2023秋《操作系统》课程实验报告

实验0

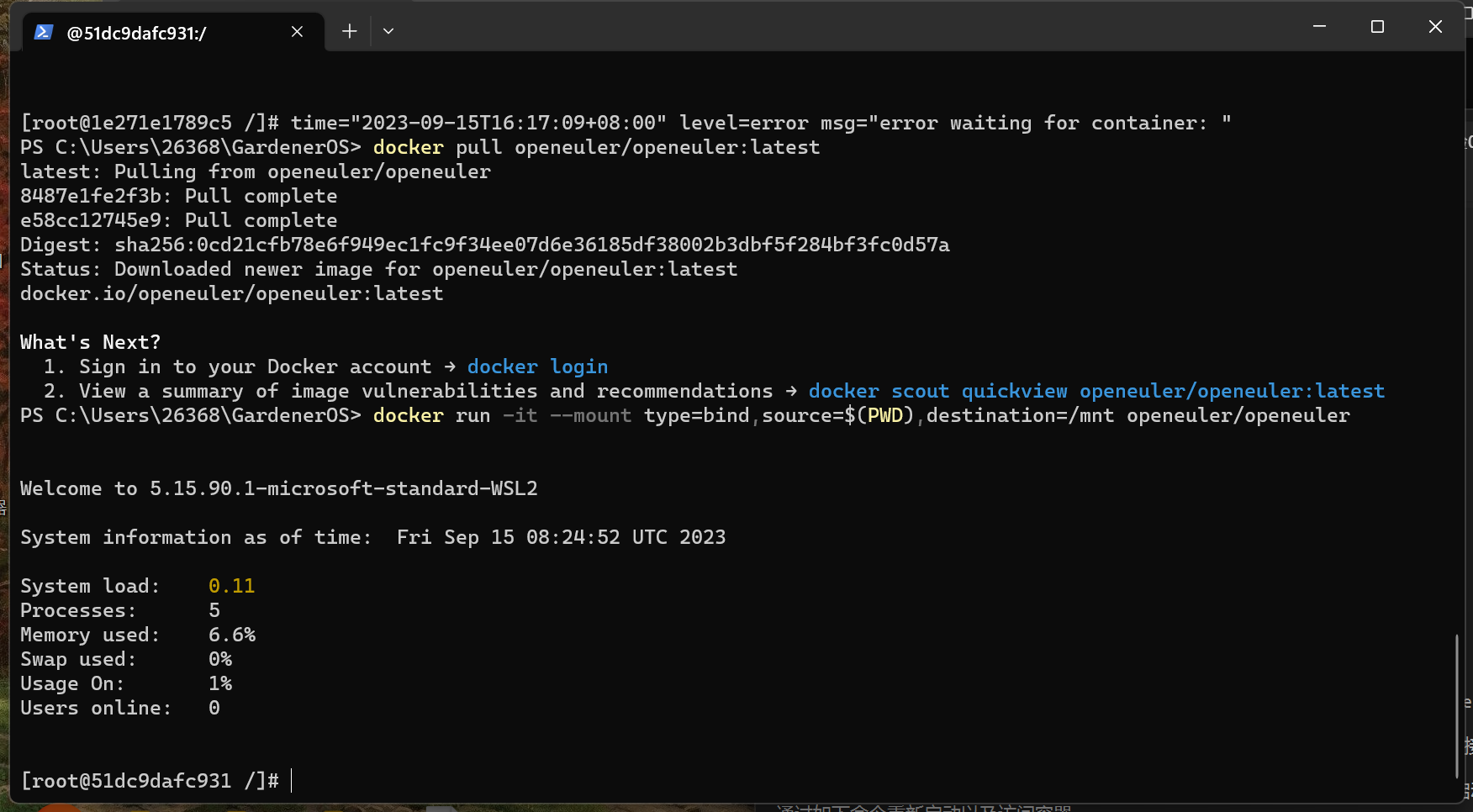
21301039 黄钰淞

1. 实验步骤
2. 准备实验基础平台

1.1首先进行docker desktop等下载，之前下载过了不再赘述

1.2拉取最新镜像

docker pull openeuler/openeuler:latest



可以看到拉取成功

1. 创建开发目录

Git相关操作参考https://blog.csdn.net/Thinkingcao/article/details/106059446

当前步骤最终截图如下



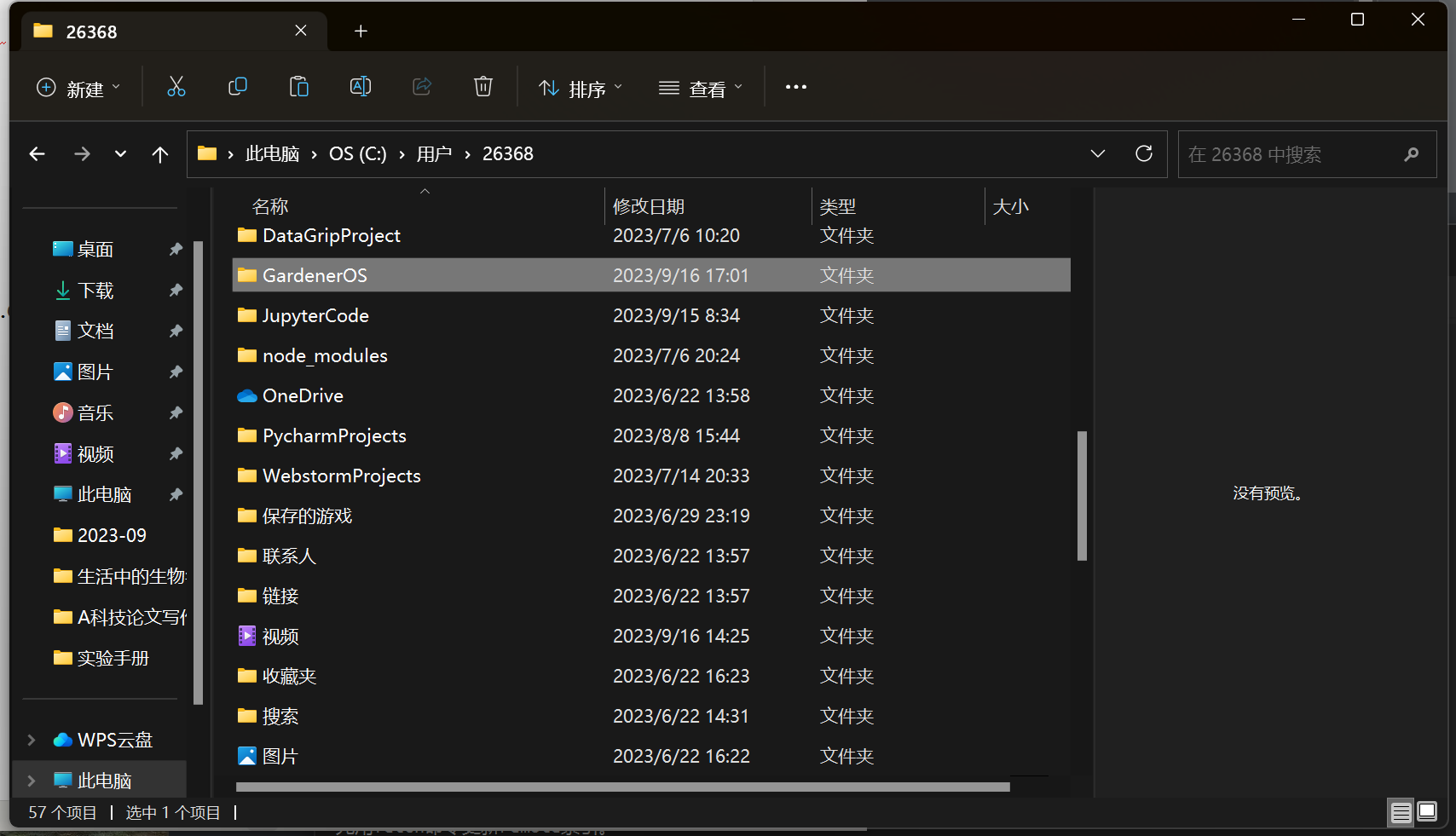
详细步骤如下

2.1创建本地开发目录

首先在用户目录下创建GardenerOS目录作为本地开发目录

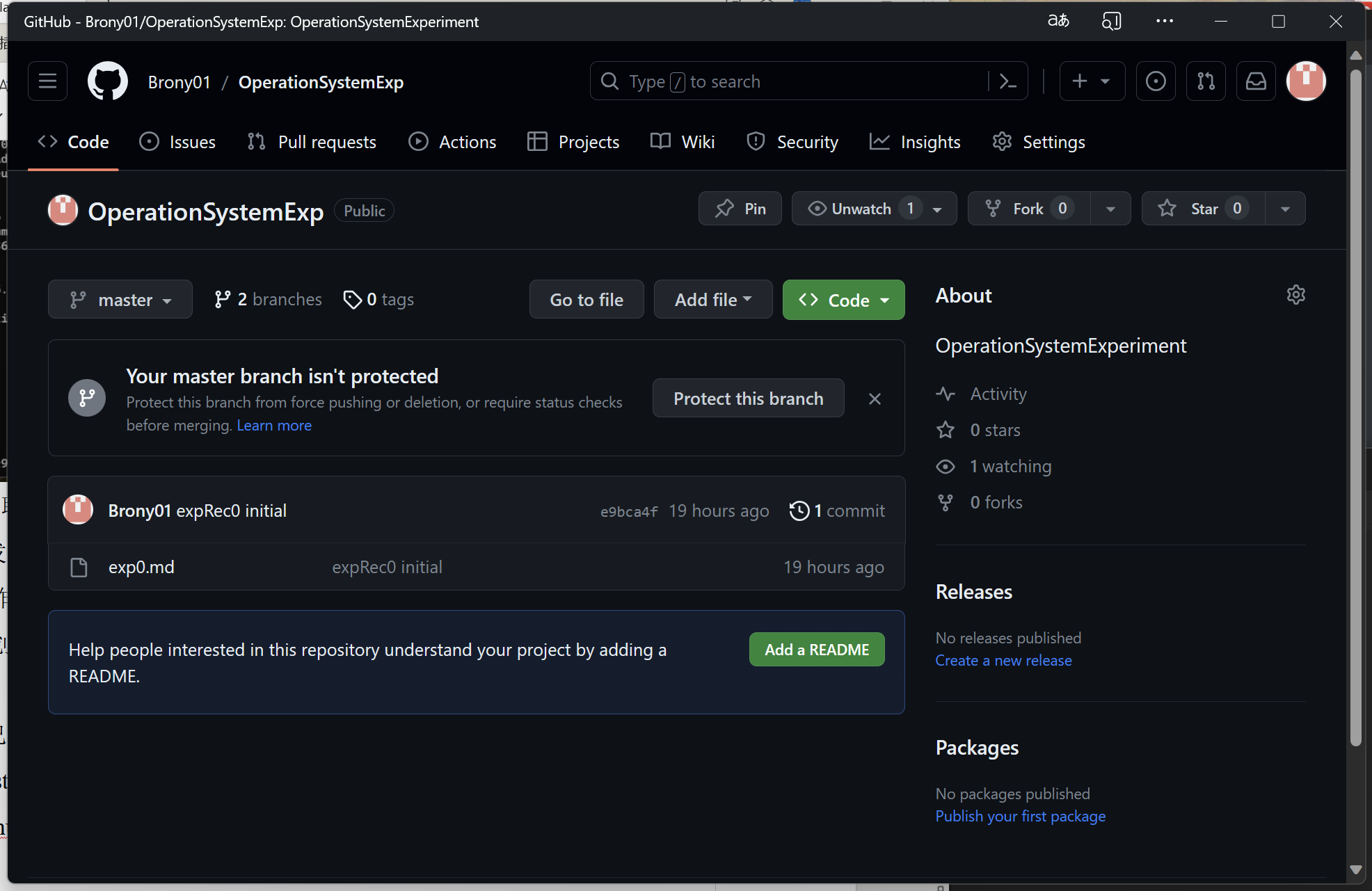
mkdir ~/GardenerOS/

可以看到已经出现GardenerOS目录



2.2远程开发目录

在GitHub创建代码仓库git@github.com:Brony01/OperationSystemExp.git



需要执行如下git相关的操作：

git init

touch .gitignore （windows下可用echo代替touch）

git add .

git commit

git remote add origin [git@github.com:Brony01/OperationSystemExp.git](mailto:git@github.com:Brony01/OperationSystemExp.git)

git push --all

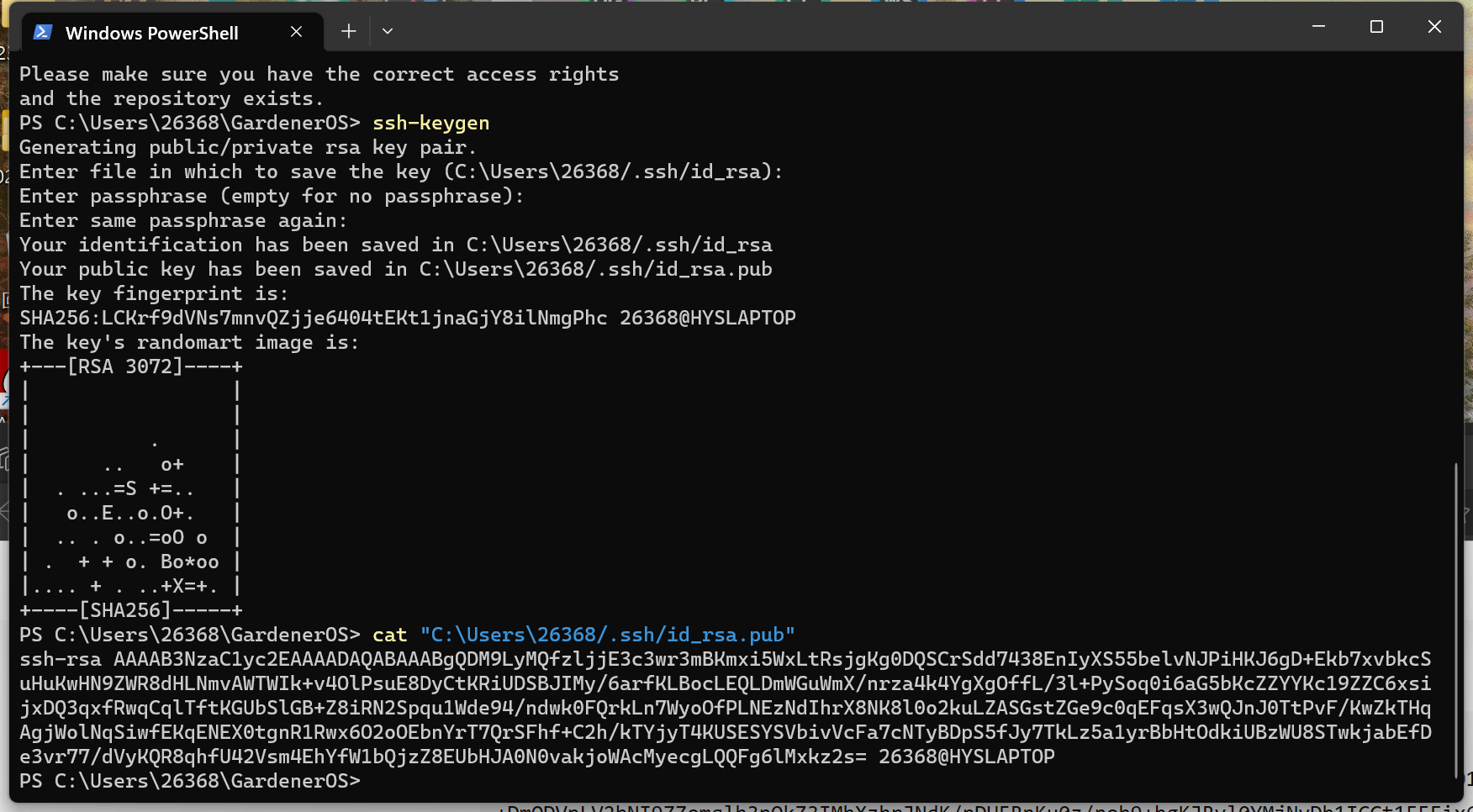


这里添加远程目录需要将本机加入ssh key否则没有访问权限

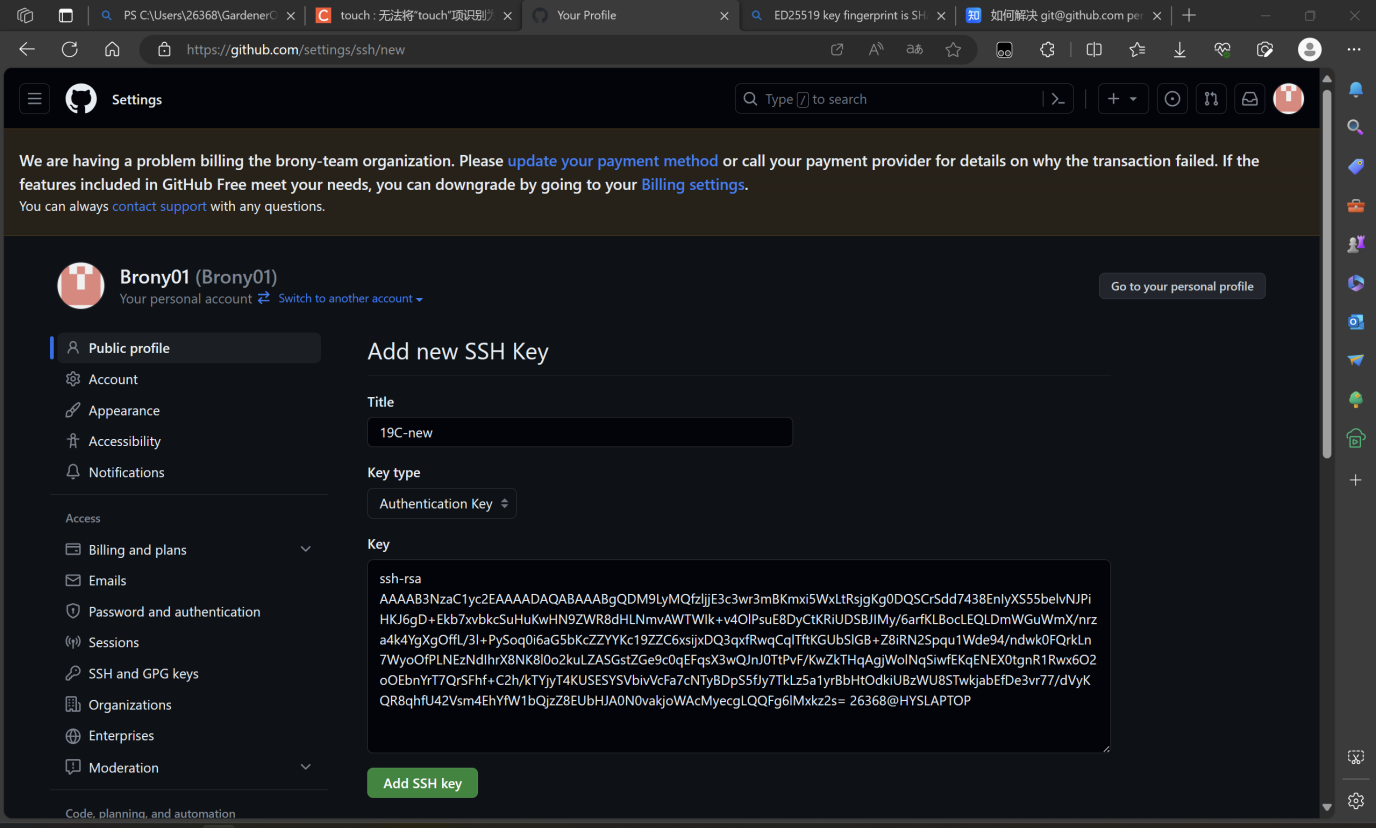
步骤如下

ssh-keygen

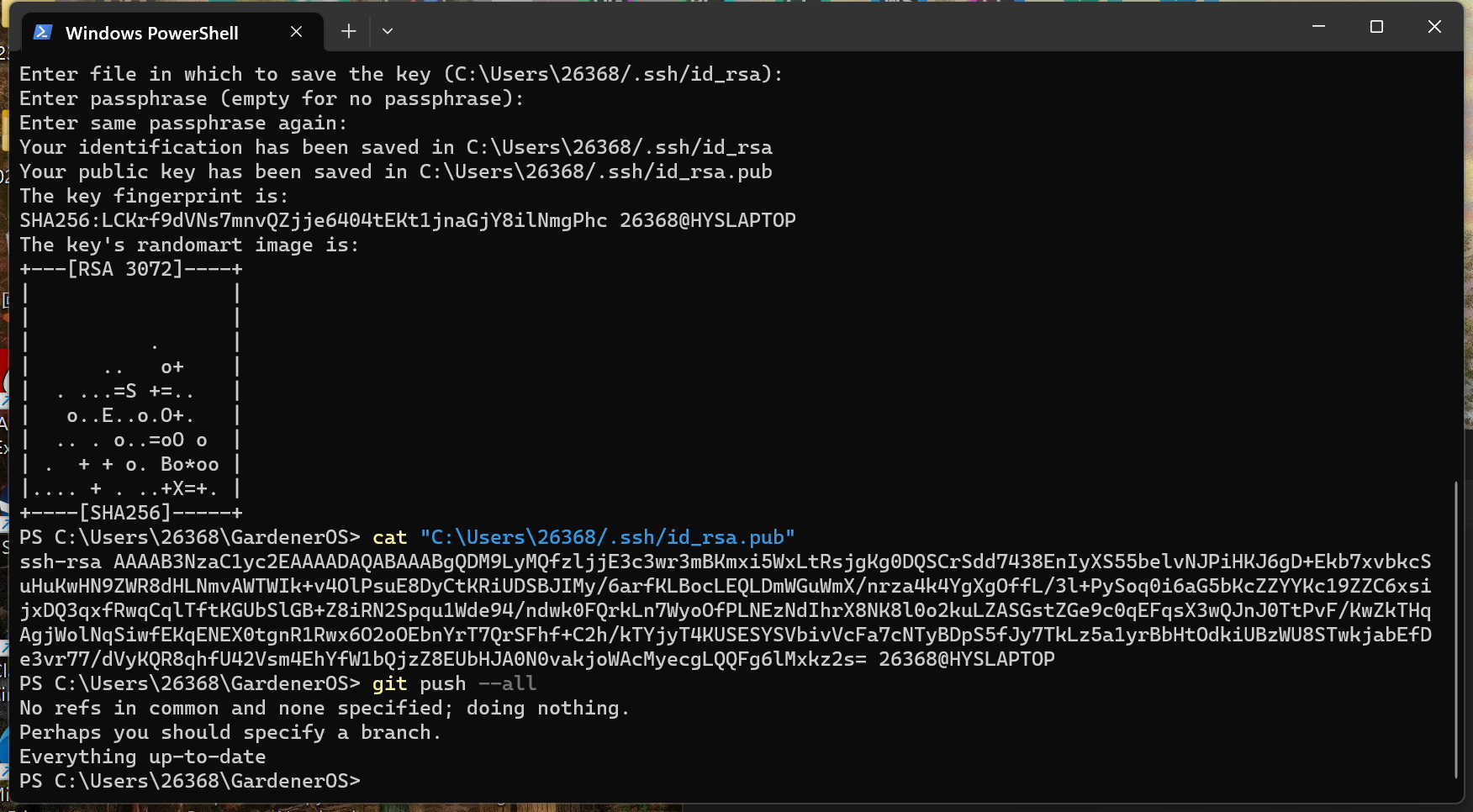
cat “C:\Users\26368/.ssh/id\_rsa.pub”



将得到的ssh-key复制到GitHub的ssh-key设置页面（选择add ssh-key）



随后可以正常访问代码仓库。



当新建的远程分支在本地不可见时，先用fetch命令更新remote索引。

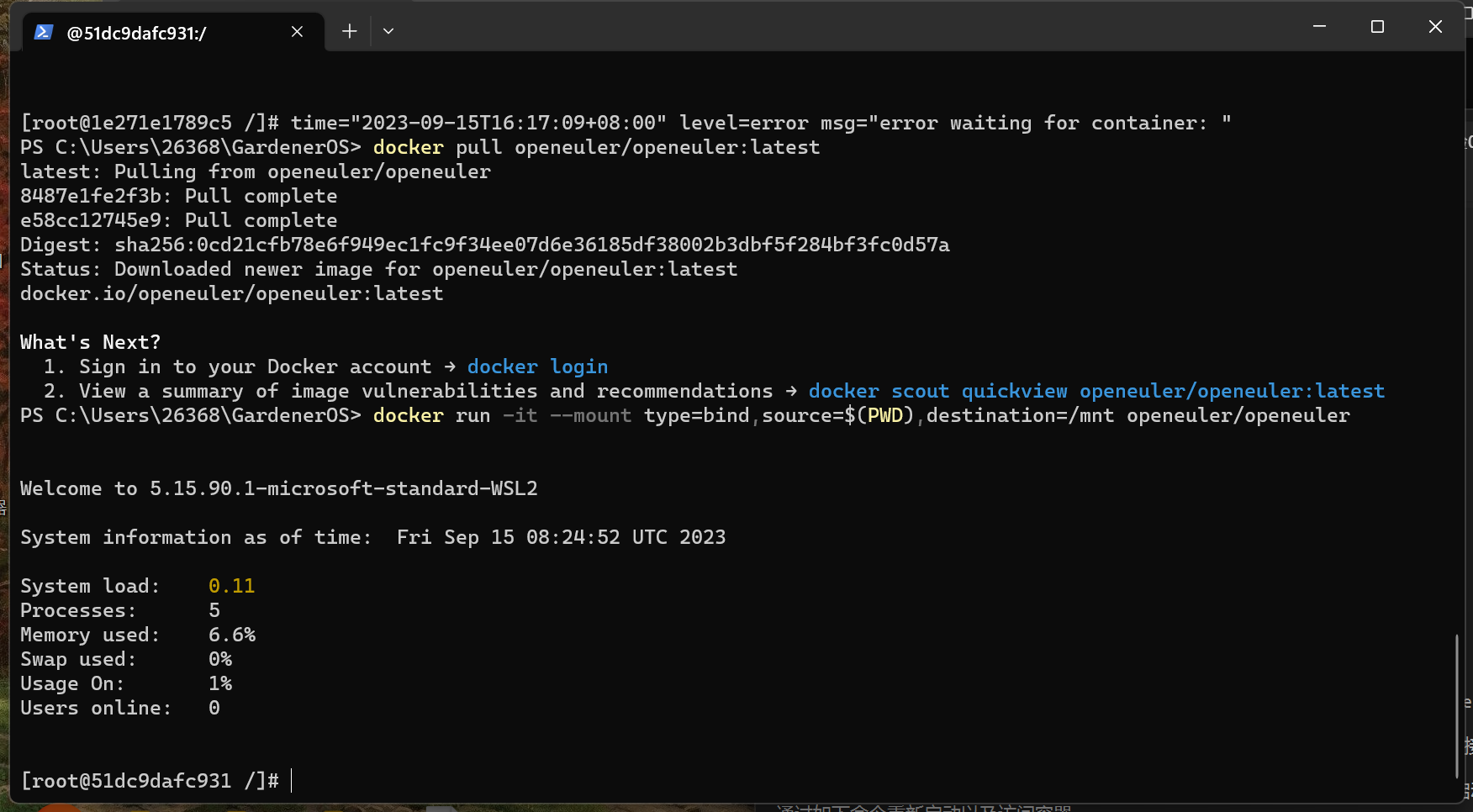
git fetch

1. 创建自己的docker镜像

首先进入自己创建的实验目录

docker run -it --mount type=bind,source=$(PWD),destination=/mnt openeuler/openeuler

通过如上命令启动容器后，本地创建的实验目录就可以直接通过容器里的/mnt目录直接访问了。



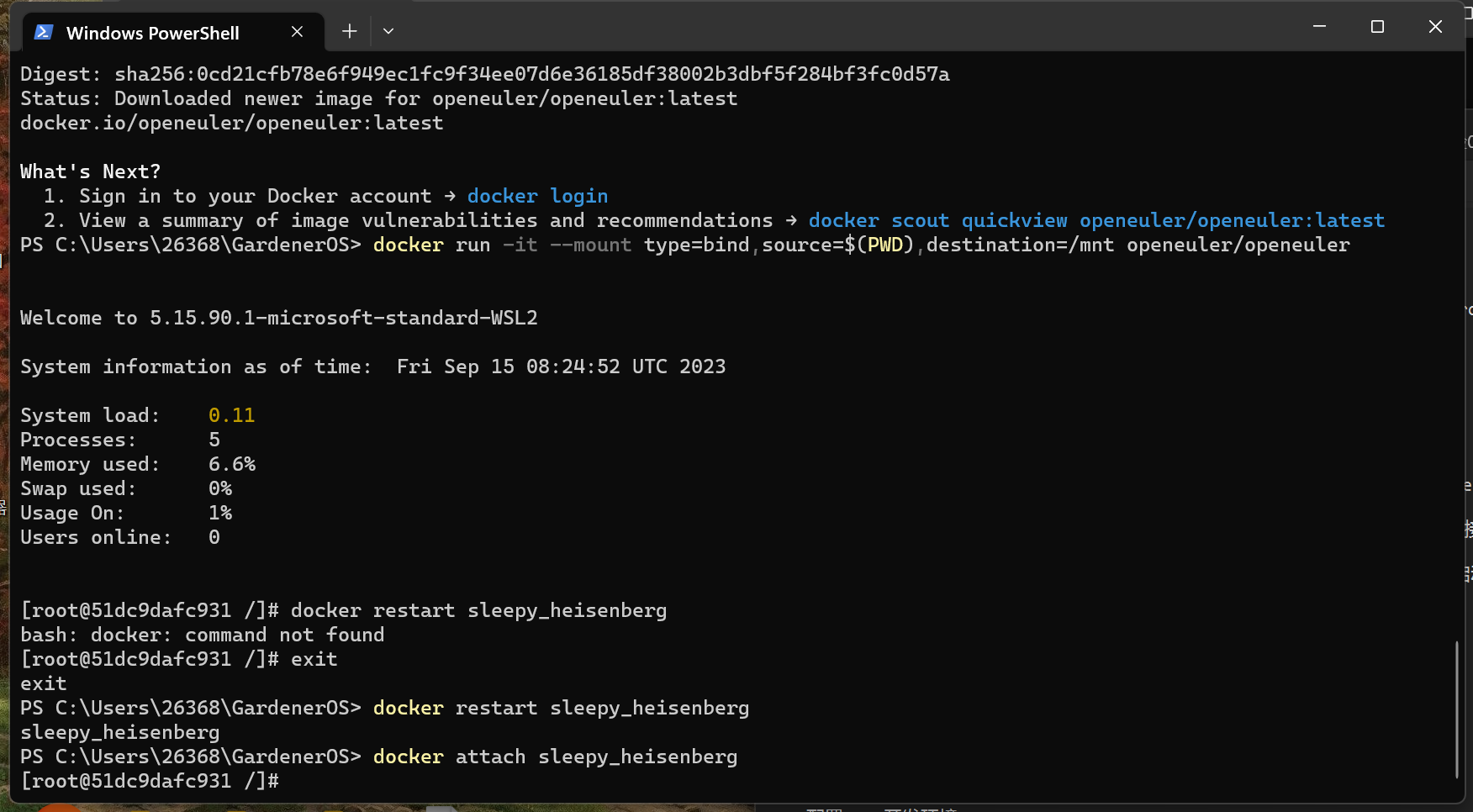
在docker desktop中查看到容器名称为sleepy\_heisenberg



重新启动以及访问容器命令如下。

docker restart sleepy\_heisenberg

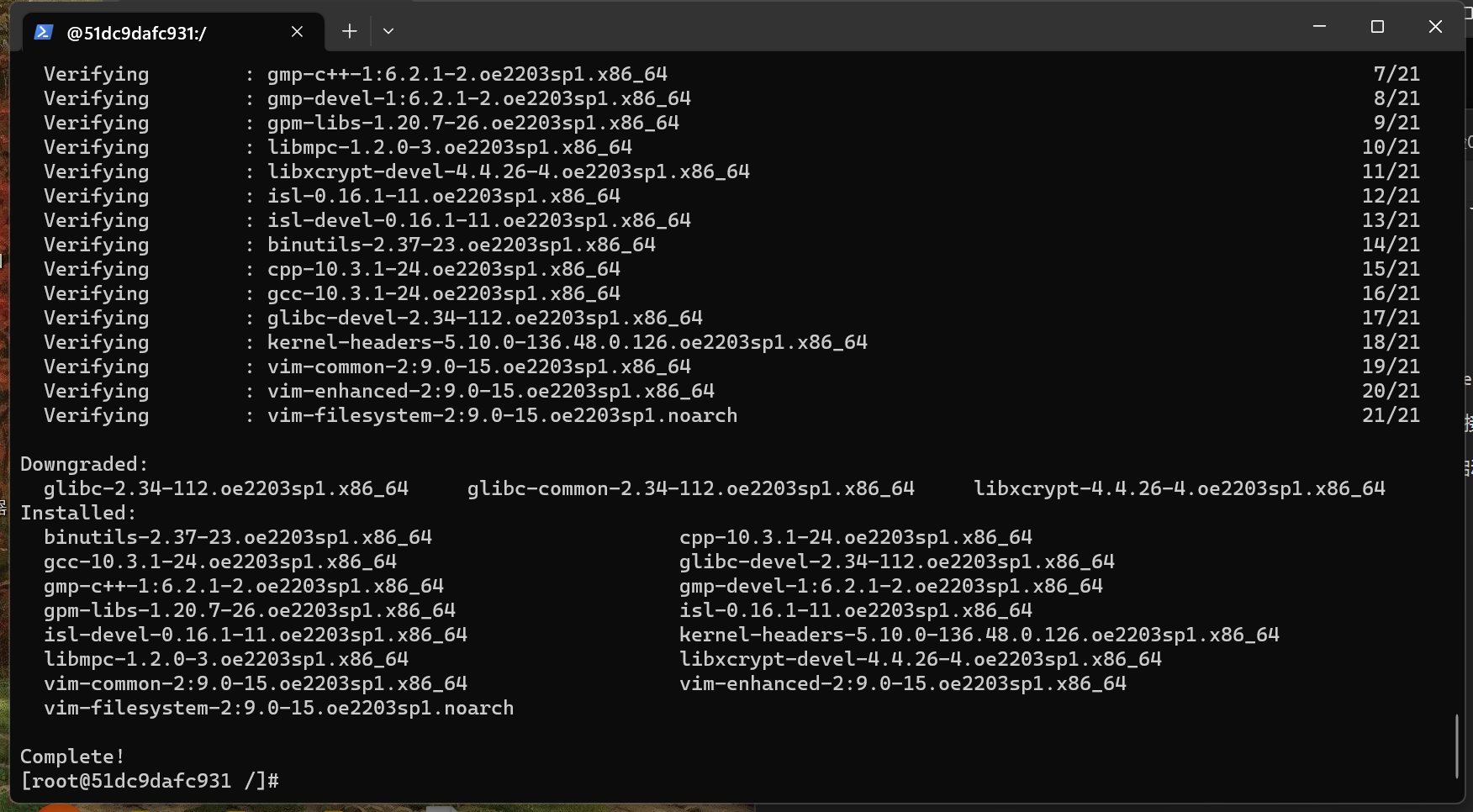
docker attach sleepy\_heisenberg



安装一些必要的工具：

dnf install curl vim gcc

中间确认输入y确认



可以看到安装成功

1. 配置Rust开发环境

4.1通过如下命令安装rust。

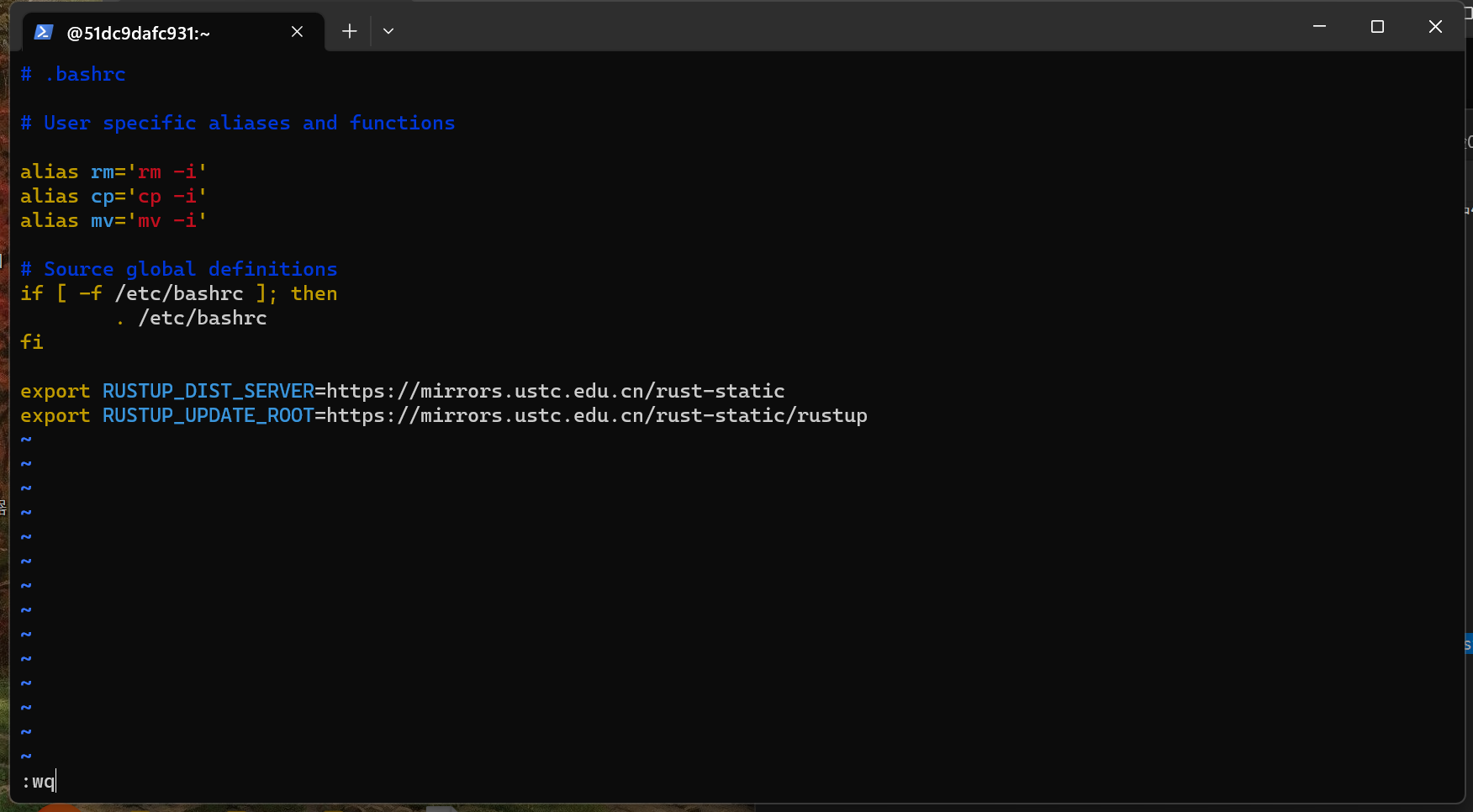
cd ~

vim .bashrc

修改.bashrc增加如下内容（增加位置如图）：

export RUSTUP\_DIST\_SERVER=https://mirrors.ustc.edu.cn/rust-static

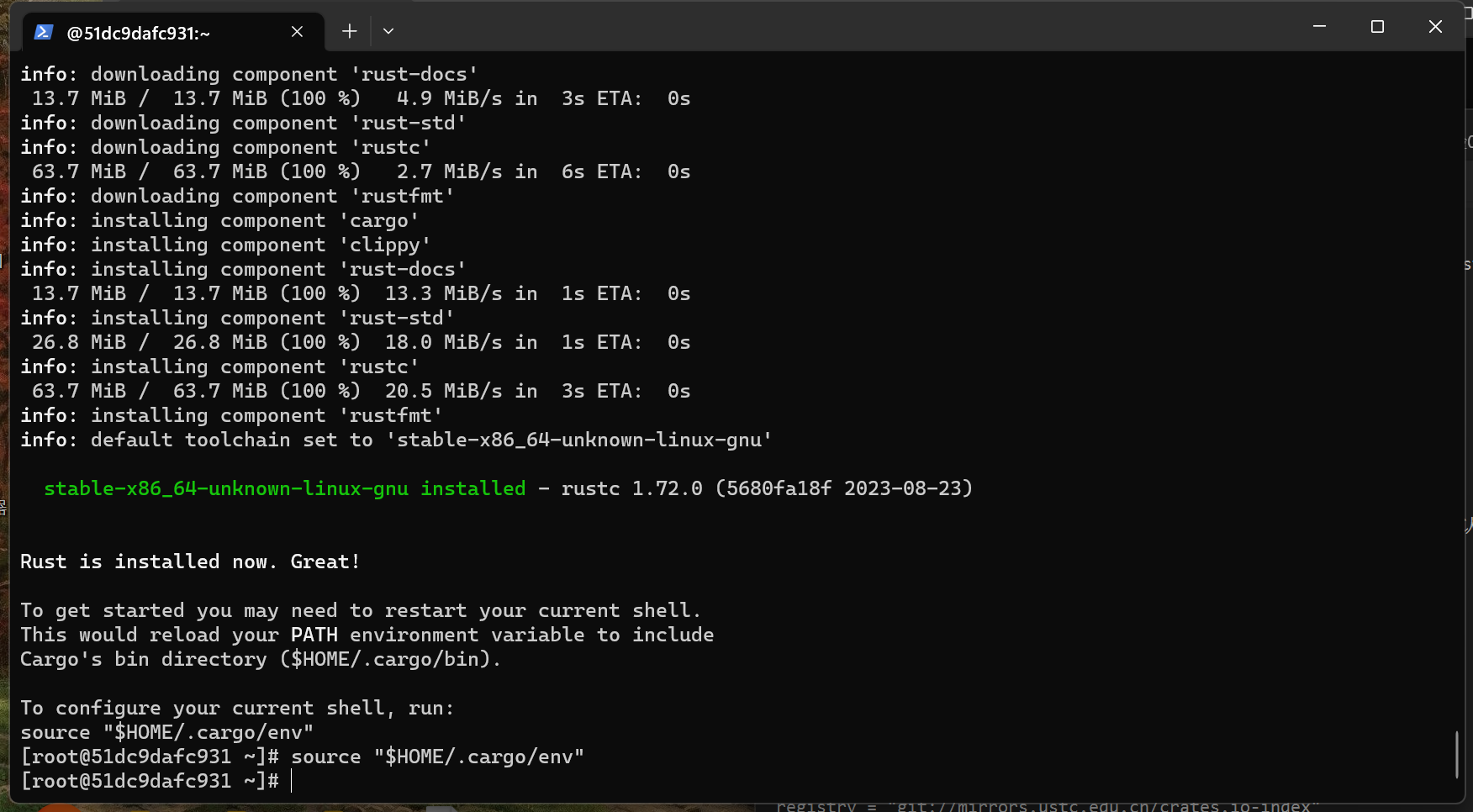
export RUSTUP\_UPDATE\_ROOT=https://mirrors.ustc.edu.cn/rust-static/rustup



然后，通过如下命令安装rust。

curl https://sh.rustup.rs -sSf | sh

选择1（默认安装）



可以看到安装完成

执行如下命令，使得安装环境启用。

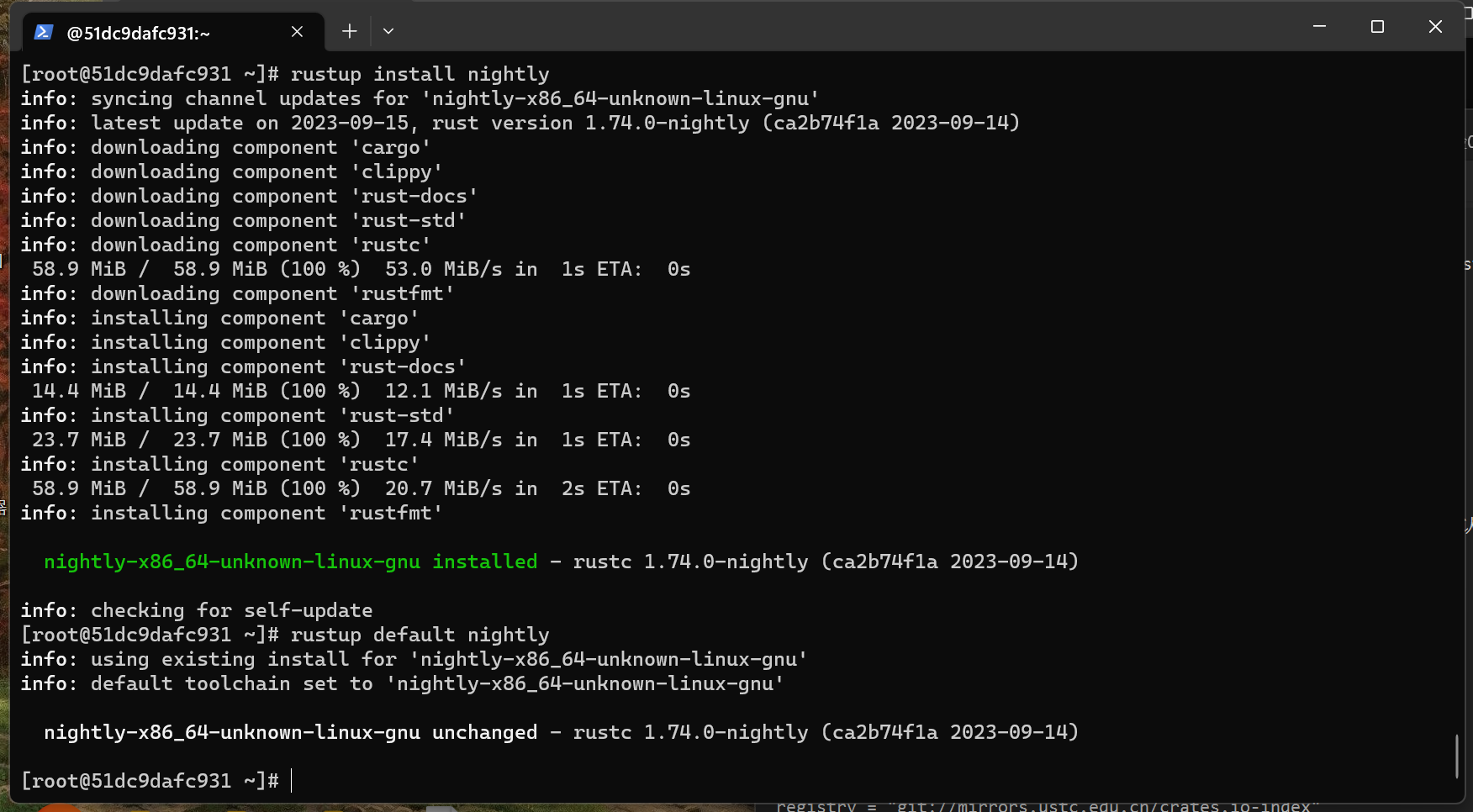
source "$HOME/.cargo/env"

操作系统实验依赖于nightly版本的rust，因此需要安装nightly版本，并将rust默认设置为nightly。

执行如下命令：

rustup install nightly

rustup default nightly



4.2切换cargo软件包镜像为tuna

vim ~/.cargo/config

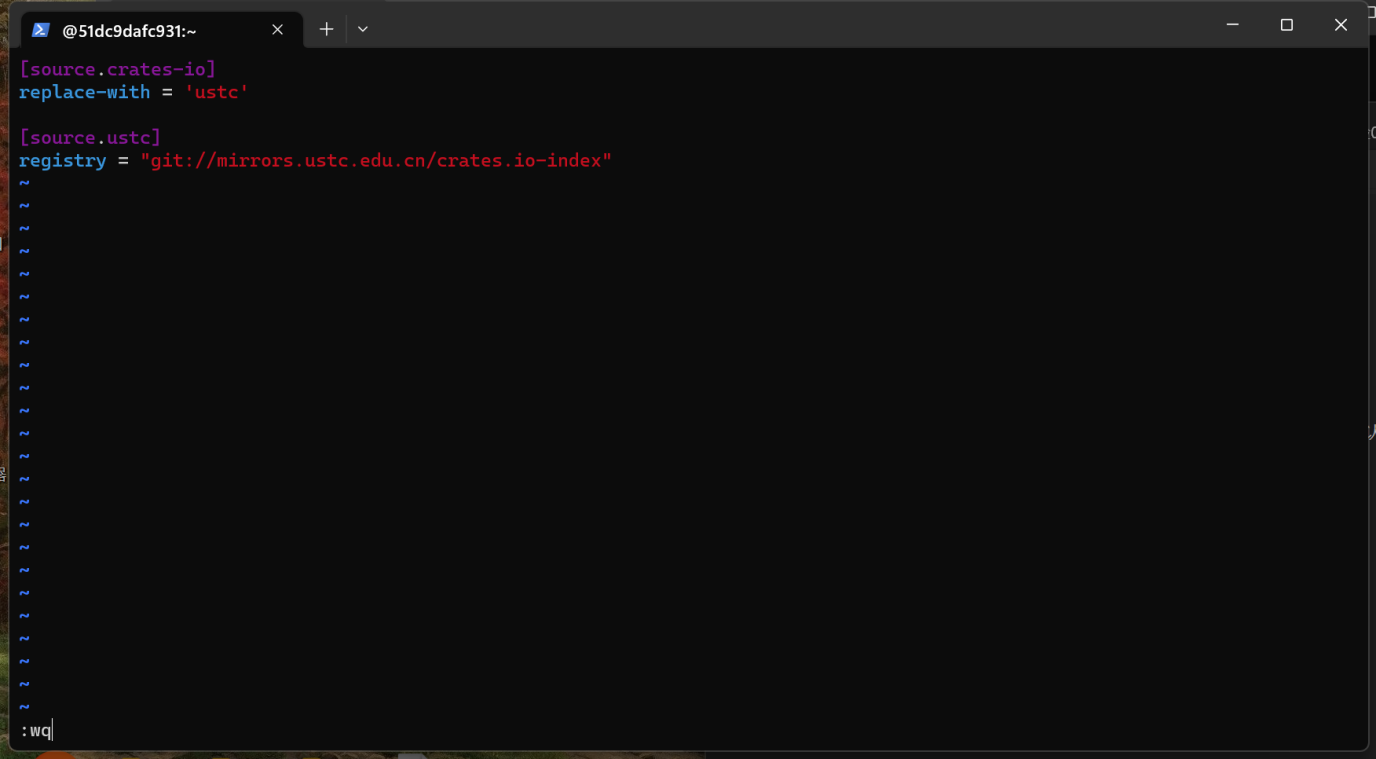
文件内容如下：

[source.crates-io]

replace-with = 'ustc'

[source.ustc]

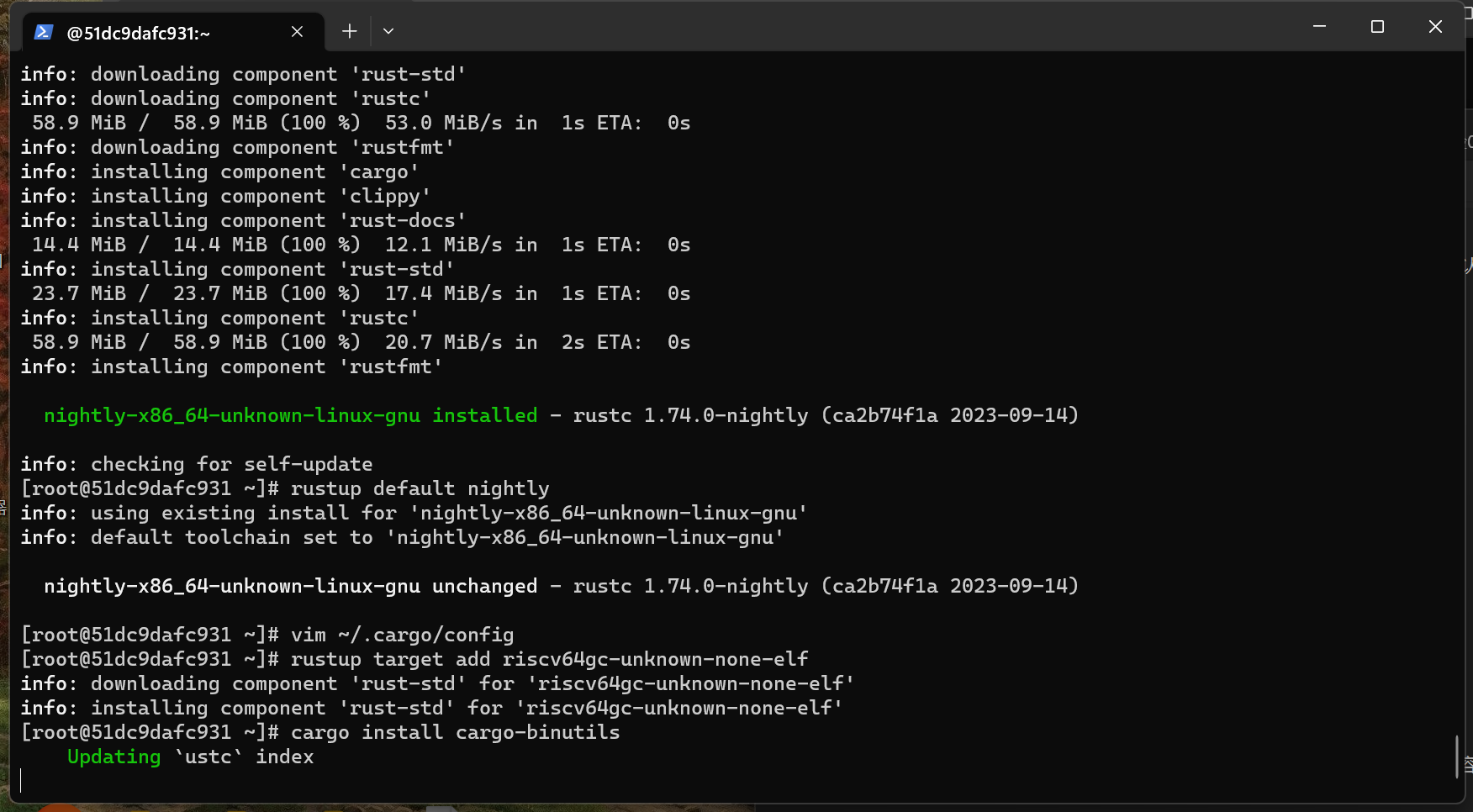
registry = "git://mirrors.ustc.edu.cn/crates.io-index"



4.3接下来安装一些Rust相关的软件包

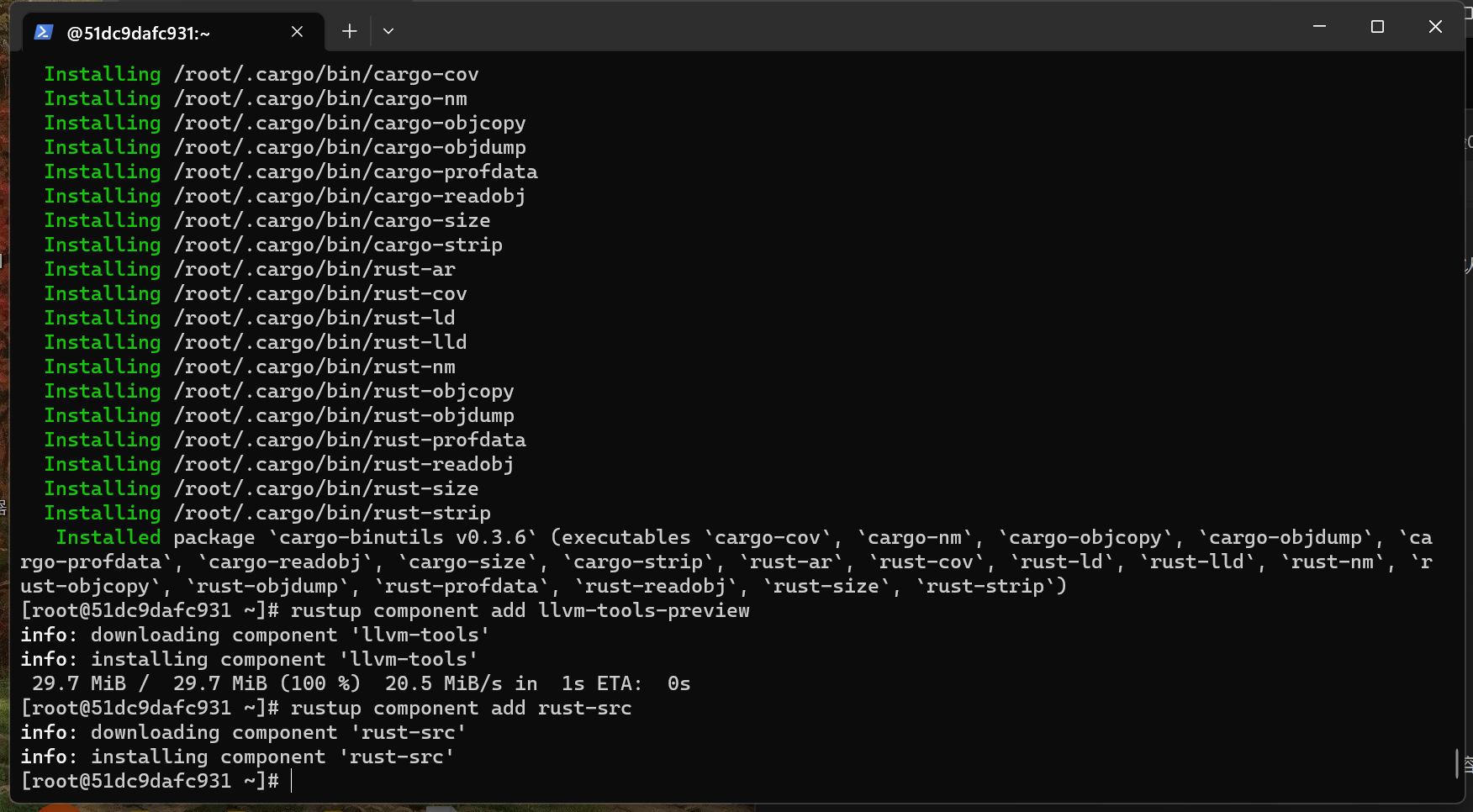
rustup target add riscv64gc-unknown-none-elf

cargo install cargo-binutils



rustup component add llvm-tools-preview

rustup component add rust-src

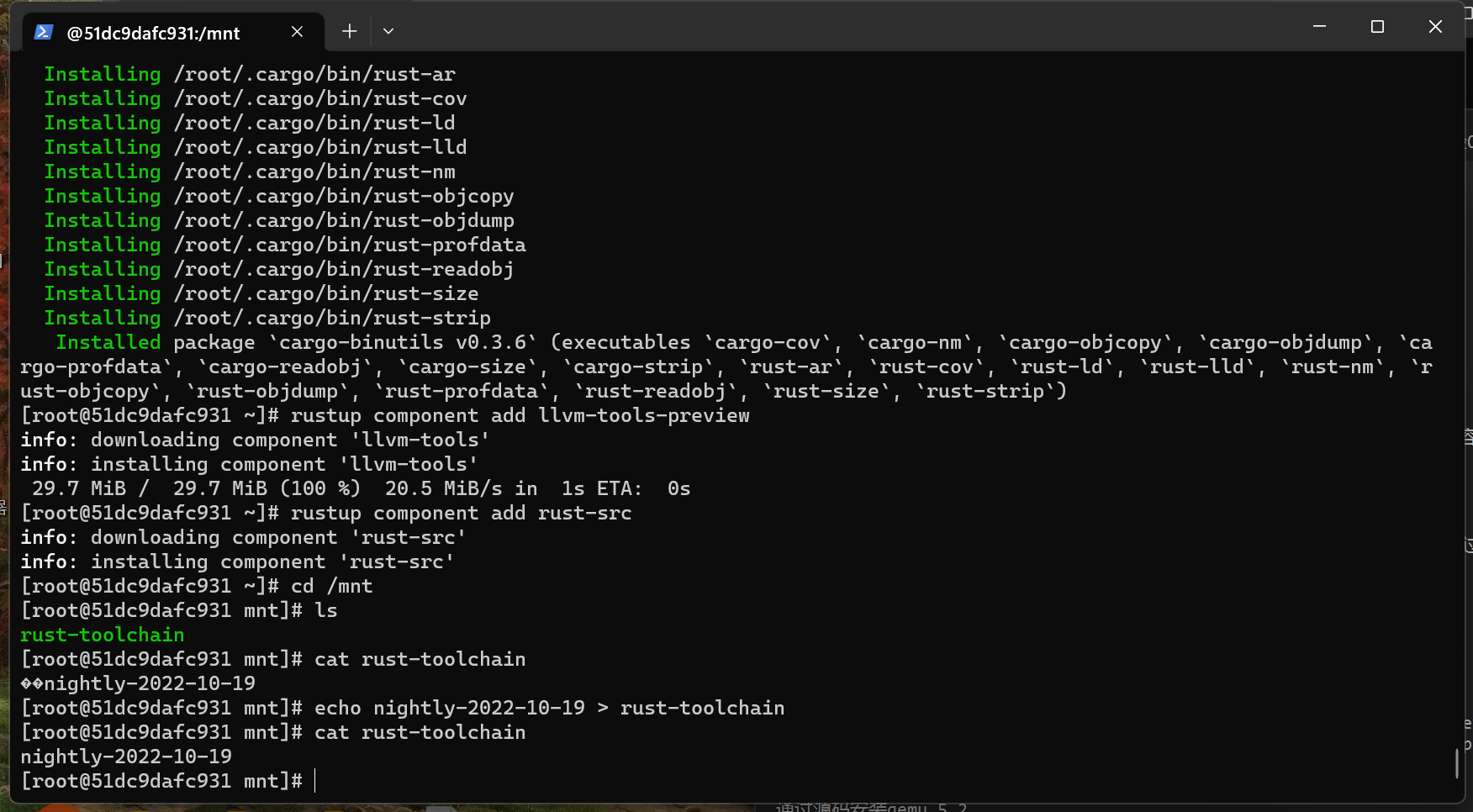


4.4限制rust的版本

在工作目录下创建一个名为 rust-toolchain 的文件，以限制rust的版本，文件内容如下：

nightly-2022-10-19

命令为echo nightly-2022-10-19 > rust-toolchain

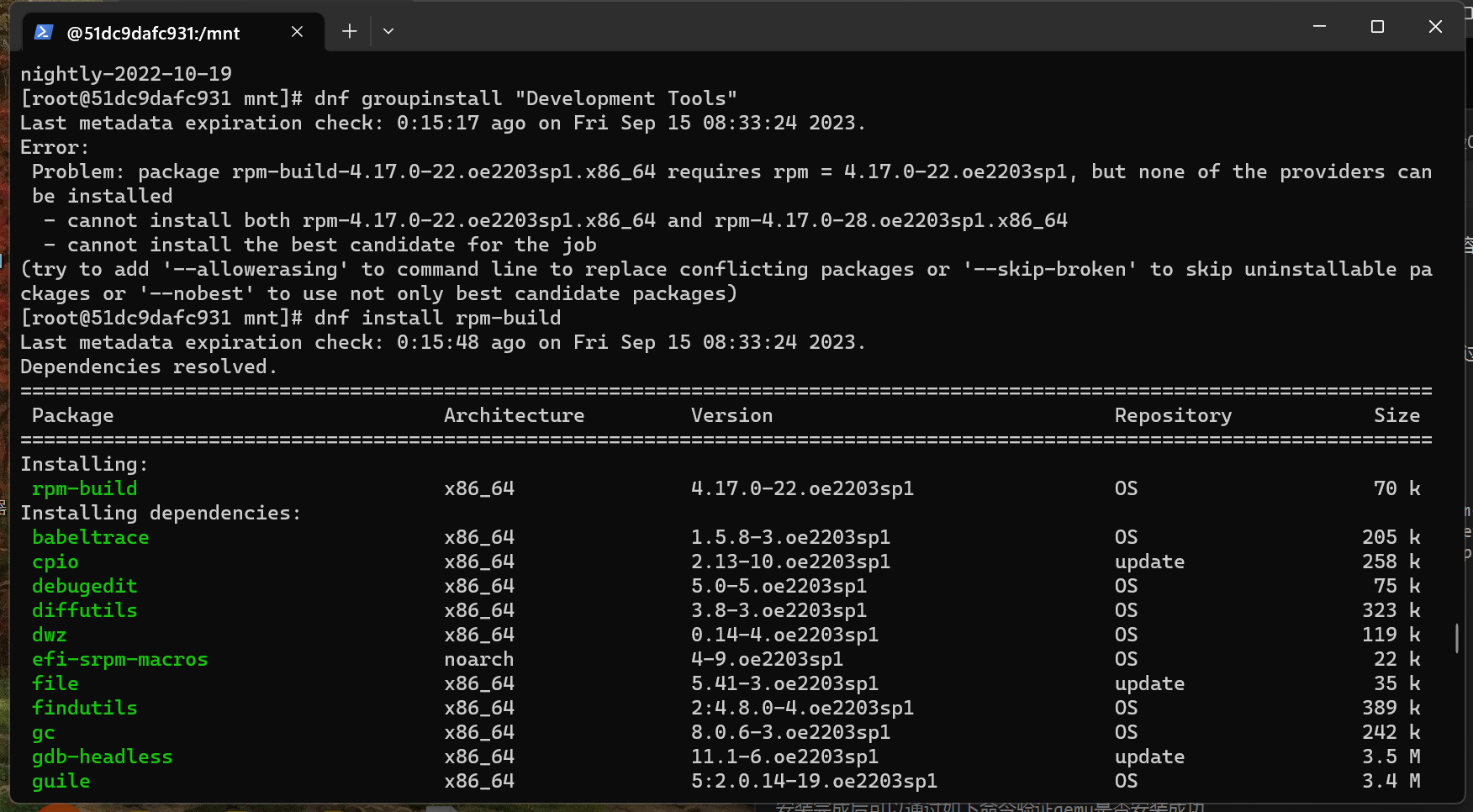


说明：访问工作目录有两种方式：在本地通过~/GardenerOS/目录访问；在容器内通过/mnt/目录访问。

1. 安装qemu

5.1执行如下命令安装基本的软件包

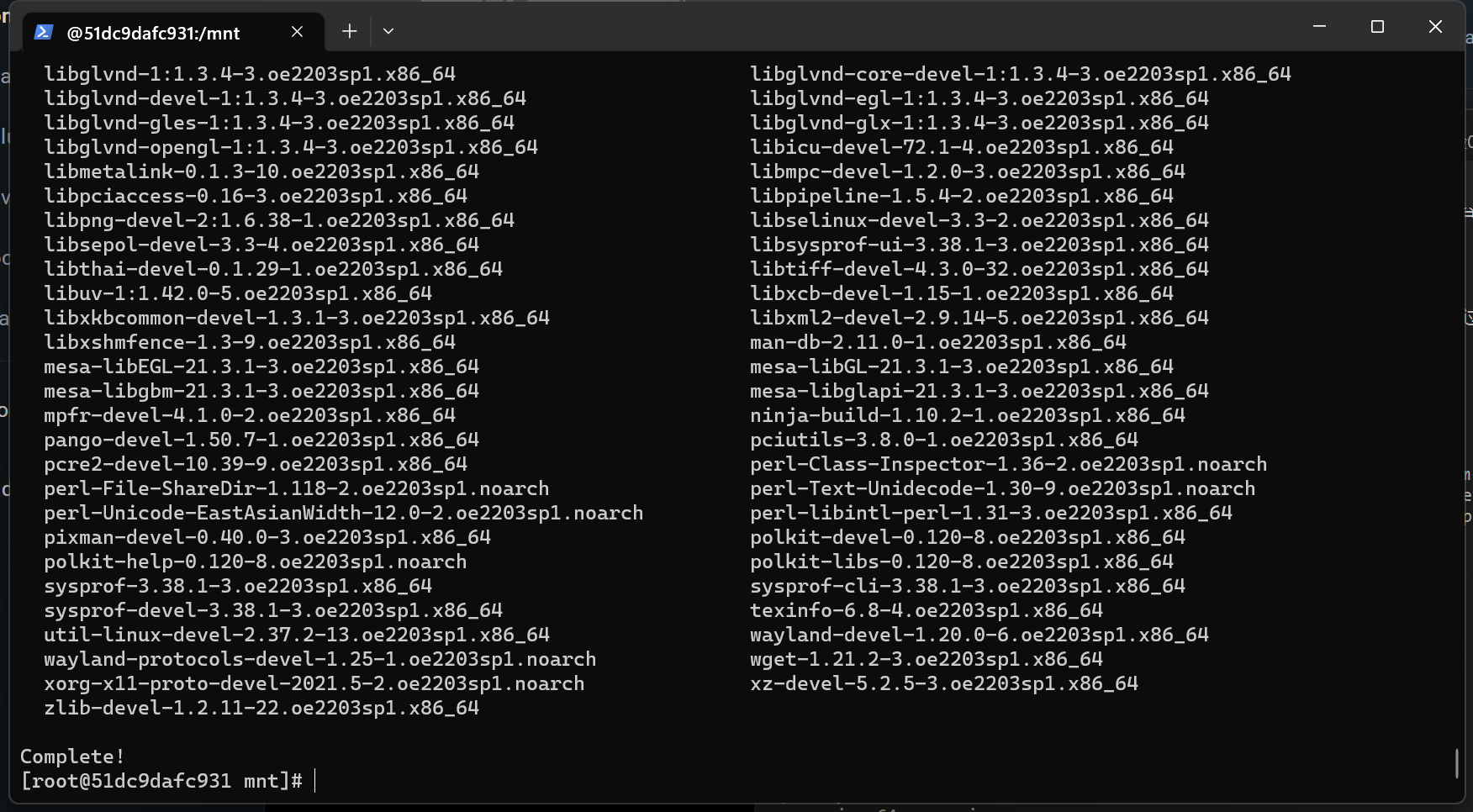
dnf groupinstall "Development Tools" 未成功——需先执行dnf install rpm-build然后再执行



dnf install autoconf automake gcc gcc-c++ kernel-devel curl libmpc-devel mpfr-devel gmp-devel \

glib2 glib2-devel make cmake gawk bison flex texinfo gperf libtool patchutils bc \

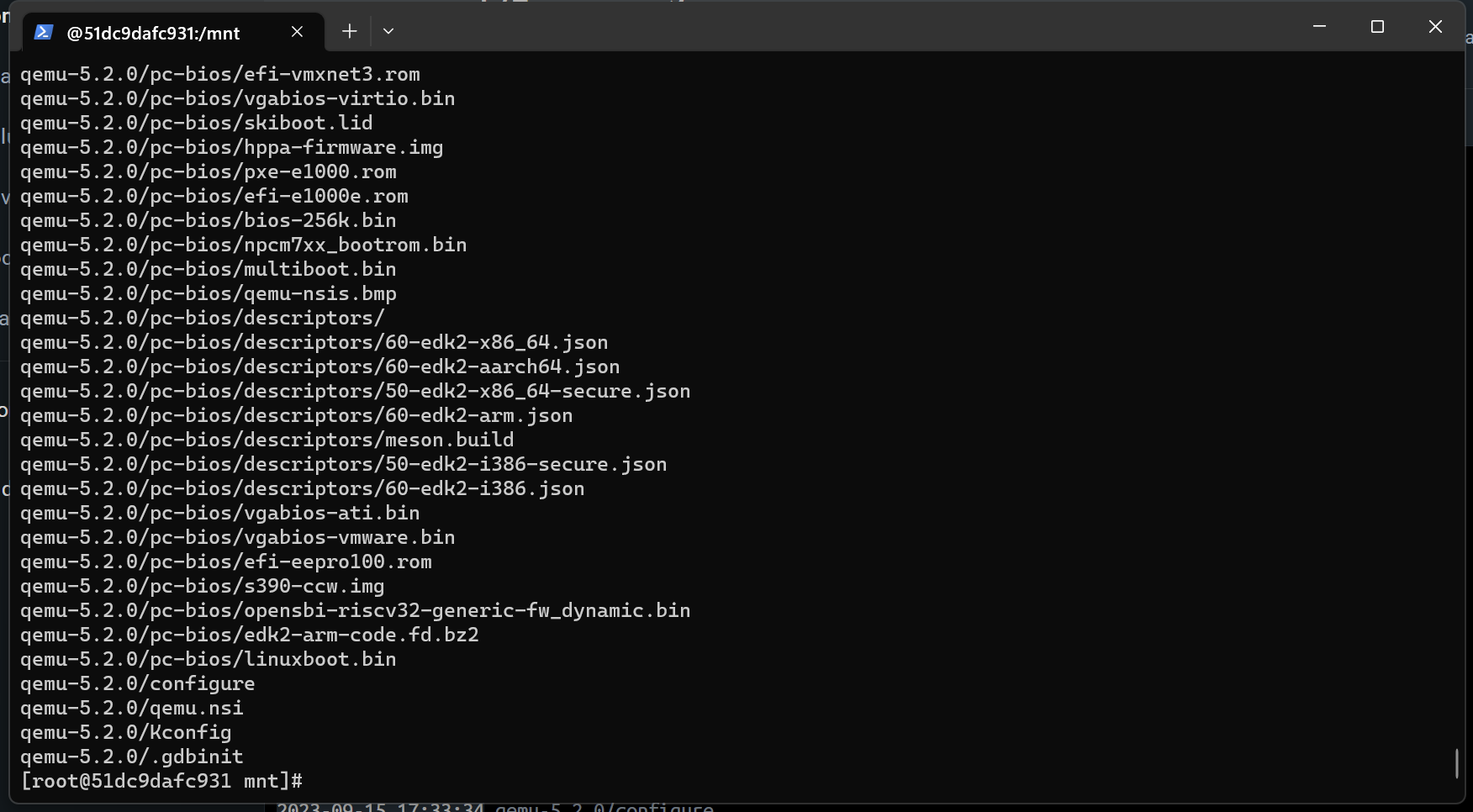
python3 ninja-build wget xz



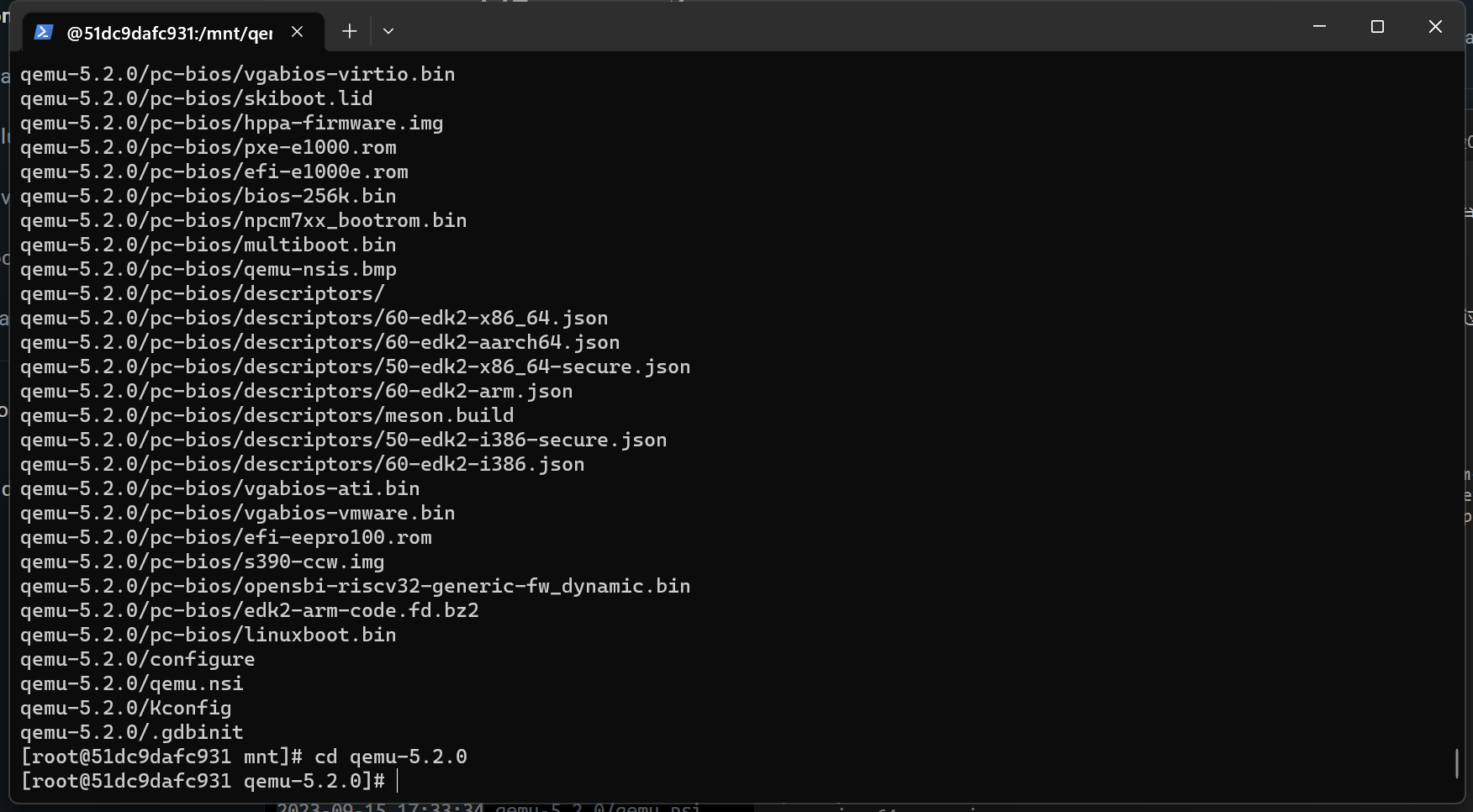
5.2通过源码安装qemu 5.2并验证安装是否成功

wget https://download.qemu.org/qemu-5.2.0.tar.xz

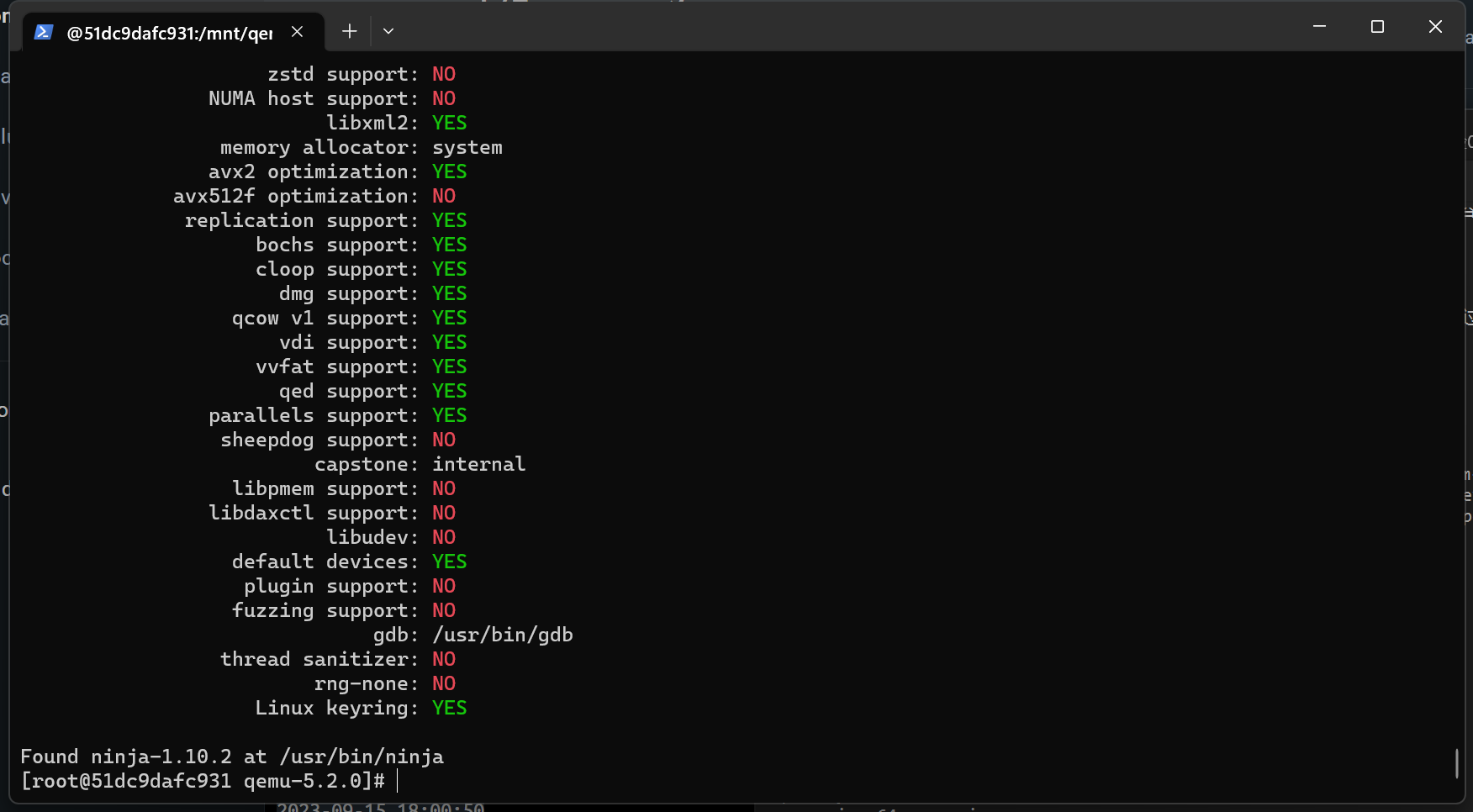
tar xvJf qemu-5.2.0.tar.xz



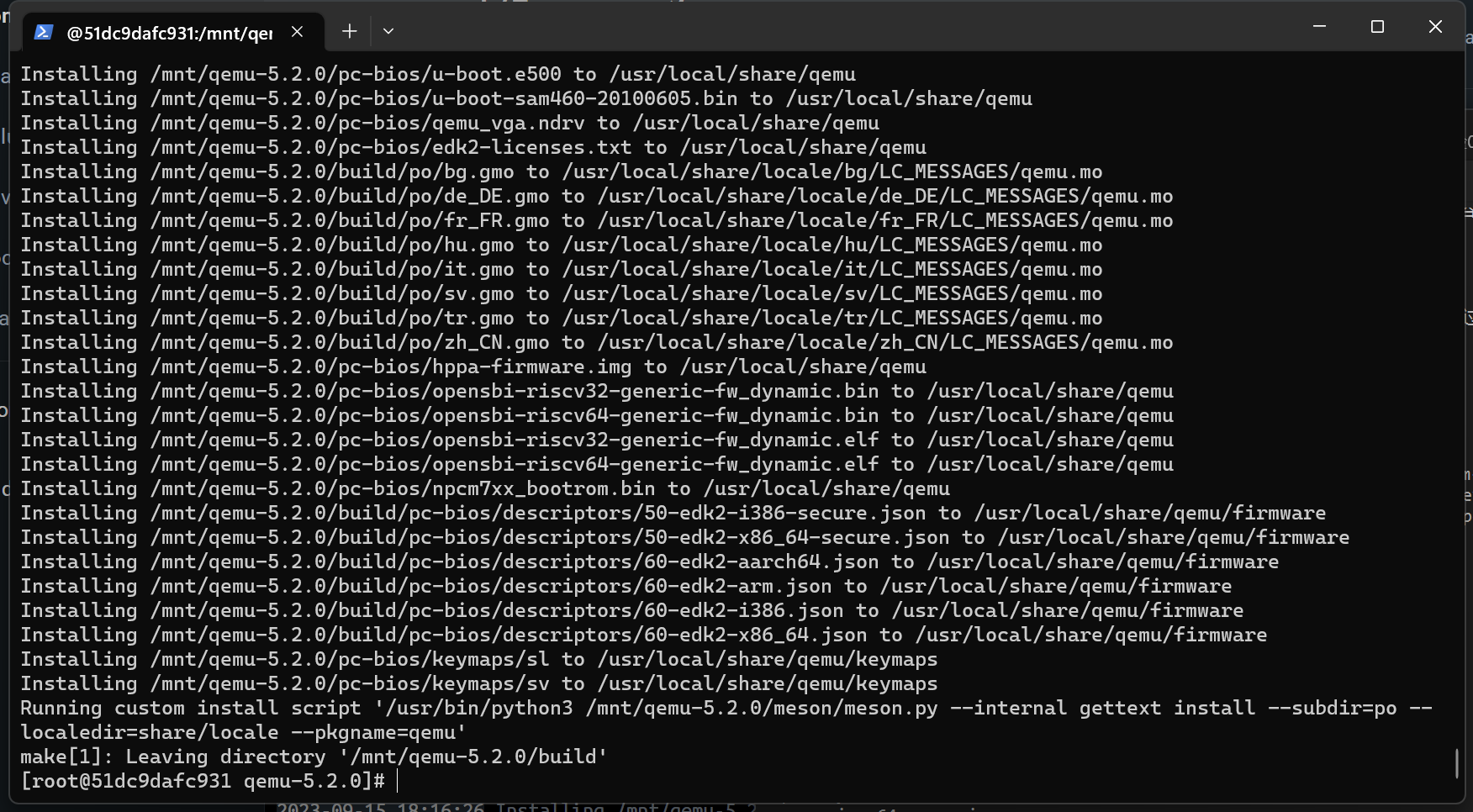
cd qemu-5.2.0



./configure --target-list=riscv64-softmmu,riscv64-linux-user



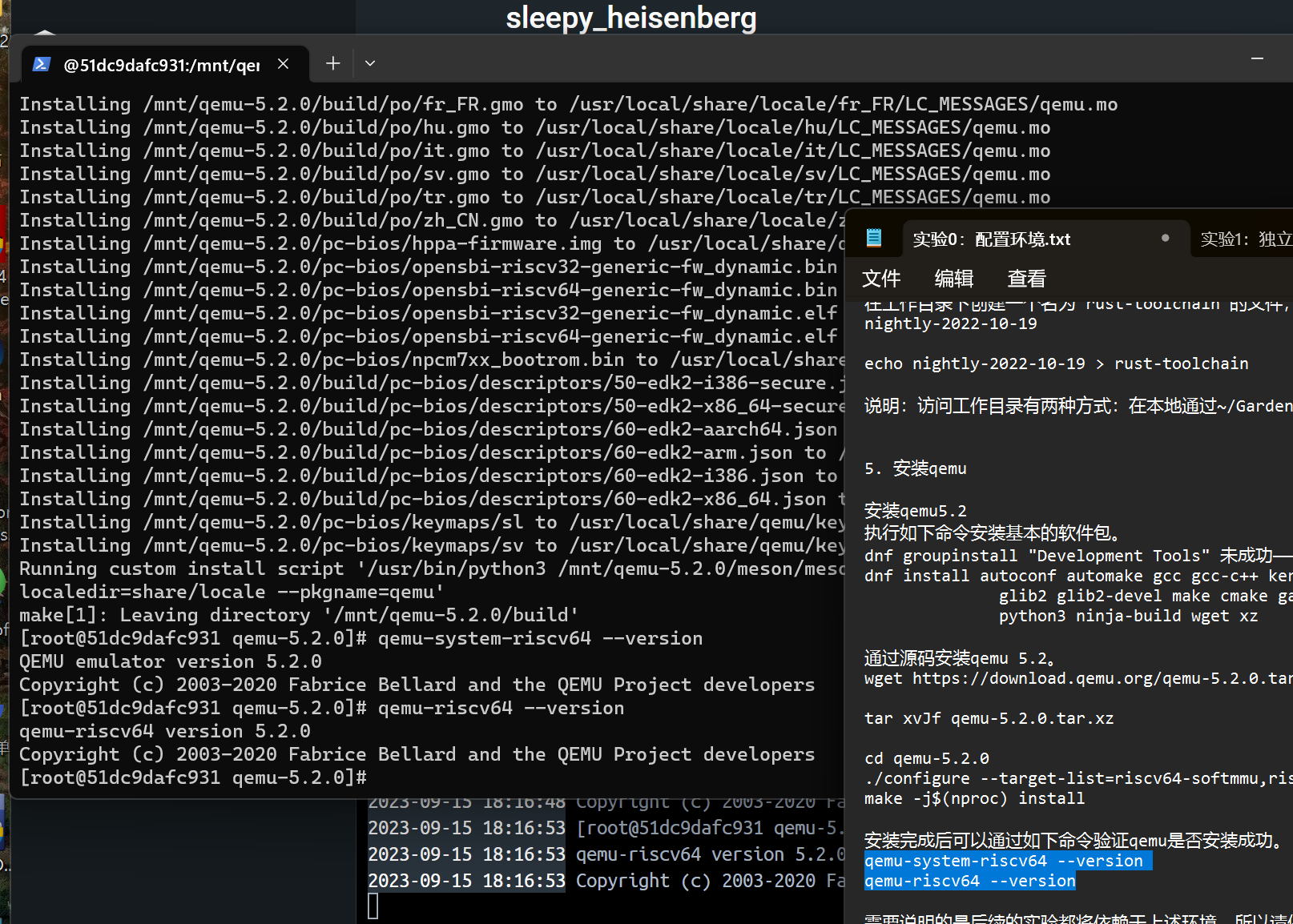
make -j$(nproc) install



安装完成后可以通过如下命令验证qemu是否安装成功。

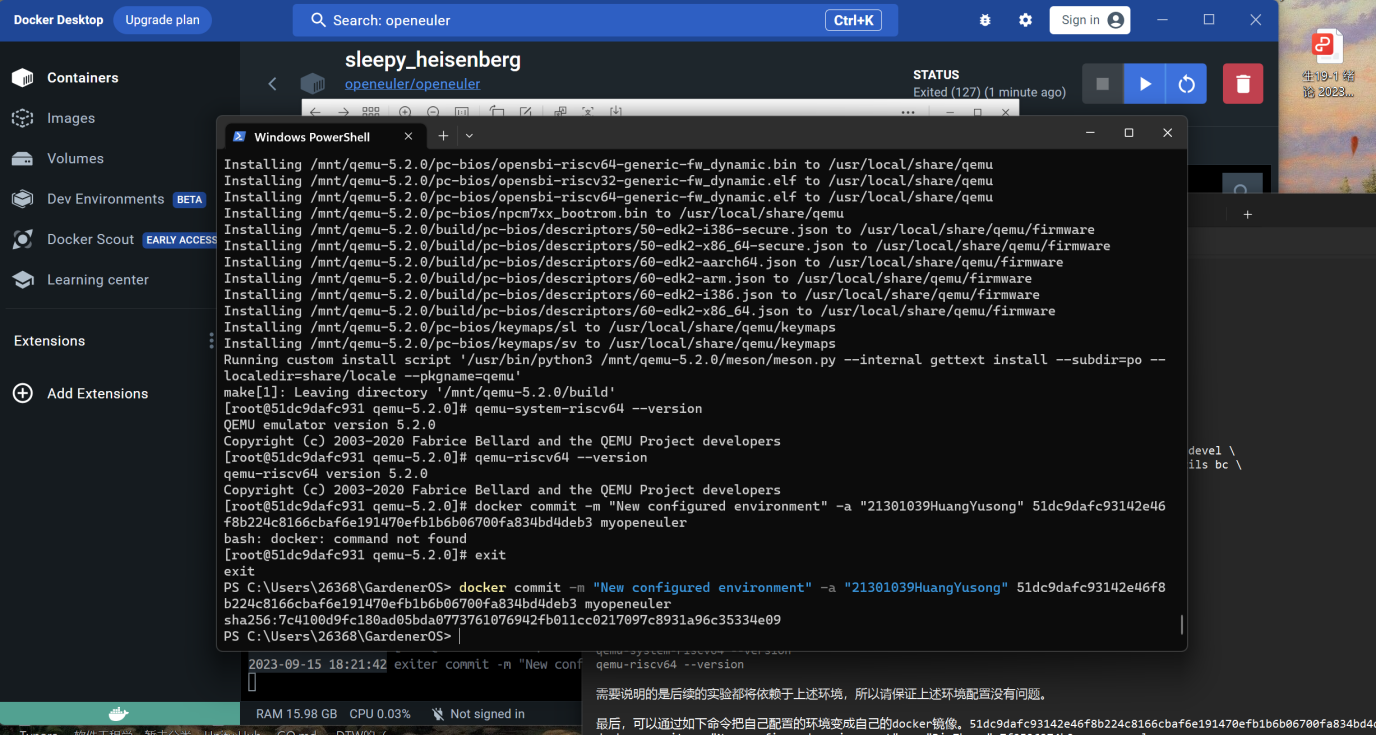
qemu-system-riscv64 --version

qemu-riscv64 --version



最后，可以通过如下命令把自己配置的环境变成自己的docker镜像。

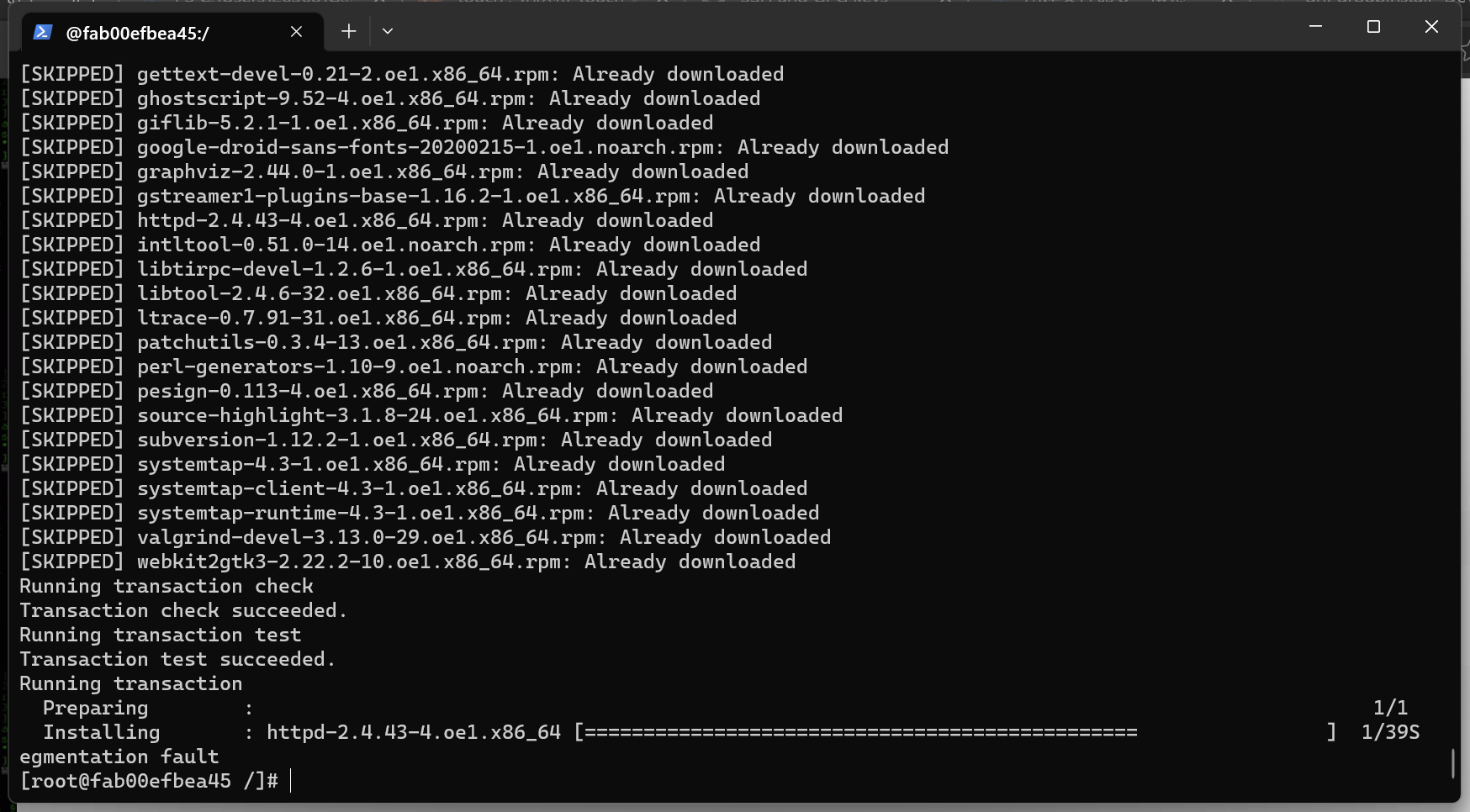
docker commit -m "New configured environment" -a "21301039HuangYusong" 51dc9dafc93142e46f8b224c8166cbaf6e191470efb1b6b06700fa834bd4deb3 myopeneuler



1. 思考问题
2. Git提交截图
3. 其他说明

遇到过如下问题（按步骤顺序）

1.拉取openeuler镜像时由于之前配置过docker镜像源的原因，直接执行docker pull openeuler/openeuler拉取到了较老版本的镜像，这导致后续步骤安装基本软件包执行dnf groupinstall "Development Tools"时会报如下错误Segmentation fault（如下图）并且一时无法找到其他报错信息。



在多次重新配置后用docker pull openeuler/openeuler:latest拉取到了最新镜像，上述问题解决。

1. 第一遍执行dnf groupinstall "Development Tools" 未成功，问题确认为缺乏依赖项，执行如下命令安装依赖项dnf install rpm-build后再执行dnf groupinstall "Development Tools"即可安装。

