

@[TOC](#)

首先去官网查看需要什么安装环境

# multipass

<https://multipass.run/docs>

```
1  # 准备条件:
2
3  # 关闭防火墙
4
5  # 初始化
6  multipass set local.driver=virtualbox
7  multipass set local.privileged-mounts=true
8
9  # 创建实例
10 multipass launch --network WLAN -d 10G -n primary
11 multipass launch --network WLAN -d 10G -n primary docker
12
13 # 挂载与取消挂载
14 multipass mount D:\doing\ubuntu primary:/home/ubuntu/zmj
15 multipass umount primary:Home
```

# docker

## install

```
1  #!/bin/sh
2  sudo apt-get remove docker docker-engine docker.io containerd runc
3
4  sudo apt-get update
5
6  sudo apt-get install -y ca-certificates curl gnupg lsb-release
7
8  sudo mkdir -p /etc/apt/keyrings
9
10 curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
   /usr/share/keyrings/docker-archive-keyring.gpg
11
12 echo \
13   "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-
   archive-keyring.gpg] https://download.docker.com/linux/ubuntu \
14   $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >
   /dev/null
15
```

```
16 sudo apt-get update
17
18 sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin
19
20
```

## dockerfile

- eg1

```
1 # FROM scratch
2 FROM ubuntu:20.04
3 RUN cp /etc/apt/sources.list /etc/apt/sources.list.bak \
4 && sed -ri "s/(archive|security).ubuntu.com/mirrors.tuna.tsinghua.edu.cn/g"
   /etc/apt/sources.list
```

- 规定

```
1 不指定目录默认在根目录
```

```
1 docker build -t ubuntu-z:20.04 .
```

## ubuntu

```
1 #!/bin/sh
2 # 更换源
3 cp /etc/apt/sources.list /etc/apt/sources.list.bak
4 sed -ri "s/(archive|security).ubuntu.com/mirrors.tuna.tsinghua.edu.cn/g"
   /etc/apt/sources.list
5 sed -ri "s/(archive|security).ubuntu.com/mirrors.aliyun.com/g" /etc/apt/sources.list
6 apt-get update
7 apt-get install --assume-yes apt-utils
8 apt-get -y install sudo vim fontconfig inetutils-ping locales wget git mysql-client
   make gcc build-essential python-dev python-setuptools python-pip python-smbus
   libncursesw5-dev \
9 libgdbm-dev libc6-dev zlib* libssl-dev libsqlite3-dev tk-dev libffi-dev
10 DEBIAN_FRONTEND=noninteractive apt-get -y install tzdata
11 # 修复依赖关系(depends)
12 sudo apt-get -f install
13
```

## dpkg

```
1 dpkg-reconfigure tzdata
```

## 环境变量

```
1 1. 什么是环境变量
2 环境变量 (environment variables) 一般是指在操作系统中用来指定操作系统运行环境的一些参数, 这些参数
   会对系统行为产生影响。
3
4 比如常用的PATH环境变量, 当要求系统运行一个程序而没有告诉它程序所在的完整路径时, 系统除了在当前目录下
   面寻找此程序外, 还会到PATH中指定的路径去找。你可以在终端使用printenv PATH查看当前PATH变量的值。
5
6 2. 用户环境变量和系统环境变量
7 Ubuntu系统包含两类环境变量: 系统环境变量和用户环境变量。系统环境变量对所有系统用户都有效, 用户环境变
   量仅仅对当前的用户有效。
8
9 用户环境变量可存储在以下文件中:
10 ~/.profile
11 ~/.bashrc, ~/.bash_profile, ~/.bash_login
12 推荐将环境变量保存在~/.profile中, 因为无论是通过控制台还是图形界面启动程序时, 都会自动执行该文件。
13 而~/.bashrc, ~/.bash_profile, ~/.bash_login这些文件, 当通过shell启动程序时, 它们也会被加载;
   但当通过图形界面环境启动程序时, 这些文件中的环境变量设置便不可用了。
14
15 系统环境变量可存储在以下文件中:
16 /etc/profile
17 /etc/profile.d (它是文件夹)
18 /etc/bash.bashrc
19 /etc/profile和/etc/profile.d都是常用的设置环境的地方。其中/etc/profile.d文件夹来源
   于/etc/profile, 在该目录下的*.sh, 即以sh为后缀的文件都会被加载。
20 类似地, 不推荐使用/etc/bash.bashrc, 因为在图形界面环境下启动程序时, 不会加载它里边的环境变量设置。
21
22 3. 设置永久环境变量实例 (以/etc/profile为例)
23 vim /etc/profile
24
25 在文件末尾处添加如下, 保存并退出:
26 export JAVA_HOME=/usr/lib/jvm/jdk1.7.0
27 export PATH=$PATH:$JAVA_HOME/bin
28 其中, export命令: 使得变量真正输出成为环境变量。
29
30 source /etc/profile
31
32
```

```
1 # C:\Program Files\Git\etc\gitconfig    系统
2 # C:\Users\zmj\.gitconfig    用户
3 git config -l
4 git config --system --list
5 git config --global -l
6 $ git config --global user.name "[name]"
7 对你的commit操作设置关联的用户名
8 $ git config --global user.email "[email address]"
9 对你的commit操作设置关联的邮箱地址
10 $ git config --global color.ui auto
11 启用有帮助的彩色命令行输出
12 git config --global user.name "钟马俊"
13 git config --global user.email 674495630@qq.com
```

## ssh

```
1 1. 安装ssh服务 : sudo apt-get install openssh-server
2 2. 查看ssh服务状态 : ps -e|grep ssh
3 3. 启动关闭: sudo service ssh start/restart/stop
4 4. 登录: ssh levi@192.168.100.5    # 登录
5 5. 退出: exit
6 6. scp拷贝
7     # 注意: `:` 后面的路径写绝对路径
8     scp demo.py levi@192.168.100.5:/home/tarena
9     # 把远程主目录下demo.py文件 复制到本地当前目录下
10    scp levi@192.168.100.5:/home/tarena/demo.py .
11    # 加上 -r 选项可以传送文件夹
12    scp -r demo levi@192.168.100.5:/home/tarena/
13 7. ssh密钥 使用方法
14    1. 在个人计算机中生产密钥对 : ssh-keygen  执行以后会在主目录下生成一个.ssh文件夹,其中包含私
    钥文件 id_rsa和公钥文件id_rsa.pub。
15    2. 在服务器主机上创建文件~/.ssh/authorized_keys, 将信任的计算机的id_rsa.pub文件内容追加到
    服务器 authorized_keys文件中。
16    chmod 600 authorized_keys
17    chmod 700 .ssh
18
19    ssh-keygen -t rsa -C "674495630@qq.com"
20
21
22 # 开启密码登录
23 vim /etc/ssh/sshd_config
24
25 PubkeyAuthentication yes #启用PublicKey认证
26 AuthorizedKeysFile .ssh/authorized_keys #PublicKey文件路径
27 PasswordAuthentication no #不适用密码认证登录    修改这项就可以
28
29 systemctl restart sshd
```

## selenium

```

1  # 配置浏览器驱动
2  【1】定义
3      phantomjs为无界面浏览器(又称无头浏览器), 在内存中进行页面加载, 高效
4  【2】下载地址
5      2.1) chromedriver : 下载对应版本
6          http://npm.taobao.org/mirrors/chromedriver/
7      2.2) geckodriver
8          https://firefox-source-docs.mozilla.org/testing/geckodriver/Support.html
9          http://npm.taobao.org/mirrors/geckodriver/
10         https://github.com/mozilla/geckodriver/releases
11      2.3) phantomjs
12         https://phantomjs.org/download.html
13  【3】Ubuntu安装
14      3.1) 下载后解压 : tar -zxvf geckodriver.tar.gz
15      3.2) 拷贝解压后文件到 /usr/bin/ (添加环境变量)
16          sudo cp geckodriver /usr/bin/
17      3.3) 添加可执行权限
18          sudo chmod 777 /usr/bin/geckodriver
19  【4】Windows安装
20      4.1) 下载对应版本的phantomjs、chromedriver、geckodriver
21      4.2) 把chromedriver.exe拷贝到python安装目录的Scripts目录下(添加到系统环境变量)
22          # 查看python安装路径: where python
23      4.3) 验证
24          cmd命令行: chromedriver
25  # 验证
26  【Ubuntu | Windows】
27  ipython3
28  from selenium import webdriver
29  webdriver.Chrome()
30  或者
31  webdriver.Firefox()
32  【mac】
33  ipython3
34  from selenium import webdriver
35  webdriver.Chrome(executable_path='/Users/xxx/chromedriver')
36  或者
37  webdriver.Firefox(executable_path='/User/xxx/geckodriver')

```

## mysqlclient

```

1  # ubuntu
2  1. 安装 mysqlclient [版本 mysqlclient 1.3.13以上, 官网目前为1.4.x]
3      - 安装前确认ubuntu是否已安装 python3-dev 和 default-libmysqlclient-dev
4
5      - sudo apt list --installed|grep -E 'libmysqlclient-dev|python3-dev'
6
7  - 若命令无输出则需要安装 - sudo apt-get install python3-dev default-libmysqlclient-dev
8
9      - 确保上述两个库已经安装, 执行 sudo pip3 install mysqlclient即可
10 # ubuntu
11 sudo apt install nginx
12

```

```
13 # ubuntu
14 查看已安装的库
15     sudo pip3 list|grep -i 'uwsgi'
16     sudo pip3 freeze|grep -i 'uwsgi'
17     -i 参数指 不区分大小写
18     如果pip3安装过uwsgi, 则会输出
19     uWSGI==2.0.18
20     uWSGI 2.0.18
```

## mysql

- client

```
1 apt-get install mysql-client
2
```

```
1 show databases;
2 use mysql;
3 update user set authentication_string=PASSWORD("自定义密码") where user='root';
4 update user set plugin="mysql_native_password";
5 flush privileges;
6 https://www.cnblogs.com/xiwusheng/p/10925669.html
7 查看: mysql -V
8 【1】修改配置文件, 允许远程连接
9 linux开启连接步骤:
10     sudo gedit /etc/mysql/mysql.conf.d/mysqld.cnf
11     将如下行注释并保存退出:
12     # bind-address = 127.0.0.1
13 windows开启连接步骤:
14 1.进入mysql命令行
15 2.输入use mysql; 回车
16 3.输入update user set host = '%' where user = 'root'; 回车
17 4.输入flush privileges; 回车
18 5.输入quit; 退出
19
20 【2】给用户授权
21     mysql -uroot -p123456
22     mysql> grant all privileges on *.* to 'root'@'%' identified by '123456' with
grant option;
23     mysql> flush privileges;
24
25 【3】重启MySQL服务
26     sudo /etc/init.d/mysql restart
27
28 【4】远程连接测试 (远程服务器上)
29     mysql -hIP地址 -uroot -p123456
```

## Important

MySQL 8.0 Server requires the Microsoft Visual C++ 2015 Redistributable Package to run on Windows platforms. Users should make sure the package has been installed on the system before installing the server. The package is available at the [Microsoft Download Center](#). Additionally, MySQL debug binaries require Visual Studio 2015 to be installed.

MySQL 8.0 Server 需要 Microsoft Visual C++ 2015 Redistributable Package 才能在 Windows 平台上运行。用户在安装服务器之前应确保系统上已安装该软件包。该软件包可从 [Microsoft 下载中心](#) 获得。此外，MySQL 调试二进制文件需要安装 Visual Studio 2015。

- my.ini

```
1 [client]
2 default-character-set=utf8
3 [mysql]
4 default-character-set=utf8
5
6 [mysqld]
7 port=3306
8 basedir=C:/Program Files/MySQL/mysql-5.7.32-winx64
9 datadir=C:/Program Files/MySQL/mysql-5.7.32-winx64/data
10 max_connections=200
11 max_connect_errors=10
12 character-set-server=utf8
13 default-storage-engine=INNODB
14 server-id=1
15 loose_mysqlx_port=33060
```

```
1 1 mysql --initialize-insecure --console
2 # mysql --defaults-file="C:\Program Files\MySQL\mysql-8.0.23-winx64\my.ini" --
  initialize-insecure --console
3 2 mysql --install
4 # .\mysql --install mysql8.0.23 --defaults-file="C:\Program Files\MySQL\mysql-
  8.0.23-winx64\my.ini"
5 # 更改注册表
6 # HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
7 3 net start mysql8.0.23
8 4 mysql -uroot -p
9 5 ALTER USER 'root'@'localhost' IDENTIFIED BY '新密码'
10 # mysqladmin -u用户名 -p旧密码 password 新密码
11 # mysql --remove
12 # sc delete mysql
13 # mysql --console # 前台启动
14 # net start mysql
15 # net stop mysql
16 # "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqladmin" -u root shutdown
17
18 # 查看启动的服务
19 net start | find /i "mysql"
20 # 查看启动的进程
```

```
21 tasklist
```

~/vimrc

```
1 set termencoding=utf-8
2 set encoding=utf8
3 set fileencodings=utf8
```

# python

## install

```
1 # Ubuntu
2
3 ##### 在线安装python
4 ubuntu 系统级组件 如下命令安装
5 sudo apt-get install 名字
6 sudo apt install software-properties-common -y
7 # 第三方维护的PPA软件源来方便的安装所需要的Python版本
8 sudo add-apt-repository ppa:deadsnakes/ppa
9 sudo apt update
10 sudo apt install python3.7
11
12 https://packaging.python.org/guides/installing-using-linux-tools/#debian-ubuntu
13 sudo apt install python3-venv python3-pip
14
15 ##### 离线安装python
16 # 安装依赖
17 https://devguide.python.org/setup/#install-dependencies
18 sudo apt-get install pkg-config
19 sudo apt-get install make gcc build-essential zlib*
20
21 tar -zxvf Python-3.6.1.tgz
22 ./configure --with-pydebug --prefix=/usr/local/bin/python3.8.2
23 make && make install
24
25
```

## pip



- pip.conf

```
1 [global]
2 index-url = https://pypi.tuna.tsinghua.edu.cn/simple
3 [install]
4 trusted-host = https://pypi.tuna.tsinghua.edu.cn
```

```
1 ##### 配置pip源
2 https://pip.pypa.io/en/stable/user_guide/#configuration
3
4
5 python -m pip install -U pip
6 -m:导入模块
7 -U: 升级。原来已经安装的包，带上U才会更新到最新版本，不带U不会装新版本。
8 pip3 - Python库的管理工具
9     sudo pip3 install uwsgi==2.0.18 -i https://pypi.tuna.tsinghua.edu.cn/simple/
10 -i 参数 指定当前 安装命令 去哪个网站下载 安装包
11     清华: https://pypi.tuna.tsinghua.edu.cn/simple
12     中国科技大学 https://pypi.mirrors.ustc.edu.cn/simple/
13     华中理工大学: http://pypi.hustunique.com/
14     山东理工大学: http://pypi.sdutlinux.org/
15     豆瓣: http://pypi.douban.com/simple/
16
17 查看已安装的库
18 sudo pip3 list|grep -i 'uwsgi'
19 -i 参数指 不区分大小写
20 pip3 show requests
21 pip freeze > requirements.txt
22
23 pip --help
24 pip list -h
25 pip list -o
26 # 卸载
27 sudo pip3 uninstall django
28
29 ##### 离线安装python第三方库
30 1、 tar -zxvf django-cors-headers-3.0.2.tar.gz
31     tar -xvf Django-2.2.12.tar.gz`
32 2、 cd django-cors-headers-3.0.2
33 3、 sudo python3 setup.py install
34 4、 pip3 freeze|grep -i 'cors'
35
36 ##### 在线安装python第三方库
37 sudo pip3 install django==2.2.12 -i https://pypi.mirrors.ustc.edu.cn/simple/
38                                     https://pypi.tuna.tsinghua.edu.cn/simple/
39                                     http://mirrors.aliyun.com/pypi/simple/
40                                     http://pypi.douban.com/simple/
41
42
43 ##### python虚拟环境安装
```

```
44 # https://docs.python.org/zh-cn/3/tutorial/venv.html
45 # 方法一
46 1 python3 -m venv tutorial-env
47 2 source tutorial-env/bin/activate
48 # 方法二
49 pycharm创建虚拟环境
50
51 ##### 多版本python中pip配置
52 # 例如: python3.6和python3.8
53 1. sudo cp pip pip3.6
54 2. sudo cp pip pip3.8
55 3. sudo vim pip3.6或者pip3.8
56 4. 更改第一行python所在路径
57
58 ##### 多版本python中的ipython库的配置
59 # 1 找到ipython的位置
60 which ipython
61 # 2 复制一份
62 例如: cp ipython ipython3.8
63 # 3 打开文件更改相应的版本号变量
64 vim ipython3.8
```

## 模块搜索路径

---

### 添加方式

1. PYTHONPATH
2. 在site-packages中创建\*.pth文件
3. sys.path
4. 将模块直接放入site-packages
5. 集成开发环境

## chrome

---

```
1 wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
2 dpkg -i google-chrome-stable_current_amd64.deb
3 sudo apt-get -f install
4 google-chrome --version
```

## jdk

---

```

1  # 解压
2  tar -zxvf jdk-11.0.15_linux-x64_bin.tar.gz -C /usr/local
3  vim .bashrc
4  # 添加一下内容 保存
5  export JAVA_HOME=/usr/local/jdk-11.0.15
6  export CLASS_PATH=.:${JAVA_HOME}/lib
7  export PATH=${JAVA_HOME}/bin:$PATH
8  # 使文件生效
9  source .bashrc
10 # 测试
11 java -version
12

```

## tomcat

```

1  tar -zxvf apache-tomcat-8.5.81.tar.gz -C /usr/local/
2  # 启动
3  ./startup.sh //直接启动
4  nohup ./startup.sh & //作为服务启动
5  ./catalina.sh run //控制台动态输出方式启动，动态的显示tomcat控制台输出信息，Ctrl+c退出并停止服务

```

- \$CATALINA\_HOME/bin/setenv.sh

```

1  # 解决jenkins日志乱码
2  JAVA_OPTS="$JAVA_OPTS -Dfile.encoding=utf-8 --illegal-access=warn -
    Dhudson.footerURL=https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-
    center.json"

```

- \$CATALINA\_HOME/webapps/manager/META-INF/context.xml

```

1  # 限制ip访问 (manager 应用)
2  <Context antiResourceLocking="false" privileged="true" >
3    <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
4      sameSiteCookies="strict" />
5    <!--
6    <Valve className="org.apache.catalina.valves.RemoteAddrValve"
7      allow="127\.\d+\.\d+|:::1|0:0:0:0:0:0:0:1" />
8    -->
9    <Valve className="org.apache.catalina.valves.RemoteAddrValve"
10     allow="127\.\d+\.\d+|:::1|0:0:0:0:0:0:0:1|192.168.1.163" />
11    <Manager sessionAttributeValueClassNameFilter="java\.lang\.(
    (?::Boolean|Integer|Long|Number|string)|org\.apache\.catalina\.filters\.CsrfPreven
    tionFilter\.$LruCache(?:\.$1)?|java\.util\.(?:Linked)?HashMap"/>
12  </Context>

```

- \$CATALINA\_HOME/conf/tomcat-users.xml

```
1 # 设置用户名和密码 (manager 应用)
2 <!--
3 <user username="admin" password="<must-be-changed>" roles="manager-gui"/>
4 <user username="robot" password="<must-be-changed>" roles="manager-script"/>
5 -->
6 <user username="admin" password="<must-be-changed>" roles="manager-gui,manager-
  status"/>
```

## jenkins

```
1 # 后台运行
2 nohup java --illegal-access=warn -
  Dhudson.footerURL=https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-
  center.json -jar jenkins.war --httpPort=8080 &
3
4 # 采用通用的URL方式, 就可以实现Jenkins的停止, 重启和重载。
5 http://[jenkins-server-address][:port]/[command] where [command] can be
6 exit to shutdown jenkins
7 restart to restart jenkins
8 reload to reload the configuration
```

- ./jenkins

```
1 # 更新插件地址
2 vim hudson.model.UpdateCenter.xml
3
4 https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-center.json
5 http://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-center.json
6 https://mirrors.aliyun.com/jenkins/updates/update-center.json
7 https://mirrors.huaweicloud.com/jenkins/updates/update-center.json
```

## jmeter

```
1 jmeter运行环境搭建
2 1 需要安装JDK
3   - JDK--Java开发工具包
4   - JRE--Java运行时环境
5   - JVM--Java虚拟机
6 2 验证机器是否安装好Java环境
```

```
7      - java -version
8      - java 验证系统的环境变量path是否设置ok
9  3. 设置环境变量-目标: 任意路径可以识别jmeter
10     - JMETER_HOME
11         eg:- D:\BaiduNetdiskDownload\apache-jmeter-5.4.1
12     - PATH
13         eg:- D:\BaiduNetdiskDownload\apache-jmeter-5.4.1\bin
14     - %JMETER_HOME%\bin-推荐使用这个
```