第一周作业

1. 梳理各 Namespace 的作用。
2. 使用 apt/yum/ 二进制安装指定版本的 Docker 。
3. 熟练使用 Docker 数据卷。
4. 熟练使用 Docker 的 bridge 和 container 模式网络。

1. 梳理各 Namespace 的作用

1.1. 什么是namespace

```
#命名空间将全局系统资源包裹在一个抽象中，使其在命名空间内的进程看来，它们是在为其提供服务。在命名空间内的进程看来，它们拥有自己的全局资源的独立实例。拥有自己的全局资源的孤立实例。
#对全局资源的改变  全局资源的变化对作为命名空间成员的其他进程是可见的，但对其他进程是不可见的。命名空间的一个用途是实现容器。
```

1.2. Namespace类型

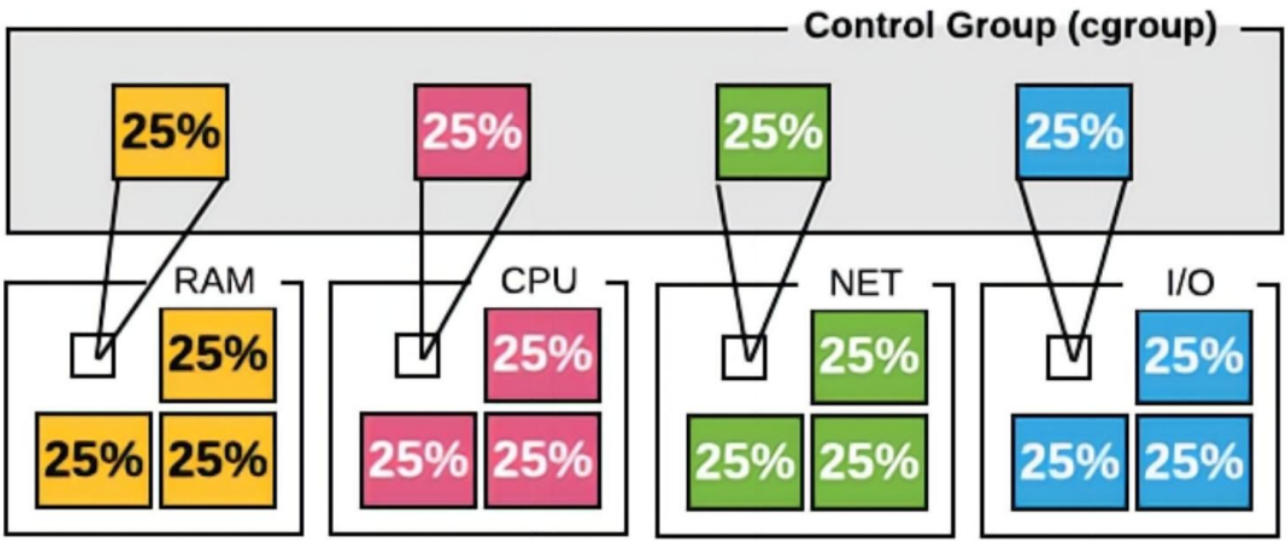
```
#下表显示了Linux上可用的命名空间类型。
      第二列显示了用于在各种API中指定命名空间类型的标志值来指定各种API中的命名空间类型。
      第三列标明了提供命名空间细节的手册页类型的详细信息。
      第四列是被命名空间类型隔离的资源的摘要。命名空间类型所隔离的资源。
```

```
       Namespace       Flag              Page              Isolates
        Cgroup     CLONE_NEWCGROUP cgroup_namespaces(7)  Cgroup root
                                                         directory
        IPC        CLONE_NEWIPC    ipc_namespaces(7)     System V IPC,
                                                         POSIX message
                                                         queues
        Network    CLONE_NEWNET    network_namespaces(7) Network
                                                         devices,
                                                         stacks, ports,
                                                         etc.
        Mount      CLONE_NEWNS     mount_namespaces(7)   Mount points
        PID        CLONE_NEWPID    pid_namespaces(7)     Process IDs
        Time       CLONE_NEWTIME   time_namespaces(7)    Boot and
                                                         monotonic
                                                         clocks
        User       CLONE_NEWUSER   user_namespaces(7)    User and group
                                                         IDs
        UTS        CLONE_NEWUTS    uts_namespaces(7)     Hostname and
                                                         NIS domain
                                                         name
```

隔离类型	功能	系统调用参数	内核版本
MNT Namespace(mount)	提供磁盘挂载点和文件系统的隔离能力	CLONE_NEWNS	Linux 2.4.19
IPC Namespace(Inter-Process Communication)	提供进程间通信的隔离能力	CLONE_NEWIPC	Linux 2.6.19
UTS Namespace(UNIX Timesharing System)	提供主机名隔离能力	CLONE_NEWUTS	Linux 2.6.19
PID Namespace(Process Identification)	提供进程隔离能力	CLONE_NEWPID	Linux 2.6.24
Net Namespace(network)	提供网络隔离能力	CLONE_NEWNET	Linux 2.6.29
User Namespace(user)	提供用户隔离能力	CLONE_NEWUSER	Linux 3.8
Time Namespace	提供时间隔离能力	CLONE_NEWTIME	Linux 5.6
Syslog Namespace	提供syslog隔离能力	syslog namespace是由华为工程师RuiXiang(瑞翔)提出的,但没有合并到linux内核中,后systemd在2020年2月实现了一个名为“journal namespace”的类似功能	
Control group (cgroup) Namespace	提供进程所属的控制组的身份隔离	Linux 4.6	

1.3. Cgroups

```
#在一个容器，如果不对其做任何资源限制，则宿主机会允许其占用无限大的内存空间，有时候会因为代码bug程序会一直申请内存，直到把宿主机内存占完，为了避免此类的问题出现，宿主机有必要对容器进行资源分配限制，比如CPU、内存等，Linux Cgroups的全称是Linux Control Groups，它最主要的作用，就是限制一个进程组能够使用的资源上限，包括CPU、内存、磁盘、网络带宽等等。此外，还能够对进程进行优先级设置，以及将进程挂起和恢复等操作。
```




```
root@castillo:~# cat /boot/config-5.15.0-52-generic |grep CGROUP |grep -v "^#" |wc -l
22
root@castillo:~# cat /boot/config-5.15.0-52-generic |grep MEM |grep -v "^#"
```

2. 使用 apt/yum/ 二进制安装指定版本的 Docker

2.1. 先确认有无旧版本，有则进行删除

```
sudo apt-get remove docker docker-engine docker.io containerd runc
```

2.2. 下载相关工具

```
root@castillo:~# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.1 LTS
Release:        22.04
Codename:       jammy
root@castillo:~# apt-get install \
>     ca-certificates \
>     curl \
>     gnupg \
>     lsb-release
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20211016).
ca-certificates set to manually installed.
lsb-release is already the newest version (11.1.0ubuntu4).
lsb-release set to manually installed.
curl is already the newest version (7.81.0-1ubuntu1.4).
curl set to manually installed.
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 28 not upgraded.
root@castillo:~#
```

2.3. 更换国内镜像源

```
root@castillo:~# curl -fsSL https://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
root@castillo:~#
```

2.4. 添加docker软件源

```
root@castillo:~# echo \
>   "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://mirrors.aliyun.com/docker-ce/linux/ubuntu \
>   $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
root@castillo:~#
```

2.5. 刷新源并查看docker版本

```
root@castillo:~#  apt-get -y update
Hit:1 http://mirrors.aliyun.com/ubuntu jammy InRelease
Get:2 http://mirrors.aliyun.com/ubuntu jammy-updates InRelease [114 kB]


Get:3 http://mirrors.aliyun.com/ubuntu jammy-backports InRelease [99.8 kB]
```

```
Get:4 http://mirrors.aliyun.com/ubuntu jammy-security InRelease [110 kB]
Get:5 https://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy InRelease [48.9 kB]
Get:6 https://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy/stable amd64 Packages [7065 B]
Fetched 380 kB in 1s (518 kB/s)
Reading package lists... Done
root@castillo:~# apt-cache madison docker-ce
 docker-ce | 5:20.10.18~3-0~ubuntu-jammy | https://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.17~3-0~ubuntu-jammy | https://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.16~3-0~ubuntu-jammy | https://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.15~3-0~ubuntu-jammy | https://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.14~3-0~ubuntu-jammy | https://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy/stable amd64 Packages
 docker-ce | 5:20.10.13~3-0~ubuntu-jammy | https://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy/stable amd64 Packages
root@castillo:~#
```

2.6. 安装指定版本docker

```
root@castillo:~# apt-get -y install docker-ce=5:20.10.18~3-0~ubuntu-jammy docker-ce-cli=5:20.10.18~3-0~ubuntu-jammy containerd.io docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras docker-scan-plugin git git-man iptables less liberror-perl libip6tc2 libltdl7 libnetfilter-conntrack3 libnfnetlink0
  libnftnl11 libslirp0 patch pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
  firewalld nftables ed diffutils-doc
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin docker-scan-plugin git git-man iptables less liberror-perl
  libip6tc2 libltdl7 libnetfilter-conntrack3 libnfnetlink0 libnftnl11 libslirp0 patch pigz slirp4netns
0 upgraded, 20 newly installed, 0 to remove and 28 not upgraded.
Need to get 114 MB of archives.
After this operation, 448 MB of additional disk space will be used.
Get:1 http://mirrors.aliyun.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://mirrors.aliyun.com/ubuntu jammy/main amd64 less amd64 590-1build1 [143 kB]
Get:3 http://mirrors.aliyun.com/ubuntu jammy/main amd64 libip6tc2 amd64 1.8.7-1ubuntu5 [20.2 kB]

Get:4 http://mirrors.aliyun.com/ubuntu jammy/main amd64 libnfnetlink0 amd64 1.0.1-
```

```
3build3 [14.6 kB]
Get:5 https://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.6.8-1 [28.1 MB]
Get:6 http://mirrors.aliyun.com/ubuntu jammy/main amd64 libnetfilter-conntrack3 amd64 1.0.9-1 [45.3 kB]
Get:7 http://mirrors.aliyun.com/ubuntu jammy/main amd64 libnftnl11 amd64 1.2.1-1build1 [65.5 kB]
Get:8 http://mirrors.aliyun.com/ubuntu jammy/main amd64 iptables amd64 1.8.7-1ubuntu5 [455 kB]
Get:9 http://mirrors.aliyun.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:10 http://mirrors.aliyun.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.5 [953 kB]
Get:11 http://mirrors.aliyun.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.5 [3132 kB]
Get:12 http://mirrors.aliyun.com/ubuntu jammy/main amd64 libltdl7 amd64 2.4.6-15build2 [39.6 kB]
Get:13 http://mirrors.aliyun.com/ubuntu jammy/main amd64 libslirp0 amd64 4.6.1-1build1 [61.5 kB]
Get:14 http://mirrors.aliyun.com/ubuntu jammy/main amd64 patch amd64 2.7.6-7build2 [109 kB]
Get:15 http://mirrors.aliyun.com/ubuntu jammy/universe amd64 slirp4netns amd64 1.0.1-2 [28.2 kB]
Get:16 https://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:20.10.18~3-0~ubuntu-jammy [41.5 MB]
Get:17 https://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy/stable amd64 docker-ce amd64 5:20.10.18~3-0~ubuntu-jammy [20.4 MB]
Get:18 https://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy/stable amd64 docker-ce-rootless-extras amd64 5:20.10.18~3-0~ubuntu-jammy [8386 kB]
Get:19 https://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy/stable amd64 docker-compose-plugin amd64 2.10.2~ubuntu-jammy [6693 kB]
Get:20 https://mirrors.aliyun.com/docker-ce/linux/ubuntu jammy/stable amd64 docker-scan-plugin amd64 0.17.0~ubuntu-jammy [3521 kB]
Fetched 114 MB in 6s (18.0 MB/s)

debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package pigz.
(Reading database ... 65247 files and directories currently installed.)
Preparing to unpack .../00-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package less.
Preparing to unpack .../01-less_590-1build1_amd64.deb ...
Unpacking less (590-1build1) ...
Selecting previously unselected package libip6tc2:amd64.
Preparing to unpack .../02-libip6tc2_1.8.7-1ubuntu5_amd64.deb ...
Unpacking libip6tc2:amd64 (1.8.7-1ubuntu5) ...
Selecting previously unselected package libnfnetlink0:amd64.
Preparing to unpack .../03-libnfnetlink0_1.0.1-3build3_amd64.deb ...
Unpacking libnfnetlink0:amd64 (1.0.1-3build3) ...
Selecting previously unselected package libnetfilter-conntrack3:amd64.
Preparing to unpack .../04-libnetfilter-conntrack3_1.0.9-1_amd64.deb ...
Unpacking libnetfilter-conntrack3:amd64 (1.0.9-1) ...

Selecting previously unselected package libnftnl11:amd64.
```

```
Preparing to unpack .../05-libnftnl11_1.2.1-1build1_amd64.deb ...
Unpacking libnftnl11:amd64 (1.2.1-1build1) ...
Selecting previously unselected package iptables.
Preparing to unpack .../06-iptables_1.8.7-1ubuntu5_amd64.deb ...
Unpacking iptables (1.8.7-1ubuntu5) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../07-containerd.io_1.6.8-1_amd64.deb ...
Unpacking containerd.io (1.6.8-1) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../08-docker-ce-cli_5%3a20.10.18~3-0~ubuntu-jammy_amd64.deb ...
Unpacking docker-ce-cli (5:20.10.18~3-0~ubuntu-jammy) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../09-docker-ce_5%3a20.10.18~3-0~ubuntu-jammy_amd64.deb ...
Unpacking docker-ce (5:20.10.18~3-0~ubuntu-jammy) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../10-docker-ce-rootless-extras_5%3a20.10.18~3-0~ubuntu-
jammy_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:20.10.18~3-0~ubuntu-jammy) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../11-docker-compose-plugin_2.10.2~ubuntu-jammy_amd64.deb ...
Unpacking docker-compose-plugin (2.10.2~ubuntu-jammy) ...
Selecting previously unselected package docker-scan-plugin.
Preparing to unpack .../12-docker-scan-plugin_0.17.0~ubuntu-jammy_amd64.deb ...
Unpacking docker-scan-plugin (0.17.0~ubuntu-jammy) ...
Selecting previously unselected package liberror-perl.
Preparing to unpack .../13-liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../14-git-man_1%3a2.34.1-1ubuntu1.5_all.deb ...
Unpacking git-man (1:2.34.1-1ubuntu1.5) ...
Selecting previously unselected package git.
Preparing to unpack .../15-git_1%3a2.34.1-1ubuntu1.5_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.5) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../16-libltdl7_2.4.6-15build2_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.6-15build2) ...
Selecting previously unselected package libslirp0:amd64.
Preparing to unpack .../17-libslirp0_4.6.1-1build1_amd64.deb ...
Unpacking libslirp0:amd64 (4.6.1-1build1) ...
Selecting previously unselected package patch.
Preparing to unpack .../18-patch_2.7.6-7build2_amd64.deb ...
Unpacking patch (2.7.6-7build2) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../19-slirp4netns_1.0.1-2_amd64.deb ...
Unpacking slirp4netns (1.0.1-2) ...
Setting up libip6tc2:amd64 (1.8.7-1ubuntu5) ...
Setting up docker-scan-plugin (0.17.0~ubuntu-jammy) ...
Setting up less (590-1build1) ...
Setting up libnftnl11:amd64 (1.2.1-1build1) ...
Setting up liberror-perl (0.17029-1) ...
Setting up containerd.io (1.6.8-1) ...


Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service →
```

```
/lib/systemd/system/containerd.service.
Setting up patch (2.7.6-7build2) ...
Setting up docker-compose-plugin (2.10.2~ubuntu-jammy) ...
Setting up libltdl7:amd64 (2.4.6-15build2) ...
Setting up docker-ce-cli (5:20.10.18~3-0~ubuntu-jammy) ...
Setting up libslirp0:amd64 (4.6.1-1build1) ...
Setting up pigz (2.6-1) ...
Setting up libnfnetlink0:amd64 (1.0.1-3build3) ...
Setting up git-man (1:2.34.1-1ubuntu1.5) ...
Setting up docker-ce-rootless-extras (5:20.10.18~3-0~ubuntu-jammy) ...
Setting up slirp4netns (1.0.1-2) ...
Setting up git (1:2.34.1-1ubuntu1.5) ...
Setting up libnetfilter-conntrack3:amd64 (1.0.9-1) ...
Setting up iptables (1.8.7-1ubuntu5) ...
update-alternatives: using /usr/sbin/iptables-legacy to provide /usr/sbin/iptables
(iptables) in auto mode
update-alternatives: using /usr/sbin/ip6tables-legacy to provide /usr/sbin/ip6tables
(ip6tables) in auto mode
update-alternatives: using /usr/sbin/iptables-nft to provide /usr/sbin/iptables
(iptables) in auto mode
update-alternatives: using /usr/sbin/ip6tables-nft to provide /usr/sbin/ip6tables
(ip6tables) in auto mode
update-alternatives: using /usr/sbin/arptables-nft to provide /usr/sbin/arptables
(arptables) in auto mode
update-alternatives: using /usr/sbin/ebtables-nft to provide /usr/sbin/ebtables
(ebtables) in auto mode
Setting up docker-ce (5:20.10.18~3-0~ubuntu-jammy) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service →
/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket →
/lib/systemd/system/docker.socket.
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend
cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78.)
debconf: falling back to frontend: Readline
Scanning processes...

Scanning candidates...

Scanning linux images...


Running kernel seems to be up-to-date.

Restarting services...
Daemons using outdated libraries
--------------------------------

  1. networkd-dispatcher.service  2. unattended-upgrades.service

(Enter the items or ranges you want to select, separated by spaces.)

```

```
Which services should be restarted?

Service restarts being deferred:
 systemctl restart networkd-dispatcher.service
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@castillo:~#
```

2.7. 启动docker确认安装是否成功

```
root@castillo:~# systemctl enable docker --now
Synchronizing state of docker.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable docker
root@castillo:~# docker info
Client:
 Context:    default
 Debug Mode: false
 Plugins:
  app: Docker App (Docker Inc., v0.9.1-beta3)
  buildx: Docker Buildx (Docker Inc., v0.9.1-docker)
  compose: Docker Compose (Docker Inc., v2.10.2)
  scan: Docker Scan (Docker Inc., v0.17.0)

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 20.10.18
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: systemd
 Cgroup Version: 2
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: runc io.containerd.runc.v2 io.containerd.runtime.v1.linux
 Default Runtime: runc

 Init Binary: docker-init
```

```
 containerd version: 9cd3357b7fd7218e4aec3eae239db1f68a5a6ec6
 runc version: v1.1.4-0-g5fd4c4d
 init version: de40ad0
 Security Options:
  apparmor
  seccomp
   Profile: default
  cgroupns
 Kernel Version: 5.15.0-52-generic
 Operating System: Ubuntu 22.04.1 LTS
 OSType: linux
 Architecture: x86_64
 CPUs: 2
 Total Memory: 3.832GiB
 Name: castillo
 ID: YUYJ:XCZZ:VP76:ENLD:AFBF:PUR7:2WMK:ENSP:VMYB:WHWT:JKCL:MFNL
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  127.0.0.0/8
 Live Restore Enabled: false

root@castillo:~#
```

配置镜像加速(略)

3. 熟练使用 Docker 数据卷

3.1. 数据卷特点

```
#数据卷可以在容器之间共享或重用数据
#数据卷中的更改可以立即生效
#数据卷中的更改不会包含在镜像的更新中
#数据卷默认会一直存在，即使容器被删除
#数据卷的生命周期一直持续到没有容器使用它为止
```

3.2. 创建存储卷

```
root@castillo:~# docker volume create nginx-data
nginx-data
root@castillo:~# docker volume list
DRIVER    VOLUME NAME
local     nginx-data
root@castillo:/var/lib/docker/volumes# ls -l
total 28
brw------- 1 root root 253, 0 Oct 22 13:09 backingFsBlockDev
-rw------- 1 root root  32768 Oct 22 13:19 metadata.db
drwx-----x 3 root root   4096 Oct 22 13:19 nginx-data
root@castillo:/var/lib/docker/volumes# pwd
/var/lib/docker/volumes
```

3.3. 创建nginx实例并挂载存储卷

```
root@castillo:~# docker run -it -d -p 80:80 -v nginx-data:/data nginx:1.20.2
Unable to find image 'nginx:1.20.2' locally
1.20.2: Pulling from library/nginx
a2abf6c4d29d: Pull complete
da03644a1293: Pull complete
dcbfc6badd70: Pull complete
3f7ccff97047: Pull complete
49e31097680b: Pull complete
c423e1dacb26: Pull complete
Digest: sha256:03f3cb0afb7bd5c76e01bfec0ce08803c495348dccce37bcb82c347b4853c00b
Status: Downloaded newer image for nginx:1.20.2
67a6df648744bb5e4167811f2ec272b8bdc1bd9aa6f4088712e7e49167460268
root@castillo:~# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
   PORTS                                  NAMES
67a6df648744   nginx:1.20.2   "/docker-entrypoint.…"   20 seconds ago   Up 13 seconds
  0.0.0.0:80->80/tcp, :::80->80/tcp   jovial_ritchie
```

3.4. ### 修改容器中文件并验证宿主机文件

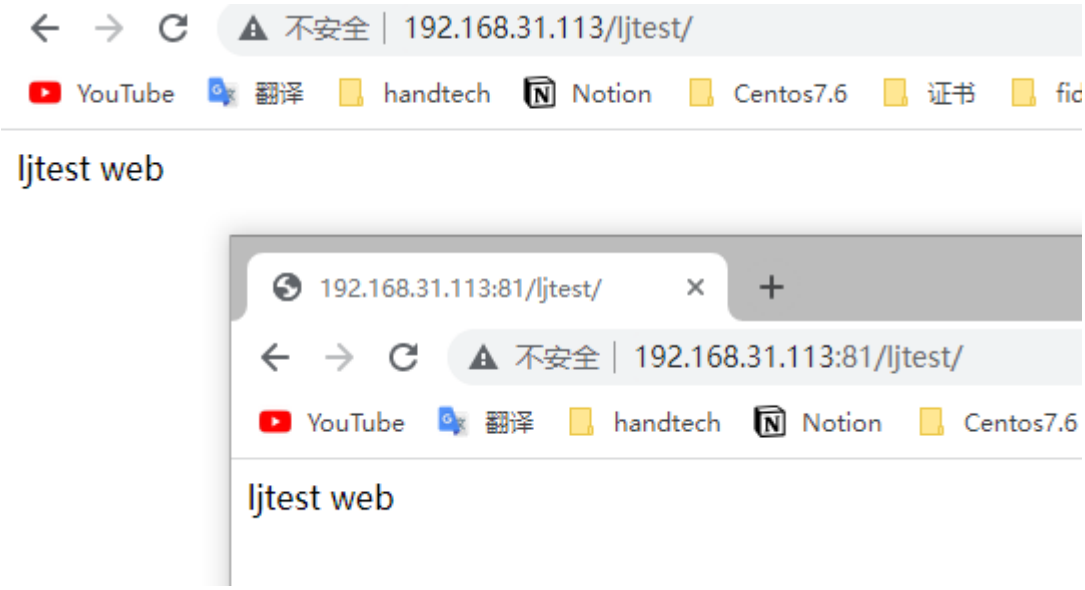
```
root@castillo:/var/lib/docker/volumes# docker exec -it 67a6df648744 bash
root@67a6df648744:/# echo "liujin web" > /data/index.html
root@67a6df648744:/# exit
exit
root@castillo:/var/lib/docker/volumes# cat
backingFsBlockDev  metadata.db        nginx-data/
root@castillo:/var/lib/docker/volumes# cat nginx-data/_data/index.html
liujin web
root@castillo:/var/lib/docker/volumes#
```

3.5. 以数据卷方式挂载

```
root@castillo:~# mkdir /data/ljtest -p
root@castillo:~# echo "ljtest web" > /data/ljtest/index.html
root@castillo:~# cat /data/ljtest/index.html
ljtest web
root@castillo:~#
```

启动两个测试容器，web1容器和web2容器，分别测试能否访问到宿主机的数据，注意使用-v参数，将宿主机目录映射到容器内部，web2的ro 表示在容器内对该目录只读，默认的权限是可读写的

```
root@castillo:~# docker run -d --name web1 -v /data/ljtest:/usr/share/nginx/html/ljtest -p 80:80 nginx:1.20.2
594c6582959dd823671eb8fc86cf179da7544d37b1da7ca8c169e1740024f326
root@castillo:~# docker run -d --name web2 -v /data/ljtest:/usr/share/nginx/html/ljtest:ro -p 81:80 nginx:1.20.2
f7d1b3a3aa3425c9c956490dad95dea31d22c1857e6e2d01cc04b1b5e72b73c3
```



3.6. 多卷挂载

```
###nginx
root@castillo:/data/ljtest# docker run -d --name web3 -v /data/nginx/conf/nginx.conf:/etc/nginx/nginx.conf:ro -p 83:80 nginx:1.20.2
443a0cb7abc82676eacad74752ac5759f703af7f103003e9ffa6cfbe078ddeea
root@castillo:/data/ljtest# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS                                   NAMES
443a0cb7abc8   nginx:1.20.2   "/docker-entrypoint.…"   4 seconds ago    Up 2 seconds    0.0.0.0:83->80/tcp, :::83->80/tcp       web3
f7d1b3a3aa34   nginx:1.20.2   "/docker-entrypoint.…"   21 minutes ago   Up 21 minutes   0.0.0.0:81->80/tcp, :::81->80/tcp       web2
594c6582959d   nginx:1.20.2   "/docker-entrypoint.…"   22 minutes ago   Up 22 minutes   0.0.0.0:80->80/tcp, :::80->80/tcp       web1
root@castillo:/data/ljtest# docker exec -it 443a0cb7abc8 bash
root@443a0cb7abc8:/# cd /etc/nginx/

root@443a0cb7abc8:/etc/nginx# mv nginx.conf nginx.confbak
```

```
mv: cannot move 'nginx.conf' to 'nginx.confbak': Device or resource busy
root@443a0cb7abc8:/etc/nginx#

###mysql
root@castillo:~# mkdir /data/mysql
root@castillo:~# ls -l
total 0
root@castillo:~# docker run -it -d -p 3306:3306 -v /data/mysql:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=111111 mysql:5.7.38
f5ac8b0f4bc03d97c99cd2fb76b5bea5c3d07d45a03018b2a7a43dc9c57d240f
root@castillo:~# docker exec -it f5ac8b0f4bc0 mysql -u root -p111111
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.38 MySQL Community Server (GPL)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

3.7. 数据管理-删除容器

```
#创建容器的时候指定参数会删除/var/lib/docker/containers/的容器数据目录，但是不会删除数据卷的内容，如下
root@castillo:/data/mysql# docker rm -f web3
web3
root@castillo:/data/mysql# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED             STATUS
        PORTS                                                  NAMES
f5ac8b0f4bc0   mysql:5.7.38   "docker-entrypoint.s…"   11 minutes ago      Up 11
minutes      0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   angry_poitras
f7d1b3a3aa34   nginx:1.20.2   "/docker-entrypoint.…"   About an hour ago   Up About
an hour   0.0.0.0:81->80/tcp, :::81->80/tcp                       web2
594c6582959d   nginx:1.20.2   "/docker-entrypoint.…"   About an hour ago   Up About
an hour   0.0.0.0:80->80/tcp, :::80->80/tcp                       web1
root@castillo:/data/mysql# ls /data/ljtest/index.html
/data/ljtest/index.html
root@castillo:/data/mysql#
```

3.7.1. 数据卷的特点及使用：

```
# 数据卷是宿主机的目录或者文件，并且可以在多个容器之间共同使用。
# 在宿主机对数据卷更改数据后会在所有容器里面会立即更新。
# 数据卷的数据可以持久保存，即使删除使用使用该容器卷的容器也不影响。
# 在容器里面的写入数据不会影响到镜像本身。
```

3.7.2. 数据卷使用场景：

```
# 容器数据持久化(mysql数据、nginx日志等类型)
# 静态web页面挂载
# 应用配置文件挂载
# 多容器间的目录或文件共享
```

3.7.3. 数据卷容器

```
#数据卷容器功能是可以让数据在多个docker容器之间共享，即先要创建一个后台运行的A容器作为Server，之后创建的B容器、C容器等都可以同时访问A容器的内容，因此数据卷容器用于为其它容器提供卷的挂载继承服务，数据卷为其它容器提供数据读写服务，A容器称为server端、其它容器成为client端：
root@castillo:/data/mysql# docker run -d --name volume-server -v /data/ljtest/:/usr/share/nginx/html/ljtest -v /data/nginx/conf/nginx.conf:/etc/nginx/nginx.conf:ro registry.cn-hangzhou.aliyuncs.com/zhangshijie/pause:3.8
Unable to find image 'registry.cn-hangzhou.aliyuncs.com/zhangshijie/pause:3.8' locally
3.8: Pulling from zhangshijie/pause
9457426d6899: Pull complete
Digest: sha256:e0cc6dba04bee00badd8b13495d4411060b5563a9499fbc20e46316328efad30
Status: Downloaded newer image for registry.cn-hangzhou.aliyuncs.com/zhangshijie/pause:3.8
982adc279d94b14d753494cf6b9d90fa88e01edbdd149ea673388ca35b65cccd
root@castillo:/data/mysql# docker ps
CONTAINER ID   IMAGE                                                  COMMAND                  CREATED          STATUS          PORTS                                                                                  NAMES
982adc279d94   registry.cn-hangzhou.aliyuncs.com/zhangshijie/pause:3.8   "/pause"                 5 seconds ago    Up 4 seconds                                                                           volume-server
f5ac8b0f4bc0   mysql:5.7.38                                           "docker-entrypoint.s…"   21 minutes ago   Up 21 minutes   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   angry_poitras
f7d1b3a3aa34   nginx:1.20.2                                           "/docker-entrypoint.…"   2 hours ago      Up 2 hours      0.0.0.0:81->80/tcp, :::81->80/tcp                                      web2
594c6582959d   nginx:1.20.2                                           "/docker-entrypoint.…"   2 hours ago      Up 2 hours      0.0.0.0:80->80/tcp, :::80->80/tcp                                      web1
root@castillo:/data/mysql# docker rm -f web1
web1
root@castillo:/data/mysql# docker rm -f web2
web2
root@castillo:/data/mysql# docker run -d --name web1 -p 80:80 --volumes-from volume-server nginx:1.20.2
629b5111709fd0d99497527c732abb5f081d2259a3189750c644167cab06a304
root@castillo:/data/mysql# docker run -d --name web2 -p 80:80 --volumes-from volume-server nginx:1.20.2
da55eab371843ba8ce269e2731011c09c0f6fb57285a4de143ac2658a3c8ee23
root@castillo:/data/mysql# docker ps
CONTAINER ID   IMAGE                                                  COMMAND
```

```
                CREATED          STATUS          PORTS
                  NAMES
25  da55eab37184   nginx:1.20.2                                                    "/docker-
    entrypoint.…"   3 seconds ago    Up 2 seconds    0.0.0.0:81->80/tcp, :::81->80/tcp
                      web2
26  629b5111709f   nginx:1.20.2                                                    "/docker-
    entrypoint.…"   59 seconds ago   Up 57 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp
                      web1
27  982adc279d94   registry.cn-hangzhou.aliyuncs.com/zhangshijie/pause:3.8   "/pause"
              3 minutes ago    Up 3 minutes
                volume-server
28  f5ac8b0f4bc0   mysql:5.7.38                                                    "docker-
    entrypoint.s…"   24 minutes ago   Up 24 minutes   0.0.0.0:3306->3306/tcp, :::3306-
    >3306/tcp, 33060/tcp   angry_poitras
```

3.7.4. 特点:

```
1  # 适用于同类服务的数据卷共享
2  # client会继承卷server挂载和挂载权限
3  # 停止卷server，也不影响已经运行的容器、甚至也不影响新建容器
4  # 删除卷server，不影响已经运行的容器，但是不能新建容器
```

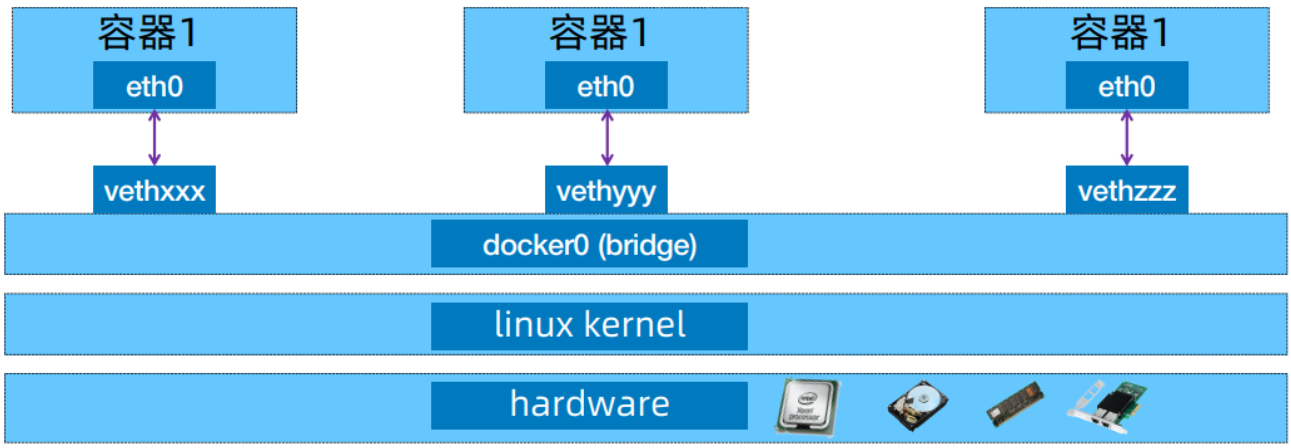
4. 熟练使用 Docker 的 bridge 和 container 模式网络

4.1. 简介

```
1  # Docker服务安装完成之后，默认在每个宿主机会生成一个名称为docker0的网卡其IP地址都是
   172.17.0.1/16，并且会生成三种不能类型的网络，如下：
2  root@castillo:/data/mysql# ifconfig docker0
3  docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
4          inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
5          inet6 fe80::42:74ff:fed3:4add  prefixlen 64  scopeid 0x20<link>
6          ether 02:42:74:d3:4a:dd  txqueuelen 0  (Ethernet)
7          RX packets 73  bytes 9403 (9.4 KB)
8          RX errors 0  dropped 0  overruns 0  frame 0
9          TX packets 90  bytes 9991 (9.9 KB)
10         TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
11
12 root@castillo:/data/mysql#
13 # 另外会额外创建三个默认网络，用于不同的使用场景：
14 root@castillo:/data/mysql# docker network list
15 NETWORK ID     NAME      DRIVER    SCOPE
16 a510745f1b02   bridge    bridge    local #桥接网络，默认使用的模式，容器基于SNAT进行地址转换
   访问宿主机以外的环境
17 3c35d3f173b0   host      host      local #host网络，直接使用宿主机的网络（ 不创建net
   namespace），性能最好，但是容器端口不能冲突
18 59faf206144d   none      null      local #空网络，容器不会分配有效的IP地址(只有一个回环网卡用
   于内部通信)，用于离线数据处理等场景
```

4.2. bridge

```
#docker的默认模式即不指定任何模式就是bridge模式，也是目前使用比较多的网络模式，此模式创建的容器会为每一个容器分配自己的网络IP等信息，并将容器连接到一个虚拟网桥与外界通信。
root@castillo:~# docker run -d --name nginx-web1-bridge-test-container -p 80:80 --net=bridge nginx:1.20.2
36f10476166c5d80c9dd3310fe3b8b94409c6d5cb1c1e96541a46e6b16543638
root@castillo:~# ls -l
total 0
root@castillo:~# docker ps
CONTAINER ID   IMAGE                                                                COMMAND
           CREATED          STATUS          PORTS
           NAMES
36f10476166c   nginx:1.20.2                                                         "/docker-entrypoint.…"   7 minutes ago    Up 7 minutes    0.0.0.0:80->80/tcp, :::80->80/tcp
                 nginx-web1-bridge-test-container
```




```
#用户自定义网桥和默认网桥之间的区别
① 用户定义的网桥在容器之间提供自动 DNS 解析。

默认网桥网络上的容器只能通过 IP 地址相互访问

用户自定义的网桥网络上，容器可以通过名称或别名相互解析。

② 用户定义的网桥提供更好的隔离。

所有未指定容器的容器都将连接到默认网桥网络。这可能是一个风险，因为不相关的堆栈/服务/容器能够进行通信。
使用用户定义的网络提供了一个作用域网络，在该网络中，只有连接到该网络的容器才能进行通信。
③ 容器可以动态地从用户自定义的网络中附加和分离。

在容器的生存期内，可以动态连接容器或将其与用户定义的网络断开连接。若要从默认网桥网络中删除容器，需要停止该容器并使用不同的网络选项重新创建它。
④ 每个用户定义的网络都会创建一个可配置的网桥。

如果容器使用默认网桥网络，则可以对其进行配置，但所有容器都使用相同的设置，例如 MTU 和规则。此外，配置默认网桥网络发生在 Docker 本身之外，并且需要重新启动 Docker。


用户自定义的网桥网络是使用 创建和配置的。如果不同的应用程序组具有不同的网络要求，则可以在创建每个用户
```

```
定义的网桥时单独配置它。
⑤ 默认网桥网络上的链接容器共享环境变量。

多个容器可以使用 Docker 卷挂载包含共享信息的文件或目录。
可以使用多个容器一起启动，并且组合文件可以定义共享变量。
可以使用 swarm 服务而不是独立容器，并利用共享密钥和配置。
```

4.3. container

```
#Container模式即容器模式，使用参数 --net=container:目标容器名称/ID 指定，使用此模式创建的容器需指定和一个已经存在的容器共享一个网络namespace，而不会创建独立的namespace，即新创建的容器不会创建自己的网卡也不会配置自己的IP，而是和一个已经存在的被指定的目标容器共享对方的IP和端口范围，因此这个容器的端口不能和被指定的目标容器端口冲突，除了网络之外的文件系统、用户信息、进程信息等仍然保持相互隔离，两个容器的进程可以通过lo网卡及容器IP进行通信。
```

4.4. 创建两个容器并验证

```
root@castillo:~# docker run -d --name nginx-container -p 80:80 --net=bridge nginx:1.22.0-alpine
Unable to find image 'nginx:1.22.0-alpine' locally
1.22.0-alpine: Pulling from library/nginx
213ec9aee27d: Pull complete
1bfd2b69cf63: Pull complete
a19f4cc2e029: Pull complete
4ae981811a6d: Pull complete
7a662f439736: Pull complete
a317c3c2c906: Pull complete
Digest: sha256:addd3bf05ec3c69ef3e8f0021ce1ca98e0eb21117b97ab8b64127e3ff6e444ec
Status: Downloaded newer image for nginx:1.22.0-alpine
c029050aed5ed612a18f6a2bdf04e7482eb198347e0e21e504725216180800fc
root@castillo:~# docker run -d --name php-container --net=container:nginx-container php:7.4.30-fpm-alpine
Unable to find image 'php:7.4.30-fpm-alpine' locally
7.4.30-fpm-alpine: Pulling from library/php
213ec9aee27d: Already exists
a600fdbc30cc: Pull complete
0cdd6cb15c0d: Pull complete
8a4c40d8aee7: Pull complete
77e67522f4fd: Pull complete
d181492ef8e9: Pull complete
b3ee11034df4: Pull complete
eb72a5cacfff: Pull complete
9acd67bcb441: Pull complete
d2d9674ad948: Pull complete
Digest: sha256:7e551f8b4666967229d03a1f00008ed5098d5cccbf8a59129203539ff745ecbb
Status: Downloaded newer image for php:7.4.30-fpm-alpine
6f1e631aa57b294a8670940cd85ef1f546ad29d74bfb70c2eb8368b7a223dde9
root@castillo:~# docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED
STATUS         PORTS                                NAMES

6f1e631aa57b   php:7.4.30-fpm-alpine   "docker-php-entrypoi…"   2 minutes ago   Up 2
```

```
minutes                                         php-container
c029050aed5e   nginx:1.22.0-alpine     "/docker-entrypoint.…"   4 minutes ago   Up 4 minutes   0.0.0.0:80->80/tcp, :::80->80/tcp   nginx-container
root@castillo:~# docker exec -it c029050aed5e bash
OCI runtime exec failed: exec failed: unable to start container process: exec: "bash": executable file not found in $PATH: unknown
root@castillo:~# docker exec -it c029050aed5e sh
/ # ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:AC:11:00:02
          inet addr:172.17.0.2  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1156 (1.1 KiB)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

/ # exit
root@castillo:~# docker exec -it 6f1e631aa57b bash
OCI runtime exec failed: exec failed: unable to start container process: exec: "bash": executable file not found in $PATH: unknown
root@castillo:~# docker exec -it 6f1e631aa57b sh
/var/www/html # ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:AC:11:00:02
          inet addr:172.17.0.2  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1156 (1.1 KiB)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

/var/www/html #


#验证namespace
root@castillo:/var/run/docker/netns# ln -s /var/run/docker/netns /var/run/netns
root@castillo:/var/run/docker/netns# ls -l /var/run/ne

needrestart/ netns/
```

```
root@castillo:/var/run/docker/netns# ls -l /var/run/ne
needrestart/ netns/
root@castillo:/var/run/docker/netns# ls -l /var/run/netns
lrwxrwxrwx 1 root root 21 Oct 22 16:49 /var/run/netns -> /var/run/docker/netns
root@castillo:/var/run/docker/netns# ip netns list
02e923db9818 (id: 0)
root@castillo:/var/run/docker/netns# ip netns exec 02e923db9818 ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
26: eth0@if27: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
group default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
root@castillo:/var/run/docker/netns#
```