

Lab1: Docker basics

Exercise 1: Install docker:

1. Log in to your VM.
2. Start terminal and elevate your privileges to root.
3. Run yum install docker.
4. After installation is finished, start docker by running this command `systemctl start docker`.
5. Also enable docker service automatic start with command `systemctl enable docker`.
6. Run `docker version` to see installed version.

```
stan@stan-VirtualBox:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap
  docker-doc rinse zfs-fuse | zfsutils git-daemon-run | git-daemon-sysvinit
  git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  bridge-utils containerd docker.io git git-man liberror-perl pigz runc
  ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 121 not upgraded.
Need to get 76,1 MB of archives.
After this operation, 307 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://bg.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63,6 kB]
Get:2 http://bg.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34,4 kB]
Get:3 http://bg.archive.ubuntu.com/ubuntu jammy-updates/main amd64 runc amd64 1.1.4-0ubuntu1~22.04.1 [4241 kB]
Get:4 http://bg.archive.ubuntu.com/ubuntu jammy-updates/main amd64 containerd amd64 1.6.12-0ubuntu1~22.04.1 [34,4 MB]
Get:5 http://bg.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 docker.io amd64 20.10.21-0ubuntu1~22.04.2 [33,2 M]
Get:6 http://bg.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26,5 kB]
Get:7 http://bg.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.8 [953 kB]
Get:8 http://bg.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.8 [3141 kB]
Get:9 http://bg.archive.ubuntu.com/ubuntu jammy/universe amd64 ubuntu-fan all 0.12.16 [35,2 kB]
(Trash) 76,1 MB in 12s (6165 kB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 199283 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7-1ubuntu3_amd64.deb ...
Unpacking bridge-utils (1.7-1ubuntu3) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.1.4-0ubuntu1~22.04.1_amd64.deb ...
Unpacking runc (1.1.4-0ubuntu1~22.04.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.6.12-0ubuntu1~22.04.1_amd64.deb ...
Unpacking containerd (1.6.12-0ubuntu1~22.04.1) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../4-docker.io_20.10.21-0ubuntu1~22.04.2_amd64.deb ...
Unpacking docker.io (20.10.21-0ubuntu1~22.04.2) ...
Selecting previously unselected package liberror-perl.
Preparing to unpack .../5-liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
```

- downloading docker IO.

```

Fetched 76,1 MB in 12s (6165 kB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 199283 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7-1ubuntu3_amd64.deb ...
Unpacking bridge-utils (1.7-1ubuntu3) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.1.4-0ubuntu1~22.04.1_amd64.deb ...
Unpacking runc (1.1.4-0ubuntu1~22.04.1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.6.12-0ubuntu1~22.04.1_amd64.deb ...
Unpacking containerd (1.6.12-0ubuntu1~22.04.1) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../4-docker.io_20.10.21-0ubuntu1~22.04.2_amd64.deb ...
Unpacking docker.io (20.10.21-0ubuntu1~22.04.2) ...
Selecting previously unselected package liberror-perl.
Preparing to unpack .../5-liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../6-git-man_1%3a2.34.1-1ubuntu1.8_all.deb ...
Unpacking git-man (1:2.34.1-1ubuntu1.8) ...
Selecting previously unselected package git.
Preparing to unpack .../7-git_1%3a2.34.1-1ubuntu1.8_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.8) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../8-ubuntu-fan_0.12.16_all.deb ...
Unpacking ubuntu-fan (0.12.16) ...
Setting up runc (1.1.4-0ubuntu1~22.04.1) ...
Setting up liberror-perl (0.17029-1) ...
Setting up bridge-utils (1.7-1ubuntu3) ...
Setting up pigz (2.6-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.8) ...
Setting up containerd (1.6.12-0ubuntu1~22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service →
/lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service →
/lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (20.10.21-0ubuntu1~22.04.2) ...
Adding group 'docker' (GID 137) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /li
b/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/sy
stemd/system/docker.socket.
Setting up git (1:2.34.1-1ubuntu1.8) ...
Processing triggers for man-db (2.10.2-1) ...

```

- downloading docker IO part2.

```

stan@stan-VirtualBox:~$ sudo snap install docker
docker 20.10.17 from Canonical✓ installed
stan@stan-VirtualBox:~$ docker --version
Docker version 20.10.21, build 20.10.21-0ubuntu1~22.04.2
stan@stan-VirtualBox:~$

```

- docker version and installing docker.

7. Run docker help to see list of available commands.

```
stan@stan-VirtualBox:~$ docker help

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default "/home/stan/.docker")
  -c, --context string  Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var and default context set with "docker context use")
  -D, --debug           Enable debug mode
  -H, --host list       Daemon socket(s) to connect to
  -l, --log-level string Set the logging level ("debug"|"info"|"warn"|"error"|"fatal") (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string    Trust certs signed only by this CA (default "/home/stan/.docker/ca.pem")
  --tlscert string       Path to TLS certificate file (default "/home/stan/.docker/cert.pem")
  --tlskey string        Path to TLS key file (default "/home/stan/.docker/key.pem")
  --tlsverify           Use TLS and verify the remote
  -v, --version         Print version information and quit

Management Commands:
  builder      Manage builds
  config        Manage Docker configs
  container     Manage containers
  context       Manage contexts
  image        Manage images
  manifest      Manage Docker image manifests and manifest lists
  network       Manage networks
  node          Manage Swarm nodes
  plugin        Manage plugins
  secret        Manage Docker secrets
  service       Manage services
  stack         Manage Docker stacks
  swarm         Manage Swarm
  system        Manage Docker
  trust         Manage trust on Docker images
  volume        Manage volumes

Commands:
  attach        Attach local standard input, output, and error streams to a running container
  build          Build an image from a Dockerfile
  commit        Create a new image from a container's changes
  cp            Copy files/folders between a container and the local filesystem
  create        Create a new container
  diff          Inspect changes to files or directories on a container's filesystem
  events        Get real time events from the server
  exec          Run a command in a running container
  export        Export a container's filesystem as a tar archive
  history        Show the history of an image
  images        List images
  import        Import the contents from a tarball to create a filesystem image
  info          Display system-wide information
  inspect       Return low-level information on Docker objects
  kill          Kill one or more running containers
  load          Load an image from a tar archive or STDIN
```

- Docker help command

```
images      List images
import      Import the contents from a tarball to create a filesystem image
info        Display system-wide information
inspect     Return low-level information on Docker objects
kill        Kill one or more running containers
load        Load an image from a tar archive or STDIN
login        Log in to a Docker registry
logout      Log out from a Docker registry
logs        Fetch the logs of a container
pause       Pause all processes within one or more containers
port        List port mappings or a specific mapping for the container
ps          List containers
pull        Pull an image or a repository from a registry
push        Push an image or a repository to a registry
rename      Rename a container
restart     Restart one or more containers
rm          Remove one or more containers
rmi         Remove one or more images
run         Run a command in a new container
save        Save one or more images to a tar archive (streamed to STDOUT by default)
search      Search the Docker Hub for images
start       Start one or more stopped containers
stats       Display a live stream of container(s) resource usage statistics
stop        Stop one or more running containers
tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top         Display the running processes of a container
unpause     Unpause all processes within one or more containers
update      Update configuration of one or more containers
version     Show the Docker version information
wait        Block until one or more containers stop, then print their exit codes

Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at https://docs.docker.com/go/guides/
```

- Docker help command part 2.

8. Search for a command (switch) that will show system-wide information for your instance of docker.
9. Test it by running docker <command you have discovered>.
10. From the output try to find where the information of number of containers and images is.
11. Also try to find whether this docker is part of a swarm

```

stan@stan-VirtualBox: ~$ sudo docker info
[sudo] password for stan:
Client:
 Context:          default
 Debug Mode: false

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 20.10.17
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
 Userxattr: false
 Logging Driver: json-file
 Cgroup Driver: systemd
 Cgroup Version: 2
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 io.containerd.runtime.v1.linux runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 10c12954828e7c7c9b6e0ea9b0c02b01407d3ae1
 runc version: de40ad0
 Security Options:
  apparmor
  seccomp
   Profile: default
 Cgroupns:
 Kernel Version: 5.19.0-35-generic
 Operating System: Ubuntu Core 18
 OSType: linux
 Architecture: x86_64
 CPUs: 4
 Total Memory: 4.982GiB
 Name: stan-VirtualBox
 ID: 602G:LOBJ:FF7D:VKGB:3KFB:3N62:L7HH:KZUC:3SUR:NG4B:BNAI:6RNK
 Docker Root Dir: /var/snap/docker/common/var-lib-docker
 Debug Mode: false
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  127.0.0.0/8
 Live Restore Enabled: false

stan@stan-VirtualBox: ~$

```

- Swarm:inactive.
- docker info command

Lab2: Creating images

Excercise1: Build a simple image:

1. Create a Docker container that executes a simple bash script. Go to your home directory and run `mkdir test`.

Run `cd test`.

2. Create a simple script. Run `vi test.sh`.

3. Write the following in your script file:

```
#!/bin/bash
```

```
sleep 30
```

```
exit 1
```

4. Save the file. In vi editor press `:wq`.

5. Create a docker file. Run `vi Dockerfile`.

6. Write the following in our Dockerfile:

```

stan@stan-VirtualBox:~$ mkdir test
stan@stan-VirtualBox:~$ cd test/
stan@stan-VirtualBox:~/test$ nano test.sh
stan@stan-VirtualBox:~/test$ nano Dockerfile
stan@stan-VirtualBox:~/test$ docker build -t my-image ./
invalid argument "my-image./" for "-t, --tag" flag: invalid reference format
See 'docker build --help'.
stan@stan-VirtualBox:~/test$ docker build -t my-image1 ./
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v:
t=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=Dockerfile&labels=%7B%7D&memory=0&memswap=0&networkmode=default&rm=1&shmsi:
nix /var/run/docker.sock: connect: permission denied
stan@stan-VirtualBox:~/test$ sudo docker build -t my-image1 ./
Sending build context to Docker daemon 3.072kB
Error response from daemon: dockerfile parse error line 3: ADD requires at least two arguments, but only one was provided. Destination could not be
stan@stan-VirtualBox:~/test$ nano Dockerfile
stan@stan-VirtualBox:~/test$ sudo docker build -t my-image1 ./
Sending build context to Docker daemon 3.072kB
Step 1/3 : FROM alpine
latest: Pulling from library/alpine
f56be85fc22e: Pull complete
Digest: sha256:124c7d2707904eea7431fffe91522a01e5a861a624ee31d03372cc1d138a3126
Status: Downloaded newer image for alpine:latest
--> 9ed4aefc74f6
Step 2/3 : ADD test.sh /
--> 72654e2ef2de
Step 3/3 : CMD /bin/bash/test.sh
--> Running in 03c379b7c431
Removing intermediate container 03c379b7c431
--> 6f60c72e00ef
Successfully built 6f60c72e00ef
Successfully tagged my-image1:latest
stan@stan-VirtualBox:~/test$

```

- Creating directory and the 2 files and editing them, running the image

FROM alpine

ADD test.sh /

CMD /bin/bash /test.sh

7. Save your Dockerfile.

8. Build your image. Run `docker build -t my-image1 ./`

9. Now spawn a container. Run `docker run -name my-test1 my-image1`.

10. Do a `docker ps -a`. Do you see your container running?

11. Do a `docker logs my-test1`. What is the output of the log?

```

stan@stan-VirtualBox:~/test$ sudo docker run --name my-test1 my-image1
/bin/sh: /bin/bash/test.sh: not found
stan@stan-VirtualBox:~/test$ docker ps -a
docker: 'ps -a' is not a docker command.
See 'docker --help'.
stan@stan-VirtualBox:~/test$ docker ps -a
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%
ck: connect: permission denied
stan@stan-VirtualBox:~/test$ sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS          PORTS          NAMES
716e7421b706   my-image1  "/bin/sh -c /bin/bas..." 2 minutes ago  Exited (127) 2 minutes ago          my-test1
stan@stan-VirtualBox:~/test$

```

- It created the container, but the image does not have bash binaries

Note: Because alpine is very light Image it does not have bash binaries.

12. Delete my-test. Run `docker rm -f my-test1`.

13. Delete my-image. Run `docker rmi -f my-image1`.


```

stan@stan-VirtualBox:~/test$ sudo docker logs my-test1
/bin/sh: /bin/bash/test.sh: not found
stan@stan-VirtualBox:~/test$ sudo docker rm -f my-test1
my-test1
stan@stan-VirtualBox:~/test$ sudo docker rmi -f my-image1
Untagged: my-image1:latest
Deleted: sha256:6f60c72e00ef0851b93e5ef50b702220c31d5f274fb1fefb60e865235ba9c356
Deleted: sha256:72654e2ef2de3b96a3424e6e122d9427a62cc47bc838bf41fa40566316da4360
Deleted: sha256:3daa008ccf2acf5fc2598779462f3ed615223086fe4d0545e844b763c0476a35
stan@stan-VirtualBox:~/test$

```

- deleting the image and the container.

Now correct your Dockerfile. In the last line replace CMD /bin/bash /test.sh with CMD /bin/sh /test.sh.

14. Build your image. Run docker build -t my-image1 ./

15. Now spawn a container again. Run docker run - --name my-test1 my-image1.

16. Do a docker ps -a. Do you see your container running?_____

```

stan@stan-VirtualBox:~/test$ sudo docker rmi -f my-image1
Untagged: my-image1:latest
Deleted: sha256:6f60c72e00ef0851b93e5ef50b702220c31d5f274fb1fefb60e865235ba9c356
Deleted: sha256:72654e2ef2de3b96a3424e6e122d9427a62cc47bc838bf41fa40566316da4360
Deleted: sha256:3daa008ccf2acf5fc2598779462f3ed615223086fe4d0545e844b763c0476a35
stan@stan-VirtualBox:~/test$ ll
total 16
drwxrwxr-x  2 stan stan 4096 Map 29 23:48 ./
drwxr-x--x 17 stan stan 4096 Map 29 23:42 ../
-rw-rw-r--  1 stan stan  50 Map 29 23:48 Dockerfile
-rw-rw-r--  1 stan stan  30 Map 29 23:43 test.sh
stan@stan-VirtualBox:~/test$ nano Dockerfile
stan@stan-VirtualBox:~/test$ sudo docker build -t my-image1 ./
Sending build context to Docker daemon  3.072kB
Step 1/3 : FROM alpine
--> 9ed4aefc74f6
Step 2/3 : ADD test.sh /
--> e5ad39a80986
Step 3/3 : CMD /bin/sh/test.sh
--> Running in 454839f0ea56
Removing intermediate container 454839f0ea56
--> 202af0d7f78e
Successfully built 202af0d7f78e
Successfully tagged my-image1:latest
stan@stan-VirtualBox:~/test$ docker run --name my-test-image1
"docker run" requires at least 1 argument.
See 'docker run --help'.

Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container
stan@stan-VirtualBox:~/test$ sudo docker run --name my-test-image1 my image1
Unable to find image 'my:latest' locally
docker: Error response from daemon: pull access denied for my, repository does not exist or may require 'docker login'
See 'docker run --help'.
stan@stan-VirtualBox:~/test$ sudo docker run --name my-test1 my-image1
/bin/sh: /bin/sh/test.sh: not found
stan@stan-VirtualBox:~/test$ sudo docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS              PORTS          NAMES
c2e4484abcca   my-image1  "/bin/sh -c /bin/sh/..." 25 seconds ago Exited (127) 23 seconds ago          my-test1
stan@stan-VirtualBox:~/test$ S

```

- Changing the Dockerfile and creating the image and container again.

17. Delete my-test. Run `docker rm -f my-test1`.

18. Delete my-image. Run `docker rmi -f my-image1`

```
stan@stan-VirtualBox:~/test$ sudo docker rm -f my-test1
my-test1
stan@stan-VirtualBox:~/test$ sudo docker rmi -f my-image1
Untagged: my-image1:latest
Deleted: sha256:202af0d7f78ecf4654d873c7c83cd0b62674dcff6eebe5b831da1d7ac6417255
Deleted: sha256:e5ad39a80986823120aa0bae1eb914d46053c3335ed8a3797c8637a651818c78
Deleted: sha256:3daa008ccf2acf5fc2598779462f3ed615223086fe4d0545e844b763c0476a35
stan@stan-VirtualBox:~/test$
```

- Deleting the test and image(2-nd time).