

Docker

Docker

Ubuntu

docker Ubuntu

Mac, windows

- [Ubuntu Docker](#)
- [CentOS Docker](#)
- [Windows Docker](#)
- [MacOS Docker](#)
- [docker](#)

Ubuntu Docker

Docker Engine-Community Ubuntu

- Xenial 16.04 (LTS)
- Bionic 18.04 (LTS)
- Cosmic 18.10
- Disco 19.04
-

Docker Engine - Community x86_64 amd64 armhf arm64 s390x IBM Z ppc64le IBM Powe

Docker docker docker.io docker-engine

```
apt remove docker docker-engine docker.io containerd runc
```

Docker Engine-Community docker-ce

Docker Engine-Community

Docker

Docker Engine-Community

Docker

Docker

apt

```
curl -fsSL http://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] http://mirrors.aliyun.com/docker-ce/linux/ubuntu
$(lsb_release -cs) stable"
```

```
curl -fsSL http://mirrors.huaweicloud.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] http://mirrors.huaweicloud.com/docker-
ce/linux/ubuntu $(lsb_release -cs) stable"
```

```
apt update
```

Docker GPG

```
curl -fsSL http://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
||
```

```
curl -fsSL http://mirrors.huaweicloud.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -
```

```
||
```

```
curl -fsSL http://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -
```

Docker Engine-Community

apt

```
sudo apt update
```

Docker Engine-Community containerd

```
sudo apt install docker-ce docker-ce-cli containerd.io -y
```

Docker :

```
docker run hello-world
```

```
Unable to find image 'hello-world:latest' locally
```

```
latest: Pulling from library/hello-world
```

```
1b930d010525: Pull
```

```
complete
```

```
Digest: sha256:c3b4ada4687bbaa170745b3e4dd8ac3f194ca95b2d0518b417fb47e5879d9b5f
```

```
Status: Downloaded newer image for hello-world:latest
```

```
Hello from Docker!
```

```
This message shows that your installation appears to be working correctly.
```

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it

to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

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

For more examples and ideas, visit:

<https://docs.docker.com/get-started/>

Shell

Docker get.docker.com test.docker.com

Docker Engine-Community

- `root` `sudo`
- `Linux` `Docker` `edge`
- `Docker`

get.docker.com

`Linux`

Docker Engine-Community

test.docker.com

```
curl -fsSL https://get.docker.com -o get-docker.sh
sh get-docker.sh
```

Docker `root`

`docker`

```
usermod -aG docker your-user
```

```
usermod -aG docker develop
```

```
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<- 'EOF'
{
  "registry-mirrors": ["https://xviwkyb8.mirror.aliyuncs.com"]
}
```

EOF

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart docker
```

CentOS Docker

Docker 64 CentOS

- CentOS 7
- CentOS 8
- ... centos-extras

overlay2

Docker docker docker-engine

```
yum remove docker \
docker-client \
docker-client-latest \
docker-common \
docker-latest \
docker-latest-logrotate \
docker-logrotate \
docker-engine
```

Docker Engine-Community

Docker

Docker Engine-Community

Docker

Docker

yum-utils

yum-config-manager

device mapper

device-mapper-persistent-data

```
yum install -y yum-utils \
device-mapper-persistent-data \
lvm2
```

```
yum-config-manager \
--add-repo \
https://download.docker.com/linux/centos/docker-ce.repo
http://mirrors.huaweicloud.com/docker-ce/linux/centos/docker-ce.repo

yum makecache fast
```

Docker Engine-Community

Docker Engine-Community containerd

```
yum install docker-ce docker-ce-cli containerd.io
```

GPG

“ Docker

Docker yum install yum update

Docker docker

Docker Engine-Community

1

```
yum list docker-ce --showduplicates | sort -r
```

docker-ce.x86_64	3:18.09.1-3.el7	docker-ce-stable
docker-ce.x86_64	3:18.09.0-3.el7	docker-ce-stable
docker-ce.x86_64	18.06.1.ce-3.el7	docker-ce-stable
docker-ce.x86_64	18.06.0.ce-3.el7	docker-ce-stable

2 docker-ce : - docker-ce:

```
yum install docker-ce-<VERSION_STRING> docker-ce-cli-<VERSION_STRING> containerd.io
```

Docker

```
systemctl start docker
systemctl enable docker
```

hello-world Docker Engine-Community

```
docker run hello-world
```

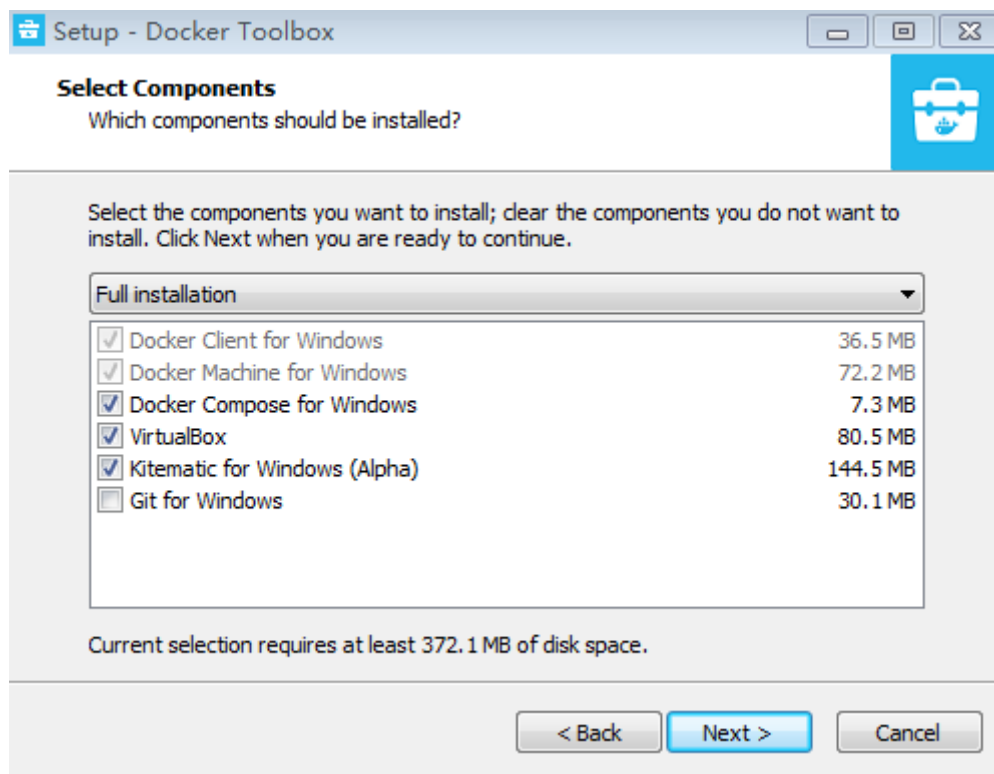
Docker root docker

```
usermod -aG docker your-user
```


Windows Docker

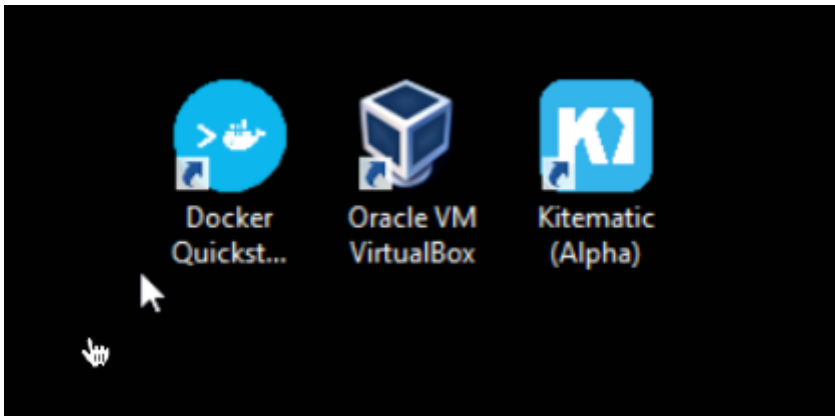
win7 win8

win7 win8 [docker toolbox](#) [windows/docker-toolbox](#)



docker toolbox

- Docker CLI - docker
- Docker Machine - Windows docker
- Docker Compose - docker-compose
- Kitematic - Docker GUI
- Docker QuickStart shell - Docker
- Oracle VM Virtualbox -



Docker QuickStart

Docker Toolbox

User Account Control

VirtualBox

Yes

```
MINGW32/c/Users/M
export DOCKER_HOST=tcp://192.168.59.103:2376
export DOCKER_CERT_PATH='C:\Users\M\boot2docker\certs\boot2docker-vm'
export DOCKER_TLS_VERIFY=1

IP address of docker VM:
192.168.59.103

setting environment variables ...
Writing C:\Users\M\boot2docker\certs\boot2docker-vm\ca.pem
Writing C:\Users\M\boot2docker\certs\boot2docker-vm\cert.pem
Writing C:\Users\M\boot2docker\certs\boot2docker-vm\key.pem
export DOCKER_HOST=tcp://192.168.59.103:2376
export DOCKER_CERT_PATH='C:\Users\M\boot2docker\certs\boot2docker-vm'
export DOCKER_TLS_VERIFY=1

You can now use 'docker' directly, or 'boot2docker ssh' to log into the VM.
Welcome to Git (version 1.9.5-preview20150319)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

MERNEALE ~
$
```

\$

```
$ docker run hello-world

Unable to find image 'hello-world:latest' locally
Pulling repository hello-world
91c95931e552: Download complete
a8219747be10: Download complete
Status: Downloaded newer image for hello-world:latest
Hello from Docker.

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker Engine CLI client contacted the Docker Engine daemon.
2. The Docker Engine daemon pulled the "hello-world" image from the Docker Hub.
   (Assuming it was not already locally available.)
```

3. The Docker Engine daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker Engine daemon streamed that output to the Docker Engine CLI client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

For more examples and ideas, visit:

<https://docs.docker.com/userguide/>

Win10

Docker

Win10

Hyper-V

Hyper-V



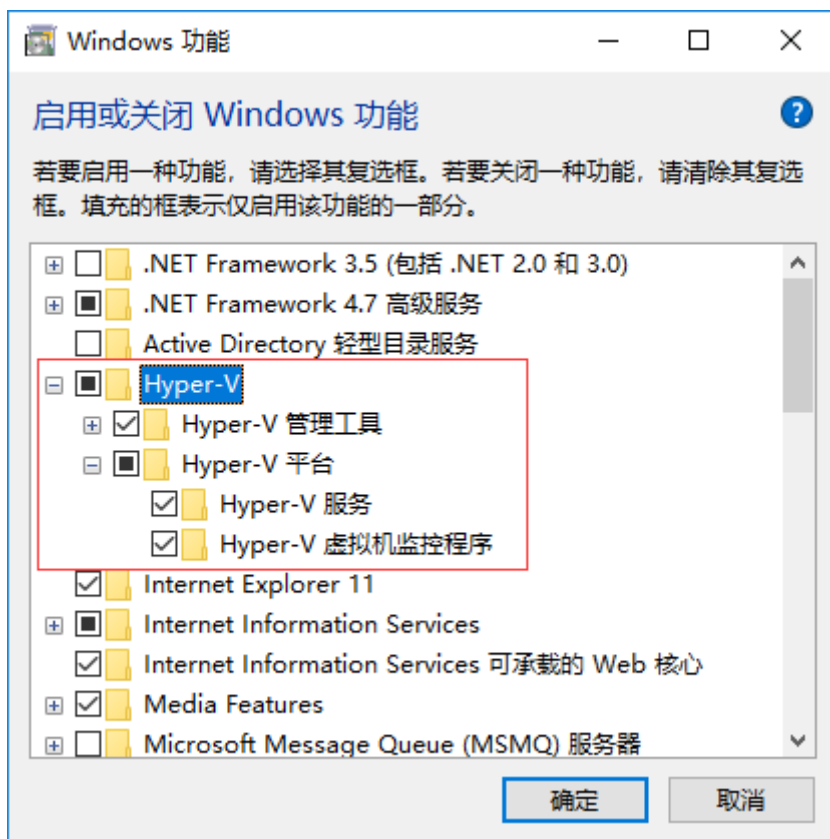
相关设置

程序和功能

- Windows

 [启用或关闭 Windows 功能](#)

- Hyper-V

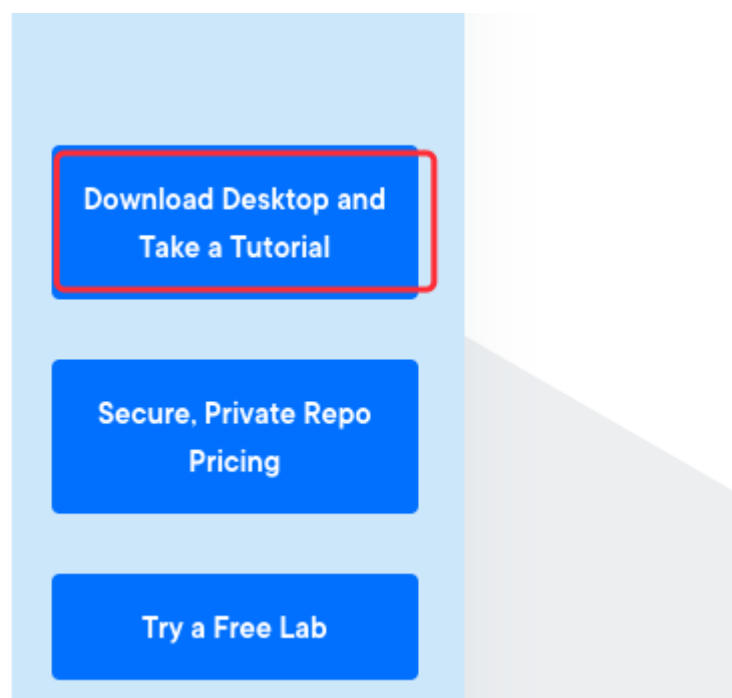


1 Toolbox

Toolboxget-docker

[Download Desktop and Take a Tutorial](#)

Windows



Welcome to Docker Hub

Download and Take a Tutorial

Get started by downloading Docker Desktop, and learn how you can build, tag and share a sample image on Hub.

[Get started with Docker Desktop](#)

Welcome, tianqixin

Let's get you started: you will need to download and install Docker Desktop to use the Docker command line interface (CLI).

Docker Desktop
The preferred choice for millions of developers that are building containerized applications

[Download Docker Desktop for Mac](#)

[Docker Desktop for Windows](#)

Looking for Docker Engine Community?

[Skip Tutorial](#)

这里根据不同平台点击下即可

2

- Docker for Windows Installer Next Finish




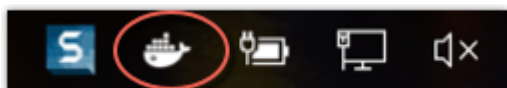
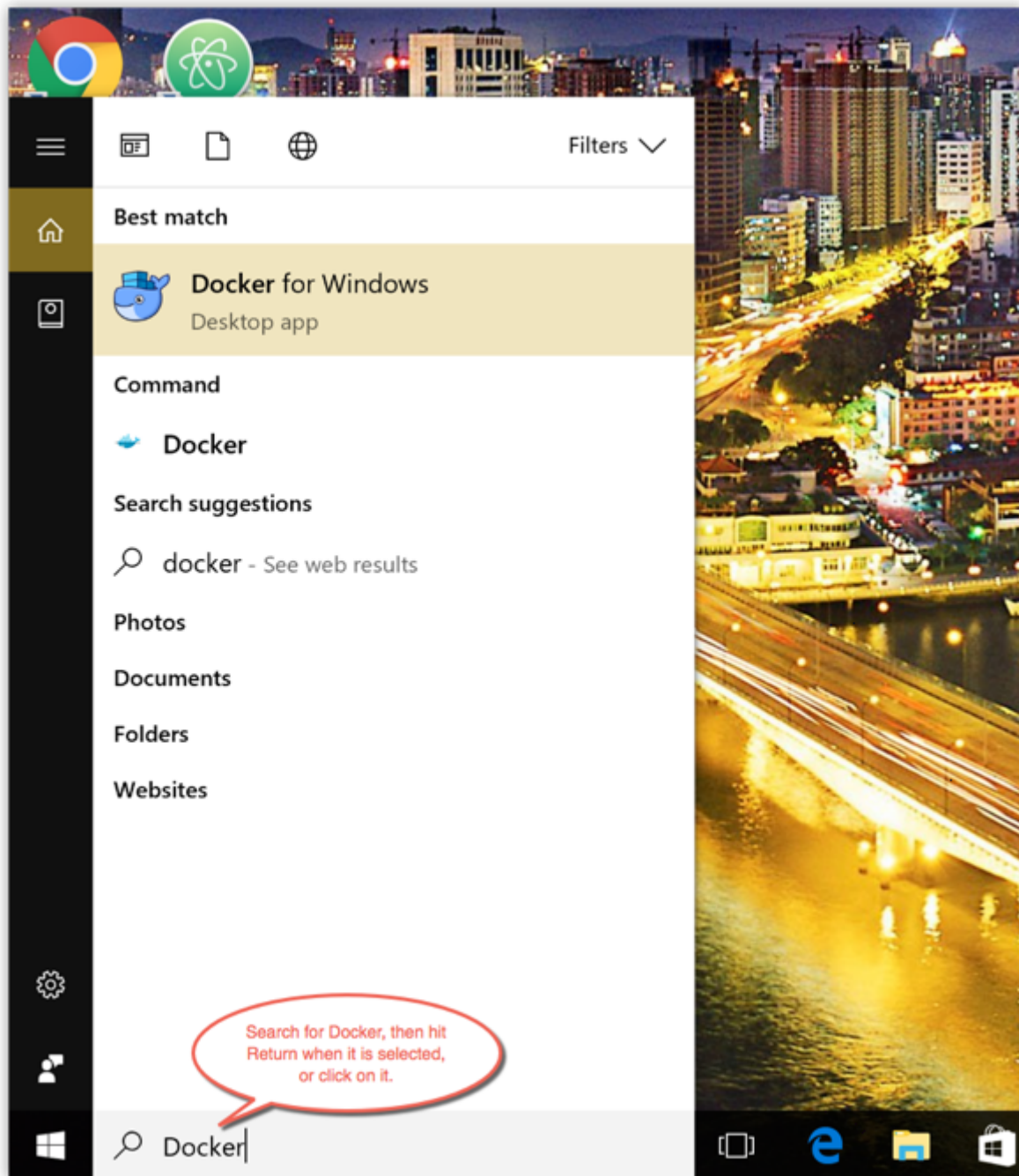
Docker for Windows 17.09.0-ce-win33

Installation succeeded

You must log out of Windows to complete installation.

Close and log out

-  Docker
- docker version docker run hello-world
- Windows Docker



Windows 10

Windows 10

Docker

Settings registry.docker.io Docker Daemon App Registry Docker

General

Shared Drives

Advanced


Network

Proxies

Daemon

Diagnose & Feedback

Reset

 Docker is running

Daemon



Configure the Docker daemon by typing a json docker daemon [configuration file](#).

☒ Advanced

This can prevent Docker from starting. Use at your own risk!

```
{  
  "registry-mirrors": [  
    "https://fd. .mirror.aliyuncs.com"  
  ],  
  "insecure-registries": []  
}
```

Docker will restart when applying these settings.

Apply

MacOS Docker

Homebrew

macOS Homebrew Docker

Homebrew Cask Docker for Mac Homebrew Cask

```
$ brew cask install docker

==> Creating Caskroom at /usr/local/Caskroom
==> We'll set permissions properly so we won't need sudo in the future
Password:      #   macOS
==> Satisfying dependencies
==> Downloading https://download.docker.com/mac/stable/21090/Docker.dmg
##### 100.0%
==> Verifying checksum for Cask docker
==> Installing Cask docker
==> Moving App 'Docker.app' to '/Applications/Docker.app'.
🍺  docker was successfully installed!
```

Docker app

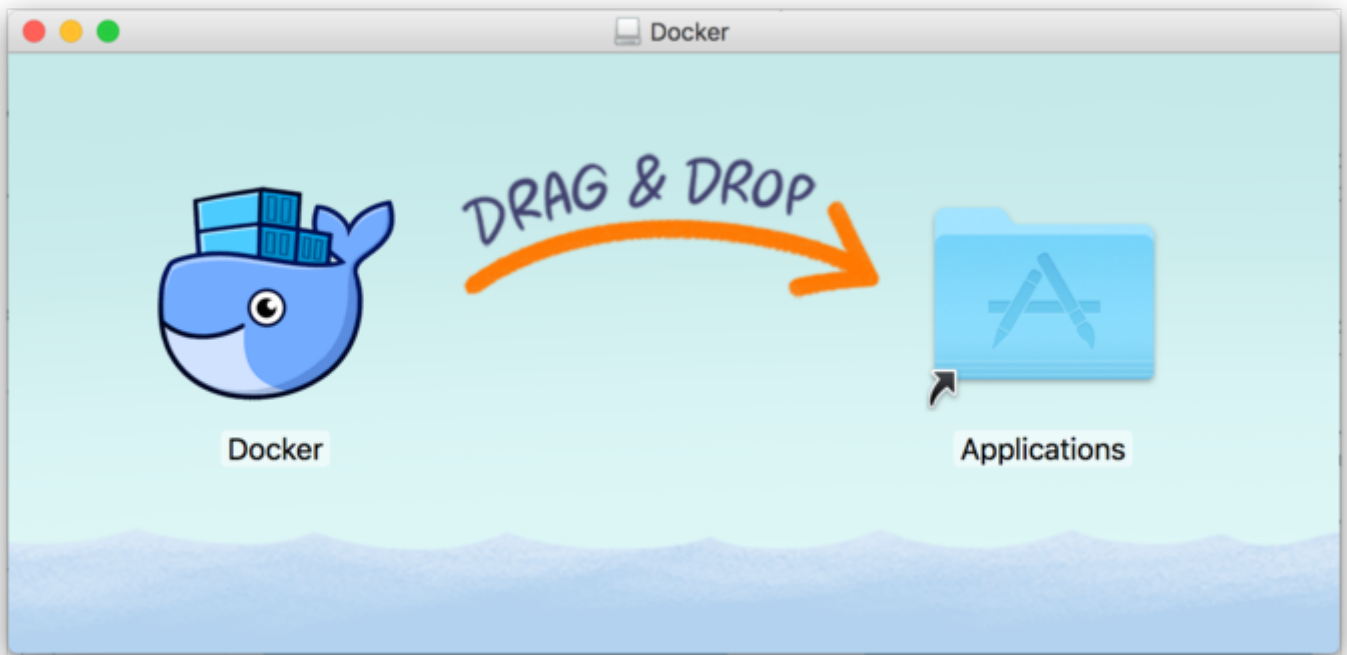
Next

 macOS

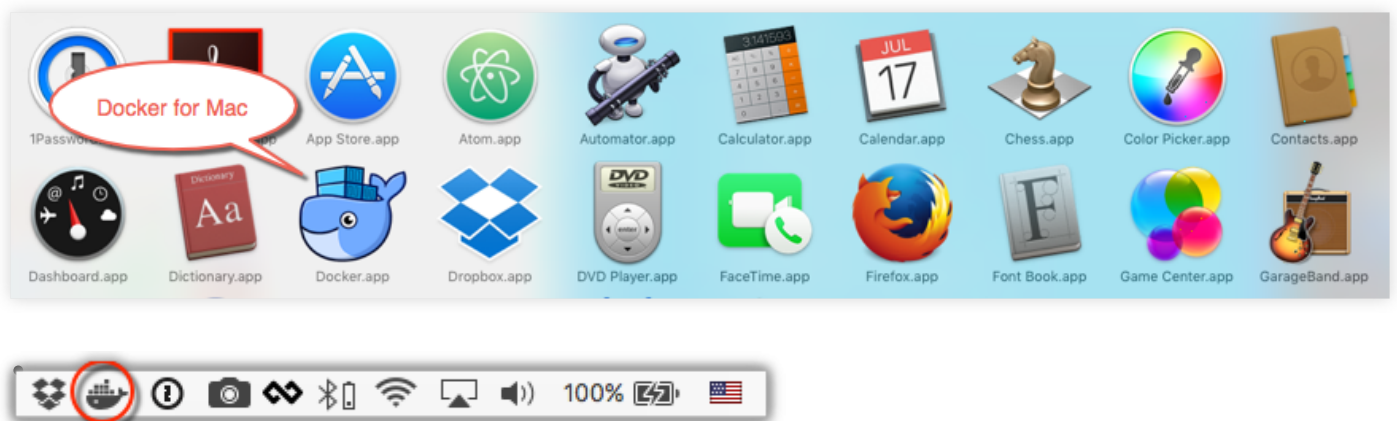
Docker

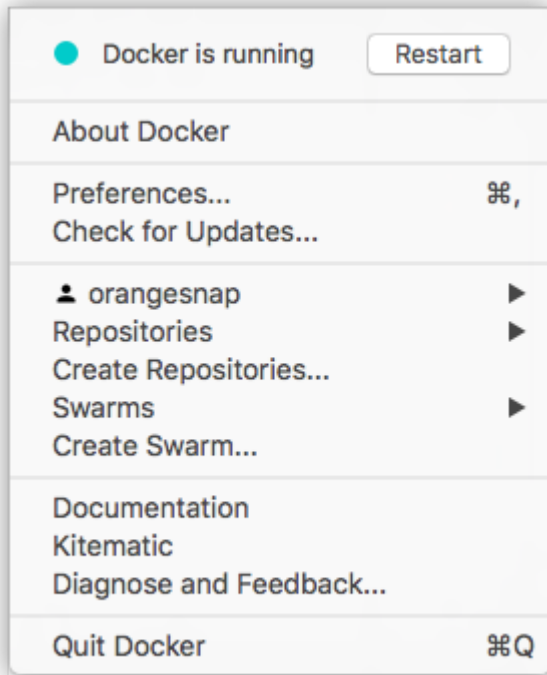
[Stable](#) [Edge](#) Docker for Mac

- macOS .dmg Application

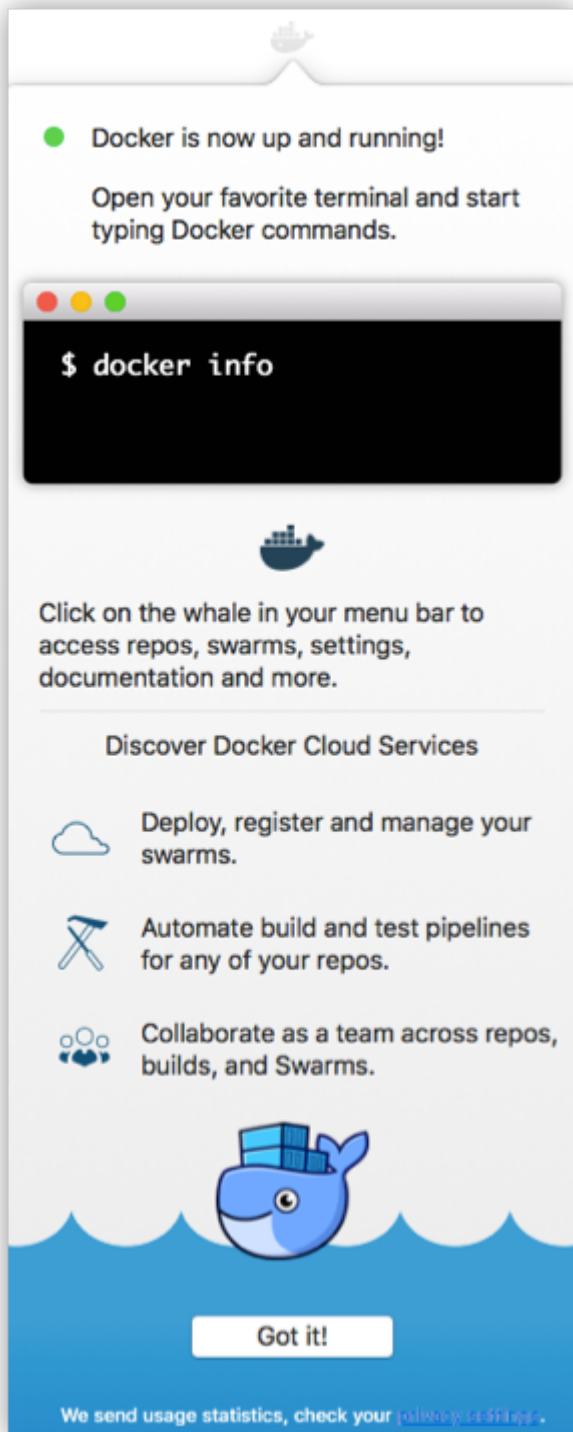


- Docker macOS





- "Got it!"



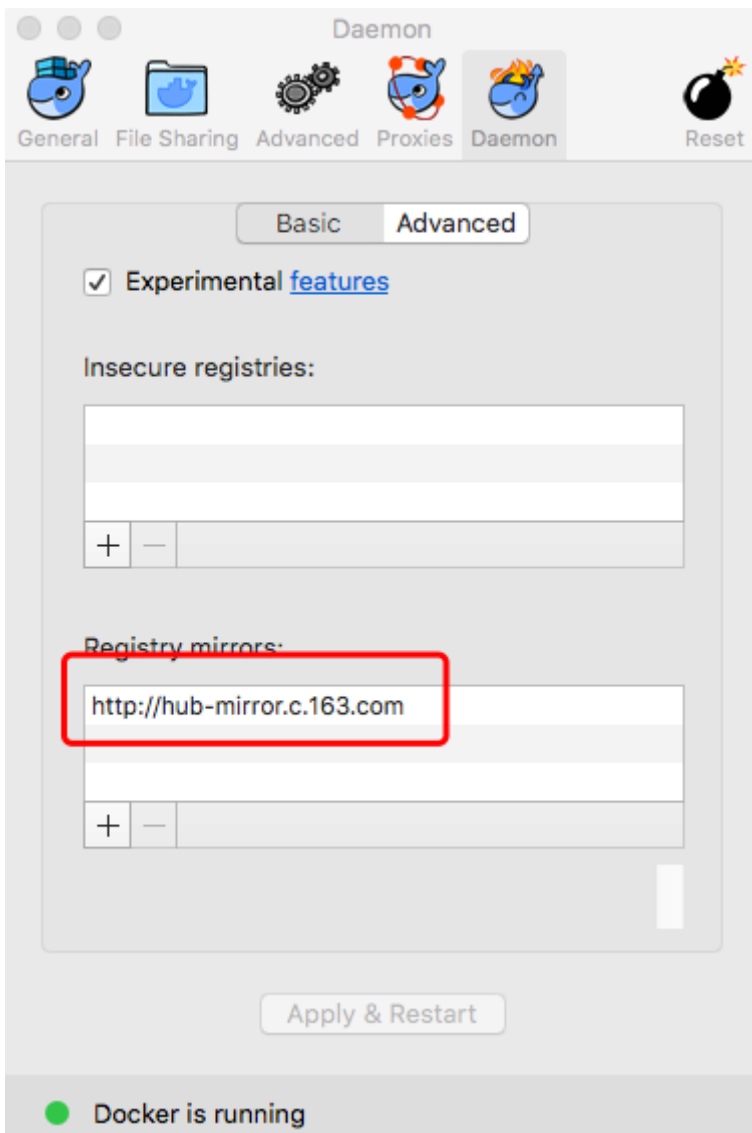
- Docker

```
$ docker --version
Docker version 17.09.1-ce, build 19e2cf6
```

hub-Docker.163

Docker for macOS Preferences... -> Daemon -> Registry mirrors

Apply



- docker info

```
$ docker info
...
Registry Mirrors:
  http://hub-mirror.c.163.com
Live Restore Enabled: false
```


docker

docker

Docker Go Apache2.0

Docker Linux

iPhone app ,

 Docker 17.03 CE Community Edition: EE Enterprise Edition:

Docker

- Web
- OpenShift Cloud Foundry PaaS

Docker

- 1 Docker Linux Docker Docker Docker
- 2 Docker Docker Docker Docker
- 3 Docker Docker Docker Docker

Docker

Docker - (C/S) API Docker

Docker Docker

Docker	

Docker	
Docker (Images)	Docker Docker
Docker (Container)	

Docker	
Docker (Client)	Docker Docker API (https://docs.docker.com/reference/api/docker_remote_api) Docker
Docker (Host)	Docker
Docker (Registry)	Docker Docker Hub(https://hub.docker.com)
Docker Machine	Docker Machine Docker Ocean Microsoft Azure Doc

Docker

“ Ubuntu Docker

“ CentOS Docker

“ Windows Docker

“ MacOS Docker

Docker

Docker

Docker

“ docker <http://www.docker.com>

Docker Windows <https://docs.docker.com/docker-for-windows/>

Docker CE() Ubuntu <https://docs.docker.com/install/linux/docker-ce/ubuntu/>

Docker mac <https://docs.docker.com/docker-for-mac/>

Docker <https://docs.docker.com/config/daemon/>

Docker <http://blog.docker.com/>

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

Docker <https://www.docker.com/open-source>

Docker

“ https://help.aliyun.com/document_detail/60750.html

<http://hub-mirror.c.163.com>

<https://registry.docker-cn.com>

ustc <https://docker.mirrors.ustc.edu.cn>

daocloud <https://www.daocloud.io/mirror#accelerator-doc>

<https://cr.console.aliyun.com/>

```
docker login --username=itqmdx@qq.com registry.cn-hangzhou.aliyuncs.com
```

1()

```
registry.cn-hangzhou.aliyuncs.com/wzhz/[image:[version]]
```

```
registry-vpc.cn-hangzhou.aliyuncs.com/wzhz/[image:[version]]
```

```
registry-internal.cn-hangzhou.aliyuncs.com/wzhz/[image:[version]]
```

1()

```
registry.cn-qingdao.aliyuncs.com/wzhz/[image:[version]]
```

```
registry-vpc.cn-qingdao.aliyuncs.com/wzhz/[image:[version]]
```

```
registry-internal.cn-qingdao.aliyuncs.com/wzhz/[image:[version]]
```

1()

```
registry.cn-shenzhen.aliyuncs.com/wzhz/[image:[version]]
```

```
registry-vpc.cn-shenzhen.aliyuncs.com/wzhz/[image:[version]]
```

```
registry-internal.cn-shenzhen.aliyuncs.com/wzhz/[image:[version]]
```

()

```
registry.cn-hongkong.aliyuncs.com/wzhz/[image:[version]]
```

```
registry-vpc.cn-hongkong.aliyuncs.com/wzhz/[image:[version]]
```

```
registry-internal.cn-hongkong.aliyuncs.com/wzhz/[image:[version]]
```

Docker

```
https://wzhz.mirror.aliyuncs.com
```

1. Docker

```
1.10.0 Docker docker-ce
```

2.

```
Docker 1.10.0
```

```
dd /etc/docker/daemon.json |
```

```
vim docker-aliyun-daemon
```

```
#!/bin/bash

sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<- 'EOF'
{
  "registry-mirrors": ["https://wzhz.mirror.aliyuncs.com"]
}
EOF

sudo systemctl daemon-reload
sudo systemctl restart docker
```

```
sudo chmod 755 docker-aliyun-daemon
```

```
./docker-aliyun-daemon
```

volume

```
# volume
$ docker volume ls -qf dangling=true

# volume
$ docker volume rm $(docker volume ls -qf dangling=true)

#
$ docker system prune
```