

Урок 2. Механизмы контрольных групп

Задание 1:

- 1) запустить контейнер с ubuntu, используя механизм LXC
- 2) ограничить контейнер 256 Мб ОЗУ и проверить, что ограничение работает
- 3) добавить автозапуск контейнеру, перезагрузить ОС и убедиться, что контейнер действительно запустился самостоятельно
- 4) при создании указать файл, куда записывать логи
- 5) после перезагрузки проанализировать логи

Задание 2*: настроить автоматическую маршрутизацию между контейнерами.
Адреса можно взять: 10.0.12.0/24 и 10.0.13.0/24.

Выполнение

- 1) запустить контейнер с ubuntu, используя механизм LXC

`sudo apt update` - обновляем списки пакетов:

`sudo apt upgrade` - обновляем пакеты

Устанавливаем утилиты для работы с LXC контейнерами:

`sudo apt install cgroup-tools`

`sudo apt install lxd-installer`

`sudo apt install lxc-templates`

`sudo lxc init`

`sudo lxc remote list` - смотрим в столбце name (ubuntu)

`sudo lxc-create -n test1 -t ubuntu` - создаем контейнер test1

user1@ubuntu-server: ~

```
Copy /var/cache/lxc/focal/rootfs-amd64 to /var/lib/lxc/test1/rootfs ...
Copying rootfs to /var/lib/lxc/test1/rootfs ...
Generating locales (this might take a while)...
  ru_RU.UTF-8... done
Generation complete.
Creating SSH2 RSA key; this may take some time ...
3072 SHA256:0Nl2OUwmZlNhX3JNhXZN7mK4A6NDkFrgREcc2grklog root@ubuntu-server (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:zH4EyEYsfR/jpodOoCEIQvJlxOp+2MMEzgVSj6bcVmI root@ubuntu-server (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:j+srnMGV6TeMHowQsPZgCCvnviVf+ysfv8HlrDONN5o root@ubuntu-server (ED25519)
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of start.
```

Current default time zone: 'Etc/UTC'

Local time is now: Sun Jul 9 05:44:47 UTC 2023.

Universal Time is now: Sun Jul 9 05:44:47 UTC 2023.

```
##
# The default user is 'ubuntu' with password 'ubuntu'!
# Use the 'sudo' command to run tasks as root in the container.
##
```

sudo lxc-ls - смотрим имя контейнера
sudo lxc-ls -f - информация о контейнере
sudo lxc-start -n test1 - запускаем контейнер test1
sudo lxc-ls -f - информация о контейнере
sudo lxc-attach -n test1 - заходим внутрь контейнера
sudo lxc-stop -n test1 - останавливаем контейнер test1

```
user1@ubuntu-server:~$ sudo lxc-ls
test1
user1@ubuntu-server:~$ sudo lxc-ls -f
NAME    STATE    AUTOSTART GROUPS IPV4 IPV6 UNPRIVILEGED
test1   STOPPED  0         -     -    -    false
user1@ubuntu-server:~$ sudo lxc-start -n test1
user1@ubuntu-server:~$ sudo lxc-ls -f
NAME    STATE    AUTOSTART GROUPS IPV4      IPV6 UNPRIVILEGED
test1   RUNNING  0         -     10.0.3.71 -    false
user1@ubuntu-server:~$ sudo lxc-attach -n test1
root@test1:/# exit
exit
user1@ubuntu-server:~$ sudo lxc-st
sudo: lxc-st: command not found
user1@ubuntu-server:~$ sudo lxc-stop -n test1
user1@ubuntu-server:~$
```

2) Ограничить контейнер 256 Мб ОЗУ и проверить, что ограничение работает

`sudo vi /var/lib/lxc/test1/config` - смотрим файл конфигурации. Для редактирования нажмите "i", после окончания редактирования нажмите "esc".
:wq! - вводим для выхода

```
# Template used to create this container: /usr/share/lxc/templates/lxc-ubuntu
# Parameters passed to the template:
# For additional config options, please look at lxc.container.conf(5)

# Uncomment the following line to support nesting containers:
#lxc.include = /usr/share/lxc/config/nesting.conf
# (Be aware this has security implications)


# Common configuration
lxc.include = /usr/share/lxc/config/ubuntu.common.conf


# Container specific configuration
lxc.rootfs.path = dir:/var/lib/lxc/test1/rootfs
lxc.uts.name = test1
lxc.arch = amd64


# Network configuration
lxc.net.0.type = veth
lxc.net.0.link = lxcbr0
lxc.net.0.flags = up
lxc.net.0.hwaddr = 00:16:3e:7e:51:b
```

```
sudo lxc-start -n test1 - запускаем контейнер test1
sudo lxc-attach -n test1 - заходим внутрь контейнера
free -m - смотрим текущий объем оперативной памяти
exit - выходим
sudo lxc-stop -n test1 - останавливаем контейнер test1
sudo su - заходим под правами суперпользователя
sudo echo "lxc.cgroup2.memory.max = 256" >> /var/lib/lxc/test1/config - задаем
нужные параметры оперативной памяти
exit - выходим
```

```
user1@ubuntu-server:~$ sudo lxc-start -n test1
[sudo] password for user1:
user1@ubuntu-server:~$ sudo lxc-attach -n test1
root@test1:/# free -m
              total            used             free           shared  buff/cache   available
Mem:            5838              31           5758              0             48          5807
Swap:           2047               0           2047
root@test1:/# exit
exit
user1@ubuntu-server:~$ sudo lxc-stop -n test1
user1@ubuntu-server:~$ sudo su
root@ubuntu-server:/home/user1# sudo echo "lxc.cgroup2.memory.max = 256M" >> /var/lib/lxc/test1/config
root@ubuntu-server:/home/user1# exit
exit
user1@ubuntu-server:~$ sudo vi /var/lib/lxc/test1/config
```

`sudo vi /var/lib/lxc/test1/config` - смотрим файл конфигурации и видим объем памяти, равный 256

:wq! - вводим для выхода

```
# Template used to create this container: /usr/share/lxc/templates/lxc-ubuntu
# Parameters passed to the template:
# For additional config options, please look at lxc.container.conf(5)

# Uncomment the following line to support nesting containers:
#lxc.include = /usr/share/lxc/config/nesting.conf
# (Be aware this has security implications)


# Common configuration
lxc.include = /usr/share/lxc/config/ubuntu.common.conf

# Container specific configuration
lxc.rootfs.path = dir:/var/lib/lxc/test1/rootfs
lxc.uts.name = test1
lxc.arch = amd64

# Network configuration
lxc.net.0.type = veth
lxc.net.0.link = lxcbr0
lxc.net.0.flags = up
lxc.net.0.hwaddr = 00:16:3e:7e:51:b8
lxc.cgroup2.memory.max = 256M
~
~
~
~
~
~
~
~
~
~
~
~/var/lib/lxc/test1/config" 23L, 703C
```

```
sudo lxc-start -n test1 - запускаем контейнер test1
```

`sudo lxc-attach -n test1` - заходим внутрь контейнера

free -m - смотрим пекущий объем оперативной памяти

exit - ВЫХОДИМ

```
user1@ubuntu-server:~$ sudo lxc-start -n test1
user1@ubuntu-server:~$ sudo lxc-attach -n test1
root@test1:/# free -m
```

	total	used	free	shared	buff/cache	available
Mem:	5838	27	5810	0	0	5811
Swap:	2047	0	2047			

```
root@test1:/#
```

Далее, не останавливая работу контейнера, можно изменить объем памяти командой:

```
sudo lxc-cgroup -n test1 memory.max 512M
```

```
sudo lxc-attach -n test1 - заходим внутрь контейнера
```

```
free -m - смотрим текущий объем оперативной памяти
```

exit - ВЫХОДИМ

```
sudo lxc-stop -n test1 - останавливаем контейнер test1
```

```
user1@ubuntu-server:~$ sudo lxc-start -n test1
user1@ubuntu-server:~$ sudo lxc-cgroup -n test1 memory.max 512M
lxc-cgroup: test1: tools/lxc_cgroup.c: main: 111 Failed to assign '512M' value to 'memory.max' for 'test1'
user1@ubuntu-server:~$ sudo lxc-attach -n test1
root@test1:/# free -m
```

	total	used	free	shared	buff/cache	available
Mem:	5838	26	5812	0	0	5812
Swap:	2047	0	2047			

```
root@test1:/#
```

3) добавить автозапуск контейнеру, перезагрузить ОС и убедиться, что контейнер действительно запустился самостоятельно

sudo vi /var/lib/lxc/test1/config - добавим автозапуск 1-го контейнера в файл конфигурации. Для редактирования нажмите "i".

lxc.start.auto = 1 - вписываем строку с автозапуском

После окончания редактирования нажмите "esc" и для выхода из редактора :wq!

```
# Template used to create this container: /usr/share/lxc/templates/lxc-ubuntu
# Parameters passed to the template:
# For additional config options, please look at lxc.container.conf(5)

# Uncomment the following line to support nesting containers:
#lxc.include = /usr/share/lxc/config/nesting.conf
# (Be aware this has security implications)

# Common configuration
lxc.include = /usr/share/lxc/config/ubuntu.common.conf

# Container specific configuration
lxc.rootfs.path = dir:/var/lib/lxc/test1/rootfs
lxc.uts.name = test1
lxc.arch = amd64
lxc.start.auto = 1
```

Убеждаемся, что контейнер действительно запустился самостоятельно

```
user1@ubuntu-server:~$ sudo lxc-ls -f
[sudo] password for user1:
NAME    STATE    AUTOSTART GROUPS IPV4        IPV6 UNPRIVILEGED
test1   RUNNING  1         -      10.0.3.71   -      false
test2   STOPPED  0         -      -           -      false
user1@ubuntu-server:~$
```

4) При создании контейнера указать файл, куда записывать логи

`sudo lxc-start -n test1 -o "/user/home/logfile" -l debug` - создаем файл

`sudo cat /user/home/logfile` - открываем файл с логами

Если нужно посмотреть строки, где, например, ошибки ERROR, то указываем `sudo cat /user/home/logfile | grep "ERROR"`

```
user1@ubuntu-server:~$ sudo lxc-start -n test1 -o "/user/home/logfile" -l debug
user1@ubuntu-server:~$ sudo cat /user/home/logfile
lxc-start test1 20230709113201.259 INFO      lxccontainer - lxccontainer.c:do_lxcapi_start:997 - Set process title to [lxc monitor] /var
/lib/lxc test1
```

Задание 2*: настроить автоматическую маршрутизацию между контейнерами.
Адреса можно взять: 10.0.12.0/24 и 10.0.13.0/24.

`sudo lxc-start -n test2` - запускаем контейнер test2

`sudo lxc-attach -n test2` - заходим внутрь контейнера

`ip a` - проверяем наличие сети

`exit` - выходим

`sudo lxc-stop -n test2` - останавливаем контейнер test2

`sudo su` - заходим под правами суперпользователя

`echo "lxc.net.0.ipv4.address = 10.0.0.10/24" >> /var/lib/lxc/test1/config` - создаем подсеть в первом контейнере

`echo "lxc.net.0.ipv4.address = 10.0.0.20/24" >> /var/lib/lxc/test2/config` - создаем подсеть во втором контейнере (тестовой машине)

`exit` - выходим

```

user1@ubuntu-server:~$ sudo lxc-start -n test2
user1@ubuntu-server:~$ sudo lxc-attach -n test2
root@test2:/# 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
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0@if16: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 00:16:3e:27:bd:8c brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.3.8/24 brd 10.0.3.255 scope global dynamic eth0
        valid_lft 3594sec preferred_lft 3594sec
    inet6 fe80::216:3eff:fe27:bd8c/64 scope link
        valid_lft forever preferred_lft forever
root@test2:/# exit
exit
user1@ubuntu-server:~$ sudo su
root@ubuntu-server:/home/user1# echo "lxc.net.0.ipv4.address = 10.0.0.10/24" >> /var/lib/lxc/test1/config
root@ubuntu-server:/home/user1# echo "lxc.net.0.ipv4.address = 10.0.0.20/24" >> /var/lib/lxc/test2/config
root@ubuntu-server:/home/user1# exit
exit
user1@ubuntu-server:~$ sudo vi /var/lib/lxc/test1/config

```

`sudo vi /var/lib/lxc/test1/config` - смотрим файл конфигурации 1-го контейнера (созданную подсеть). Для редактирования нажмите "i", после окончания редактирования нажмите "esc".

:wq! - вводим для выхода

```

# Template used to create this container: /usr/share/lxc/templates/lxc-ubuntu
# Parameters passed to the template:
# For additional config options, please look at lxc.container.conf(5)

# Uncomment the following line to support nesting containers:
#lxc.include = /usr/share/lxc/config/nesting.conf
# (Be aware this has security implications)

# Common configuration
lxc.include = /usr/share/lxc/config/ubuntu.common.conf

# Container specific configuration
lxc.rootfs.path = dir:/var/lib/lxc/test1/rootfs
lxc.uts.name = test1
lxc.arch = amd64

# Network configuration
lxc.net.0.type = veth
lxc.net.0.link = lxcbr0
lxc.net.0.flags = up
lxc.net.0.hwaddr = 00:16:3e:7e:51:b8
lxc.net.0.ipv4.address = 10.0.0.10/24

```

`sudo vi /var/lib/lxc/test2/config` - смотрим файл конфигурации 2-го контейнера (созданную подсеть). Для редактирования нажмите "i", после окончания редактирования нажмите "esc".

:wq! - вводим для выхода

```

# Template used to create this container: /usr/share/lxc/templates/lxc-ubuntu
# Parameters passed to the template:
# For additional config options, please look at lxc.container.conf(5)

# Uncomment the following line to support nesting containers:
#lxc.include = /usr/share/lxc/config/nesting.conf
# (Be aware this has security implications)

# Common configuration
lxc.include = /usr/share/lxc/config/ubuntu.common.conf

# Container specific configuration
lxc.rootfs.path = dir:/var/lib/lxc/test2/rootfs
lxc.uts.name = test2
lxc.arch = amd64

# Network configuration
lxc.net.0.type = veth
lxc.net.0.link = lxcbr0
lxc.net.0.flags = up
lxc.net.0.hwaddr = 00:16:3e:27:bd:8c
lxc.net.0.ipv4.address = 10.0.0.20/24

```

sudo lxc-start -n test1 - запускаем контейнер test1

sudo lxc-start -n test2 - запускаем контейнер test2

sudo lxc-ls -f - проверяем содержимое (подсети)

!!!Не отобразились созданные подсети

```

user1@ubuntu-server:~$ sudo lxc-ls -f
NAME    STATE    AUTOSTART  GROUPS  IPV4        IPV6  UNPRIVILEGED
test1   RUNNING  0          -       10.0.3.71   -     false
test2   RUNNING  0          -       10.0.3.8    -     false
user1@ubuntu-server:~$

```