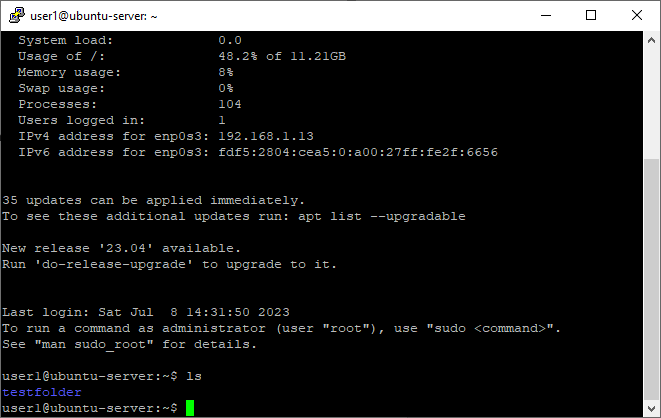
Запускаем изолированный процесс bash

Для этого создаем свою директорию

mkdir testfolder

ls

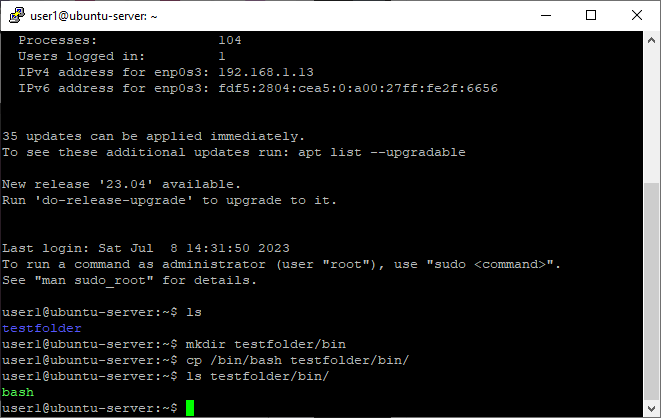


Bash лежит в директории /bin, поэтому в нашей директории testfolder создаем директорию /bin и копируем туда файлы bin/bash:

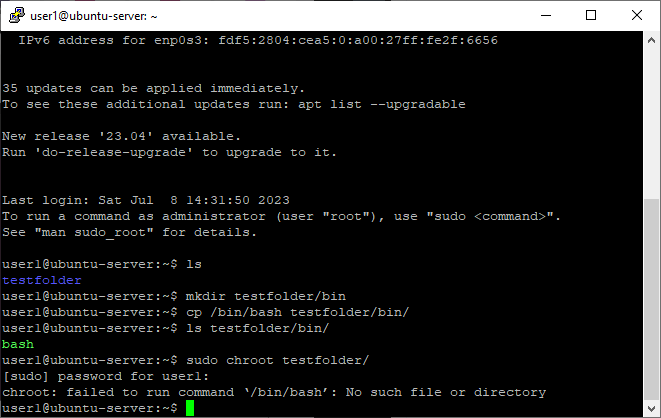
mkdir testfolder/bin

cp /bin/bash testfolder/bin/

ls testfolder/bin/

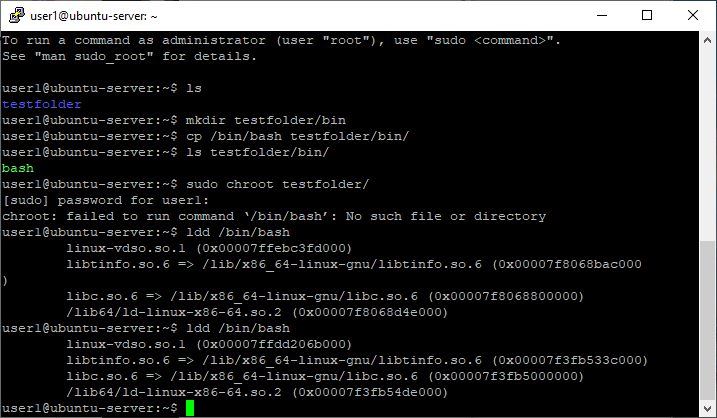


Изолировать наш новый каталог не получится, т.к. нет необходимых библиотек:



Смотрим, какие библиотеки нужно загрузить:

ldd /bin/bash



Создаем соответствующие каталоги, куда будем загружать библиотеки, и копируем:

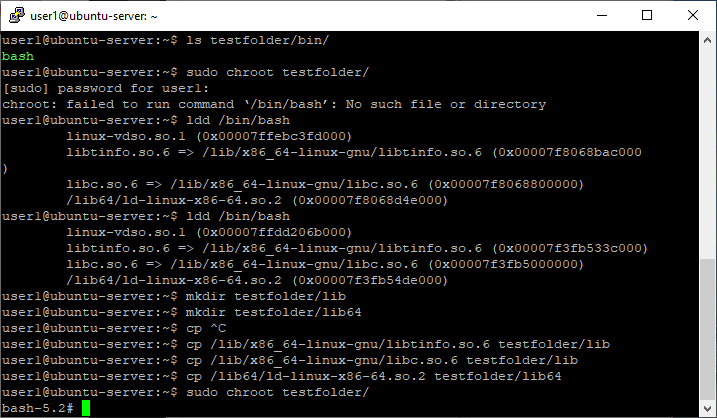
mkdir testfolder/lib

mkdir testfolder/lib64

cp /lib/x86\_64-linux-gnu/libtinfo.so.6 testfolder/lib

cp /lib/x86\_64-linux-gnu/libc.so.6 testfolder/lib

cp /lib64/ld-linux-x86-64.so.2 testfolder/lib64



Изолировали файловую директорию (рисунок выше):

sudo chroot testfolder/

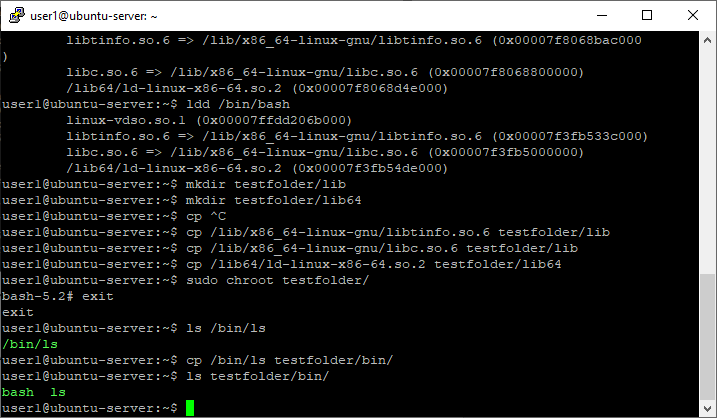
2) Теперь добавим команду ls в нашу изолированную директорию:

Эта команда лежит в каталоге /bin/ls:

ls /bin/ls

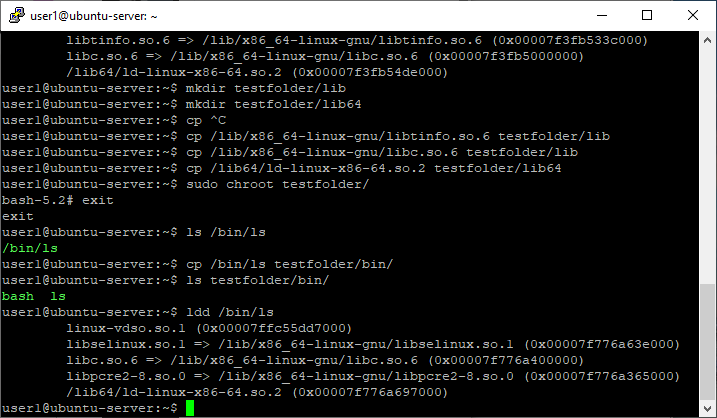
Копируем ее в наш каталог:

cp /bin/ls testfolder/bin/



посмотрим, какие библиотеки нужны будут для работы команды ls:

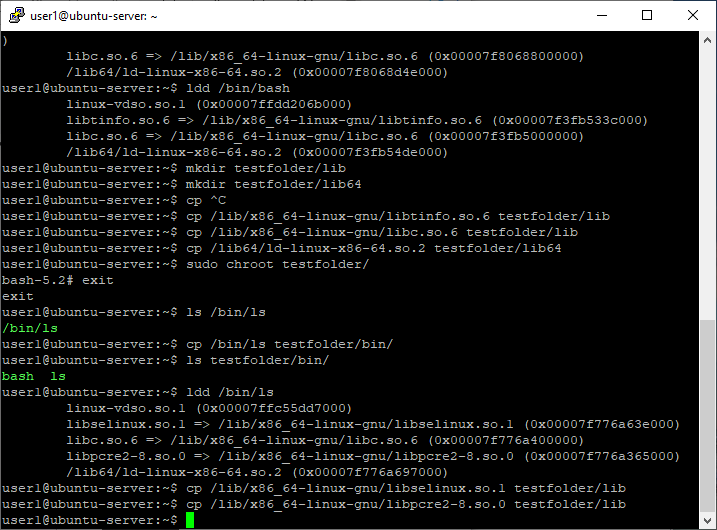
ldd /bin/ls



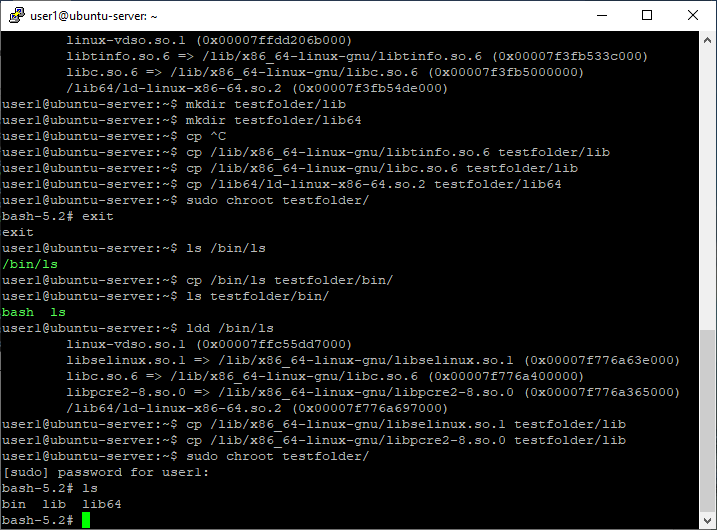
Скопируем недостающие для работы команды ls библиотеки:

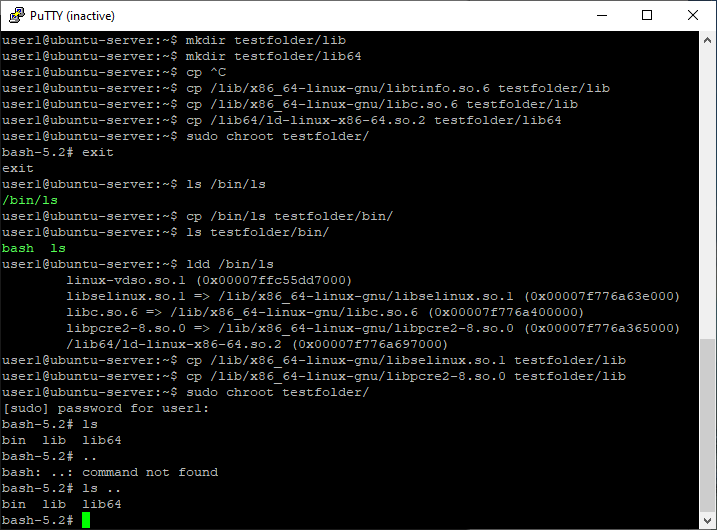
cp /lib/x86\_64-linux-gnu/libselinux.so.1 testfolder/lib

cp /lib/x86\_64-linux-gnu/libpcre2-8.so.0 testfolder/lib

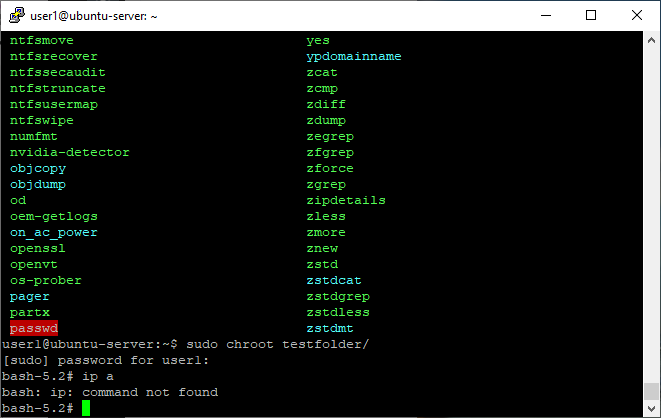


Теперь команда ls заработала



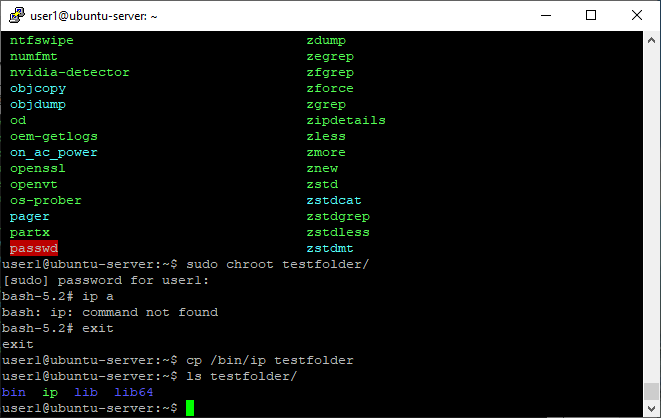


3) скопируем команду ip

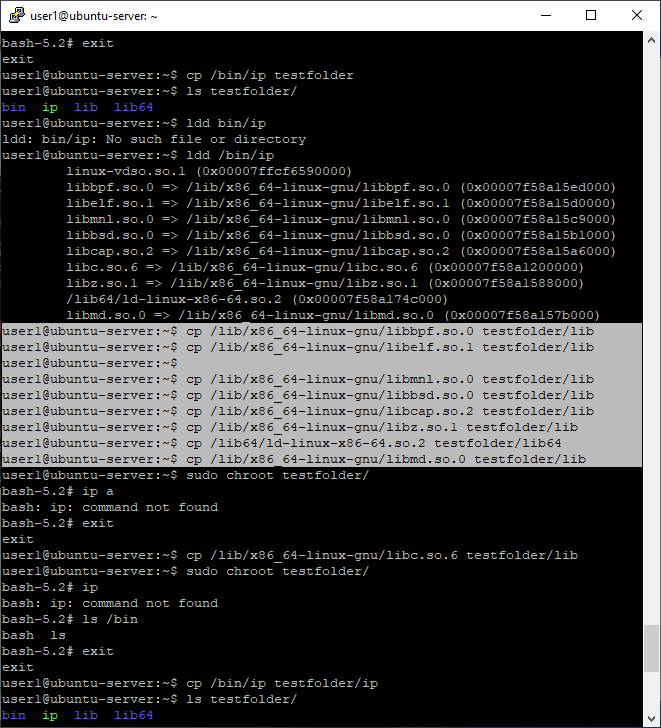


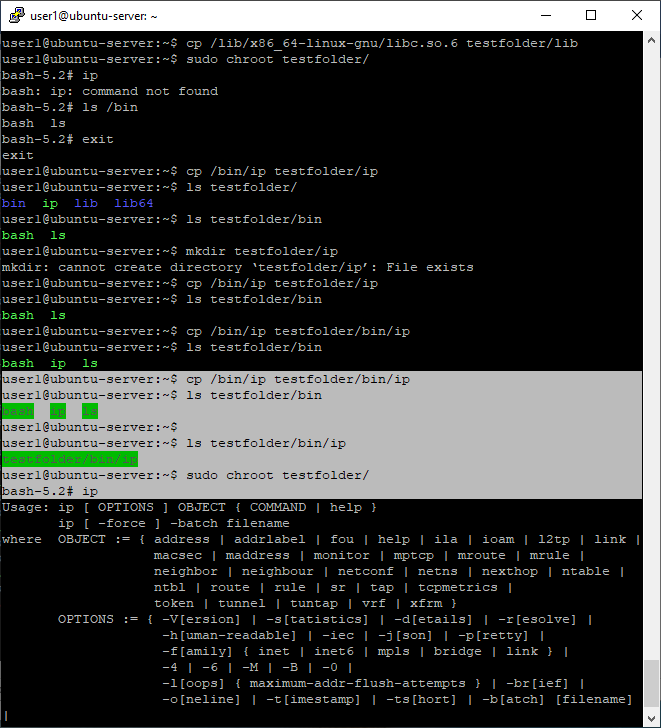
cp /bin/ip testfolder

ls testfolder/



Скопируем библиотеки для ip:

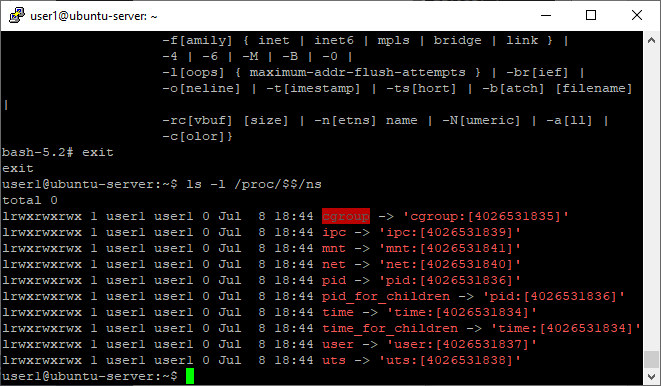




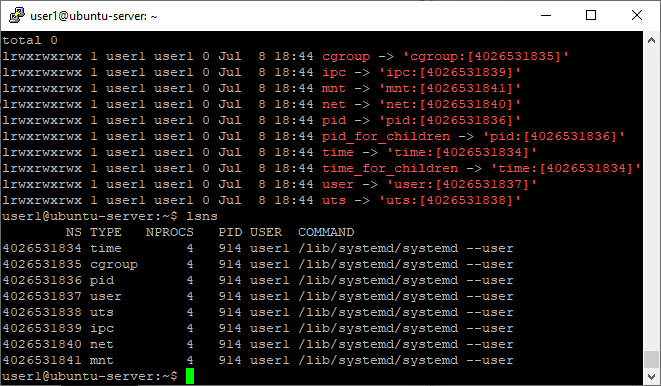
Посмотрим все процессы внутри текущего namespace

ls -l /proc/$$/ns

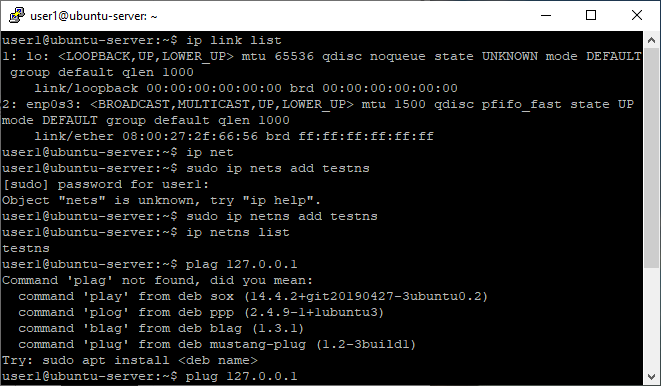
группы namespace:



Посмотреть PID:



2. Создадим namespace с помощью утилиты ip (используется для изолирования сети)



Создали новый namespace:

sudo ip netns add testns

посмотрели созданный namespace:

ip netns list

список интерфейсов в родительском пространстве:

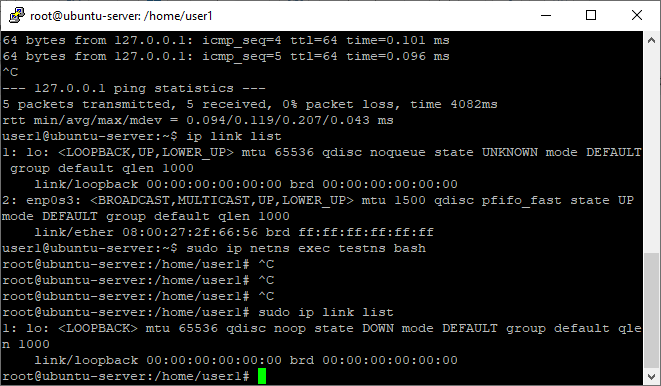
ip link list

проваливаемся в новый созданный интерфейс (здесь мы уже root):

sudo ip netns exec testns bash

и смотрим список интерфейсов:

sudo ip link list

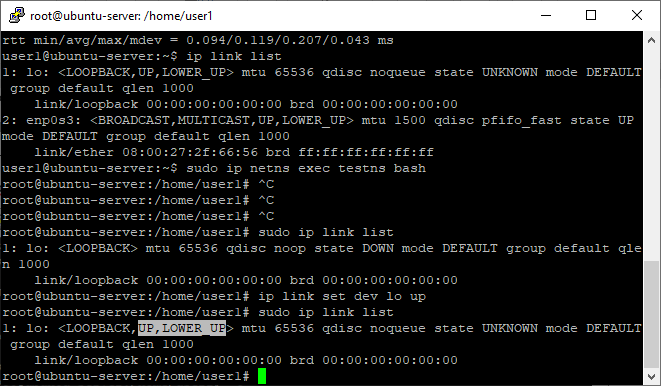


Поднимем наш loopback в новом интерфейсе:

ip link set dev lo up

и посмотрим его статус:

sudo ip link list



Создадим 2 виртуальных Ethernet-порта: один в родительском namespace, другой – в дочернем.

После этого будем гонять пинги между этими двумя пространствами. Пинги к виртуальному интерфейсу родительского namespace будут также приходить к вирт.интерфейсу дочернего, т.е. создадим тоннель.

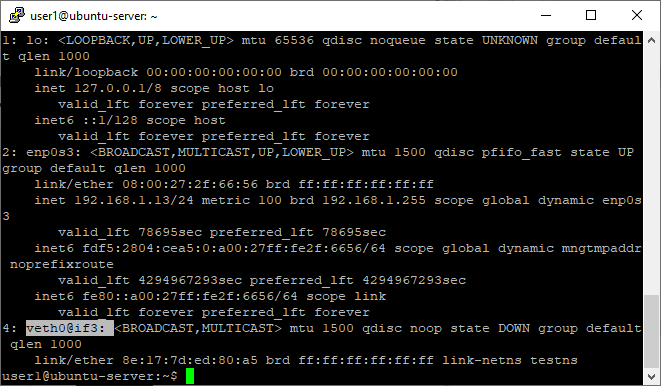
ip link add veth0 type veth peer name veth1 (это выполнить в родительском пространстве (под user1)) – добавили вирт.интерфейс в родительском

прилинкуем его к testns:

sudo ip link set veth1 netns testns

посмотрим новый инерфейс:

ip a

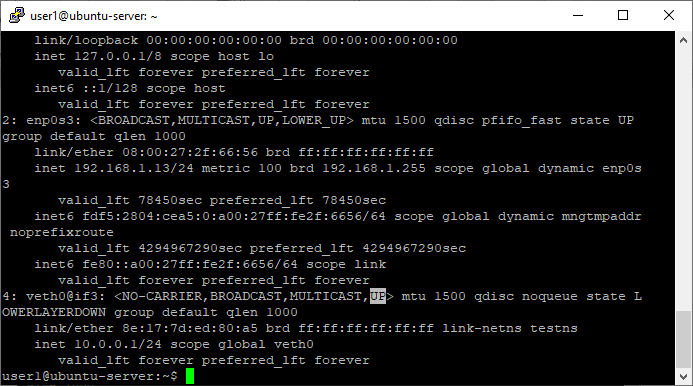


Добавим ему ip-адрес:

sudo ip addr add 10.0.0.1/24 dev veth0

поднимем интерфейс:

sudo ip link set dev veth0 up



Теперь пройдем в наше новое пространство:

sudo ip netns exec testns bash

создадим вирт интерфейс в дочернем:

ip addr add 10.0.0.2/24 dev veth1

поднимем его:

sudo ip link set dev veth1 up

отправим пинг в родительский интерфейс:

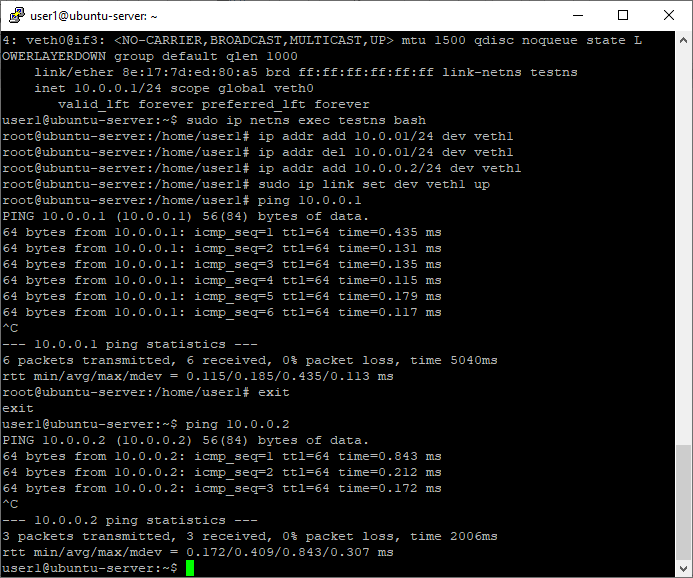
ping 10.0.0.1 (у родительского интерфейса был адрес 10.0.0.1)

перейдем в родительский интерфейс:

exit

и отправим пинг в дочерний интерфейс:

ping 10.0.0.2

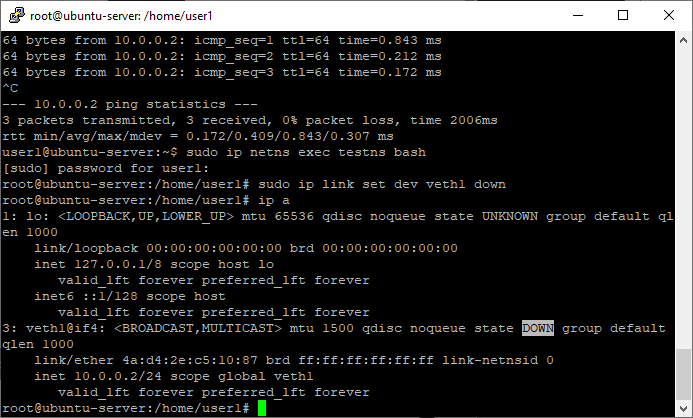


Таким образом, мы между двумя изолированными интерфейсами (контейнерами) настроили сеть.

Положим сеть дочернего интерфейса:

sudo ip netns exec testns bash (переход в дочерний интерфейс)

sudo ip link set dev veth1 down

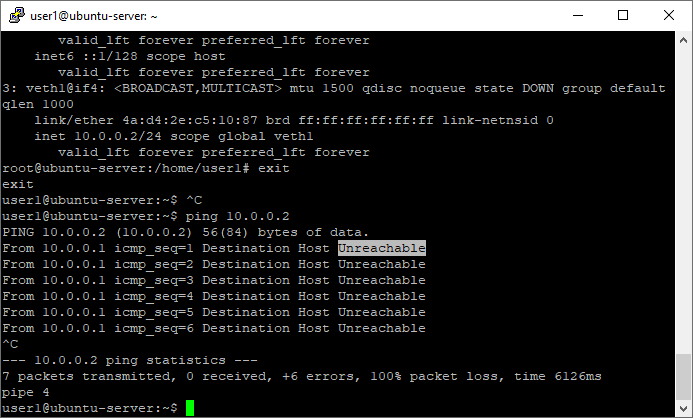


Перейдем в родительский интерфейс и запустим пинг:

exit

ping 10.0.0.2

сеть положили, поэтому даже не находит такой хост:



Удаление виртуальных интерфейсов:

sudo ip netns del testns

проверим список интерфейсов:

ip netns list

Удаление портов у интерфейсов:

ip link dell veth0

3. Утилита unshare для создания и изолирования namespace (универсальная утилита)

Мы создали еще один изолированный дочерний net space, только он неименной, он не сохранится под своим уникальным именем:

sudo unshare --net /bin/bash

при выходе из него (exit), он удаляется, потому что это просто процесс

аналог docker-контейнера без доп возможностей самого docker

sudo unshare --mount --uts --ipc --net --pid --user /bin/bash

