Допустим, нам необходимо создать контрольную группу и работать в ней. Перед началом работы необходимо установить ПО на систему.

|  |
| --- |
| apt install cgroup-tools |

Следом воспользуемся командой, которую изучили на прошлом семинаре - запустим изолированно bash:

|  |
| --- |
| sudo unshare --fork --pid --mount-proc bash |

И проверим, что все действительно идет по плану:

|  |
| --- |
| ps -aux |

Далее - давайте создадим в системе контрольную группу:

|  |
| --- |
| cgcreate -a $USER -g memory:mytestgroup -g cpu:mytestgroup |

Ну и проверим, что она создалась:

|  |
| --- |
| ls /sys/fs/cgroup/mytestgroup/ |

Здесь показать файлы, напомнить о том, что к чему, для чего и куда. Указать ограничение памяти в 200 Мб. Повторно открыть файл и обратить внимание на изменение содержимого.

Далее - вход в группу:

|  |
| --- |
| cgexec -g memory:mygroup bash |

Собственно, таким образом и происходит взаимодействие с контрольными группами. По факту же, сегодня практика больше будет касаться LXD-контейнеров. Почему они:

1. Они максимально похожи на системную обвязку. В частности - на контрольные группы
2. Они также имеют сходство с контейнерами. В частности - с Docker.

## Демо 1

### Задание

Если у вас новая машина, то перед первым использованием необходимо выполнить lxd init

Затем посмотреть выводы:

|  |
| --- |
| lxc storage list  lxc network list  lxc remote list  lxc image -c dasut list ubuntu: | head -n 11  cat /etc/default/lxc-net  cat /etc/lxc/default.conf  ip a (напоследок обязательно) |

Для начала необходимо повторить задание лекции: создать контейнер, сделать ему ограничение в 200Мб ОЗУ.

### Пример решения

|  |
| --- |
| lxc-create -n test123 -t ubuntu  lxc-stop -n test123  vi /var/lib/lxc/test123/config  <<<--->>>  lxc.cgroup2.memory.max = 256M  lxc.start.auto = 1  <<<--->>>  lxc-ls -f  free -m #внутри контейнера |

## Задача 1

### Задание

* Необходимо создать 2 контейнера и показать возможность взаимодействия между собой (командой пинг)

### Пример решения

|  |
| --- |
| lxc-create -n container1 -t ubuntu  lxc-create -n container2 -t ubuntu  << ping from container1 to container2 >> |

## Задача 2

### Задание

Добавить в каждый контейнер статический IP-адрес из той же подсети через config-file, показанный ранее на семинаре.

### Пример решения

Это примерный вариант решения, главное – чтобы логика была верной.

|  |
| --- |
| lxc.net.0.ipv4.address = 10.0.0.10/24 |

## Задача 3

### Тайминг

Объяснить задачу – 3-5 минут

Работа с преподавателем – 10-20 минут

Обсуждение решения - 5 минут

### Задание

Создать контейнер, проверить, что он создался и запустить команду без входа в контейнер, используя lxc-execute

### Пример решения

Это примерный вариант решения, главное – чтобы логика была верной.

|  |
| --- |
| lxc-execute /bin/ls |

## Задача 4

### Задание

Запустить контейнер, обязательно убедиться, что мы внутри контейнера!

Для этого проверить hostname, вывести список запущенных процессов.

Выполнить команду rm -rf /

Дождаться выполнения.

Остановить контейнер и просмотреть логи, которые предлагает lxc.

### Пример решения

Это примерный вариант решения, главное – чтобы логика была верной.

|  |
| --- |
| lxc-attach -n test123  rm -rf /  exit  lxc-stop -n test123  lxc-start -n test123  lxc-start -n test123 –logfile file.log |