Grid y Computación de Altas Prestaciones
GRyCAP

Prácticas.
Contenedores Docker (I)

Máster Universitario en Computación en la Nube y de Altas Prestaciones

Objetivos

Grid y Computación de Altas Prestaciones GRyCAP

- Instalar un **entorno de trabajo (Docker-CE)** para la implementación y gestión de contenedores.
- Entender cómo usar de forma apropiada las funcionalidades básicas proporcionadas por la **tecnología docker** para gestionar:
  - **procesos (batch)** empaquetados en contenedores**.**
  - **los puertos** de los contenedores para las comunicaciones.
- Probar y testear dichas opciones a través de un host local desplegado en la nube mediante Docker-CE.

Contenidos

Grid y Computación de Altas Prestaciones GRyCAP

P1. Instalar Docker-CE (Community Edition).

P2. Gestión Básica de Imágenes.

P3. Gestión Básica de Contenedores.

P4. Gestión de Puertos en Contenedores.

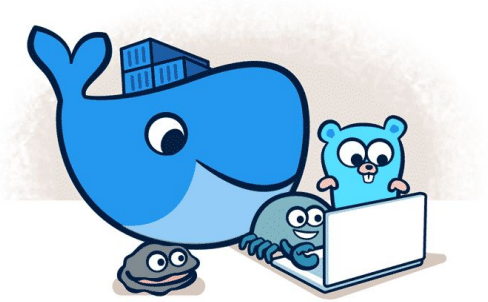
PA1. Solver Octave Básico.

PA2. Servicio Web Tomcat.

P1. Instalar Docker-CE

Grid y Computación de Altas Prestaciones
GRyCAP

- En esta práctica vas a desplegar una **máquina virtual (MV) en Microsoft Azure** sobre la que trabajar.
- Sobre la MV instalarás **Docker-CE (Community Edition)** para la implementación y gestión de contenedores.
  - se trata de un entorno de desarrollo y ejecución de contenedores **gratuito**.
  - te permitirá crear, testear, depurar, empaquetar y desplegar contenedores con la **tecnología Docker.**

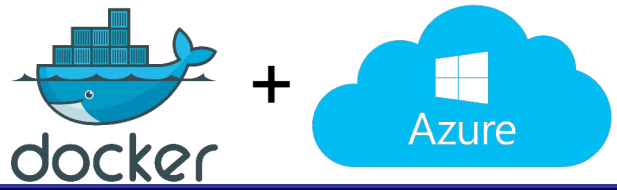


P1. Instalar Docker-CE
Desplegando una MV

Grid y Computación de Altas Prestaciones GRyCAP

- Despliega una Máquina Virtual en Windows Azure con las siguientes características:

Grupo de recursos: Crea un grupo con nombre "gr<usuario_upvnet> "(Ej. gr_josegqui) o reutiliza el que ya tengas
Nombre Máquina Virtual: MVdocker-<usuario_upvnet> (Ej. MVdocker-josegqui).
Región: Spain Central (selecciona otra región de europa en caso de error por límite de cuotas, p.ej. *West Europe*).
Opciones de disponibilidad: No se requiere redundancia de la infraestructura.
Tipo de seguridad: Estándar.
Imagen: Ubuntu Server 22.04 LTS - x64 gen. 2.
Arquitectura de VM: x64.
Instancia de Azure spot: No (Opción desmarcada).
Tamaño: Standard_B2s (Tamaño Standard_B2s - 2 vcpu, 4 GiB de memoria).
Usuario y contraseña o claves SSH (más seguro)
Permitir puerto de entrada ssh
Tamaño del disco del SO: Valor predeterminado de la imagen (30 GiB)
Tipo de Disco: HDD
Habilita "Eliminar IP pública y NIC cuando se elimine la VM" en el apartado de "Redes".
Deshabilita cualquier opción de supervisión.



P1. Instalar Docker-CE
Pasos para instalar Docker Sobre Ubuntu

Grid y Computación de Altas Prestaciones
GRyCAP

- Primero actualiza los paquetes existentes de tu MV mediante la herramienta ***apt-get*** :

```
$ sudo apt-get update
```

- Para instalar Docker-CE es necesario instalar algunos paquetes como requisito previo, que permitan a APT usar paquetes a través de HTTPS:

```
$ sudo apt-get install ca-certificates curl
```

- Para instalar el repositorio oficial de Docker se debe de instalar la clave de GPG en la MV:

```
$ sudo install -m 0755 -d /etc/apt/keyrings
$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
$ sudo chmod a+r /etc/apt/keyrings/docker.asc
```

Nota: también puedes seguir la guía oficial de instalación: https://docs.docker.com/engine/install/

P1. Instalar Docker-CE
Pasos para instalar Docker Sobre ubuntu

Grid y Computación de Altas Prestaciones
GRyCAP

- Agrega el repositorio de Docker a las fuentes de ***APT***:

```
$ echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

- Actualiza la base de datos de los paquetes de Docker en el repositorio recién agregado:

```
$ sudo apt-get update
```

P1. Instalar Docker-CE
Pasos para instalar Docker Sobre ubuntu

Grid y Computación de Altas Prestaciones GRyCAP

- Instala el Docker-CE (Engine y client) en la MV:

```
$ sudo apt-get install docker-ce docker-ce-cli
```

- Testea que se ha instalado correctamente y que el Servicio está *active* y en ejecución, listo para utilizarse.

```
$ sudo systemctl status docker
```

```
amcaar@MVdocker-amcaar:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2024-10-20 20:00:33 UTC; 28min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 3307 (dockerd)
      Tasks: 9
     Memory: 26.4M
        CPU: 440ms
     CGroup: /system.slice/docker.service
             └─3307 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

P1. Instalar Docker-CE
Pasos para instalar Docker Sobre ubuntu

Grid y Computación de Altas Prestaciones
GRyCAP

- Puedes probar a ejecutar un contenedor de prueba, donde se descargará una nueva imagen de contenedor, se ejecutará un contenedor docker y se imprimirá un mensaje de bienvenida:

```
$ sudo docker run hello-world
```

```
amcaar@MVdocker-amcaar:~$ sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

P1. Instalar Docker-CE

Grid y Computación de Altas Prestaciones GRyCAP

- Aunque este último paso no sea necesario, es conveniente por practicidad para evitar utilizar ***sudo*** en todos los comandos docker.
- Se debe de agregar el usuario de la MV al grupo docker (creado en el proceso de instalación). Posteriormente, tendremos que reestablecer el Servicio.

```
$ sudo usermod -aG docker <tu_usuario>
```

- Comprueba que tu usuario de la MV se ha agregado al grupo docker

```
$ id -nG <tu_usuario>
```

- Reestablece el Servicio docker y la sesión de usuario.

```
$ sudo service docker restart
```

- Realiza un *Logout*, y vuelve a entrar en la sesión.

P1. Instalar Docker-CE
Evidencias

Grid y Computación de Altas Prestaciones
GRyCAP

- Al final de esta actividad práctica deberás de tener instalado **Docker-CE** en la MV que has desplegado en la nube.
- Verifica que la instalación ha sido realizada con éxito ejecutando los siguientes comandos:

```
$ sudo docker --version
$ sudo systemctl status docker
```

- Para el portafolio, hay que recoger las siguientes evidencias:
  - Captura de pantalla del terminal con la salida de los comandos anteriores.

P1. Instalar Docker-CE
Evidencias

- Ejemplo documento Portafolio:

Práctica 1

```
josegqui@MVdocker-josegqui:~$ sudo docker --version
Docker version 28.4.0, build d8eb465
```

```
josegqui@MVdocker-josegqui:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Tue 2025-09-16 07:04:45 UTC; 26min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 922 (dockerd)
      Tasks: 8
     Memory: 502.5M (peak: 549.6M)
        CPU: 8.222s
     CGroup: /system.slice/docker.service
             └─922 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Contenidos

Grid y Computación de Altas Prestaciones GRyCAP

P1. Instalar Docker-CE (Community Edition).

P2. Gestión Básica de Imágenes.

P3. Gestión Básica de Contenedores.

P4. Gestión de Puertos en Contenedores.

PA1. Solver Octave Básico.

PA2. Servicio Web Tomcat.

P2. Gestión Básica de Imágenes
Comandos Básicos Docker

Grid y Computación de Altas Prestaciones
GRyCAP

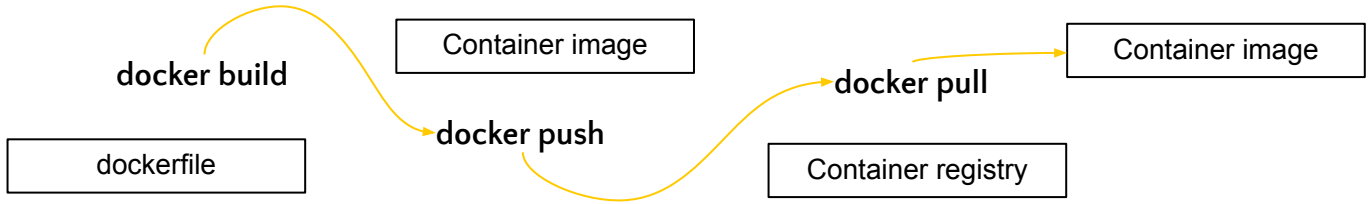
- El objetivo de esta práctica es:
  - entender cómo funcionan los **comandos** fundamentales de **Docker** para la **consulta, descarga y borrado en el registro local de imágenes**.
  - utilizar los comandos de Docker relacionados con la gestión básica de imágenes.

Uso

- Listado imagenes (docker images)
- Descarga desde Docker Hub (docker pull <imagen>)
- Eliminar imagen (docker rmi)
- Consulta listado (docker ps)
- Consulta Contenedor (docker Inspect)
- Crear Contenedor (docker create)
- Activación (docker start)
- Parada (docker stop)
- Ejecución (docker Run)
- Borrado (docker rm)
- Consulta puertos (docker port)
- ejecución (docker attach)
- Ejecución (docker exec)
- Volúmenes (docker cp)

Creación Imágenes

- Creación del contenedor
  - Vía creación del Dockerfile
    - Creación de la imagen (docker build).
  - Vía actualización
    - Descarga (docker pull) y actualización.
    - docker commit.
- Publicación (docker push)



P2. Gestión Básica de Imágenes
Docker images (Registro Imagenes Local)

Grid y Computación de Altas Prestaciones GRyCAP

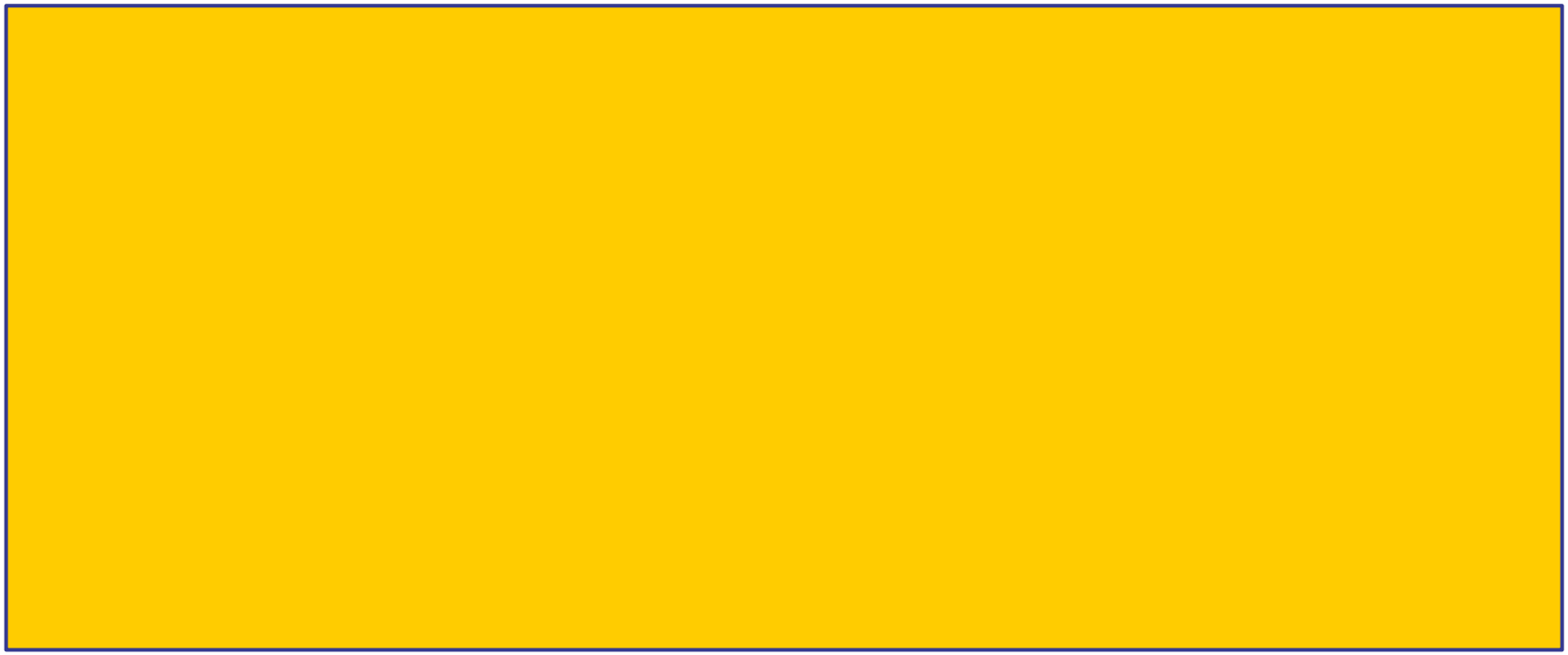
- ***docker images*** muestra las imágenes docker que hay en el registro local del host (en nuestro caso la MV de Azure donde habéis instalado el **Docker-CE)**.

docker images

P2. Gestión Básica de Imágenes
Docker images (Registro Imagenes Local)

Grid y Computación de Altas Prestaciones
GRyCAP

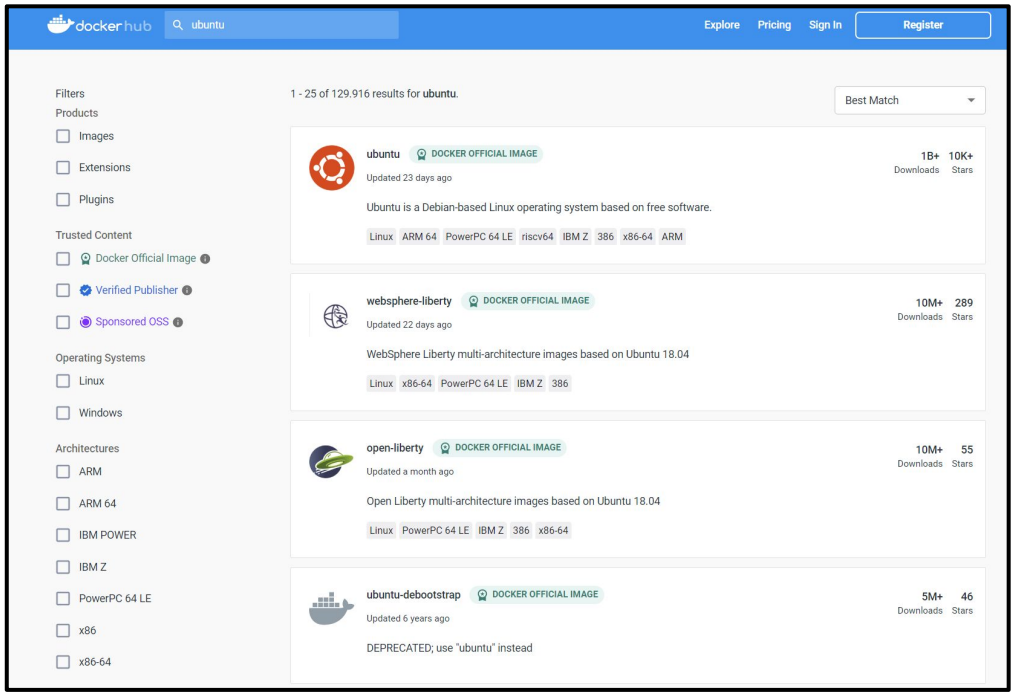
- **Tarea 2.1:** Muestra todas las imágenes existentes del registro del host. Deberías de tener la imagen de Hello-World.



P2. Gestión Básica de Imágenes
Registro Imágenes (Docker Hub)

Grid y Computación de Altas Prestaciones
GRyCAP

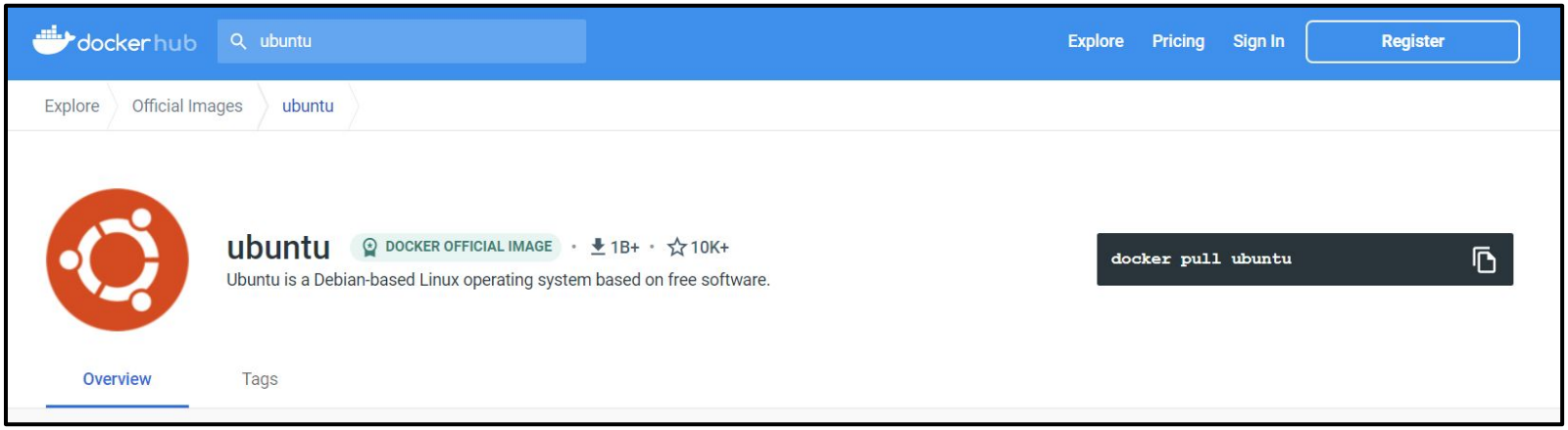
- Docker Hub (https://hub.docker.com) es un repositorio (registro de imágenes) en la nube de imágenes docker.
- Proporciona a los Docker-CE de imágenes de contenedores que pueden ser descargadas al registro local del host para su uso.



P2. Gestión Básica de Imágenes
Registro Imagenes (Docker Hub)

Grid y Computación de Altas Prestaciones
GRyCAP

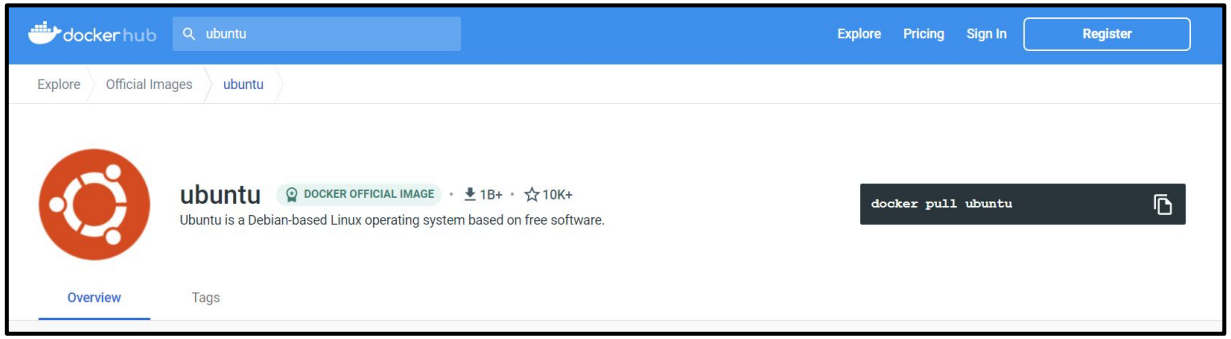
- En Docker hub existen imágenes ***públicas*** que pueden ser descargadas por cualquier usuario o bien imágenes de uso ***privado*** que solo podrán ser descargadas por usuarios registrados.



P2. Gestión Básica de Imágenes
Registro Imágenes (Docker Hub)
Docker pull

Grid y Computación de Altas Prestaciones GRyCAP

- ***docker pull*** descarga imágenes desde el repositorio de la nube ***Docker hub*** al repositorio local para su uso.
- Permite especificar qué versión concreta de una imagen queremos descargar a través de los TAGS.



docker pull <nombre_imagen>:<etiqueta/s>

- **Parámetros:**

<nombre_imagen> → Nombre de la imagen a descargar desde Docker Hub.

<etiqueta/s> → Etiqueta/s (metadato) asociado a la imagen a descargar desde Docker Hub. Si no se introduce etiqueta se asume "latest".

P2. Gestión Básica de Imágenes

Registro Imágenes (Docker Hub)

Docker pull

Grid y Computación de Altas Prestaciones GRyCAP

- **Tarea 2.2**: Descarga una imagen de **ubuntu** sin

¿que imagen descarga si no especificas etiqueta?

P2. Gestión Básica de Imágenes
Registro Imágenes (Docker Hub)
Docker pull

Grid y Computación de Altas Prestaciones GRyCAP

- Para descargar una versión concreta lo puedes especificar con los **tags.** Para ello puedes navegar por **Docker Hub** y buscar la versión que más te interese.

P2. Gestión Básica de Imágenes

Registro Imágenes (Docker Hub)

Docker pull

- **Tarea 2.3:** Descarga una imagen de Ubuntu versión focal (utiliza la etiqueta correspondiente)**.**

- **Tarea 2.4:** Descarga la última versión de una imagen de un servidor web nginx (no utilices etiqueta) desde **Docker Hub.**

P2. Gestión Básica de Imágenes
Docker rmi (Registro Imagenes Local)

Grid y Computación de Altas Prestaciones GRyCAP

- ***docker rmi*** elimina una imagen de contenedor del registro local.

docker rmi <opciones> <[nombre_imagen:etiqueta/s][IDImagen]>

- **Parámetros:**

  <nombre_imagen> → Etiqueta de la imagen a eliminar.

  <etiqueta/s> → Si no se indica este campo, por defecto se asume ":latest".

  <IDImagen> → Identificador de la imagen a eliminar.

- **Opciones**

  -f → fuerza a la eliminación (por ejemplo si un contenedor asociado a la imagen está en marcha).

P2. Gestión Básica de Imágenes
Docker rmi (Registro Imagenes Local)

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 2.5:** Elimina la imagen de hello-world**. Primero intentalo sin utilizar la opción -f.**

¿qué ocurre?

P2. Gestión Básica de Imágenes
Evidencias

Grid y Computación de Altas Prestaciones GRyCAP

- Al final de esta actividad práctica deberás de tener el entorno **Docker-CE** con una serie de imágenes de docker descargadas en tu registro local.
- Verifica que están todas las imágenes descargadas con:

```
josegqui@MVdocker-josegqui:~$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED         SIZE
ubuntu       latest    6d79abd4c962   6 days ago      78.1MB
nginx        latest    41f689c20910   4 weeks ago     192MB
ubuntu       focal     b7bab04fd9aa   5 months ago    72.8MB
```

- Para el portafolio, hay que recoger las siguientes evidencias:
  - Captura de pantalla del terminal con la salida del comando anterior donde aparezca tu usuario y las imágenes descargadas.
  - Texto donde indique la tarea y el comando o comandos utilizados.

P2. Gestión Básica de Imágenes
Evidencias

- Ejemplo documento Portafolio:

Práctica 2

```
josegqui@MVdocker-josegqui:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
ubuntu        latest    6d79abd4c962   6 days ago      78.1MB
nginx         latest    41f689c20910   4 weeks ago     192MB
ubuntu        focal     b7bab04fd9aa   5 months ago    72.8MB
```

- Tarea 2.1.
  <comando>
- Tarea 2.2.
  <comando>
- Tarea 2.3.
  <comando>
- Tarea 2.4.
  <comando>
- Tarea 2.5.
  <comando>

Contenidos

Grid y Computación de Altas Prestaciones GRyCAP

P1. Instalar Docker-CE (Community Edition).

P2. Gestión Básica de Imágenes.

P3. Gestión Básica de Contenedores.

P4. Gestión de Puertos en Contenedores.

PA1. Solver Octave Básico.

PA2. Servicio Web Tomcat.

P3. Gestión Básica de Contenedores
Comandos Básicos Docker

Grid y Computación de Altas Prestaciones
GRyCAP

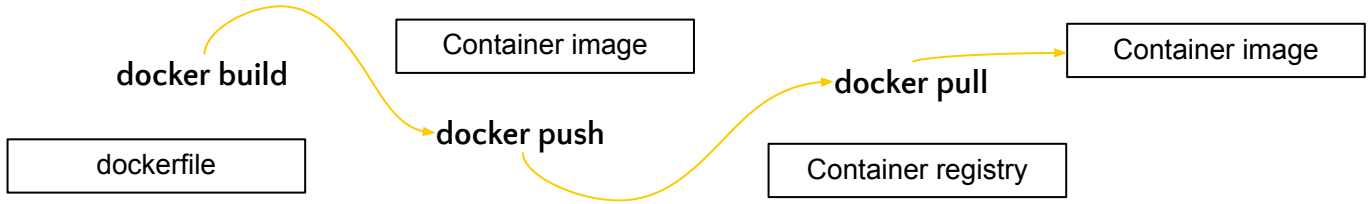
- El objetivo de esta práctica es:
  - entender cómo funcionan los **comandos** fundamentales de **Docker** para la consulta, creación, ejecución, parado y borrado de los contenedores.
  - utilizar los comandos de Docker para la gestión básica de contenedores.

Uso

- Listado imagenes (docker images)
- Descarga desde Docker Hub (docker pull <imagen>)
- Eliminar imagen (docker rmi)
- Consulta listado (docker ps)
- Consulta Contenedor (docker Inspect)
- Crear Contenedor (docker create)
- Activación (docker start)
- Parada (docker stop)
- Ejecución (docker Run)
- Borrado (docker rm)
- Consulta puertos (docker port)
- ejecución (docker attach)
- Ejecución (docker exec)
- Volúmenes (docker cp)

Creación Imágenes

- Creación del contenedor
  - Vía creación del Dockerfile
    - Creación de la imagen (docker build).
  - Vía actualización
    - Descarga (docker pull) y actualización.
    - docker commit.
- Publicación (docker push)



P3. Gestión Básica de Contenedores
Docker ps

Grid y Computación de Altas Prestaciones GRyCAP

- El comando ***docker ps*** muestra un listado de los contenedores existentes (**no de imágenes**), así como información básica acerca de estos, tales como su ID, nombre de contenedor, el nombre de la imagen sobre la que se basa el contenedor, la hora en la que se creó el contenedor o el estado en el que se encuentra (creado, en ejecución, parado etc….), entre otros.

docker ps <opciones>

- **Opciones:**

  -a → muestra de todos los contenedores la información asociada, independientemente de su estado. Si no se pone esta opción, solo muestra los que están en marcha (Up …).

P3. Gestión Básica de Contenedores
Docker ps

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 3.1:** Muestra los contenedores existentes junto con la información asociada.

```
josegqui@MVdocker-josegqui:~$ docker ps -a
CONTAINER ID   IMAGE         COMMAND    CREATED          STATUS                      PORTS     NAMES
ecb9ff309d34   hello-world   "/hello"   48 minutes ago   Exited (0) 48 minutes ago             unruffled_napier
```

P3. Gestión Básica de Contenedores
Docker inspect

Grid y Computación de Altas Prestaciones GRyCAP

- El comando ***docker inspect*** devuelve ***toda la información*** conocida en formato JSON sobre un objeto de Docker (Ej. Imágenes, contenedores etc…)

docker inspect <opciones> <[etiqueta_contenedor][IDContenedor][nombre_imagen][IDImagen]>

- **Parámetros:**

  <etiqueta_contenedor> → Etiqueta del contenedor a poner en marcha que se le asignó en el momento de su creación.

  <IDContenedor> → Identificador del contenedor (Completo o resumido)

- **Opciones**

  --format → sirve para filtrar y dar formato a la salida de los comandos usando plantillas de Go. Algunos filtros básicos de utilidad pueden ser:

  {{.Id}} → ID completo del contenedor.
  {{.Name}} → nombre.
  {{.State.Status}} → estado (running, exited, etc.).
  {{.Config.Image}} → imagen base.
  {{.NetworkSettings.IPAddress}} → IP interna.
  {{.NetworkSettings.Ports}} → puertos expuestos/publicados.
  {{.Config.Entrypoint}} → ENTRY POINT de una imagen o contenedor.
  {{.Config.Cmd}} → CMD de una imagen o contenedor.

  Ejemplos de uso: **docker inspect --format {{.Id}} 309e108ead56**

P3. Gestión Básica de Contenedores
Docker inspect

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 3.2:** Muestra toda la información del contenedor "hello-world".

P3. Gestión Básica de Contenedores
Docker inspect

- **Tarea 3.3:** Muestra la información relativa sólo al estado del contenedor "hello-world".

P3. Gestión Básica de Contenedores

Docker create

Conceptos ENTRYPOINT y CMD

Grid y Computación de Altas Prestaciones GRyCAP

- Las imágenes de contenedores creadas en un repositorio, vienen por lo general con una serie de parámetros predefinidos que hay que tener en cuenta a la hora de crear y ejecutar los contenedores. Destacamos los siguientes:

  - **ENTRYPOINT:** El ENTRYPOINT de una imagen Docker es el programa o proceso principal que Docker ejecuta automáticamente cuando inicias un contenedor creado a partir de esa imagen. Puede que el ENTRYPOINT de una imagen no esté definido y esté vacio.

  - **CMD:** Dependiendo de si hay definido un ENTRYPOINT o no, actúa de diferente manera.
    - Si hay definido un ENTRYPOINT, el CMD indican los argumentos que se le pasan al ENTRYPOINT.
    - Si NO hay definido un ENTRYPOINT, el CMD define el programa o proceso principal junto con los argumentos que se les quiera pasar.

- Los contenedores, cuando se crean, heredan estos parámetros predefinidos en las imágenes, aunque pueden ser modificados en el proceso de creación de contenedor.

- Tanto el **ENTRYPOINT** como el **CMD** de una imagen o contenedor lo podemos consultar con **docker inspect** de la siguiente manera:

  --format='{{.Config.Entrypoint}}' --format='{{.Config.Cmd}}'

P3. Gestión Básica de Contenedores
Docker create

Grid y Computación de Altas Prestaciones GRyCAP

- El comando ***docker create*** crea el contenedor a partir de una imagen que tengamos en el registro local (p.e. Una de las imágenes ubuntu o nginx descargadas previamente de **Docker Hub**).
- Si la imagen no está en el registro local, la busca en **Docker Hub** y la descarga **(Docker pull)** automáticamente si existe en este registro. En caso contrario, falla.
- Al crear el contenedor, **NO** lo pone en funcionamiento y se queda parado (estado *created*) a la espera que alguien lo lance y ponga en funcionamiento mediante algún comando específico (**docker start** o **docker run**).

docker create <opciones> <[nombre_imagen:etiqueta/s][IDImagen]> <comando>

- **Parámetros:**

<nombre_imagen:etiqueta> → Imagen de contenedor a utilizar para crear el contenedor. Si no está en el registro local, lo busca en docker Hub, lo descarga al registro local y después crea el contenedor.

<IDImagen> → Identificador de la imagen a eliminar.

<cmd> → Si se indica, se reescribe el CMD de la imagen. Sinó, el contenedor creado hereda el de la imagen.

- **Opciones**

--entrypoint → Si se indica, se reescribe el ENTRYPOINT de la imagen. Sinó, el contenedor creado hereda el de la imagen.

--name <nombre_contenedor> → Etiqueta que se le quiere dar al contenedor para su referencia dentro del entorno de trabajo ofrecido por Docker en el host.

P3. Gestión Básica de Contenedores
Docker create

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 3.4:** Consulta el ENTRYPOINT y el CMD de la imagen ubuntu:focal.

¿qué observas?

P3. Gestión Básica de Contenedores
Docker create

- **Tarea 3.5:** Crea un contenedor basándose en la imagen *ubuntu:focal*. Llama a este nuevo contenedor mi_con1. Reescribe el ENTRYPOINT de la imagen para que el contenedor ejecute el comando */bin/hostname.*

- Si consultamos los contenedores creados, **¿que valor te llama la atención de la columna STATUS y ID?**

- **¿que valor te llama la atención de la columna COMMAND?**

P3. Gestión Básica de Contenedores
Docker create

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 3.6:** Crea un contenedor basándose en la imagen *ubuntu:focal.* Llama a este nuevo contenedor mi_con2. Reescribe el ENTRYPOINT de la imagen para que el contenedor ejecute el comando */bin/hostname* y el CMD a "josegqui.mucnap.cc.upv.es".

- Si consultamos los contenedores creados, **¿que valor te llama la atención de la columna COMMAND?**

P3. Gestión Básica de Contenedores
Docker create

- **Tarea 3.7:** Crea un contenedor, llámalo mi_cont3. Debe de estar basado en la versión focal de ubuntu. Reescribe solo el CMD al comando **/bin/*hostname***.

- Si consultamos los contenedores, **¿que valores te llaman la atención de la columna COMMAND?**

- **¿Cual es el identificador del contenedor?**

P3. Gestión Básica de Contenedores
Docker create

- **Tarea 3.8:** Crea un contenedor basado en nginx (servidor web) y llámalo mi_con1_web. Verás que el estado del contenedor creado es “Created” al realizar esta acción. No sobrescribas ni el ENTRYPOINT y ni el CMD.

Relativo a la columna COMMAND, ¿qué observas?

Relativo a la columna STATUS, ¿qué observas?

¿Qué echas en falta en la salida de ps -a con la creación de este último “Docker create” relativo al servidor nginx?

P3. Gestión Básica de Contenedores
Docker start

Grid y Computación de Altas Prestaciones
GRyCAP

- ***docker start*** pone en funcionamiento un contenedor, **siempre en background (segundo plano),** que ejecuta un proceso asociado **(combinación ENTRYPOINT + CMD)** en su creación (**docker create**).
- El contenedor debe estar previamente **creado** (***docker create***) y **parado**.
- Al finalizar la ejecución del proceso asociado al contenedor, este pasa al estado a "Exited" e indica el tiempo desde que se paró la ejecución del contenedor la última vez.
- Un contenedor que está en estado "Exited", puede volverse a lanzar mediante **docker start** tantas veces como se desee.

docker start <opciones> <[nombre_contenedor][IDContenedor]>

- **Parámetros:**

  <nombre_contenedor> → Etiqueta del contenedor a poner en marcha que se le asignó en el momento de su creación.

  <IDContenedor> → Identificador del contenedor (Completo o resumido)

- **Opciones:**

  -a: captura la salida/error estándar (STDERR, STDOUT) del proceso asociado (comando) al contenedor. Si se quiere capturar la entrada estándar (STDIN) hay que crear el contenedor con una opción específica (opción -i en docker create).

P3. Gestión Básica de Contenedores
Docker start

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 3.9:** Arranca dos veces el contenedor mi_con1. Primero no muestres la salida estándar del contenedor, luego haz lo mismo mostrando la salida y error estandard (opción -a) de la ejecución del contenedor.

¿Qué observas en ambas salidas del comando docker start?

P3. Gestión Básica de Contenedores
Docker start

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 3.10:** Arranca el contenedor mi_con1_web y muestra la información del proceso.

- **¿Que observas en STATUS?**

- **En el contenedor mi_con1_web que está en marcha, ¿qué información nueva te llama la atención?**

P3. Gestión Básica de Contenedores
Docker start

Grid y Computación de Altas Prestaciones
GRyCAP

Tarea 3.11: El contenedor está en marcha escuchando en el puerto 80. Pon la la siguiente dirección web "**http://<IP_MV_de_Azure>/**". **¿Puedes acceder al servidor web nginx desde tu navegador local web?** en su caso **¿qué crees que te haría falta?**

P3. Gestión Básica de Contenedores
Docker stop

Grid y Computación de Altas Prestaciones GRyCAP

- El comando **docker stop** para un contenedor que esté en funcionamiento (STATUS up ..) y lo pasa a parado (STATUS Exited …).

docker stop <[nombre_contenedor][IDContenedor]>

- **Parámetros:**

  <nombre_contenedor> → Etiqueta del contenedor a poner en marcha que se le asignó en el momento de su creación.

  <IDContenedor> → Identificador del contenedor (Completo o resumido)

P3. Gestión Básica de Contenedores
Docker stop

Tarea 3.12: Para el contenedor mi_con1_web que tienes en marcha y muestra el estado del contenedor.

- **¿Qué observas en STATUS?**

P3. Gestión Básica de Contenedores
Docker run

Grid y Computación de Altas Prestaciones GRyCAP

- ***docker run*** es la ejecución de **docker create + docker start** en una única operación y de forma atómica.
- **Permite opciones que docker start no permite como la ejecución de un contenedor en primer plano.**
- ***docker run*** por defecto lanza en primer plano, aunque se puede forzar a lanzar el background (como docker start) utilizando una opción específica.

docker run <opciones> <nombre_imagen:etiqueta/s> <comando>

- **Opciones nuevas:**

  -d → Contenedor se ejecuta en background.

TODAS LAS OPCIONES y ARGUMENTOS DE docker create SON OPCIONES Y ARGUMENTOS DE docker run

P3. Gestión Básica de Contenedores
Docker run

- **Tarea 3.13:** Crea y lanza un nuevo contenedor mi_con3. El contenedor debe de estar basado en la imagen ubuntu focal y debe de ejecutar un proceso asociado al comando /bin/hostname.

- **¿Que ocurre?**

P3. Gestión Básica de Contenedores
Docker run

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 3.14:** Crea y lanza con **docker run** un contenedor *mi_con3_run* construido igual que el contenedor *mi_con3*.

- **Ejecuta con *docker start* el contenedor mi_con3 dos veces (con y sin la opción -a). Tanto mi_con3 y mi_con3_run ejecutan el mismo comando. ¿qué observas si comparas las dos salidas de docker start y la salida de docker run?**

P3. Gestión Básica de Contenedores
Docker run

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 3.15:** Crea y lanza con **docker run** un contenedor *mi_con3_run2* construido igual que el contenedor *mi_con3 y mi_con3_run*, utilizando la opción -d.

- **¿Qué ocurre?**

P3. Gestión Básica de Contenedores
Docker rm

Grid y Computación de Altas Prestaciones
GRyCAP

- ***docker rm*** elimina uno o más contenedores.

docker rm <opciones> <[nombre_contenedor][IDContenedor]>

- **Parámetros:**

  <nombre_contenedor> → Etiqueta del contenedor a poner en marcha que se le asignó en el momento de su creación.

  <IDContenedor> → Identificador del contenedor (Completo o resumido)

- **Opciones:**

  -f → fuerza la eliminación de un contenedor en marcha.

P3. Gestión Básica de Contenedores
Docker rm

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 3.16:** Elimina el contenedor mi_con1_web que tienes en marcha y muestra el estado del contenedor.

- **¿Has podido eliminar el contenedor?¿porque?**

P3. Gestión Básica de Contenedores
Evidencias

Grid y Computación de Altas Prestaciones
GRyCAP

- Al final de esta actividad práctica deberás de tener el entorno **Docker-CE** con una serie de contenedores.
- Verifica que están todos los contenedores descargadas con docker ps -a:

```
josegqui@MVdocker-josegqui:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS                          PORTS     NAMES
e958135b28ba   ubuntu:focal   "/bin/hostname"          About a minute ago   Exited (0) About a minute ago             mi_con3_run2
4394256e41f2   ubuntu:focal   "/bin/hostname"          5 minutes ago        Exited (0) 3 minutes ago                  mi_con3_run
fb06e155d0ca   ubuntu:focal   "/bin/hostname"          21 minutes ago       Created                                   mi_con3
c9498d3a048c   ubuntu:focal   "/bin/hostname joseg…"   23 minutes ago       Created                                   mi_con2
f078c8dc04db   ubuntu:focal   "/bin/hostname"          25 minutes ago       Exited (0) 13 minutes ago                 mi_con1
d2fd989ebc2f   hello-world    "/hello"                 About an hour ago    Exited (0) About an hour ago              competent_dijkstra
```

- Para el portafolio, hay que recoger las siguientes evidencias:
  - Captura de pantalla del terminal con la salida del comando anterior donde aparezca tu usuario.
  - Texto donde indique la tarea y el comando o comandos utilizados.

Contenidos

Grid y Computación de Altas Prestaciones GRyCAP

P1. Instalar Docker-CE (Community Edition).

P2. Gestión Básica de Imágenes.

P3. Gestión Básica de Contenedores.

P4. Gestión de Puertos en Contenedores.

PA1. Solver Octave Básico.

PA2. Servicio Web Tomcat.

P4. Gestión de Puertos en Contenedores
Comandos Básicos Docker

Grid y Computación de Altas Prestaciones
GRyCAP

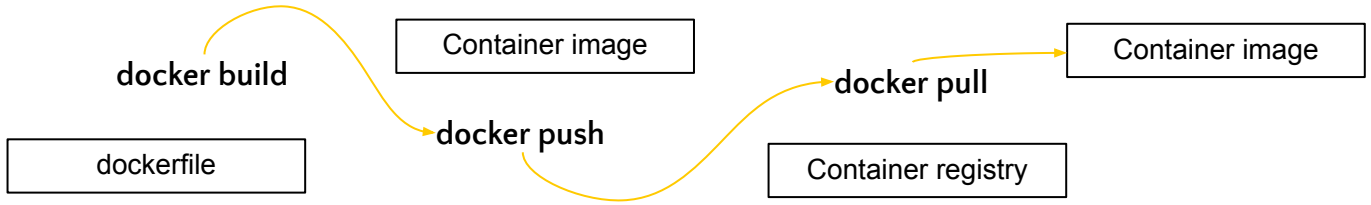
- El objetivo de esta práctica:
  - entender cómo funcionan los **comandos** fundamentales de **Docker** para la consulta y mapeo de puertos entre contenedores y el host.
  - utilizar los comandos de Docker para la gestión de puertos.

Uso

- Listado imagenes (docker images)
- Descarga desde Docker Hub (docker pull <imagen>)
- Eliminar imagen (docker rmi)
- Consulta listado (docker ps)
- Consulta Contenedor (docker Inspect)
- Crear Contenedor (docker create)
- Activación (docker start)
- Parada (docker stop)
- Ejecución (docker Run)
- Borrado (docker rm)
- Consulta puertos (docker port)
- ejecucion (docker attach)
- Ejecución (docker exec)
- Volúmenes (docker cp)

Creación Imágenes

- Creación del contenedor
  - Vía creación del Dockerfile
    - Creación de la imagen (docker build).
  - Vía actualización
    - Descarga (docker pull) y actualización.
    - docker commit.
- Publicación (docker push)



P4. Gestión de Puertos en Contenedores
Docker create

Grid y Computación de Altas Prestaciones GRyCAP

- ***docker create*** crea un contenedor a partir de una imagen que tengamos en el registro local.
- El comando ***docker run*** tiene implícito **docker create,** por lo que las opciones de gestión de puertos también se pueden utilizar en este comando de forma análoga.
- En tareas anteriores relativos al servidor web de nginx:
  - Se mostraba metainformación relativa a los puertos expuestos por el contenedor (no mapeos de puertos) **cuando estaba en marcha.**
  - No teníamos los puertos mapeados desde el host al contenedor.
- Para solucionar estos problemas, debemos de crear el contenedor de forma adecuada mediante "**docker create**" con las opciones específicas.

docker create <opciones> <nombre_imagen:etiqueta/s> <comando>

- Opciones nuevas para la gestión de puertos

  --expose → Indica el puerto expuesto por el contenedor, **pero NO mapea puertos**.

  --publish o -p <puerto_host>:<puerto_contenedor> → Mapea un puerto del host indicado al puerto del contenedor indicado. Se utiliza un [-p][--publish] por cada puerto mapeado.

  -P → **publica automáticamente todos los puertos EXPOSE** en puertos aleatorios del host.

P4. Gestión de Puertos en Contenedores
Docker create

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 4.1:** Crea un contenedor basado en nginx (servidor web). Llámalo mi_con2_web indicando que el puerto expuesto del contenedor es el 80 y que realice un mapeo del puerto 8080 del host al puerto 80 del contenedor. Después muestra el contenedor.

Relativo al campo PORTS, ¿que observas?

P4. Gestión de Puertos en Contenedores
Docker create

- **Tarea 4.2:** Arranca el contenedor basado en nginx (servidor web) mi_con2_web con *docker start* y muestra la información del contenedor.

- **¿qué observas relativo a los puertos?**

P4. Gestión de Puertos en Contenedores
Docker create

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 4.3:** **Accede al servidor web desde tu navegador local "http://<IP_MV_de_Azure>:8080"¿Puedes acceder al servidor web nginx desde tu navegador local web?**

P4. Gestión de Puertos en Contenedores
Docker create

Tarea 4.4: Modifica el contenedor basado en nginx (servidor web) de la tarea anterior (mi_con2_web)añadiendo el mapeo del puerto 8043 del host al puerto 443 del contenedor.

- **¿Qué ocurre?**

P4. Gestión de Puertos en Contenedores
Docker create/start

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 4.5:** Crea un contenedor basado en nginx (servidor web). Llámalo mi_con3_web indicando que los puertos expuestos del contenedor sean el 80 y el 443, y que realice un mapeo de los puertos 8080 y 8443 del host a los puertos 80 y 443 del contenedor. Después arranca y muestra el contenedor.

- **¿Qué ocurre?**

P4. Gestión de Puertos en Contenedores
Docker create/start

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 4.6:** Haz los pasos pertinentes para poder arrancar el contenedor mi_con3_web y poder acceder desde tu navegador local.

¿Puedes acceder desde el navegador desde el puerto 8443? ¿por qué?

P4. Gestión de Puertos en Contenedores
Docker create/start

Grid y Computación de Altas Prestaciones
GRyCAP

- **Tarea 4.7:** Crea y arranca un contenedor mi_con4_web que exponga los puertos 80 y 443 como los ejercicios anteriores, pero que mapee a puertos libres del host de forma aleatoria.

- **¿que ocurre con los puertos?**

P4. Gestión de Puertos en Contenedores
docker inspect/ports

Grid y Computación de Altas Prestaciones
GRyCAP

- Ya conocemos el comando ***docker inspect,*** el cual permite consultar toda la información disponible de un objeto docker.
- Puedes filtrar con la opción ***--format {{.NetworkSettings.Ports}}*** los puertos expuestos y mapeados por un contenedor.

```
docker inspect --format '{{.NetworkSettings.Ports}}' <[NombreContenedor][IDcontenedor/]>
```

- Del mismo modo, el comando ***docker ports*** muestra la misma información.

```
docker ports <[NombreContenedor][IDContenedor]>
```

P4. Gestión de Puertos en Contenedores
Docker inspect/port

- **Tarea 4.8:** Muestra la información relativa a los puertos en el contenedor mi_con3_web con los comandos *inspect* y *port*

P4. Gestión de Puertos en Contenedores
Evidencias

Grid y Computación de Altas Prestaciones
GRyCAP

- Al final de esta actividad práctica deberás de tener el entorno **Docker-CE** con una serie de contenedores creados.
- Verifica que están todos los contenedores con docker ps -a:

```
josegqui@MVdocker-josegqui:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS                      PORTS                                                                                  NAMES
987a610f63f1   nginx:latest   "/docker-entrypoint.…"   3 minutes ago    Up 3 minutes                0.0.0.0:32768->80/tcp, [::]:32768->80/tcp, 0.0.0.0:32769->443/tcp, [::]:32769->443/tcp   mi_con4_web
b0bb2087b4bb   nginx:latest   "/docker-entrypoint.…"   7 minutes ago    Up 5 minutes                0.0.0.0:8080->80/tcp, [::]:8080->80/tcp, 0.0.0.0:8443->443/tcp, [::]:8443->443/tcp     mi_con3_web
03997d24aba4   nginx:latest   "/docker-entrypoint.…"   14 minutes ago   Exited (0) 5 minutes ago                                                                                   mi_con2_web
e958135b28ba   ubuntu:focal   "/bin/hostname"          20 minutes ago   Exited (0) 20 minutes ago                                                                                  mi_con3_run2
4394256e41f2   ubuntu:focal   "/bin/hostname"          24 minutes ago   Exited (0) 22 minutes ago                                                                                  mi_con3_run
fb06e155d0ca   ubuntu:focal   "/bin/hostname"          39 minutes ago   Created                                                                                                    mi_con3
c9498d3a048c   ubuntu:focal   "/bin/hostname joseg…"   41 minutes ago   Created                                                                                                    mi_con2
f078c8dc04db   ubuntu:focal   "/bin/hostname"          44 minutes ago   Exited (0) 32 minutes ago                                                                                  mi_con1
d2fd989ebc2f   hello-world    "/hello"                 2 hours ago      Exited (0) 2 hours ago                                                                                     competent_dijkstra
```

- Para el portafolio, hay que recoger las siguientes evidencias:
  - Captura de pantalla del terminal con la salida del comando anterior donde aparezca tu usuario.
  - Texto donde indique la tarea y el comando o comandos utilizados.

Contenidos

Grid y Computación de Altas Prestaciones GRyCAP

P1. Instalar Docker-CE (Community Edition).

P2. Gestión Básica de Imágenes.

P3. Gestión Básica de Contenedores.

P4. Gestión de Puertos en Contenedores.

PA1. Solver Octave Básico.

PA2. Servicio Web Tomcat.

PA1. Contenedor Para Ejecuciones Solver Octave Básico

Grid y Computación de Altas Prestaciones GRyCAP

- Monta una solución a través de un contenedor (llámalo solver_octave_basico) que te permita ejecutar la siguiente operación matemática a través de GNU Octave:

  $\sqrt{25}$ + sen(π/2).

- Para ello sigue los siguientes pasos:
  - Busca en docker Hub una imagen de contenedor que pueda servirte para realizar el cálculo con la última versión de GNU Octave.
  - Crea y configura un contenedor para que ejecute esta operación matemática desde octave cuando arranque.
  - Lanza el contenedor para que muestre el resultado por pantalla. El resultado debería ser 6.
- **La evidencia va a ser el propio contenedor creado.**

Contenidos

Grid y Computación de Altas Prestaciones GRyCAP

P1. Instalar Docker-CE (Community Edition).

P2. Gestión Básica de Imágenes.

P3. Gestión Básica de Contenedores.

P4. Gestión de Puertos en Contenedores.

PA1. Solver Octave Básico.

PA2. Servicio Web Tomcat.

PA2. Servicio Web Tomcat

- Monta una solución a través de un contenedor (llámalo servidor_web_tomcat) que ofrezca un servicio web que escuche en el puerto 8000 para servir peticiones http.
- Para ello sigue los siguientes pasos:
  - Busca en docker Hub una imagen oficial de tomcat que pueda servirte.
  - Crea un contenedor para que lance el servicio web tomcat y lo exponga en el contenedor en el puerto 8000.
- Desde tu ordenador local ejecuta un curl (curl http://<IP Pública>:8000) o con el navegador web accede a dicha URL.
- **La evidencia va a ser el propio contenedor creado.**