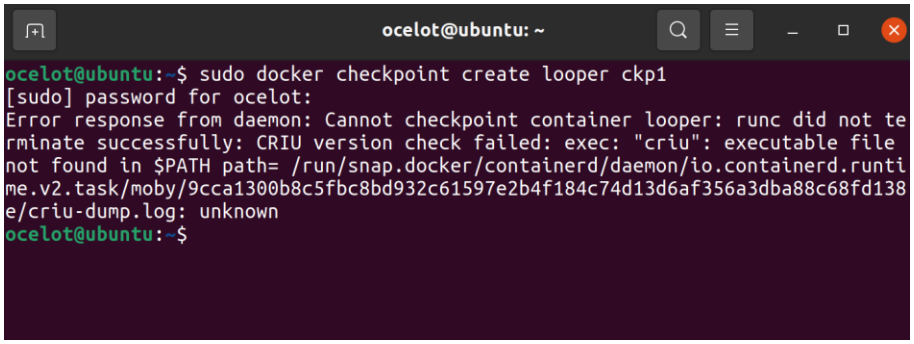
分布式系统作业-2 利用 criu 实现容器迁移

19335015 陈恩婷

1. 没有解决的 Bug

一开始尝试使用 docker 和 criu，就出现了以下报错:



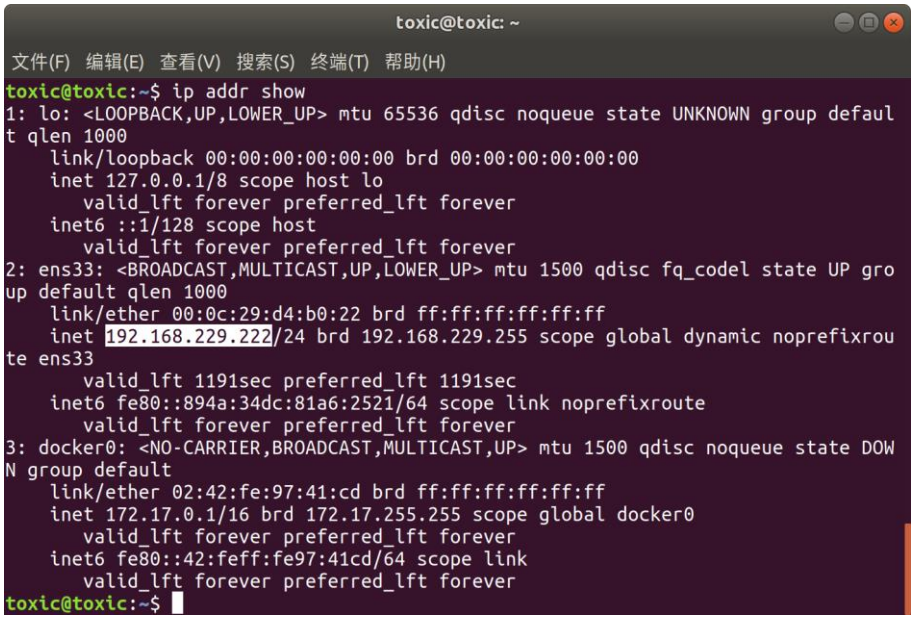
本人在老师和助教的指导下，把所有的能想到的方法都试了个遍，还是没有办法解决，主要尝试过的有:

a. 重装 docker 和 criu，使用源码安装指定版本;
b. 换用 Ubuntu 18.04

可能是需要在系统中安装相关依赖的原因（实在找不出来是缺了什么包），这些尝试都没有什么效果。最终的办法是借用了同学刚好不会报错的 Ubuntu 18.04 虚拟机来完成实验。

2. 在可以正常运行 docker 和 criu 的虚拟机下完成实验

需要准备两台虚拟机，分别用 ip addr show 找到他们的 ip 地址



其中客户机的 ip 地址为 192.168.229.129，服务器的 ip 地址为 192.168.229.222。

然后开始实验，具体步骤大致如下：

（1） 完成配置

Step 1: Configure NFS on host & client server

- **Downloading and Installing the Components**
- On the host:

```
$ sudo apt-get update
$ sudo apt-get install nfs-kernel-server
```

- On the client:

```
$ sudo apt-get update
$ sudo apt-get install nfs-common
```

- **Configuring the NFS Exports on the Host Server**

```
$ sudo nano /etc/exports
# Add the below line to exports file:
/home       129.59.1.2(rw,sync,no_root_squash,no_subtree_check)
# Then restart nfs-kernel-server service
$ sudo systemctl restart nfs-kernel-server
```

- **Adjusting the Firewall on the Host**

```
# First, check firewall status
$ sudo ufw status
# If ufw is inactive, use the below command to enable ufw:
$ sudo ufw enable
# Make ufw allow incoming and outgoing:
$ sudo ufw default allow incoming
$ sudo ufw default allow outgoing
# Make client server can access host server
$ sudo ufw allow from 129.59.1.2 to any port nfs
# Check ufw status
$ sudo ufw status numbered
```

- **Creating the Mount Points on the Client**

```
$ sudo mkdir -p /nfs/home
```

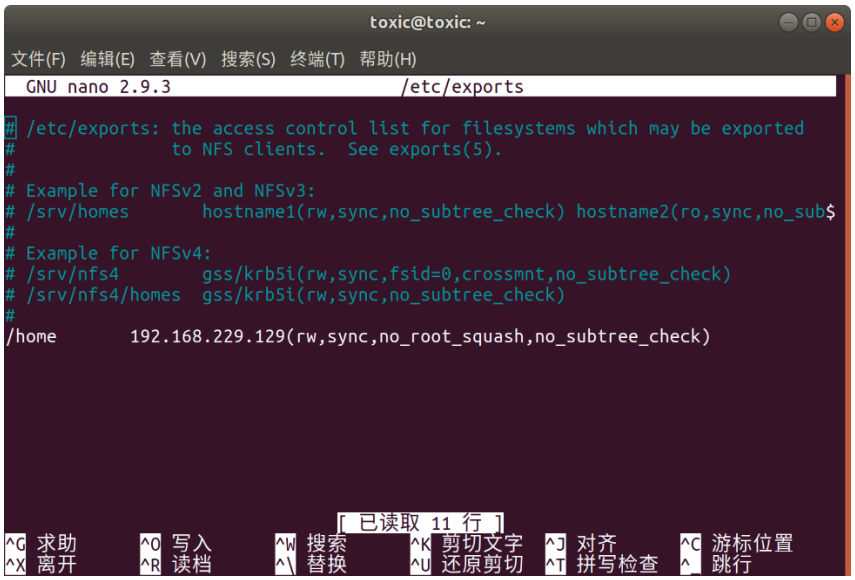
- **Mounting the Directory on the Client**

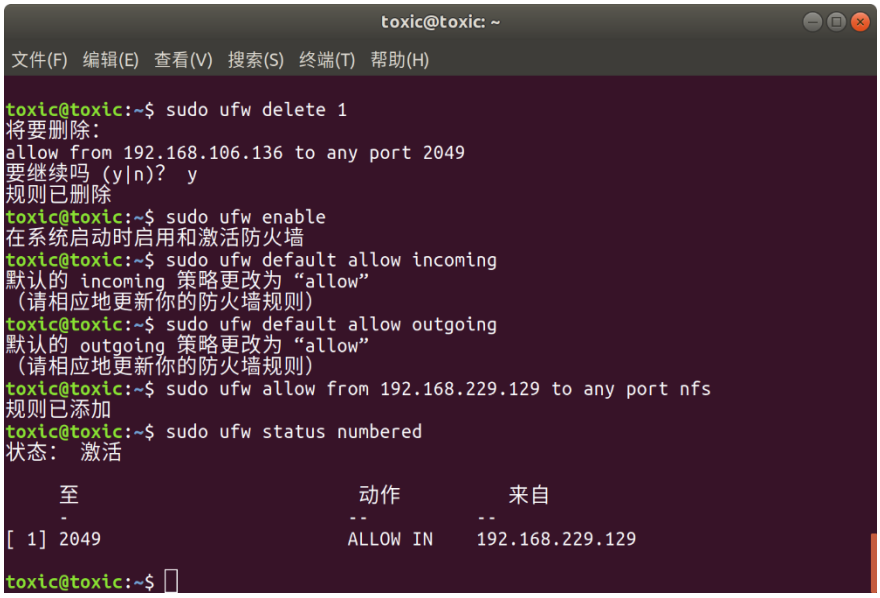
```
sudo mount 129.59.1.1:/home /nfs/home
```

- **Checking the Mounted Directory on the Client**

```
$ df -h
```

过程比较简单，将 ip 地址换成自己两台虚拟机的地址即可，这里附上几张截图：





(2) 安装 docker 和 criu

这一步本人在自己的虚拟机上已经反复尝试了多次官方源和源码安装，可惜没有留下截图，方法比较简单，这里不再赘述

(3) 测试容器迁移

- Test your live migration

1. On the host

```
$ docker run -d --name looper2 --security-opt seccomp:unconfined busybox \
        /bin/sh -c 'i=0; while true; do echo $i; i=$(expr $i + 1); sleep 1; done'

# wait a few seconds to give the container an opportunity to print a few lines, then
$ docker checkpoint create --checkpoint-dir=/home/ubuntu/Container-Checkpoints/ looper2 check

# check your container & print log file
$ docker ps
$ docker logs looper2
```

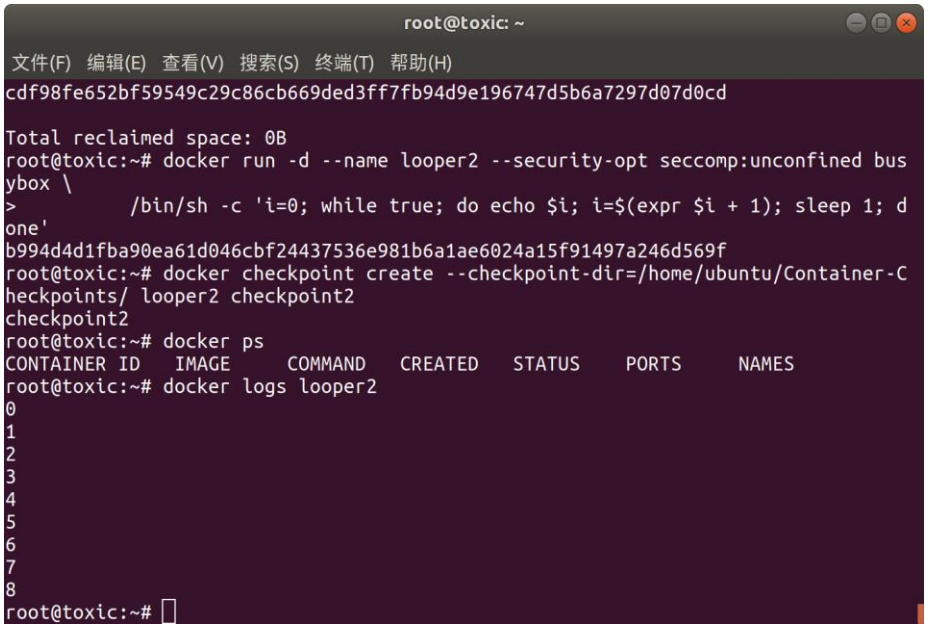
2. On the client

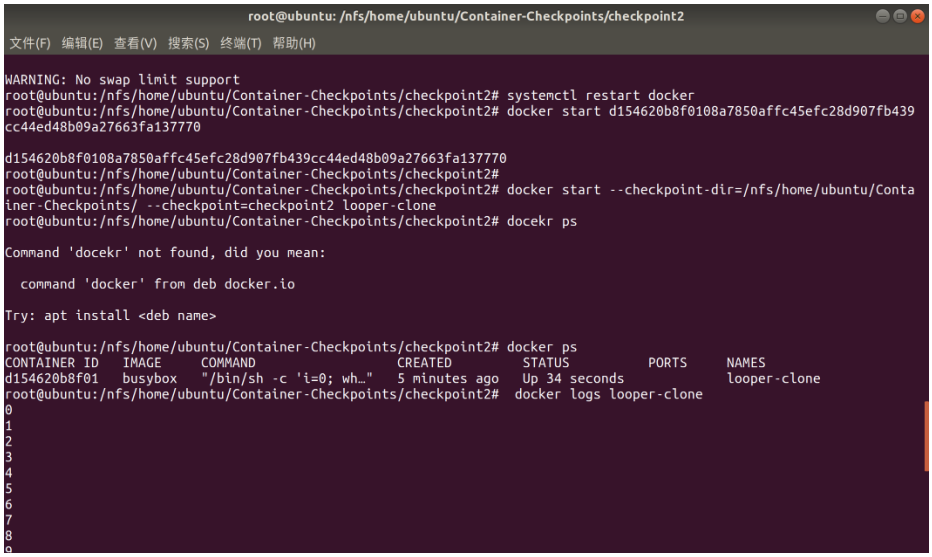
```
$ docker create --name looper-clone --security-opt seccomp:unconfined busybox \
        /bin/sh -c 'i=0; while true; do echo $i; i=$(expr $i + 1); sleep 1; done'

$ docker start --checkpoint-dir=/nfs/home/ubuntu/Container-Checkpoints/ --checkpoint=checkpoi

# check your container
$ docekr ps
$ docker logs looper-clone
```

这个是报错最难解决的一步，主要就是第一部分给出的报错截图。不过在同学的虚拟机上面，这一部分倒是很顺利，非常快就完成了，以下是截图：





如图所示，成功完成了容器的迁移。