# Docker Commands

Contents

[1. Correr una imagen. 2](#_Toc33006946)

[2. Correr una imagen y entrar al bash 2](#_Toc33006947)

[3. Ver las imágenes 2](#_Toc33006948)

[4. Ver las containers que están en ejecución 3](#_Toc33006949)

[5. Ver los container que ya han sido terminados (Exited) 3](#_Toc33006950)

[6. Para el container 3](#_Toc33006951)

[7. Eliminar Containers 4](#_Toc33006952)

[8. Eliminar Imágenes 4](#_Toc33006953)

[9. Ingresar a un contenedor que ya está en ejecución en modo Bash. 4](#_Toc33006954)

[10. Copiar archivos a un contenedor linux 4](#_Toc33006955)

[11. Crear una imagen a partir de un contenedor 5](#_Toc33006956)

[12. Agregar TAG a una imagen recién creada de un contenedor 5](#_Toc33006957)

[13. Correr una imagen con Puerto 5](#_Toc33006958)

[14. Docker Build 6](#_Toc33006959)

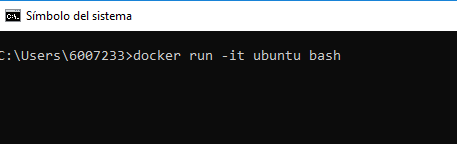
## Correr una imagen.

\>docker run **[Image Name/ Image Id]**

\>docker run ubuntu

## Correr una imagen y entrar al bash

\>docker run -it **[Image Name/ Image Id]** bash

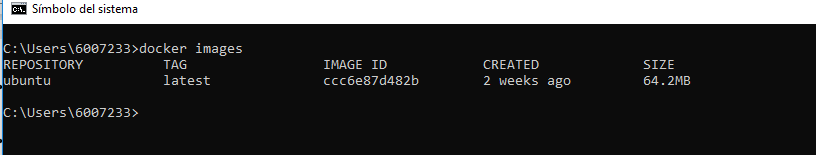


Ejemplo:

\> docker run -it -p 8082:8080 --name ubuntu\_tomcat2 img\_ubuntu\_tomcat bash

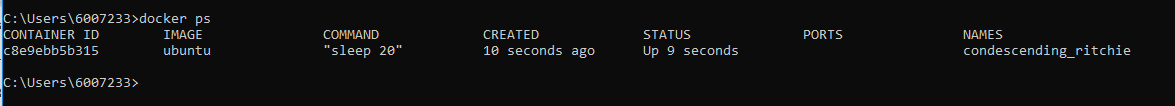
## Ver las imágenes

\>docker images



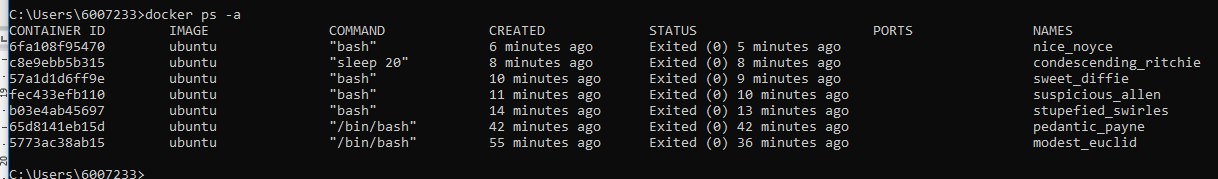
## Ver las containers que están en ejecución

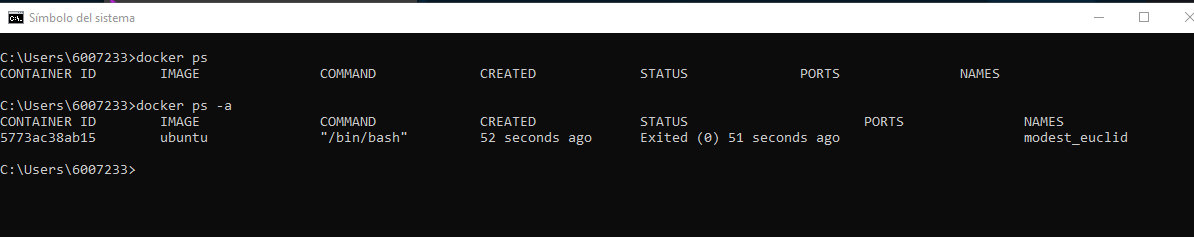
\>docker ps -a



## Ver los container que ya han sido terminados (Exited)

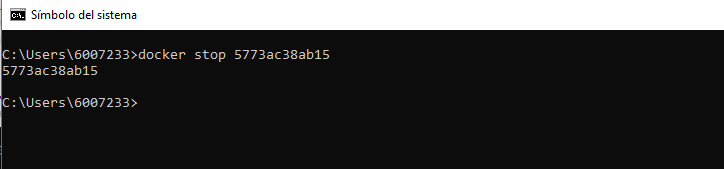
\>docker ps -a





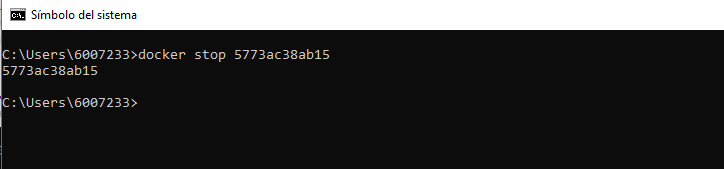
## Para el container

\>docker stop **[Container Id]**



## Eliminar Containers

\>docker rm **[Container Id]**

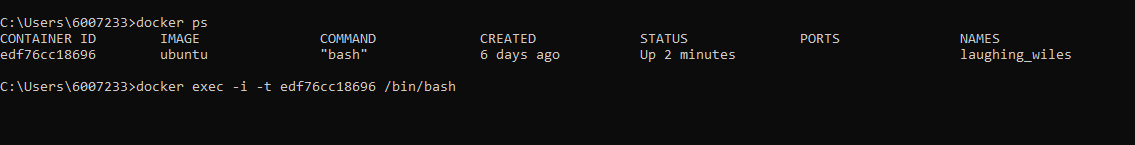


## Eliminar Imágenes

\>docker rmi **[Image Id]**

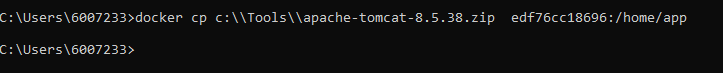
## Ingresar a un contenedor que ya está en ejecución en modo Bash.

\>docker exec -i -t **[Container Id]** /bin/bash



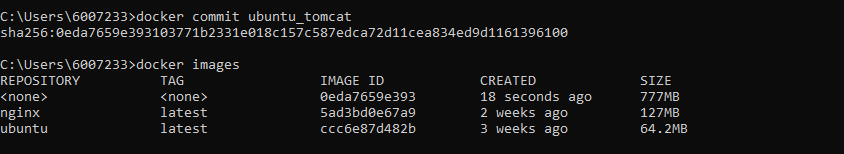
## Copiar archivos a un contenedor linux

\> docker cp **[Source File] [Container Id]**:**[path]**



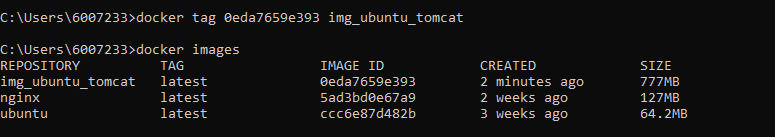
## Crear una imagen a partir de un contenedor

\>docker commit **[Container Name]**



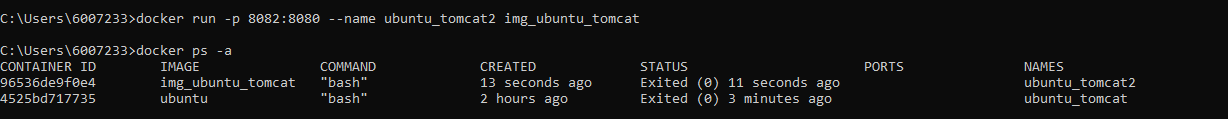
## Agregar TAG a una imagen recién creada de un contenedor

\>docker tag **[Image Id] [Tag to Add]**



## Correr una imagen con Puerto

\>docker run -p 8081:8080 --name ubuntu\_tomcat ubuntu



## Docker Build

Archive dockerfile

FROM java

CMD ["java", "-version"]

\>docker build -t hello-java .

* The docker build command builds the image. -t provides a name for the image. hello-java is the name of the image. “.” is the context for the command. This context is used as the base directory for copying any files to the image. No files are copied in this case.
* The java image is downloaded from Docker Hub. It also downloads the complete chain of base images.
* The CMD instruction added as a new layer to the image.