

# 再見了VMware，一款更輕量級的虛擬機！

計算機工作原理 2022-01-17 11:00

以下文章來源於陌溪教你學編程，作者陌溪



**陌溪教你學編程**

三本院校出身，字節跳動後台開發工程師，開源項目蘑菇博客作者，分享一些我寫開源...

來自公眾號：陌溪教你學編程

大家好，我是**陌溪**。

最近，打算在 **Mac** 電腦上安裝一個虛擬機，用來學習 **K8S**。

首先想到的是 **Vmware** 和 **Parallels Desktop**，然後打開瀏覽器找了找，發現都只提供付費版本，只有 **Parallels Desktop** 提供了 **14** 天的免費試用。



無法白嫖

然後，開始嘗試使用免費 **VirtualBox**，但是在安裝 **CentOS** 的時候，貌似因為 **MacOS** 最新版本的原因，總是報錯，花費了好幾天的時間，試過各種方案，最終解決無果，就放棄了。。

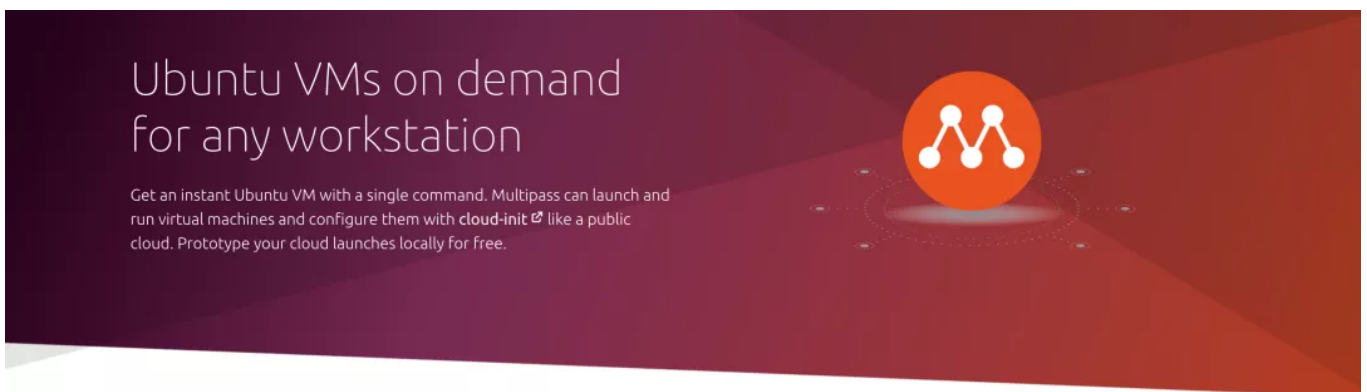
就在陌溪要放棄的時候，突然在 **Ubuntu** 官網進行沖浪的時候，發現了一款叫 **Multipass** 的東西。

它可以快速在電腦上快速搭建一個輕量級的虛擬機，並且相比於 **Vmware** 更加輕量，只需一行命令快速創建 **Ubuntu** 虛擬機。

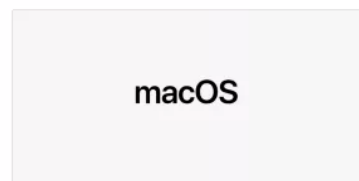
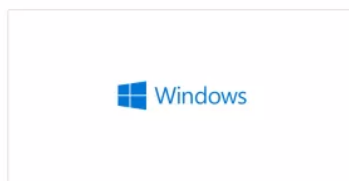
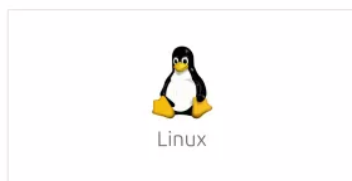
**Multipass** 是一個輕量虛擬機管理器，是由 **Ubuntu** 運營公司 **Canonical** 所推出的開源項目。運行環境支持 **Linux**、**Windows**、**macOS**。在不同的操作系統上，使用的是不同的虛擬化技術。在 **Linux** 上使用的是 **KVM**、**Window** 上使用 **Hyper-V**、**macOS** 中使用 **HyperKit** 以最小開銷運行VM，支持在筆記本模擬小型雲。

同時，**Multipass** 提供了一個命令行界面來啟動和管理 **Linux** 實例。下載一個全新的鏡像需要幾秒鐘的時間，並且在幾分鐘內就可以啟動並運行 **VM**。（直呼好傢伙，大大降低的我們安裝虛擬機的成本了）

Multipass官網：<https://multipass.run/>



Select OS to get started



Multipass官網

## 開始使用

在使用 **Multipass** 之前，首先需要安裝 **Multipass** 工具，可以打開官網進行下載

<https://multipass.run/>

[https://mp.weixin.qq.com/s/KodOpSL52BIL28\\_0rCwXhQ](https://mp.weixin.qq.com/s/KodOpSL52BIL28_0rCwXhQ)

選擇對應的操作系統，因為我的是 **Mac** 電腦，所以選擇 macOS，點擊 **Download** 開始下載

Select OS to get started



1 Install Multipass on MacOS

Download Multipass for MacOS

選擇對應版本安裝

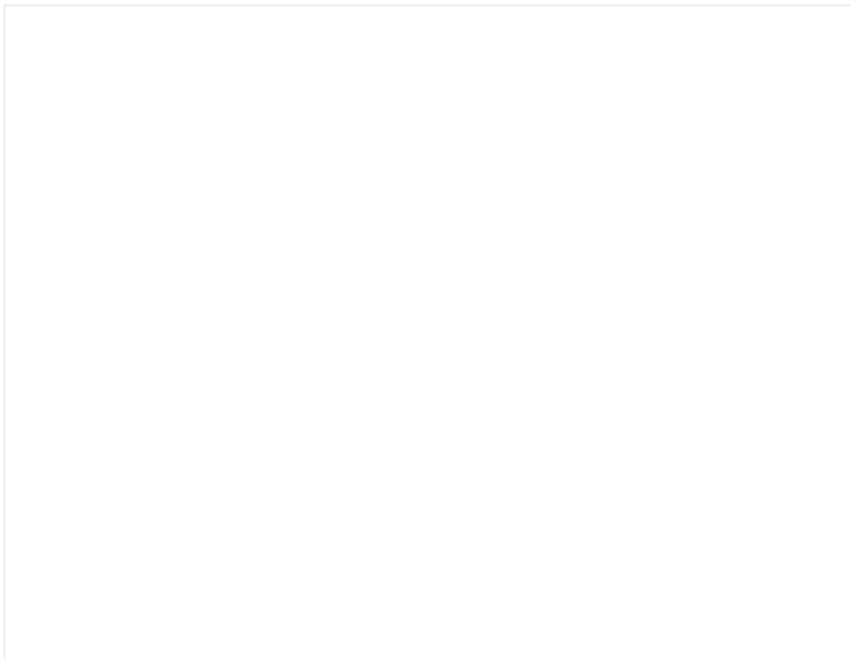
如果電腦裝了 **brew** 可以直接命令行下載安裝：

```
brew cask install multipass
```

下載安裝成功後，執行以下命令可以查看當前軟件版本：

```
multipass --version
```

安裝成功後，點擊運行，即可打開 **Multipass** 客戶端，這裡能夠快速的通過 **Open Shell** 創建和啟動一個虛擬機



Multipass客戶端

下面，我們來介紹如何通過命令的方式，創建一台虛擬機

## 創建Ubuntu虛擬機

首先，通過以下指令查看可供下載的 **Ubuntu** 鏡像

```
multipass find
```

運行成功後，可以看到下面的這些鏡像列表，包含各種版本的

Image	Aliases	Version	Description
core	core16	20200818	Ubuntu Core 16
core18		20200812	Ubuntu Core 18
18.04	bionic	20211109	Ubuntu 18.04 LTS
20.04	focal,lts	20211118	Ubuntu 20.04 LTS
21.04	hirsute	20211119	Ubuntu 21.04
21.10	impish	20211103	Ubuntu 21.10
appliance:adguard-home		20200812	Ubuntu AdGuard Home Appli
appliance:mosquitto		20200812	Ubuntu Mosquitto Applianc
appliance:nextcloud		20200812	Ubuntu Nextcloud Applianc
appliance:openhab		20200812	Ubuntu openHAB Home Appli
appliance:plexmediaserver		20200812	Ubuntu Plex Media Server
anbox-cloud-appliance		latest	Anbox Cloud Appliance
minikube		latest	minikube is local Kuberne

下载最新版的 **Ubuntu** 镜像并运行，初次创建时需要下载 **Ubuntu** 镜像，网络畅通的情况下，稍等片刻即可。

```
multipass launch -n vm01 -c 1 -m 1G -d 10G
```

自定义配置创建可以参考如下方式：

```
-n, --name: 名称  
-c, --cpus: cpu核心数，默认：1  
-m, --mem: 内存大小，默认：1G  
-d, --disk: 硬盘大小，默认：5G
```

## 操作虚拟机

### 查看虚拟机列表

虚拟机创建完成后，可以使用 **multipass list** 命令进行查看虚拟机列表

```
% multipass list  
Name                State      IPv4          Image  
vm01                Running    192.168.64.2  Ubuntu 20.04 LTS
```

可以看到目前正在运行一台 **Ubuntu 20.04** 版本的虚拟机，并且对应的 **IP** 地址为 **192.168.64.2**

### 外部操作虚拟机

通过 **multipass exec** 命令在实例内执行给定的命令。第一个参数是运行命令的实例，也就是 **vm01**

我们通过 **exec** 命令，就可以在外部操作刚刚创建的虚拟机，例如查看内部所处的目录，执行 **pwd** 命令

```
# multipass exec vm01 pwd  
/home/ubuntu
```

## 查看虚拟机信息

通过 **multipass info** 命令，即可查看当前运行的虚拟机信息

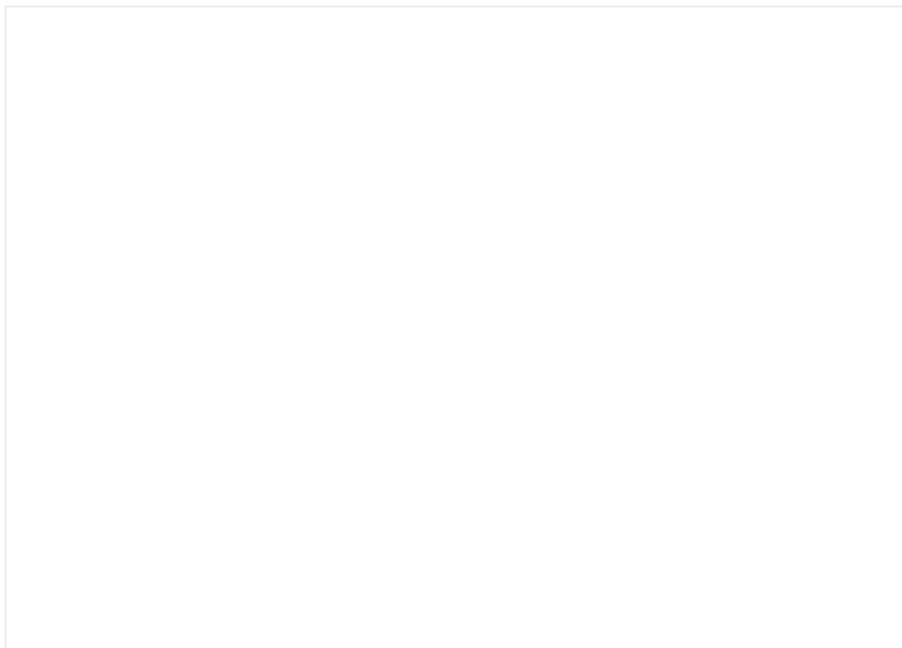
```
# multipass info vm01
Name:          vm01
State:         Running
IPv4:          192.168.64.2
Release:       Ubuntu 20.04.3 LTS
Image hash:    91740d72ffff (Ubuntu 20.04 LTS)
Load:          0.00 0.00 0.00
Disk usage:    1.3G out of 9.5G
Memory usage:  170.9M out of 1.9G
Mounts:        --
```

## 进入虚拟机

通过 **multipass shell** 命令，即可进入到虚拟机内部

```
multipass shell vm01
```

运行后的结果如下所示，可以看到目前虚拟机的一些系统配置信息，以及内存和磁盘的使用情况



进入虚拟机内部

同时会运行一个新的 **Ubuntu** 系统，此时便进入到了**Ubuntu** 环境中，在里面可以执行相关的 **linux** 指令

当然如果不想进入系统内部，也可以通过上述提到的 **multipass exce** 命令，来操作 **Ubuntu** 系统

首先执行下面命令，给系统设置一个 **root** 密码，设置好密码后，使用 **su root** 切换到 **root** 用户

```
# 设置密码
sudo passwd
# 切换 root
su root
```

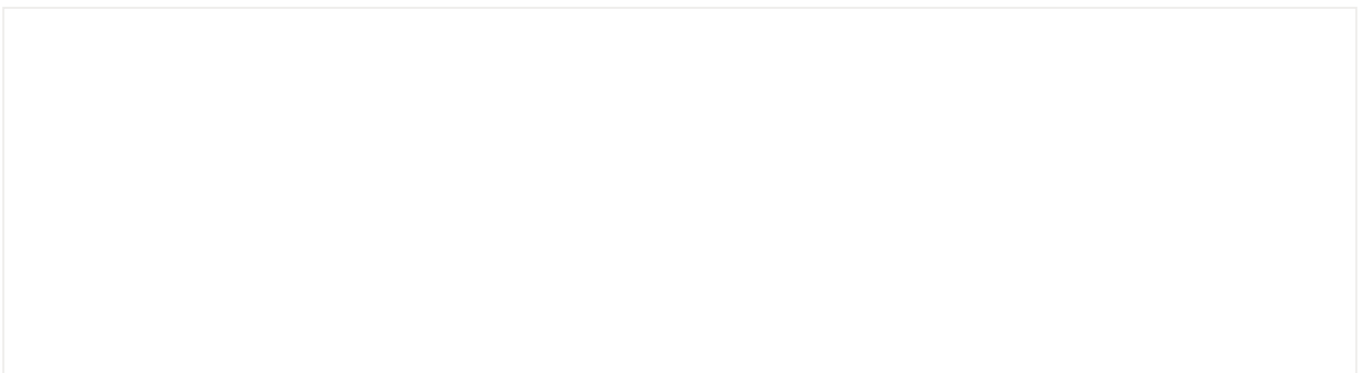
**ubuntu** 是使用 **apt-get** 来进行包管理的，首先更新一下 **apt-get**，然后安装 **nginx**

```
# 更新apt
apt-get update
# 安装 nginx
apt-get install nginx
```

安装好 **nginx** 后，可以到 **/etc/nginx** 目录即可看到刚刚的 **nginx** 配置信息

```
root@vm01:/lost+found# cd /etc/nginx/
root@vm01:/etc/nginx# ls
conf.d          fastcgi_params  koi-win        modules-available  nginx.conf      scgi_params
fastcgi.conf    koi-utf         mime.types     modules-enabled    proxy_params    sites-availab
```

下面我们可以测试一下，**nginx** 是否安装成功，在 **mac** 上打开浏览器，输入虚拟机的 **ip** 地址 **192.168.64.2**



可以看出，目前 **nginx** 已經成功運行了，後續的話，可能會考慮在上面部署更多的應用，感興趣的小伙伴歡迎關注～

## 挂载数据卷

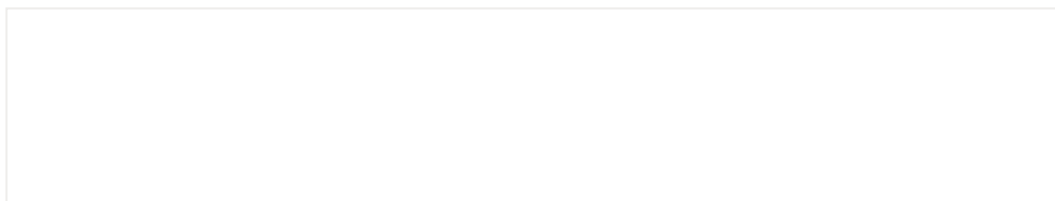
**multipass** 還提供和 **Docker** 一樣的挂载数据卷的功能，能夠與外部宿主機的文件保持同步。

```
# 挂载格式
multipass mount 宿主机目录 实例名:虚拟机目录
```

下面，我在用戶的目錄下，創建一個 **hello** 目錄

```
# 创建hello目录
mkdir hello
# 挂载
multipass mount /Users/moxi/hello vm01:/hello
```

挂载完成后，我們回到实例 **vm01** 中，可以看出多了一個 **hello** 文件夾，說明已經成功挂载～



挂载成功

因此，以後在 **vm01** 的 **hello文件夾** 中創建的文件，都會在的宿主機的 **hello文件夾** 同步顯示

## 卸载数据卷

如果以後不需要用到挂载了，可以使用 **unmount** 命令卸載

```
#卸載数据卷
multipass umount 容器名
```



## 传输文件

除了使用上述的 **mount** 挂载卷的方式实现文件的交互，同时还可以通过 **transfer** 命令，将宿主机的文件，发送到虚拟机内部

```
multipass transfer 主机文件 容器名:容器目录
```

例如，将 **hello.txt** 发送到

```
multipass transfer hello.txt vm01:/home/ubuntu/
```

## 删除和释放实例

使用下面的命令，可以开启、停止、删除和释放实例

```
# 启动实例
multipass start vm01
# 停止实例
multipass stop vm01
# 删除实例 (删除后，还会存在)
multipass delete vm01
# 释放实例 (彻底删除)
multipass purge vm01
```

## 容器配置自动化

为了保持开发环境和线上环境一致性 同时节省部署时间 **multipass** 给我们提供了 **--cloud-init** 选项进行容器启动初始化配置：

```
multipass launch --name ubuntu --cloud-init config.yaml
```

上面 **config.yaml** 则是容器的初始化配置文件，例如，我们想在初始化容器的时候，自动下载安装 **Node.js**，内容如下：

```
#cloud-config
```

```
runcmd:  
- curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -  
- sudo apt-get install -y nodejs
```

**runcmd** 可以指定容器 **首次启动** 时运行的命令

凡是用户自定义的cloud-init的配置文件,必须以#cloud-config开头, 这是cloud-init识别它的方式。

**yaml** 配置文件可以参考下面的文章

<https://cloudinit.readthedocs.io/en/latest/topics/examples.html?highlight=lock-passwd#ir>

更多关于 **multipass** 的高阶的技巧, 欢迎访问 **multipass** 官方文档

<https://multipass.run/docs/>

好了, 本期关于 **Multipass** 的基础学习就到这里了, 感兴趣的小伙伴环境下载自己动手学习呢~

我是陌溪, 我们下期再见~

## 总结

陌溪使用了一阵子后, **Multipass** 在使用起来很简洁直观, 在整体操作上和 **docker** 类似, 如果想在自己电脑上快速搭建一个 **Linux** 系统用于学习的话, 选择使用 **Multipass** 还是非常方便的。

唯一不足的地方就是, **Multipass** 是由 **Ubuntu** 背後的 **Canonical** 公司開發的, 因此 **Multipass** 所使用的鏡像也都是 **Ubuntu** 鏡像 (這也可以理解, 畢竟支持自家的產品), 不過對於習慣了使用 **CentOS** 的用戶來說, 還是用些許不太習慣。不過瑕不掩瑜, **Multipass** 在我看來還是一款非常棒的軟件~

--- EOF ---

[閱讀原文](#)

喜歡此內容的人還喜歡

為什麼阿里巴巴/騰訊不去開發被卡脖子的工業軟件？

計算機工作原理