

# Instalación Docker

MARKEL ORALLO

## Versión instalada de Docker

Si ejecutamos el comando `docker --version`:

**Versión:** 20.10.7

**Build:** f0df350

```
markel@MacBook-Pro-de-Markel ~ % docker --version
Docker version 20.10.7, build f0df350
```

Si ejecutamos el comando `docker version`:

Nos muestra información detallada sobre la versión de docker que tenemos instalada tanto como para cliente como para servidor.

```
markel@MacBook-Pro-de-Markel ~ % docker version
Client:
 Cloud integration: 1.0.17
 Version:          20.10.7
 API version:      1.41
 Go version:       go1.16.4
 Git commit:       f0df350
 Built:            Wed Jun  2 11:56:22 2021
 OS/Arch:          darwin/amd64
 Context:          default
 Experimental:     true

Server: Docker Engine - Community
 Engine:
  Version:          20.10.7
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.13.15
  Git commit:       b0f5bc3
  Built:            Wed Jun  2 11:54:58 2021
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.4.6
  GitCommit:       d71fcd7d8303cbf684402823e425e9dd2e99285d
 runc:
  Version:          1.0.0-rc95
  GitCommit:       b9ee9c6314599f1b4a7f497e1f1f856fe433d3b7
 docker-init:
  Version:          0.19.0
  GitCommit:       de40ad0
 Kubernetes:
  Version:          Unknown
  StackAPI:         Unknown
markel@MacBook-Pro-de-Markel ~ %
```

## Información del sistema Docker en mi sistema

Por la parte del cliente nos muestra que esta en el contexto predeterminado, que no tiene habilitado el modo de depuración y que tiene instalados tres plugins.

```
markel@MacBook-Pro-de-Markel ~ % docker system info
Client:
 Context:      default
 Debug Mode:   false
 Plugins:
  buildx: Build with BuildKit (Docker Inc., v0.5.1-docker)
  compose: Docker Compose (Docker Inc., v2.0.0-beta.6)
  scan: Docker Scan (Docker Inc., v0.8.0)
```

Por la parte de servidor nos muestra más información de la que podemos destacar el numero de contenedores y su estado, numero de imágenes y versión del servidor de docker.

```
Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 14
 Server Version: 20.10.7
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 io.containerd.runtime.v1.linux runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: d71fcd7d8303cbf684402823e425e9dd2e99285d
 runc version: b9ee9c6314599f1b4a7f497e1f1f856fe433d3b7
 init version: de40ad0
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 5.10.25-linuxkit
 Operating System: Docker Desktop
 OSType: linux
 Architecture: x86_64
 CPUs: 8
 Total Memory: 1.94GiB
 Name: docker-desktop
 ID: EYMQ:KZ53:5Y57:ZADQ:HFE2:ZWMS:DLE3:ZYEG:2V7K:FWU2:VCR3:NAGM
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 HTTP Proxy: http.docker.internal:3128
 HTTPS Proxy: http.docker.internal:3128
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  127.0.0.0/8
 Live Restore Enabled: false
```

## Primera ejecución de un contenedor docker

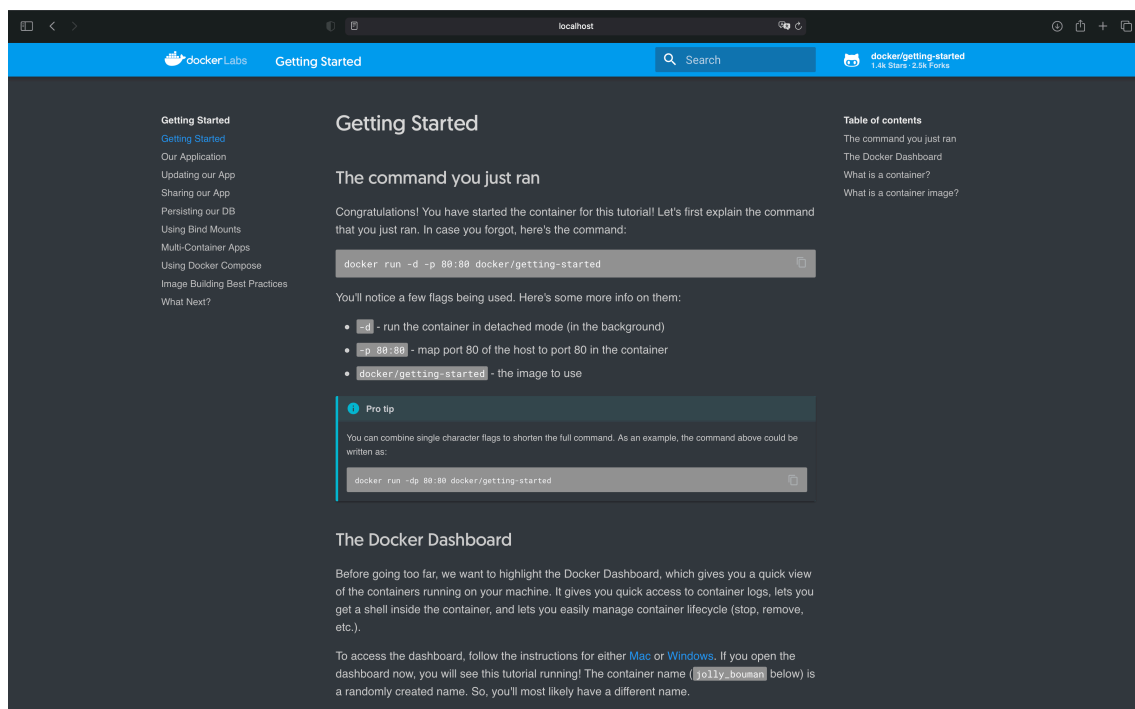
Para la primera ejecución de un contenedor docker ejecutamos el comando `docker run -p 80:80 -d docker/getting-started`

Donde la `-p` es para mapear el puerto 80 del host con el puerto 80 del contenedor y el `-d` para ejecutar el contenedor en segundo plano.

Como podemos ver en la salida, docker no encuentra la imagen en nuestro registro local y se la descarga mediante un pull de Docker Hub.

```
markel@MacBook-Pro-de-Markel ~ % docker run -p 80:80 -d docker/getting-started
Unable to find image 'docker/getting-started:latest' locally
latest: Pulling from docker/getting-started
540db60ca938: Pull complete
0ae30075c5da: Pull complete
9da81141e74e: Pull complete
b2e41dd2ded0: Pull complete
7f40e809fb2d: Pull complete
758848c48411: Pull complete
23ded5c3e3fe: Pull complete
38a847d4d941: Pull complete
Digest: sha256:10555bb0c50e13fc4dd965ddb5f00e948ffa53c13ff15dc8c85b7ab65e1f240b
Status: Downloaded newer image for docker/getting-started:latest
79f827efd8f79e7c1bb6cb159c0ef8656648124a18108f6340a9a8b9ebc51fc7
```

Una vez termina el comando accedemos a <http://localhost> y accedemos al contenedor que nos muestra información sobre como iniciarse en docker.



The screenshot shows the Docker 'Getting Started' page in a web browser. The page has a blue header with the Docker Labs logo and a search bar. The main content area is dark gray and contains the following sections:

- Getting Started**: A sidebar menu with links to 'Getting Started', 'Our Application', 'Updating our App', 'Sharing our App', 'Persisting our DB', 'Using Bind Mounts', 'Multi-Container Apps', 'Using Docker Compose', 'Image Building Best Practices', and 'What Next?'.
- The command you just ran**: A section with the command `docker run -d -p 80:80 docker/getting-started` in a code block. Below it, it says 'You'll notice a few flags being used. Here's some more info on them:' and lists the flags:
  - `-d` - run the container in detached mode (in the background)
  - `-p 80:80` - map port 80 of the host to port 80 in the container
  - `docker/getting-started` - the image to use
- Pro tip**: A section with a tip about combining flags to shorten the command. It shows the shortened command `docker run -dp 80:80 docker/getting-started` in a code block.
- The Docker Dashboard**: A section explaining the Docker Dashboard and how to access it. It mentions that the container name is randomly created and provides a link to the Docker Dashboard.

## Descarga de una imagen de Docker Hub

Para descargar imágenes de docker hub se utiliza el comando *docker pull imagen:versión\_de\_la\_imagen*.

En este caso particular descargaremos la ultima versión de la imagen de Ubuntu, por lo tanto el comando será *docker pull ubuntu:latest* .

```
markel@MacBook-Pro-de-Markel ~ % docker pull ubuntu:latest
latest: Pulling from library/ubuntu
35807b77a593: Pull complete
Digest: sha256:9d6a8699fb5c9c39cf08a0871bd6219f0400981c570894cd8cbea30d3424a31f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```