EPAM University Programs

DevOps external course

Module 2 Virtualization and Cloud Basic

TASK 2.4

Danylenko - Homework

Работа с lxc в Ubuntu

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

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

1. Установить lxc

```
danylenko@VM2:~$ sudo apt install lxd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer requ
  efibootmgr libfwup1 libwayland-egl1-mesa
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ebtables libuv1 lxd-client xdelta3
Suggested packages:
  criu lxd-tools
The following NEW packages will be installed:
  ebtables libuv1 lxd lxd-client xdelta3
0 upgraded, 5 newly installed, 0 to remove and 3 not upgraded.
Need to get 8 440 kB of archives.
After this operation, 32,4 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ua.archive.ubuntu.com/ubuntu bionic-updates/main amd64 ebtabl
```

2. Запустить lxc launch для любой из версий Убунту

```
danylenko@VM2:~$ lxc launch ubuntu:18.04 x1
Creating x1
Starting x1
```

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

```
danylenko@VM2:~$ lxc list
+------+---------+----------------------+-----------------------------------
---------+------------+-----------+
| NAME |  STATE  |         IPV4         |                    IPV6
        |    TYPE    | SNAPSHOTS |
+------+---------+----------------------+-----------------------------------
---------+------------+-----------+
| x1   | RUNNING | 10.170.209.251 (eth0) | fd42:c3d0:19a1:a826:216:3eff:fe1a:699
f (eth0) | PERSISTENT | 0         |
+------+---------+----------------------+-----------------------------------
---------+------------+-----------+
```

4. Зайдите в контейнер с командной строкой bash /bin/bash

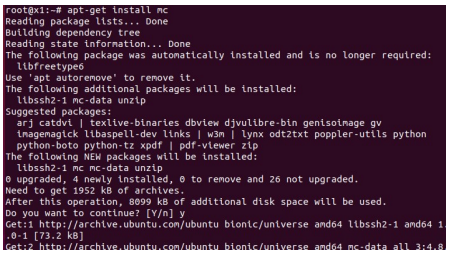
```
danylenko@VM2:~$ lxc exec x1 bash
root@x1:~#
```

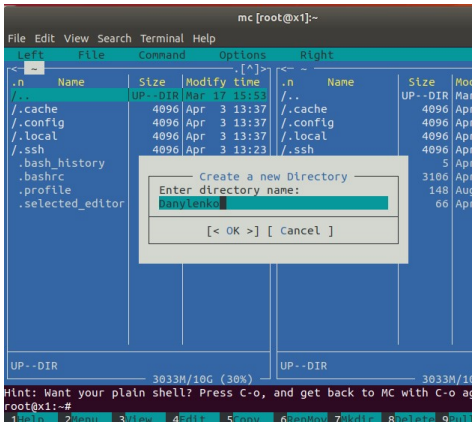
5. Запустите обновление apt-get update

```
danylenko@VM2:~$ lxc exec x1 -- /bin/bash
root@x1:~# apt-get update
Hit:1 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [677
```

```
Fetched 18.5 MB in 17s (1078 kB/s)
Reading package lists... Done
```

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





7. Загрузите в контейнер файл

```
danylenko@VM2:~$ lxc file push /home/danylenko/Desktop/folder/test_file x1/home/
test_file

danylenko@VM2:~$ lxc exec -- cat /home/test_file
Error: not found
danylenko@VM2:~$ lxc exec x1 -- cat /home/test_file
danylenko
```

и скачайте с контейнера другой файл

```
danylenko@VM2:~$ lxc exec x1 -- nano /home/remote_file
danylenko@VM2:~$ lxc file pull x1/home/remote_file /home/danylenko/Desktop/folde
r/
danylenko@VM2:~$ cat /home/danylenko/Desktop/folder/remote_file
danylenko-remote-file
danylenko@VM2:~$
```

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

```
danylenko@VM2:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
   Active: active (running) since Sun 2020-04-05 11:06:34 EEST; 1min 39s ago
     Docs: https://docs.docker.com
 Main PID: 11167 (dockerd)
    Tasks: 8
   CGroup: /system.slice/docker.service
           └─11167 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai
```

2. Запустить поиск сконфигурированных решений для “ubuntu”

```
danylenko@VM2:~$ docker search ubuntu
NAME                                                    DESCRIPTION                                     STARS
               OFFICIAL            AUTOMATED
ubuntu                                                  Ubuntu is a Debian-based Linux operating sys…   10715
               [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]
neurodebian                                             NeuroDebian provides neuroscience research s…   68
```

3. Скачать любой из образов на локальную машину.

```
danylenko@VM2:~$ 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
```

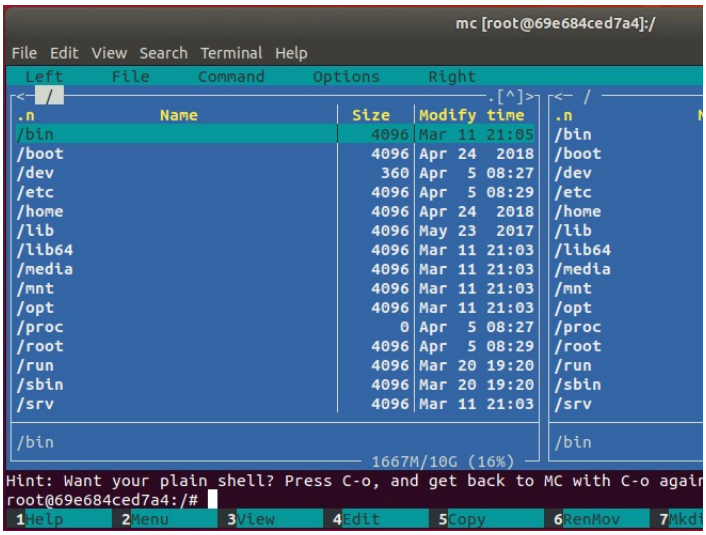
4. Запустить команду просмотра загруженных на компьютер образов.

```
danylenko@VM2:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              4e5021d210f6        2 weeks ago         64.2MB
hello-world         latest              fce289e99eb9        15 months ago       1.84kB
```

5. Запустите обновление apt-get update

```
danylenko@VM2:~$ docker run -it ubuntu
root@69e684ced7a4:/# apt-get update
Get:1 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [835 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [37.0 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [870 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [7904 B]
Get:10 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [186 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [1161 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [50.4 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1367 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [12.2 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [2496 B]
Get:18 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [4247 B]
Fetched 17.7 MB in 16s (1108 kB/s)
Reading package lists... Done
```

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



7. Загрузите в контейнер файл и скачайте с контейнера другой файл.

```
danylenko@VM2:~$ docker exec -it busy_bhaskara ls /home/
remote_file
failed to resize tty, using default size
                                        danylenko@VM2:~$
danylenko@VM2:~$ ls /home/danylenko
Desktop  Documents  Downloads  examples.desktop  local_file  Music  Pictures  Public  Templates  Videos
danylenko@VM2:~$ docker cp /home/danylenko/local_file busy_bhaskara:/home/local_file
danylenko@VM2:~$ docker cp busy_bhaskara:/home/remote_file /home/danylenko/
danylenko@VM2:~$ cat /home/danylenko/remote_file
danylenko_remote_file
danylenko@VM2:~$ docker exec -it busy_bhaskara cat /home/local_file
danylenko local file
```

8. Прочитать документацию и кратко описать основные 7 команд Dockerfile

FROM imageName AS name - sets starting image from which build process will start, also additional images, can divide build process on stages and use resulting build of some stage as a parent of other stages.

ENV name=value – sets global variables for whole build process and all stages.

ARG name=default value – sets allowed arguments list that can be used with docker build command, after it. Scope is until after next FROM command – next stage start. ARG need to be used again, inside of each stage for variable value to be available.

RUN - Executes commands in new layer on top of current image, and commits result to the image. Have Shell and Exec calls. Exec form can execute commands directly without shell, but also without Shell context and Shell variables.

ENTRYPOINT и CMD – Do not participate in build process, but are used to set command execution that will be called when built image will run. Have shell and Exec forms of calls. Exec form runs command without shell with PID 1 and allows this process to get commands sent to it though Docker from Host. CMD is replaced with commands\args from run comand line after image name. ENTRYPOINT sets not replaceable by default part of command and, with additional replaceable part from CMD, allows presenting the image as executable application. (Still can be replaced with --entrypoint)

ONBUILD sets instruction that will run upon use of this image as a base of next build in FROM command. Instructions will be triggered right after FROM command, before all following commands.

HEALTHCHECK used to set specific instructions that to make checks of working state of processes in running image that can't be connected directly to image state. Checks can be set to be done in intervals, with timeouts, and number of retries, after which image state in Docker will change to Unhealthy.

COPY, ADD – commands that are used to copy files from context sent by user with build command, to container. From best practices mostly COPY should be used. ADD should be only used for local .tar extraction. Also COPY can use files from previous stages of build.

Работа с Kubernetes в Ubuntu

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

1. Установить microk8s

```
danylenko@VM2:~$ sudo snap install microk8s --classic --channel=latest/stable
microk8s v1.18.0 from Canonical✓ installed
```

2. Проверьте статус

```
danylenko@VM2:~$ microk8s status
microk8s is running
addons:
cilium: disabled
dashboard: disabled
dns: disabled
fluentd: disabled
gpu: disabled
helm: disabled
helm3: disabled
ingress: disabled
istio: disabled
jaeger: disabled
knative: disabled
kubeflow: disabled
linkerd: disabled
metallb: disabled
metrics-server: disabled
prometheus: disabled
rbac: disabled
registry: disabled
storage: disabled
```

и команды менеджера кластера

```
danylenko@VM2:~$ microk8s kubectl get no
NAME   STATUS   ROLES    AGE   VERSION
vm2    Ready    <none>   17m   v1.18.0
danylenko@VM2:~$ microk8s add-node
Join node with: microk8s join 10.0.2.15:25000/YjDFHqRBsYGXJYeYrheChieFBorexppg

If the node you are adding is not reachable through the default interface you ca
 microk8s join 10.0.2.15:25000/YjDFHqRBsYGXJYeYrheChieFBorexppg
 microk8s join 10.0.3.1:25000/YjDFHqRBsYGXJYeYrheChieFBorexppg
 microk8s join 172.18.0.1:25000/YjDFHqRBsYGXJYeYrheChieFBorexppg
 microk8s join 10.1.95.0:25000/YjDFHqRBsYGXJYeYrheChieFBorexppg
danylenko@VM2:~$ microk8s join 10.0.2.15:25000/YjDFHqRBsYGXJYeYrheChieFBorexppg
[sudo] password for danylenko:
danylenko@VM2:~$ microk8s kubectl get no
This MicroK8s deployment is acting as a node in a cluster. Please use the microk
danylenko@VM2:~$ microk8s leave
Stopped.
Started.
Enabling pod scheduling
node/vm2 already uncordoned
danylenko@VM2:~$ microk8s kubectl get no
NAME   STATUS   ROLES    AGE   VERSION
vm2    Ready    <none>   23m   v1.18.0
```

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

```
danylenko@VM2:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu-danylenko    local               33d234e084d9        5 minutes ago       187MB
ubuntu              latest              4e5021d210f6        2 weeks ago         64.2MB
ubuntu              local               4e5021d210f6        2 weeks ago         64.2MB
hello-world         latest              fce289e99eb9        15 months ago       1.84kB
danylenko@VM2:~$ docker save ubuntu-danylenko:local > lastimage.tar
danylenko@VM2:~$ microk8s ctr image import lastimage.tar
unpacking docker.io/library/ubuntu-danylenko:local (sha256:369c9601801da08e111e32ef771d8546d1bf723ca4f9fbc565636c8d8266c525)...done
danylenko@VM2:~$
```

5. Запустите образ и убедитесь, что он работает.

```
danylenko@VM2:~$ microk8s kubectl run -it danylenko-ubuntu-mc --image=ubuntu-danylenko:local
If you don't see a command prompt, try pressing enter.
root@danylenko-ubuntu-mc:/# mc

root@danylenko-ubuntu-mc:/# date
Sun Apr  5 16:14:44 UTC 2020
root@danylenko-ubuntu-mc:/# exit
exit

root@danylenko-ubuntu-mc:/# exit
exit
Session ended, resume using 'kubectl attach danylenko-ubuntu-mc -c danylenko-ubuntu-mc -i -t' command when the pod is running
danylenko@VM2:~$ microk8s kubectl get pod danylenko-ubuntu-mc
NAME                 READY   STATUS    RESTARTS   AGE
danylenko-ubuntu-mc  1/1     Running   1          2m52s
danylenko@VM2:~$
```