

Docker CE 软件仓库镜像使用帮助

注意：本镜像只提供 Debian/Ubuntu/Fedora/CentOS/RHEL 的 docker 软件包，非 dockerhub

环境准备：x86-64 架构的 Docker 运行环境。

因为 LoongArch32 的环境配置比较复杂，尤其是它的 QEMU 需要补充大量依赖，因此助教为了节省同学们的工作量打包了一个 Docker 镜像，它封装了所有运行我们的 uCore 实验所需的环境，并已发布到 Docker Hub.

这里推荐使用 Windows 的同学使用 WSL2 或 Linux 虚拟机安装 Docker，使用 Linux 的同学直接安装即可，使用 macOS（含 Apple Silicon）的同学使用 Linux 虚拟机安装 Docker。

Docker 自动安装

Docker 提供了一个自动配置与安装的脚本，支持 Debian、RHEL、SUSE 系列及衍生系统的安装。但是采用自动安装的方式网速通常会很慢，因此不推荐。这里依旧提供了自动安装的方法，有兴趣的同学可以尝试。或者跳转至 [Ubuntu 用户安装 Docker](#) 查看其他安装方法。

以下内容假定：

- (1) 您为 root 用户，或有 sudo 权限，或知道 root 密码；
- (2) 您系统上有 curl 或 wget；

```
export DOWNLOAD_URL="https://mirrors.tuna.tsinghua.edu.cn/docker-ce"

# 如您使用 curl

curl -fsSL https://get.docker.com/ | sudo -E sh

# 如您使用 wget

wget -O- https://get.docker.com/ | sudo -E sh
```

Ubuntu 用户安装 Docker

如果你过去安装过 docker，先删掉：

```
for pkg in docker.io docker-doc docker-compose podman-docker containerd runc; do sudo apt-get remove $pkg; done
```

首先安装依赖：

```
sudo apt-get update

sudo apt-get install ca-certificates curl gnupg
```

信任 Docker 的 GPG 公钥并添加仓库：

```
sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

```
echo \  
  
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://mirrors.tuna.tsinghua.edu.cn/docker-ce/linux/ubuntu \  
  
"${(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \  
  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

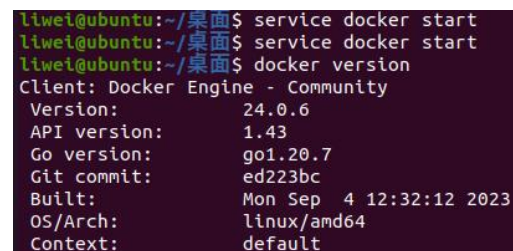
最后安装：

```
sudo apt-get update  
  
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

查看 docker 版本：

```
service docker start    #启动 docker  
  
docker version          #查看 docker 版本
```

docker 版本示例截图：



```
liwei@ubuntu:~/桌面$ service docker start  
liwei@ubuntu:~/桌面$ service docker start  
liwei@ubuntu:~/桌面$ docker version  
Client: Docker Engine - Community  
Version: 24.0.6  
API version: 1.43  
Go version: go1.20.7  
Git commit: ed223bc  
Built: Mon Sep 4 12:32:12 2023  
OS/Arch: linux/amd64  
Context: default
```

配置 Docker(可选)

1. 配置 Docker 组的用户权限:

如果同学们想要在 Linux 中不通过 root 用户 (sudo) 就能直接访问 Docker，以及在运行于用户权限的 VSCode 中直接使用 Docker，可以将自己的用户加入 Docker 创建的组中，操作如下：

```
sudo usermod -aG docker $USER    #将当前用户添加至用户组  
  
newgrp docker                    #更新用户组
```

之后重启电脑即可全局生效，否则每次打开终端需要使用 newgrp docker 才能使权限生效。

2. 配置镜像网站:

对于许多同学的网络环境而言可能从 Docker Hub 官方下载较慢，这里推荐将 Docker Hub 改为南京大学镜像站（目前教育网内唯一 Docker Hub 镜像）。

Linux

在配置文件 /etc/docker/daemon.json 中加入：

```
{  
  "registry-mirrors": ["https://docker.nju.edu.cn/"]  
}
```

注意，如果 /etc/docker/daemon.json 文件不存在，则在指定位置自行创建！

重新启动 docker：

```
sudo systemctl restart docker.service
```

命令行执行 **docker info** 命令，如果从输出中看到如下内容，说明配置成功：

```
Registry Mirrors:
https://docker.nju.edu.cn/
```

下载 Docker 镜像并创建实验环境容器

1. 下载 **chenyy/la32r-env** 镜像：

在终端执行以下命令：

```
docker pull chenyy/la32r-env
```

等待一段时间后终端显示：

```
Using default tag: latest

latest: Pulling from chenyy/la32r-env
d19f32bd9e41: Pull complete
947bd14d4597: Pull complete
04014e0c378d: Pull complete
Digest: sha256:11ab22262972676faa4baeb21043c7a1d073d3fffc26a6ebe9a45391b0f5da82
Status: Downloaded newer image for chenyy/la32r-env:latest
docker.io/chenyy/la32r-env:latest
```

2. 基于 **chenyy/la32r-env** 创建 **la32r-docker** 容器：

```
docker run -dit \

    --name la32r-docker \

    --user=$(id -u $USER):$(id -g $USER) \

    --net=host \

    --workdir="/home/$USER" \

    --volume="/home:/home" \

    --volume="/root:/root" \

    --volume="/mnt:/mnt" \

    --volume="/etc/group:/etc/group:ro" \

    --volume="/etc/passwd:/etc/passwd:ro" \

    --volume="/etc/shadow:/etc/shadow:ro" \

    --volume="/etc/sudoers.d:/etc/sudoers.d:ro" \

    -e LANG=en_US.UTF-8 \

    -e LANGUAGE=en_US.UTF-8 \

    -e LC_ALL=en_US.UTF-8 \

    chenyy/la32r-env
```

这里我们对 Docker 进行了一些配置，简单说就是将整个/home 目录在主机与 Docker 容器之间共享，从而 Docker 能够直接访问用户文件。同时将用户与组权限进行同步，方便在 Docker 中直接访问对应的文件。

举个例子，如果我们 docker 容器内启动了 nginx，但是没有进行 挂载，那么我们只能进入容器里进行修改 nginx.conf

配置，查看日志，更改静态包。但是如果启动时，进行文件和文件夹的挂载，那么我们直接操作宿主机下对应的文件和文件夹就可以得到我们想要的。

3. 启动 **la32r-docker** 容器：

```
docker start la32r-docker
```

4. 进入 la32r-docker 的 Shell：

```
docker exec -it la32r-docker /bin/zsh
```

VSCode 中配置 Docker 插件

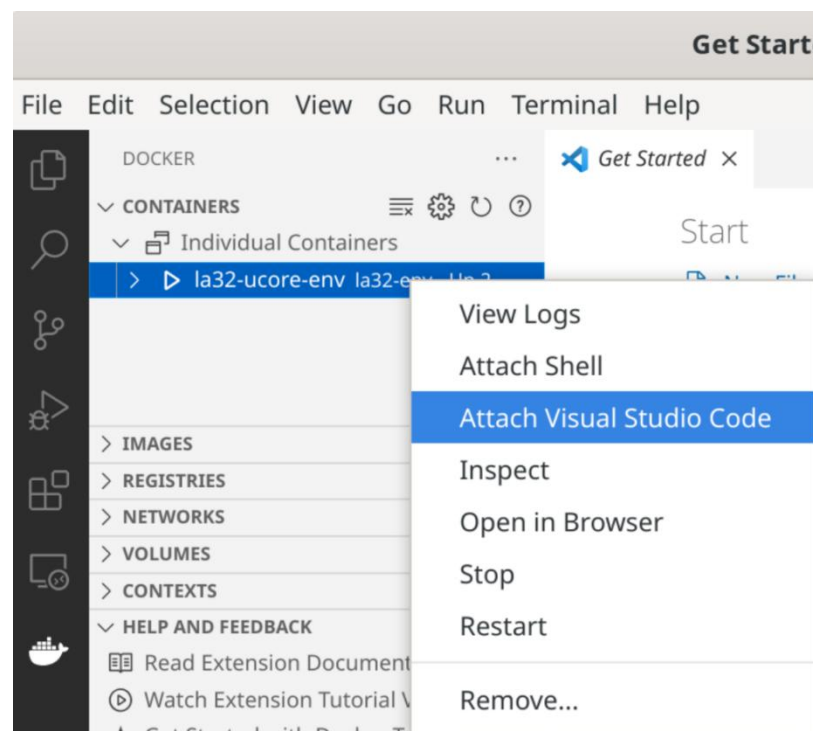
同学们可以在 VSCode 的插件中搜索 Docker，并通过该插件连接到容器中完成操作。

这样我们可以在 VSCode 的插件中找到对应 Docker，可以进行 Attach Shell 或 Attach VSCode。方便我们后续在 Host 端的 VSCode 完成代码编写，并在 Docker 中使用对应环境完成编译与调试。

1. 在 ubuntu 虚拟机中下载 linux 版的 vscode。

2. 在 vscode 中搜索插件 docker 和 dev controller。

安装完毕后会在左侧边栏显示出 docker 的图标 ，同时可以看到我们之前拉下来的 la32r 的镜像。



3. 如果插件安装完毕后 docker 没有显示出之前拉下来的镜像，则可能的原因是我们没有将 docker 加入到用户组中，首先关闭 vscode，然后输入以下命令来解决：

```
sudo usermod -aG docker $USER #将当前用户添加至用户组
```

```
newgrp docker    #更新用户组
```

重启 vscode 后即可正常使用。