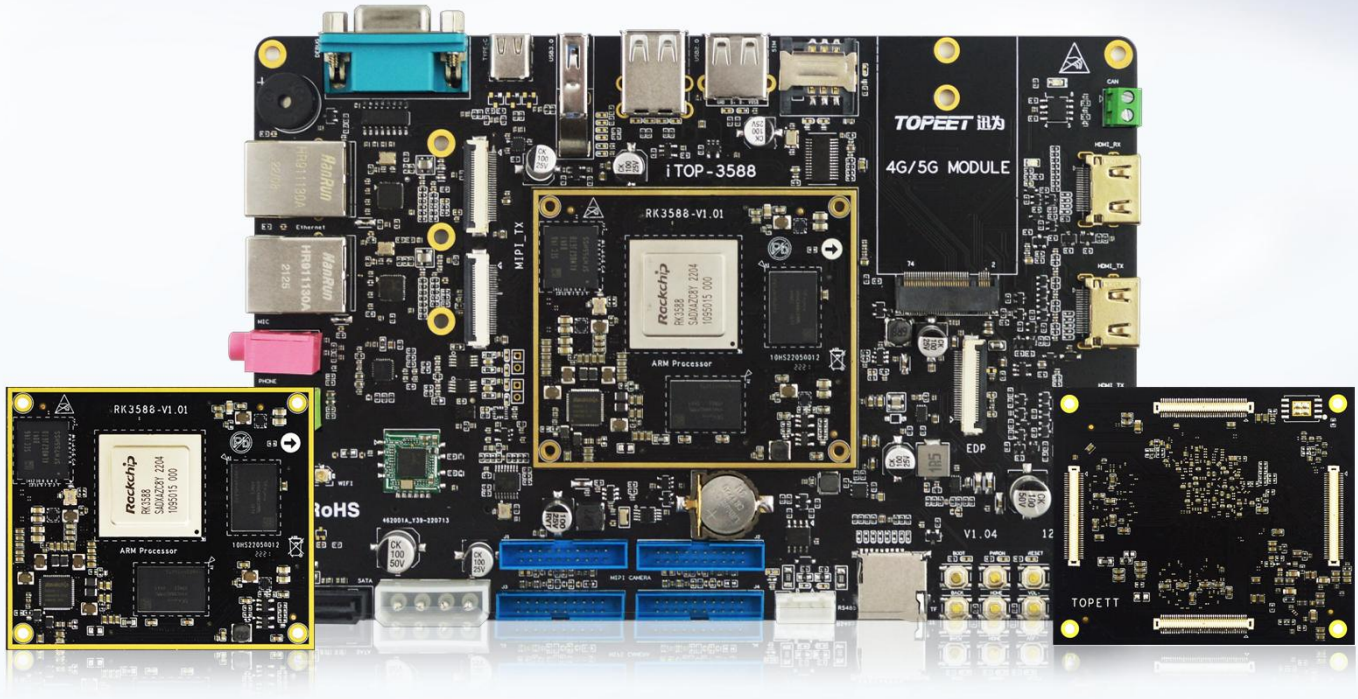
TOPEET北京迅为

强大的 AI 能力 更快更强

超长供货周期 | 7X24 小时稳定运行 | 8K 视频编解码



iTOP-RK3588 开发板使用手册

八核 64 位 CPU | 主频 2.4GHz | NPU 算力 6T | 4800 安防级别 ISP

更新记录

更新版本	修改内容
V1.0	初版

目录

更新记录 2
目录 3
版权声明 4
更多帮助 5
第 1 章 Docker 配置教程 6
1.1 docker 简介 6
1.2 Ubuntu 安装 Docker 7
1.2.1 设置 Docker 仓库 7
1.2.2 安装 Docker Engine-Community 9
1.2.3 使用中科大镜像加速器 10
1.2.4 把 Docker 配置为普通用户访问 11
1.3 Docker 安装 Ubuntu 20.04 12
1.4 Docker 安装常用工具 13
1.5 下载文件到 docker 16
1.6 Docker 常用命令 17

版权声明

更多帮助

注意事项与维护

- 请注意和遵循标注在产品上的所有警示和指引信息；
- 请勿带电插拔核心板及外围模块；
- 使用产品之前，请仔细阅读本手册，并妥善保管，以备将来参考；
- 请使用配套电源适配器，以保证电压、电流的稳定；
- 请勿在冷热交替环境中使用本产品，避免结露损坏元器件；
- 请保持产品干燥，如果不慎被任何液体泼溅或浸润，请立刻断电并充分晾干；
- 请勿使用有机溶剂或腐蚀性液体清洗本产品；
- 请勿在多尘、脏乱的环境中使用本产品，如果长期不使用，请包装好本产品；
- 如果在震动场景使用，请做好核心板与底板的固定，避免核心板跌落损坏；
- 请勿在通电情况下，插拔核心板及外围模块(特别是串口模块)；
- 请勿自行维修、拆解本产品，如产品出现故障应及时联系本公司进行维修；
- 请勿自行修改或使用未经授权的配件，由此造成的损坏将不予保修；

资料的更新

为了确保您的资料是最新状态，请密切关注我们的动态，我们将会通过微信公众号和 QQ 群推送。

关注“迅为电子”微信公众号，不定期分享教程、资料和行业干货及产品一线资料。

迅为新媒体账号

官网：https://www.topeetboard.com

知乎 https://www.zhihu.com/people/topeetabc123

CSDN：https://blog.csdn.net/BeiJingXunWei

售后服务政策

1. 如产品使用过程中出现硬件故障可根据售后服务政策进行维修
2. 服务政策：参见官方网售后服务说明 https://www.topeetboard.com/sydymfl/Service/bx.html

送修地址：

1. 地址：北京市海淀区永翔北路 9 号中国航发大厦三层
2. 联系人：迅为开发板售后服务部
3. 电话：010-85270716
4. 邮编：100094
5. 邮寄须知：建议使用顺丰、圆通或韵达，且不接受任何到付

技术支持范围

1. 了解产品的软、硬件资源提供情况咨询
2. 产品的软、硬件手册使用过程中遇到的问题
3. 下载和烧写更新系统过程中遇到的问题
4. 产品用户的资料丢失、更新后重新获取
5. 产品的故障判断及售后维修服务。

PS：（由于嵌入式系统知识范围广泛，我们无法保证对各种问题都能一一解答，部分内容无法供技术支持，只能提供建议。 ）

技术支持

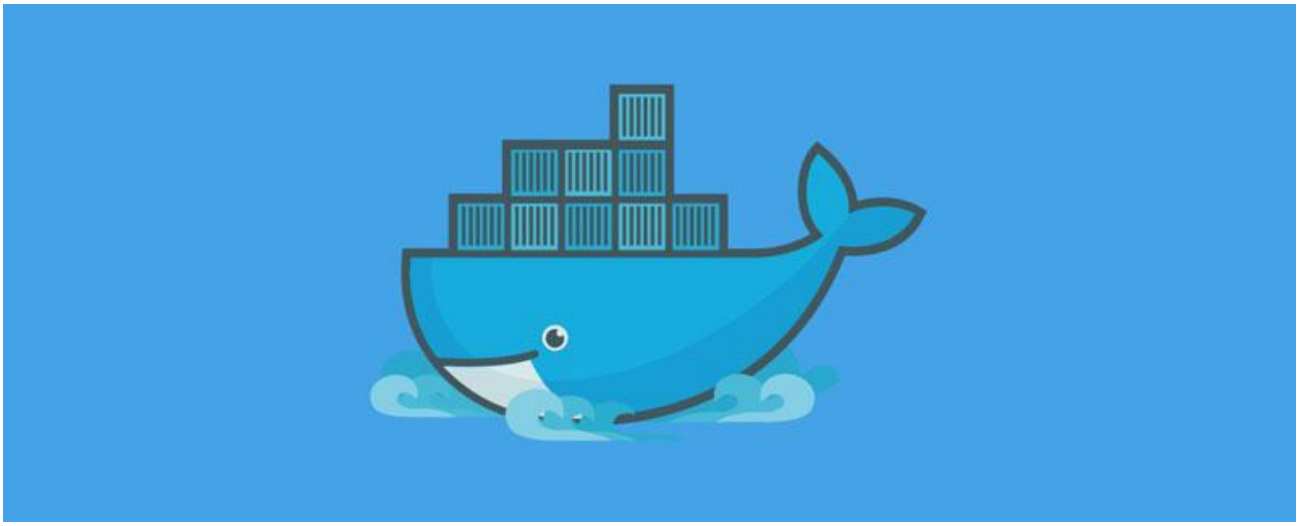
1. 周一至周五：（法定节假日除外）
上午 9:00 ~ 11:30 / 下午 13:30 ~ 17:30

2. QQ 技术交流群： 824412014
822183461
95631883
861311530

第 1 章 Docker 配置教程

1.1 docker 简介

Docker 是一种容器，那么什么是容器呢？



在使用迅为 RK3588 开发板的时候，我们一般采用的是虚拟机安装 Ubuntu20.04 来编译 Android 源码或者 linux 源码，但是编译源码最让人头疼的是主机环境问题。假如我手上有很多块开发板，每个开发板都使用不同的编译环境，而我本地电脑已经有一个编译环境了，那怎么办呢？

有没有一种统一，虚拟的软件硬件平台，客户可以直接使用这个平台来编译源码？

答案是有的，像这样的平台就是容器，容器有很多种，Docker 是其中比较好用的。

在使用 docker 之前，我们需要了解 Docker 的三个基本概念：镜像（Image）、容器（Container）、仓库（Repository）。接下来我们依次讲解这三个概念。

Docker 镜像

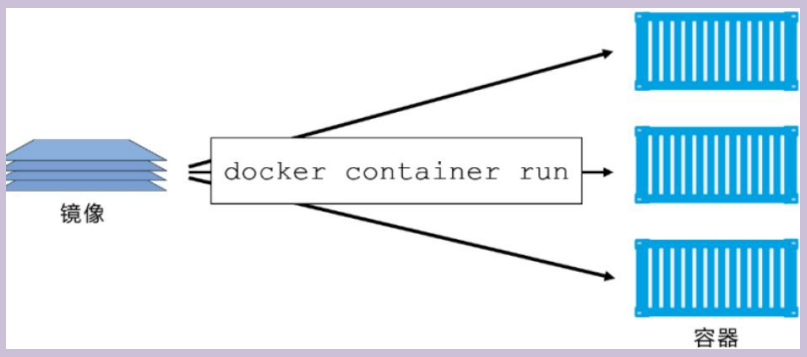
Docker 镜像（Image）类似于虚拟机的镜像，可以将它理解为一个面向 Docker 引擎的只读模板，包含了文件系统。镜像就是一个环境包，这个环境包可以移动到任意的 Docker 平里运行。

镜像是创建 Docker 容器的基础，通过版本管理和增量的文件系统，Docker 提供了一套十分简单的机制来创建和更新现有的镜像。用户可以从网上下载一个已经做好的应用镜像，并通过命令直接使用。总之，应用运行是需要环境的，而镜像就是来提供这种环境。

Docker 容器

Docker 容器（Container）类似于一个轻量级的沙箱子（因为 Docker 是基于 Linux 内核的虚拟技术，所以消耗资源十分少），Docker 利用容器来运行和隔离应用。

容器是镜像的运行时实例，一个镜像可以启动多个容器。它可以被启动、开始、停止、删除。每个容器都是相互隔离的、保证安全的平台。如下图所示：



Docker 仓库

Docker 仓库（Repository）类似与代码仓库，是 Docker 集中存放镜像文件的场所。

每个仓库可以包含多个标签，每个标签对应一个镜像。通常，一个仓库会包含同一个软件不同版本的镜像，而标签就常用于对应该软件的各个版本。

1.2 Ubuntu 安装 Docker

接下来我们在虚拟机的 Ubuntu(任何版本)中安装 docker。

1.2.1 设置 Docker 仓库

（1）更新 apt 包索引，输入以下命令：

```
sudo apt-get update
```

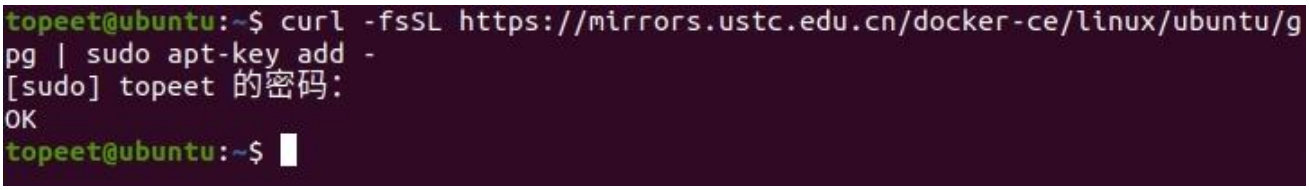
```
topeet@ubuntu:~$ sudo apt-get update
[sudo] topeet 的密码:
命中:1 http://mirrors.aliyun.com/ubuntu focal InRelease
获取:2 http://mirrors.bwbot.org stable InRelease [5,525 B]
获取:3 http://mirrors.aliyun.com/ubuntu focal-updates InRelease [114 kB]
命中:4 http://packages.microsoft.com/repos/code stable InRelease
获取:5 http://mirrors.aliyun.com/ubuntu focal-backports InRelease [108 kB]
获取:6 http://mirrors.aliyun.com/ubuntu focal-security InRelease [114 kB]
获取:7 http://mirrors.aliyun.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [278 kB]
获取:8 http://mirrors.aliyun.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [391 kB]
获取:9 http://mirrors.aliyun.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [944 B]
获取:10 http://mirrors.aliyun.com/ubuntu focal-backports/main amd64 DEP-11 Metadata [8,008 B]
获取:11 http://mirrors.aliyun.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [30.5 kB]
获取:12 http://mirrors.aliyun.com/ubuntu focal-security/main amd64 DEP-11 Metadata [40.7 kB]
获取:13 http://mirrors.aliyun.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [77.8 kB]
获取:14 http://mirrors.aliyun.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2,464 B]
已下载 1,171 kB，耗时 8秒 (154 kB/s)
正在读取软件包列表... 完成
topeet@ubuntu:~$
```

（2）安装 apt 依赖包，用于通过 HTTPS 来获取仓库，输入以下命令：

```
sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg-agent \
    software-properties-common
```

（3）添加 Docker 的官方 GPG 密钥，输入以下命令：

```
curl -fsSL https://mirrors.ustc.edu.cn/docker-ce/linux/ubuntu/gpg | sudo apt-key add -
```



（4）9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88 通过搜索指纹的后 8 个字符，验证现在是否拥有带有指纹的密钥，输入以下命令：

```
sudo apt-key fingerprint 0EBFCD88
```

（5）使用以下指令设置稳定版仓库。

```
sudo add-apt-repository \
    "deb [arch=amd64] https://mirrors.ustc.edu.cn/docker-ce/linux/ubuntu/ \
   $(lsb_release -cs) \
   stable"
```

```
topeet@ubuntu:~$ sudo add-apt-repository \
>    "deb [arch=amd64] https://mirrors.ustc.edu.cn/docker-ce/linux/ubuntu/ \
>   $(lsb_release -cs) \
>   stable"
获取:1 http://mirrors.bwbot.org stable InRelease [5,525 B]
获取:2 https://mirrors.ustc.edu.cn/docker-ce/linux/ubuntu focal InRelease [57.7
kB]
命中:3 http://mirrors.aliyun.com/ubuntu focal InRelease
命中:4 http://mirrors.aliyun.com/ubuntu focal-updates InRelease
命中:5 http://packages.microsoft.com/repos/code stable InRelease
获取:6 https://mirrors.ustc.edu.cn/docker-ce/linux/ubuntu focal/stable amd64 Pac
kages [18.5 kB]
命中:7 http://mirrors.aliyun.com/ubuntu focal-backports InRelease
命中:8 http://mirrors.aliyun.com/ubuntu focal-security InRelease
已下载 81.7 kB，耗时 1秒 (114 kB/s)
正在读取软件包列表... 完成
topeet@ubuntu:~$
```

1.2.2 安装 Docker Engine-Community

（1）更新 apt 包索引，输入以下命令：

```
sudo apt-get update
```

```
topeet@ubuntu:~$ sudo apt-get update
命中:1 http://mirrors.aliyun.com/ubuntu focal InRelease
获取:2 http://mirrors.bwbot.org stable InRelease [5,525 B]
命中:3 http://mirrors.aliyun.com/ubuntu focal-updates InRelease
命中:4 https://mirrors.ustc.edu.cn/docker-ce/linux/ubuntu focal InRelease
命中:5 http://mirrors.aliyun.com/ubuntu focal-backports InRelease
命中:6 http://mirrors.aliyun.com/ubuntu focal-security InRelease
命中:7 http://packages.microsoft.com/repos/code stable InRelease
已下载 5,525 B，耗时 1秒 (7,453 B/s)
正在读取软件包列表... 完成
topeet@ubuntu:~$
```

（2）安装最新版本的 Docker Engine-Community 和 containerd，输入以下命令：

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

```
topeet@ubuntu:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了：
  binutils-aarch64-linux-gnu cpp-9-aarch64-linux-gnu cpp-aarch64-linux-gnu
  gcc-10-cross-base gcc-9-aarch64-linux-gnu-base gcc-9-cross-base
  libasan5-arm64-cross libatomic1-arm64-cross libc6-arm64-cross
  libc6-dev-arm64-cross libfwupdplugin1 libgcc-9-dev-arm64-cross
  libgcc-s1-arm64-cross libgomp1-arm64-cross libitm1-arm64-cross
  liblsan0-arm64-cross libstdc++6-arm64-cross libtsan0-arm64-cross
  libubsan1-arm64-cross linux-libc-dev-arm64-cross
使用'sudo apt autoremove'来卸载它(它们)。
将会同时安装下列软件：
  docker-ce-rootless-extras docker-scan-plugin pigz slirp4netns
```

（3）测试 Docker 是否安装成功，输入以下指令，打印出以下信息则安装成功

```
sudo docker run hello-world
```

```
topeet@ubuntu:~$ sudo docker run hello-world

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/

topeet@ubuntu:~$
```

1. 2. 3 使用中科大镜像加速器

（1）添加镜像源：

```
sudo vim /etc/docker/daemon.json
```

输入以下内容：

```
{
      "registry-mirrors": ["https://docker.mirrors.ustc.edu.cn"]
}
```

```
{
    "registry-mirrors": ["https://docker.mirrors.ustc.edu.cn"]
}
```

（2）重启 docker 服务：

```
sudo systemctl restart docker
```

```
topeet@ubuntu:~$
topeet@ubuntu:~$ sudo systemctl restart docker
topeet@ubuntu:~$
```

1.2.4 把 Docker 配置为普通用户访问

（1）添加 docker 用户组，输入以下命令：

```
sudo groupadd docker
```

（2）将登陆用户加入到 docker 用户组中：

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

（3）更新用户组（这一步非常重要）：

```
newgrp docker
```

```
topeet@ubuntu:~$ newgrp docker
topeet@ubuntu:~$
```

（4）重启 docker 服务：

```
sudo systemctl enable docker
sudo systemctl restart docker
```

```
topeet@ubuntu:~$ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /lib/systemd
/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable docker
topeet@ubuntu:~$
topeet@ubuntu:~$
topeet@ubuntu:~$ sudo systemctl restart docker
```

（5）直接普通用户运行 hello-world，输入以下命令：

```
docker run hello-world
```

```
topeet@ubuntu:~$ docker run hello-world

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/

topeet@ubuntu:~$
```

1.3 Docker 安装 Ubuntu 20.04

Docker 下载安装 Ubuntu20.04，输入以下命令：

```
sudo apt update
docker pull ubuntu:20.04
```

```
topeet@ubuntu:~$ docker pull ubuntu:20.04
20.04: Pulling from library/ubuntu
527f5363b98e: Pull complete
Digest: sha256:f2034e7195f61334e6caff6ecf2e965f92d11e888309065da85ff50c617732b8
Status: Downloaded newer image for ubuntu:20.04
docker.io/library/ubuntu:20.04
topeet@ubuntu:~$
```

切换 Shell 到 Ubuntu 20.04，输入以下命令：

```
docker container run -p 8000:3000 -it ubuntu:20.04 /bin/bash
```

```
topeet@ubuntu:~$ docker container run -p 8000:3000 -it ubuntu:20.04 /bin/bash
root@4d3d85984e86:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@4d3d85984e86:/#
root@4d3d85984e86:/#
root@4d3d85984e86:/#
```

-p 参数：容器的 3000 端口映射到本机的 8000 端口。

-it 参数：容器的 Shell 映射到当前的 Shell，然后你在本机窗口输入的命令，就会传入容器。

ubuntu:20.04：image 文件的名字

/bin/bash：容器启动以后，内部第一个执行的命令。这里是启动 Bash，保证用户可以使用 Shell。

1.4 Docker 安装常用工具

输入以下命令安装 ubuntu20.04 常用工具。

```
apt update
apt install byobu vim-gtk inetutils-ping net-tools wget cpio unzip rsync xz-utils bc time
```

```
root@bc1d9b59bc0b:/#
root@bc1d9b59bc0b:/# apt install byobu vim-gtk inetutils-ping net-tools wget cpi
o unzip rsync xz-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme ca-certificates dbus file fontconfig fontconfig-config
  fonts-dejavu-core fonts-lato gawk gettext-base gtk-update-icon-cache
  hicolor-icon-theme humanity-icon-theme javascript-common krb5-locales
  libapparmor1 libatk1.0-0 libatk1.0-data libavahi-client3
  libavahi-common-data libavahi-common3 libbsd0 libcairo2 libcroco3 libcups2
  libdatrie1 libdbus-1-3 libevent-2.1-6 libexpat1 libfontconfig1 libfreetype6
  libfribidi0 libgail-common libgail18 libgdbm-compat4 libgdbm5
  libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libglib2.0-0
  libglib2.0-data libgpm2 libgraphite2-3 libgssapi-krb5-2 libgtk2.0-0
```

过程中需要配置地区，选择亚洲上海：

```
Geographic area: 6

Please select the city or region corresponding to your time zone.

  1. Aden          24. Dubai         47. Kuching       70. Shanghai
  2. Almaty        25. Dushanbe      48. Kuwait        71. Singapore
  3. Amman         26. Famagusta     49. Macau         72. Srednekolymsk
  4. Anadyr        27. Gaza          50. Magadan       73. Taipei
  5. Aqtau         28. Harbin        51. Makassar      74. Tashkent
  6. Aqtobe        29. Hebron        52. Manila        75. Tbilisi
  7. Ashgabat      30. Ho_Chi_Minh   53. Muscat        76. Tehran
  8. Atyrau        31. Hong_Kong     54. Nicosia       77. Tel_Aviv
  9. Baghdad       32. Hovd          55. Novokuznetsk  78. Thimphu
  10. Bahrain      33. Irkutsk       56. Novosibirsk   79. Tokyo
  11. Baku         34. Istanbul      57. Omsk          80. Tomsk
  12. Bangkok      35. Jakarta       58. Oral          81. Ujung_Pandang
  13. Barnaul      36. Jayapura      59. Phnom_Penh    82. Ulaanbaatar
  14. Beirut       37. Jerusalem     60. Pontianak     83. Urumqi
  15. Bishkek      38. Kabul         61. Pyongyang     84. Ust-Nera
  16. Brunei       39. Kamchatka     62. Qatar         85. Vientiane
  17. Chita        40. Karachi       63. Qostanay      86. Vladivostok
  18. Choibalsan   41. Kashgar       64. Qyzylorda     87. Yakutsk
  19. Chongqing    42. Kathmandu     65. Rangoon       88. Yangon
  20. Colombo      43. Khandyga      66. Riyadh        89. Yekaterinburg
  21. Damascus     44. Kolkata       67. Sakhalin      90. Yerevan
  22. Dhaka        45. Krasnoyarsk   68. Samarkand
  23. Dili         46. Kuala_Lumpur  69. Seoul
Time zone: 70
Progress: [ 66%] [#########################################.....................]
```

设置超级用户：

```
apt install sudo
```

```
root@bc1d9b59bc0b:/#
root@bc1d9b59bc0b:/# apt install sudo
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  sudo
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 428 kB of archives.
After this operation, 1765 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 sudo amd64 1.8.
21p2-3ubuntu1.4 [428 kB]
Fetched 428 kB in 4s (105 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package sudo.
(Reading database ... 27354 files and directories currently installed.)
Preparing to unpack .../sudo_1.8.21p2-3ubuntu1.4_amd64.deb ...
Unpacking sudo (1.8.21p2-3ubuntu1.4) ...
Setting up sudo (1.8.21p2-3ubuntu1.4) ...
root@bc1d9b59bc0b:/#
```

```
visudo
```

添加自己的用户名，保存并退出：

```
topeet     ALL=(ALL:ALL) ALL
```

```
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/
sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d

topeet  ALL=(ALL:ALL) ALL
-- INSERT --                                    32,25-26      Bot
```

切换到普通用户，输入以下命令：

```
adduser topeet
su topeet
```

```
root@bc1d9b59bc0b:/#
root@bc1d9b59bc0b:/# adduser topeet
Adding user `topeet' ...
Adding new group `topeet' (1000) ...
Adding new user `topeet' (1000) with group `topeet' ...
Creating home directory `/home/topeet' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for topeet
Enter the new value, or press ENTER for the default
        Full Name []: topeet
        Room Number []: topeet
        Work Phone []: topeet
        Home Phone []: topeet
        Other []: topeet
Is the information correct? [Y/n] Y
root@bc1d9b59bc0b:/#
root@bc1d9b59bc0b:/# su topeet
topeet@bc1d9b59bc0b:/$
topeet@bc1d9b59bc0b:/$
```

1.5 下载文件到 docker

1 将网盘上的源码 SDK 下载下来，然后在 Windows 上解压为*.tar.gz 格式的压缩包，然后将此压缩包拷贝到虚拟机的 ubuntu 上。

2 根据《【北京迅为】itop-3588 开发板从零搭建 ubuntu 开发环境以及使用手册.doc》安装必要的依赖。

3 在 Ubuntu 20.04 新建一个目录，输入以下命令：

```
cd ~
mkdir project
cd project/
pwd
/home/topeet/project
```

```
topeet@bc1d9b59bc0b:/$
topeet@bc1d9b59bc0b:/$ cd ~
topeet@bc1d9b59bc0b:~$ mkdir project
topeet@bc1d9b59bc0b:~$ cd project/
topeet@bc1d9b59bc0b:~/project$ pwd
/home/topeet/project
topeet@bc1d9b59bc0b:~/project$
```

4 主机使用 docker 命令把需要使用的文件夹(比如源码)拷贝到 Docker 下的 Ubuntu 20.04，输入命令参考以下命令，自行根据实际情况进行修改。

```
docker cp /home/topeet/topeet-qt5.14.2/ bc1d9b59bc0b:/home/topeet/project/
```

```
root@ubuntu:/home/topeet#
root@ubuntu:/home/topeet# docker cp /home/topeet/topeet-qt5.14.2/ bc1d9b59bc0b:/
home/topeet/project/
root@ubuntu:/home/topeet#
```

在 Docker 下的 Ubuntu 20.04 中，文件被拷贝好，如下所示：

```
topeet@bc1d9b59bc0b:~/project$
topeet@bc1d9b59bc0b:~/project$ cd topeet-qt5.14.2/
topeet@bc1d9b59bc0b:~/project/topeet-qt5.14.2$ ls
aarch64.tar  demo  qt-opensource-linux-x64-5.14.2.run  topeet-qt5.14.2-aarch64
topeet@bc1d9b59bc0b:~/project/topeet-qt5.14.2$
```

5 我们可以使用 docker 命令将文件(比如源码)拷贝到 Docker 下的 Ubuntu 20.04 中，我们也可以使用 docker 命令将 Docker 下的 Ubuntu 20.04 中的文件(比如源码编译好的镜像)拷贝到虚拟机中。

```
Docker cp bc1d9b59bc0b:/home/topeet/project/* bc1d9b59bc0b:/home/topeet/project/
/home/topeet/
```

1.6 Docker 常用命令

查看容器

我们输入以下命令查看容器：

```
docker ps -a
```

```
root@ubuntu:/home/topeet# docker ps -a
CONTAINER ID   IMAGE             COMMAND       CREATED              STATUS                          PORTS     NAMES
b4b1fe2e96c0   ubuntu:20.04      "/bin/bash"   About a minute ago   Exited (0) About a minute ago             admiring_hermann
4d3d85984e86   ubuntu:20.04      "/bin/bash"   6 minutes ago        Exited (0) 2 minutes ago                  stoic_ganguly
e0fc21c64832   hello-world       "/hello"      8 minutes ago        Exited (0) 8 minutes ago                  serene_dubinsky
b58a27348e6a   hello-world       "/hello"      10 minutes ago       Exited (0) 10 minutes ago                 sad_feistel
4dfd76f4d776   ubuntu20_topeet   "/bin/bash"   6 weeks ago          Exited (0) 6 weeks ago                    hardcore_swartz
3a214cb3f92a   ubuntu20_topeet   "/bin/bash"   6 weeks ago          Exited (130) 6 weeks ago                  affectionate_swirles
9bd421bacf66   hello-world       "/hello"      6 weeks ago          Exited (0) 6 weeks ago                    unruffled_shaw
root@ubuntu:/home/topeet#
```

运行容器

执行以下命令运行容器：

```
docker start b4b1fe2e96c0
```

停止容器

如果容器的状态是“up”。表示它正在运行，可以执行以下命令停止此容器

```
docker stop b4b1fe2e96c0
```

```
root@ubuntu:/home/topeet#
root@ubuntu:/home/topeet# docker start b4b1fe2e96c0
b4b1fe2e96c0
root@ubuntu:/home/topeet#
root@ubuntu:/home/topeet#
root@ubuntu:/home/topeet# docker stop b4b1fe2e96c0
b4b1fe2e96c0
root@ubuntu:/home/topeet#
```

运行一个容器

停止一个容器

进入容器

输入以下命令进入容器，注意!如果从这个容器退出，容器不会停止。

```
docker exec -it b4b1fe2e96c0 /bin/bash
```

```
root@ubuntu:/home/topeet# docker start b4b1fe2e96c0
b4b1fe2e96c0
root@ubuntu:/home/topeet#
root@ubuntu:/home/topeet#
root@ubuntu:/home/topeet# docker exec -it b4b1fe2e96c0 /bin/bash
root@b4b1fe2e96c0:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@b4b1fe2e96c0:/#
root@b4b1fe2e96c0:/#
root@b4b1fe2e96c0:/#
```

退出容器

输入 exit 命令退出容器，如下图所示，使用 exec 命令从容器中退出，容器不会停止。

```
root@b4b1fe2e96c0:/#
root@b4b1fe2e96c0:/# exit
exit
root@ubuntu:/home/topeet# docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED         STATUS          PORTS                                       NAMES
b4b1fe2e96c0   ubuntu:20.04   "/bin/bash"   4 minutes ago   Up 36 seconds   0.0.0.0:8000->3000/tcp, :::8000->3000/tcp   admiring_hermann
root@ubuntu:/home/topeet#
root@ubuntu:/home/topeet#
```

导出容器

如果要导出本地某个容器，可以使用以下命令，导出容器 b4b1fe2e96c0 快照到本地文件 ubuntu.tar。

```
docker export b4b1fe2e96c0 > ubuntu.tar
```

```
root@ubuntu:/home/topeet# docker export b4b1fe2e96c0 > ubuntu.tar
root@ubuntu:/home/topeet# ls ubuntu.tar
ubuntu.tar
root@ubuntu:/home/topeet#
```

导入容器快照

可以使用 docker import 从容器快照文件中再导入为镜像，以下实例将快照文件 ubuntu.tar 导入到镜像 test/ubuntu:v1，输入以下命令：

```
cat docker/ubuntu.tar | docker import - test/ubuntu:v1
```

```
root@ubuntu:/home/topeet#
root@ubuntu:/home/topeet# cat ubuntu.tar | docker import - test/ubuntu:v1
sha256:276b348aa8c9e95a86055990edf04dcb9725840798f62ebe82a766108472f268
root@ubuntu:/home/topeet#
```

```
root@ubuntu:/home/topeet# docker images
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
test/ubuntu   v1        276b348aa8c9   2 minutes ago    7.97GB
ubuntu        18.04     71cb16d32be4   4 days ago       63.1MB
hello-world   latest    feb5d9fea6a5   12 months ago    13.3kB
root@ubuntu:/home/topeet#
```

删除容器

删除容器使用 docker rm 命令

```
docker rm -f b4b1fe2e96c0
```