

OFFICIAL REPOSITORY

docker (/r/ /docker/) ☆

Last pushed: 16 days ago

[Repo Info \(/ /docker/\)](#)

## Short Description

Docker in Docker!

## Full Description

### Supported tags and respective Dockerfile links

- 18.09.0 , 18.09 , 18 , stable , test , latest (18.09/Dockerfile)  
(<https://github.com/docker-library/docker/blob/91bbc4f7b06c06020d811dafb2266bcd7cf6c06d/18.09/Dockerfile>)
- 18.09.0-dind , 18.09-dind , 18-dind , stable-dind , test-dind , dind  
(18.09/dind/Dockerfile) (<https://github.com/docker-library/docker/blob/91bbc4f7b06c06020d811dafb2266bcd7cf6c06d/18.09/dind/Dockerfile>)
- 18.09.0-git , 18.09-git , 18-git , stable-git , test-git , git  
(18.09/git/Dockerfile) (<https://github.com/docker-library/docker/blob/91bbc4f7b06c06020d811dafb2266bcd7cf6c06d/18.09/git/Dockerfile>)
- 18.06.1-ce , 18.06.1 , 18.06 , edge (18.06/Dockerfile)  
(<https://github.com/docker-library/docker/blob/fe2ca76a21fdc02cbb4974246696ee1b4a7839dd/18.06/Dockerfile>)
- 18.06.1-ce-dind , 18.06.1-dind , 18.06-dind , edge-dind  
(18.06/dind/Dockerfile) (<https://github.com/docker-library/docker/blob/fe2ca76a21fdc02cbb4974246696ee1b4a7839dd/18.06/dind/Dockerfile>)
- 18.06.1-ce-git , 18.06.1-git , 18.06-git , edge-git (18.06/git/Dockerfile)  
(<https://github.com/docker-library/docker/blob/595ad0c92090937dcb7c200900fb97e36d36c412/18.06/git/Dockerfile>)

### Quick reference

- **Where to get help:**  
[the Docker Community Forums \(https://forums.docker.com/\)](https://forums.docker.com/), [the Docker Community Slack \(https://blog.docker.com/2016/11/introducing-docker-community-directory-docker-community-slack/\)](https://blog.docker.com/2016/11/introducing-docker-community-directory-docker-community-slack/), or [Stack Overflow \(https://stackoverflow.com/search?tab=newest&q=docker\)](https://stackoverflow.com/search?tab=newest&q=docker)
- **Where to file issues:**  
<https://github.com/docker-library/docker/issues> (<https://github.com/docker-library/docker/issues>)
- **Maintained by:**  
[Tianon \(of the Docker Project\) \(https://github.com/docker-library/docker\)](https://github.com/docker-library/docker)
- **Supported architectures:** (more info (<https://github.com/docker-library/official-images#architectures-other-than-amd64>))  
[amd64 \(https://hub.docker.com/r/amd64/docker/\)](https://hub.docker.com/r/amd64/docker/), [arm32v6 \(https://hub.docker.com/r/arm32v6/docker/\)](https://hub.docker.com/r/arm32v6/docker/), [arm64v8 \(https://hub.docker.com/r/arm64v8/docker/\)](https://hub.docker.com/r/arm64v8/docker/), [ppc64le \(https://hub.docker.com/r/ppc64le/docker/\)](https://hub.docker.com/r/ppc64le/docker/), [s390x \(https://hub.docker.com/r/s390x/docker/\)](https://hub.docker.com/r/s390x/docker/)
- **Published image artifact details:**  
[repo-info repo's repos/docker/ directory \(https://github.com/docker-library/repo-info/blob/master/repos/docker\)](https://github.com/docker-library/repo-info/blob/master/repos/docker) ([history \(https://github.com/docker-library/repo-info/commits/master/repos/docker\)](https://github.com/docker-library/repo-info/commits/master/repos/docker))  
 (image metadata, transfer size, etc)
- **Image updates:**  
[official-images PRs with label library/docker \(https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fdocker\)](https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fdocker)  
[official-images repo's library/docker file \(https://github.com/docker-library/official-images/blob/master/library/docker\)](https://github.com/docker-library/official-images/blob/master/library/docker) ([history \(https://github.com/docker-library/official-images/commits/master/library/docker\)](https://github.com/docker-library/official-images/commits/master/library/docker))
- **Source of this description:**  
[docs repo's docker/ directory \(https://github.com/docker-library/docs/tree/master/docker\)](https://github.com/docker-library/docs/tree/master/docker) ([history \(https://github.com/docker-library/docs/commits/master/docker\)](https://github.com/docker-library/docs/commits/master/docker))
- **Supported Docker versions:**  
[the latest release \(https://github.com/docker/docker-ce/releases/latest\)](https://github.com/docker/docker-ce/releases/latest) (down to 1.6 on a best-effort basis)

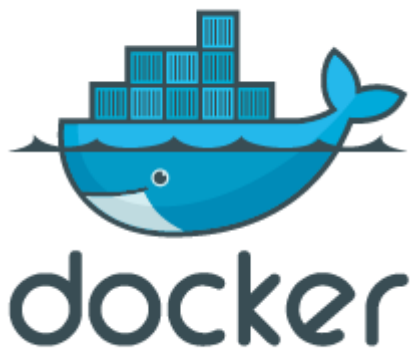
## What is Docker in Docker?

Although running Docker inside Docker is generally not recommended, there are some legitimate use cases, such as development of Docker itself.

*Docker is an open-source project that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of operating-system-level virtualization on Linux, Mac OS and Windows.*

[wikipedia.org/wiki/Docker\\_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))

[\(https://en.wikipedia.org/wiki/Docker\\_\(software\)\)](https://en.wikipedia.org/wiki/Docker_(software))



Before running Docker-in-Docker, be sure to read through [Jérôme Petazzoni's excellent blog post on the subject](https://jpetazzo.github.io/2015/09/03/do-not-use-docker-in-docker-for-ci/) (<https://jpetazzo.github.io/2015/09/03/do-not-use-docker-in-docker-for-ci/>), where he outlines some of the pros and cons of doing so (and some nasty gotchas you might run into).

If you are still convinced that you need Docker-in-Docker and not just access to a container's host Docker server, then read on.

## How to use this image

```

tianon@nameless ~ $ docker images docker | less
tianon@nameless ~ $ docker run -it --privileged --name some-docker -d docker:dind
fe95ad1a89f41f4f03cae53b27dcd674cea04a60dc43e1150468b10fcd032a2a
tianon@nameless ~ $ docker logs some-docker
INFO[0000] Listening for HTTP on unix (/var/run/docker.sock)
WARN[0000] /\ DON'T BIND ON ANY IP ADDRESS WITHOUT setting -tlsverify IF YOU DON'T KNOW WHAT YOU'RE DOING /\
INFO[0000] Listening for HTTP on tcp (0.0.0.0:2375)
WARN[0000] Running modprobe bridge nf_nat failed with message: , error: exit status 1
INFO[0000] Loading containers: start.

INFO[0000] Loading containers: done.
INFO[0000] Daemon has completed initialization
INFO[0000] Docker daemon                                commit=786b29d execdriver=native-0.2 graphdriver=vfs version=1.7.1
tianon@nameless ~ $ docker run -it --rm --link some-docker:docker docker version
Client version: 1.7.1
Client API version: 1.19
Go version (client): go1.4.2
Git commit (client): 786b29d
OS/Arch (client): linux/amd64
Server version: 1.7.1
Server API version: 1.19
Go version (server): go1.4.2
Git commit (server): 786b29d
OS/Arch (server): linux/amd64
tianon@nameless ~ $ docker run -it --rm --link some-docker:docker docker sh
/ #

```



<https://asciinema.org/a/24707>

## Start a daemon instance

**IMPORTANT:** this image defaults to `--storage-driver=vfs`, which will be very slow and inefficient (but is the only driver which is guaranteed to work regardless of your underlying filesystem). Which driver you should use varies depending on your needs, but a good rule of thumb is that your DinD instance should be using the same driver as your host (which can be seen under `Storage Driver` in the output of `docker info`). See the "Custom daemon flags" section below for how to specify your storage driver.

```
$ docker run --privileged --name some-docker -d docker:stable-dind
```

**Note:** `--privileged` is required for Docker-in-Docker to function properly, but it should be used with care as it provides full access to the host environment, as explained [in the relevant section of the Docker documentation](https://docs.docker.com/engine/reference/run/#runtime-privilege-and-linux-capabilities) (<https://docs.docker.com/engine/reference/run/#runtime-privilege-and-linux-capabilities>).

This image includes `EXPOSE 2375` (the Docker port), so standard container linking will make it automatically available to the linked containers (as the following examples illustrate).

## Connect to it from a second container

```
$ docker run --rm --link some-docker:docker docker:edge version
Client:
Version:      17.05.0-ce
API version:  1.27 (downgraded from 1.29)
Go version:   go1.7.5
Git commit:   89658be
Built:        Fri May  5 15:36:11 2017
OS/Arch:      linux/amd64

Server:
Version:      17.03.1-ce
API version:  1.27 (minimum version 1.12)
Go version:   go1.7.5
Git commit:   c6d412e
Built:        Tue Mar 28 00:40:02 2017
OS/Arch:      linux/amd64
Experimental: false
```

```
$ docker run -it --rm --link some-docker:docker docker:edge sh
/ # docker version
```

```
Client:
Version:      17.05.0-ce
API version:  1.27 (downgraded from 1.29)
Go version:   go1.7.5
Git commit:   89658be
Built:        Fri May  5 15:36:11 2017
OS/Arch:      linux/amd64

Server:
Version:      17.03.1-ce
API version:  1.27 (minimum version 1.12)
Go version:   go1.7.5
Git commit:   c6d412e
Built:        Tue Mar 28 00:40:02 2017
OS/Arch:      linux/amd64
Experimental: false
```

```
$ docker run --rm --link some-docker:docker docker info
Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
Images: 0
Server Version: 17.03.1-ce
Storage Driver: vfs
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
  Volume: local
  Network: bridge host macvlan null overlay
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Init Binary: docker-init
containerd version: 4ab9917febca54791c5f071a9d1f404867857fcc
runc version: 54296cf40ad8143b62dbcaa1d90e520a2136ddfe
init version: 949e6fa
Security Options:
  seccomp
   Profile: default
Kernel Version: 4.4.63-gentoo
Operating System: Alpine Linux v3.5 (containerized)
OSType: linux
Architecture: x86_64
CPUs: 8
Total Memory: 31.4 GiB
Name: 393376fdc461
ID: FDP3:4GDT:L2WP:D4CC:UAW5:RHNA:4Z4G:WQYY:YWBE:7RER:LV7E:USY5
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): false
Registry: https://index.docker.io/v1/
WARNING: bridge-nf-call-iptables is disabled
WARNING: bridge-nf-call-ip6tables is disabled
Experimental: false
Insecure Registries:
  127.0.0.0/8
Live Restore Enabled: false
```

```
$ docker run --rm -v /var/run/docker.sock:/var/run/docker.sock dock
Client:
Version:      17.05.0-ce
API version:  1.28 (downgraded from 1.29)
Go version:   go1.7.5
Git commit:   89658be
Built:        Fri May  5 15:36:11 2017
OS/Arch:      linux/amd64

Server:
Version:      17.04.0-ce
API version:  1.28 (minimum version 1.12)
Go version:   go1.8
Git commit:   4845c56
Built:        Thu Apr 27 07:51:43 2017
OS/Arch:      linux/amd64
Experimental: false
```

## Custom daemon flags

```
$ docker run --privileged --name some-overlay-docker -d docker:dind
```

## Where to Store Data

Important note: There are several ways to store data used by applications that run in Docker containers. We encourage users of the `docker` images to familiarize themselves with the options available, including:

- Let Docker manage the storage of your data by writing to disk on the host system using its own internal volume management (<https://docs.docker.com/engine/tutorials/dockervolumes/#adding-a-data-volume>). This is the default and is easy and fairly transparent to the user. The downside is that the files may be hard to locate for tools and applications that run directly on the host system, i.e. outside containers.
- Create a data directory on the host system (outside the container) and mount this to a directory visible from inside the container (<https://docs.docker.com/engine/tutorials/dockervolumes/#mount-a-host-directory-as-a-data-volume>). This places the files in a known location on the host system, and makes it easy for tools and applications on the host system to access the files. The downside is that the user needs to make sure that the directory exists, and that e.g. directory permissions and other security mechanisms on the host system are set up correctly.

The Docker documentation is a good starting point for understanding the different storage options and variations, and there are multiple blogs and forum postings that discuss and give advice in this area. We will simply show the basic procedure here for the latter option

above:

1. Create a data directory on a suitable volume on your host system, e.g. `/my/own/var-lib-docker`.
2. Start your `docker` container like this:

```
$ docker run --privileged --name some-docker -v /my/own/var-lib-
```

The `-v /my/own/var-lib-docker:/var/lib/docker` part of the command mounts the `/my/own/var-lib-docker` directory from the underlying host system as `/var/lib/docker` inside the container, where Docker by default will write its data files.

## License

View [license information](#)

(<https://github.com/docker/docker/blob/eb7b2ed6bbe3fbef588116d362ce595d6e35fc43/LICENSE>) for the software contained in this image.

As with all Docker images, these likely also contain other software which may be under other licenses (such as Bash, etc from the base distribution, along with any direct or indirect dependencies of the primary software being contained).

Some additional license information which was able to be auto-detected might be found in the [repo-info repository's docker/ directory](#) (<https://github.com/docker-library/repo-info/tree/master/repos/docker>).

As for any pre-built image usage, it is the image user's responsibility to ensure that any use of this image complies with any relevant licenses for all software contained within.

Docker Pull Command



```
docker pull docker
```