----+---------------------+----------------+
10 rows in set (0.00 sec)

mysql> _
```

5. Mostra o contido da táboa `wp_users`.

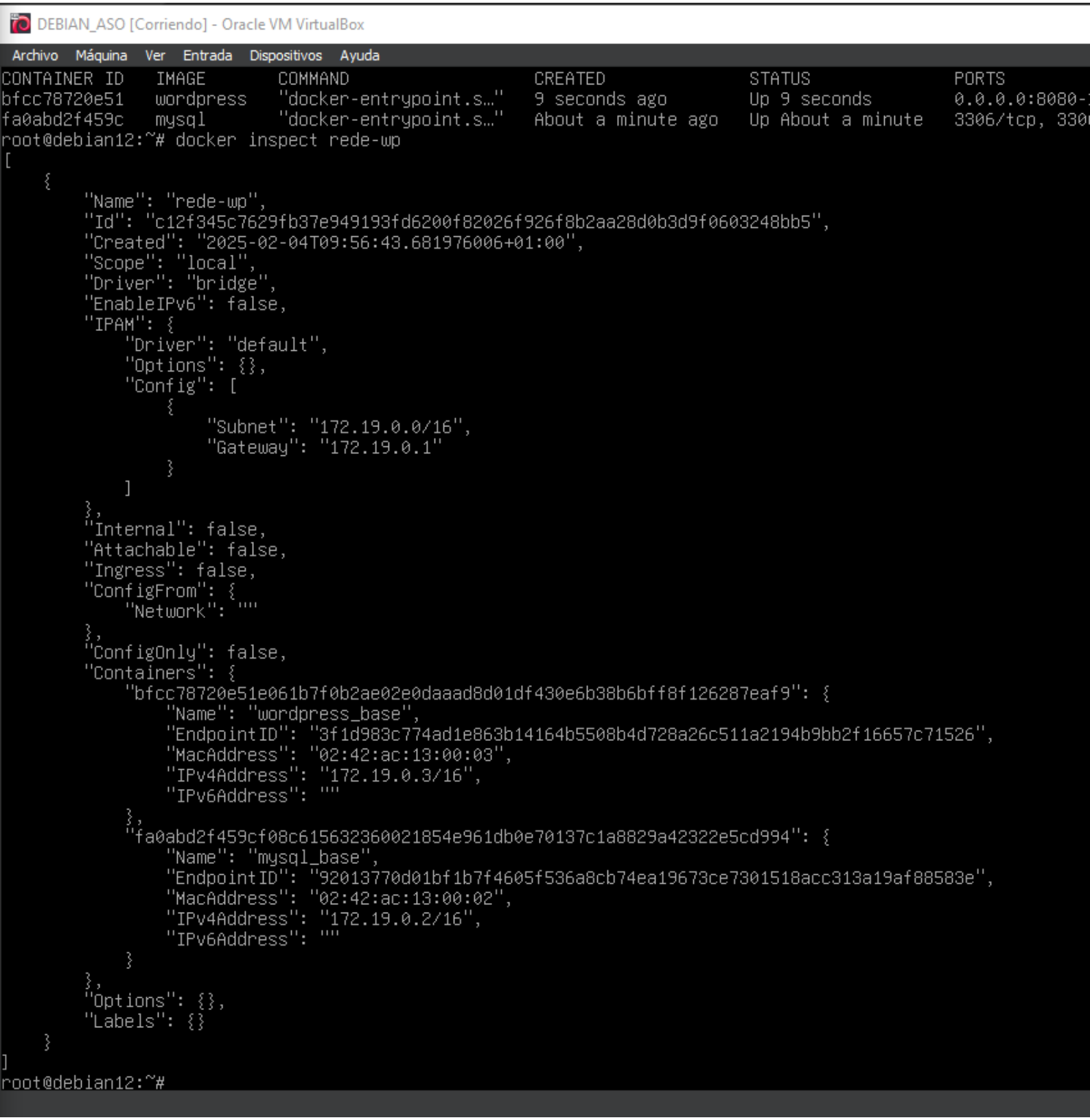
```
mysql> select * from wp_users;
+----+------------+------------------------------------+--------------+----------------------+----------------------
-------+-------------+--------------+
| ID | user_login | user_pass                          | user_nicename | user_email           | user_url
on_key | user_status | display_name |
+----+------------+------------------------------------+--------------+----------------------+----------------------
-------+-------------+--------------+
|  1 | rubenrf    | $P$BSzh2zpkc0C126Q.OXyRUn1OqRJnJH1 | rubenrf       | rubenaso@wordpress.com | http://172.21.32.55:80
       |           0 | rubenrf      |
+----+------------+------------------------------------+--------------+----------------------+----------------------
-------+-------------+--------------+
1 row in set (0.00 sec)

mysql> _
```

6. Agora repetir apartados anteriores definindo as carpetas persistentes `wp-html` (para os ficheiros de Wordpress) e `wp-db`,no home do usuario que ten permiso de execución dos contedores, e creando unha rede propia (en vez de usar –`link`) chamada `rede-wp`. Mapea o porto 8081 ao porto 80 do contedor Wordpress.

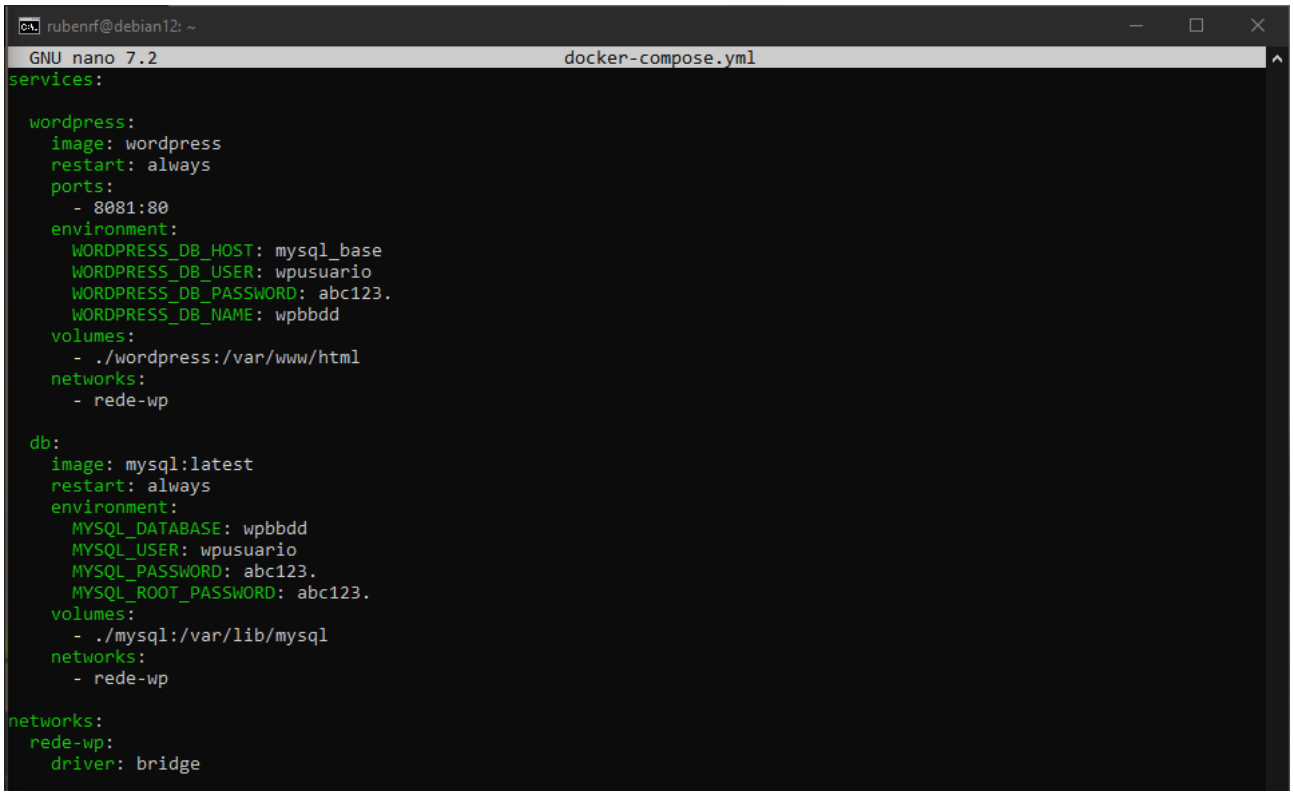
```
root@debian12:~# docker network create rede-wp
c12f345c7629fb37e949193fd6200f82026f926f8b2aa28d0b3d9f0603248bb5
root@debian12:~# docker inspect rede-wp
[
    {
        "Name": "rede-wp",
        "Id": "c12f345c7629fb37e949193fd6200f82026f926f8b2aa28d0b3d9f0603248bb5",
        "Created": "2025-02-04T09:56:43.681976006+01:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.19.0.0/16",
                    "Gateway": "172.19.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {},
        "Labels": {}
    }
]
root@debian12:~# _
```

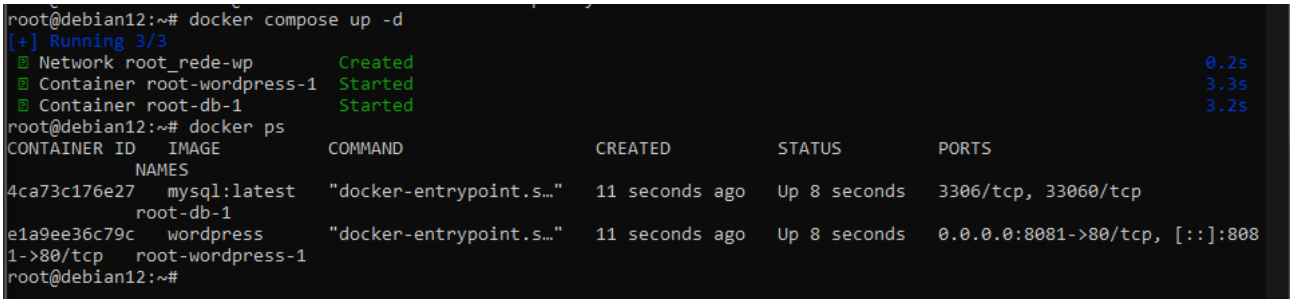
```
root@debian12:~# docker rm -f mysql_base
mysql_base
root@debian12:~# docker rm -f wordpress_base
wordpress_base
root@debian12:~# docker run -d --name mysql_base --network rede-wp -e MYSQL_DATABASE=wpbbdd -e MYSQL_USER=wpusuario -e MYSQL_PASSWORD=abc123. -e MYSQL_ROOT_PASS
WORD=abc123. -v ./mysql:/var/lib/mysql mysql
fa0abd2f459cf08c615632360021854e961db0e70137c1a8829a42322e5cd994
root@debian12:~# docker run -d --name wordpress_base -p 8080:80 --network rede-wp -e WORDPRESS_DB_PASSWORD=abc123. -e WORDPRESS_DB_USER=wpusuario -e WORDPRESS_D
B_NAME=wpbbdd -e WORDPRESS_DB_HOST=mysql_base -v ./wordpress:/var/www/html wordpress
bfcc78720e51e061b7f0b2ae02e0daaad8d01df430e6b38b6bff8f126287eaf9
root@debian12:~# docker ps
CONTAINER ID   IMAGE       COMMAND                  CREATED             STATUS             PORTS                                   NAMES
bfcc78720e51   wordpress   "docker-entrypoint.s…"   9 seconds ago       Up 9 seconds       0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   wordpress_base
fa0abd2f459c   mysql       "docker-entrypoint.s…"   About a minute ago   Up About a minute   3306/tcp, 33060/tcp                       mysql_base
root@debian12:~# _
```

7. Inspecciona a nova rede e mostra que enderezo IP lle asociou a cada contedor.



8. Repite o paso 6 con Docker Compose creando `docker-compose.yml` correspondente. Deben crearse a rede e tamén o mapeando ás carpetas locais igual que no apartado 6. Executa o orquestador de contedores co comando `docker compose`.





9. Inspecciona a nova rede e mostra que enderezo IP lle asociou a cada contedor.

```
root@debian12:~# docker network inspect root_rede-wp
[
    {
        "Name": "root_rede-wp",
        "Id": "ad539569a97e99b6f5d74495cb22b69fdecc8090b5adb9642a06af3e1ba9149b",
        "Created": "2025-02-05T13:45:57.842236165+01:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.22.0.0/16",
                    "Gateway": "172.22.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "45c4ce1f50e83cfe0fc9efff9d48fb665839b87a146513dde4c0a6aa2513f996": {
                "Name": "root-db-1",
                "EndpointID": "9bd4f2d48ebe0284e63a2cbff1afa600568742e7479c22ef67e3fffce00ab0c4",
                "MacAddress": "02:42:ac:16:00:02",
                "IPv4Address": "172.22.0.2/16",
                "IPv6Address": ""
            },
            "86657613decbae388abcdf5eab8b1adc943b98a2f62077ffc0a3360af4638bda": {
                "Name": "root-wordpress-1",
                "EndpointID": "db06a2c16356929e6cdbaf9b1f6d64df9d7fcc6726a72ed4fc32810b1853e597",
                "MacAddress": "02:42:ac:16:00:03",
                "IPv4Address": "172.22.0.3/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {
            "com.docker.compose.config-hash": "c193021ee73413e4205e33faecd8a8def200e0c2e11ffe430f869be18d196083",
            "com.docker.compose.network": "rede-wp",
            "com.docker.compose.project": "root",
            "com.docker.compose.version": "2.32.3"
        }
    }
]
root@debian12:~#
```

10. Mostra os contedores executados co orquestador

```
rubenrf@debian12: ~
]
root@debian12:~# docker compose ps
NAME              IMAGE          COMMAND                  SERVICE     CREATED          STATUS          PORTS
root-db-1         mysql:latest   "docker-entrypoint.s…"   db          10 minutes ago   Up 10 minutes   3306/tcp, 330
60/tcp
root-wordpress-1  wordpress      "docker-entrypoint.s…"   wordpress   10 minutes ago   Up 10 minutes   0.0.0.0:8081-
>80/tcp, [::]:8081->80/tcp
root@debian12:~#
root@debian12:~#
```

11. Para os contedores executados co orquestador.

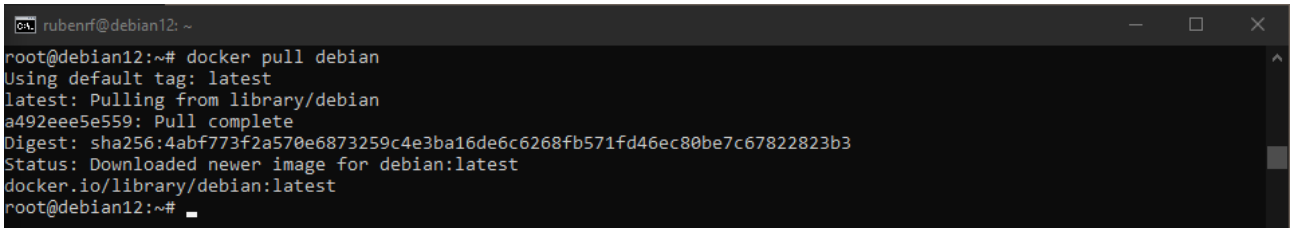
```
rubenrf@debian12: ~
[+] Stopping 2/2 docker compose stop
 ⠿ Container root-wordpress-1  Stopped                                    1.8s
 ⠿ Container root-db-1         Stopped                                    2.7s
root@debian12:~#
```

3.2. Creación de imaxes

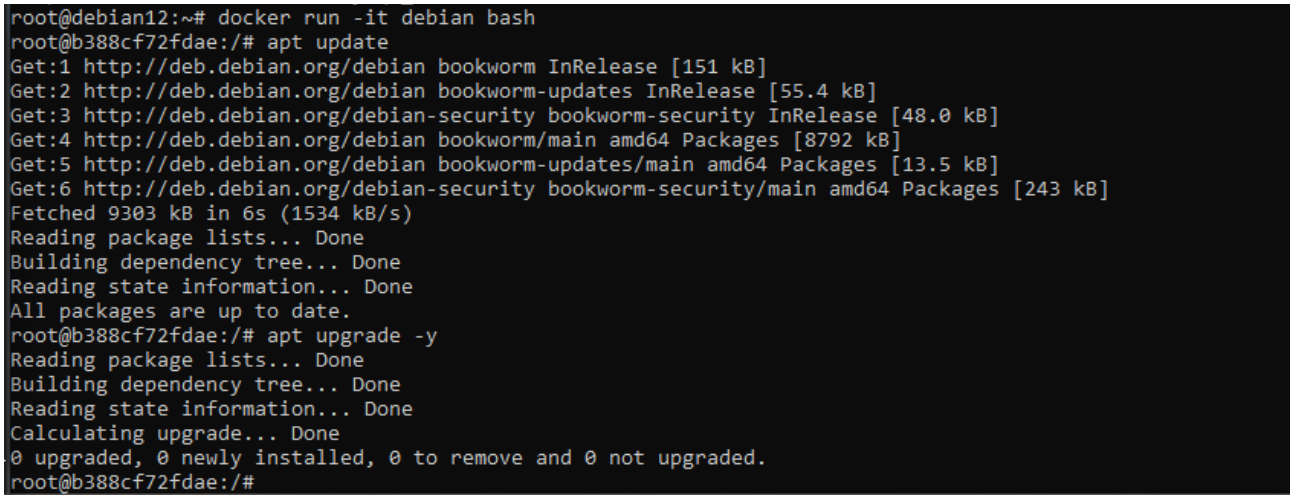
Completa os seguintes apartados facendo as capturas de imaxe necesarias que demostren o correcto funcionamento dos contedores.

3.2.1. A partir dun contedor

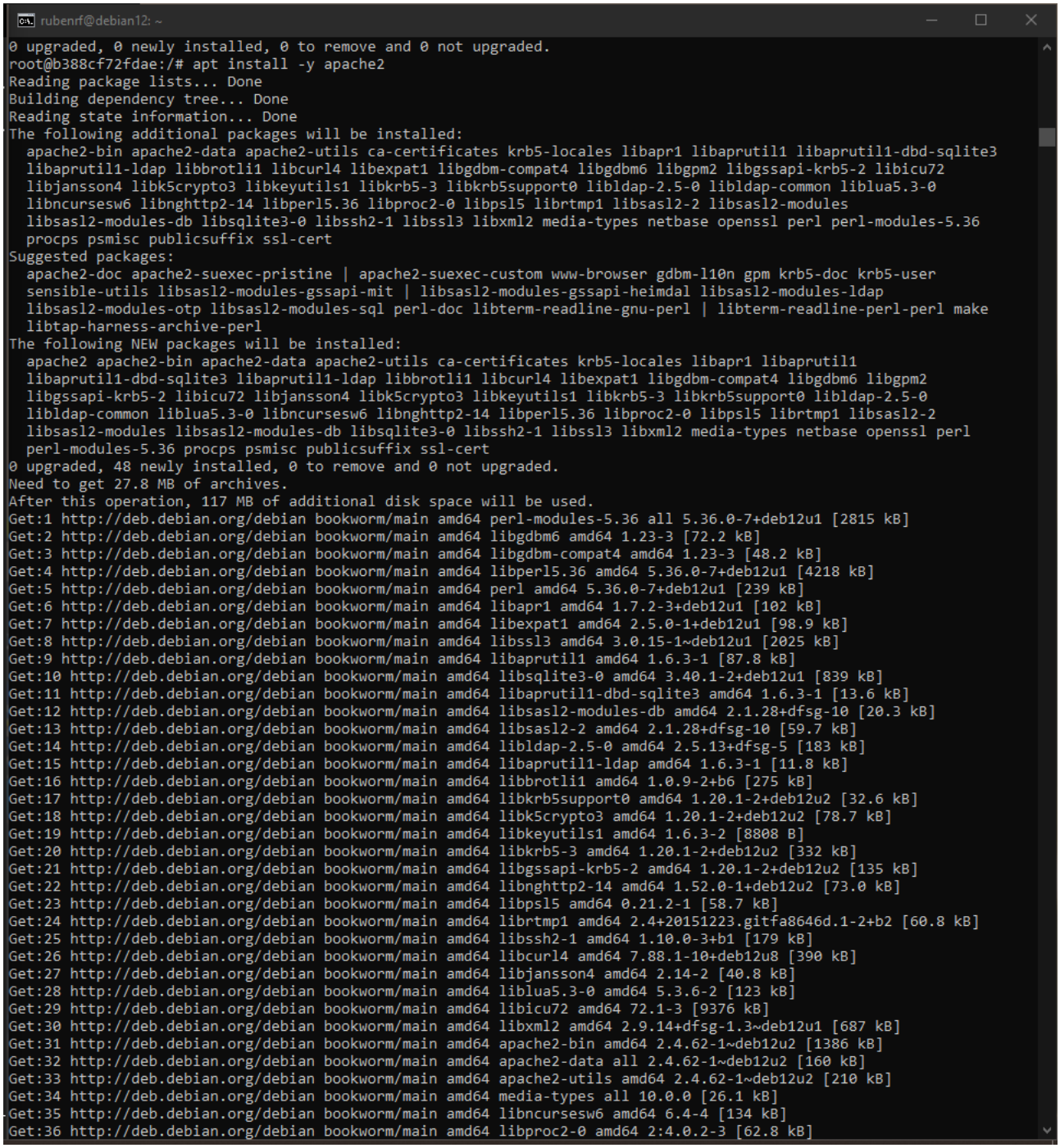
1. Descarga a última imaxe de Debian.



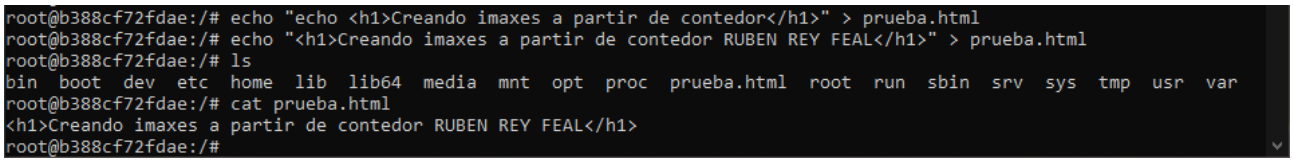
2. Lanza no contedor os comandos apt para actualizar os repositorios e logo actualizar todos os paquetes sen que pida confirmación.



3. Instala o paquete `apache2` sen que pida confirmación.



4. Crea un ficheiro HTML simple que poña como título de nivel 1 «Creando imaxes a partir de contedor» e engade o teu nome e apelidos.



5. Copia ao contedor ese ficheiro html co nome ao cartafol `/var/www/html/` do contedor. Comproba que é accesible dende o navegador.

```
root@b388cf72fdae:/# cp prueba.html /var/www/html/
```

6. Inicia o servidor apache2 co comando `bash -c "apache2ctl -D FOREGROUND"`.

```
root@debian12:~# docker exec loving_jepsen bash -c "apache2ctl -D FOREGROUND"
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'S
erverName' directive globally to suppress this message
httpd (pid 26) already running
root@debian12:~#
```

7. Garda os cambios nunha nova imaxe co formato `teunomeiniciais/meuapache:v1` (cambiando `teunomeiniciais` polo teu nome e iniciais dos teu apelidos. Por exemplo: elenafc/meuapache:v1)

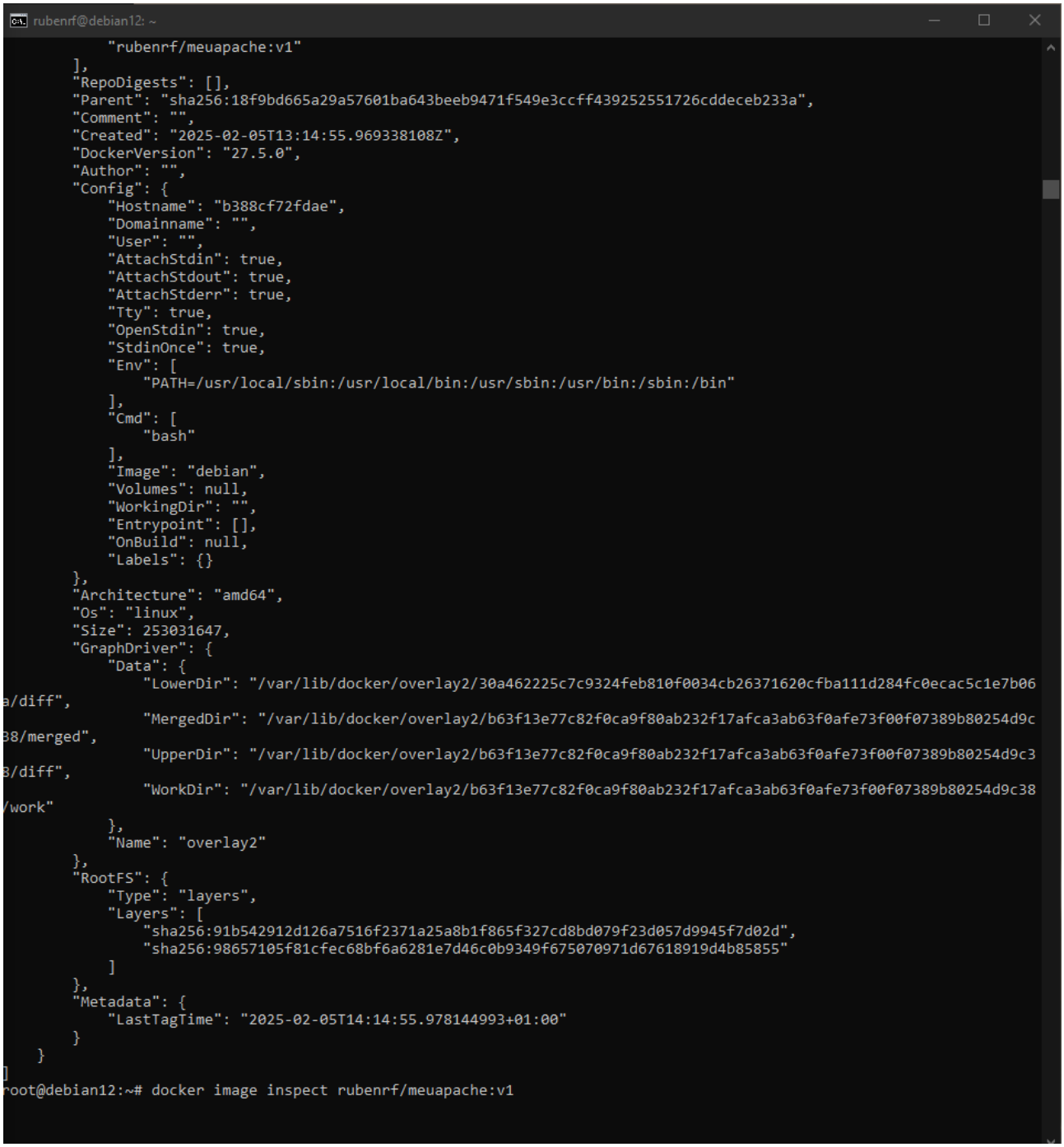
```
root@debian12:~# docker commit loving_jepsen rubenrf/meuapache:v1
sha256:c41e973f52078fc48a3cf3180634d3da306a4c9162df3146a9deee4e406f32ce
root@debian12:~#
```

```
root@debian12:~# docker images
REPOSITORY                TAG       IMAGE ID       CREATED          SIZE
rubenrf/meuapache         v1        c41e973f5207   39 seconds ago   253MB
debian                    latest    18f9bd665a29   2 days ago       117MB
personalizado_nginx       latest    168d5c990769   7 days ago       192MB
httpd                     latest    f7d8bafbd9a9   12 days ago      148MB
mysql                     latest    a52cba19e8cc   13 days ago      797MB
nginx                     prueba    a0d9e3b3dcc2   2 weeks ago      192MB
web_server-nginx          latest    58d3f6196d31   2 weeks ago      192MB
node                      latest    89871f29e084   2 weeks ago      1.12GB
redis                     latest    4075a3f8c3f8   4 weeks ago      117MB
portainer/portainer-ce    2.21.5    0c03664af9ed   6 weeks ago      308MB
bash                      latest    2a658e2e2bab   8 weeks ago      14.5MB
mongo                     latest    f08e39122805   2 months ago     855MB
nginx                     latest    9bea9f2796e2   2 months ago     192MB
postgres                  latest    9a0ce6be5dd4   2 months ago     435MB
wordpress                 latest    c012b71a41fc   2 months ago     701MB
busybox                   latest    af4709625109   4 months ago     4.27MB
nginx                     1.26.2    0dcfd986e814   5 months ago     188MB
hello-world               latest    d2c94e258dcb   21 months ago    13.3kB
root@debian12:~#
```

8. Elimina o contedor actual.

```
root@debian12:~# docker ps
CONTAINER ID   IMAGE       COMMAND                  CREATED         STATUS         PORTS
          NAMES
b388cf72fdae   debian      "bash"                   14 minutes ago  Up 4 minutes
          loving_jepsen
bfcc78720e51   wordpress   "docker-entrypoint.s…"   28 hours ago    Up 26 minutes  0.0.0.0:8080->80/tcp, [::]:8080-
>80/tcp   wordpress_base
root@debian12:~# docker rm -f loving_jepsen
loving_jepsen
root@debian12:~#
```

9. Inspecciona a imaxe.



10. Revisa as capas da nova imaxe (comando `history`).

```
root@debian12:~# docker history rubenrf/meuapache:v1
IMAGE          CREATED         CREATED BY                                      SIZE      COMMENT
c41e973f5207   3 minutes ago   bash                                            137MB
18f9bd665a29   2 days ago      # debian.sh --arch 'amd64' out/ 'bookworm' '…   117MB     debuerreotype 0.15
root@debian12:~#
```

11. Crea un novo contedor mapeando os portos ao porto 8084. Comproba que o ficheiro web creado anteriormente segue accesible.

```
root@debian12:~# docker run -d --name miapache -p 8084:80 rubenrf/meuapache:v1
89c2459d7579f237a2793e109753d76b16c755475e5dc3c1b82a7314cf539125
root@debian12:~#
```

3.2.2. Usando un Dockerfile

1. Crea un cartafol chamado `novocontedor`. Accede a ese cartafol. Crea un ficheiro HTML simple que poña como título de nivel 1 «Creando imaxes con **Dockerfile**» e engade o teu nome e apelidos.

```
root@debian12:~/novocontedor# nano index.html
root@debian12:~/novocontedor# cat index.html
<h1>Creando imaxes con Dockerfile RUBEN REY FEAL </h1>
root@debian12:~/novocontedor#
```

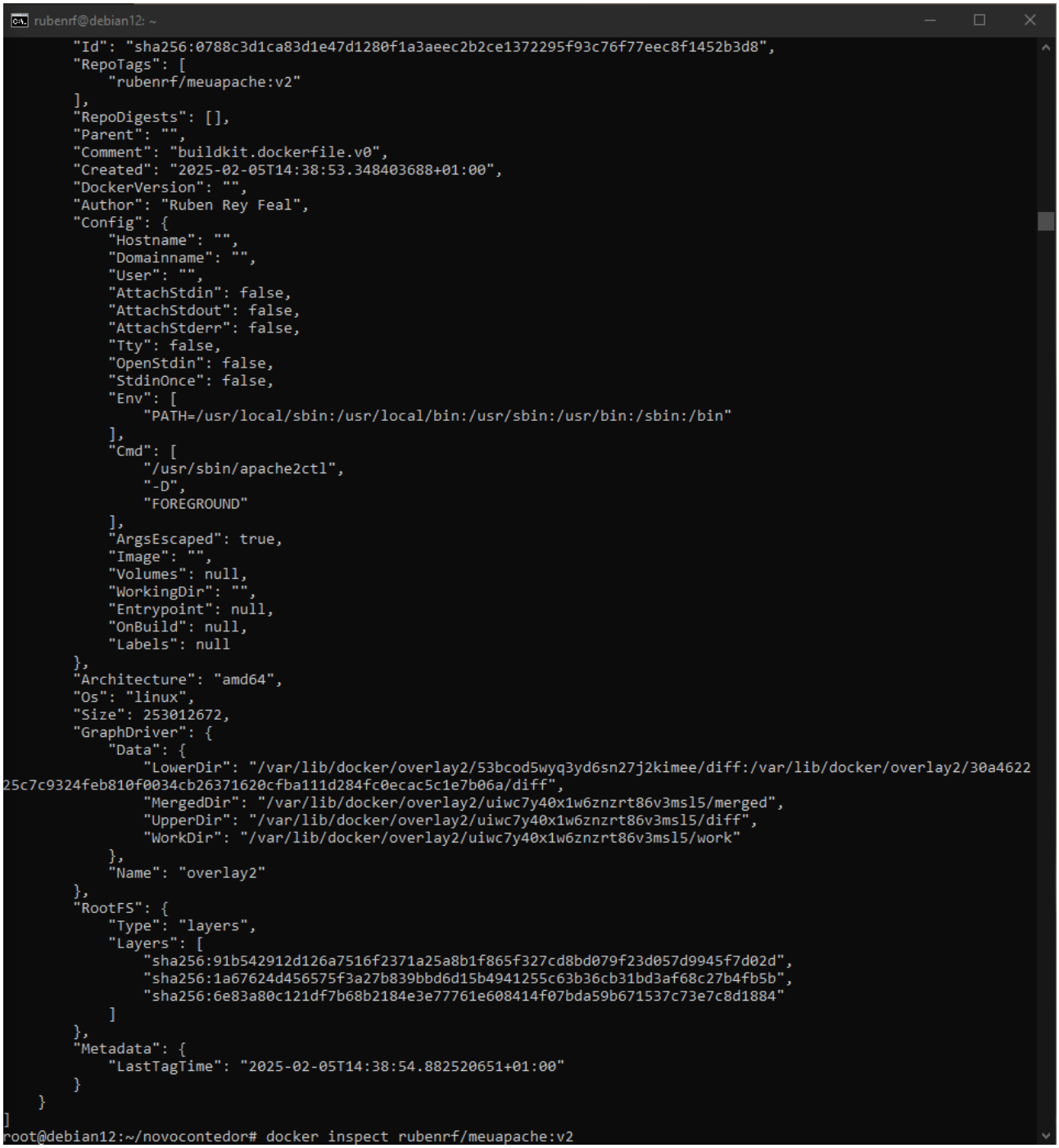
2. Crea un *docker file* que reproduza os apartados anteriores. Deberá ter as instrucións: FROM, MAINTAINER, RUN, COPY e CMD.

```
rubenrf@debian12: ~
  GNU nano 7.2                                  Dockerfile *
FROM debian
MAINTAINER Ruben Rey Feal
RUN apt update && apt upgrade -y && apt install apache2 -y
COPY index.html /var/www/html
CMD ["/usr/sbin/apache2ctl","-D","FOREGROUND"]
```

3. Crea a nova imaxe co comando `docker build` co nome `teunomeiniciais/meuapache:v2` (cambiando `teunomeiniciais` polo teu nome e iniciais dos teu apelidos. Por exemplo: elenafc/meuapache:v2).

```
root@debian12:~/novocontedor# docker build -t rubenrf/meuapache:v2 .
[+] Building 46.5s (8/8) FINISHED                                                docker:default
 => [internal] load build definition from Dockerfile                                      0.1s
 => => transferring dockerfile: 213B                                                      0.0s
 => [internal] load metadata for docker.io/library/debian:latest                          0.0s
 => [internal] load .dockerignore                                                         0.0s
 => => transferring context: 2B                                                           0.0s
 => [1/3] FROM docker.io/library/debian:latest                                            0.1s
 => [internal] load build context                                                         0.2s
 => => transferring context: 92B                                                          0.0s
 => [2/3] RUN apt update && apt upgrade -y && apt install apache2 -y                     44.2s
 => [3/3] COPY index.html /var/www/html                                                   0.1s
 => exporting to image                                                                    1.6s
 => => exporting layers                                                                   1.5s
 => => writing image sha256:0788c3d1ca83d1e47d1280f1a3aeec2b2ce1372295f93c76f77eec8f1452b3d8  0.0s
 => => naming to docker.io/rubenrf/meuapache:v2                                           0.0s

 1 warning found (use docker --debug to expand):
 - MaintainerDeprecated: Maintainer instruction is deprecated in favor of using label (line 2)
root@debian12:~/novocontedor#
```

4. Inspecciona a nova imaxe.



5. Revisa as capas da nova imaxe.

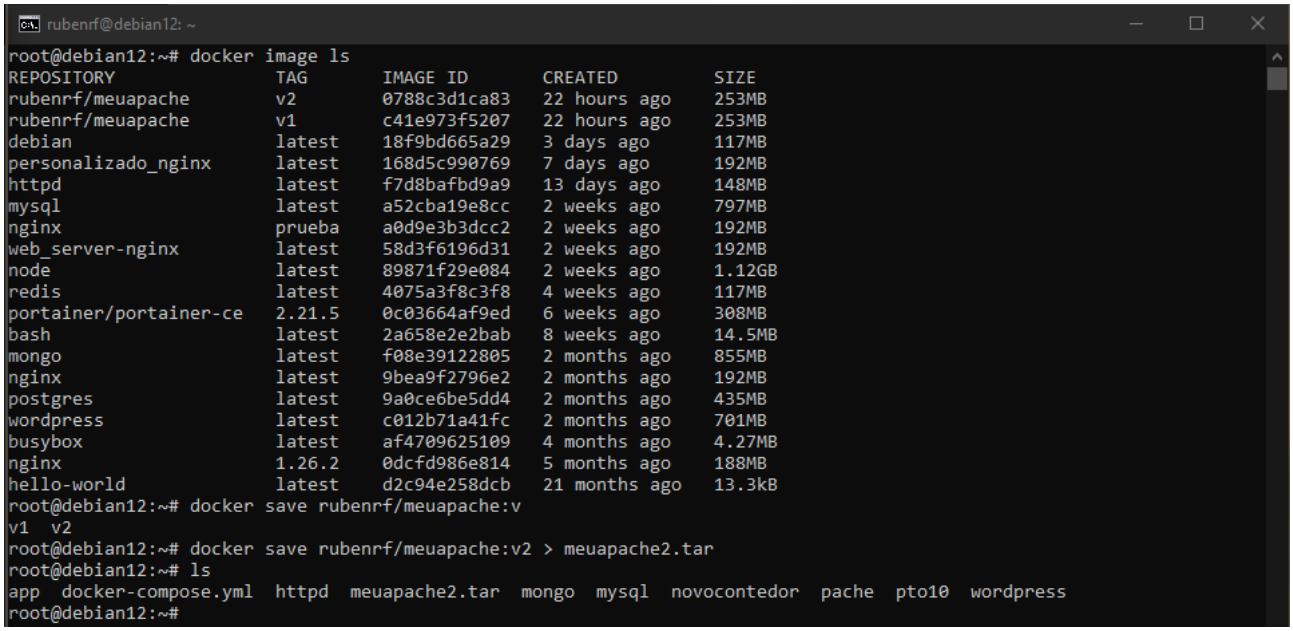
```
root@debian12:~/novocontedor# docker history rubenrf/meuapache:v2
IMAGE          CREATED              CREATED BY                                      SIZE      COMMENT
0788c3d1ca83   About a minute ago   CMD ["/usr/sbin/apache2ctl" "-D" "FOREGROUND…   0B        buildkit.dockerfile.v0
<missing>      About a minute ago   COPY index.html /var/www/html # buildkit        55B       buildkit.dockerfile.v0
<missing>      About a minute ago   RUN /bin/sh -c apt update && apt upgrade -y …   136MB     buildkit.dockerfile.v0
<missing>      About a minute ago   MAINTAINER Ruben Rey Feal                       0B        buildkit.dockerfile.v0
<missing>      2 days ago           # debian.sh --arch 'amd64' out/ 'bookworm' '…   117MB     debuerreotype 0.15
root@debian12:~/novocontedor#
```

6. Executa o novo contedor coa imaxe creada e mapeado o porto 8085 e comproba o acceso ao ficheiro HTML creado anteriormente

```
root@debian12:~/novocontedor# docker run -d --name miapache2 -p 8085:80 rubenrf/meuapache:v2
87adf2a54fe80f80b07aa27b011a6520f7ba6bc6b3d5b83628708ec1c0d59898
root@debian12:~/novocontedor# docker ps
CONTAINER ID   IMAGE                 COMMAND                  CREATED         STATUS          PORTS
                  NAMES
87adf2a54fe8   rubenrf/meuapache:v2   "/usr/sbin/apache2ct…"   4 seconds ago   Up 3 seconds    0.0.0.0:8085->80/tcp,
[::]:8085->80/tcp   miapache2
bfcc78720e51   wordpress             "docker-entrypoint.s…"   29 hours ago    Up 50 minutes   0.0.0.0:8080->80/tcp,
[::]:8080->80/tcp   wordpress_base
root@debian12:~/novocontedor#
```

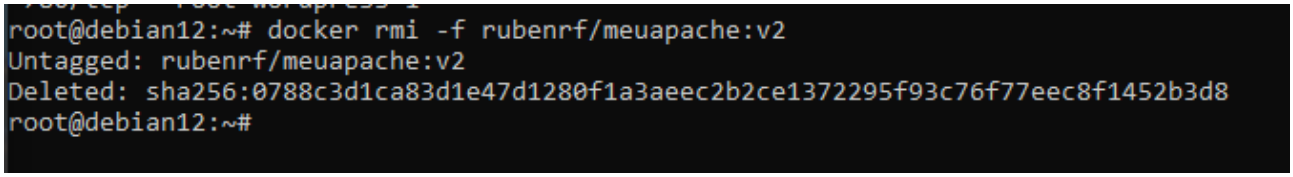
3.3. Distribución da imaxe

1. Salva a nova imaxe creada co nome meuapache2.tar.



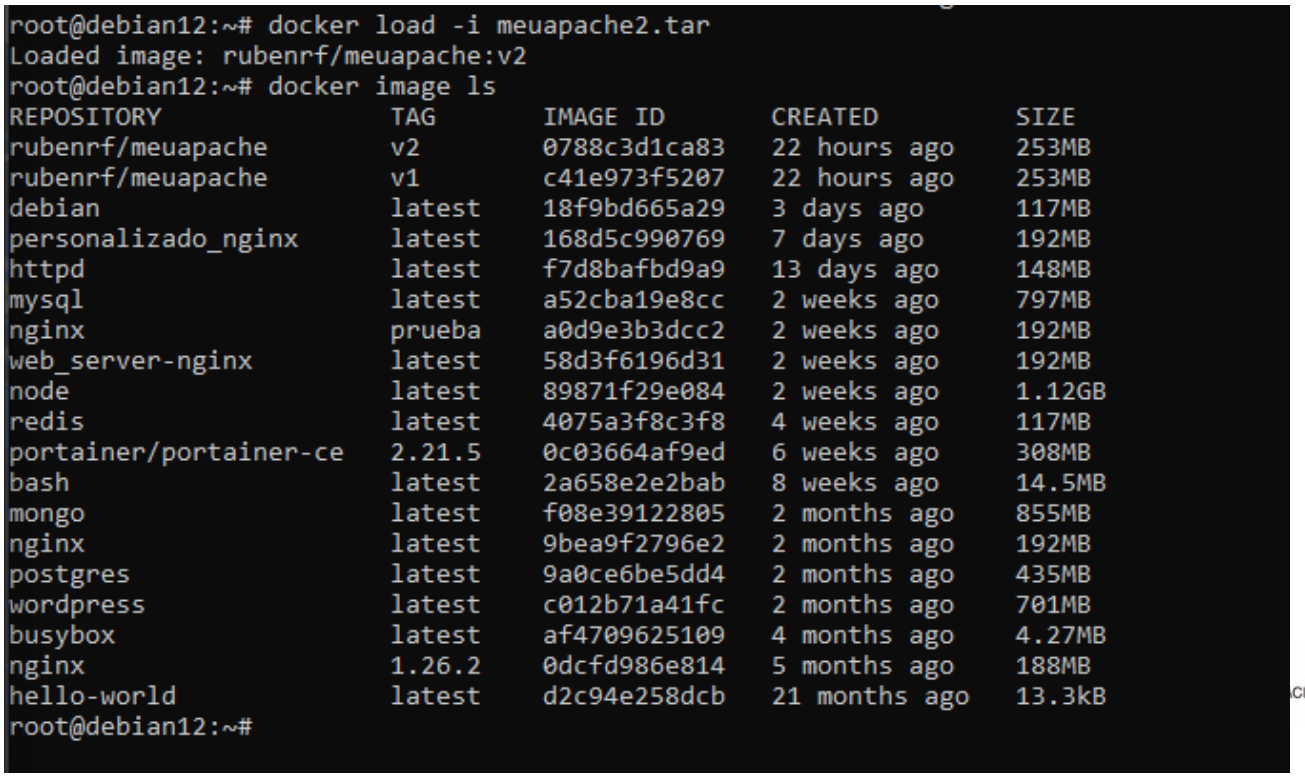
```
rubenrf@debian12: ~
root@debian12:~# docker image ls
REPOSITORY                TAG       IMAGE ID       CREATED         SIZE
rubenrf/meuapache         v2        0788c3d1ca83   22 hours ago    253MB
rubenrf/meuapache         v1        c41e973f5207   22 hours ago    253MB
debian                    latest    18f9bd665a29   3 days ago      117MB
personalizado_nginx       latest    168d5c990769   7 days ago      192MB
httpd                     latest    f7d8bafbd9a9   13 days ago     148MB
mysql                     latest    a52cba19e8cc   2 weeks ago     797MB
nginx                     prueba    a0d9e3b3dcc2   2 weeks ago     192MB
web_server-nginx          latest    58d3f6196d31   2 weeks ago     192MB
node                      latest    89871f29e084   2 weeks ago     1.12GB
redis                     latest    4075a3f8c3f8   4 weeks ago     117MB
portainer/portainer-ce    2.21.5    0c03664af9ed   6 weeks ago     308MB
bash                      latest    2a658e2e2bab   8 weeks ago     14.5MB
mongo                     latest    f08e39122805   2 months ago    855MB
nginx                     latest    9bea9f2796e2   2 months ago    192MB
postgres                  latest    9a0ce6be5dd4   2 months ago    435MB
wordpress                 latest    c012b71a41fc   2 months ago    701MB
busybox                   latest    af4709625109   4 months ago    4.27MB
nginx                     1.26.2    0dcfd986e814   5 months ago    188MB
hello-world               latest    d2c94e258dcb   21 months ago   13.3kB
root@debian12:~# docker save rubenrf/meuapache:v
v1  v2
root@debian12:~# docker save rubenrf/meuapache:v2 > meuapache2.tar
root@debian12:~# ls
app  docker-compose.yml  httpd  meuapache2.tar  mongo  mysql  novocontedor  pache  pto10  wordpress
root@debian12:~#
```

2. Borra a imaxe «`teunomeiniciais/meuapache:`**v2**».



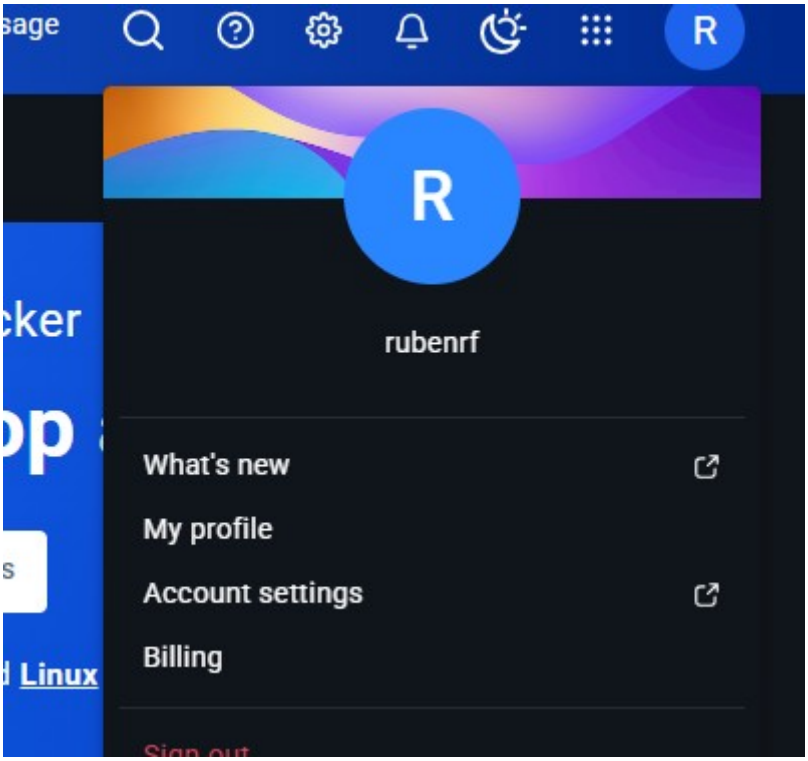
```
root@debian12:~# docker rmi -f rubenrf/meuapache:v2
Untagged: rubenrf/meuapache:v2
Deleted: sha256:0788c3d1ca83d1e47d1280f1a3aeec2b2ce1372295f93c76f77eec8f1452b3d8
root@debian12:~#
```

3. Carga de novo a imaxe dende o ficheiro tar creado antes.



```
root@debian12:~# docker load -i meuapache2.tar
Loaded image: rubenrf/meuapache:v2
root@debian12:~# docker image ls
REPOSITORY                TAG       IMAGE ID       CREATED         SIZE
rubenrf/meuapache         v2        0788c3d1ca83   22 hours ago    253MB
rubenrf/meuapache         v1        c41e973f5207   22 hours ago    253MB
debian                    latest    18f9bd665a29   3 days ago      117MB
personalizado_nginx       latest    168d5c990769   7 days ago      192MB
httpd                     latest    f7d8bafbd9a9   13 days ago     148MB
mysql                     latest    a52cba19e8cc   2 weeks ago     797MB
nginx                     prueba    a0d9e3b3dcc2   2 weeks ago     192MB
web_server-nginx          latest    58d3f6196d31   2 weeks ago     192MB
node                      latest    89871f29e084   2 weeks ago     1.12GB
redis                     latest    4075a3f8c3f8   4 weeks ago     117MB
portainer/portainer-ce    2.21.5    0c03664af9ed   6 weeks ago     308MB
bash                      latest    2a658e2e2bab   8 weeks ago     14.5MB
mongo                     latest    f08e39122805   2 months ago    855MB
nginx                     latest    9bea9f2796e2   2 months ago    192MB
postgres                  latest    9a0ce6be5dd4   2 months ago    435MB
wordpress                 latest    c012b71a41fc   2 months ago    701MB
busybox                   latest    af4709625109   4 months ago    4.27MB
nginx                     1.26.2    0dcfd986e814   5 months ago    188MB
hello-world               latest    d2c94e258dcb   21 months ago   13.3kB
root@debian12:~#
```

4. Crea un usuario en Docker Hub (opcional).



5. Identifícate en Docker Hub dende o terminal da túa máquina virtual (opcional).

```
root@debian12:~# docker login -u rubenrf
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/#credential-stores

Login Succeeded
```

6. Sube a imaxe a docker Hub login/meuapache:v1 (opcional).

```
root@debian12:~# docker push rubenrf/meuapache:v2
The push refers to repository [docker.io/rubenrf/meuapache]
6e83a80c121d: Pushed
1a67624d4565: Pushed
91b542912d12: Mounted from library/debian
v2: digest: sha256:4fe34d87c1bf1a052fe6e5f2a25d45b2af34c8a790842d6edfbc26fdea7663fc size: 948
root@debian12:~#
```

```
root@debian12:~# docker image tag rubenrf/meuapache:v2 rubenreyfeal/meuapache:v2
root@debian12:~# docker image ls
REPOSITORY               TAG       IMAGE ID       CREATED         SIZE
rubenrf/meuapache        v2        0788c3d1ca83   23 hours ago    253MB
rubenreyfeal/meuapache   v2        0788c3d1ca83   23 hours ago    253MB
rubenrf/meuapache        v1        c41e973f5207   23 hours ago    253MB
```

Tarefa 4. Monitorización Docker

Completa os seguintes apartados facendo as capturas de imaxe necesarias que demostren o correcto funcionamento dos contedores e a súa monitorización.

1. Poñer en marcha un contedor cunha imaxe de `nginx` en segundo plano (modo *detached*). A continuación inspeccionar a configuración e os detalles de dito contedor relativos á **rede**, **volumes**, **memoria** e **CPU**. Tentar obter os datos anteriores por separado usando o formato de notación *json*.

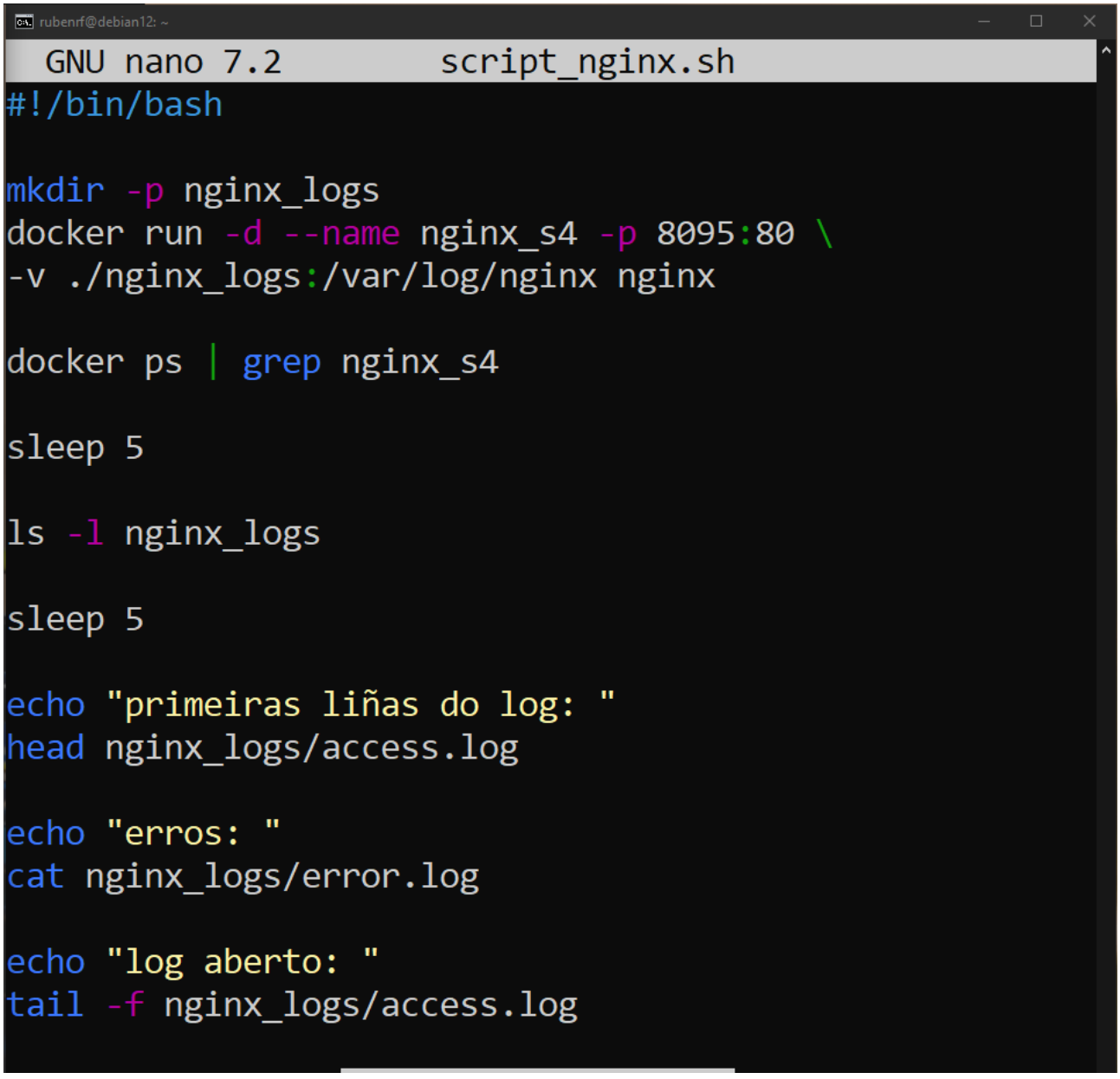
```
root@debian12:~# docker run -d --name nginx_1 nginx
00bd7b23357a07f66460163721092828eb5578583acd8e46c21e2027f01e0d84
root@debian12:~#
```

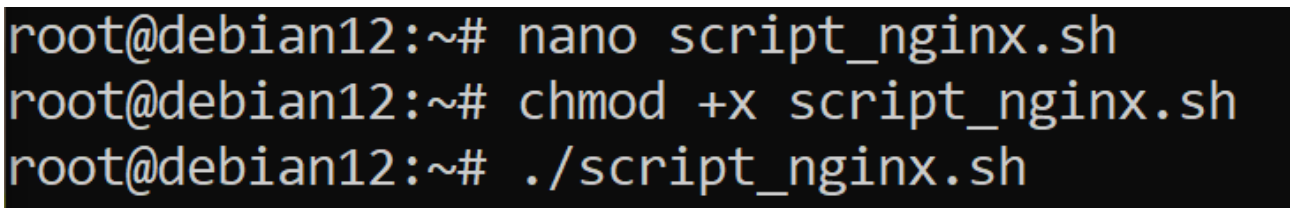
```
root@debian12:~# docker inspect nginx_1 -f '{{json .NetworkSettings}}'
{"Bridge":"","SandboxID":"f248da5747a2b6ce9f10bc7edc382227d577900751dbf242b091d8e9c8b65d41","SandboxKey":"/var/run/docker/netns/f248da5747a2","Ports":{"80/tcp":null},"HairpinMode":false,"LinkLocalIPv6Address":"","LinkLocalIPv6PrefixLen":0,"SecondaryIPAddresses":null,"SecondaryIPv6Addresses":null,"EndpointID":"aecbf000aed1ccb1acade38a1ee80bdcf6e8eee0da684688c68da94412e08399","Gateway":"172.17.0.1","GlobalIPv6Address":"","GlobalIPv6PrefixLen":0,"IPAddress":"172.17.0.2","IPPrefixLen":16,"IPv6Gateway":"","MacAddress":"02:42:ac:11:00:02","Networks":{"bridge":{"IPAMConfig":null,"Links":null,"Aliases":null,"MacAddress":"02:42:ac:11:00:02","DriverOpts":null,"NetworkID":"f36bd953bc036c97acff940f9ab3fc0bc8788ae3da80c9060c7b78686fd5bf10","EndpointID":"aecbf000aed1ccb1acade38a1ee80bdcf6e8eee0da684688c68da94412e08399","Gateway":"172.17.0.1","IPAddress":"172.17.0.2","IPPrefixLen":16,"IPv6Gateway":"","GlobalIPv6Address":"","GlobalIPv6PrefixLen":0,"DNSNames":null}}}
root@debian12:~# docker inspect nginx_1 -f '{{json .Mounts}}'
[]
root@debian12:~# docker inspect nginx_1 -f '{{json .HostConfig.Memory}}'
0
root@debian12:~# docker inspect nginx_1 -f '{{json .HostConfig.CpuShares}}'
0
root@debian12:~#
```

2. Obter estatísticas de uso de recursos do sistema en tempo real para os contedores activos no sistema. Reparar no uso de CPU e de memoria de cada un dos contedores.

```
root@debian12:~# docker stats nginx_1 --no-stream
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT     MEM %     NET I/O       BLOCK I/O         PIDS
00bd7b23357a   nginx_1   0.00%     3.871MiB / 1.921GiB   0.20%     1.53kB / 0B   8.57MB / 12.3kB   2
root@debian12:~#
```

3. Automatizar a execución dun contedor `nginx` cun volume de log para almacenar os logs do servidor no host. Facelo creando un directorio `nginx_logs` para almacenar os logs e executando o contedor `nginx` del tal modo que monte nese directorio `nginx_logs` como volume para o directorio `/var/log/nginx` o contedor. Verificar que os logs se almacenan correctamente no directorio mapeado.





4. Crear un volume en Docker e executar un contedor coa imaxe `busybox` que faga uso de dito volume. Empregar por exemplo un directorio chamado `/data` do contedor. Faga tamén que a execución sexa en segundo plano (*detached*). Usar o comando *docker inspect* para verificar a montaxe do volume no contedor.

```
root@debian12:~# docker run -d --name busybox_s3 -v busybox_3:/data busybox sleep 30000
2b6e3905f52a881ce1a7f0eb0d41022694bae7147acf95b6beab17651f089ffe
root@debian12:~# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS
                                NAMES
2b6e3905f52a   busybox        "sleep 30000"            4 seconds ago    Up 4 seconds
                                busybox_s3
a4b1a7a75c28   nginx          "/docker-entrypoint.…"   4 minutes ago    Up 4 minutes    0.0.0.
0:8095->80/tcp, [::]:8095->80/tcp   nginx_s4
00bd7b23357a   nginx          "/docker-entrypoint.…"   29 minutes ago   Up 29 minutes   80/tcp
                                nginx_1
45c4ce1f50e8   mysql:latest   "docker-entrypoint.s…"   24 hours ago     Up 2 hours      3306/t
cp, 33060/tcp                   root-db-1
86657613decb   wordpress      "docker-entrypoint.s…"   24 hours ago     Up 2 hours      0.0.0.
0:8081->80/tcp, [::]:8081->80/tcp   root-wordpress-1
root@debian12:~#
```

```
root@debian12:~# docker inspect busybox_s3
```

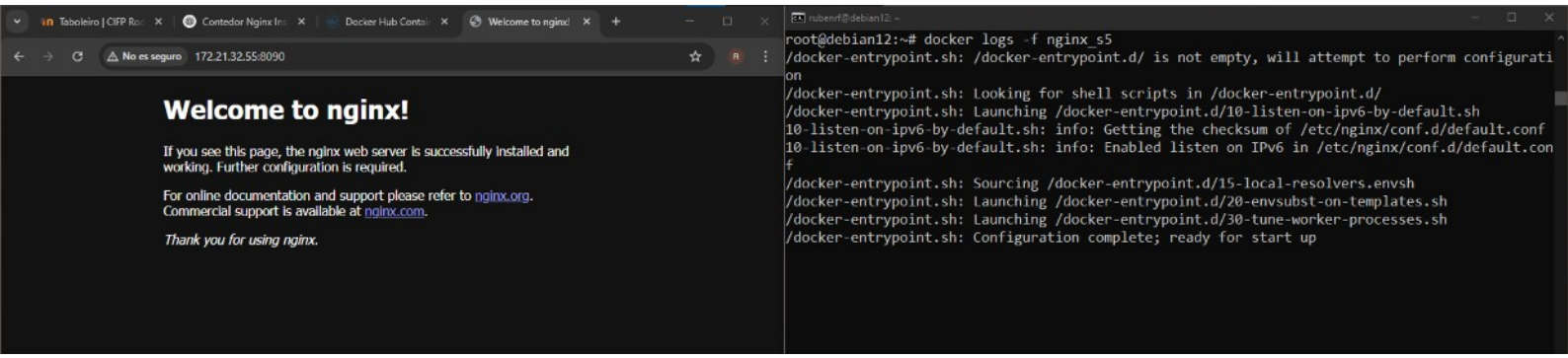
```
    "Mounts": [
        {
            "Type": "volume",
            "Name": "busybox_3",
            "Source": "/var/lib/docker/volumes/busybox_3/_data",
            "Destination": "/data",
            "Driver": "local",
            "Mode": "z",
            "RW": true,
            "Propagation": ""
        }
    ],
```

5. Poñer en marcha un contedor cunha imaxe de `nginx` en segundo plano (modo *detached*), mapeando o porto 80 do contedor a un porto coñecido no anfitrión (por exemplo, 8090). Utilicar o comando `docker logs` para visualizar os rexistros de actividade de dito contedor. Fixarse no tipo de información que proporcionan os *logs* dun contedor e como se poden usar para detectar problemas. Buscar o modo de visualizar só as últimas 10 liñas dos *logs* dun contedor.

```
root@debian12:~# mkdir nginx_logs2
root@debian12:~# docker run -d --name nginx_s5 -v ./nginx_logs2:/var/log/nginx -p 8090:80 ngin
x
25981547555ea0745ba740307b5443120665a07ddaf0e85f5aeeec5fa4940e01
```

```
root@debian12:~# docker logs --tail 10 nginx_s5
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configurati
on
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.con
f
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
root@debian12:~#
```

6. Repetir o escenario do anterior exercicio, pero agora empregar o comando `docker logs` para visualizar os rexistros de actividade en tempo real ó mesmo tempo que se accede mediante un navegador ao servizo. Observar que ocorre cada vez que solicita a páxina raíz http://IP-MAQUINA:8090.



7. Poñer en marcha un *stack* creando un ficheiro `docker-compose.yml` que levante tres servizos: **Prometheus**, **Grafana** e **Node Exporter** para monitorizar o sistema anfitrión (os nomes das imaxes son, respectivamente, `prom/prometheus:latest`, `grafana/grafana:latest` e `prom/node_exporter:latest`). Xa cos servizos en marcha, acceder a Grafana no URL http://IP-MÁQUINA:3000 e engadir Prometheus coma fonte de datos. Configurar un *dashboard* para visualizar métricas coma uso de memoria, tráfico de rede e carga do sistema (OPCIONAL).

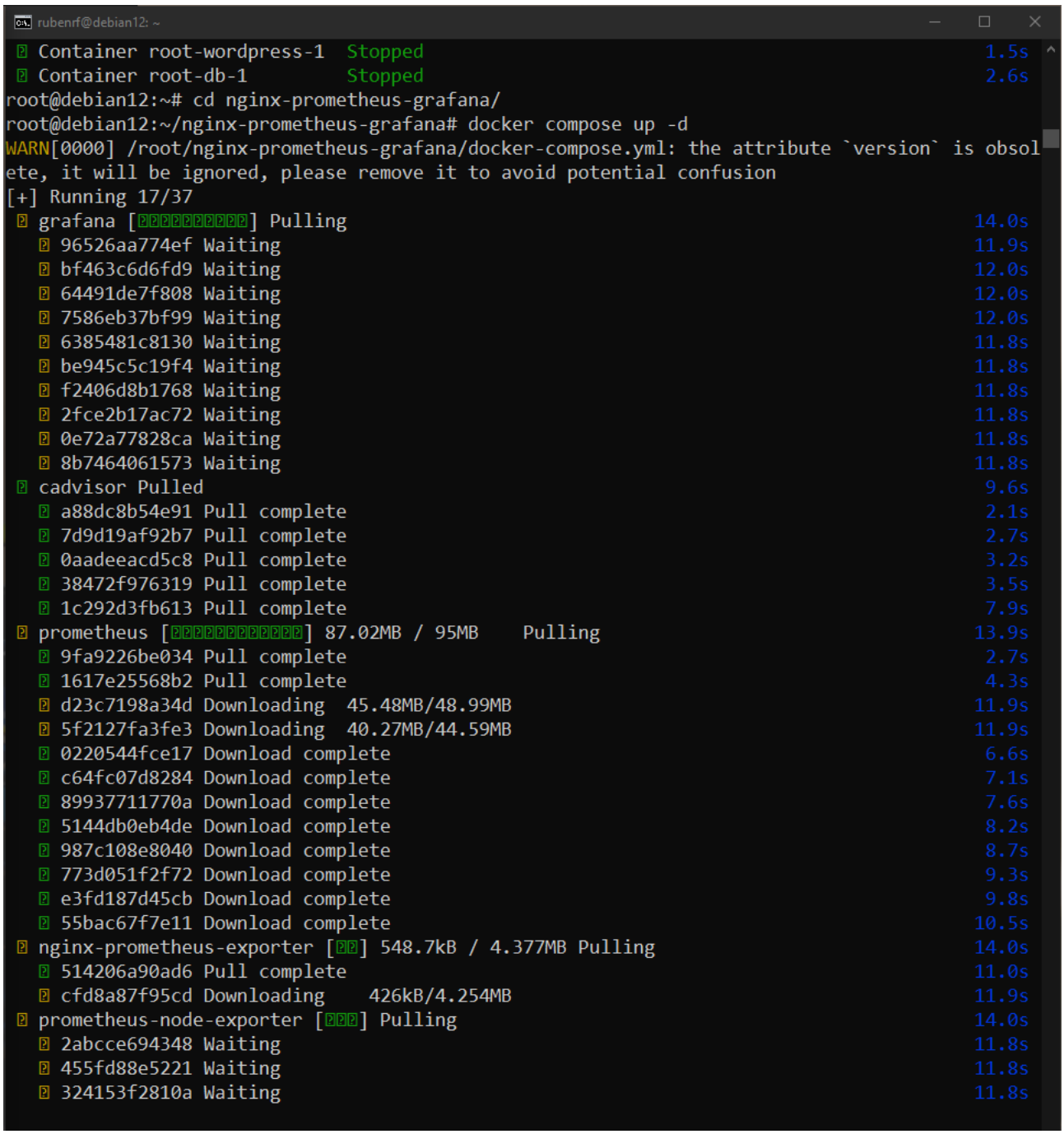
8. Extender o exercicio anterior para que facer a monitorización dun par de contedores adicionais con imaxes de **Nginx** e **Mysql**. Configurar alertas para emitir notificacións cando a carga do sistema ou o uso de memoria superen un límite (OPCIONAL).

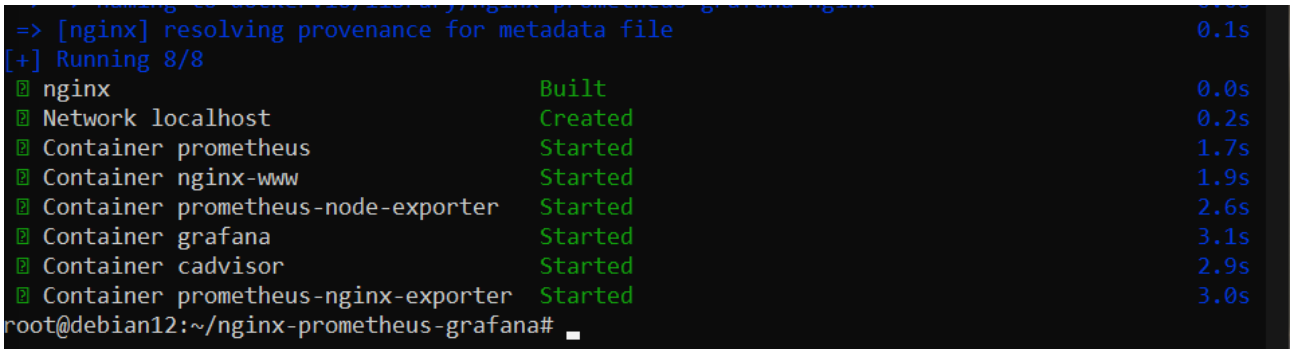
```
root@debian12:~# git clone https://github.com/rvva/nginx-prometheus-grafana/
Clonando en 'nginx-prometheus-grafana'...
remote: Enumerating objects: 66, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 66 (delta 6), reused 9 (delta 4), pack-reused 48 (from 1)
Recibiendo objetos: 100% (66/66), 138.69 KiB | 755.00 KiB/s, listo.
Resolviendo deltas: 100% (22/22), listo.
root@debian12:~#
```

```
root@debian12:~# mkdir -p nginx-prometheus-grafana/{prometheus,grafana}/data
root@debian12:~# mkdir nginx-prometheus-grafana/nginx/log
root@debian12:~# ls nginx-prometheus-grafana/
docker-compose.yml  grafana  html  LICENSE  nginx  prometheus  README.md
root@debian12:~# ls nginx-prometheus-grafana/grafana/
data
root@debian12:~# ls nginx-prometheus-grafana/prometheus/
data  prometheus.yml
root@debian12:~# ls nginx-prometheus-grafana/nginx/
conf  Dockerfile  log
root@debian12:~#
```

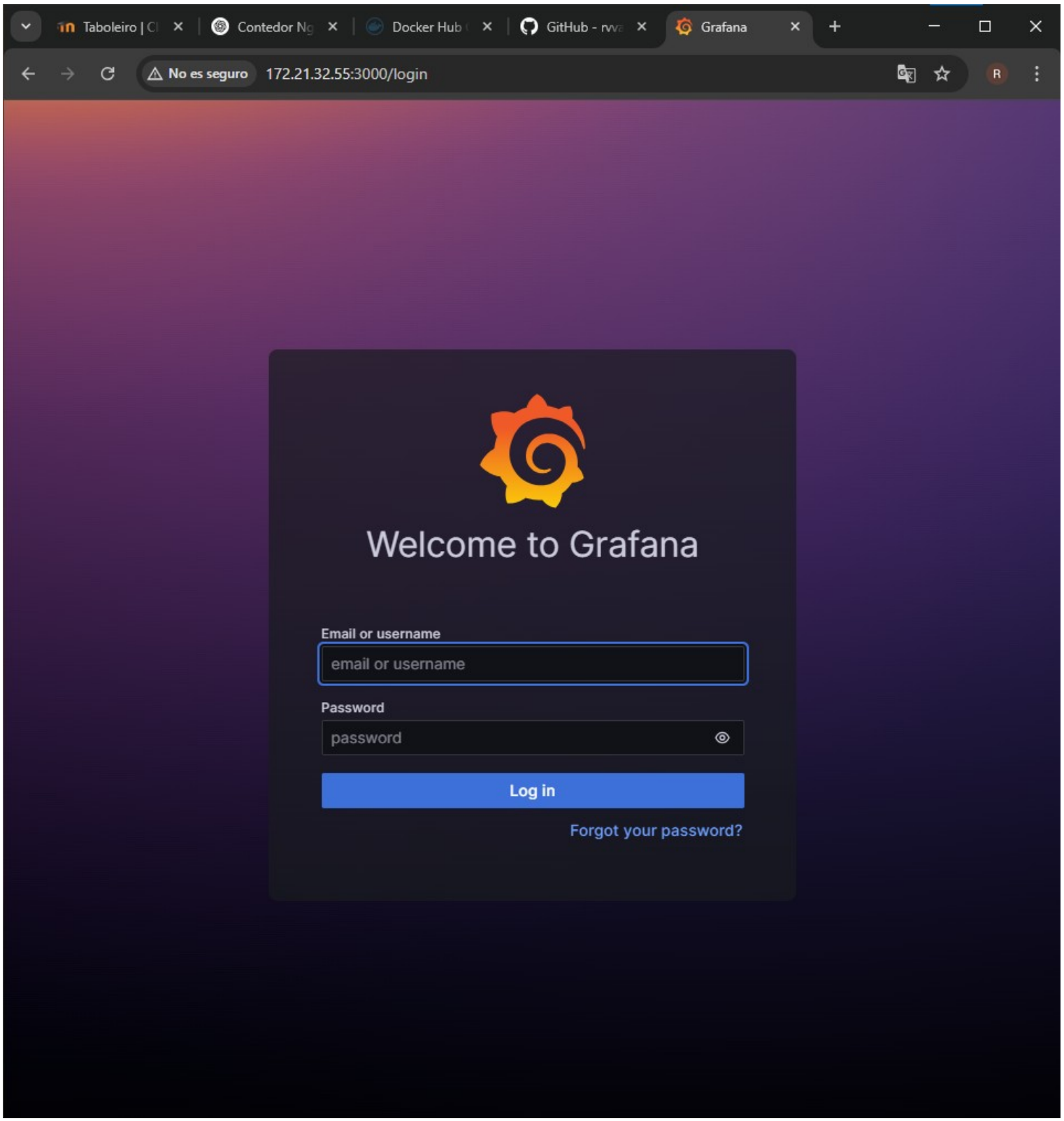
```
root@debian12:~# ls -a nginx-prometheus-grafana/
.  ..  docker-compose.yml  .env  .git  grafana  html  LICENSE  nginx  prometheus  README.md
root@debian12:~# cat .env
cat: .env: No existe el fichero o el directorio
root@debian12:~# cat nginx-prometheus-grafana/.env
GF_SECURITY_ADMIN_USER=user
GF_SECURITY_ADMIN_PASSWORD=password
root@debian12:~# nano nginx-prometheus-grafana/.env
root@debian12:~#
```

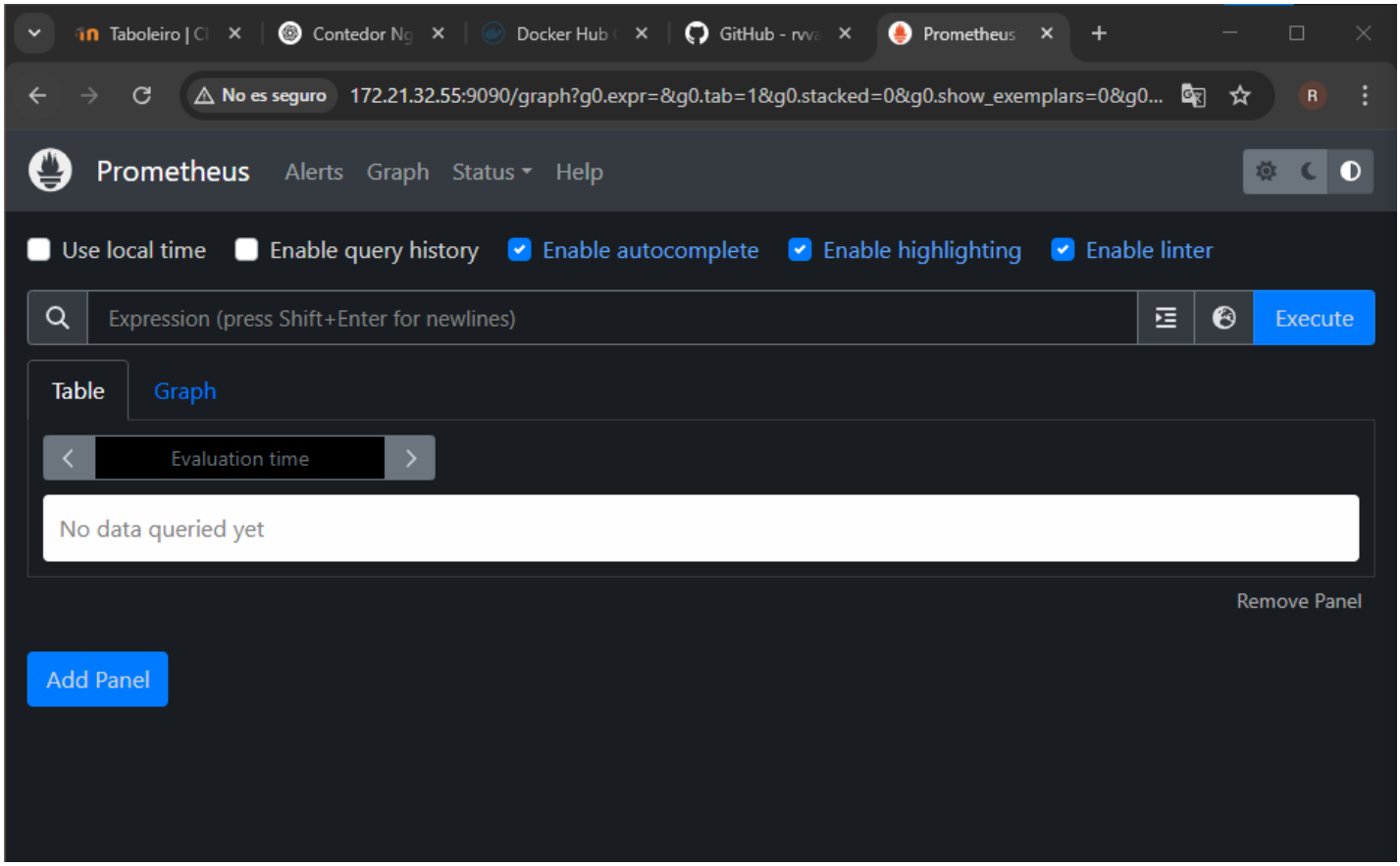
```
rubenrf@debian12: ~
  GNU nano 7.2                    nginx-prometheus-grafana/.env *
GF_SECURITY_ADMIN_USER=rubenrf
GF_SECURITY_ADMIN_PASSWORD=abc123.
```

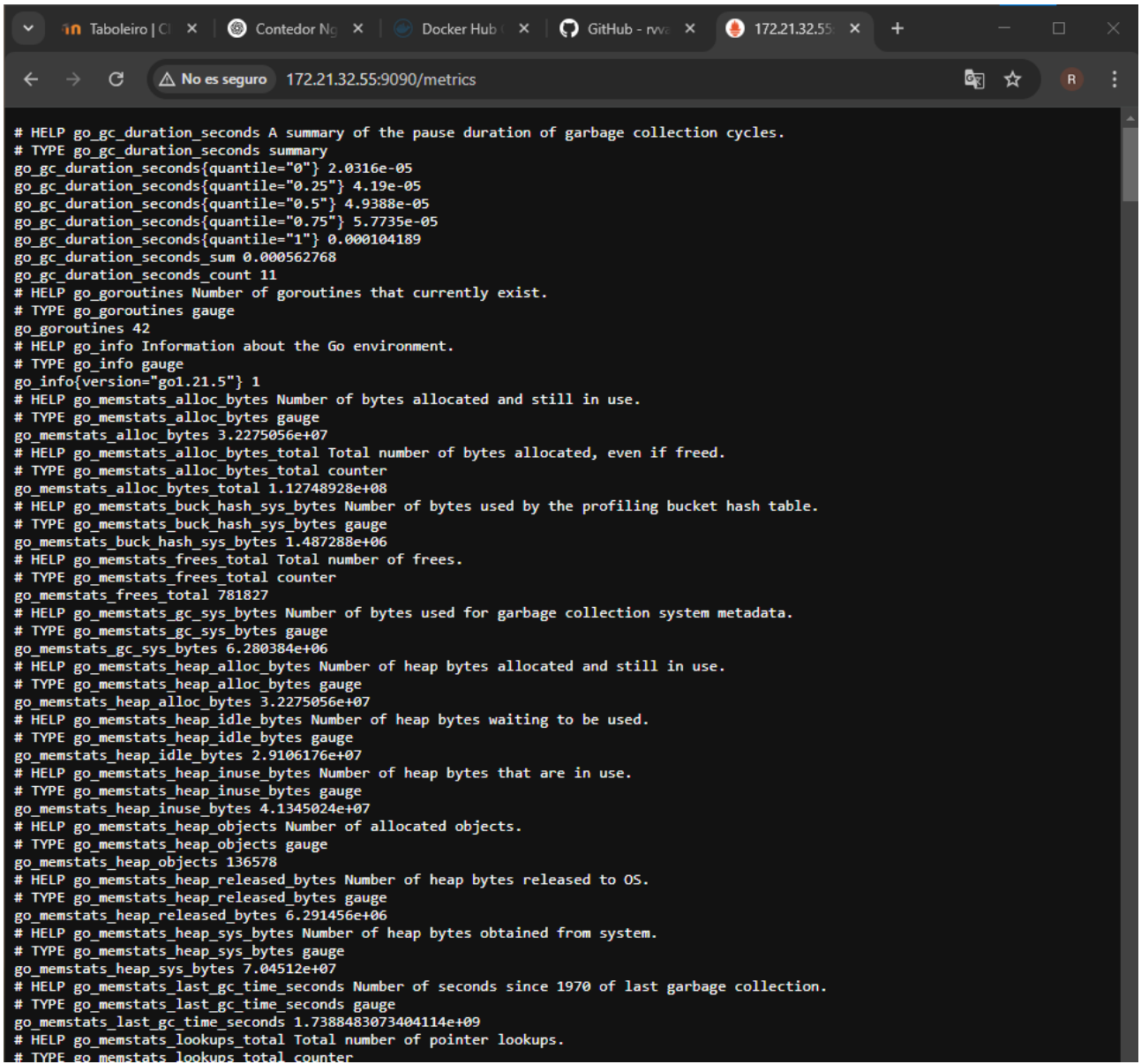

rubenrf@debian12: ~
Container root-wordpress-1 Stopped 1.5s
Container root-db-1 Stopped 2.6s
root@debian12:~# cd nginx-prometheus-grafana/
root@debian12:~/nginx-prometheus-grafana# docker compose up -d
WARN[0000] /root/nginx-prometheus-grafana/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 17/37
grafana [] Pulling 14.0s
96526aa774ef Waiting 11.9s
bf463c6d6fd9 Waiting 12.0s
64491de7f808 Waiting 12.0s
7586eb37bf99 Waiting 12.0s
6385481c8130 Waiting 11.8s
be945c5c19f4 Waiting 11.8s
f2406d8b1768 Waiting 11.8s
2fce2b17ac72 Waiting 11.8s
0e72a77828ca Waiting 11.8s
8b7464061573 Waiting 11.8s
cadvisor Pulled 9.6s
a88dc8b54e91 Pull complete 2.1s
7d9d19af92b7 Pull complete 2.7s
0aadeeacd5c8 Pull complete 3.2s
38472f976319 Pull complete 3.5s
1c292d3fb613 Pull complete 7.9s
prometheus [] 87.02MB / 95MB Pulling 13.9s
9fa9226be034 Pull complete 2.7s
1617e25568b2 Pull complete 4.3s
d23c7198a34d Downloading 45.48MB/48.99MB 11.9s
5f2127fa3fe3 Downloading 40.27MB/44.59MB 11.9s
0220544fce17 Download complete 6.6s
c64fc07d8284 Download complete 7.1s
89937711770a Download complete 7.6s
5144db0eb4de Download complete 8.2s
987c108e8040 Download complete 8.7s
773d051f2f72 Download complete 9.3s
e3fd187d45cb Download complete 9.8s
55bac67f7e11 Download complete 10.5s
nginx-prometheus-exporter [] 548.7kB / 4.377MB Pulling 14.0s
514206a90ad6 Pull complete 11.0s
cfd8a87f95cd Downloading 426kB/4.254MB 11.9s
prometheus-node-exporter [] Pulling 14.0s
2abcce694348 Waiting 11.8s
455fd88e5221 Waiting 11.8s
324153f2810a Waiting 11.8s

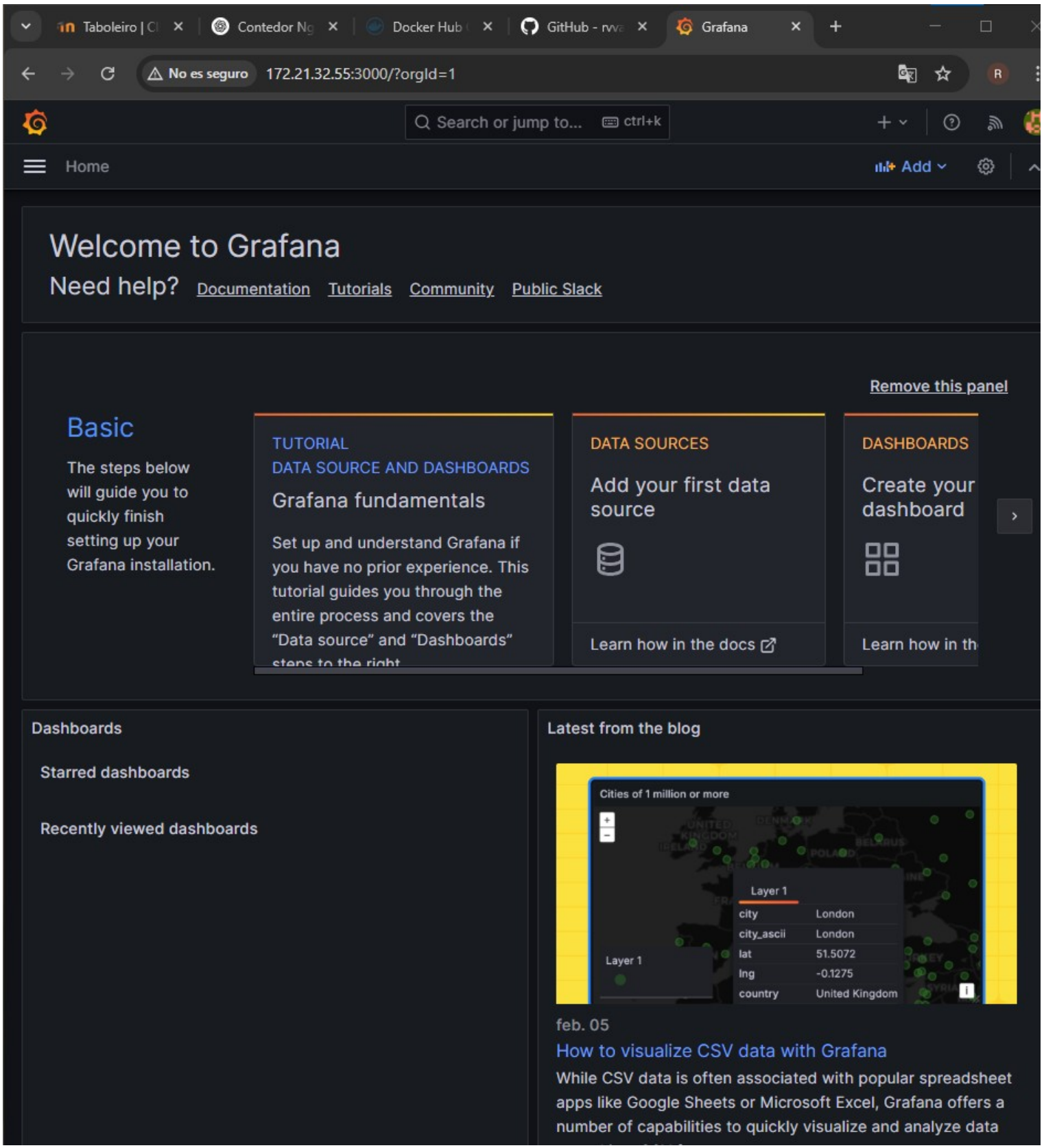

=> [nginx] resolving provenance for metadata file 0.1s
[+] Running 8/8
nginx Built 0.0s
Network localhost Created 0.2s
Container prometheus Started 1.7s
Container nginx-www Started 1.9s
Container prometheus-node-exporter Started 2.6s
Container grafana Started 3.1s
Container cadvisor Started 2.9s
Container prometheus-nginx-exporter Started 3.0s
root@debian12:~/nginx-prometheus-grafana#

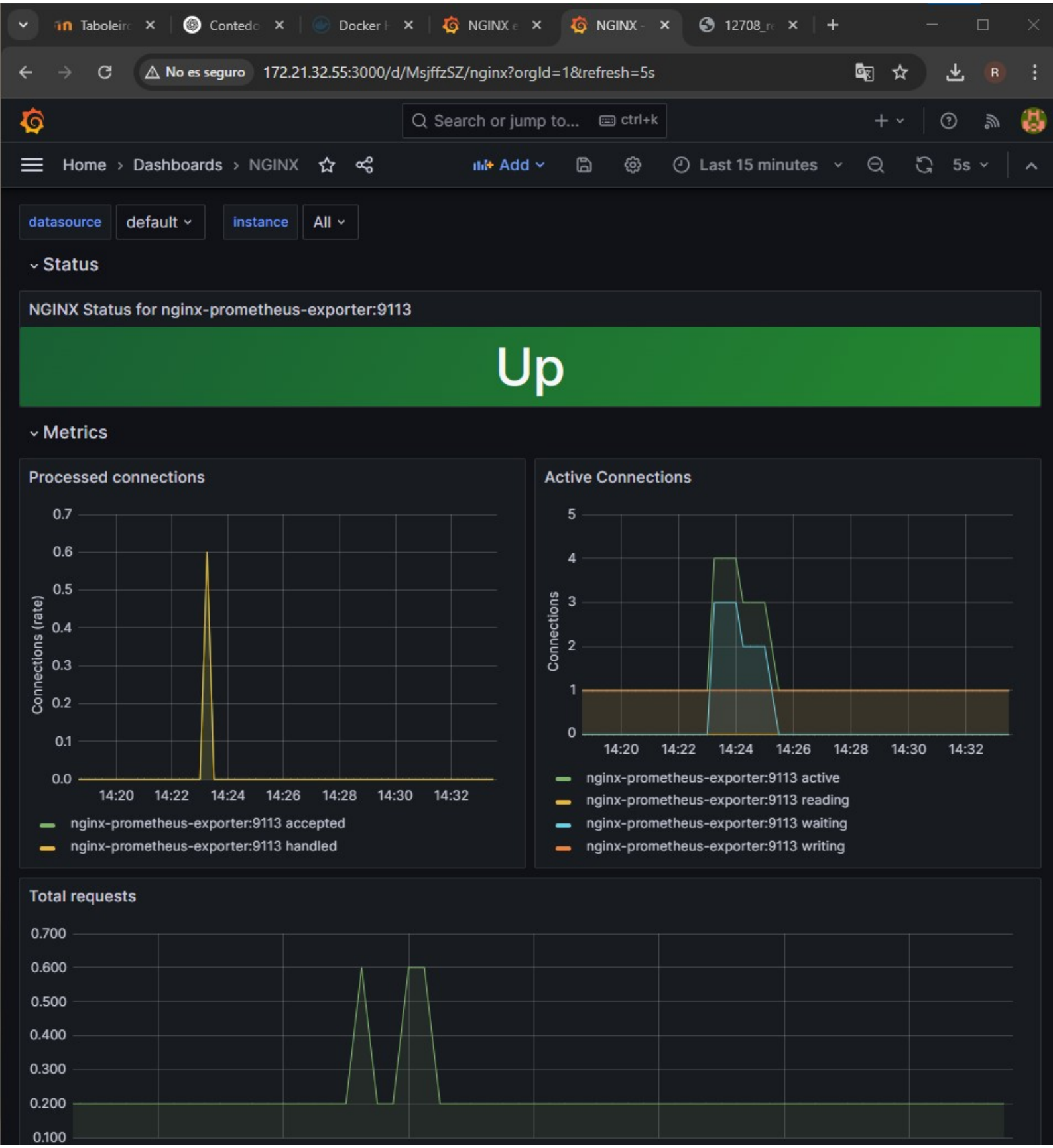
```
rubenrf@debian12: ~
root@debian12:~/nginx-prometheus-grafana# docker compose ps
WARN[0000] /root/nginx-prometheus-grafana/docker-compose.yml: the attribute `version` is obsol
ete, it will be ignored, please remove it to avoid potential confusion
NAME                          IMAGE                                    COMMAND                  SER
VICE                      CREATED         STATUS                   PORTS
cadvisor                      gcr.io/cadvisor/cadvisor:v0.47.2         "/usr/bin/cadvisor -…"   cad
visor                     2 minutes ago   Up 2 minutes (healthy)   8080/tcp
grafana                       grafana/grafana:10.0.10                  "/run.sh"                gra
fana                      2 minutes ago   Up 2 minutes             0.0.0.0:3000->3000/tcp, :::3
000->3000/tcp
nginx-www                     nginx-prometheus-grafana-nginx           "/docker-entrypoint.…"   ngi
nx                        2 minutes ago   Up 2 minutes             0.0.0.0:80->80/tcp, :::80->8
0/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp
prometheus                    prom/prometheus:v2.45.2                  "/bin/prometheus --c…"   pro
metheus                   2 minutes ago   Up 2 minutes             0.0.0.0:9090->9090/tcp, :::9
090->9090/tcp
prometheus-nginx-exporter     nginx/nginx-prometheus-exporter:1.0      "/usr/bin/nginx-prom…"   ngi
nx-prometheus-exporter    2 minutes ago   Up 2 minutes             9113/tcp
prometheus-node-exporter      prom/node-exporter:v1.7.0                "/bin/node_exporter …"   pro
metheus-node-exporter     2 minutes ago   Up 2 minutes             9100/tcp
root@debian12:~/nginx-prometheus-grafana# docker ps
CONTAINER ID   IMAGE                                    COMMAND                  CREATED
STATUS                  PORTS
      NAMES
5fba5f260df8   nginx/nginx-prometheus-exporter:1.0      "/usr/bin/nginx-prom…"   3 minutes ago
Up 3 minutes            9113/tcp
      prometheus-nginx-exporter
26b297d49806   gcr.io/cadvisor/cadvisor:v0.47.2         "/usr/bin/cadvisor -…"   3 minutes ago
Up 3 minutes (healthy)   8080/tcp
      cadvisor
8b36054289d2   grafana/grafana:10.0.10                  "/run.sh"                3 minutes ago
Up 3 minutes            0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
      grafana
597a6988b4da   prom/node-exporter:v1.7.0                "/bin/node_exporter …"   3 minutes ago
Up 3 minutes            9100/tcp
      prometheus-node-exporter
9ed5f2f87061   nginx-prometheus-grafana-nginx           "/docker-entrypoint.…"   3 minutes ago
Up 3 minutes            0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/
tcp   nginx-www
18b012cc7d34   prom/prometheus:v2.45.2                  "/bin/prometheus --c…"   3 minutes ago
Up 3 minutes            0.0.0.0:9090->9090/tcp, :::9090->9090/tcp
      prometheus
root@debian12:~/nginx-prometheus-grafana#
```

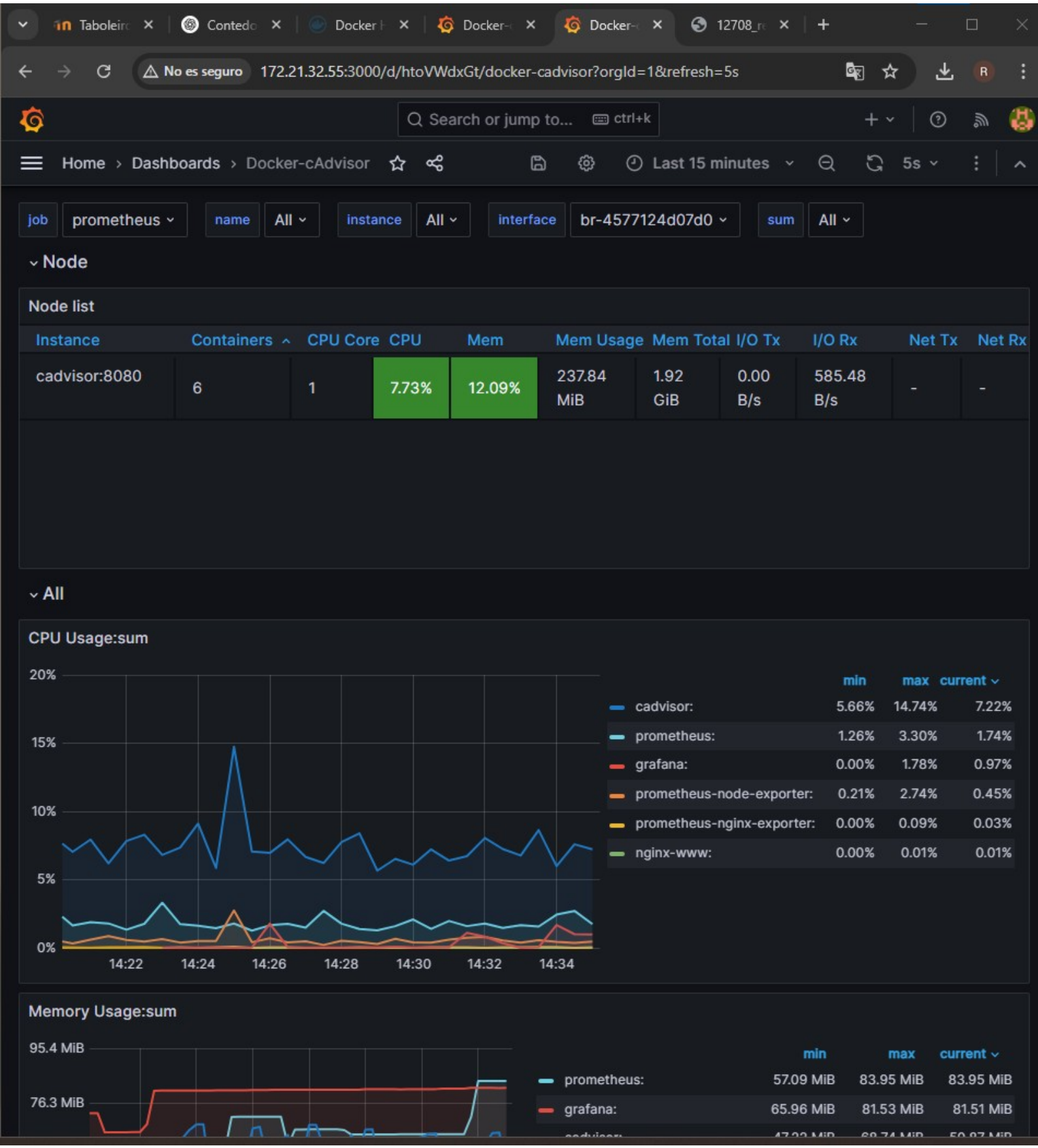

Taboleiro | Cl
Contedor Ng
Docker Hub
GitHub - rvva
Grafana
No es seguro
172.21.32.55:3000/login
Welcome to Grafana
Email or username
email or username
Password
password
Log in
Forgot your password?

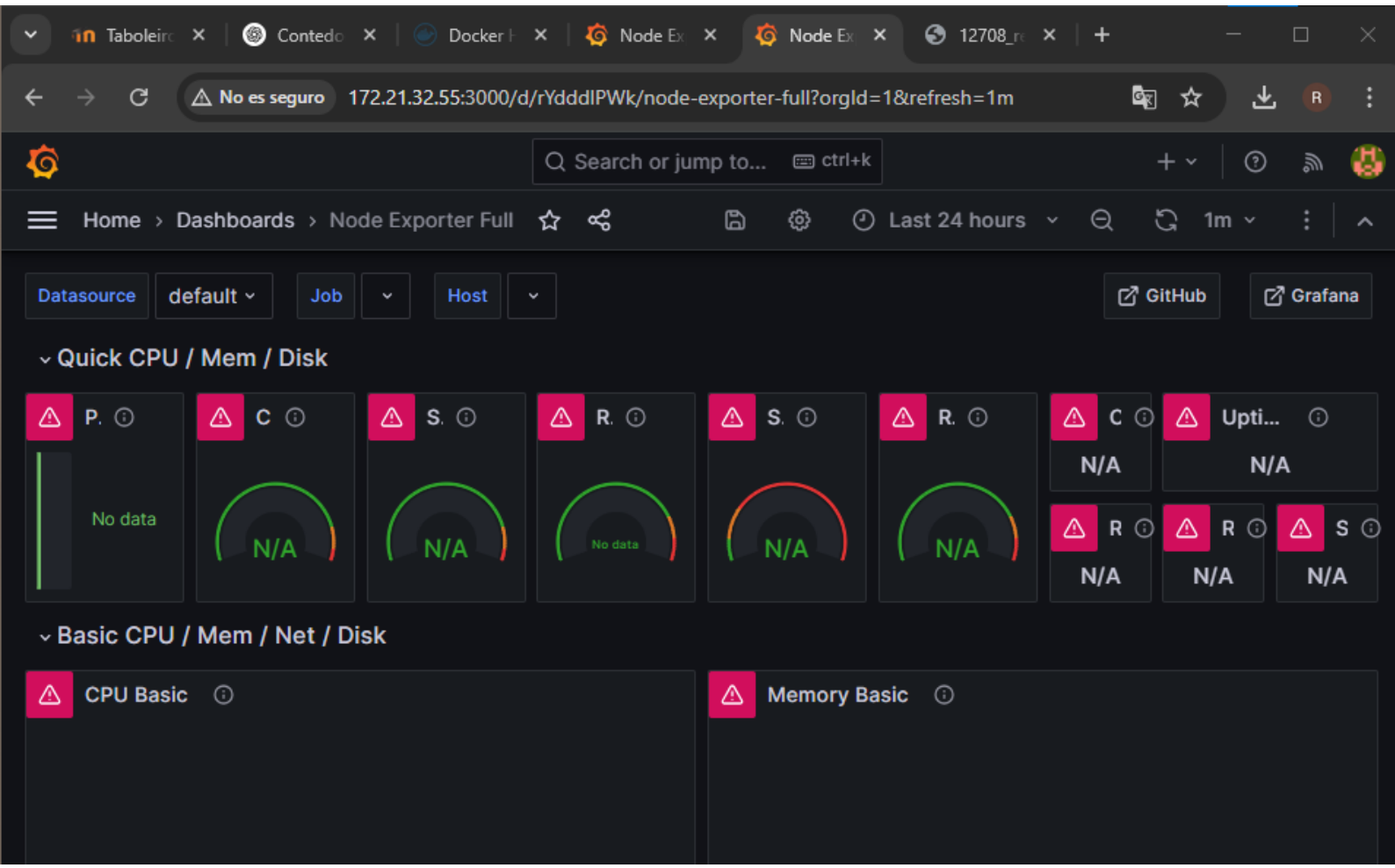

Taboleiro | Cl
Contedor Ng
Docker Hub
GitHub - rvv
Prometheus
No es seguro
172.21.32.55:9090/graph?g0.expr=&g0.tab=1&g0.stacked=0&g0.show_exemplars=0&g0...
Prometheus
Alerts
Graph
Status
Help
Use local time
Enable query history
Enable autocomplete
Enable highlighting
Enable linter
Expression (press Shift+Enter for newlines)
Execute
Table
Graph
Evaluation time
No data queried yet
Remove Panel
Add Panel

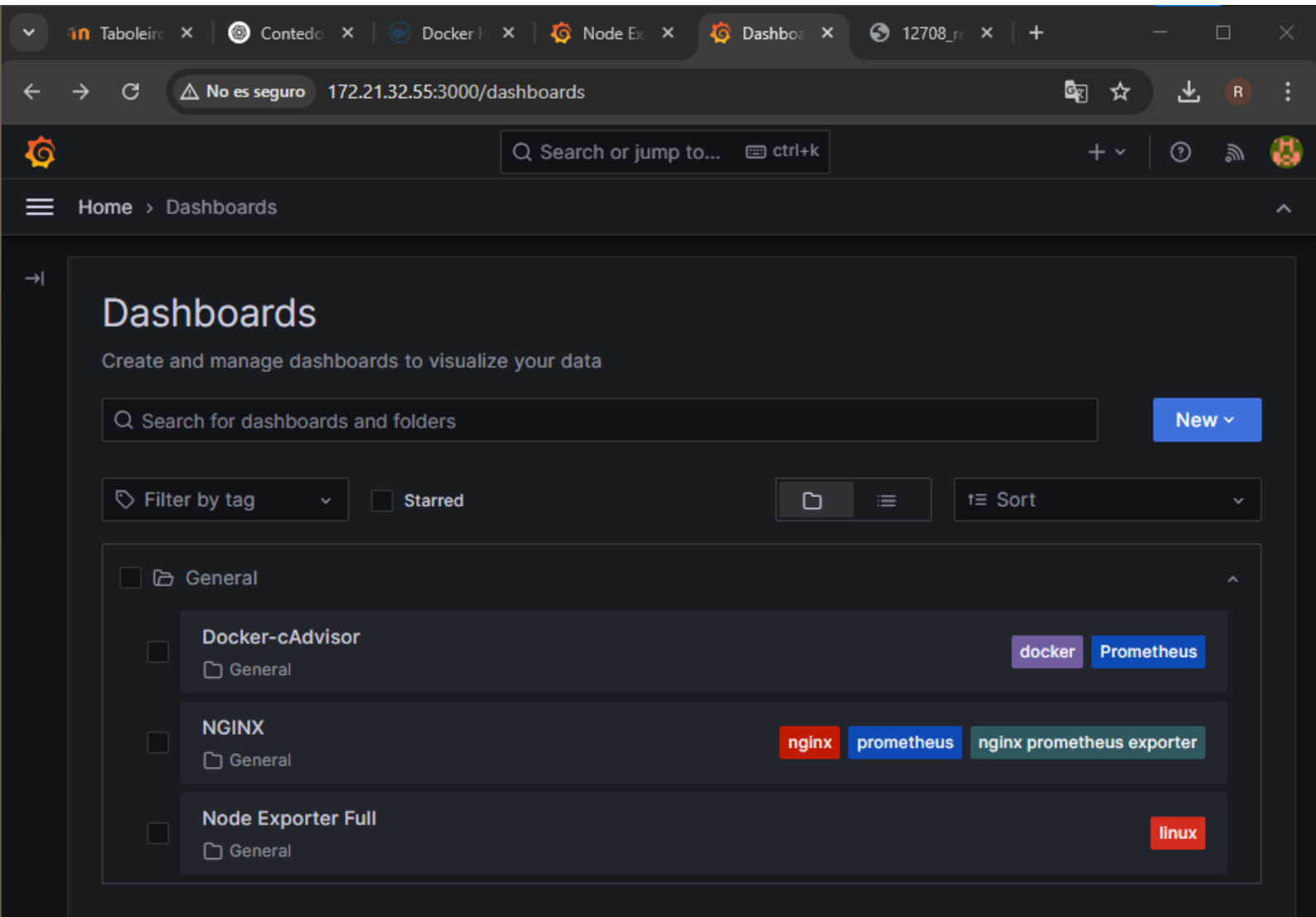

Taboleiro | Cl
Contedor Ng
Docker Hub
GitHub - rvv
172.21.32.55
No es seguro
172.21.32.55:9090/metrics
HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 2.0316e-05
go_gc_duration_seconds{quantile="0.25"} 4.19e-05
go_gc_duration_seconds{quantile="0.5"} 4.9388e-05
go_gc_duration_seconds{quantile="0.75"} 5.7735e-05
go_gc_duration_seconds{quantile="1"} 0.000104189
go_gc_duration_seconds_sum 0.000562768
go_gc_duration_seconds_count 11
HELP go_goroutines Number of goroutines that currently exist.
TYPE go_goroutines gauge
go_goroutines 42
HELP go_info Information about the Go environment.
TYPE go_info gauge
go_info{version="go1.21.5"} 1
HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 3.2275056e+07
HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 1.12748928e+08
HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.487288e+06
HELP go_memstats_frees_total Total number of frees.
TYPE go_memstats_frees_total counter
go_memstats_frees_total 781827
HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 6.280384e+06
HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 3.2275056e+07
HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 2.9106176e+07
HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 4.1345024e+07
HELP go_memstats_heap_objects Number of allocated objects.
TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 136578
HELP go_memstats_heap_released_bytes Number of heap bytes released to OS.
TYPE go_memstats_heap_released_bytes gauge
go_memstats_heap_released_bytes 6.291456e+06
HELP go_memstats_heap_sys_bytes Number of heap bytes obtained from system.
TYPE go_memstats_heap_sys_bytes gauge
go_memstats_heap_sys_bytes 7.04512e+07
HELP go_memstats_last_gc_time_seconds Number of seconds since 1970 of last garbage collection.
TYPE go_memstats_last_gc_time_seconds gauge
go_memstats_last_gc_time_seconds 1.7388483073404114e+09
HELP go_memstats_lookups_total Total number of pointer lookups.
TYPE go_memstats_lookups_total counter


Taboleiro | Cl
Contedor Ng
Docker Hub
GitHub - rvva
Grafana
No es seguro
172.21.32.55:3000/?orgId=1
Search or jump to... ctrl+k
Home
Add
Welcome to Grafana
Need help? Documentation Tutorials Community Public Slack
Remove this panel
Basic
The steps below will guide you to quickly finish setting up your Grafana installation.
TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right
DATA SOURCES
Add your first data source
Learn how in the docs
DASHBOARDS
Create your dashboard
Learn how in th
Dashboards
Starred dashboards
Recently viewed dashboards
Latest from the blog
Cities of 1 million or more
Layer 1
city London
city_ascii London
lat 51.5072
lng -0.1275
country United Kingdom
Layer 1
feb. 05
How to visualize CSV data with Grafana
While CSV data is often associated with popular spreadsheet apps like Google Sheets or Microsoft Excel, Grafana offers a number of capabilities to quickly visualize and analyze data


No es seguro
172.21.32.55:3000/d/MsjffzSZ/nginx?orgId=1&refresh=5s
Search or jump to... ctrl+k
Home › Dashboards › NGINX
Add
Last 15 minutes
5s
datasource
default
instance
All
Status
NGINX Status for nginx-prometheus-exporter:9113
Up
Metrics
Processed connections
Connections (rate)
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
14:20
14:22
14:24
14:26
14:28
14:30
14:32
nginx-prometheus-exporter:9113 accepted
nginx-prometheus-exporter:9113 handled
Active Connections
Connections
5
4
3
2
1
0
14:20
14:22
14:24
14:26
14:28
14:30
14:32
nginx-prometheus-exporter:9113 active
nginx-prometheus-exporter:9113 reading
nginx-prometheus-exporter:9113 waiting
nginx-prometheus-exporter:9113 writing
Total requests
0.700
0.600
0.500
0.400
0.300
0.200
0.100


No es seguro
172.21.32.55:3000/d/htoVWdxGt/docker-cadvisor?orgId=1&refresh=5s
Search or jump to... ctrl+k
Home › Dashboards › Docker-cAdvisor
Last 15 minutes
5s
job prometheus
name All
instance All
interface br-4577124d07d0
sum All
Node
Node list
Instance Containers CPU Core CPU Mem Mem Usage Mem Total I/O Tx I/O Rx Net Tx Net Rx
cadvisor:8080 6 1 7.73% 12.09% 237.84 MiB 1.92 GiB 0.00 B/s 585.48 B/s - -
All
CPU Usage:sum
20%
15%
10%
5%
0%
14:22
14:24
14:26
14:28
14:30
14:32
14:34
min max current
cadvisor: 5.66% 14.74% 7.22%
prometheus: 1.26% 3.30% 1.74%
grafana: 0.00% 1.78% 0.97%
prometheus-node-exporter: 0.21% 2.74% 0.45%
prometheus-nginx-exporter: 0.00% 0.09% 0.03%
nginx-www: 0.00% 0.01% 0.01%
Memory Usage:sum
95.4 MiB
76.3 MiB
min max current
prometheus: 57.09 MiB 83.95 MiB 83.95 MiB
grafana: 65.96 MiB 81.53 MiB 81.51 MiB


No es seguro
172.21.32.55:3000/d/rYdddlPWk/node-exporter-full?orgId=1&refresh=1m
Search or jump to... ctrl+k
Home › Dashboards › Node Exporter Full
Last 24 hours
1m
Datasource
default
Job
Host
GitHub
Grafana
Quick CPU / Mem / Disk
No data
N/A
Basic CPU / Mem / Net / Disk
CPU Basic
Memory Basic


No es seguro
172.21.32.55:3000/dashboards
Search or jump to... ctrl+k
Home › Dashboards
Dashboards
Create and manage dashboards to visualize your data
Search for dashboards and folders
New
Filter by tag
Starred
Sort
General
Docker-cAdvisor
General
docker
Prometheus
NGINX
General
nginx
prometheus
nginx prometheus exporter
Node Exporter Full
General
linux

Tarefa 5. Siglas

Busca e traduce as seguintes siglas **relacionados coa UD**:

	Siglas	Significado	Tradución
1	**LPD**	Ley de Protección de Datos	Lei de Protección de Datos
2	**LPR**	Ley de Propiedad Registral	Lei de Propiedade Rexistral
3	**IPP**	Indicador Público de Renta de Efectos Múltiples	Indicador Público de Renda de Efectos Múltiples
4	**PDL**	Proyecto de Ley	Proxecto de Lei
5	**CUPS**	Código Universal del Punto de Suministro	Código Universal do Punto de Subministración
6	**PDF**	Portable Document Format	Formato de Documento Portátil
7	**XPS**	XML Paper Specification	Especificación de Papel XML