

# Docker

@pvavilin

15 января 2023 г.

# Outline

# chroot

Запустим Python в chroot

```
echo -e "$(ldd /usr/bin/python3 | \
    egrep -o '/lib.*\.[0-9]')"
/lib/x86_64-linux-gnu/libc.so.6
/lib/x86_64-linux-gnu/libpthread.so.0
/lib/x86_64-linux-gnu/libdl.so.2
/lib/x86_64-linux-gnu/libutil.so.1
/lib/x86_64-linux-gnu/libm.so.6
/lib/x86_64-linux-gnu/libexpat.so.1
/lib/x86_64-linux-gnu/libz.so.1
/lib64/ld-linux-x86-64.so.2
```

# chroot

```
copy_to_chroot () {  
    local name="${1}"  
    mkdir -p $(dirname "chroot${name}")  
    for i in $(ldd ${name} | \  
        grep -o '/lib.*\.[0-9]'); do  
        echo "chroot${i}"  
        mkdir -p $(dirname "chroot${i}")  
        /bin/cp $i "chroot${i}";  
    done  
    mkdir -p chroot/bin  
    /bin/cp ${name} chroot/bin/  
}
```

# virtual machine

```
vbox.sh
```

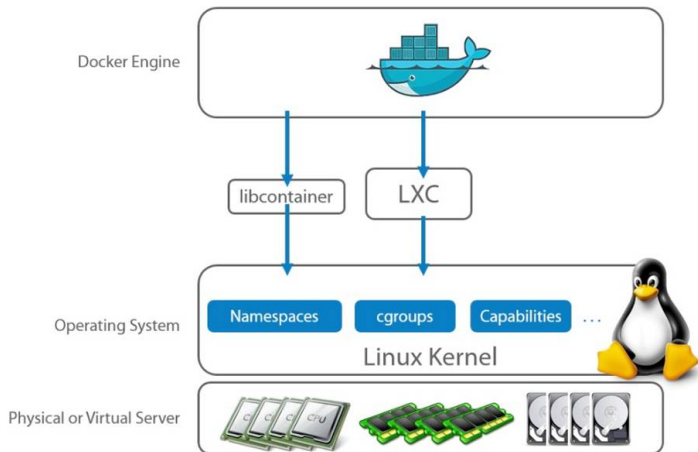
# vagrant

```
mkdir vagrant_project  
cd vagrant_project  
vagrant init generic/ubuntu2004  
vagrant up
```

# Linux Namespaces

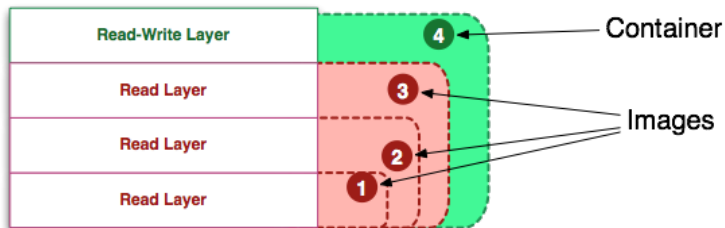
TryTry

# Docker



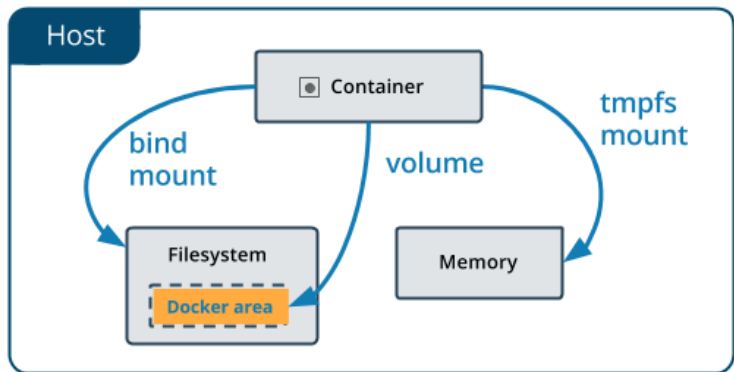


# Слои



# Монтирование

[Документация](#)



# Где всё на самом деле хранится?

## docker volumes

/var/lib/docker

docker system df

- TYPE	TOTAL	ACTIVE	SIZE	RECLAIM
- Images	4	0	1.558GB	1.558GB
- Containers	0	0	0B	0B
- Local Volumes	3	0	45.46MB	45.46MB
- Build Cache	28	0	2.499MB	2.499MB

# полезные команды для работы с Docker

- очистка для пересборки "на чистую"

```
docker system prune --all
docker volume prune
```

- посмотреть контейнеры

```
# запущенные
docker container ls
# и остановленные в том числе
docker container ls -a
```

- пересобрать контейнеры

```
docker compose up --build --no-deps <
```

# docker compose / docker-compose

## Официальная инструкция по установке плагина

```
echo "--docker-compose--"
docker-compose version
echo "--docker compose--"
docker compose version
--docker-compose--
docker-compose version 1.29.2, build 5becea4c
docker-py version: 5.0.0
CPython version: 3.7.10
OpenSSL version: OpenSSL 1.1.0l 10 Sep 2019
--docker compose--
Docker Compose version v2.5.0
```

# Как попасть на запущенный контейнер?

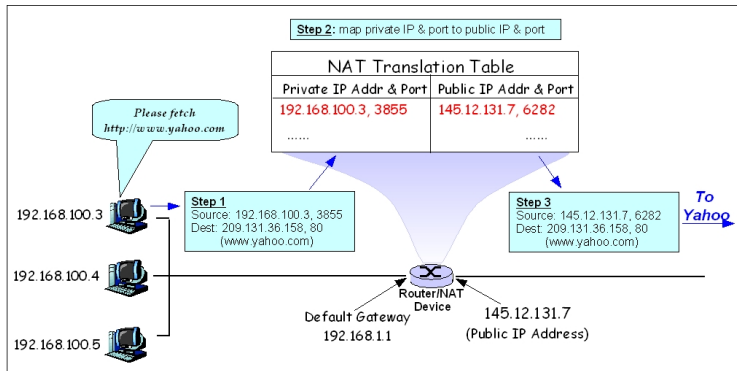
Если вы не настроили доступ по SSH на контейнер, то всегда можно туда зайти таким способом

```
docker exec -it <node> /bin/bash
```

# NAT (Network Address Translation)

docker ports

IPv4-адресов мало, а портов ещё меньше!



## немного про NGINX

```
location / {  
    proxy_set_header Host $host;  
    proxy_set_header X-Real-IP \  
        $remote_addr;  
    proxy_set_header X-Forwarded-For \  
        $proxy_add_x_forwarded_for;  
    proxy_set_header X-Forwarded-Proto \  
        $scheme;  
/*  
 *   web это доменное имя известное  
 *   только внутри Docker-сети  
*/  
    proxy_pass http://web:8080;  
}
```



# Собрать свой образ с нуля?

baseimage

# Запустить контейнер с графической системой?

Можно

# Запустить контейнер в контейнере?

- в Linux точно можно
- лучше так не делать (могут возникнуть технические сложности)
- это может быть нужно когда у тебя CI система сама запускает сборку в контейнере, а в процессе сборки может создавать контейнер. тогда надо просто пробрасывать docker daemon из хоста

# Дополнительная литература

- namespaces
- cgroups
- Linux контейнеры
- Образы и контейнеры Docker в картинках

# Вопросы-ответы

