

MilkV Duo —— Start Out

1. 准备一个linux环境

docker安装

在windows环境下，可以安装Docker Desktop for Windows，通过docker拉取ubuntu镜像并创建容器得到linux环境，[docker下载地址](#)

Install Docker Desktop on Windows

Welcome to Docker Desktop for Windows. This page contains information about Docker Desktop for Windows system requirements, download URL, instructions to install and update Docker Desktop for Windows.

Docker Desktop for Windows

For checksums, see [Release notes](#)

Docker Desktop terms

Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than \$10 million USD in annual revenue) requires a paid subscription.

System requirements

You must meet the following requirements to successfully install Docker Desktop on Windows:

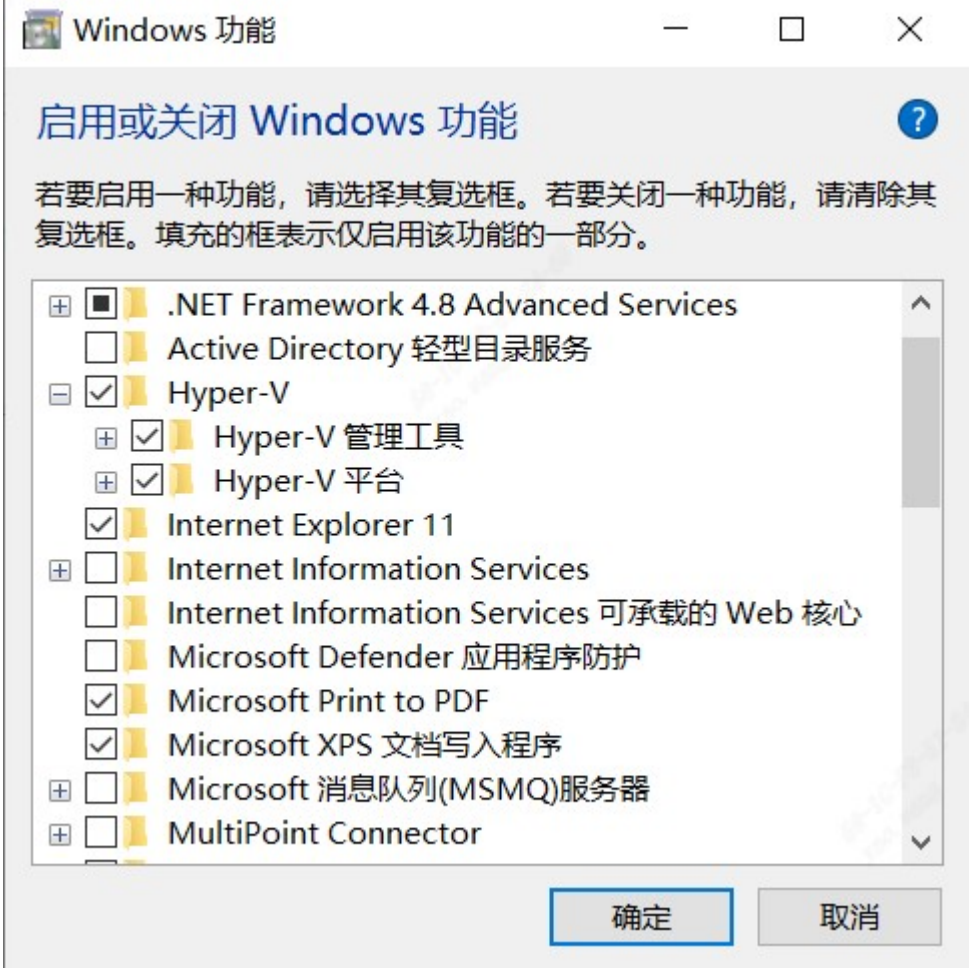
WSL 2 backend

Hyper-V backend and Windows containers

在Windows下运行docker需要相关依赖，即如图中所示，需要使用WSL2后端或者Hyper-V后端作为运行依赖
Hyper-V后端的启用方式如下：

- 1. 控制面板 —— 程序 —— 启用或关闭Windows功能

2. 找到Hyper-V，勾选Hyper-V管理工具和Hyper-V平台，等待系统文件配置完成后重启电脑



然后即可安装下载好Docker Desktop for Windows，在安装指引中根据选择的后端进行相应的勾选

安装完成后，需要重启电脑，然后即可使用docker

拉取ubuntu镜像

在命令行中输入`docker pull ubuntu:20.04`，即可拉取ubuntu20.04的镜像，然后可以使用`docker images`查看拉取到的镜像

接着，在命令行中输入`docker run -it ubuntu:20.04 /bin/bash`，即可创建并运行一个容器，当终端中出现`root@<string>:/#`时，则表示容器创建完成且已经进入容器，其中，string是容器的id，即containerId

若需要退出在容器内退出容器，输入`exit`即可，若需要在容器外再次进入容器，则在终端中依次输入`docker start <containerId>`，`docker attach <containerId>`即可，其中<containerId>为需要进入的容器对应的containerId

若需要查看containerId，在终端中输入`docker ps -a`会列出所有容器的信息，输入`docker ps -n num`会列出最近的num个容器的信息

若需要对容器重命名(改变NAMES)，在终端中输入`docker rename <old_name> <new_name>`即可

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
092c6b16284e	ubuntu:20.04	"/bin/bash"	3 hours ago	Exited (0)	2 hours ago	ubuntu20.04_duo
7df1ee2b0cf4	ubuntu:20.04	"/bin/bash"	8 hours ago	Exited (0)	3 hours ago	ubuntu20.04_milk_v
e859f0cfd07c	hello-world	"/hello"	8 hours ago	Exited (0)	8 hours ago	funny_darwin

更多指令可参考docker官方教程

2. 配置duo-buildroot-sdk

准备编译环境

启动创建好的ubuntu20.04容器，首先需要安装sudo

```
apt-get update
apt-get install sudo
```

安装编译依赖工具

```
sudo apt install pkg-config build-essential ninja-build automake autoconf libtool
wget curl git gcc libssl-dev bc slib squashfs-tools android-sdk-libsparse-utils
android-sdk-ext4-utils jq python3-distutils tclsh scons parallel ssh-client tree
python3-dev python3-pip device-tree-compiler ssh cpio fakeroot libncurses5 flex
bison libncurses5-dev genext2fs rsync unzip dosfstools mtools
```

安装过程中，需要选择输入时区以进行下一步，依照提示输入即可

安装cmake

```
wget https://github.com/Kitware/CMake/releases/download/v3.26.4/cmake-3.26.4-
linux-x86_64.sh
chmod +x cmake-3.26.4-linux-x86_64.sh
sudo ./cmake-3.26.4-linux-x86_64.sh --skip-license --prefix=/usr/local/
```

然后输入`cmake --version`进行验证，若能输出对应版本号，则表示安装成功

获取SDK

```
git clone https://github.com/milkv-duo/duo-buildroot-sdk.git
```

进行编译

执行一键编译脚本`build_milkv.sh`

```
cd duo-buildroot-sdk/  
./build_milkv.sh
```

编译成功后可以在/duo-buildroot-sdk/out目录下找到milkv-duo-xxxxxxx-xxxx.img文件，即为后续要用到的镜像文件

至此，基本开发环境搭建完毕。