

从 VMware 安装 Ubuntu 22.04 到 RM 视觉项目环境配置培训手册

版本：V1.0

编写日期：2025-08-08

编写人：张俊杰

适用对象：26 赛季视觉组新人

目录

1. 培训目标.....	3
2. 系统与软件要求.....	3
3. VMware 虚拟机安装与配置	3
4. Ubuntu 系统安装.....	4
5. 系统基础配置	5
6. 视觉项目开发环境搭建	5
7. 项目代码拉取与运行测试	19

1. 培训目标

- 熟悉 VMware 虚拟机安装与 Ubuntu 系统配置流程
- 掌握视觉项目所需环境的安装步骤
- 能够独立完成代码拉取与运行测试


2. 系统与软件要求

项目	版本/要求
VMware	17.x 及以上
Ubuntu	22.04 LTS
Python	3.8+
CUDA（可选）	11.8 / 12.x

3. VMware 虚拟机安装与配置

1. 下载 VMware Workstation 并安装（Windows 平台）

- 官方地址：<https://www.vmware.com>（下面所有的安装包均会提供）

 VMware17.6.exe	2024/9/3 17:39	应用程序	251,368 KB
--	----------------	------	------------

这里只需要管理员运行，然后跟着引导安装即可

2. 创建 Ubuntu 虚拟机

- 分配 CPU/内存（推荐 2 核/8GB）
- 网络建议 NAT 或桥接

3. 网络配置注意事项：公司内网桥接模式可能需要申请 IP


```
Activities Terminal 8月8 15:36
julyjelly@julyjelly: ~
Length: 51518 (50K) [application/octet-stream]
Saving to: '/tmp/fishinstall/tools/base.py'
/tmp/fishinstall/t 100%[=====] 50.30K --KB/s in 0.04s
2025-08-08 15:30:09 (1.13 MB/s) - '/tmp/fishinstall/tools/base.py' saved [51518/51518]

Run CMD Task [dmg --print-architecture]
[-][0.00s] CMD Result:success

Run CMD Task [wget http://mirror.fishros.com/install/tools/translation/translator.py -O /tmp/fishinstall/tools/translation/translator.py --no-check-certificate]
[-][0.00s] --2025-08-08 15:30:09-- http://mirror.fishros.com/install/tools/translation/translator.py [200] Connecting to mirror.fishros.com (mirror.fishros.com)[47.119.165.109] [-][0.00s] CMD Result:success

Run CMD Task [wget http://mirror.fishros.com/install/tools/translation/assets/zh_CN.py -O /tmp/fishinstall/tools/translation/assets/zh_CN.py --no-check-certificate]
[-][0.00s] --2025-08-08 15:30:11-- http://mirror.fishros.com/install/tools/translation/assets/zh_CN.py [200] Connecting to mirror.fishros.com (mirror.fishros.com)[47.119.165.109] [-][0.00s] CMD Result:success

Run CMD Task [wget http://mirror.fishros.com/install/tools/translation/assets/en_US.py -O /tmp/fishinstall/tools/translation/assets/en_US.py --no-check-certificate]
[-][0.00s] --2025-08-08 15:30:11-- http://mirror.fishros.com/install/tools/translation/assets/en_US.py [200] Connecting to mirror.fishros.com (mirror.fishros.com)[47.119.165.109] [-][0.00s] CMD Result:success

Run CMD Task [wget https://fishros.org.cn/forum/topic/1733 -O /tmp/t1733.q --timeout 10 && rm -rf /tmp/t1733]
[-][0.00s] CMD Result:success

以为您切快速查看当前所在国家语言:zh_CN
系统语言通过:
=====
#####欢迎使用一键安装工具, 人生苦短, 三省吾身, 省时省力省心#####
#####一键安装已开源, 请放心使用: https://github.com/fishros/install #####
=====

// 开单有益 // 开山有路
// 可以多看看小鱼的文档 // 关注a站看mcs机器人

Run Choose Task: 请输入括号内的数字
---公共工具, 等待安装---
ros相关:
(1) 一键安装(推荐): ros支持ros/ros2, 树莓派jetson
(2) 一键安装(rosdep): 小鱼rosdep, 支持交叉编译
(3) 一键安装(rosdep): 小鱼rosdep, 支持交叉编译
(4) 一键安装: ros环境(快速安装ros环境设置, 自动生成环境选择)
(5) 一键安装: Cartographer(18.20测试通过, 16未测, updateTime 20240125)
(6) 一键安装: ros Docker版(支持所有版本ros/ros2)
(7) 一键安装: 系统自带ros ( ! 警告 ! 仅供特殊情况下载使用)

常用软件:
(8) 一键安装: github桌面版(小鱼常用的github客户端)
(9) 一键安装: nodejs环境
(10) 一键安装: vscode开发工具
(11) 一键安装: docker
(12) 一键安装: 微信(可以在Linux上使用的微信)
(13) 一键安装: PlatformIO MicroOS开发环境(支持fishbot)
(14) 一键安装: QQ for Linux

配置工具:
(15) 一键配置: 系统源(更换系统源, 支持全版本ubuntu系统)
(16) 一键配置: python国内源
(17) 一键配置: Docker代理(支持vms+代理服务器两种模式)

[0] quit
请输入[ ]内的数字以选择
```

输入数字 1，然后回车

```
Activities Terminal 8月8 15:38
julyjelly@julyjelly: ~
以为您切快速查看当前所在国家语言:zh_CN
系统语言通过:
=====
#####欢迎使用一键安装工具, 人生苦短, 三省吾身, 省时省力省心#####
#####一键安装已开源, 请放心使用: https://github.com/fishros/install #####
=====

// 开单有益 // 开山有路
// 可以多看看小鱼的文档 // 关注a站看mcs机器人

Run Choose Task: 请输入括号内的数字
---公共工具, 等待安装---
ros相关:
(1) 一键安装(推荐): ros支持ros/ros2, 树莓派jetson
(2) 一键安装(rosdep): 小鱼rosdep, 支持交叉编译
(3) 一键安装(rosdep): 小鱼rosdep, 支持交叉编译
(4) 一键安装: ros环境(快速安装ros环境设置, 自动生成环境选择)
(5) 一键安装: Cartographer(18.20测试通过, 16未测, updateTime 20240125)
(6) 一键安装: ros Docker版(支持所有版本ros/ros2)
(7) 一键安装: 系统自带ros ( ! 警告 ! 仅供特殊情况下载使用)

常用软件:
(8) 一键安装: github桌面版(小鱼常用的github客户端)
(9) 一键安装: nodejs环境
(10) 一键安装: vscode开发工具
(11) 一键安装: docker
(12) 一键安装: 微信(可以在Linux上使用的微信)
(13) 一键安装: PlatformIO MicroOS开发环境(支持fishbot)
(14) 一键安装: QQ for Linux

配置工具:
(15) 一键配置: 系统源(更换系统源, 支持全版本ubuntu系统)
(16) 一键配置: python国内源
(17) 一键配置: Docker代理(支持vms+代理服务器两种模式)

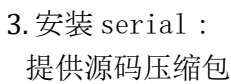
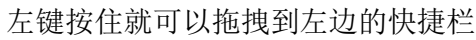
[0] quit
请输入[ ]内的数字以选择:
Run CMD Task [wget http://mirror.fishros.com/install/tools/tool_install_ros.py -O /tmp/fishinstall/tools/tool_install_ros.py --no-check-certificate]
[-][0.07s] CMD Result:success 100%|#####| connected.

Run CMD Task [wget http://mirror.fishros.com/install/tools/tool_config_rosenv.py -O /tmp/fishinstall/tools/tool_config_rosenv.py --no-check-certificate]
[-][0.00s] CMD Result:success 100%|#####| connected.

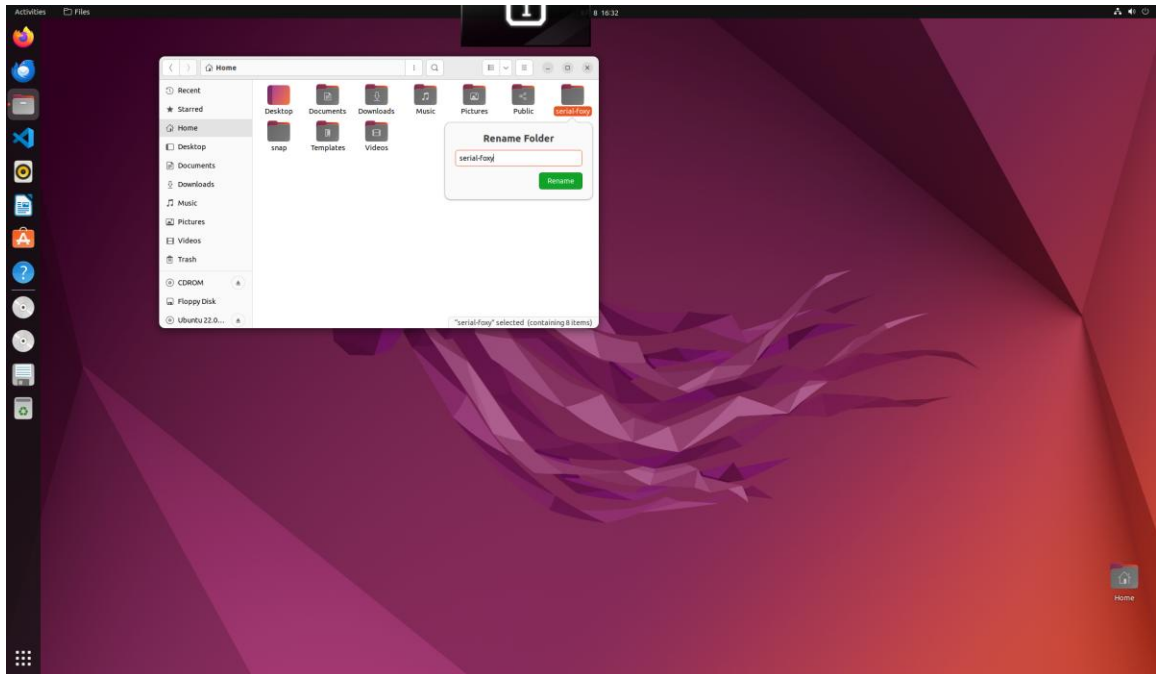
Run CMD Task [wget http://mirror.fishros.com/install/tools/tool_config_system_source.py -O /tmp/fishinstall/tools/tool_config_system_source.py --no-check-certificate]
[-][0.00s] CMD Result:success 100%|#####| connected.

欢迎使用一键安装ros和ros2, 支持树莓派jetson, 本工具由作者小鱼提供
欢迎使用ros开箱工具, 本工具由( 角哥mcs )小鱼贡献。
小鱼和角哥提供快速安装ros环境, 如果遇到问题请快速选择:
Run Choose Task: 请输入括号内的数字
新手建议安装时一定要一定要换源并清理三方源, 换源!!!系统默认国外源容易失败!!
(1) 更换系统源并清理三方源
(2) 不更换系统源
[0] quit
请输入[ ]内的数字以选择:
Run CMD Task [python ros_update]
[-][1.41s] m113 http://mirror.fishros.com/install/tools/tool_config_system_source.py [200] Connecting to mirror.fishros.com (mirror.fishros.com)[47.119.165.109] [-][0.00s] CMD Result:success
```

这里我选择不换源，如果感觉速度很慢可以选择换源，静等片刻，然后选择 1 安装 ros-humble



将其解压到 home 目录下，重命名为 serial



双击进去里面有个 **readme.pdf** 像这样，安装他的引导，打开终端输入下列命令即可

Get the code:

```
# open a new terminal
cd serial
mkdir build
```

Build:

```
cd build
cmake ..
make
```

Install:

```
sudo make install
```

这里就是编译这里的源码，然后把库载入系统

A screenshot of a Linux terminal window displaying the installation of the 'serial' module. The user starts by cloning a repository and running 'make build'. This triggers a series of checks for compilers (C, CXX) and features. After successful checks, it generates build files. Then, the user runs 'make install', which prompts for a password. Finally, they run 'sudo make install' to complete the installation. The output shows various system paths being updated and configuration files being installed.

```
july@julykali:~/serial$ git clone https://github.com/mroldano/pySerial.git
Cloning into 'pySerial':
remote: Enumerating objects: 17, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 17 (delta 17), reused 0 (delta 0), compressed 0 bytes.
Unpacking objects: 100% (17/17), done.
july@julykali:~/serial$ cd build
july@julykali:~/serial/build$ make
-- The C compiler identification is GNU 11.4.0
-- The CXX compiler identification is GNU 11.4.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Found Boost: 1.81.2 (/opt/ros/noetic/share/ament_cmake/boost)
-- Found Python3: /usr/bin/python3 (found version "3.10.12") found components: Interpreter
Configuring done
Generating done
Build files have been written to: /home/july/serial/build
july@julykali:~/serial/build$ make
[ 23%] Building CXX object cmakefiles/serial_dir/src/serial.cc.o
[ 50%] Linking CXX shared library libserial.so
[ 75%] Building CXX object cmakefiles/serial_dir/src/npy/tst_ports/tst_ports_linux.cc.o
[100%] Linking CXX shared library libserial.so
[100%] Built target serial
july@julykali:~/serial/build$ sudo make install
[sudo] password for july:
Installing cmake generated dependencies of target serial
[100%] Built target serial
Install the project...
-- Install configuration: ""
-- Installing: /usr/local/share/serial/environment/library_path.sh
-- Installing: /usr/local/share/serial/environment/library_path.dv
-- Installing: /usr/local/lib/libserial.so
-- Installing: /usr/local/include/serial/serial.h
-- Installing: /usr/local/include/serial/vsdist.h
-- Installing: /usr/local/share/ament_index/resource_index/package_run_dependencies/serial
-- Installing: /usr/local/share/ament_index/resource_index/parent_prefix_path/serial
-- Installing: /usr/local/share/serial/environment/ament_prefix_path.sh
-- Installing: /usr/local/share/serial/environment/ament_prefix_path.dv
-- Installing: /usr/local/share/serial/environment/path.sh
-- Installing: /usr/local/share/serial/environment/path.dv
-- Installing: /usr/local/share/serial/local_setup.bash
-- Installing: /usr/local/share/serial/local_setup.sh
-- Installing: /usr/local/share/serial/local_setup.zsh
-- Installing: /usr/local/share/serial/local_setup.dv
-- Installing: /usr/local/share/serial/package.dv
-- Installing: /usr/local/share/ament_index/resource_index/package_scripts
-- Installing: /usr/local/share/serial/cmake/ament_cmake_export_include_directories-extras.cmake
-- Installing: /usr/local/share/serial/cmake/ament_cmake_export_libraries-extras.cmake
-- Installing: /usr/local/share/serial/cmake/serialconfig.cmake
-- Installing: /usr/local/share/serial/cmake/serialconfig-version.cmake
-- Installing: /usr/local/share/serial/package.xml
july@julykali:~/serial/build$
```

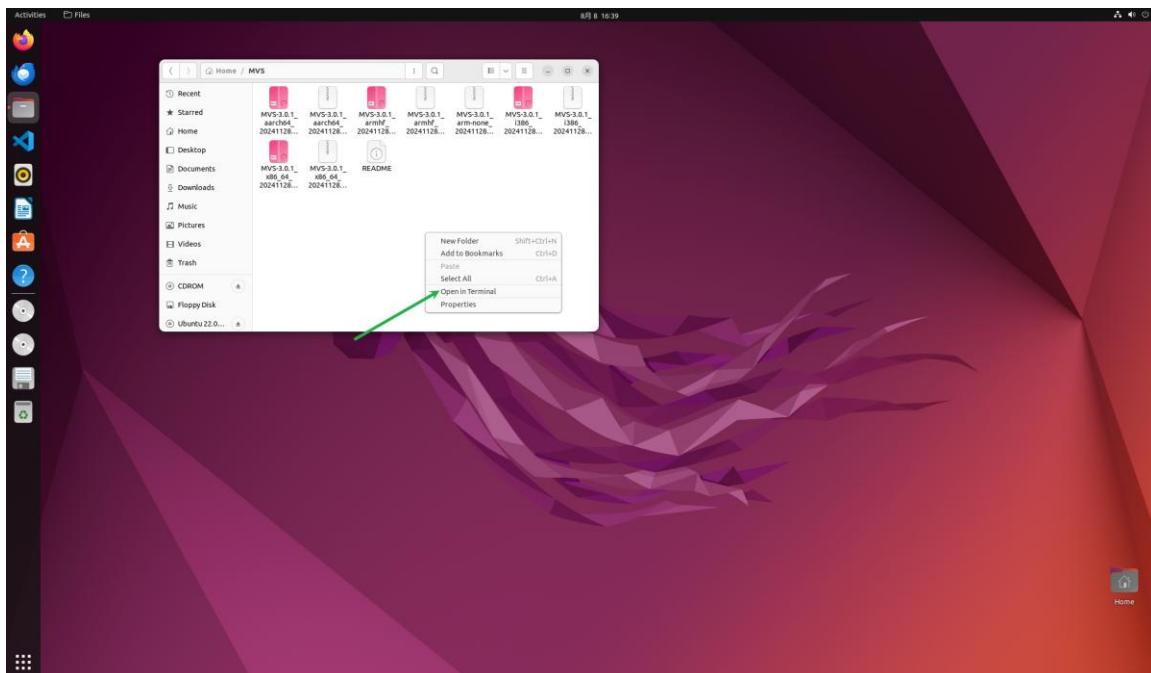
4. 安装 海康工业相机驱动 MVS :

提供驱动压缩包, 同样解压到 home 目录

 MVS.zip	2025/8/8 16:24	压缩(zipped)文件夹	545,160 KB
---	----------------	---------------	------------

解压后双击打开，在 **MVS** 目录下右键在终端打开输入如下命令：

```
sudo dpkg -i MVS-3.0.1_x86_64_20241128.deb
```





The screenshot shows a Kali Linux desktop environment. The top panel includes the 'Activities' button, a search bar, and the system clock displaying '8/8 16:41'. The terminal window, titled 'julyjolly@julyjolly: ~mvs', shows the following commands and output:

```
julyjolly@julyjolly:~$ sudo dpkg -i MVS-3.0.1_
MVS-3.0.1_3arch64_20241128.deb  MVS-3.0.1_1i386_20241128.deb
MVS-3.0.1_ernbf_20241128.deb  MVS-3.0.1_x86_64_20241128.deb
julyjolly@julyjolly:~$ sudo dpkg -i MVS-3.0.1_x86_64_20241128.deb
[sudo] password for julyjolly:
Selecting previously unselected package mvs.
Preparing to unpack MVS-3.0.1_x86_64_20241128.deb ...
Unpacking mvs (2022-18-24) ...
Setting up mvs (2022-18-24) ...
Install mvs, please wait...
cp: cannot stat '/opt/mvs/bin/fonts/*': No such file or directory
Set up the SDK environment...
Adding rules for vendor ID 2bdf.
The /etc/udev/rules.d/80-drivers-SDK-2bdf.rules rule has been created.
Adding rules for virtual serial device.
The /etc/udev/rules.d/80-drivers-SDK-virtualserial.rules rule has been created.
create l1nk to dynamic library
Starting execute script...
Setting env's memory size to 2000
Setting socket maximum buffer size to 10485760
Configuration of the rg_filter.
For more information, read the README knowledge note:
https://access.redhat.com/knowledge/p/shortitem/53093
Supported modes:
0 - No source validation (recommended).
1 - RFC3704 Strict Reverse Path.
2 - RFC3704 Loose Reverse Path.
Setting the mode to No source validation.
The network stack will be restarted.
Install MVS complete!
Tips: You should be launch a new terminal or execute source command for the bash environment!
julyjolly@julyjolly:~$
```

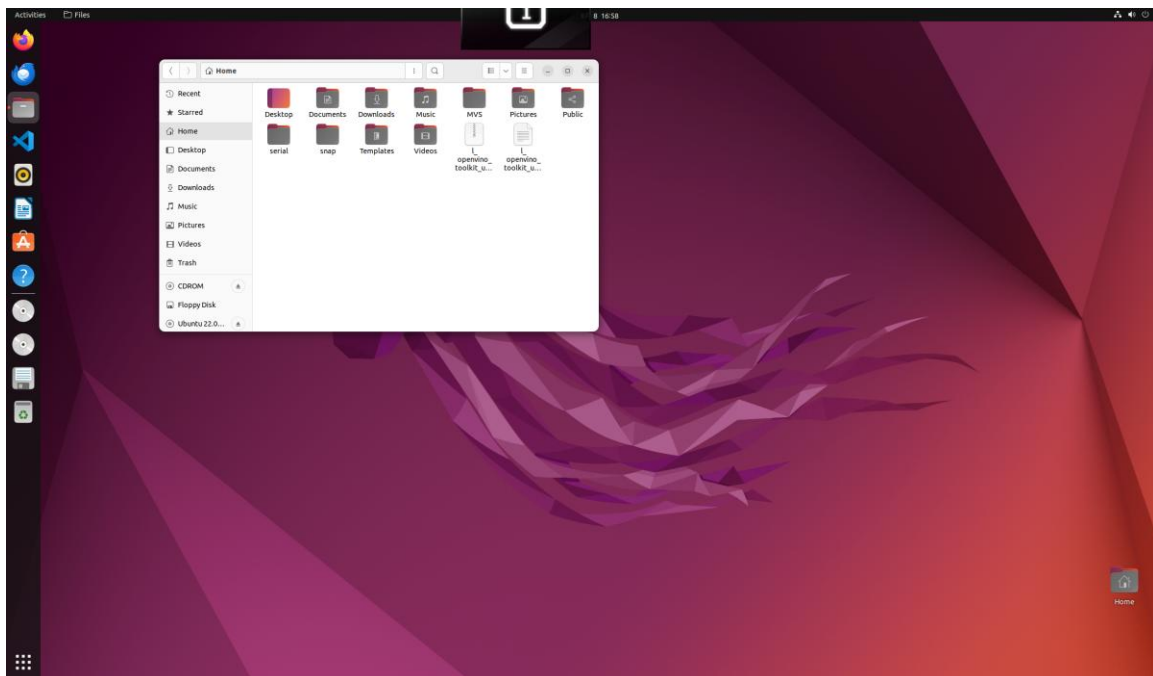
[illegible]

6. 安装 cpu 加速 openvino2023 :

提供关键 openvino2023 库压缩包及校验包

 I_openvino_toolkit_ubuntu22_2023.3.0.137...	2025/4/6 11:52	压缩存档文件夹	52,420 KB
 I_openvino_toolkit_ubuntu22_2023.3.0.137...	2025/4/6 11:51	SHA256 文件	1 KB

拷贝到 home 目录下，打开终端

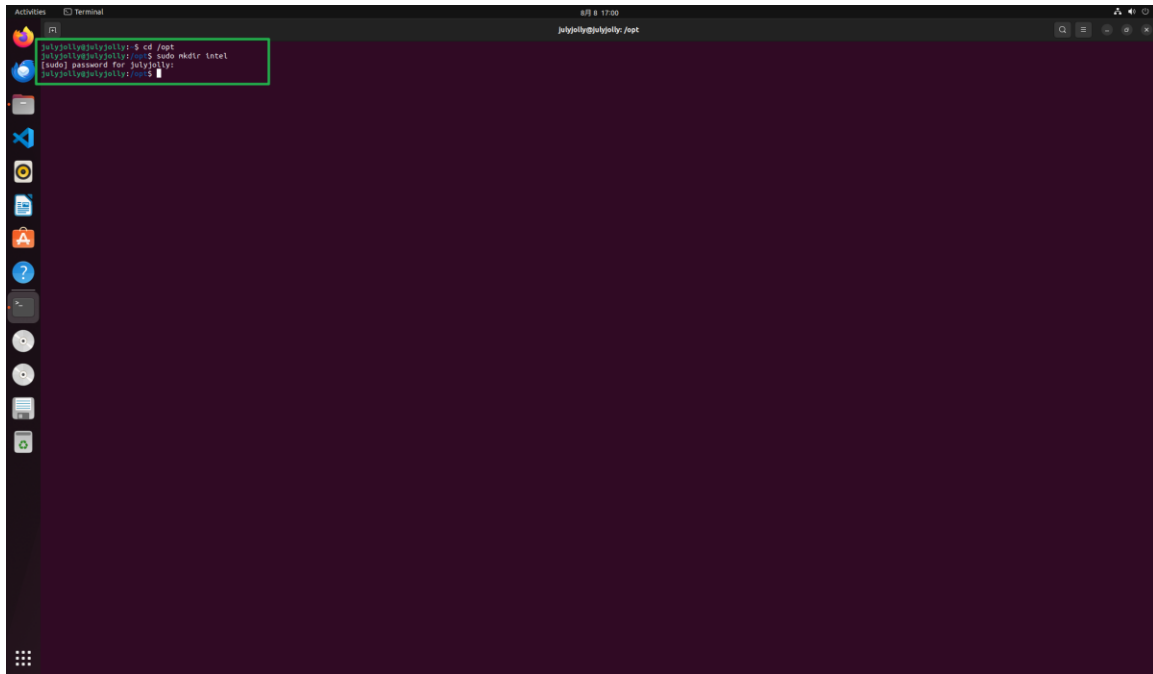


输入

```
cd /opt
```

```
sudo mkdir intel
```

创建存放 openvino 库的文件夹



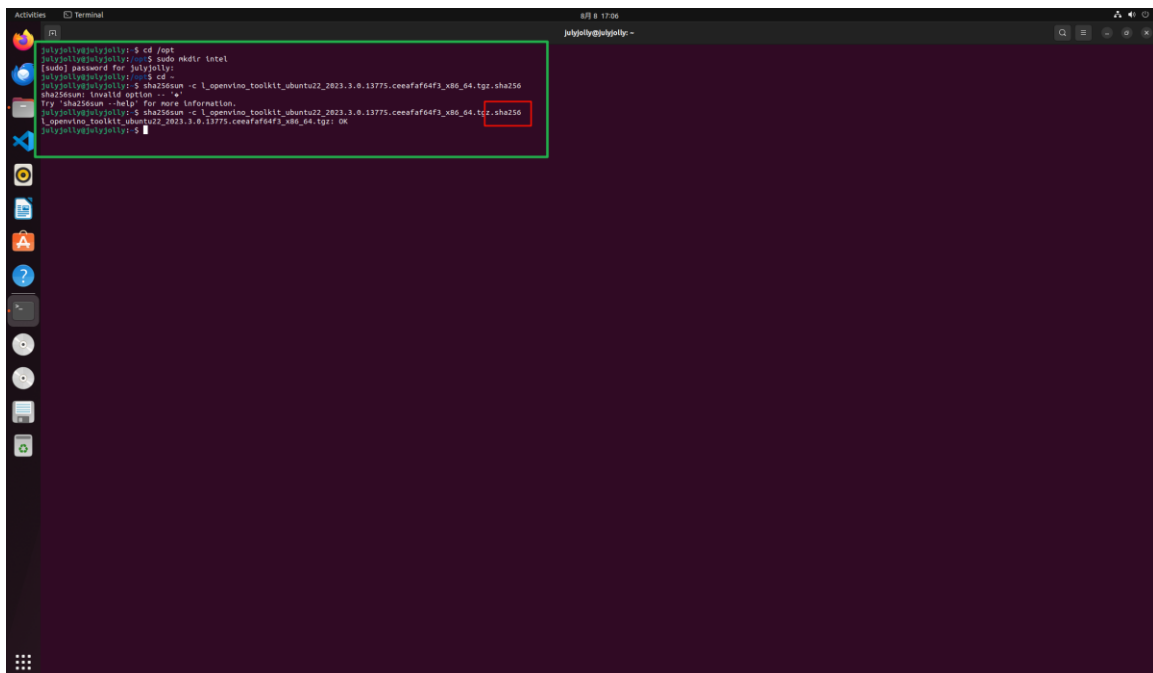
```
julyjuly@julyjuly:~$ cd /opt
julyjuly@julyjuly:~/opt$ sudo mkdir intel
[sudo] password for julyjuly:
julyjuly@julyjuly:~/opt$
```

输入: `cd ~` 返回 home 目录

输入: `sha256sum -c`

`l_openvino_toolkit_ubuntu22_2023.3.0.13775.cceafaf64f3_x86_64.tgz.sha256`

这个命令用来校验库，第一次复制进去可能会打印报错日志，第二遍手敲 `sha256sum -c l_op` 在 `tab` 补齐即可，补齐别忘了 `.sha256`



```
julyjuly@julyjuly:~$ cd /opt
julyjuly@julyjuly:~/opt$ sudo mkdir intel
[sudo] password for julyjuly:
julyjuly@julyjuly:~/opt$ cd ~
julyjuly@julyjuly:~$ sha256sum -c l_openvino_toolkit_ubuntu22_2023.3.0.13775.cceafaf64f3_x86_64.tgz.sha256
sha256sum: l_openvino_toolkit_ubuntu22_2023.3.0.13775.cceafaf64f3_x86_64.tgz: OK
julyjuly@julyjuly:~$
```

输入

```
sudo tar xf l_openvino_toolkit_ubuntu22_2023.3.0.13775.cceafaf64f3_x86_64.tgz -C /opt/intel
```

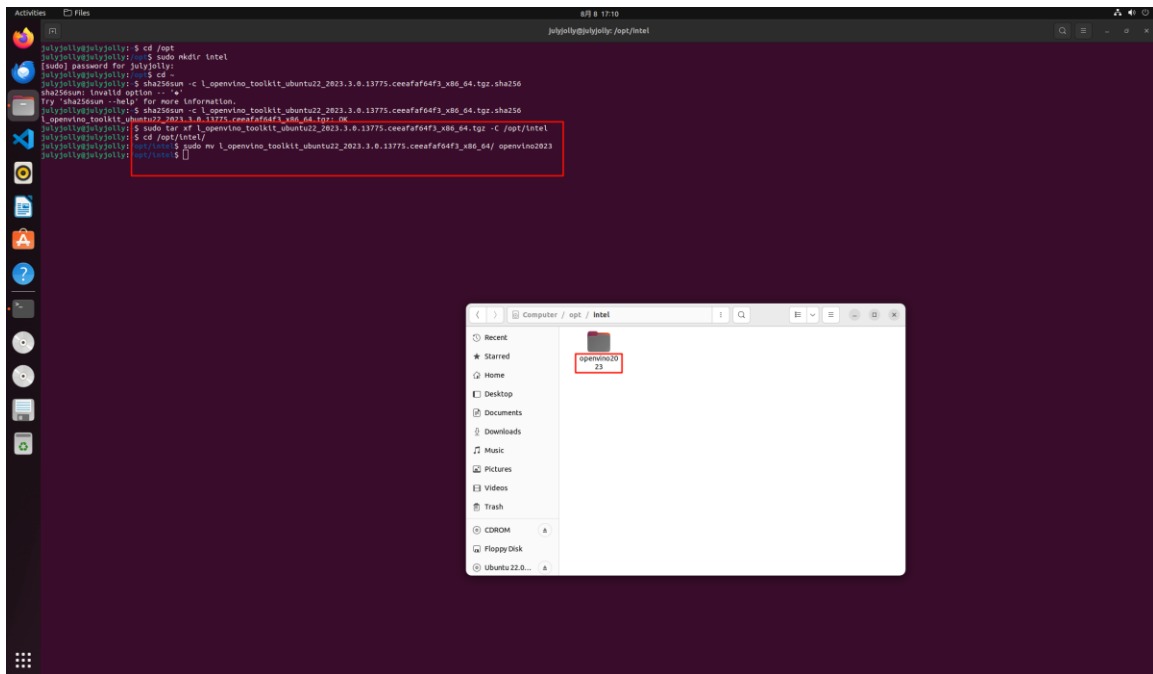
将 openvino 库解压到 intel 文件夹

然后，输入

```
cd /opt/intel/
```

```
sudo mv l_openvino_toolkit_ubuntu22_2023.3.0.13775.cceafaf64f3_x86_64/ openvino2023
```

将 intel 的 openvino 改名为 openvino2023



输入 `cd ~` 回到 home 目录，接着依次输入下面命令行安装 openvino APT

```
wget https://apt.repos.intel.com/intel-gpg-keys/GPG-PUB-KEY-INTEL-SW-PRODUCTS.PUB
```

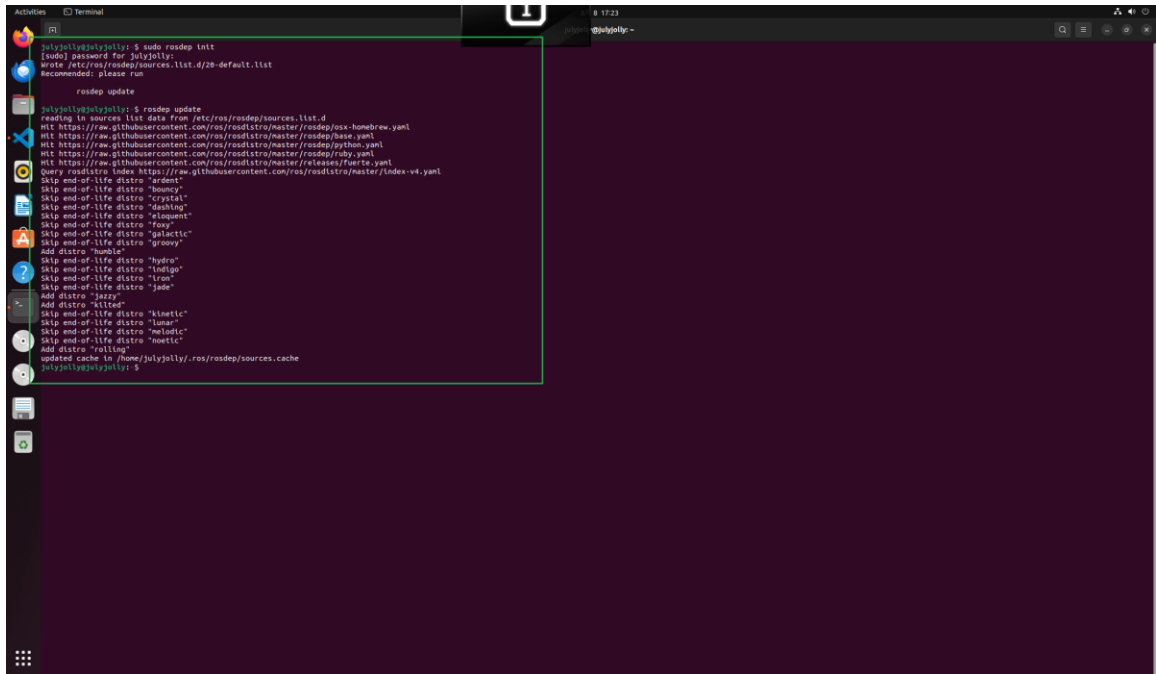
```
sudo apt-key add GPG-PUB-KEY-INTEL-SW-PRODUCTS.PUB
```

```
echo "deb https://apt.repos.intel.com/opencvino/2023 ubuntu22 main" | sudo tee /etc/apt/sources.list.d/intel-openvino-2023.list
```

```
sudo apt update
```

```
apt-cache search openvino
```

```
sudo apt install openvino-2023.1.0
```

```
julyjolly@julyjolly:~$ sudo rosdep init
[sudo] password for julyjolly:
wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run

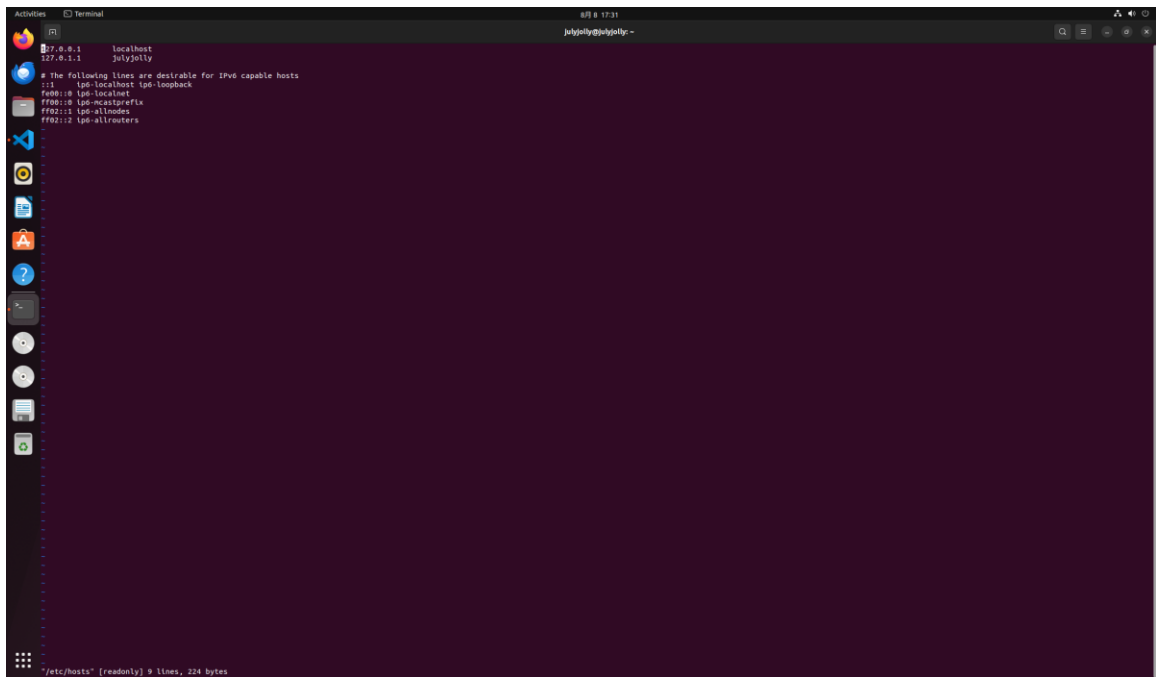
rosdep update

julyjolly@julyjolly:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/oxz-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/patches.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuente.yaml
Query rosdistro index https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "argent"
Skip end-of-life distro "boucny"
Skip end-of-life distro "cristal"
Skip end-of-life distro "dashing"
Skip end-of-life distro "eloquent"
Skip end-of-life distro "fox"
Skip end-of-life distro "galactic"
Skip end-of-life distro "groovy"
Add distro "humble"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "lunar"
Skip end-of-life distro "jade"
Add distro "jazzy"
Add distro "kinetic"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Skip end-of-life distro "melodic"
Skip end-of-life distro "noetic"
Add distro "rolling"
update cache in /home/julyjolly/.ros/rosdep/sources.cache
julyjolly@julyjolly:~$
```

我这里运气好一次就通过了，如果没通过也别着急在 `/etc/hosts` 文件中添加相应的 ip 即可，

输入 `vi /etc/hosts`

进去之后就是这样的



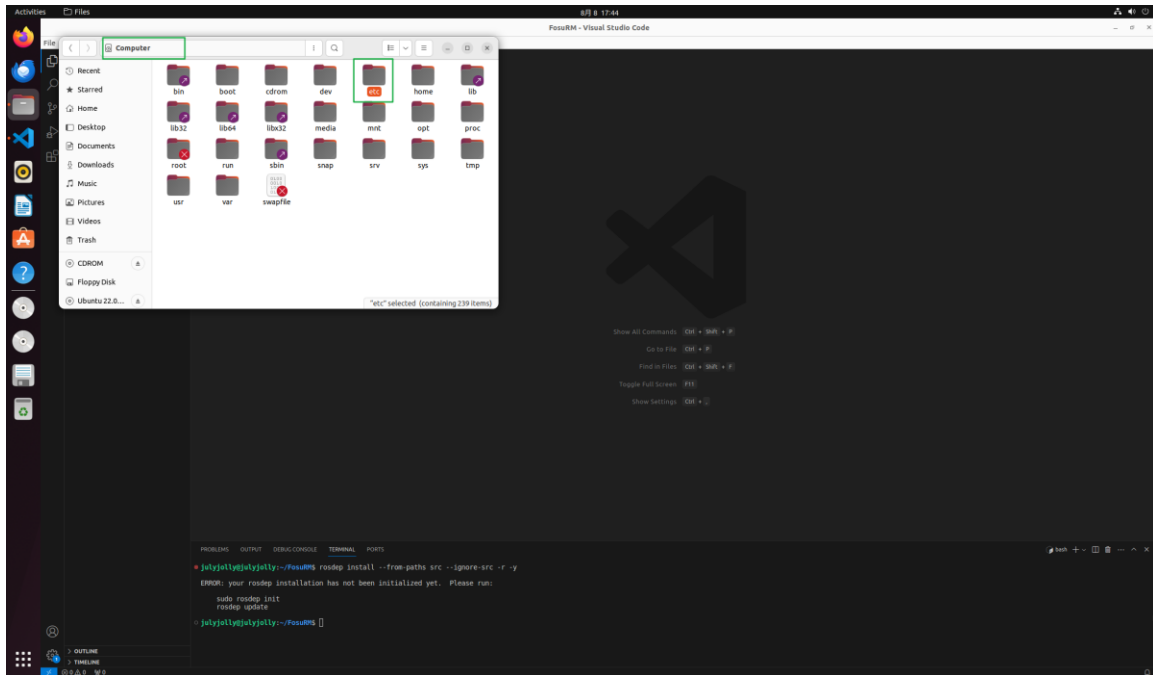
```
127.0.0.1 localhost
127.0.1.1 julyjolly

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe80:: ip6-localhost
ff02:: ip6-mcastprefix
ff02:: ip6-allnodes
ff02:: ip6-allrouters

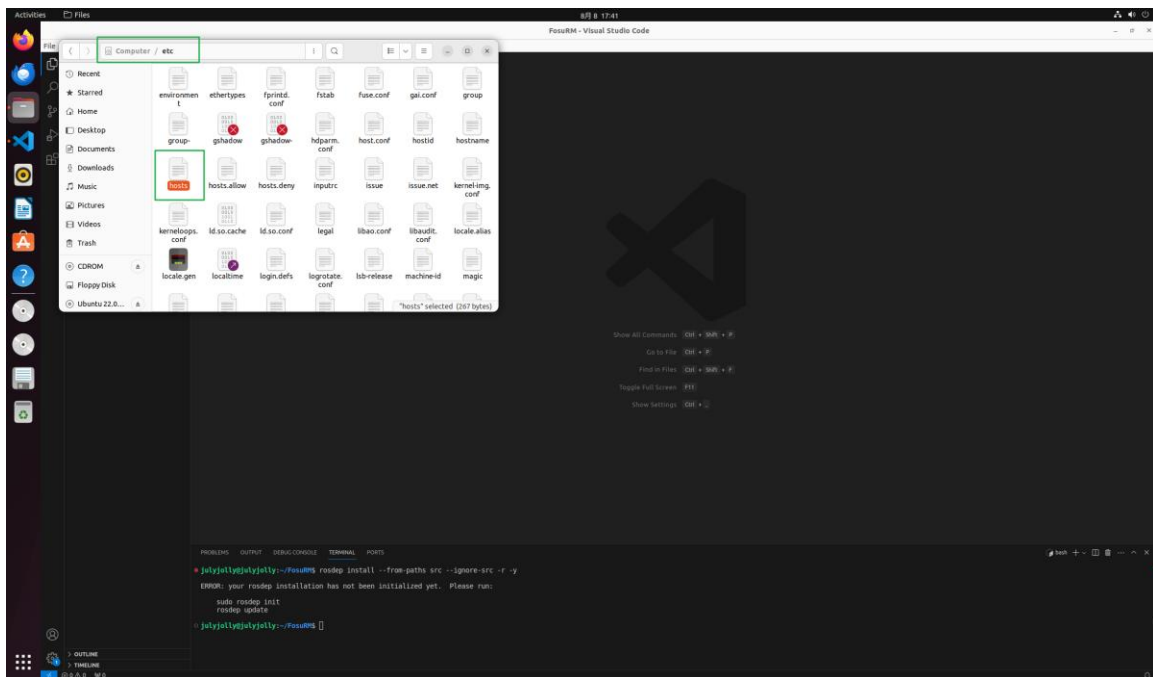
/etc/hosts [readonly] 9 lines, 224 bytes
```

然后在最后一行输入 [185.199.111.133 raw.githubusercontent.com](https://raw.githubusercontent.com) 就可以了

vi 会很难使用，也可以在文件夹中找到/etc 这个目录，如何找到/etc，只需在 home 目录下，按住你删除文字的那个按键，按两下就会跳转到



双击打开 hosts 再在最后一行添加 185.199.111.133 raw.githubusercontent.com 然后保存就好，保存的时候应该要你输入用户的密码。



8.一些疏漏的 ros-humble 包：

只要在终端输入

```
sudo apt install ros-humble-joint-state-publisher
```

```
sudo apt install ros-humble-asio-cmake-module
```

7. 项目代码拉取与运行测试

按照战队代码 readme 编译运行，测试环境配置是否完全

能量机关识别效果

```
./build/armor_detector/test_buff_detector
```

