实验0 - 配置环境

所用设备及系统：Macbook Pro M2 Max, MacOS Ventura 13.5.2

因为我有魔法，所以省略了各种镜像源配置相关步骤🥳

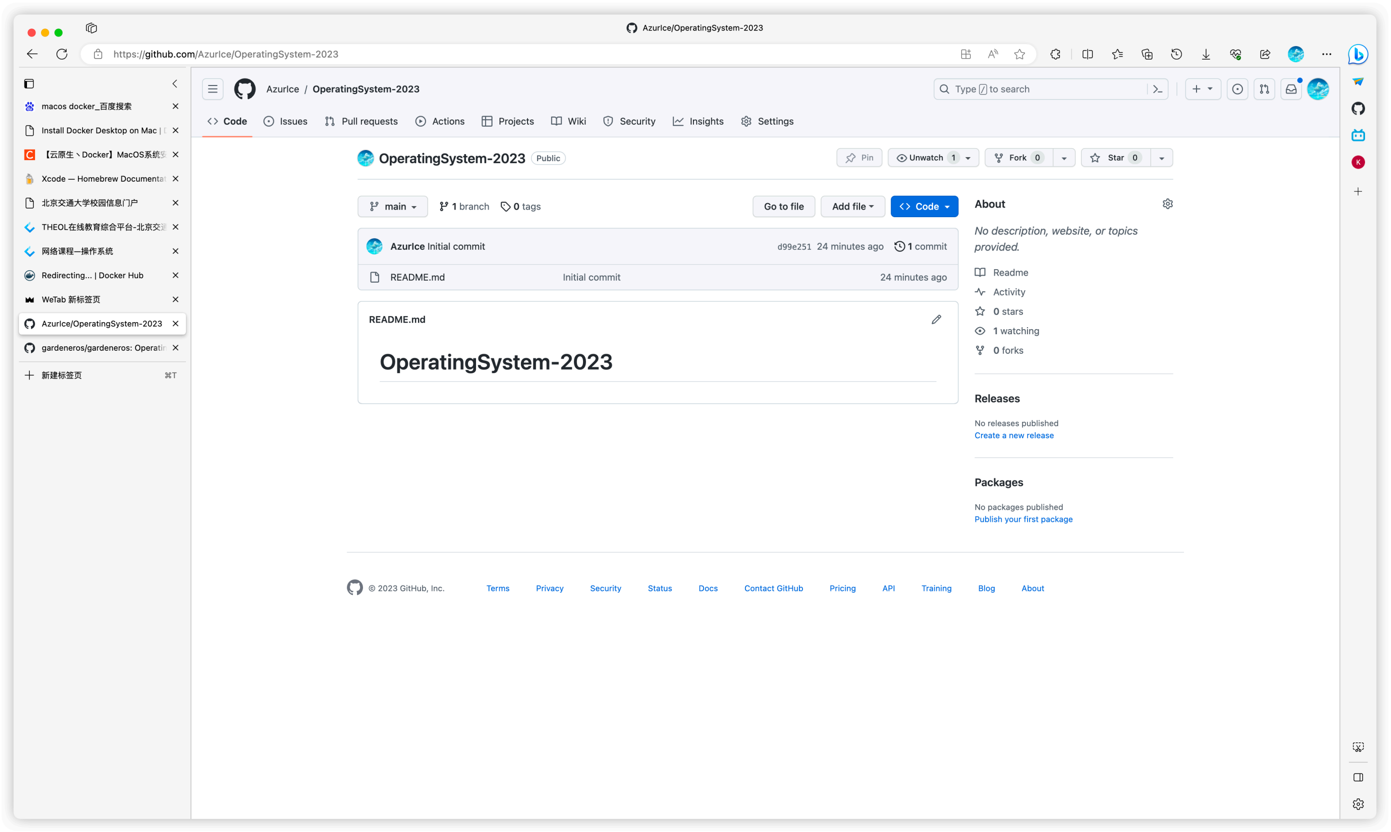
一、安装 docker

https://www.docker.com/

二、创建容器及项目

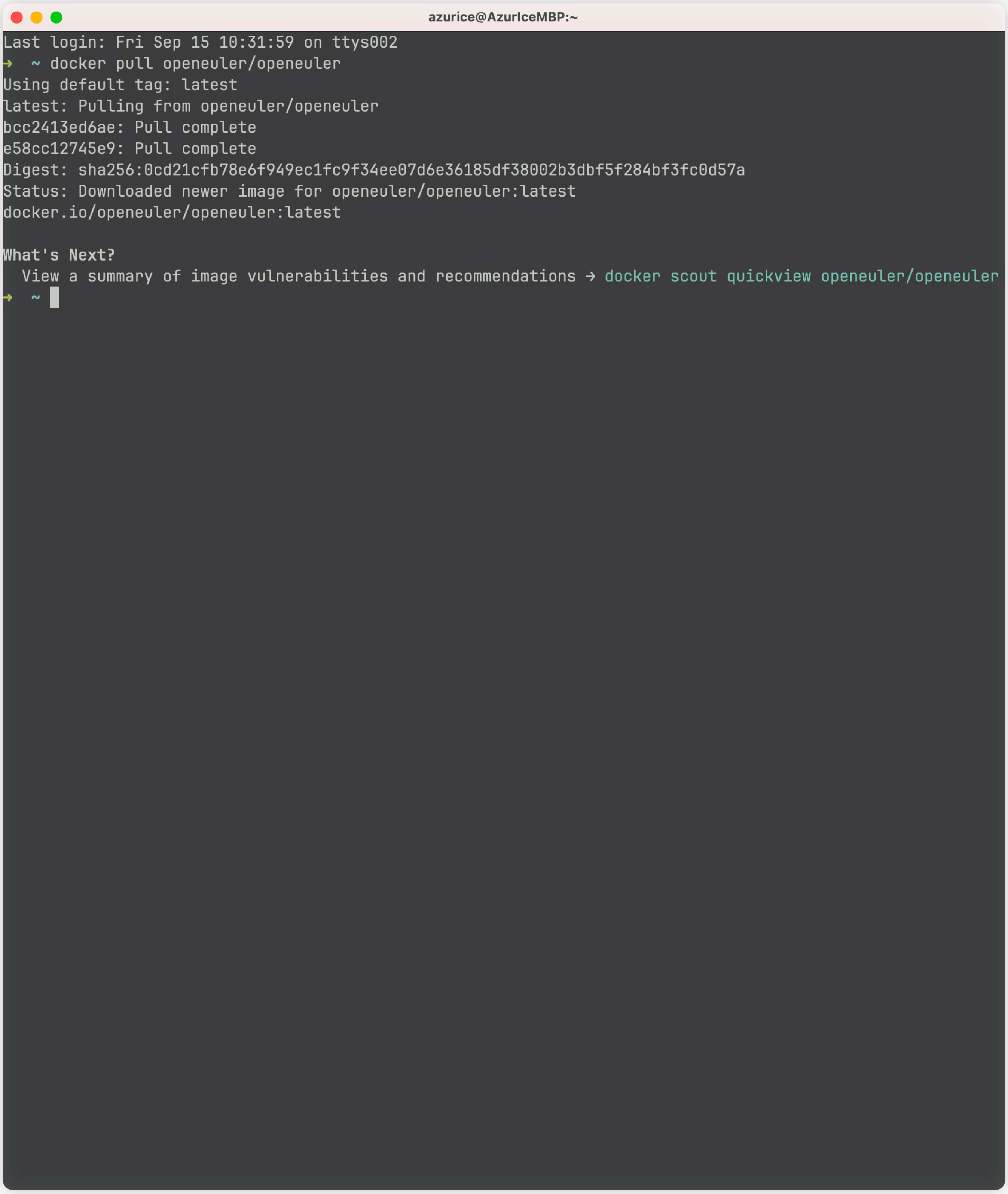
https://github.com/AzurIce/OperatingSystem-2023

首先创建一个项目：



获取 openeuler 的 docker 镜像：

```
docker pull openeuler/openeuler
```



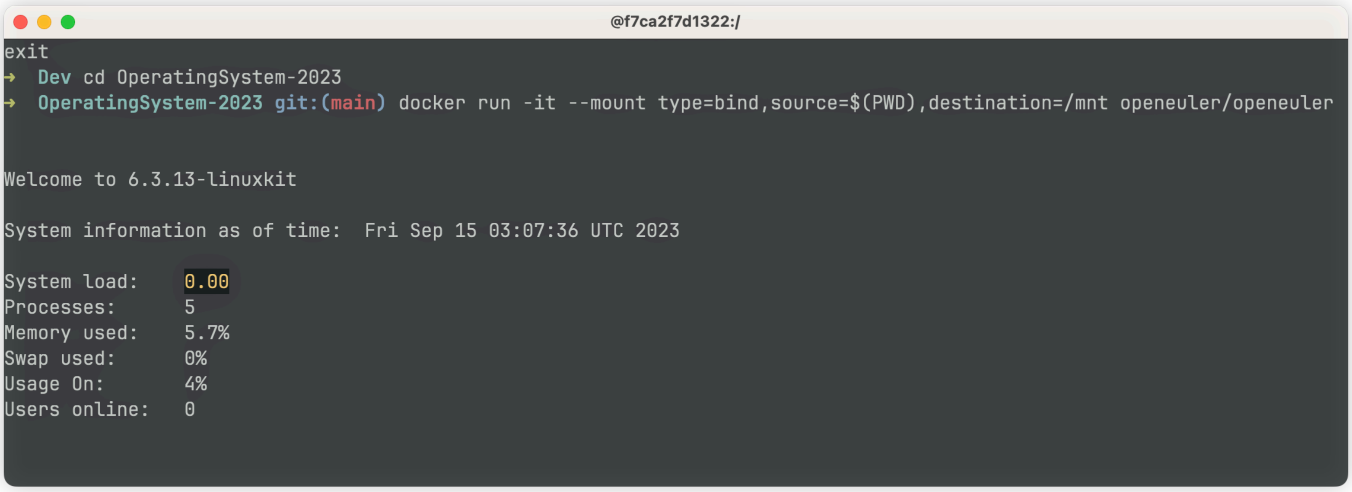
azurice@AzurIceMBP:~

```
Last login: Fri Sep 15 10:31:59 on ttys002
➜  ~ docker pull openeuler/openeuler
Using default tag: latest
latest: Pulling from openeuler/openeuler
bcc2413ed6ae: Pull complete
e58cc12745e9: Pull complete
Digest: sha256:0cd21cfb78e6f949ec1fc9f34ee07d6e36185df38002b3dbf5f284bf3fc0d57a
Status: Downloaded newer image for openeuler/openeuler:latest
docker.io/openeuler/openeuler:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview openeuler/openeuler
➜  ~ 
```

进入到项目目录，创建容器并启动：

```
git clone git@github.com:AzurIce/OperatingSystem-2023.git
cd OperatingSystem-2023
docker run -it --mount type=bind,source=$(PWD),destination=/mnt openeuler/openeuler
```



```
exit
➜  Dev cd OperatingSystem-2023
➜  OperatingSystem-2023 git:(main) docker run -it --mount type=bind,source=$(PWD),destination=/mnt openeuler/openeuler


Welcome to 6.3.13-linuxkit

System information as of time:  Fri Sep 15 03:07:36 UTC 2023

System load:    0.00
Processes:      5
Memory used:    5.7%
Swap used:      0%
Usage On:       4%
Users online:   0
```

然后便进入了 openeuler 环境，并可以通过 `/mnt` 目录访问到项目文件夹。

三、开发环境配置

1. 必要软件

```
dnf install curl vim gcc git
```

2. Rust 开发环境

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

> 可以直接在这一步选择 nightly 版本，或之后再通过下面命令设置：
>
> ```
> rustup install nightly
> rustup default nightly
> ```

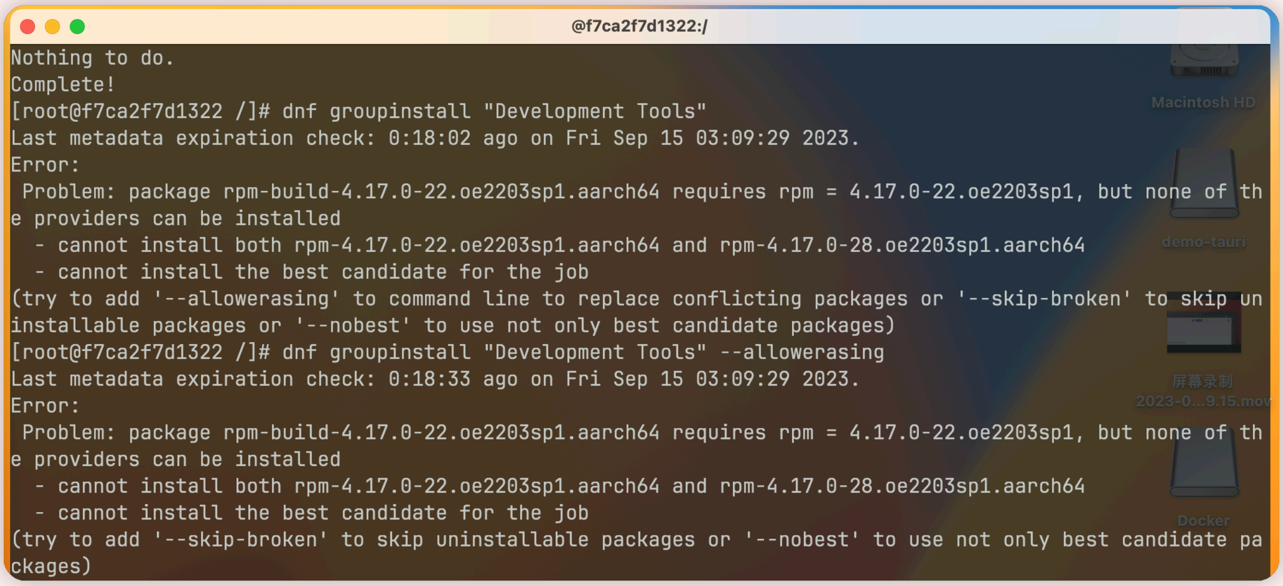
```
rustup target add riscv64gc-unknown-none-elf
cargo install cargo-binutils
rustup component add llvm-tools-preview
rustup component add rust-src
```

然后在 项目目录下心间 `rust-toolchain` 文件，写入 `nightly-2022-10-19` 来固定我们到时候将要使用的 rust 版本

3. 安装一些基本的软件包

```
dnf groupinstall "Development Tools"
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
```

坑1:



```
Nothing to do.
Complete!
[root@f7ca2f7d1322 /]# dnf groupinstall "Development Tools"
Last metadata expiration check: 0:18:02 ago on Fri Sep 15 03:09:29 2023.
Error:
 Problem: package rpm-build-4.17.0-22.oe2203sp1.aarch64 requires rpm = 4.17.0-22.oe2203sp1, but none of th
e providers can be installed
  - cannot install both rpm-4.17.0-22.oe2203sp1.aarch64 and rpm-4.17.0-28.oe2203sp1.aarch64
  - cannot install the best candidate for the job
(try to add '--allowerasing' to command line to replace conflicting packages or '--skip-broken' to skip un
installable packages or '--nobest' to use not only best candidate packages)
[root@f7ca2f7d1322 /]# dnf groupinstall "Development Tools" --allowerasing
Last metadata expiration check: 0:18:33 ago on Fri Sep 15 03:09:29 2023.
Error:
 Problem: package rpm-build-4.17.0-22.oe2203sp1.aarch64 requires rpm = 4.17.0-22.oe2203sp1, but none of th
e providers can be installed
  - cannot install both rpm-4.17.0-22.oe2203sp1.aarch64 and rpm-4.17.0-28.oe2203sp1.aarch64
  - cannot install the best candidate for the job
(try to add '--skip-broken' to skip uninstallable packages or '--nobest' to use not only best candidate pa
ckages)
```

先执行一次 `dnf distro-sync` 即可

4. 从源码安装 qemu

```
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。

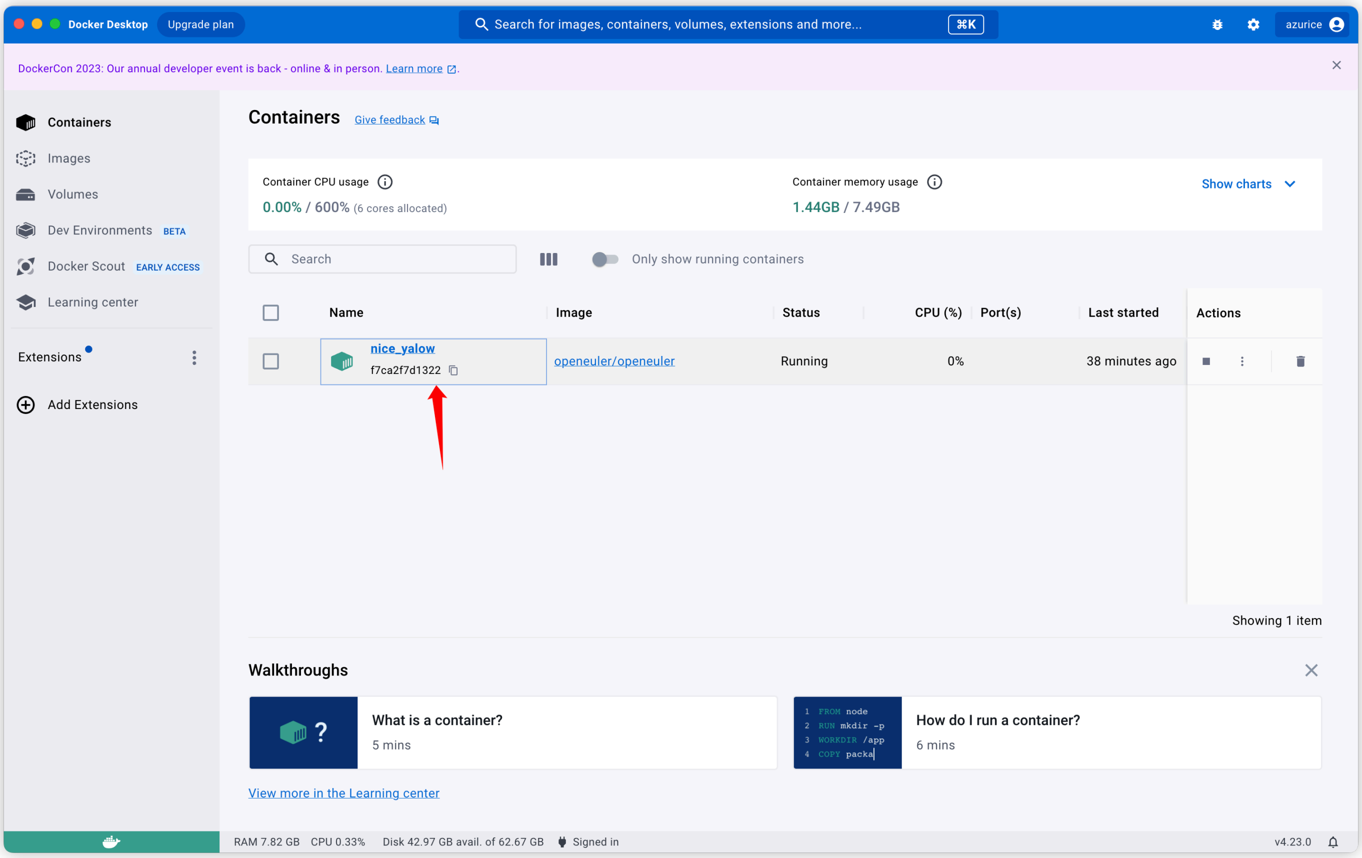
```
qemu-system-riscv64 --version
qemu-riscv64 --version
```

四、保存配置到 docker 镜像中

在docker外（自己的操作系统中）的终端内运行：

```
docker commit -m "Configured environment" -a "AzurIce" f7ca2f7d1322077670897839a7a68e5954d5530338117fac026abc6395003405 my_openeuler
```

那一大长串hash字符串来源于这里：



DockerCon 2023: Our annual developer event is back - online & in person. Learn more.

Containers

Container CPU usage 0.00% / 600% (6 cores allocated)

Container memory usage 1.44GB / 7.49GB

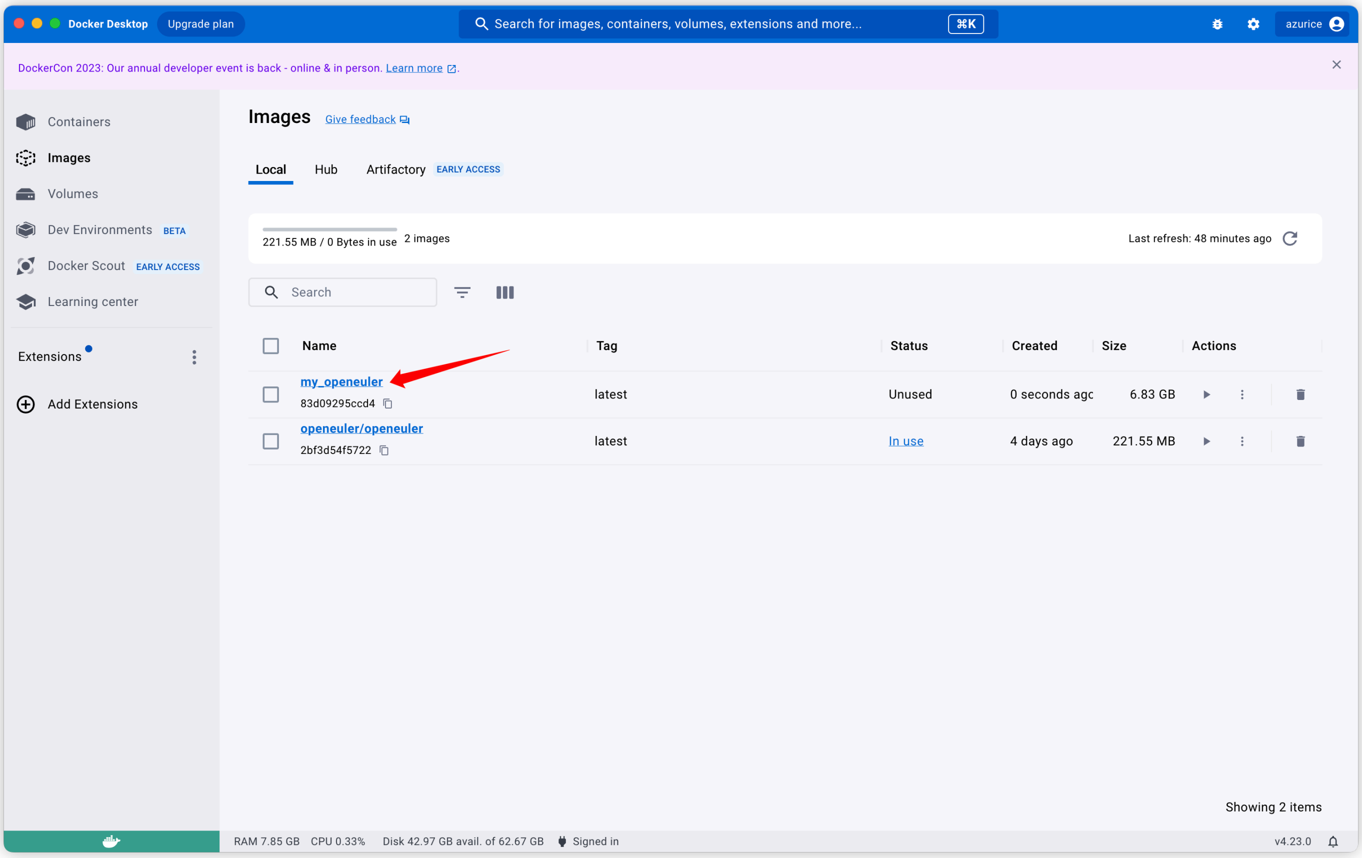
Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
nice_yalow f7ca2f7d1322	openeuler/openeuler	Running	0%		38 minutes ago	

What is a container? 5 mins

How do I run a container? 6 mins

RAM 7.82 GB CPU 0.33% Disk 42.97 GB avail. of 62.67 GB Signed in v4.23.0

然后可以在 Images 中看到我们刚刚创建的镜像：



DockerCon 2023: Our annual developer event is back - online & in person. Learn more.

221.55 MB / 0 Bytes in use 2 images

Last refresh: 48 minutes ago

Name	Tag	Status	Created	Size	Actions
my_openeuler 83d09295ccd4	latest	Unused	0 seconds agc	6.83 GB	
openeuler/openeuler 2bf3d54f5722	latest	In use	4 days ago	221.55 MB	

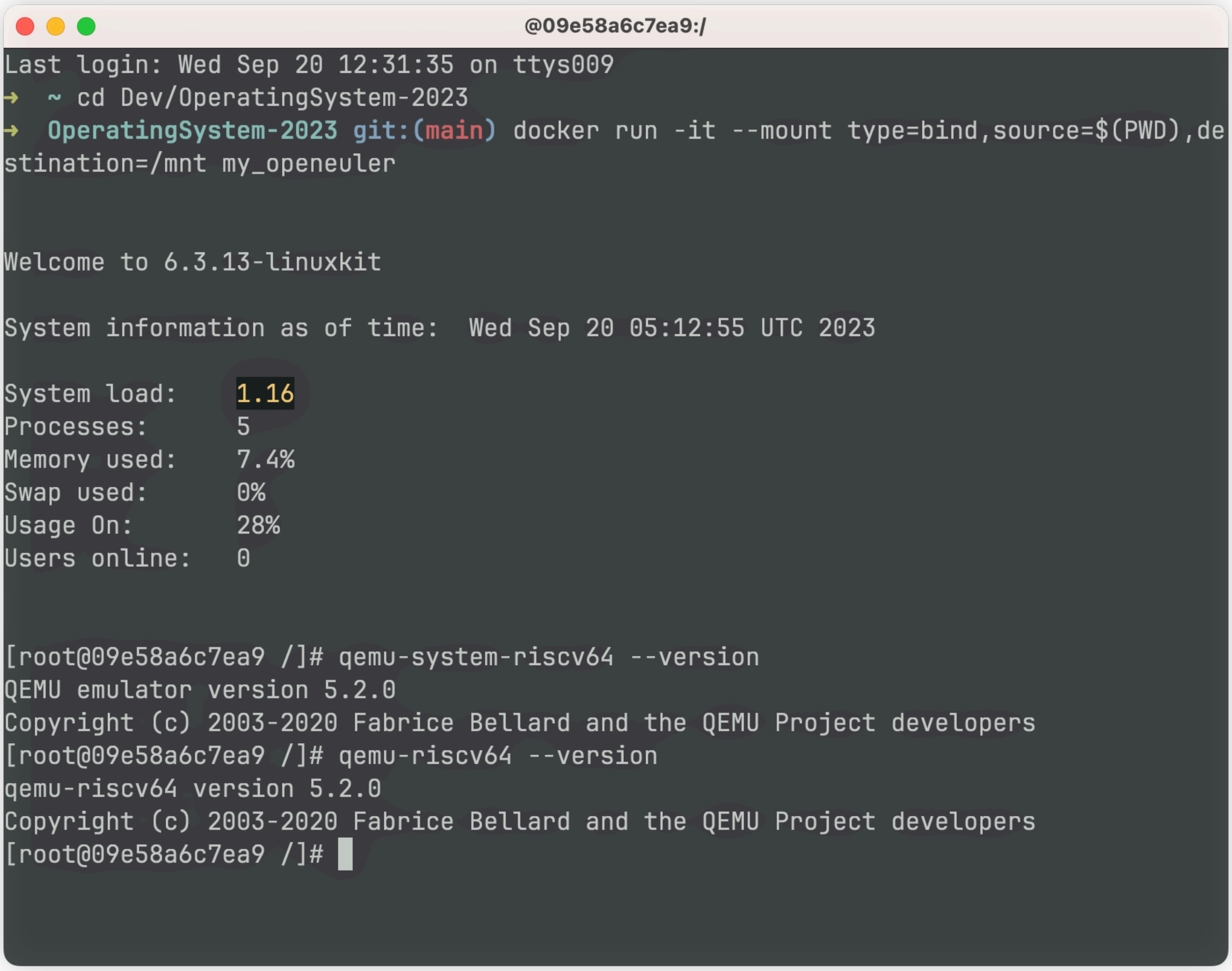
RAM 7.85 GB CPU 0.33% Disk 42.97 GB avail. of 62.67 GB Signed in

可以使用

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

```

来用刚才的镜像创建一个容器并运行，其环境正是刚才保存时的环境：



```
@09e58a6c7ea9:/
Last login: Wed Sep 20 12:31:35 on ttys009
➜  ~ cd Dev/OperatingSystem-2023
➜  OperatingSystem-2023 git:(main) docker run -it --mount type=bind,source=$(PWD),de
stination=/mnt my_openeuler


Welcome to 6.3.13-linuxkit

System information as of time:  Wed Sep 20 05:12:55 UTC 2023

System load:    1.16
Processes:      5
Memory used:    7.4%
Swap used:      0%
Usage On:       28%
Users online:   0


[root@09e58a6c7ea9 /]# qemu-system-riscv64 --version
QEMU emulator version 5.2.0
Copyright (c) 2003-2020 Fabrice Bellard and the QEMU Project developers
[root@09e58a6c7ea9 /]# qemu-riscv64 --version
qemu-riscv64 version 5.2.0
Copyright (c) 2003-2020 Fabrice Bellard and the QEMU Project developers
[root@09e58a6c7ea9 /]#
```