



Docker的安装之Windows

系统要求

[Docker Desktop for Windows](#) 支持 64 位版本的 Windows 10 Pro，且必须开启 Hyper-V（若版本为 v1903 及以上则无需开启 Hyper-V），或者 64 位版本的 Windows 10 Home v1903 及以上版本。

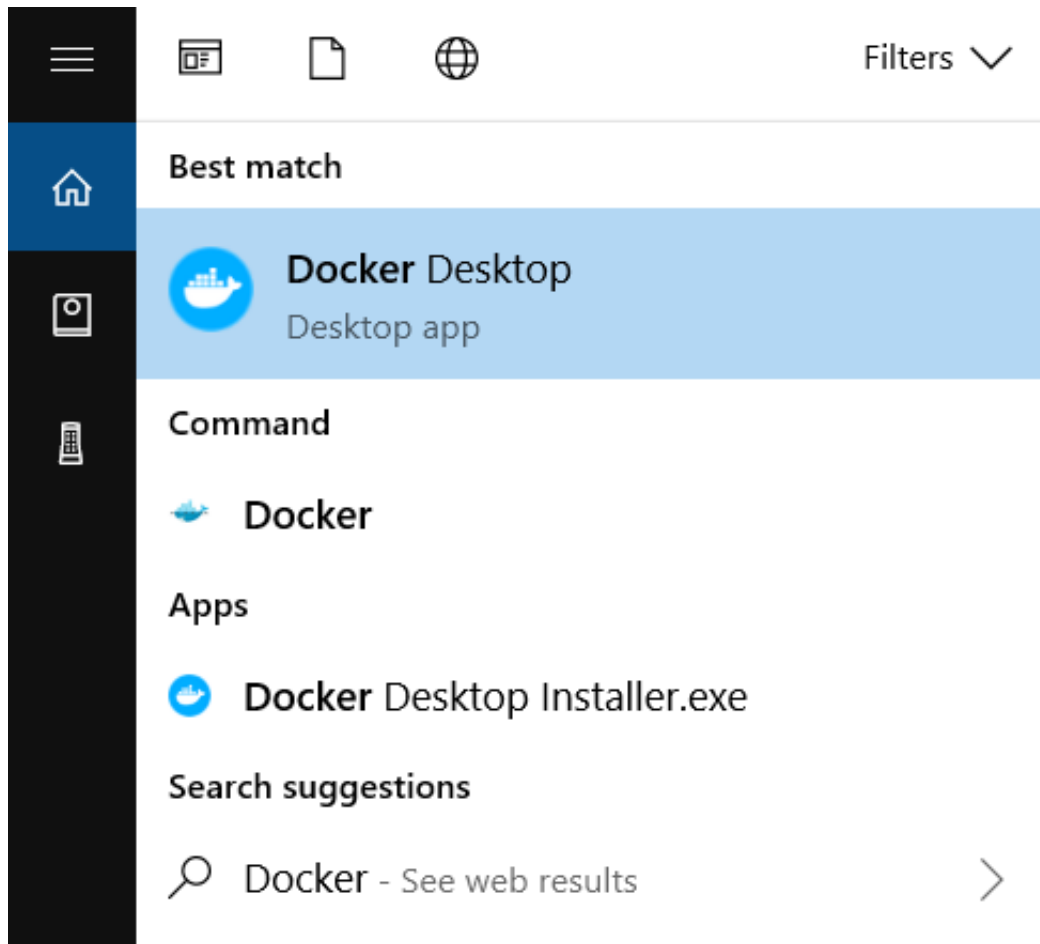
安装

1. 从这个链接[下载Windows版本的Docker](#)。
2. 它通常会下载到您的Downloads文件夹中，或者您可以从Web浏览器底部的“最近下载”栏中运行它。
3. 出现提示时，请确保在“配置”页面上选择了“**启用Hyper-V Windows功能或为WSL 2安装所需的Windows组件**”选项。
4. 按照安装向导上的说明授权安装程序并继续进行安装。
5. 安装成功后，单击“**关闭**”以完成安装过程。
6. 如果您的管理员帐户与用户帐户不同，则必须将该用户添加到**docker-users**组。以管理员身份运行**计算机管理**，然后导航到**本地用户和组>组>docker-users**。右键单击以将该用户添加到该组。注销并重新登录，以使更改生效。

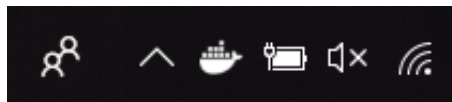
启动Docker

1. Docker Desktop在安装后不会自动启动。要启动Docker Desktop，请搜索

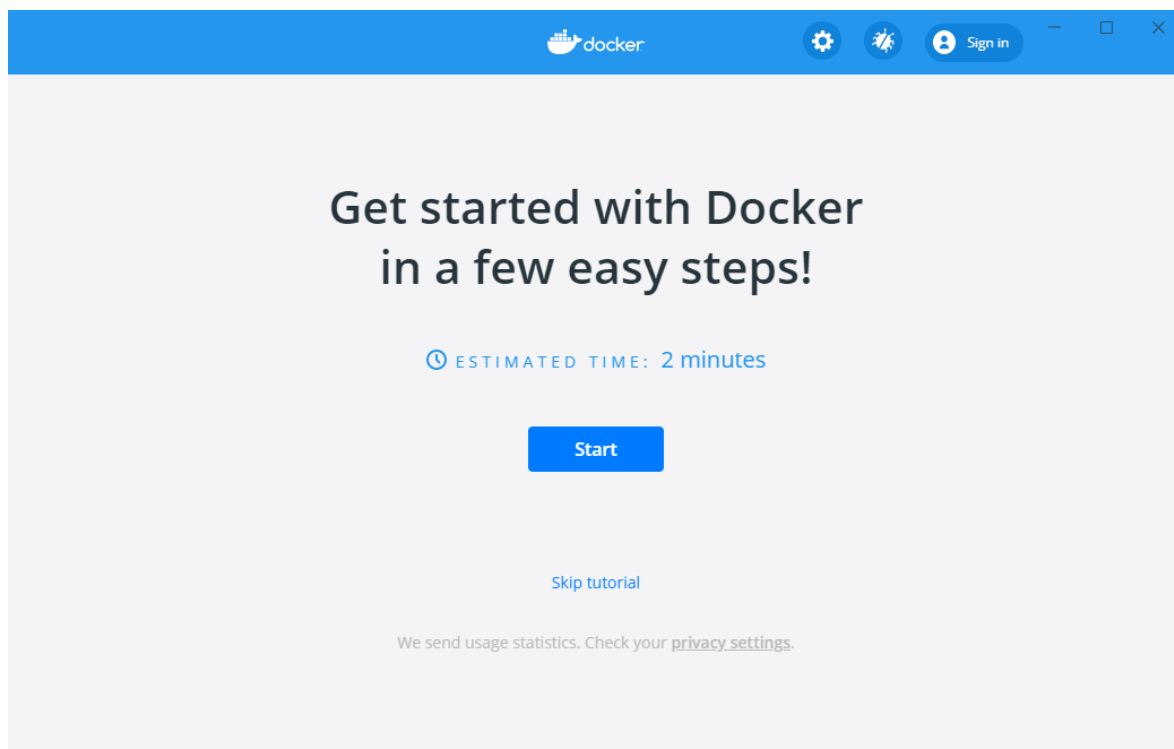
Docker，然后在搜索结果中选择**Docker Desktop**。



2. 当状态栏中的鲸鱼图标保持稳定时，Docker桌面将运行，并且可以从任何终端窗口访问。



3. 初始化完成后，Docker Desktop将启动入门教程。本教程包括一个简单的练习，以构建示例Docker映像，将其作为容器运行，将映像推送并保存到Docker Hub。



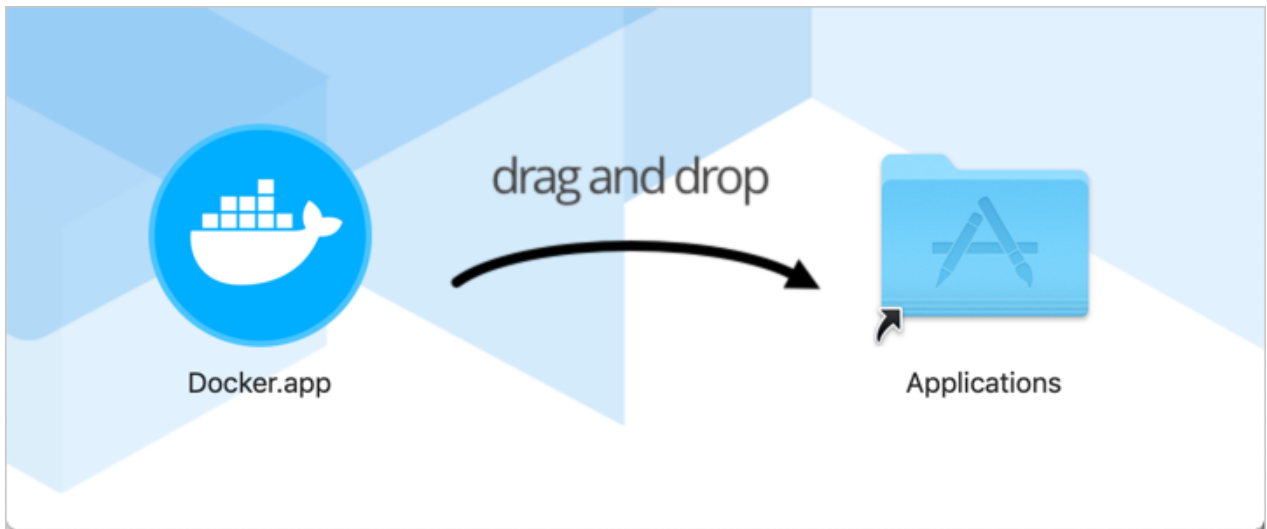
Docker的安装之MAC

系统要求

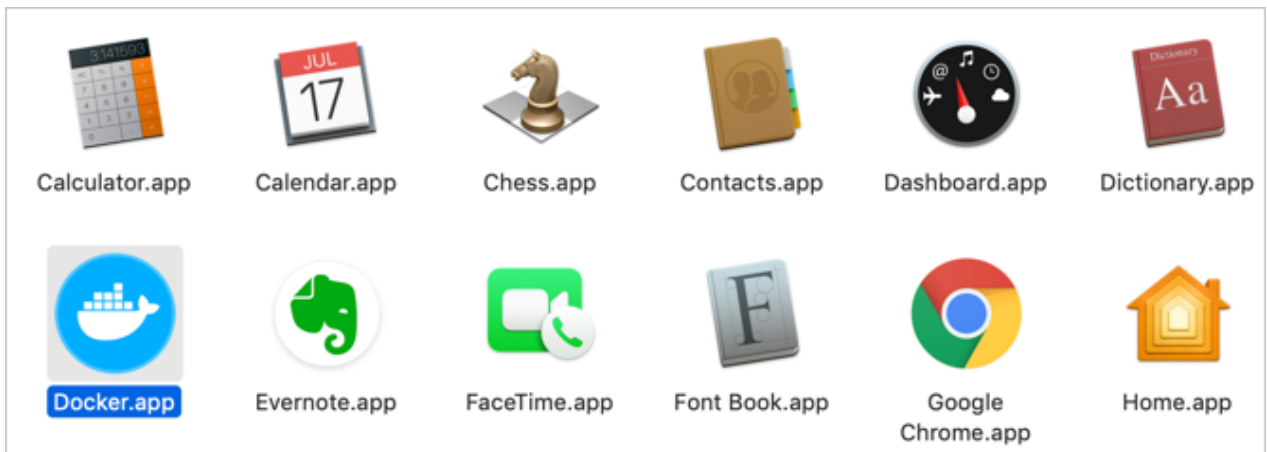
1. **macOS必须为10.14或更高版本。**也就是：Mojave, Catalina, or Big Sur。
我们建议升级到最新版本的macOS。
如果将macOS升级到10.15版后遇到任何问题，则必须安装最新版本的Docker Desktop才能与此版本的macOS兼容。
2. 至少4 GB的RAM。
3. 不得安装版本4.3.30之前的VirtualBox，因为它与Docker Desktop不兼容。

安装

1. 从[这个网页](#)安装MAC版的Docker。
2. 双击Docker.dmg打开安装程序，然后将Docker图标拖到Applications文件夹。



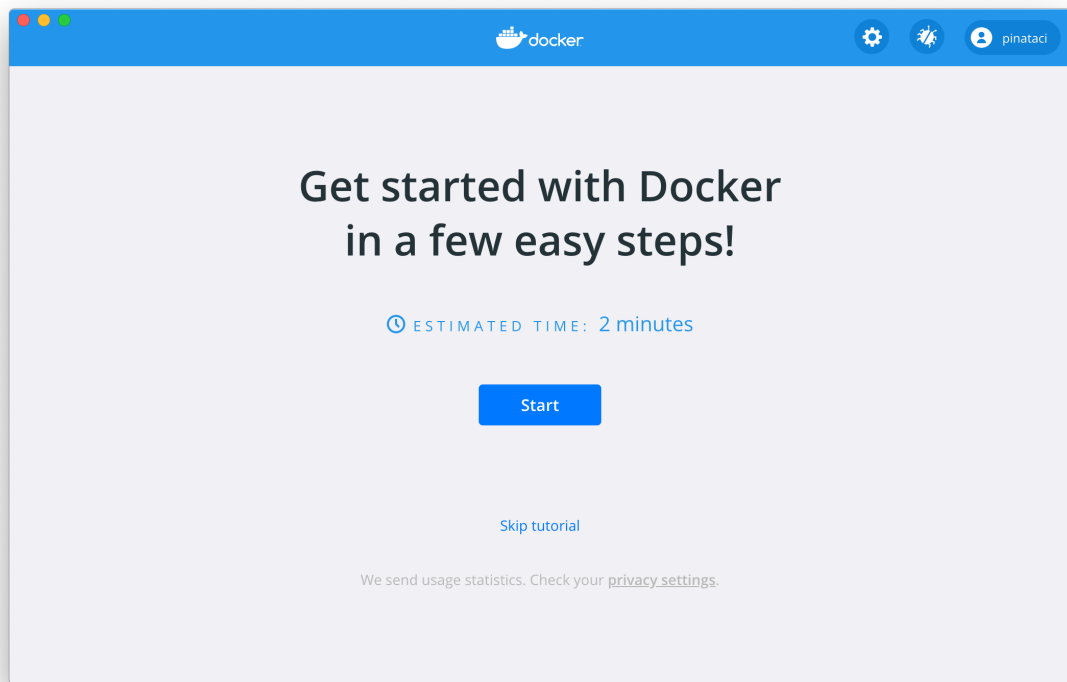
3. 双击Docker.appApplications文件夹以启动Docker。（在下面的示例中，Applications文件夹处于“网格”视图模式。）



4. 顶部状态栏中的Docker菜单指示Docker Desktop正在运行，并且可以从终端进行访问



5. 如果刚刚安装了该应用程序，则Docker Desktop将启动入门教程。本教程包括一个简单的练习，以构建示例Docker映像，将其作为容器运行，将映像推送并保存到 Docker Hub。



Docker的安装之Ubuntu

系统要求

Docker 支持以下版本的 Ubuntu 操作系统：

- Ubuntu Groovy 20.10
- Ubuntu Focal 20.04 (LTS)
- Ubuntu Bionic 18.04 (LTS)
- Ubuntu Xenial 16.04 (LTS)

Docker 可以安装在 64 位的 x86 平台或 ARM 平台上。

安装

卸载旧版本

旧版本的 Docker 称为 docker 或者 docker-engine，使用以下命令卸载旧版本：

- 如果没有旧版本可以PASS这一part

```
$ sudo apt-get remove docker \
    docker-engine \
    docker.io
```

使用APT安装

1. 由于 apt 源使用 HTTPS 以确保软件下载过程中不被篡改。因此，我们首先需要添加使用 HTTPS 传输的软件包以及 CA 证书。

警告：切勿在没有配置 Docker APT 源的情况下直接使用 apt 命令安装 Docker。

```
$ sudo apt-get update

$ sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
```

2. 为了确认所下载软件包的合法性，需要添加软件源的 GPG 密钥。

```
$ curl -fsSL https://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | \
sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

# 官方源
# $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg
--dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

鉴于国内网络问题，强烈建议使用国内源，官方源请在注释中查看。

3. 然后，我们需要向 `sources.list` 中添加 Docker 软件源：

```
$ echo \
    "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-
keyring.gpg] https://mirrors.aliyun.com/docker-ce/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null

# 官方源
# $ echo \
#    "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-
keyring.gpg] https://download.docker.com/linux/ubuntu \
#    $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
```

以上命令会添加稳定版本的 Docker APT 镜像源，如果需要测试版本的 Docker 请将 `stable` 改为 `test`。

4. 安装Docker

更新 apt 软件包缓存，并安装 docker-ce:

```
$ sudo apt-get update  
  
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

启动Docker

```
$ sudo systemctl enable docker  
$ sudo systemctl start docker
```

建立Docker的用户组

默认情况下，docker 命令会使用 [Unix socket](#) 与 Docker 引擎通讯。而只有 root 用户和 docker 组的用户才可以访问 Docker 引擎的 Unix socket。出于安全考虑，一般 Linux 系统上不会直接使用 root 用户。因此，更好地做法是将需要使用 docker 的用户加入 docker 用户组。

建立 docker 组:

```
$ sudo groupadd docker
```

将当前用户加入 docker 组:

```
$ sudo usermod -aG docker $USER
```

退出当前终端并重新登录。

测试安装成功与否

```
$ docker run --rm hello-world  
  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
b8dfde127a29: Pull complete  
Digest:  
sha256:308866a43596e83578c7dfa15e27a73011bdd402185a84c5cd7f32a88b501a2  
4  
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/>

若能正常输出以上信息，则说明安装成功。