

Uso de comandos en la terminal:

Paso	Comando
1.	<code>docker search ubuntu</code>
2.	<code>docker pull ubuntu</code>
3.	<code>docker pull ubuntu</code>
4.	<code>docker images</code>
5.	<code>docker run ubuntu echo "hello world"</code> <code>docker run hello-world</code>
6.	<code>docker ps</code>
7.	<code>docker run -it ubuntu bash</code>
8.	<code>exit</code>
9.	<code>docker exec -it &lt;ID_del_Contenedor&gt; bash</code>
10.	<code>docker start &lt;ID_del_Contenedor&gt;</code>
11.	<code>docker exec -it &lt;ID_del_Contenedor&gt; bash</code>
12.	<code>ls</code>
13.	<code>cat /etc/os-release</code>
14.	<code>touch andrei_archivo_prueba.txt</code>
15.	<code>echo "Hola, mundo de DAM/DAW, de parte de Andrei!" &gt; andrei_archivo_prueba.txt</code>
16.	<code>cat andrei_archivo_prueba.txt</code>
17.	<code>echo "Otra línea de texto en mi curso de Entornos" &gt;&gt; andrei_archivo_prueba.txt</code> <code>head andrei_archivo_prueba.txt</code> <code>tail andrei_archivo_prueba.txt</code> <code>more andrei_archivo_prueba.txt</code> <code>less andrei_archivo_prueba.txt</code> <code>grep "Entornos" andrei_archivo_prueba.txt</code>
18.	<code>echo "Puedo conectarme y modificar archivos en mi contenedor ID 7dfae9d60cba de ubuntu con el nombre de keen_jepsen" &gt;&gt; andrei_archivo_prueba.txt</code>

Estos pasos cubren la mayoría de las operaciones básicas que puedes realizar con Docker, desde la búsqueda de imágenes hasta la creación, ejecución y manipulación de contenedores.

Paso 1. Buscar con la terminal:  
docker search ubuntu + enter

```
PS C:\Users\andri> docker search ubuntu
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
ubuntu	Ubuntu is a Debian-based Linux operating sys...	16683	[OK]	
websphere-liberty	WebSphere Liberty multi-architecture images ...	297	[OK]	
open-liberty	Open Liberty multi-architecture images based...	62	[OK]	
neurodebian	NeuroDebian provides neuroscience research s...	105	[OK]	
ubuntu-debootstrap	DEPRECATED; use "ubuntu" instead	52	[OK]	
ubuntu-upstart	DEPRECATED; as is Upstart (find other proces...	115	[OK]	
ubuntu/nginx	Nginx, a high-performance reverse proxy & we...	103		
ubuntu/squid	Squid is a caching proxy for the Web. Long-t...	70		
ubuntu/cortex	Cortex provides storage for Prometheus. Long...	4		
ubuntu/apache2	Apache, a secure & extensible open-source HT...	65		
ubuntu/kafka	Apache Kafka, a distributed event streaming ...	37		
ubuntu/prometheus	Prometheus is a systems and service monitori...	51		
ubuntu/bind9	BIND 9 is a very flexible, full-featured DNS...	64		
ubuntu/mysql	MySQL open source fast, stable, multi-thread...	54		
ubuntu/zookeeper	Zookeeper maintains configuration informatio...	12		
ubuntu/postgres	PostgreSQL is an open source object-relatio...	31		
ubuntu/redis	Redis, an open source key-value store. Long...	20		
ubuntu/grafana	Grafana, a feature rich metrics dashboard & ...	9		
ubuntu/jre	Distroless Java runtime based on Ubuntu. Lon...	9		
ubuntu/dotnet-aspnet	Chiselled Ubuntu runtime image for ASP.NET a...	13		
ubuntu/memcached	Memcached, in-memory keyvalue store for sma...	5		
ubuntu/dotnet-deps	Chiselled Ubuntu for self-contained .NET & A...	11		
ubuntu/prometheus-alertmanager	Alertmanager handles client alerts from Prom...	9		
ubuntu/dotnet-runtime	Chiselled Ubuntu runtime image for .NET apps...	13		
ubuntu/cassandra	Cassandra, an open source NoSQL distributed ...	2		

Paso 2. Descargar Ubuntu :  
docker search Ubuntu

```
PS C:\Users\andri> docker search ubuntu
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
ubuntu	Ubuntu is a Debian-based Linux operating sys...	16683	[OK]	
websphere-liberty	WebSphere Liberty multi-architecture images ...	297	[OK]	
open-liberty	Open Liberty multi-architecture images based...	62	[OK]	
neurodebian	NeuroDebian provides neuroscience research s...	105	[OK]	
ubuntu-debootstrap	DEPRECATED; use "ubuntu" instead	52	[OK]	
ubuntu-upstart	DEPRECATED; as is Upstart (find other proces...	115	[OK]	
ubuntu/nginx	Nginx, a high-performance reverse proxy & we...	103		
ubuntu/squid	Squid is a caching proxy for the Web. Long-t...	70		
ubuntu/cortex	Cortex provides storage for Prometheus. Long...	4		
ubuntu/apache2	Apache, a secure & extensible open-source HT...	65		
ubuntu/kafka	Apache Kafka, a distributed event streaming ...	37		
ubuntu/prometheus	Prometheus is a systems and service monitori...	51		
ubuntu/bind9	BIND 9 is a very flexible, full-featured DNS...	64		
ubuntu/mysql	MySQL open source fast, stable, multi-thread...	54		
ubuntu/zookeeper	Zookeeper maintains configuration informatio...	12		
ubuntu/postgres	PostgreSQL is an open source object-relatio...	31		
ubuntu/redis	Redis, an open source key-value store. Long...	20		
ubuntu/grafana	Grafana, a feature rich metrics dashboard & ...	9		
ubuntu/jre	Distroless Java runtime based on Ubuntu. Lon...	9		
ubuntu/dotnet-aspnet	Chiselled Ubuntu runtime image for ASP.NET a...	13		
ubuntu/memcached	Memcached, in-memory keyvalue store for sma...	5		
ubuntu/dotnet-deps	Chiselled Ubuntu for self-contained .NET & A...	11		
ubuntu/prometheus-alertmanager	Alertmanager handles client alerts from Prom...	9		
ubuntu/dotnet-runtime	Chiselled Ubuntu runtime image for .NET apps...	13		
ubuntu/cassandra	Cassandra, an open source NoSQL distributed ...	2		

```
PS C:\Users\andri> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
aece8493d397: Pull complete
Digest: sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

Paso 3 . Instalar Ubuntu :  
docker pull Ubuntu

```
PS C:\Users\andri> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
aece8493d397: Pull complete
Digest: sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

Paso 4 . verificar las imagenes :  
docker images

```
PS C:\Users\andri> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mysql	latest	a3b6608898d6	2 weeks ago	596MB
ubuntu	latest	e4c58958181a	5 weeks ago	77.8MB
hello-world	latest	9c7a54a9a43c	6 months ago	13.3kB

```
PS C:\Users\andri>
```

Paso 5. Utilizar la imagen de ubunthu

para ejecutar el programa echo y ejecutar hello-word

```
PS C:\Users\andri> docker run ubuntu echo "hello world"
hello world
```

Paso 6. La ejecución de la imagen que hemos instalado. Cuando ejecutamos una imagen creamos un contenedor, (se puede dar la posibilidad de ejecutar múltiples veces la imagen), tenemos diferentes identificadores para ello.

docker ps

docker ps -a

```
PS C:\Users\andri> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
PS C:\Users\andri> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
985ea6a6cc5e   ubuntu    "echo 'hello world'"    17 minutes ago Exited (0) 6 minutes ago                thirsty_lehmann
5fb34be1aa9e   mysql:latest "docker-entrypoint.s..." 6 days ago    Exited (255) 34 minutes ago 3306/tcp, 33060/tcp mi-mysql
4ae476a72472   hello-world "/hello"                6 days ago    Exited (0) 2 days ago                nostalgic_gould
76b2359b09b0   hello-world "/hello"                2 weeks ago   Exited (0) 2 weeks ago                cool_lichterman
```

Por ejemplo ejecutas la imagen de Ubuntu una vez y tenemos un ID, ejecutamos otra vez y obtendremos otra ID ( del otro contenedor), al final tendremos múltiples procesos y cada uno se diferenciará con su propio ID de su contenedor.

Paso 7. Iniciamos un contenedor Docker basado en la imagen de Ubuntu y abriremos un Shell Bash dentro de ese contenedor.

docker run -it Ubuntu bash

```
PS C:\Users\andri> docker run -it ubuntu bash
root@7dfae9d60cba:/# |
```

Este comando que has ejecutado muestra que has iniciado un contenedor Docker basado en la imagen de Ubuntu y has abierto un shell Bash dentro de ese contenedor.

Desglosemos cada parte del comando:

1. **docker run**: Este comando se utiliza para crear y ejecutar un contenedor a partir de una imagen de Docker.
2. **-it**: Estas opciones son para ejecutar el contenedor en modo interactivo (**-i**) y asignarle un terminal (**-t**). Esto permite que puedas interactuar directamente con el shell del contenedor.
3. **ubuntu**: Especifica la imagen de Docker que se utilizará para crear el contenedor. En este caso, estás utilizando la imagen oficial de Ubuntu.
4. **bash**: Este es el comando que se ejecutará dentro del contenedor. En lugar de ejecutar el comando predeterminado de la imagen, que podría ser simplemente iniciar un servicio, estás especificando que desees iniciar un shell Bash interactivo dentro del contenedor.

Después de ejecutar este comando, la salida muestra que estás dentro del shell Bash del contenedor. La parte `root@7dfae9d60cba:/#` indica que el usuario actual es el usuario

root dentro del contenedor, y el prompt (#) sugiere que estás ejecutando comandos con privilegios de superusuario.

Paso 8. Salimos escribimos

*exit*

```
root@7dfae9d60cba:/# exit
exit
```

Paso 9. Volver a entrar en el contenedor:

Si has salido de un contenedor y deseas volver a entrar, puedes utilizar el siguiente comando:

...

```
docker exec -it <ID_del_Contenedor> bash
```

...

Reemplaza `*ID\_del\_Contenedor*` con el ID o el nombre del contenedor al que deseas entrar.

Por ejemplo, si tu contenedor tiene el ID *7dfae9d60cba*, el comando sería:

...

```
docker exec -it 7dfae9d60cba bash
```

...

Este comando utiliza `docker exec` para ejecutar un comando (en este caso, `bash`) dentro del contenedor especificado. La opción `-it` se utiliza para ejecutar el comando de manera interactiva y asignar un terminal.

Después de ejecutar este comando, estarás dentro del shell Bash del contenedor y podrás ejecutar comandos dentro de él. Cuando hayas terminado, puedes salir del contenedor con el comando `exit`.

Paso 10. Arrancamos el contenedor

```
docker start 985ea6a6cc5e
```

```
PS C:\Users\andri> docker start 985ea6a6cc5e
985ea6a6cc5e
```

Paso 11. Ejecutamos el contenedor. `docker exec -it <ID_del_Contenedor> bash`

En mi caso, antes he debido consultar y después ejecutar:

```
docker exec -it keen_jepsen bash
```

```
PS C:\Users\andri> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS              PORTS          NAMES
7dfae9d60cba   ubuntu    "bash"                  24 minutes ago Up 45 seconds           keen_jepsen
985ea6a6cc5e   ubuntu    "echo 'hello world'"    48 minutes ago Exited (0) 2 minutes ago          thirsty_lehmann
5fb34be1aa9e   mysql:latest "docker-entrypoint.s..." 6 days ago    Exited (255) About an hour ago    3306/tcp, 33060/tcp  mi-mysql
4ae476a72472   hello-world "/hello"                6 days ago    Exited (0) 2 days ago          nostalgic_gould
76b2359b09b0   hello-world "/hello"                2 weeks ago   Exited (0) 2 weeks ago          cool_lichterman

PS C:\Users\andri> docker exec -it keen_jepsen bash
root@7dfae9d60cba:/#
```

Paso 12. Estoy dentro del contenedor, así que vamos a  
Escribimos  
Ls

```
root@d4ae9d60cba:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
```

Esto significa que con el comando **ls** - has listado los directorios presentes en el sistema de archivos del contenedor. Estos son los directorios estándar de un sistema Linux.

Paso 13. Ver el contenido del archivo `/etc/os-release`

`cat /etc/os-release`

```
root@d4ae9d60cba:/# cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.3 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.3 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
```

Lo que has ejecutado, `cat /etc/os-release`, muestra el contenido del archivo `/etc/os-release` en el sistema de archivos del contenedor. Este archivo proporciona información sobre la distribución del sistema operativo.

Abajo podemos ver el desglose de los campos clave en la salida:

- **PRETTY\_NAME**: Proporciona una descripción amigable y legible para humanos de la versión del sistema operativo. En este caso, es "Ubuntu 22.04.3 LTS".
- **NAME**: Es el nombre de la distribución del sistema operativo. En este caso, "Ubuntu".
- **VERSION\_ID**: Proporciona la versión específica del sistema operativo. Aquí, es "22.04".
- **VERSION**: Ofrece una descripción más detallada de la versión del sistema operativo. En este caso, es "22.04.3 LTS (Jammy Jellyfish)".
- **VERSION\_CODENAME**: Es el nombre en código de la versión específica del sistema operativo. En este caso, "jammy".
- **ID**: Es la identificación única de la distribución del sistema operativo. Aquí, "ubuntu".

- **ID\_LIKE**: Indica qué otras distribuciones de Linux son similares a esta. En este caso, "debian".
- **HOME\_URL**, **SUPPORT\_URL**, **BUG\_REPORT\_URL**: Proporcionan enlaces a la página de inicio, soporte y reporte de errores de la distribución, respectivamente.
- **PRIVACY\_POLICY\_URL**: Proporciona el enlace a la política de privacidad de la distribución.
- **UBUNTU\_CODENAME**: Es el nombre en código de la versión de Ubuntu. En este caso, "jammy".

En resumen, la salida de `cat /etc/os-release` proporciona información detallada sobre la versión y la distribución del sistema operativo Ubuntu que se está ejecutando en el contenedor.

## Seguir los pasos personalizar los resultados para vuestro terminal:

Paso 14. Cada alumno pondrá su nombre.

Crear un archivo. `touch mi_archivo.txt`

`touch andrei_archivo_prueba.txt`

Paso 15. Ver el listado de archivos y comprobar la creación de mi archivo.

Con el comando

`ls`

```
root@7d+ae9d60cba:/# ls
andrei_archivo_prueba.txt bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
```

Paso 16. Modificar mi archivo creado.

Crear y manipular un archivo de texto, y luego visualizar su contenido. Vamos a trabajar con el archivo que has creado, "andrei\_archivo\_prueba.txt". Asumiendo que estás dentro del contenedor, aquí tienes algunos pasos:

### 1. **Agregar Contenido al Archivo:**

Utiliza `echo` para agregar contenido al archivo. Ejecuta:

...

`echo "Hola, mundo de DAM/DAW, de parte de Andrei!" > andrei_archivo_prueba.txt`

...

### 2. **Visualizar el Contenido del Archivo:**

Puedes usar el comando `cat` para visualizar el contenido del archivo

`cat andrei_archivo_prueba.txt`

```
root@7d+ae9d60cba:/# echo "Hola, mundo de DAM/DAW, de parte de Andrei!" > andrei_archivo_prueba.txt
root@7d+ae9d60cba:/# cat andrei_archivo_prueba.txt
Hola, mundo de DAM/DAW, de parte de Andrei!
```

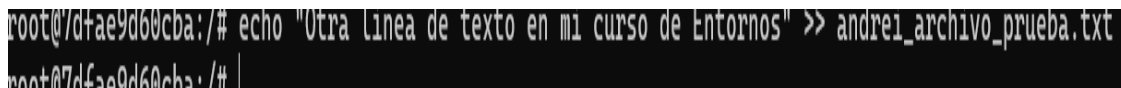
3. Usar el comando ``echo`` en combinación con redirección (`>>`) para agregar contenido al final de un archivo. Además, puedes experimentar con varios comandos de edición y visualización utilizando ``echo``. Aquí hay algunos ejemplos adicionales:

1. **\*\*Agregar otra línea al archivo:\*\***

Utiliza ``echo`` con redirección para agregar otra línea al final del archivo:

...

`echo "Otra línea de texto en mi curso de Entornos" >> andrei_archivo_prueba.txt`

A terminal window with a black background and green text. The prompt is 'root@7dfae9d60cba:/' and the command entered is 'echo "Otra línea de texto en mi curso de Entornos" >> andrei\_archivo\_prueba.txt'. The cursor is at the end of the command line.

```
root@7dfae9d60cba:/# echo "Otra línea de texto en mi curso de Entornos" >> andrei_archivo_prueba.txt
root@7dfae9d60cba:/#
```

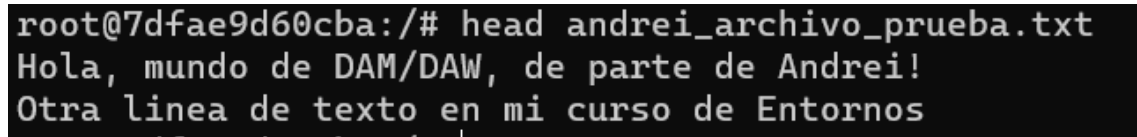
2. **\*\*Visualizar las primeras líneas del archivo:\*\***

Usa el comando ``head`` para mostrar las primeras líneas del archivo:

...

`head andrei_archivo_prueba.txt`

...

A terminal window with a black background and green text. The prompt is 'root@7dfae9d60cba:/' and the command entered is 'head andrei\_archivo\_prueba.txt'. The output shows three lines: 'Hola, mundo de DAM/DAW, de parte de Andrei!', 'Otra línea de texto en mi curso de Entornos', and a partially visible third line.

```
root@7dfae9d60cba:/# head andrei_archivo_prueba.txt
Hola, mundo de DAM/DAW, de parte de Andrei!
Otra línea de texto en mi curso de Entornos
root@7dfae9d60cba:/#
```

3. **\*\*Visualizar las últimas líneas del archivo:\*\***

Usa el comando ``tail`` para mostrar las últimas líneas del archivo:

```bash

`tail andrei_archivo_prueba.txt`

...

4. **\*\*Visualizar el archivo página por página:\*\***

Usa el comando ``more`` para visualizar el archivo página por página:

```bash

`more andrei_archivo_prueba.txt`

...

Puedes avanzar página por página presionando la barra espaciadora y salir presionando la tecla ``q``.

5. **\*\*Visualizar el archivo con paginación avanzada:\*\***

Usa el comando ``less`` para visualizar el archivo con paginación avanzada:

```
```bash
less andrei_archivo_prueba.txt
```
```

Puedes navegar hacia arriba y hacia abajo utilizando las teclas de flecha, y salir presionando la tecla `q`.

6. **\*\*Buscar texto en el archivo:\*\*** vamos a buscar palabra "Entornos"

Usa el comando `grep` para buscar líneas que contengan un texto específico:

```
```bash
grep "Entornos" andrei_archivo_prueba.txt
```
```

Esto imprimirá las líneas que contienen la palabra "Entornos" en el archivo.

```
root@7dfae9d60cba:/# grep "Entornos" andrei_archivo_prueba.txt
Otra línea de texto en mi curso de Entornos
```

7. Modificar archivo por la segunda vez:

```
echo "Puedo conectarme y modificar archivos en mi contenedor ID 7dfae9d60cba de ubuntu con el nombre de keen_jepsen " >> andrei_archivo_prueba.txt
```

```
root@7dfae9d60cba:/# echo "Puedo conectarme y modificar archivos en mi contenedor ID 7dfae9d60cba de ubuntu con el nombre de keen_jepsen " >> andrei_archivo_prueba.txt
```

8. buscar el nombre del contenedor que he puesto en el archivo en el paso anterior.

```
root@7dfae9d60cba:/# grep "keen_jepsen" andrei_archivo_prueba.txt
Puedo conectarme y modificar archivos en mi contenedor ID 7dfae9d60cba de ubuntu con el nombre de keen_jepsen
```

Estos son solo algunos ejemplos para que experimentes con diferentes comandos de terminal y veas cómo afectan el archivo.

Sube en aules las capturas de pantallas de tu terminal para cada paso.

Pone numero de paso, el comando y su resultado, demostrando que has logrado entender y practicar los comandos para esta practica de Docker.



