EPAM University Programs

DevOps external course

Module 2 Virtualization and Cloud Basic

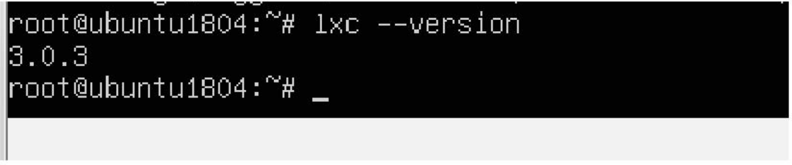
TASK 2.4

Работа с lxc в Ubuntu

Documentation - https://help.ubuntu.com/lts/serverguide/lxd.html

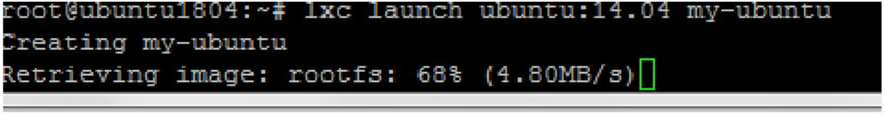
https://linuxcontainers.org/lxd/getting-started-cli/

1. Установить lxc (screenshot)



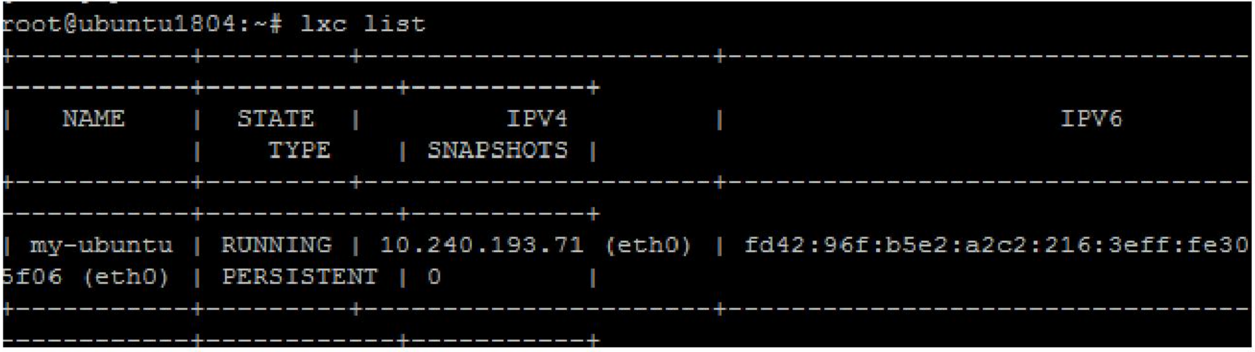
```
root@ubuntu1804:~# lxc --version
3.0.3
root@ubuntu1804:~#
```

2. Запустить lxc launch для любой из версий Убунту (screenshot)



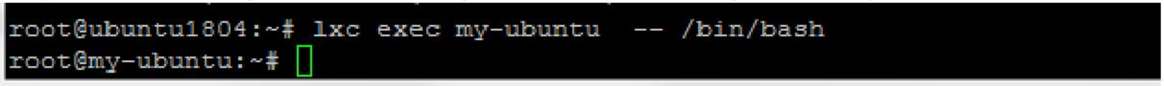
```
root@ubuntu1804:~# lxc launch ubuntu:14.04 my-ubuntu
Creating my-ubuntu
Retrieving image: rootfs: 68% (4.80MB/s)
```

3. По окончании загрузки убедиться, что машина стартовала lxc list (screenshot)



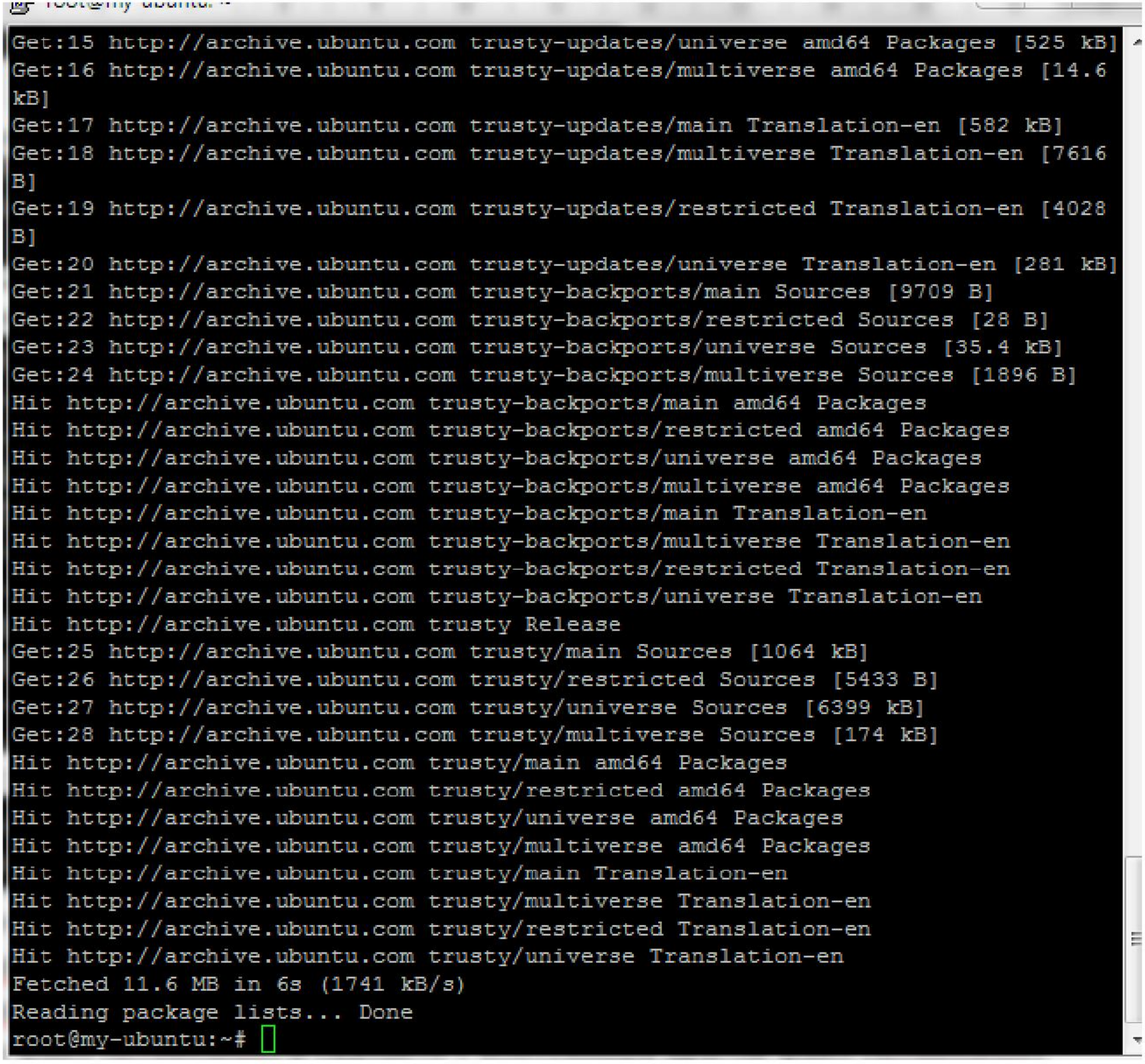
```
root@ubuntu1804:~# lxc list
+-----------+---------+----------------------+-----------------------------------
-----------+------------+-----------+
|   NAME    |  STATE  |         IPV4         |                              IPV6
           |    TYPE    | SNAPSHOTS |
+-----------+---------+----------------------+-----------------------------------
-----------+------------+-----------+
| my-ubuntu | RUNNING | 10.240.193.71 (eth0) | fd42:96f:b5e2:a2c2:216:3eff:fe30
5f06 (eth0) | PERSISTENT | 0         |
+-----------+---------+----------------------+-----------------------------------
-----------+------------+-----------+
```

4. Зайдите в контейнер с командной строкой bash /bin/bash (screenshot)



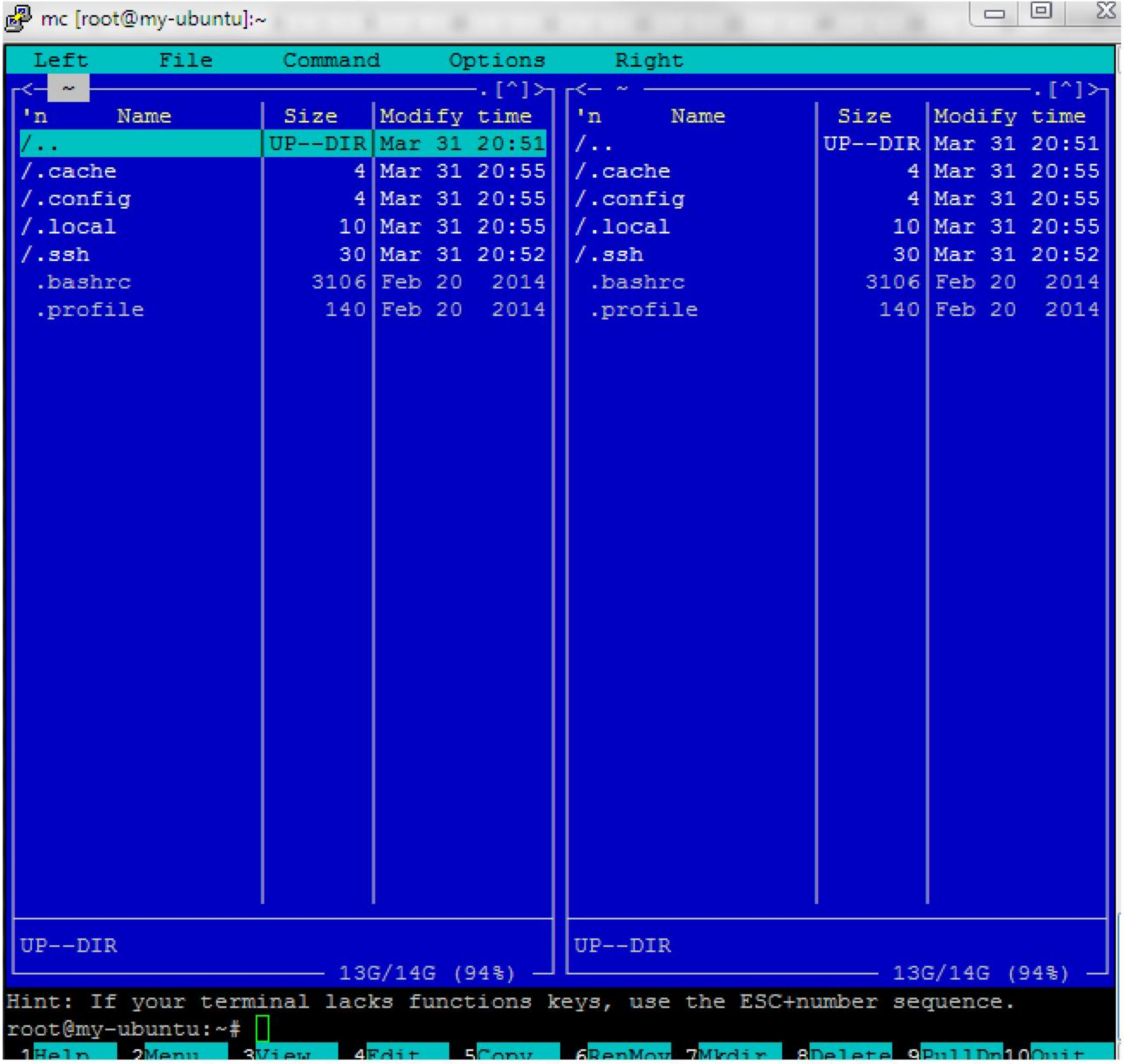
```
root@ubuntu1804:~# lxc exec my-ubuntu  -- /bin/bash
root@my-ubuntu:~#
```

5. Запустите обновление apt-get update (screenshot)



```
Get:15 http://archive.ubuntu.com trusty-updates/universe amd64 Packages [525 kB]
Get:16 http://archive.ubuntu.com trusty-updates/multiverse amd64 Packages [14.6
kB]
Get:17 http://archive.ubuntu.com trusty-updates/main Translation-en [582 kB]
Get:18 http://archive.ubuntu.com trusty-updates/multiverse Translation-en [7616
B]
Get:19 http://archive.ubuntu.com trusty-updates/restricted Translation-en [4028
B]
Get:20 http://archive.ubuntu.com trusty-updates/universe Translation-en [281 kB]
Get:21 http://archive.ubuntu.com trusty-backports/main Sources [9709 B]
Get:22 http://archive.ubuntu.com trusty-backports/restricted Sources [28 B]
Get:23 http://archive.ubuntu.com trusty-backports/universe Sources [35.4 kB]
Get:24 http://archive.ubuntu.com trusty-backports/multiverse Sources [1896 B]
Hit http://archive.ubuntu.com trusty-backports/main amd64 Packages
Hit http://archive.ubuntu.com trusty-backports/restricted amd64 Packages
Hit http://archive.ubuntu.com trusty-backports/universe amd64 Packages
Hit http://archive.ubuntu.com trusty-backports/multiverse amd64 Packages
Hit http://archive.ubuntu.com trusty-backports/main Translation-en
Hit http://archive.ubuntu.com trusty-backports/multiverse Translation-en
Hit http://archive.ubuntu.com trusty-backports/restricted Translation-en
Hit http://archive.ubuntu.com trusty-backports/universe Translation-en
Hit http://archive.ubuntu.com trusty Release
Get:25 http://archive.ubuntu.com trusty/main Sources [1064 kB]
Get:26 http://archive.ubuntu.com trusty/restricted Sources [5433 B]
Get:27 http://archive.ubuntu.com trusty/universe Sources [6399 kB]
Get:28 http://archive.ubuntu.com trusty/multiverse Sources [174 kB]
Hit http://archive.ubuntu.com trusty/main amd64 Packages
Hit http://archive.ubuntu.com trusty/restricted amd64 Packages
Hit http://archive.ubuntu.com trusty/universe amd64 Packages
Hit http://archive.ubuntu.com trusty/multiverse amd64 Packages
Hit http://archive.ubuntu.com trusty/main Translation-en
Hit http://archive.ubuntu.com trusty/multiverse Translation-en
Hit http://archive.ubuntu.com trusty/restricted Translation-en
Hit http://archive.ubuntu.com trusty/universe Translation-en
Fetched 11.6 MB in 6s (1741 kB/s)
Reading package lists... Done
root@my-ubuntu:~#
```

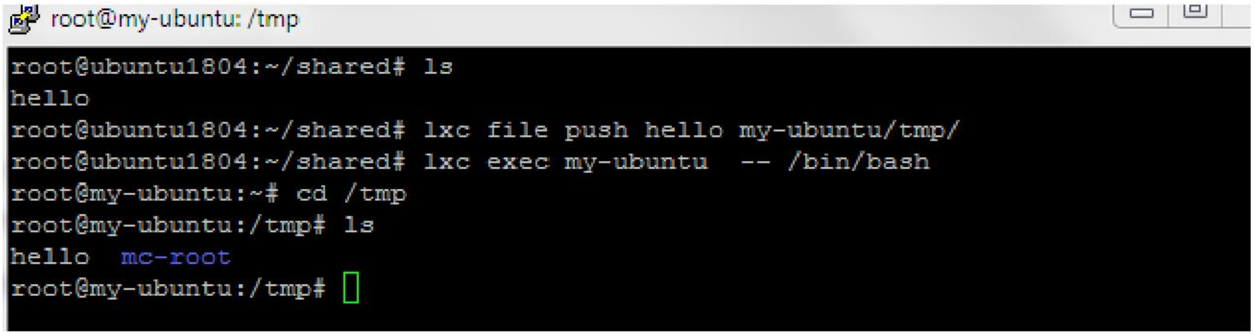
6. Установите (apt-get install) любую программу в контейнер. Например mc. Проверьте работоспособность. (screenshot)



mc [root@my-ubuntu]:~

```
  Left       File       Command      Options      Right
┌<─ ~ ──────────────────────────────────.[^]>┐┌<─ ~ ──────────────────────────────────.[^]>┐
│'n      Name          │ Size   │Modify time  ││'n      Name          │ Size   │Modify time  │
│/..                   │UP--DIR │Mar 31 20:51 ││/..                   │UP--DIR │Mar 31 20:51 │
│/.cache               │      4 │Mar 31 20:55 ││/.cache               │      4 │Mar 31 20:55 │
│/.config              │      4 │Mar 31 20:55 ││/.config              │      4 │Mar 31 20:55 │
│/.local               │     10 │Mar 31 20:55 ││/.local               │     10 │Mar 31 20:55 │
│/.ssh                 │     30 │Mar 31 20:52 ││/.ssh                 │     30 │Mar 31 20:52 │
│ .bashrc              │   3106 │Feb 20  2014 ││ .bashrc              │   3106 │Feb 20  2014 │
│ .profile             │    140 │Feb 20  2014 ││ .profile             │    140 │Feb 20  2014 │
├──────────────────────────────────────────────┤├──────────────────────────────────────────────┤
│UP--DIR                                       ││UP--DIR                                       │
└───────────────────────────── 13G/14G (94%) ─┘└───────────────────────────── 13G/14G (94%) ─┘
Hint: If your terminal lacks functions keys, use the ESC+number sequence.
root@my-ubuntu:~#
 1Help  2Menu  3View  4Edit  5Copy  6RenMov 7Mkdir  8Delete 9PullDn10Quit
```

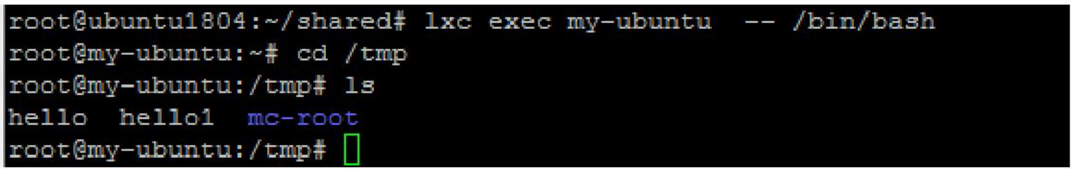
7. Загрузите в контейнер файл (screenshot) и скачайте с контейнера другой файл (screenshot).



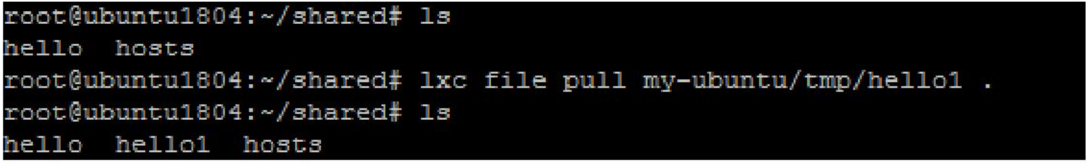
root@my-ubuntu: /tmp

```
root@ubuntu1804:~/shared# ls
hello
root@ubuntu1804:~/shared# lxc file push hello my-ubuntu/tmp/
root@ubuntu1804:~/shared# lxc exec my-ubuntu  -- /bin/bash
root@my-ubuntu:~# cd /tmp
root@my-ubuntu:/tmp# ls
hello  mc-root
root@my-ubuntu:/tmp#
```

создали файл, вытягиваем его



```
root@ubuntu1804:~/shared# lxc exec my-ubuntu  -- /bin/bash
root@my-ubuntu:~# cd /tmp
root@my-ubuntu:/tmp# ls
hello  hello1  mc-root
root@my-ubuntu:/tmp# 
```



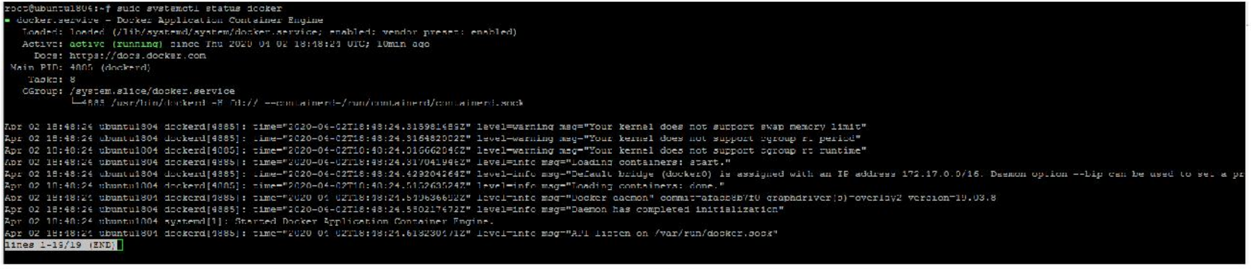
```
root@ubuntu1804:~/shared# ls
hello  hosts
root@ubuntu1804:~/shared# lxc file pull my-ubuntu/tmp/hello1 .
root@ubuntu1804:~/shared# ls
hello  hello1  hosts
```

Работа с Docker в Ubuntu

Documentation - https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04

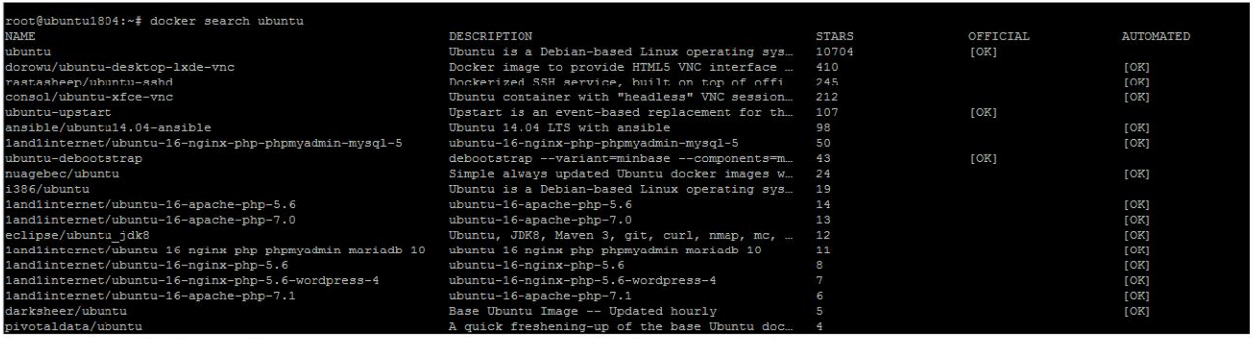
https://docs.docker.com

1. Установить docker (screenshot)



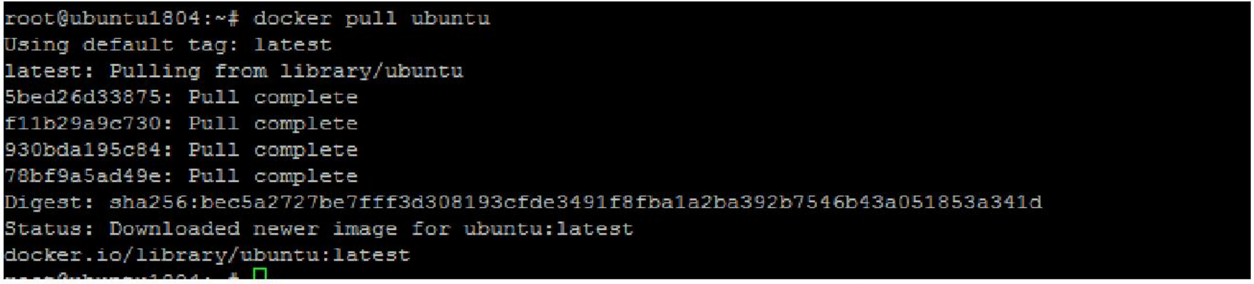
```
root@ubuntu1804:~# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2020-04-02 18:48:24 UTC; 10min ago
     Docs: https://docs.docker.com
 Main PID: 4885 (dockerd)
    Tasks: 8
   CGroup: /system.slice/docker.service
           └─4885 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Apr 02 18:48:24 ubuntu1804 dockerd[4885]: time="2020-04-02T18:48:24.315981489Z" level=warning msg="Your kernel does not support swap memory limit"
Apr 02 18:48:24 ubuntu1804 dockerd[4885]: time="2020-04-02T18:48:24.316482002Z" level=warning msg="Your kernel does not support cgroup rt period"
Apr 02 18:48:24 ubuntu1804 dockerd[4885]: time="2020-04-02T18:48:24.316662046Z" level=warning msg="Your kernel does not support cgroup rt runtime"
Apr 02 18:48:24 ubuntu1804 dockerd[4885]: time="2020-04-02T18:48:24.317041946Z" level=info msg="Loading containers: start."
Apr 02 18:48:24 ubuntu1804 dockerd[4885]: time="2020-04-02T18:48:24.429204264Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daemon option --bip can be used to set a pr
Apr 02 18:48:24 ubuntu1804 dockerd[4885]: time="2020-04-02T18:48:24.515263524Z" level=info msg="Loading containers: done."
Apr 02 18:48:24 ubuntu1804 dockerd[4885]: time="2020-04-02T18:48:24.519636652Z" level=info msg="Docker daemon" commit=afacb8b7f0 graphdriver(s)=overlay2 version=19.03.8
Apr 02 18:48:24 ubuntu1804 dockerd[4885]: time="2020-04-02T18:48:24.550217472Z" level=info msg="Daemon has completed initialization"
Apr 02 18:48:24 ubuntu1804 systemd[1]: Started Docker Application Container Engine.
Apr 02 18:48:24 ubuntu1804 dockerd[4885]: time="2020-04-02T18:48:24.613230171Z" level=info msg="API listen on /var/run/docker.sock"
lines 1-19/19 (END)
```

2. Запустить поиск сконфигурированных решений для “ubuntu”(screenshot)



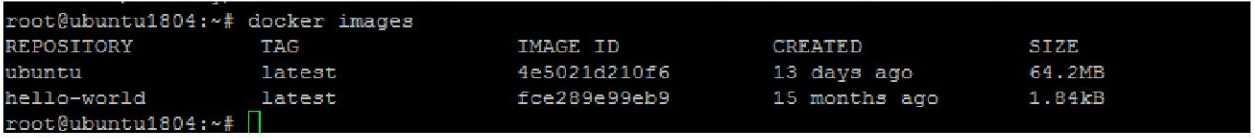
```
root@ubuntu1804:~# docker search ubuntu
NAME                                                      DESCRIPTION                                     STARS               OFFICIAL            AUTOMATED
ubuntu                                                    Ubuntu is a Debian-based Linux operating sys…   10704               [OK]
dorowu/ubuntu-desktop-lxde-vnc                            Docker image to provide HTML5 VNC interface …   410                                     [OK]
rastasheep/ubuntu-sshd                                    Dockerized SSH service, built on top of offi…   245                                     [OK]
consol/ubuntu-xfce-vnc                                    Ubuntu container with "headless" VNC session…   212                                     [OK]
ubuntu-upstart                                            Upstart is an event-based replacement for th…   107                 [OK]
ansible/ubuntu14.04-ansible                               Ubuntu 14.04 LTS with ansible                   98                                      [OK]
1and1internet/ubuntu-16-nginx-php-phpmyadmin-mysql-5      ubuntu-16-nginx-php-phpmyadmin-mysql-5          50                                      [OK]
ubuntu-debootstrap                                        debootstrap --variant=minbase --components=m…   43                  [OK]
nuagebec/ubuntu                                           Simple always updated Ubuntu docker images w…   24                                      [OK]
i386/ubuntu                                               Ubuntu is a Debian-based Linux operating sys…   19
1and1internet/ubuntu-16-apache-php-5.6                    ubuntu-16-apache-php-5.6                        14                                      [OK]
1and1internet/ubuntu-16-apache-php-7.0                    ubuntu-16-apache-php-7.0                        13                                      [OK]
eclipse/ubuntu_jdk8                                       Ubuntu, JDK8, Maven 3, git, curl, nmap, mc, …   12                                      [OK]
1and1internet/ubuntu-16-nginx-php-phpmyadmin-mariadb-10   ubuntu-16-nginx-php-phpmyadmin-mariadb-10       11                                      [OK]
1and1internet/ubuntu-16-nginx-php-5.6                     ubuntu-16-nginx-php-5.6                         8                                       [OK]
1and1internet/ubuntu-16-nginx-php-5.6-wordpress-4         ubuntu-16-nginx-php-5.6-wordpress-4             7                                       [OK]
1and1internet/ubuntu-16-apache-php-7.1                    ubuntu-16-apache-php-7.1                        6                                       [OK]
darksheer/ubuntu                                          Base Ubuntu Image -- Updated hourly             5                                       [OK]
pivotaldata/ubuntu                                        A quick freshening-up of the base Ubuntu doc…   4
```

3. Скачать любой из образов на локальную машину. (screenshot)



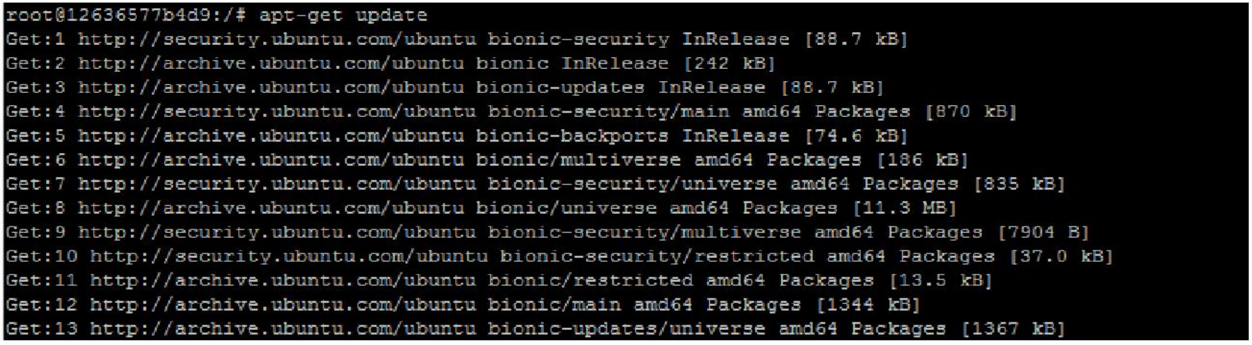
```
root@ubuntu1804:~# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
5bed26d33875: Pull complete
f11b29a9c730: Pull complete
930bda195c84: Pull complete
78bf9a5ad49e: Pull complete
Digest: sha256:bec5a2727be7fff3d308193cfde3491f8fba1a2ba392b7546b43a051853a341d
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
root@ubuntu1804:~# 
```

4. Запустить команду просмотра загруженных на компьютер образов. (screenshot)



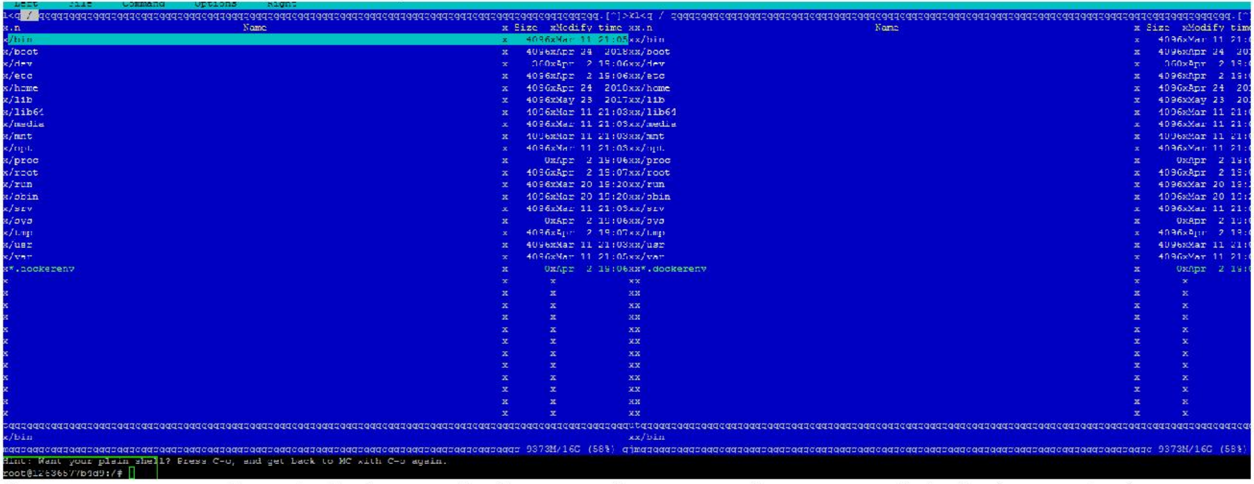
```
root@ubuntu1804:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              4e5021d210f6        13 days ago         64.2MB
hello-world         latest              fce289e99eb9        15 months ago       1.84kB
root@ubuntu1804:~# 
```

5. Запустите обновление apt-get update (screenshot)



```
root@12636577b4d9:/# apt-get update
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [870 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [186 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [835 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [7904 B]
Get:10 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [37.0 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1367 kB]
```

6. Установите (apt-get install) любую программу в контейнер. Например mc. Проверьте работоспособность. (screenshot)

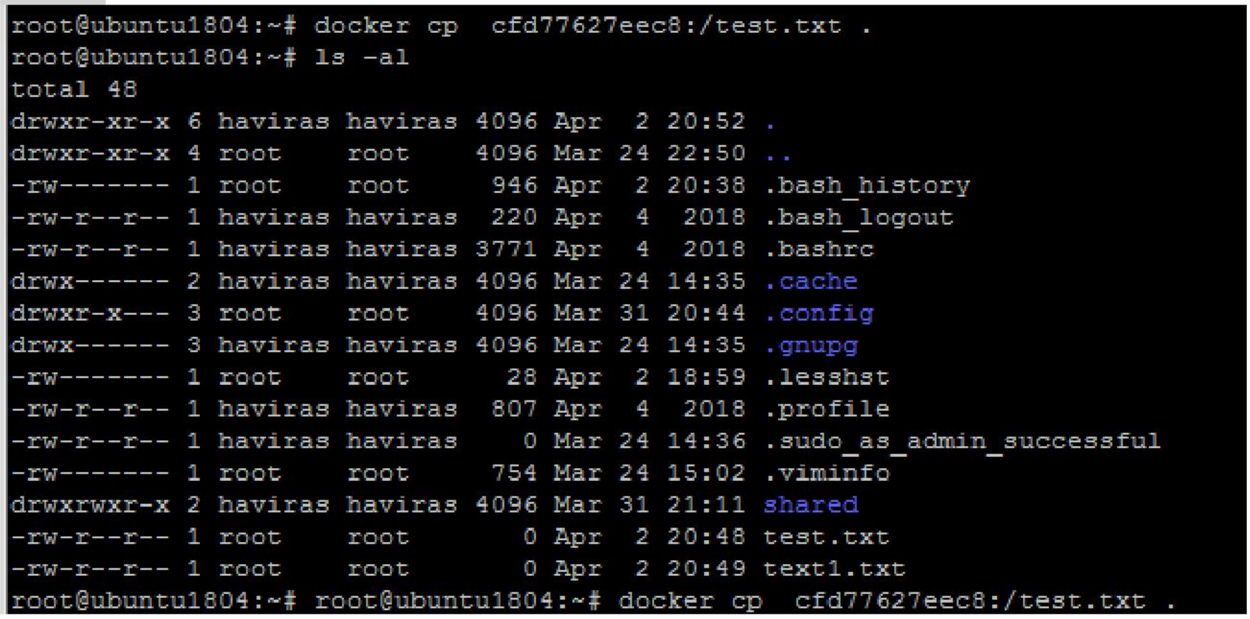


```
.n                              Name                  Size  Modify time
/bin                                                  4096 Mar 11 21:05
/boot                                                 4096 Apr 24  2018
/dev                                                   360 Apr  2 19:06
/etc                                                  4096 Apr  2 19:06
/home                                                 4096 Apr 24  2018
/lib                                                  4096 May 23  2017
/lib64                                                4096 Mar 11 21:03
/media                                                4096 Mar 11 21:03
/mnt                                                  4096 Mar 11 21:03
/opt                                                  4096 Mar 11 21:03
/proc                                                    0 Apr  2 19:06
/root                                                 4096 Apr  2 19:07
/run                                                  4096 Mar 20 19:20
/sbin                                                 4096 Mar 20 19:20
/srv                                                  4096 Mar 11 21:03
/sys                                                     0 Apr  2 19:06
/tmp                                                  4096 Apr  2 19:07
/usr                                                  4096 Mar 11 21:03
/var                                                  4096 Mar 11 21:05
*.dockerenv                                              0 Apr  2 19:06
/bin
9373M/16G (58%)
Hint: Want your plain shell? Press C-o, and get back to MC with C-o again.
root@12636577b4d9:/#
```

7. Загрузите в контейнер файл (screenshot) и скачайте с контейнера другой файл (screenshot).

```
root@ubuntu1804:~# docker run -itd ubuntu

cfd77627eec864e9a2db3f292541a48a24915f15b81434fa6f66016a6cabf4e9

root@ubuntu1804:~# docker ps

CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS               NAMES


cfd77627eec8        ubuntu              "/bin/bash"         3 seconds ago       Up 1 second
hungry_moore

root@ubuntu1804:~# docker exec -it cfd77627eec8 sh

# id

uid=0(root) gid=0(root) groups=0(root)

# exit

root@ubuntu1804:~# docker ps

CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS               NAMES


cfd77627eec8        ubuntu              "/bin/bash"         22 seconds ago      Up 20 seconds
hungry_moore

root@ubuntu1804:~# docker exec -it cfd77627eec8 sh
```

```
# touch test.txt

# ls -a

.  ..  .dockerenv  bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin
srv  sys  test.txt  tmp  usr  var

# ls -al

total 72

-rw-r--r--   1 root root    0 Apr  2 20:48 test.txt

# exit

root@ubuntu1804:~# docker ps -a

CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS               NAMES

cfd77627eec8        ubuntu              "/bin/bash"         About a minute ago   Up About a minute
hungry_moore

root@ubuntu1804:~# touch text1.txt

root@ubuntu1804:~# docker cp text1.txt cfd77627eec8:/root

root@ubuntu1804:~# docker exec -it cfd77627eec8 sh

# ls -al

total 72

-rw-r--r--   1 root root    0 Apr  2 20:48 test.txt

# cd /root

# ls

text1.txt
```



```
root@ubuntu1804:~# docker cp  cfd77627eec8:/test.txt .
root@ubuntu1804:~# ls -al
total 48
drwxr-xr-x 6 haviras haviras 4096 Apr  2 20:52 .
drwxr-xr-x 4 root    root    4096 Mar 24 22:50 ..
-rw------- 1 root    root     946 Apr  2 20:38 .bash_history
-rw-r--r-- 1 haviras haviras  220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 haviras haviras 3771 Apr  4  2018 .bashrc
drwx------ 2 haviras haviras 4096 Mar 24 14:35 .cache
drwxr-x--- 3 root    root    4096 Mar 31 20:44 .config
drwx------ 3 haviras haviras 4096 Mar 24 14:35 .gnupg
-rw------- 1 root    root      28 Apr  2 18:59 .lesshst
-rw-r--r-- 1 haviras haviras  807 Apr  4  2018 .profile
-rw-r--r-- 1 haviras haviras    0 Mar 24 14:36 .sudo_as_admin_successful
-rw------- 1 root    root     754 Mar 24 15:02 .viminfo
drwxrwxr-x 2 haviras haviras 4096 Mar 31 21:11 shared
-rw-r--r-- 1 root    root       0 Apr  2 20:48 test.txt
-rw-r--r-- 1 root    root       0 Apr  2 20:49 text1.txt
root@ubuntu1804:~# root@ubuntu1804:~# docker cp  cfd77627eec8:/test.txt .
```

Прочитать документацию и кратко описать основные 7 команд Dockerfile

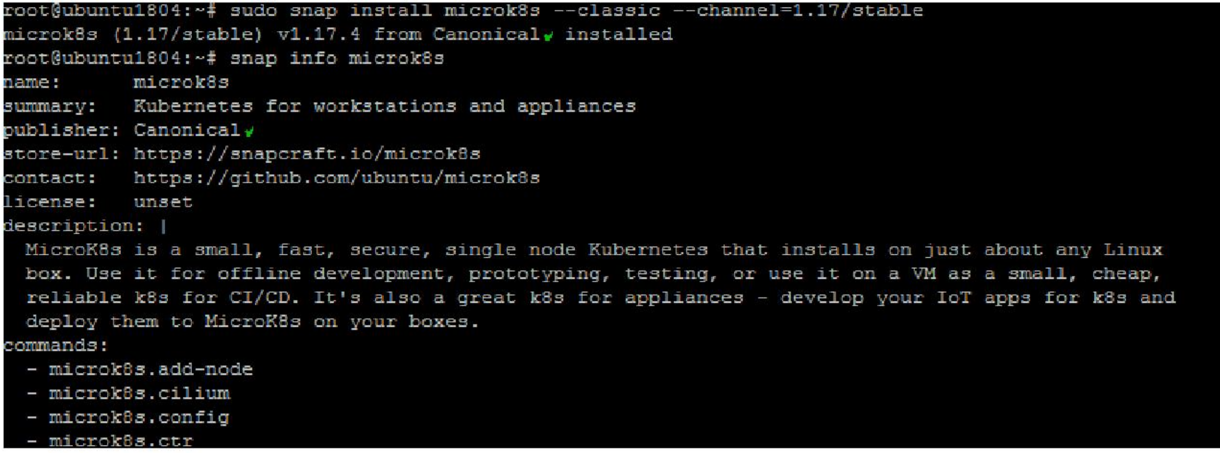
exec – запустить в существующем контейнере команду
attach - зайти в существующий контейнер
kill – остановить контейнер
create - создать контейнер
images – посмотреть доступные образы в репозитории
pull – загрузить шаблоны из репозитория
ps – просмотр состояния контейнеров
rm – удалить контейнер
run – запустить команды в контейнере
commit – создать новый образ из измнений в первоначальном контейнере

8.

Работа с Kubernetes в Ubuntu

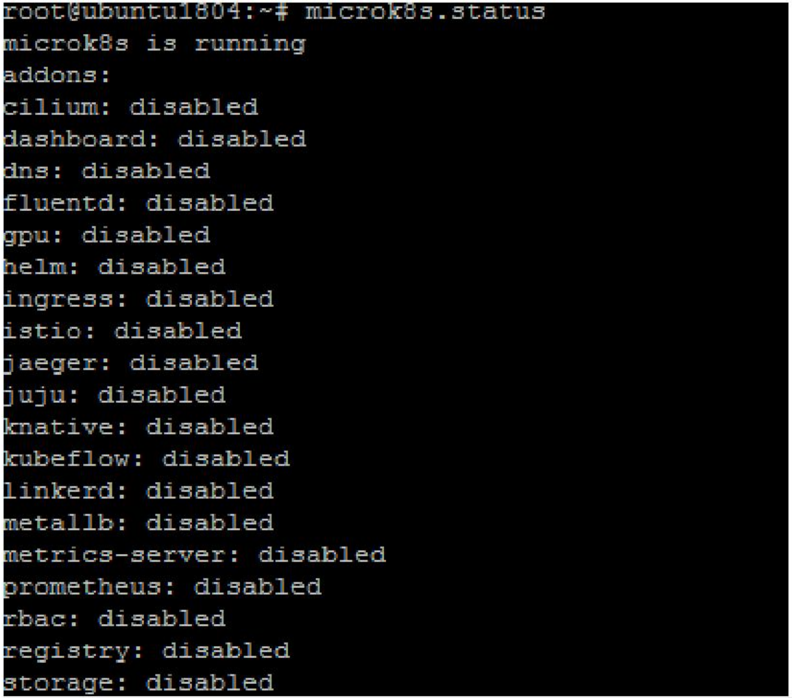
https://ubuntu.com/kubernetes/install ; https://microk8s.io/docs/

1. Установить microk8s (screenshot)

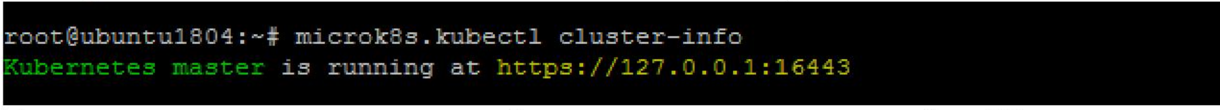


```
root@ubuntu1804:~# sudo snap install microk8s --classic --channel=1.17/stable
microk8s (1.17/stable) v1.17.4 from Canonical✓ installed
root@ubuntu1804:~# snap info microk8s
name:      microk8s
summary:   Kubernetes for workstations and appliances
publisher: Canonical✓
store-url: https://snapcraft.io/microk8s
contact:   https://github.com/ubuntu/microk8s
license:   unset
description: |
  MicroK8s is a small, fast, secure, single node Kubernetes that installs on just about any Linux
  box. Use it for offline development, prototyping, testing, or use it on a VM as a small, cheap,
  reliable k8s for CI/CD. It's also a great k8s for appliances - develop your IoT apps for k8s and
  deploy them to MicroK8s on your boxes.
commands:
  - microk8s.add-node
  - microk8s.cilium
  - microk8s.config
  - microk8s.ctr
```

2. Проверьте статус (screenshot) и команды менеджера кластера (screenshot).

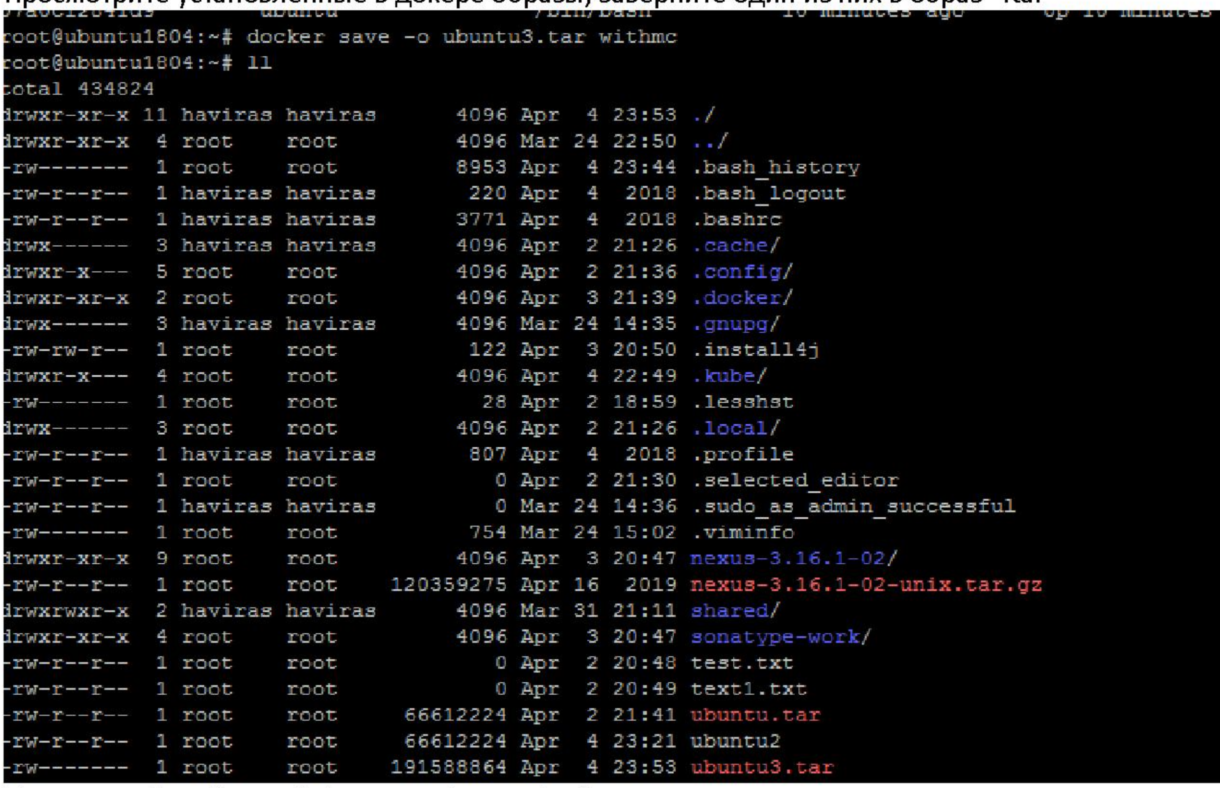


```
root@ubuntu1804:~# microk8s.status
microk8s is running
addons:
cilium: disabled
dashboard: disabled
dns: disabled
fluentd: disabled
gpu: disabled
helm: disabled
ingress: disabled
istio: disabled
jaeger: disabled
juju: disabled
knative: disabled
kubeflow: disabled
linkerd: disabled
metallb: disabled
metrics-server: disabled
prometheus: disabled
rbac: disabled
registry: disabled
storage: disabled
```



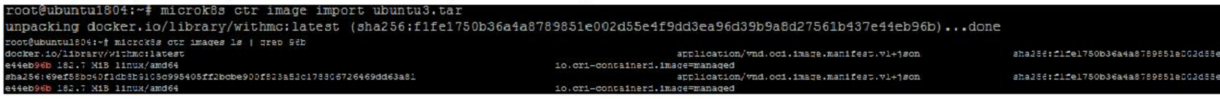
```
root@ubuntu1804:~# microk8s.kubectl cluster-info
Kubernetes master is running at https://127.0.0.1:16443
```

3. Просмотрите установленные в докере образы; заверните один из них в образ *.tar



```
root@ubuntu1804:~# docker save -o ubuntu3.tar withmc
root@ubuntu1804:~# ll
total 434824
drwxr-xr-x 11 haviras haviras      4096 Apr  4 23:53 ./
drwxr-xr-x  4 root    root         4096 Mar 24 22:50 ../
-rw-------  1 root    root         8953 Apr  4 23:44 .bash_history
-rw-r--r--  1 haviras haviras       220 Apr  4  2018 .bash_logout
-rw-r--r--  1 haviras haviras      3771 Apr  4  2018 .bashrc
drwx------  3 haviras haviras      4096 Apr  2 21:26 .cache/
drwxr-x---  5 root    root         4096 Apr  2 21:36 .config/
drwxr-xr-x  2 root    root         4096 Apr  3 21:39 .docker/
drwx------  3 haviras haviras      4096 Mar 24 14:35 .gnupg/
-rw-rw-r--  1 root    root          122 Apr  3 20:50 .install4j
drwxr-x---  4 root    root         4096 Apr  4 22:49 .kube/
-rw-------  1 root    root           28 Apr  2 18:59 .lesshst
drwx------  3 root    root         4096 Apr  2 21:26 .local/
-rw-r--r--  1 haviras haviras       807 Apr  4  2018 .profile
-rw-r--r--  1 root    root            0 Apr  2 21:30 .selected_editor
-rw-r--r--  1 haviras haviras         0 Mar 24 14:36 .sudo_as_admin_successful
-rw-------  1 root    root          754 Mar 24 15:02 .viminfo
drwxr-xr-x  9 root    root         4096 Apr  3 20:47 nexus-3.16.1-02/
-rw-r--r--  1 root    root    120359275 Apr 16  2019 nexus-3.16.1-02-unix.tar.gz
drwxrwxr-x  2 haviras haviras      4096 Mar 31 21:11 shared/
drwxr-xr-x  4 root    root         4096 Apr  3 20:47 sonatype-work/
-rw-r--r--  1 root    root            0 Apr  2 20:48 test.txt
-rw-r--r--  1 root    root            0 Apr  2 20:49 text1.txt
-rw-r--r--  1 root    root     66612224 Apr  2 21:41 ubuntu.tar
-rw-r--r--  1 root    root     66612224 Apr  4 23:21 ubuntu2
-rw-------  1 root    root    191588864 Apr  4 23:53 ubuntu3.tar
```

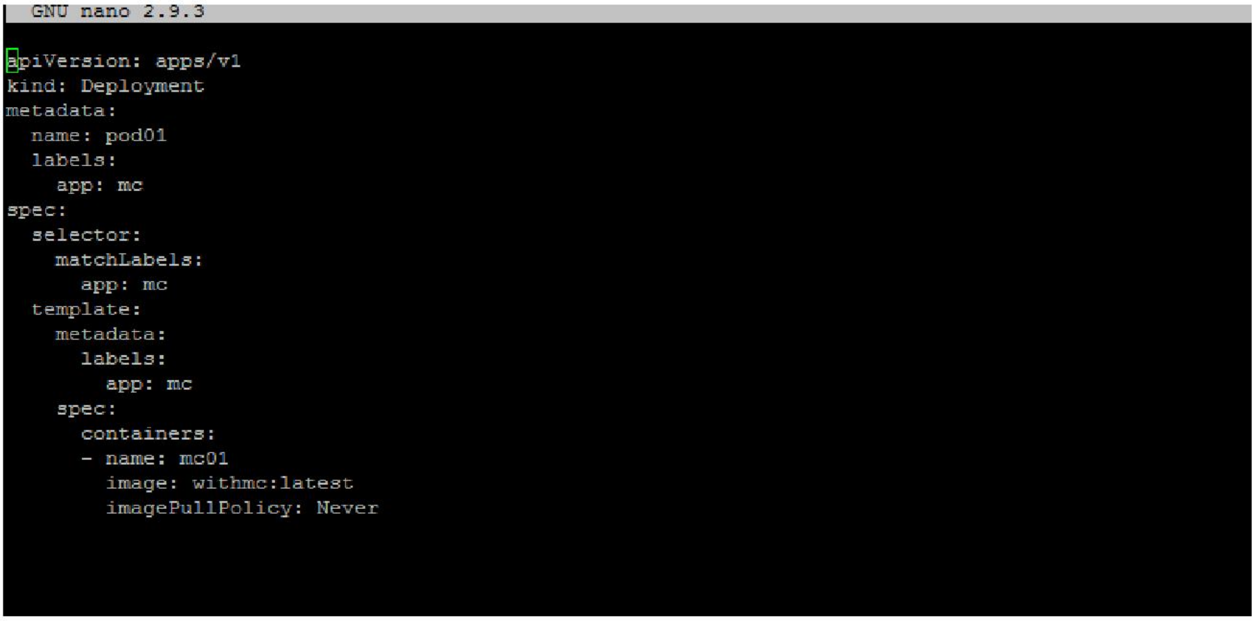
4. Импортируйте образ в Kubernetes (screenshot)



```
root@ubuntu1804:~# microk8s ctr image import ubuntu3.tar
unpacking docker.io/library/withmc:latest (sha256:f1fe1750b36a4a8789851e002d55e4f9dd3ea96d39b9a8d27561b437e44eb96b)...done
root@ubuntu1804:~# microk8s ctr images ls | grep 96b
docker.io/library/withmc:latest                                                    application/vnd.oci.image.manifest.v1+json            sha256:f1fe1750b36a4a8789851e002d55e
e44eb96b 182.7 MiB linux/amd64                                       io.cri-containerd.image=managed
sha256:69ef58bc40f1db8b9105c995405ff2bcbe900f823a52c17830 6726469dd63a81          application/vnd.oci.image.manifest.v1+json            sha256:f1fe1750b36a4a8789851e002d55e
e44eb96b 182.7 MiB linux/amd64                                       io.cri-containerd.image=managed
```

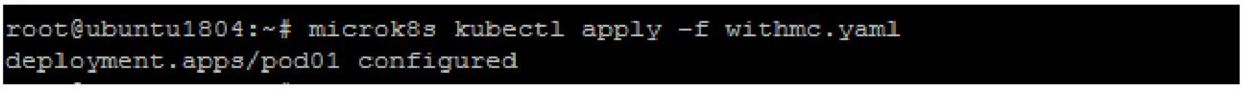
5. Запустите образ и убедитесь, что он работает. (screenshot)

Объявил yaml



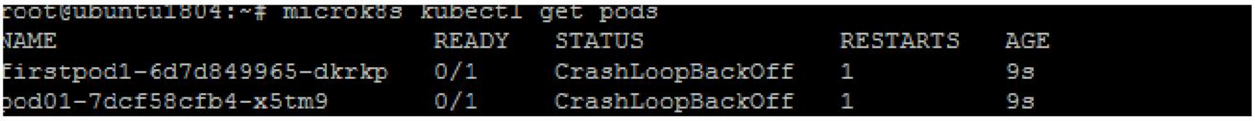
```
  GNU nano 2.9.3

apiVersion: apps/v1
kind: Deployment
metadata:
  name: pod01
  labels:
    app: mc
spec:
  selector:
    matchLabels:
      app: mc
  template:
    metadata:
      labels:
        app: mc
    spec:
      containers:
      - name: mc01
        image: withmc:latest
        imagePullPolicy: Never
```

применил его



```
root@ubuntu1804:~# microk8s kubectl apply -f withmc.yaml
deployment.apps/pod01 configured
```

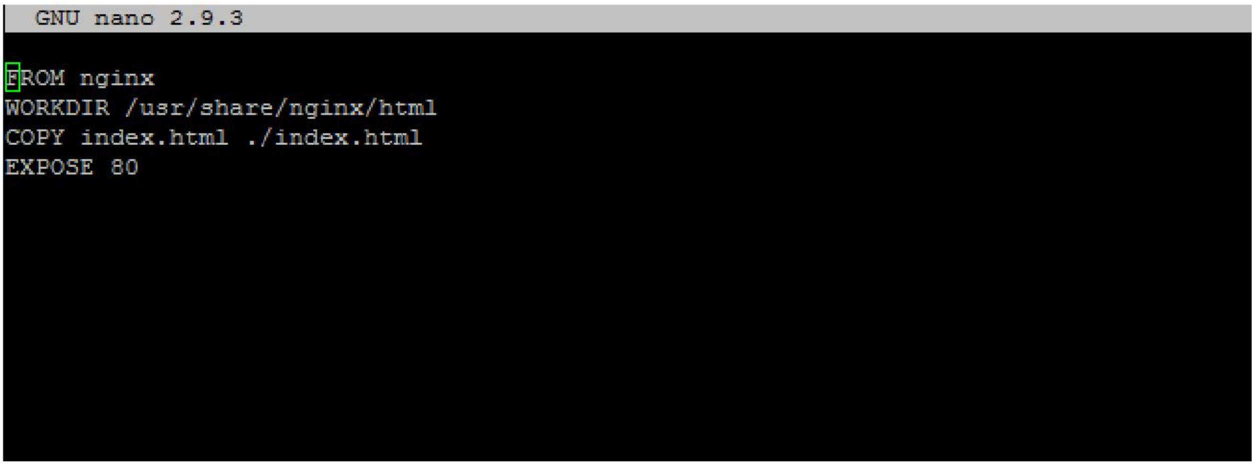
kubectl выдал статус контейнера CrashLoopBackOff потому что mc сразу завершает себя



```
root@ubuntu1804:~# microk8s kubectl get pods
NAME                          READY   STATUS             RESTARTS   AGE
firstpod1-6d7d849965-dkrkp    0/1     CrashLoopBackOff   1          9s
pod01-7dcf58cfb4-x5tm9        0/1     CrashLoopBackOff   1          9s
```

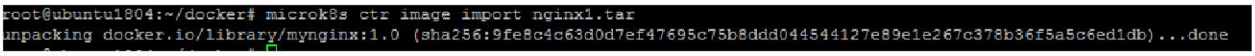
Поэтому лучше использовать постоянно работающий сервис, например nginx

Создаем докер файл



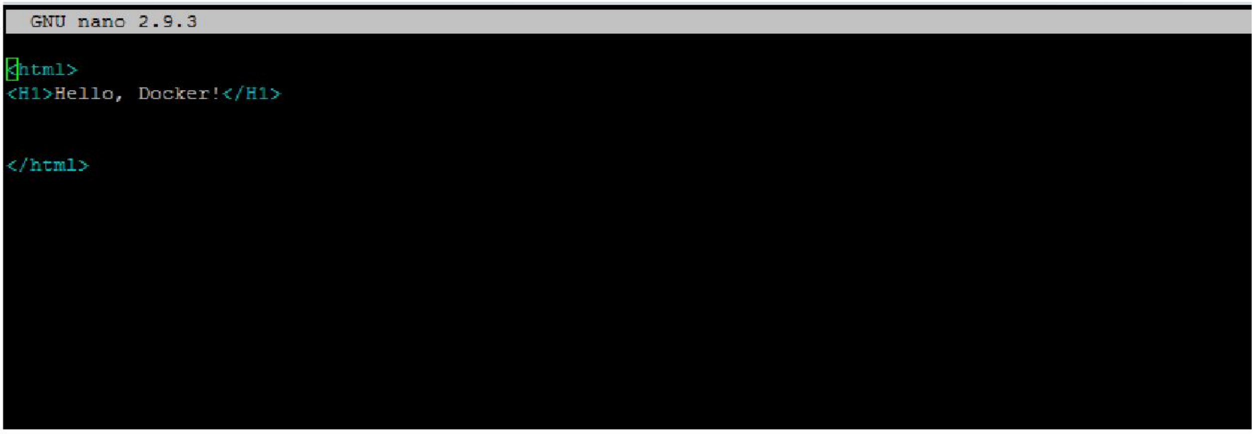
```
  GNU nano 2.9.3

FROM nginx
WORKDIR /usr/share/nginx/html
COPY index.html ./index.html
EXPOSE 80
```

Импортируем в кубернетс



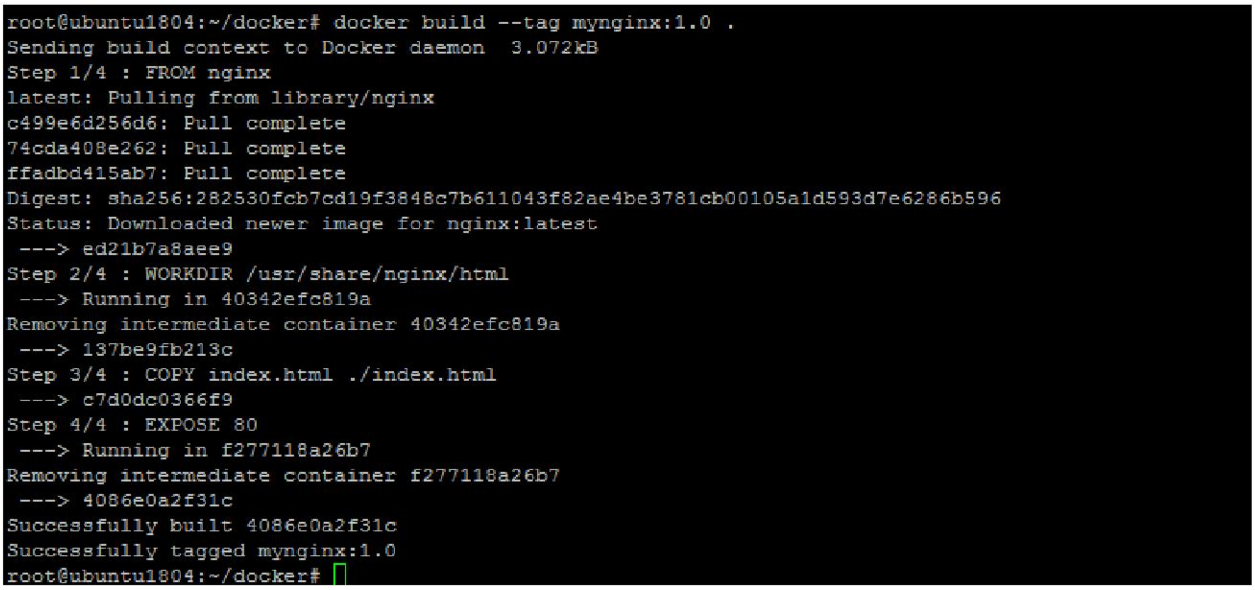
```
root@ubuntu1804:~/docker# microk8s ctr image import nginx1.tar
unpacking docker.io/library/mynginx:1.0 (sha256:9fe8c4c63d0d7ef47695c75b8ddd044544127e89e1e267c378b36f5a5c6ed1db)...done
```

Создаем index.html с произвольным содержанием



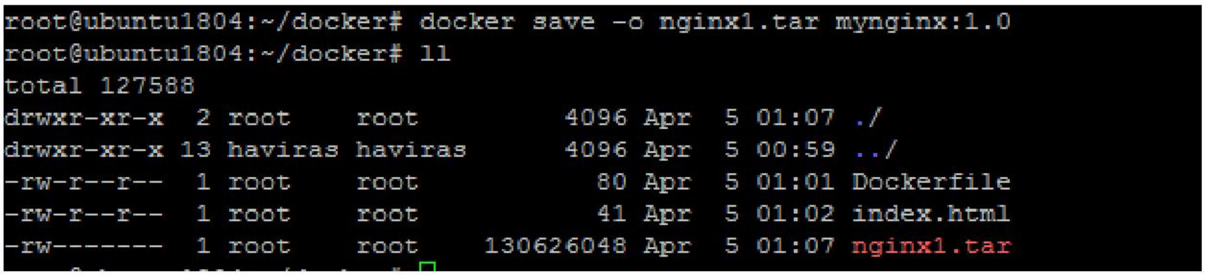
```
  GNU nano 2.9.3

<html>
<H1>Hello, Docker!</H1>


</html>
```

Создаем образ



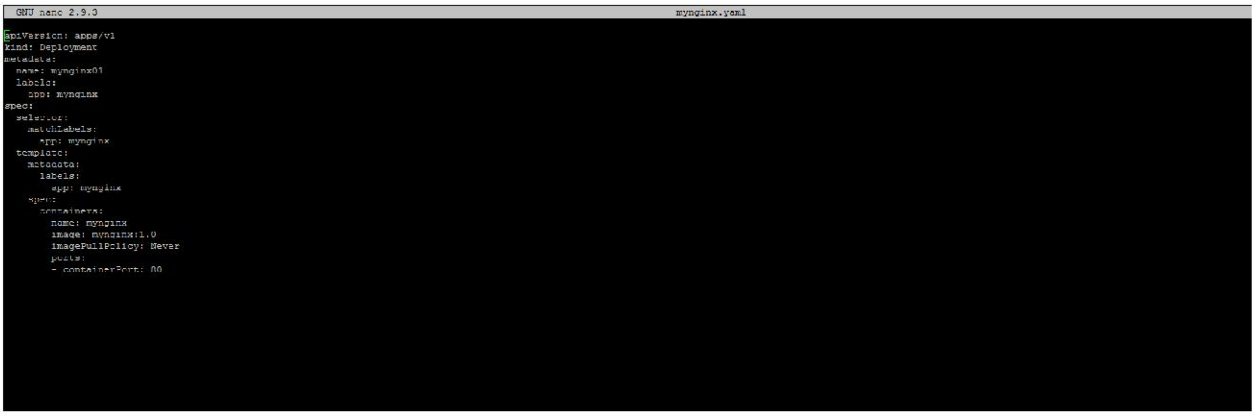
```
root@ubuntu1804:~/docker# docker build --tag mynginx:1.0 .
Sending build context to Docker daemon  3.072kB
Step 1/4 : FROM nginx
latest: Pulling from library/nginx
c499e6d256d6: Pull complete
74cda408e262: Pull complete
ffadbd415ab7: Pull complete
Digest: sha256:282530fcb7cd19f3848c7b611043f82ae4be3781cb00105a1d593d7e6286b596
Status: Downloaded newer image for nginx:latest
 ---> ed21b7a8aee9
Step 2/4 : WORKDIR /usr/share/nginx/html
 ---> Running in 40342efc819a
Removing intermediate container 40342efc819a
 ---> 137be9fb213c
Step 3/4 : COPY index.html ./index.html
 ---> c7d0dc0366f9
Step 4/4 : EXPOSE 80
 ---> Running in f277118a26b7
Removing intermediate container f277118a26b7
 ---> 4086e0a2f31c
Successfully built 4086e0a2f31c
Successfully tagged mynginx:1.0
root@ubuntu1804:~/docker#
```

Экспортируем в tar



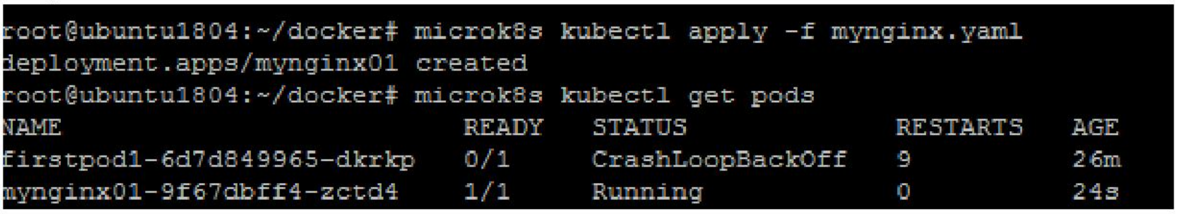
```
root@ubuntu1804:~/docker# docker save -o nginx1.tar mynginx:1.0
root@ubuntu1804:~/docker# ll
total 127588
drwxr-xr-x  2 root    root          4096 Apr  5 01:07 ./
drwxr-xr-x 13 haviras haviras       4096 Apr  5 00:59 ../
-rw-r--r--  1 root    root            80 Apr  5 01:01 Dockerfile
-rw-r--r--  1 root    root            41 Apr  5 01:02 index.html
-rw-------  1 root    root     130626048 Apr  5 01:07 nginx1.tar
```

Описали pod в yaml файле



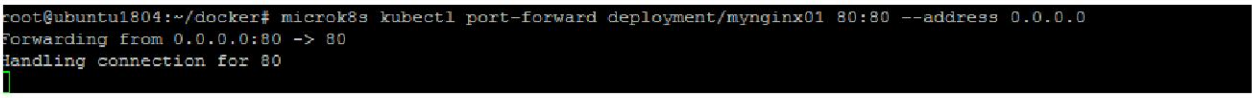
```
  GNU nano 2.9.3                                                          mynginx.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  name: mynginx01
  labels:
    app: mynginx
spec:
  selector:
    matchLabels:
      app: mynginx
  template:
    metadata:
      labels:
        app: mynginx
    spec:
      containers:
        name: mynginx
        image: mynginx:1.0
        imagePullPolicy: Never
        ports:
        - containerPort: 80
```

Запускаем под



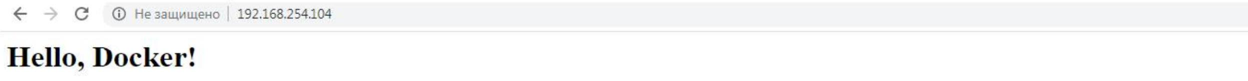
```
root@ubuntu1804:~/docker# microk8s kubectl apply -f mynginx.yaml
deployment.apps/mynginx01 created
root@ubuntu1804:~/docker# microk8s kubectl get pods
NAME                            READY   STATUS             RESTARTS   AGE
firstpod1-6d7d849965-dkrkp      0/1     CrashLoopBackOff   9          26m
mynginx01-9f67dbff4-zctd4       1/1     Running            0          24s
```

Пробрасываем порт (это неправильно! Надо использовать cubefwd?)



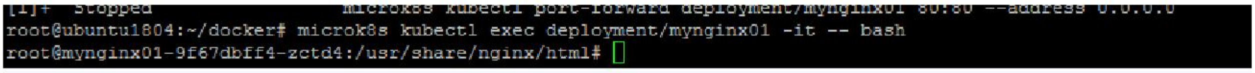
```
root@ubuntu1804:~/docker# microk8s kubectl port-forward deployment/mynginx01 80:80 --address 0.0.0.0
Forwarding from 0.0.0.0:80 -> 80
Handling connection for 80
```

Вывод в браузер



Hello, Docker!

А вообще в контейнер можно зайти вот так



```
[1]+  Stopped                 microk8s kubectl port-forward deployment/mynginx01 80:80 --address 0.0.0.0
root@ubuntu1804:~/docker# microk8s kubectl exec deployment/mynginx01 -it -- bash
root@mynginx01-9f67dbff4-zctd4:/usr/share/nginx/html#
```