

# Getting started with Docker

---

Docker can package an application and its dependencies in a virtual container that can run on any host. Docker differs from virtualization since a Docker container does not need an operating system. Thus, a Docker container starts very quickly and takes less disk space than a virtual machine.

## Introduction

This tutorial will guide you to:

- Discover Docker commands
- Create and use a container

## Installation

Docker is available in many versions and for several uses:

- Desktop
- Cloud providers
- Server

This tutorial uses the Docker desktop

Follow the installation procedure: <https://www.docker.com/>

## Working with Docker images

Docker containers are built from Docker images. By default, Docker pulls these images from Docker Hub: <https://hub.docker.com/>

Among all the images, the hub contains a basic one named hello-world:  
[https://hub.docker.com/\\_/hello-world/](https://hub.docker.com/_/hello-world/)

To use this Docker image type this command:

```
docker run hello-world
```

The command

```
docker search
```

searches for images in the Docker hub. As an example search for the Ubuntu images with:

```
docker search ubuntu
```

Then download one of the images with:

```
docker pull ubuntu
```

Check the local images with:

```
docker images
```

You can then run a container in the interactive mode by combining i and t option with:

```
docker run -it ubuntu
```

A few remarks:

- As you can see, the container starts very quickly.
- Thanks to the interactive mode, you are inside the container! So you can interact with the Linux using the ls command, etc.
- Don't forget to exit the container with: exit

The command

```
docker rmi [imageID]
```

will remove an image, while

```
docker stop [Container ID]
```

will stop a running container. Get the list of container using:

- docker ps
- docker ps -a                      Show all containers (default shows just running)

```
docker start [Image ID or image name]
```

will start a stopped container.

Modify an image is easy. In the next section, you will install Java inside the Ubuntu container then deduce from it a new image. Start by updating Ubuntu with:

```
apt-get update && apt-get upgrade
```

then install Java with:

```
apt-get install default-jdk
```

In the end test the installation using the java command.

The updated container can be "saved" as a new image using commit:

```
docker commit [ContainerID] [NewImageName]
```

Fun and useful isn't it?

## **Publish a Docker image in the Docker hub**

*Docker Hub* is the world's largest library and community for container images:

<https://hub.docker.com/>

To publish yours images to the Docker hub, create an account first.

Then you should tag your local image with: `docker tag imageID  
yourDockerHubName/imageName:version`

Example: `docker tag 1dsd512s0d efrei/my-service:1`

Then login to docker before pushing your image: `docker login`

In the end, push your image with: `docker push yourDockerHubName/imageName:version`