Riscv

下载 QEMU 源代码并构建

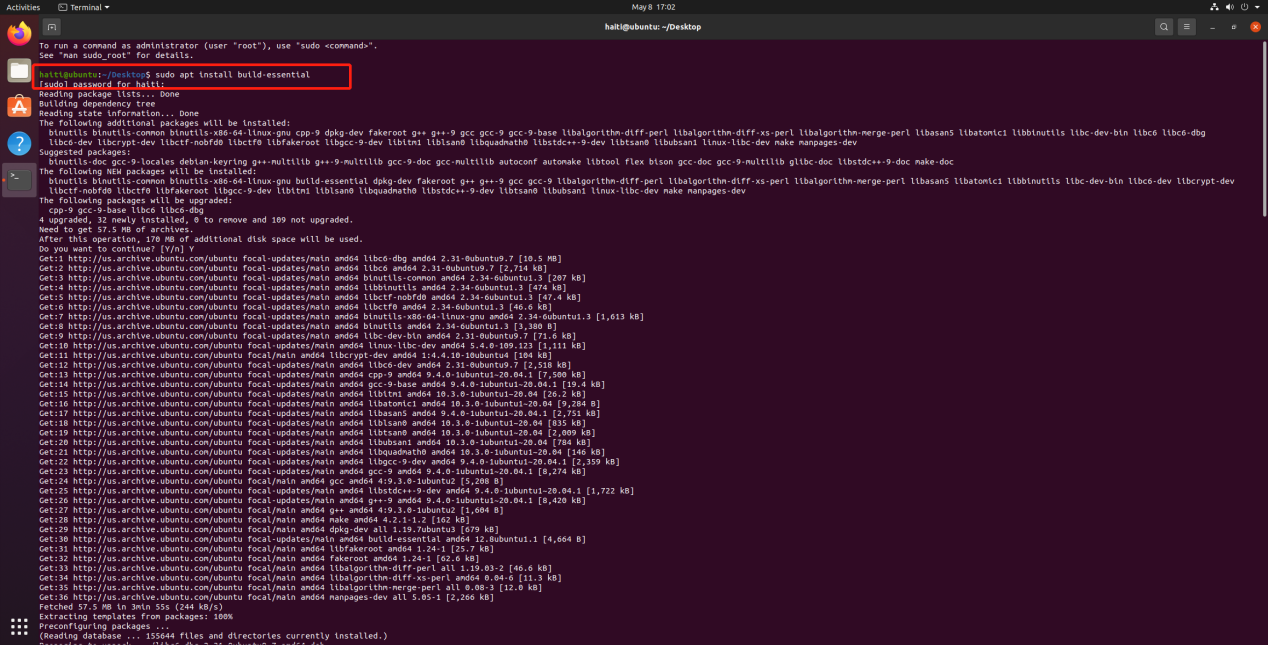
$ sudo apt install build-essential 安装必要的构建工具

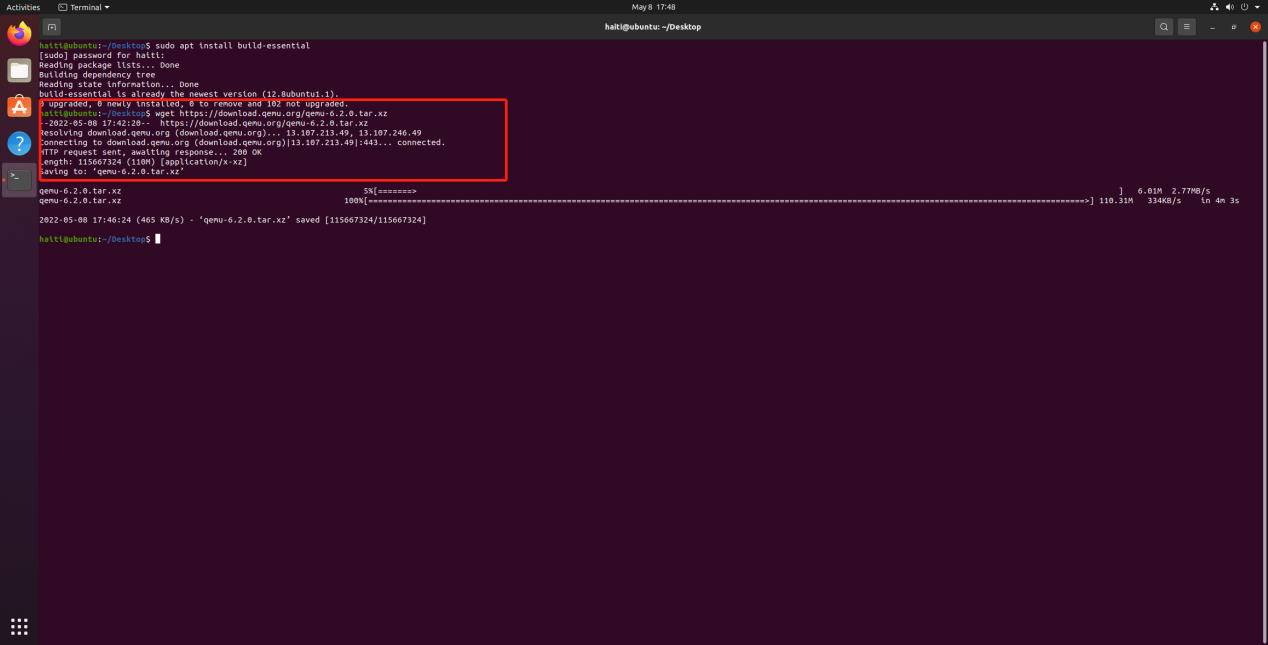
$ wget https://download.qemu.org/qemu-6.2.0.tar.xz 下载最新 QEMU 源码包，请将 <latest> 替换为目前最新的 QEMU 版本

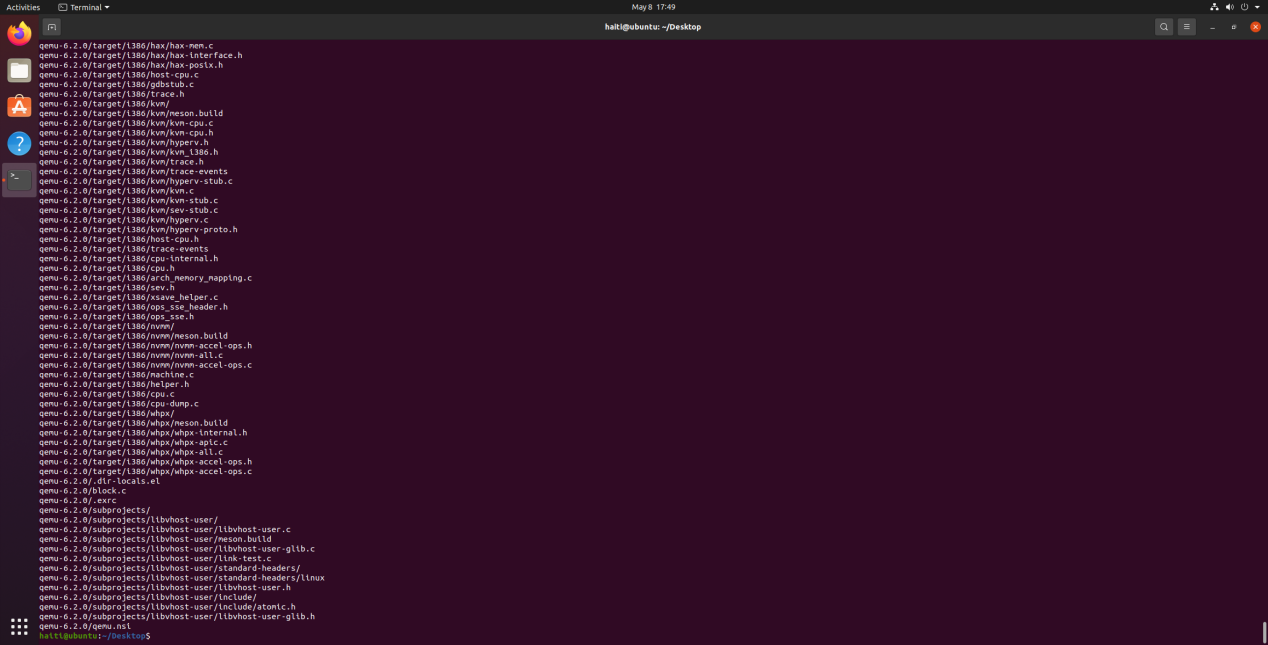
$ tar xvJf qemu-6.2.0.tar.xz

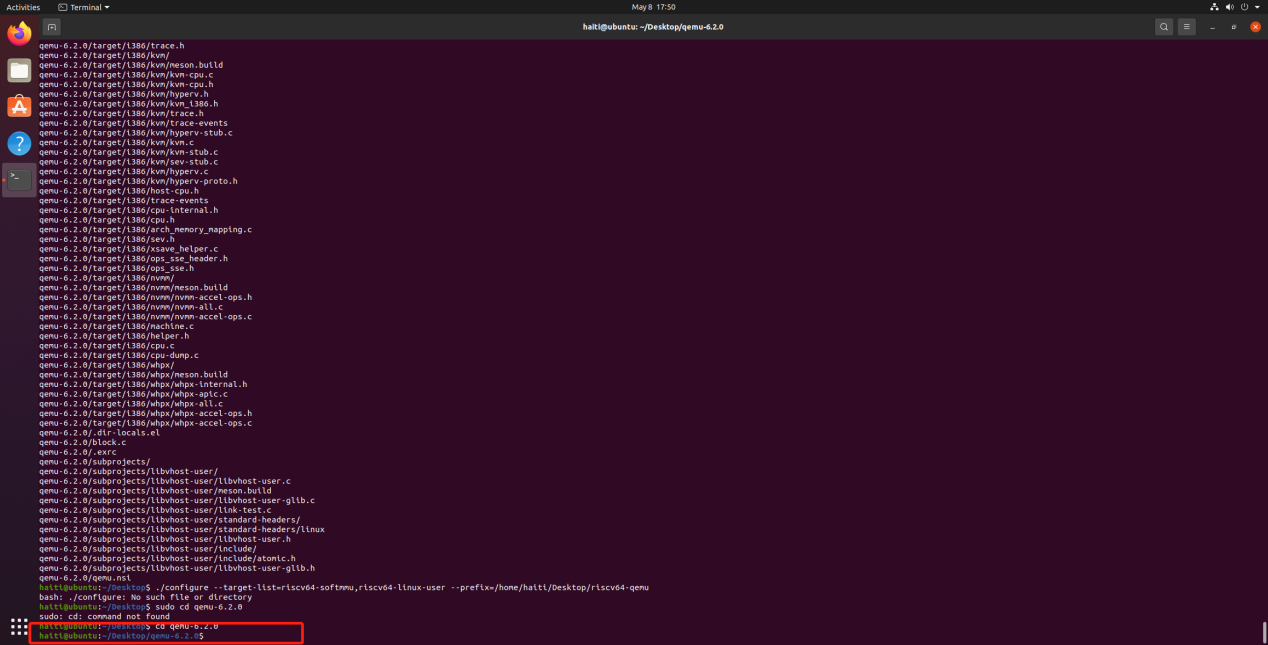
解压刚刚下载的源码包

$ cd qemu-6.2.0

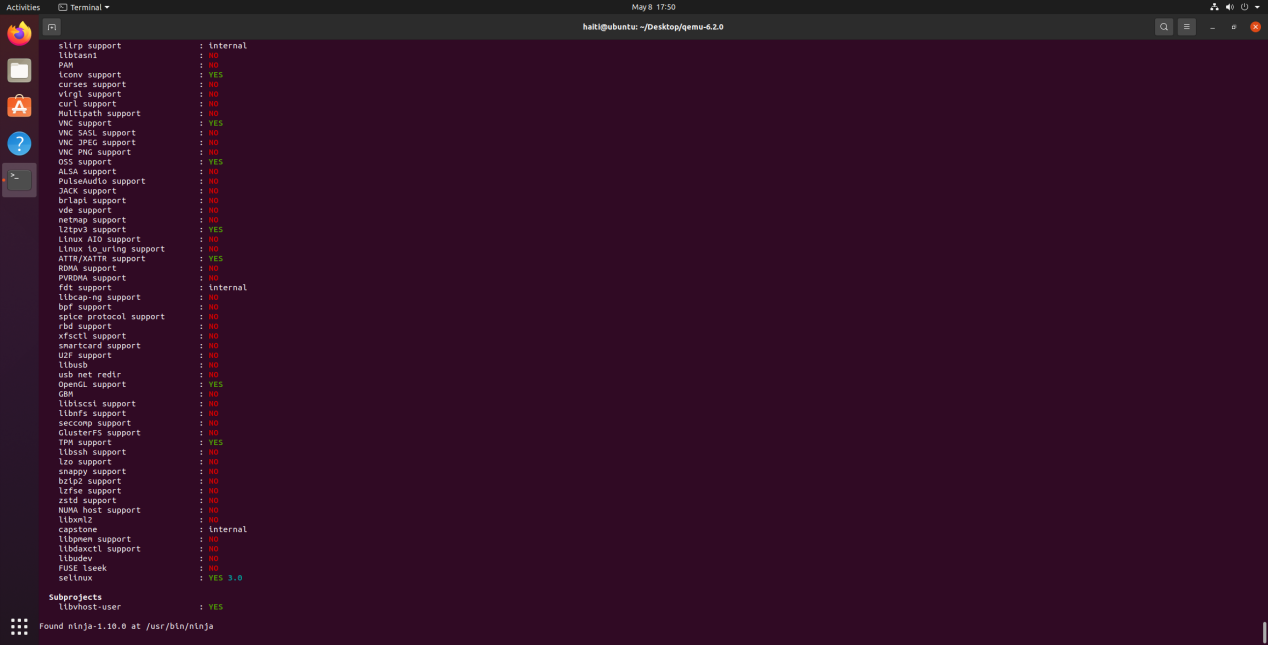






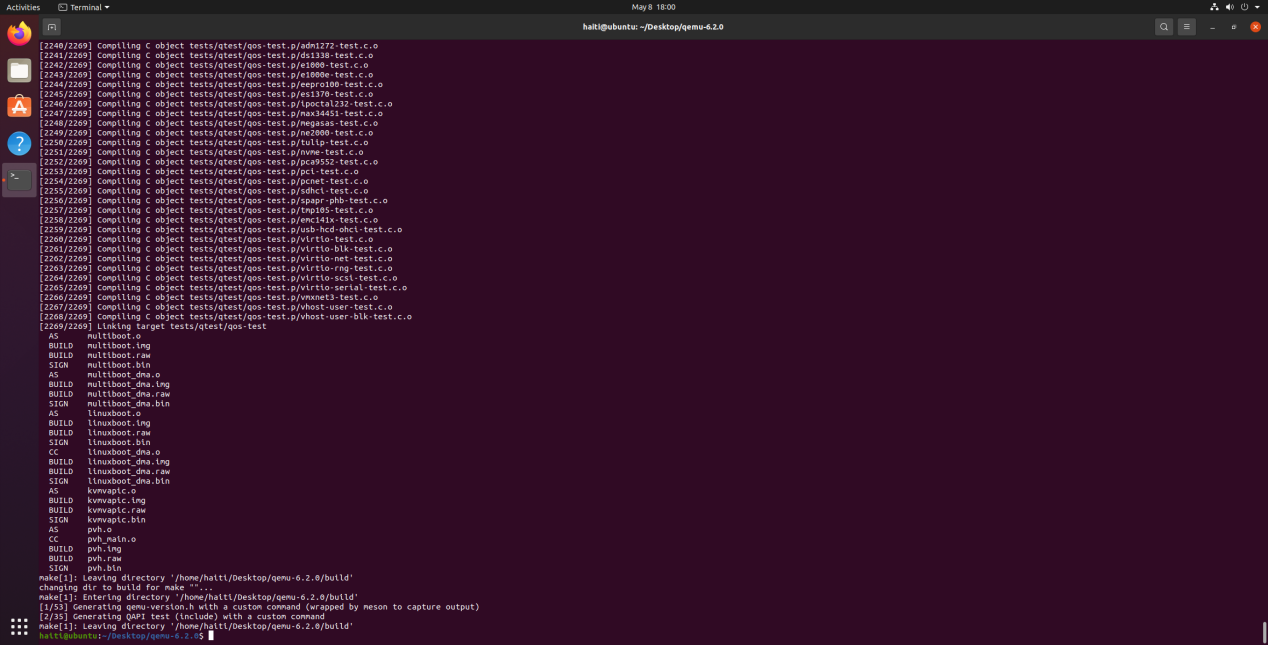


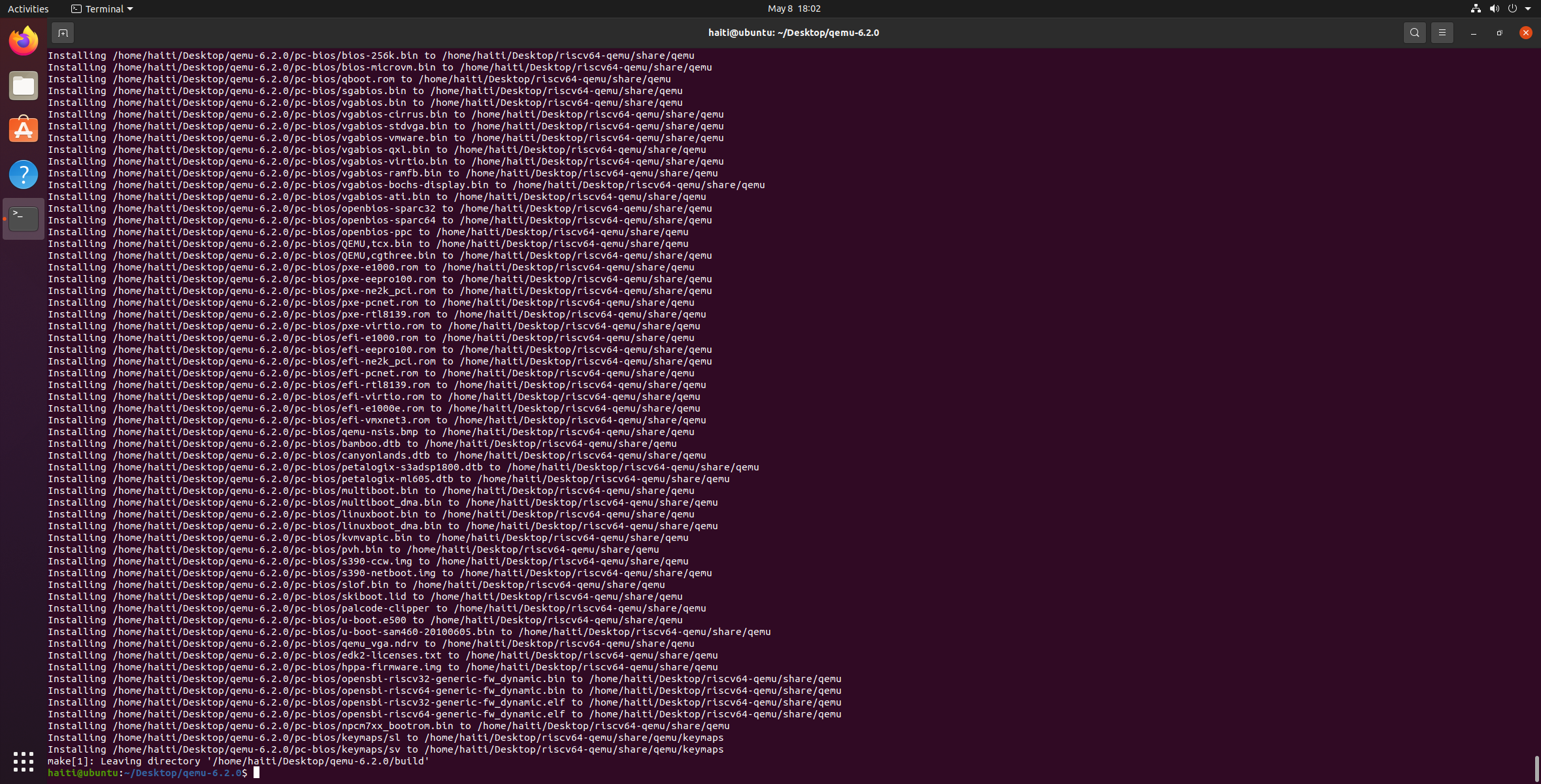
./configure --target-list=riscv64-softmmu,riscv64-linux-user --prefix=/home/haiti/Desktop/riscv64-qemu



$ make

$ make install 如果 --prefix 指定的目录位于根目录下，则需要在 ./configure 前加入 sudo



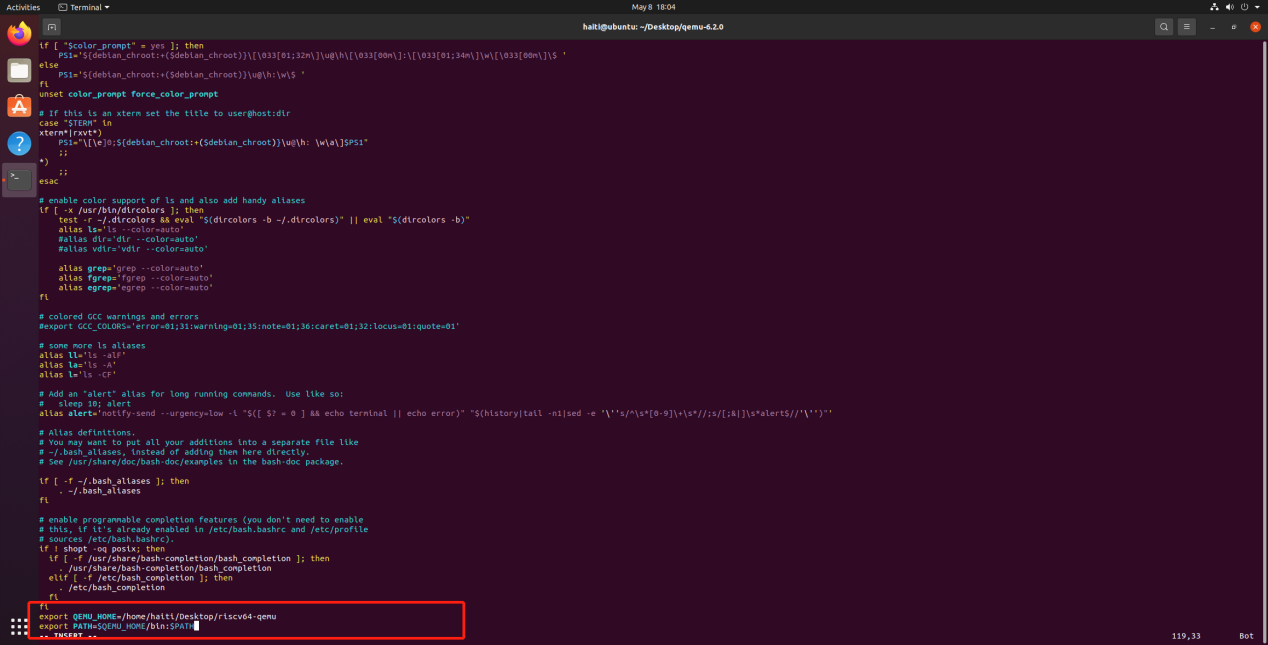


配置环境变量

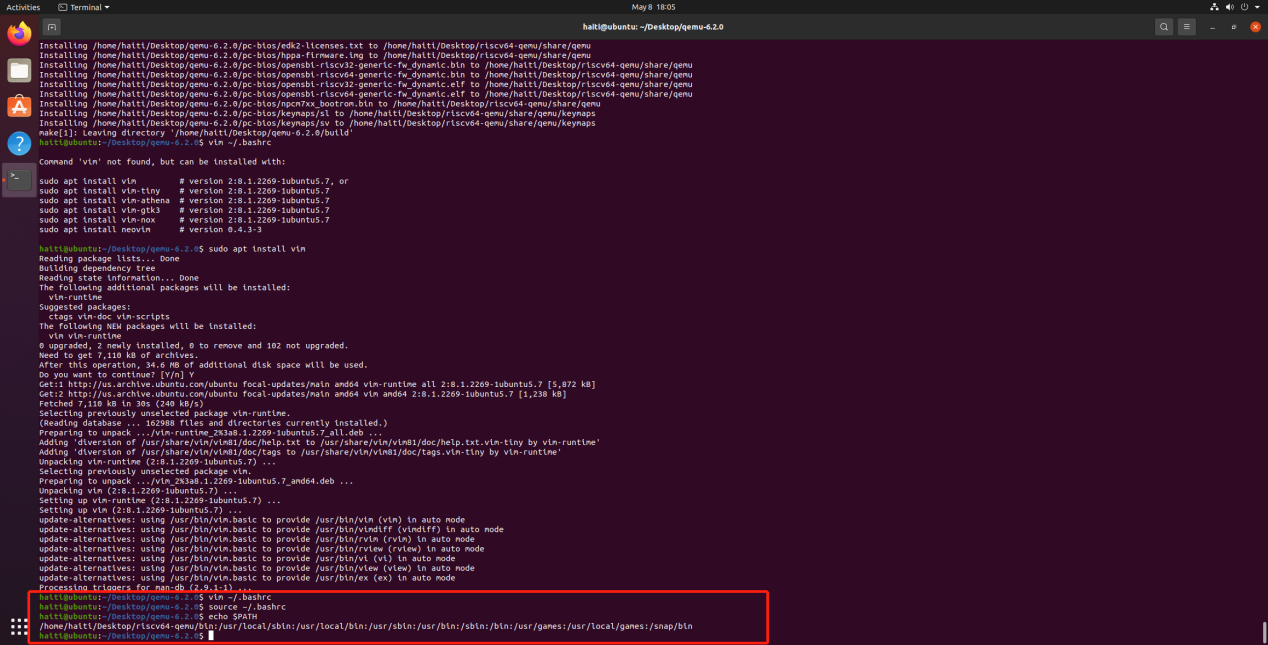
$ vim ~/.bashrc 在文末添加：

export QEMU\_HOME=/home/haiti/Desktop/riscv64-qemu

export PATH=$QEMU\_HOME/bin:$PATH

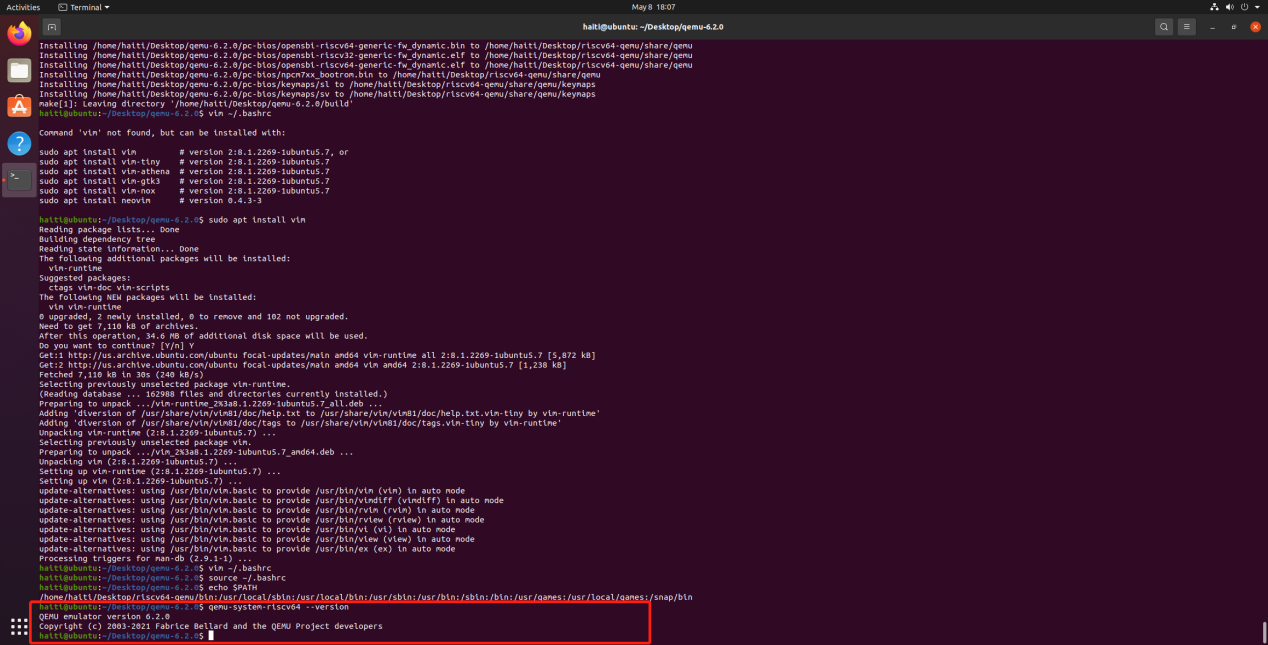


$ source ~/.bashrc $ echo $PATH



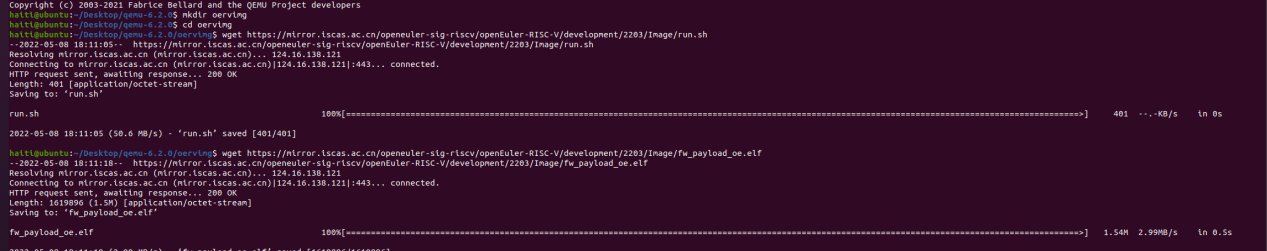
验证安装是否正确

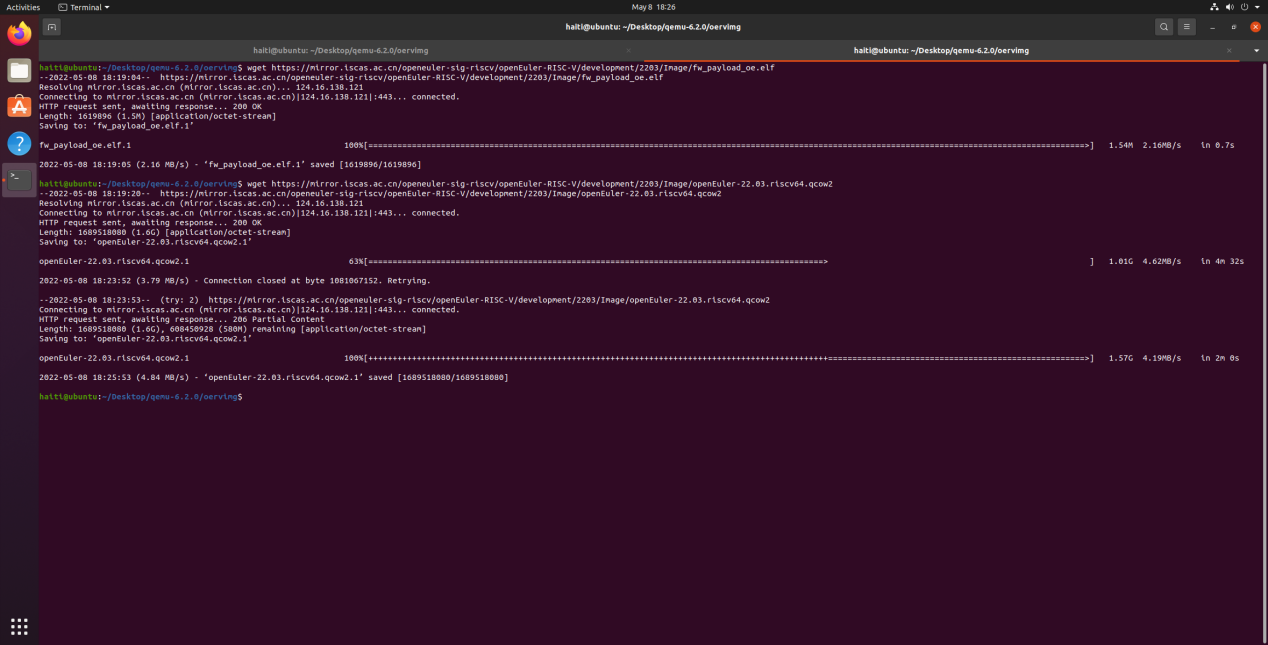
qemu-system-riscv64 --version



镜像安装

mkdir oervimgcd oervimgwget https://mirror.iscas.ac.cn/openeuler-sig-riscv/openEuler-RISC-V/development/2203/Image/run.shwget https://mirror.iscas.ac.cn/openeuler-sig-riscv/openEuler-RISC-V/development/2203/Image/fw\_payload\_oe.elfwget https://mirror.iscas.ac.cn/openeuler-sig-riscv/openEuler-RISC-V/development/2203/Image/openEuler-22.03.riscv64.qcow2





qemu-system-riscv64 \

-nographic -machine virt \

-smp 8 -m 4G \

-kernel fw\_payload\_oe\_docker.elf \

-drive file=openEuler-preview.riscv64.qcow2,format=qcow2,id=hd0 \

-object rng-random,filename=/dev/urandom,id=rng0 \

-device virtio-rng-device,rng=rng0 \

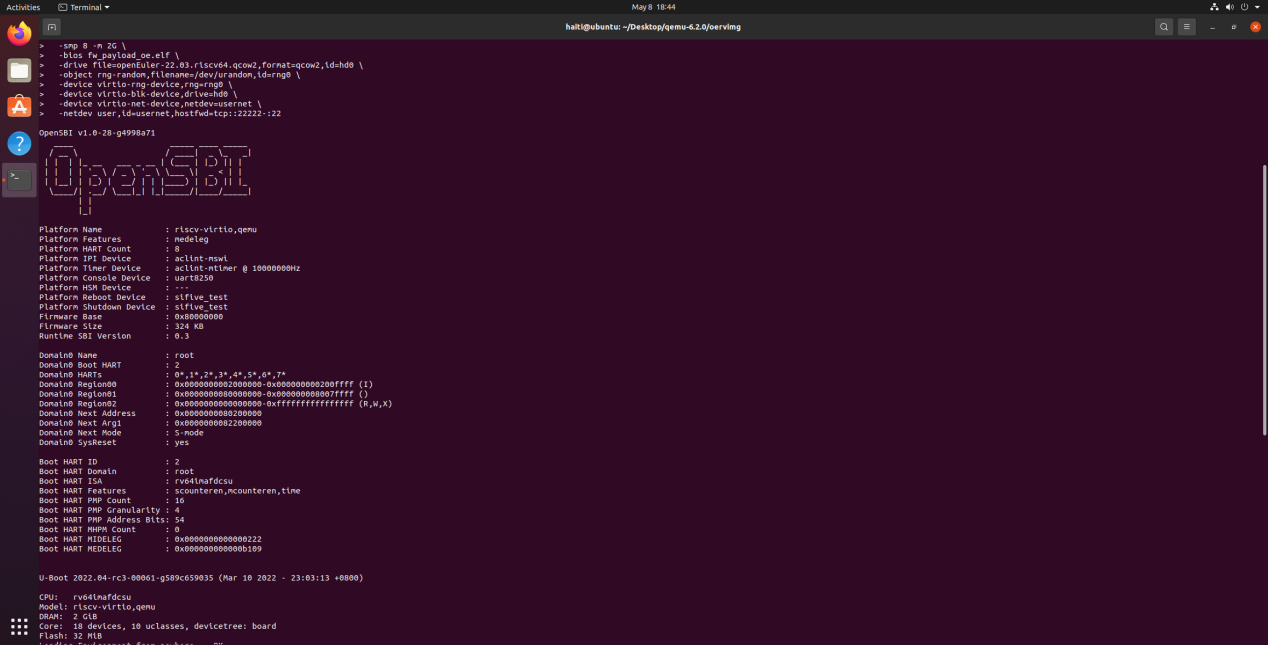
-device virtio-blk-device,drive=hd0 \

-device virtio-net-device,netdev=usernet \

-netdev user,id=usernet,hostfwd=tcp::12055-:22 \

-append 'root=/dev/vda1 rw console=ttyS0 systemd.default\_timeout\_start\_sec=600 selinux=0 highres=off mem=4096M earlycon' \

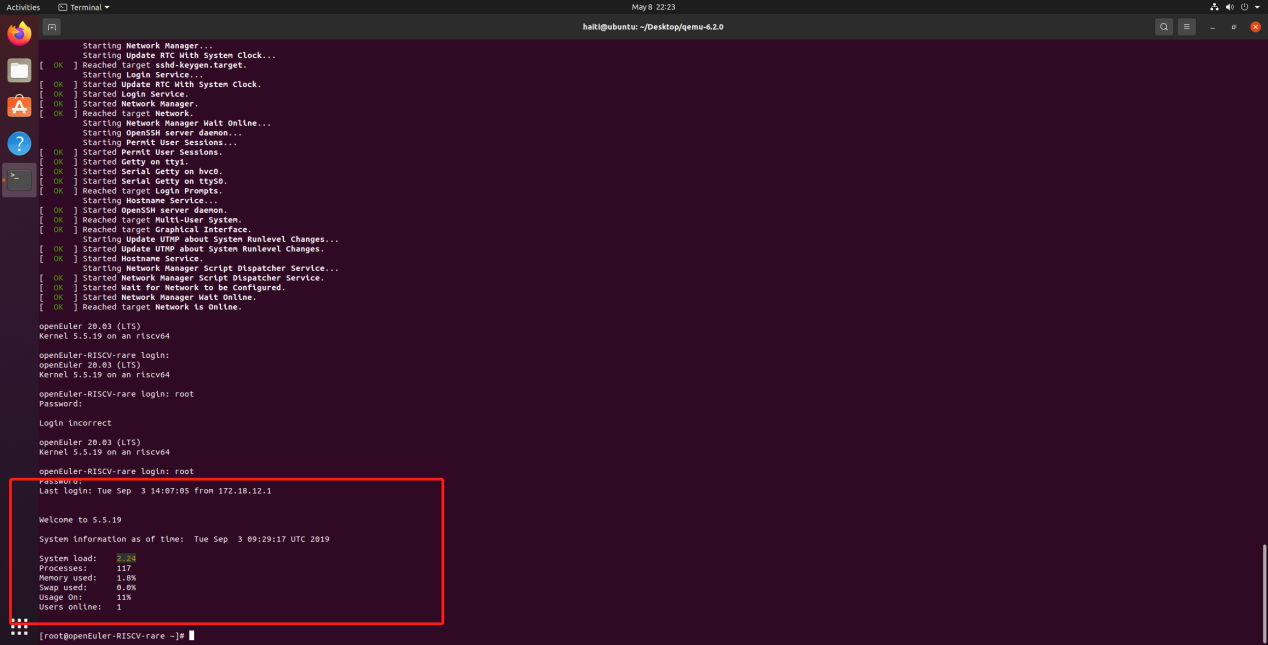
-bios none



欢迎使用

登录用户：root

默认密码：openEuler12#$



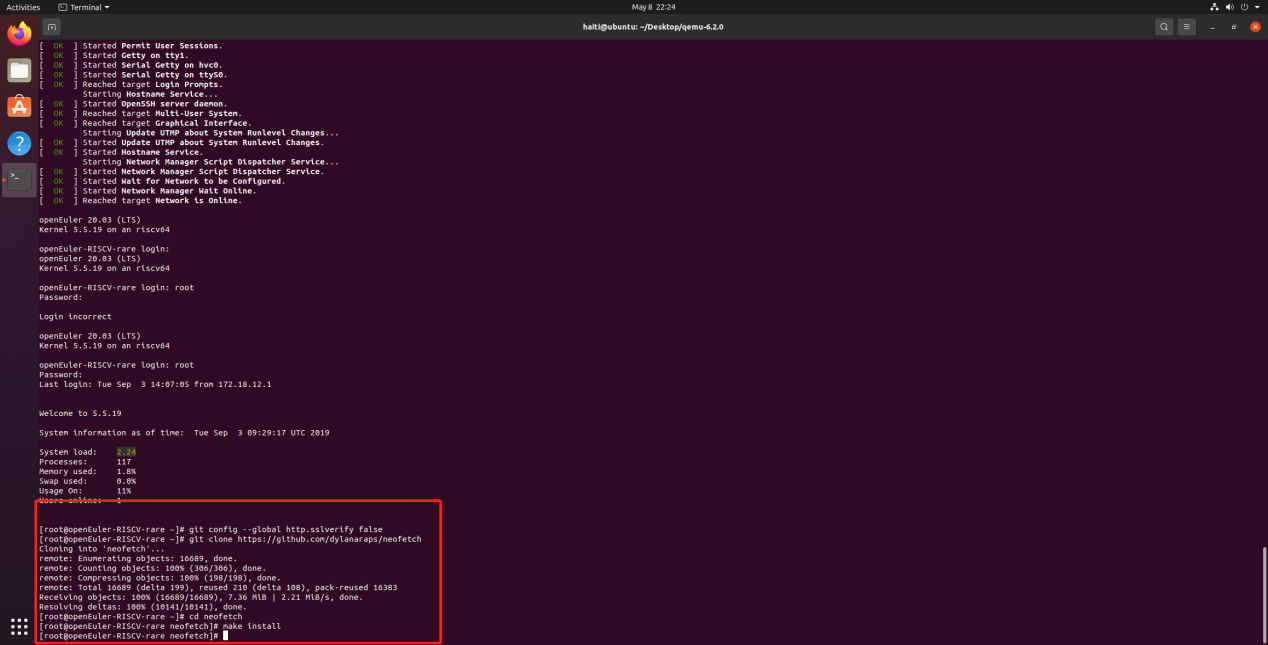
安装neotch

git config --global http.sslverify false

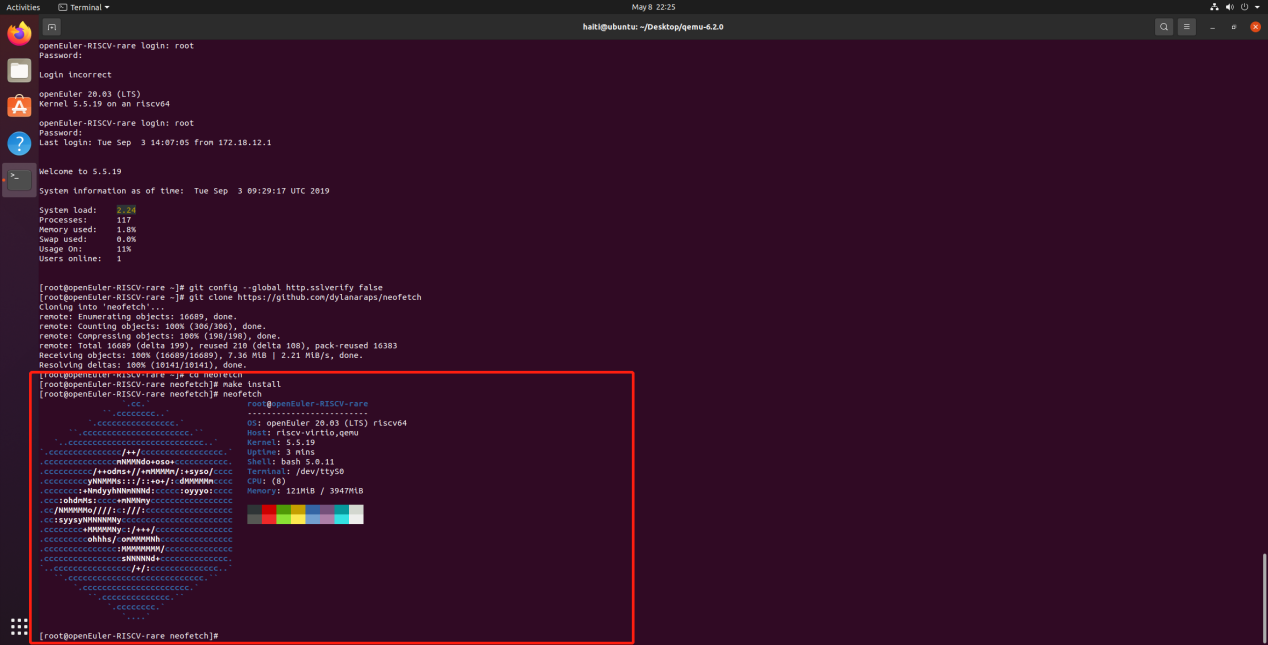
git clone https://github.com/dylanaraps/neofetch

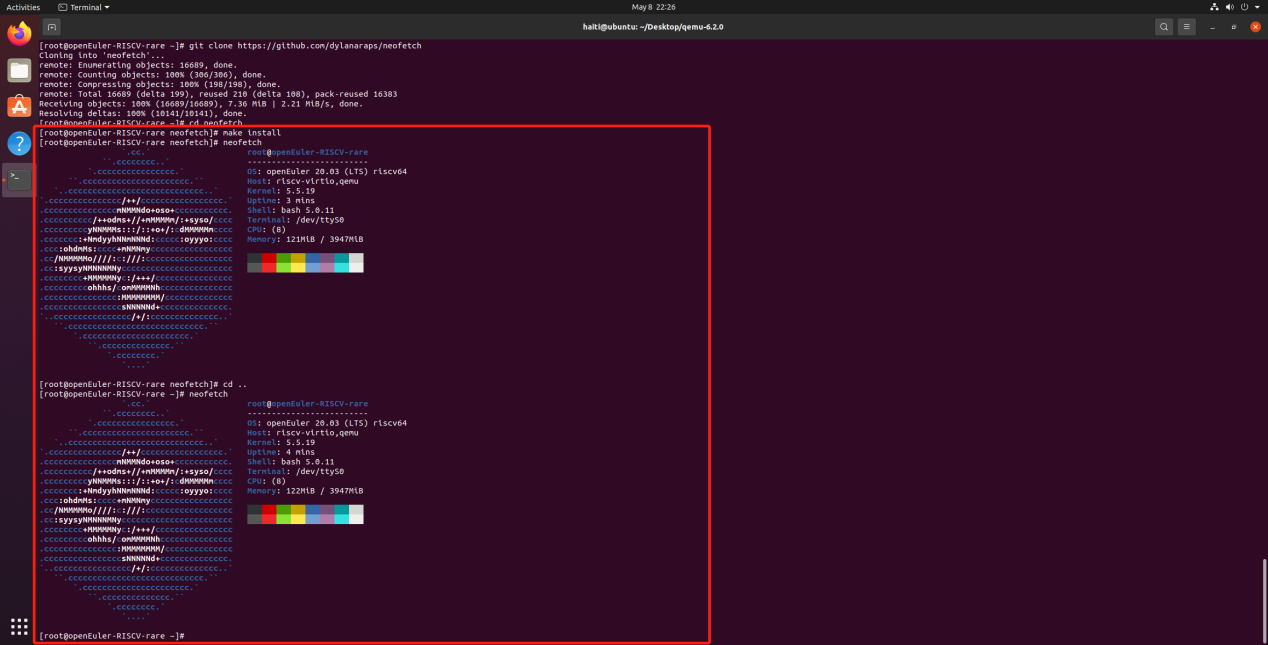
cd neofetch

make install



运行界面：





Docker

增加软件源:

[base1]

Name=base1

baseurl=http://119.3.219.20:82/openEuler:/Mainline:/RISC-V/standard\_riscv64/

enabled=1

gpgcheck=0

[base2]

Name=base2

baseurl=http://121.36.3.168:82/home:/pandora/openEuler\_stage1/

enabled=1

gpgcheck=0

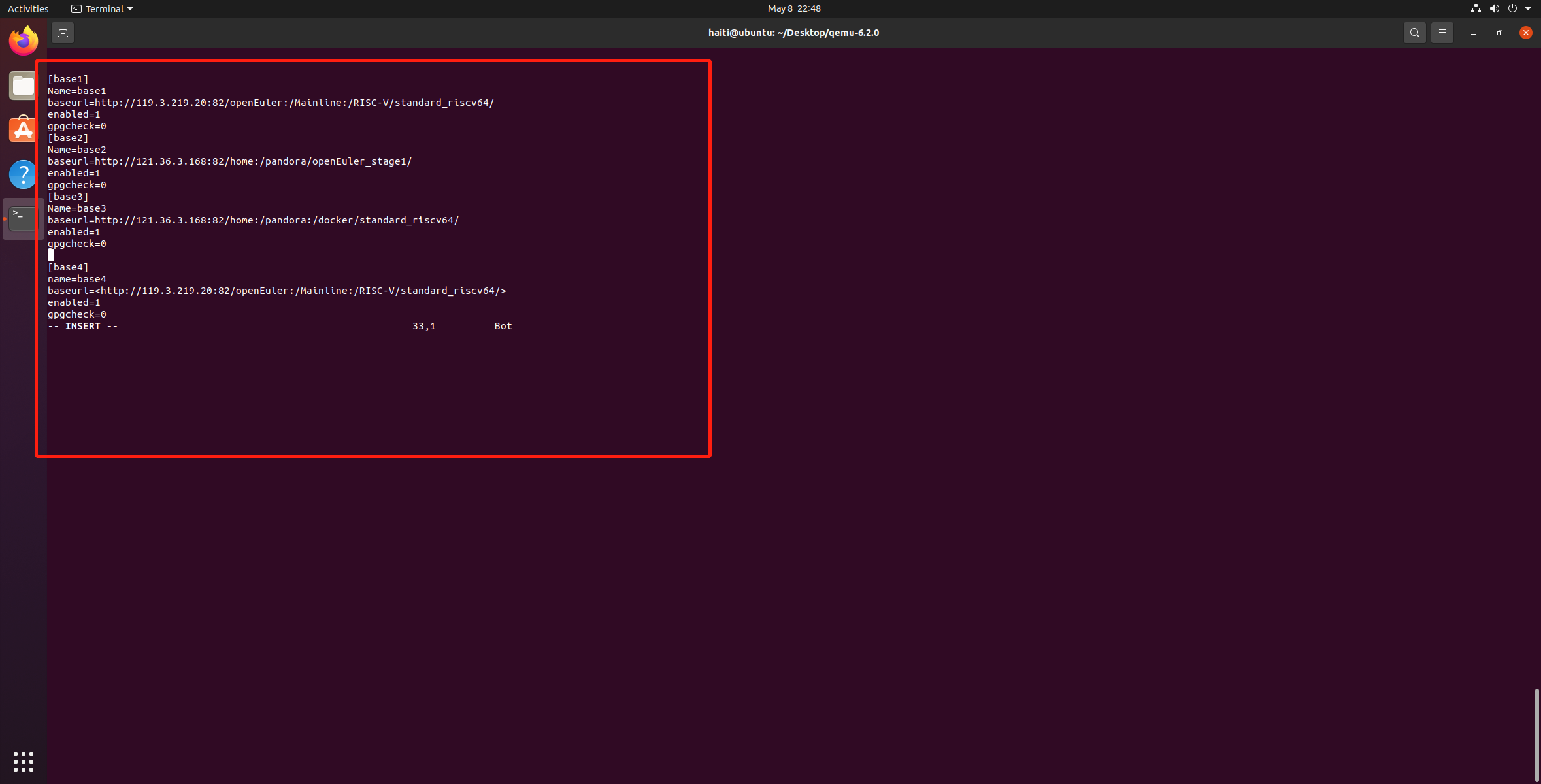
[base3]

Name=base3

baseurl=http://121.36.3.168:82/home:/pandora:/docker/standard\_riscv64/

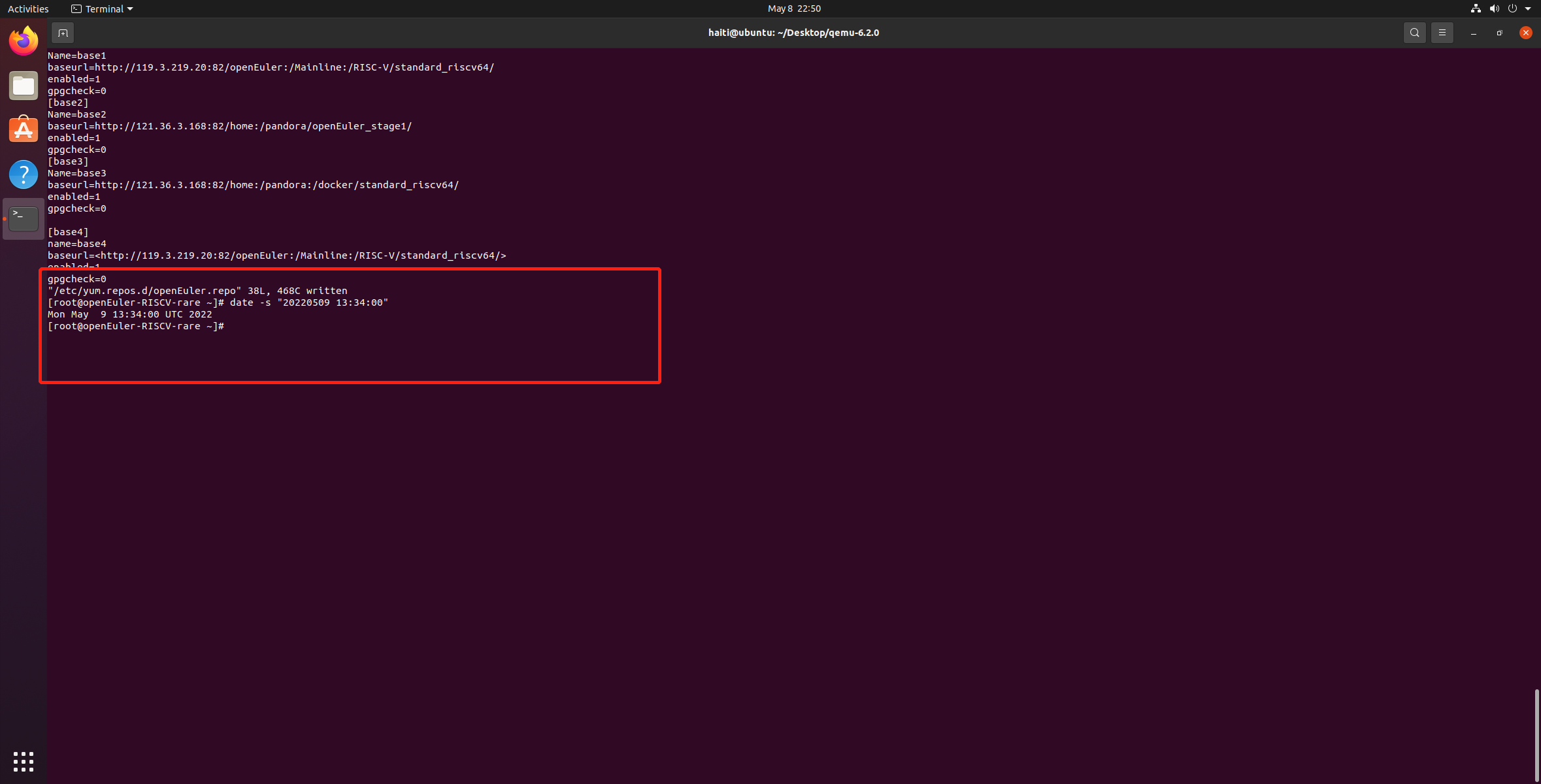
enabled=1

gpgcheck=0



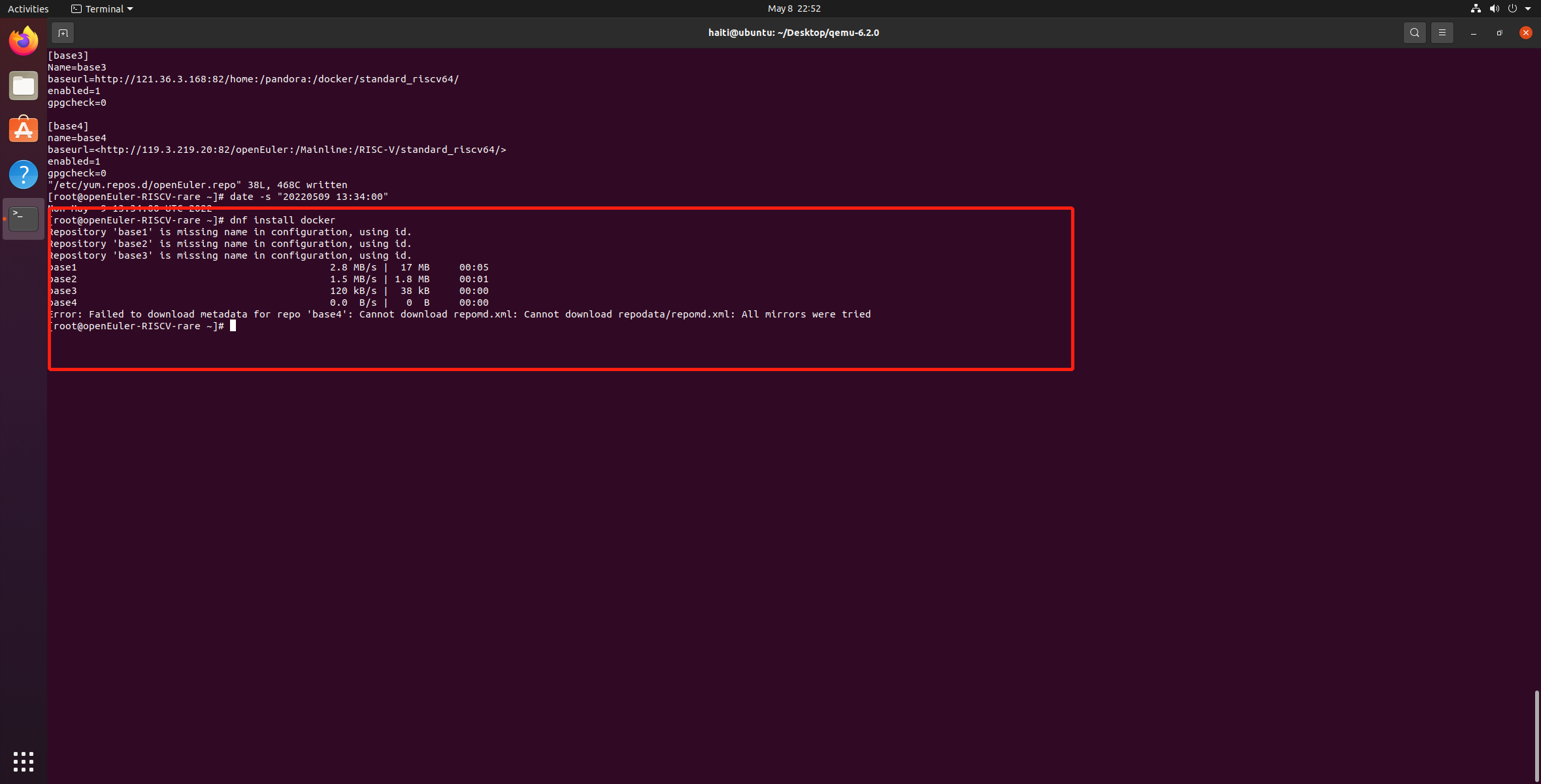
修改时间：

修改日期指令：date -s "20220509 13:34:00"（xx:xx:xx指的是时、分、秒）



用 dnf 安装 docker 软件包：

dnf install docker



测试阶段

1.验证 docker 是否已经成功安装，以文字形式记录输入命令和输出结果。

验证命令：docker -v

拉取 docker 官方仓库提供的 [hello-world image](https://gitee.com/link?target=https://hub.docker.com/_/hello-world) 生成并运行容器，以文字形式记录整个过程的输入命令和输出结果。

docker pull hello-world

docker run hello-world

docker run -it ubuntu bash

其他docker命令测试

d**ocker ps查看当前容器**

**docker image 查看已获取的镜像**

osc的安装

增加 dnf repo，需要的 dnf repo 地址为：

<http://119.3.219.20:82/openEuler:/Mainline:/RISC-V/standard_riscv64/>

①：vim /etc/yum.repos.d/openEuler.repo

②：

[base4]

name=base4

baseurl=<http://119.3.219.20:82/openEuler:/Mainline:/RISC-V/standard\_riscv64/>

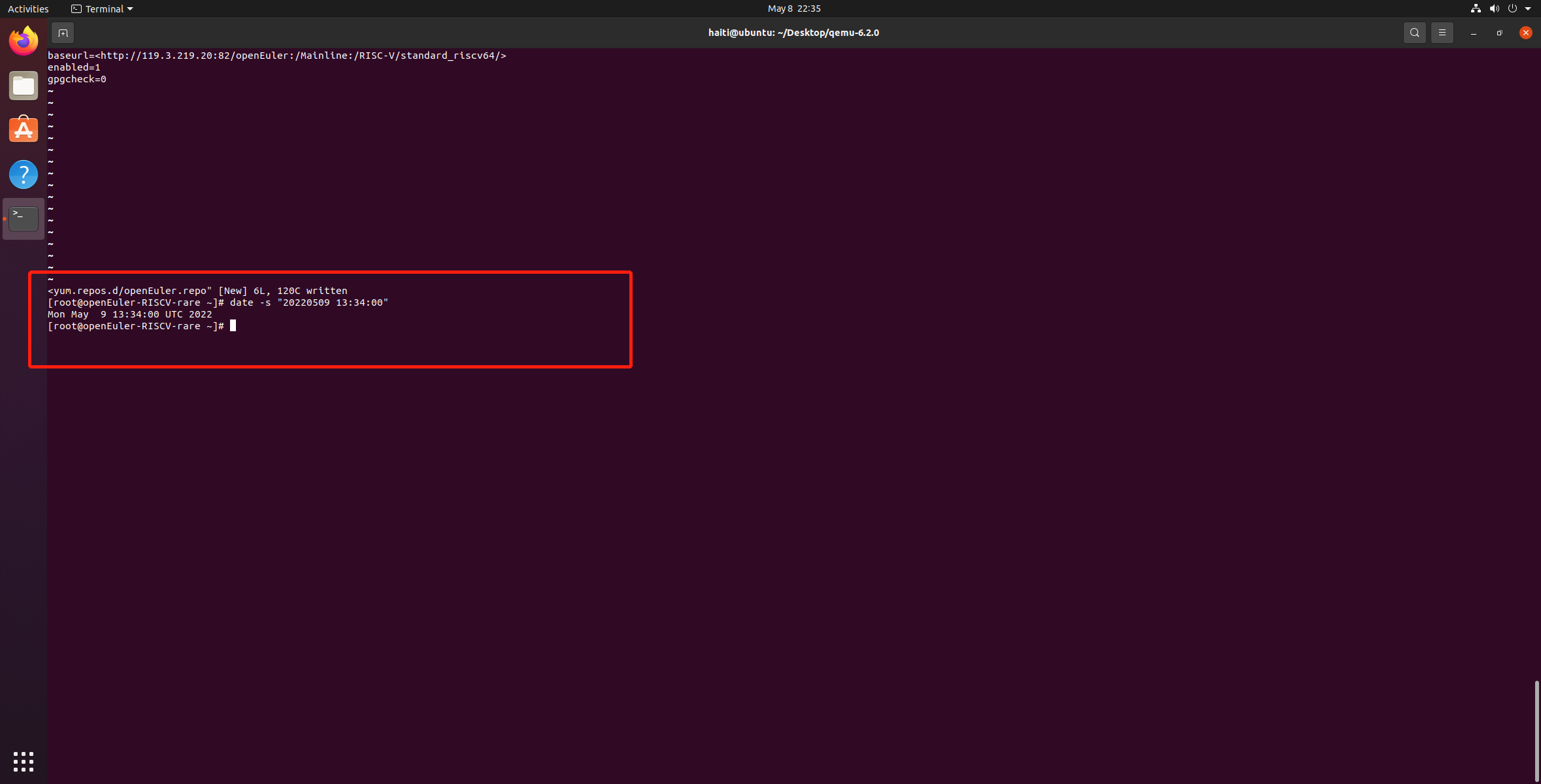
enabled=1

gpgcheck=0

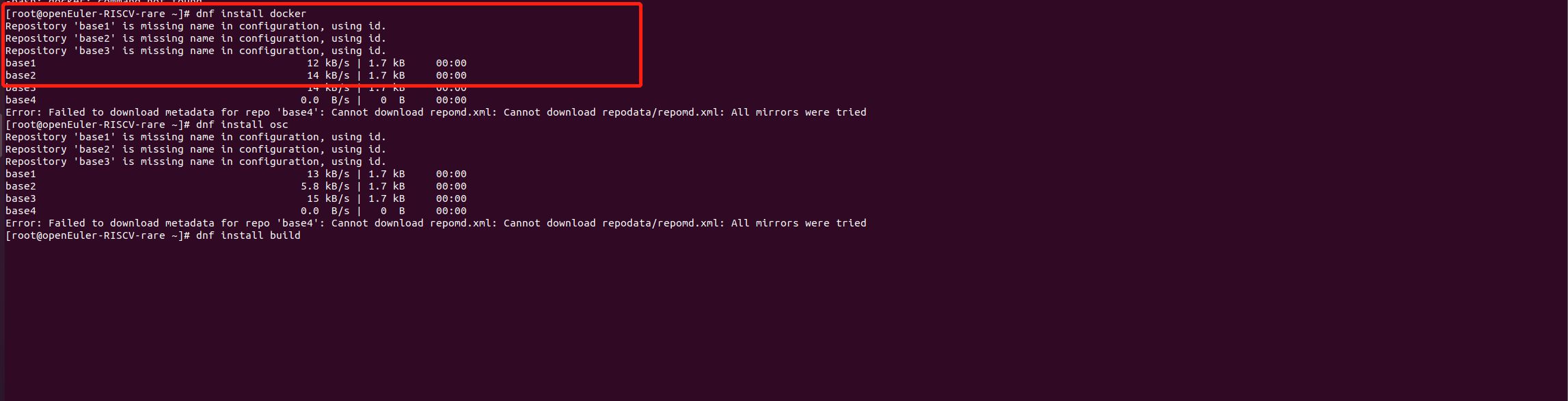


用 dnf 安装 osc 软件包、 build 软件包和 sudo 软件包，使用 dnf 以前需要用 date 命令设置正确的时间。

①修改日期指令：date -s "20220509 13:34:00"（xx:xx:xx指的是时、分、秒）

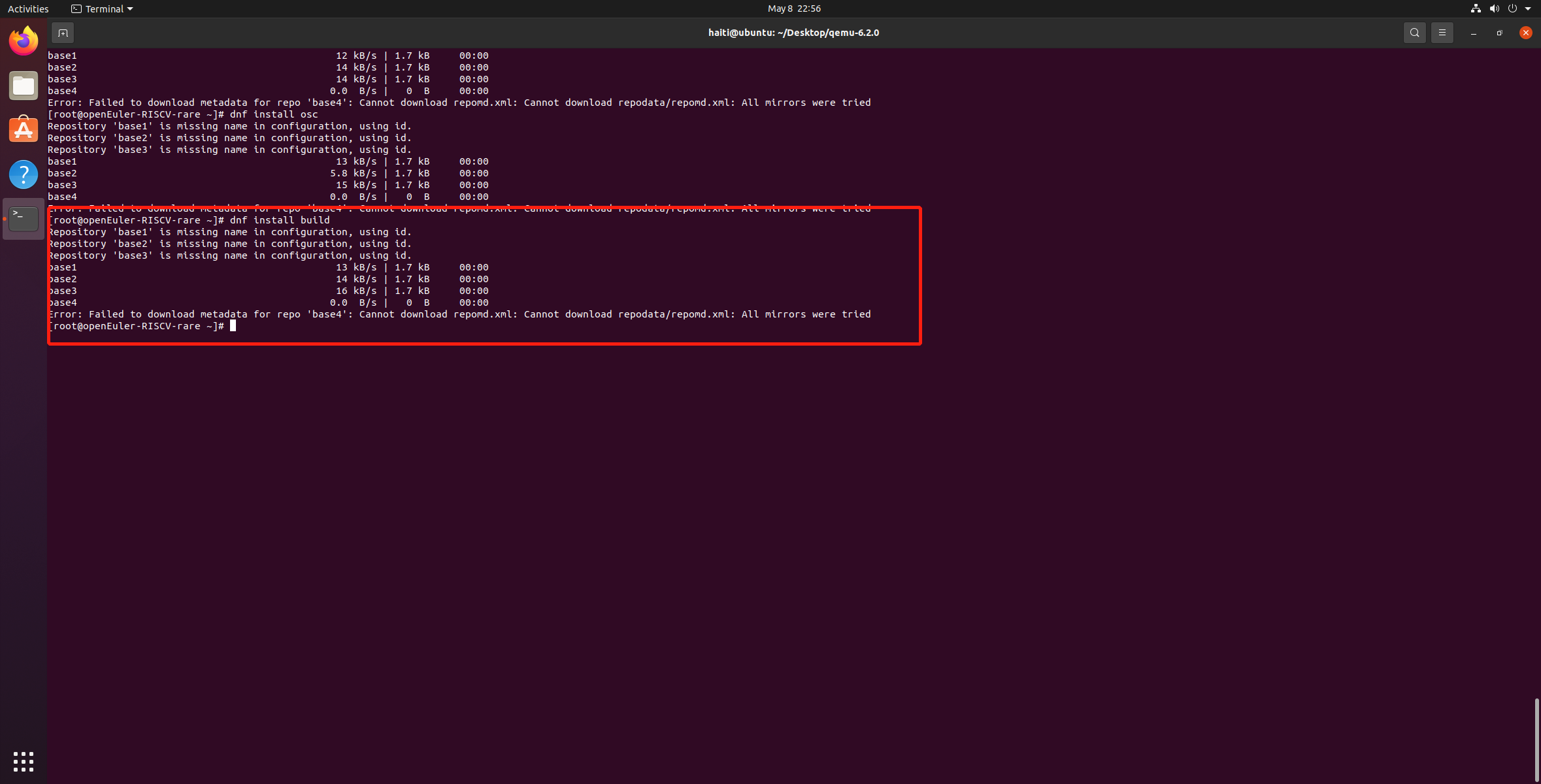
安装包指令：

dnf install osc报错：镜像丢失



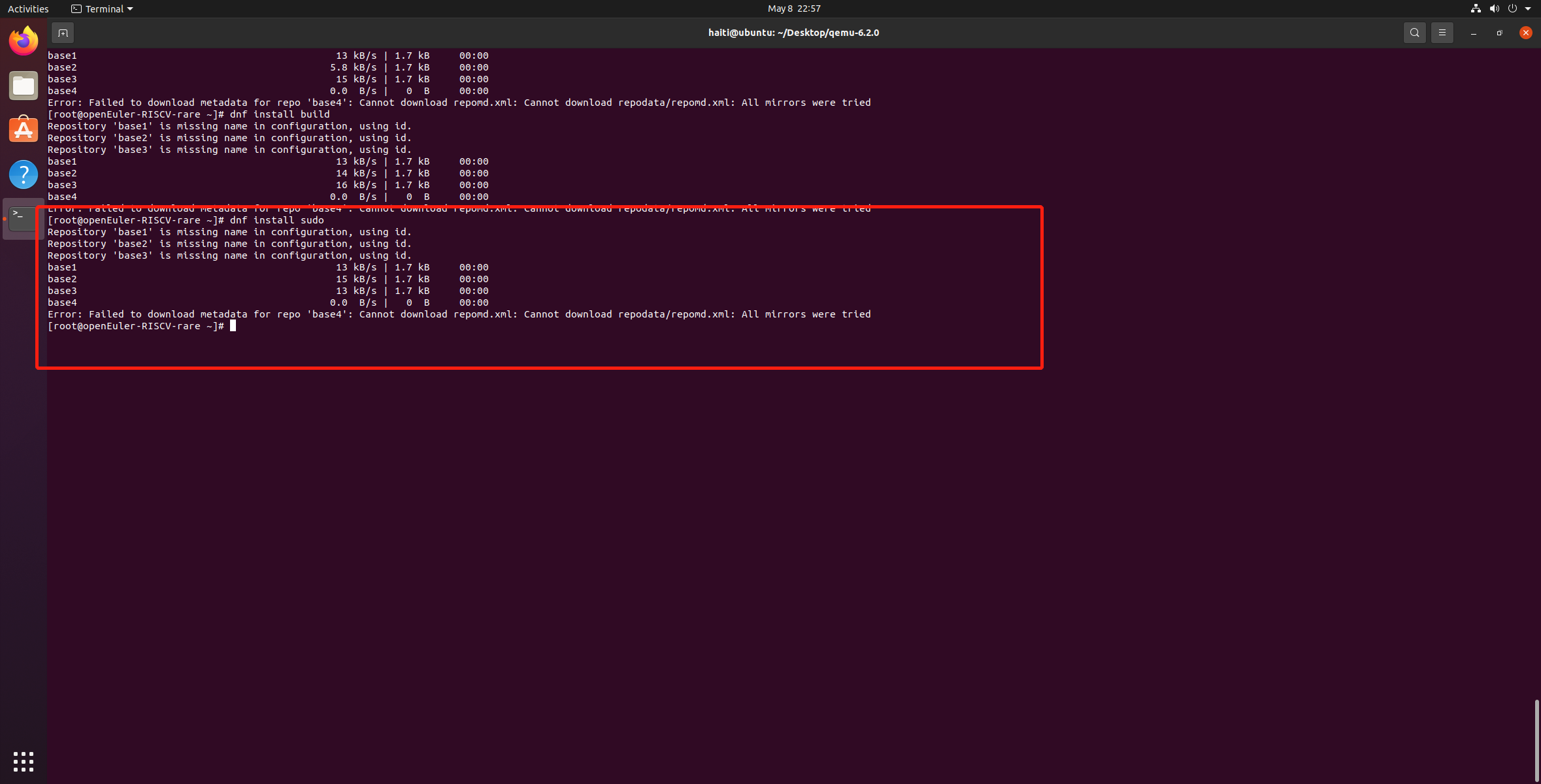
报错：镜像丢失

dnf install build



dnf install sudo

报错：镜像丢失



4.去 https://build.openeuler.org/ 注册一个账号。

复制网址，进行注册。

5.使用 osc co openEuler:Mainline:RISC-V/aalib 将 aalib 包的 \_service 文件下载下来。第一次使用，会提示输入用户名和密码，这里填写上面注册的用户名和密码。最后会失败，需要在 ~/.config/osc/oscrc 配置文件中把 apiurl 改成 https://build.openeuler.org/，no\_verify = 1 取消注释，然后把最下面 [https://api.opensuse.org] 改为 [https://build.openeuler.org]。

①osc co openEuler:Mainline:RISC-V/aalib，输入4中已注册的账号和密码。

略

②vim ~/.config/osc/oscrc进入文件，修改apiurl 改成 https://build.openeuler.org/，no\_verify = 1 取消注释，然后把最下面 [https://api.opensuse.org] 改为 [https://build.openeuler.org]。

即：

[general]

apiurl = https://build.openeuler.org/

no\_verify = 1

[https://build.openeuler.org/]

user=

pass=

（安装后进入文件查看的，开始时忘记截图了）

**测试阶段**

1. 再次运行 osc co openEuler:Mainline:RISC-V/aalib，以文字形式记录输入命令和输出结果。

命令：osc co openEuler:Mainline:RISC-V/aalib

1. 进入 openEuler:Mainline:RISC-V/aalib 文件夹，运行 osc up -S 命令，下载 aalib 包的源代码，以文字形式记录输入命令和输出结果。

命令：cd osc co openEuler:Mainline:RISC-V/aalib

命令：osc up -S

1. 运行 rm -f \_service;for file in `ls | grep -v .osc`;do new\_file=${file##\*:};mv $file $new\_file;done，然后运行 osc build 构建 aalib 包，记录最后几行构建成功的信息。

命令：rm -f \_service;for file in `ls | grep -v .osc`;do new\_file=${file##\*:};mv $file $new\_file;done

命令：osc build

1. 以上只是 osc 的基础使用，测试者可以参考 [https://en.opensuse.org/openSUSE:OSC](https://gitee.com/link?target=https://en.opensuse.org/openSUSE:OSC) 运行更多 osc 命令，并以文字形式记录整个过程的输入命令和输出结果。

命令：osc list

命令：osc prjresults

命令：osc status