

0. 内核校验

先确认内核是否支持docker，这里用一个脚本检验一下。脚本源地址：

<https://github.com/moby/moby/blob/master/contrib/check-config.sh>

用wget下载：

```
 wget https://github.com/moby/moby/raw/master/contrib/check-config.sh
```

拷贝到目标平台直接运行脚本：

```
 chmod +x check-config.sh  
 ./check-config.sh
```

结果主要是两部分，Generally Necessary: 表示必要的配置，如果有显示missing的地方，需要在内核配置中打开，重新编译烧录内核以支持Docker

Optional Features: 是可选配置，根据需要打开

1. 正式安装

使用官方源安装 Docker 以下操作需要在 root 用户下完成，请使用 `sudo -i` 或 `su root` 切换到 root 用户进行操作。

在尝试安装新版本之前卸载旧版本，一定要卸载旧的版本，否则有奇怪的问题不负责

首先更新系统，安装一些必要的软件包：

```
 apt update  
 apt install curl vim wget gnupg dpkg apt-transport-https lsb-release ca-certificates
```

接下来看烧录的系统：

Ubuntu：

添加 Docker 的清华源 GPG 密钥：

```
 curl -fsSL https://mirrors.tuna.tsinghua.edu.cn/docker-ce/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
```

设置存储库：

```
echo \  
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]  
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
```

Debian:

从 Docker 官方下载 GPG 密钥，然后将这个密钥解密并保存到系统的密钥环中；配置软件源，以便能够从 Docker 官方仓库安装 Docker：

```
curl -sSL https://download.docker.com/linux/debian/gpg | gpg --dearmor >  
/usr/share/keyrings/docker-ce.gpg  
  
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-  
ce.gpg] https://download.docker.com/linux/debian $(lsb_release -sc) stable" >  
/etc/apt/sources.list.d/docker.list
```

国内机器可以用清华源：

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

没有权限就chmod 666添加权限。

安装步骤

然后更新系统后即可安装 Docker CE 和 Docker Compose 插件：

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

到安装这步如果报错的话，直接执行下面“※修改docker配置”的步骤，修改配置后还是会报错，使用dockerd查找原因，如果像下面一样的错误：

```
failed to start daemon: Error initializing network controller: error obtaining controller instance: failed to register "bridge" driver: failed to create NAT chain DOCKER: iptables failed: iptables --wait -t nat -N DOCKER: iptables v1.8.2 (nf_tables): CHAIN_ADD failed (No such file or directory): chain PREROUTING
```

执行下面的命令：

```
update-alternatives --set iptables /usr/sbin/iptables-legacy  
update-alternatives --set ip6tables /usr/sbin/ip6tables-legacy
```

然后尝试重启docker

```
systemctl restart docker
```

此时可以使用 docker version 命令检查是否安装成功：

```
root@debian ~ # docker version  
Client: Docker Engine - Community  
  Version:          26.1.4  
  API version:     1.45  
  Go version:      go1.21.11  
  Git commit:       5650f9b  
  Built:            Wed Jun  5 11:29:22 2024  
  OS/Arch:          linux/amd64  
  Context:          default  
  
Server: Docker Engine - Community  
  Engine:  
    Version:          26.1.4  
    API version:     1.45 (minimum version 1.24)  
    Go version:      go1.21.11  
    Git commit:       de5c9cf  
    Built:            Wed Jun  5 11:29:22 2024b  
    OS/Arch:          linux/amd64  
    Experimental:    true  
  containerd:  
    Version:          1.6.33  
    GitCommit:        d2d58213f83a351ca8f528a95fb145f5654e957  
    runc:  
      Version:         1.1.12  
      GitCommit:       v1.1.12-0-g51d5e94  
  docker-init:
```

```
Version:          0.19.0
GitCommit:        de40ad0
```

如果需要某个特定用户可以用 Docker rootless 模式运行 Docker，那么可以把这个用户也加入 docker 组：

```
apt install docker-ce-rootless-extras
sudo usermod -aG docker skyman          #将skyman用户加入docker组
```

安装 Docker Compose

因为我们已经安装了 docker-compose-plugin，所以 Docker 目前已经自带 docker compose 命令，基本上可以替代 docker-compose：

```
root@debian ~ # docker compose version
Docker Compose version v2.27.1
```

如果某些镜像或命令不兼容，则我们还可以单独安装 Docker Compose。

我们可以使用 Docker 官方发布的 Github 直接安装最新版本：

```
curl -L https://github.com/docker/compose/releases/latest/download/docker-compose-
Linux-x86_64 > /usr/local/bin/docker-compose
chmod +x /usr/local/bin/docker-compose          #给下载好的docker-compose文件赋予
执行权限
```

此时可以使用 docker-compose version 命令检查是否安装成功：

```
root@debian ~ # docker-compose version
Docker Compose version v2.27.1
```

※修改 Docker 配置

以下配置会增加镜像地址：

```
cat > /etc/docker/daemon.json << EOF
{
  "registry-mirrors" : [
    "https://docker.registry.cyou",
    "https://docker-cf.registry.cyou",
    "https://dockercf.jsdelivr.fyi",
    "https://docker.jsdelivr.fyi",
    "https://dockertest.jsdelivr.fyi",
    "https://mirror.aliyuncs.com",
    "https://dockerproxy.com",
    "https://mirror.baidubce.com",
    "https://docker.m.daocloud.io",
    "https://docker.nju.edu.cn",
  ]
}
```

```
"https://docker.mirrors.sjtug.sjtu.edu.cn",
"https://docker.mirrors.ustc.edu.cn",
"https://mirror.iscas.ac.cn",
"https://docker.rainbond.cc",
"https://do.nark.eu.org",
"https://dc.j8.work",
"https://dockerproxy.com",
"https://gst6rzl9.mirror.aliyuncs.com",
"https://registry.docker-cn.com",
"http://hub-mirror.c.163.com",
"http://mirrors.ustc.edu.cn/",
"https://mirrors.tuna.tsinghua.edu.cn/",
"http://mirrors.sohu.com/"
],
"insecure-registries" : [
    "registry.docker-cn.com",
    "docker.mirrors.ustc.edu.cn"
],
"debug": true,
"experimental": false
}
EOF
```

然后重启 Docker 服务：

```
systemctl restart docker
```

测试docker能否工作：

```
docker run hello-world
```

原文：<https://zhuanlan.zhihu.com/p/31251918552>