

# Introduccion

---

## Containers

- Might be described as creating a small software package that can run a particular app and its associated processes

Container becomes portable and can run on all types of computers in the same way without compatibility issues and dependencies. This is why Docker refers to a container as a **Standardized unit of Software**.

### Benefits

- Not disruptive to a system
- Easy clean up afterwards
- No conflicting versions of software or libraries
- Lightweight: Optimized for a medium-powered laptop.
- Can run multiple containers on one server
- Orchestrators can help organize and run multiple containers, as Kubernetes or Openshift

### VM vs Containers

VMs might have different apps present on a single VM. There might be a web server and a database along with their backup scripts.

The container is using the underlying host computer OS, and not starting its own whenever it launches

### Keys

- Isolate containers from one another
- control groups which provide resource kernel quotas. Limit one container from using up all the resources, such as the CPU or disk reads and writes on the host machine.

### Dockerfile Basic Sample

In order to create a Docker image, a Dockerfile is required. This is a simple script that determines how an image is built from scratch.

Once you use Docker to build an image from a Dockerfile, you can run as many identical containers from it as you wish.

## Base Images and Layering

Base images are an important part of Docker and involve an image which you build your own image upon. Base images usually come with a tiny but functional OS.

Docker uses a layered filesystem that allows to add a base image without making your image much larger.

Whenever you want to create a container image, it will be used. Docker will know which base image to use when you specify a FROM line in the Dockerfile.

## Microservices

Docker containers are better suited to a microservice architecture than a traditional monolithic architecture because a container can run autonomously without relying on the other centralized processes.

```
FROM debian:stable
LABEL author=LinuxTrainingAcademy
LABEL Version=dockerbook

RUN apt update &&
    apt install -y nano &&
    apt clean

CMD ["/bin/bash"]
```