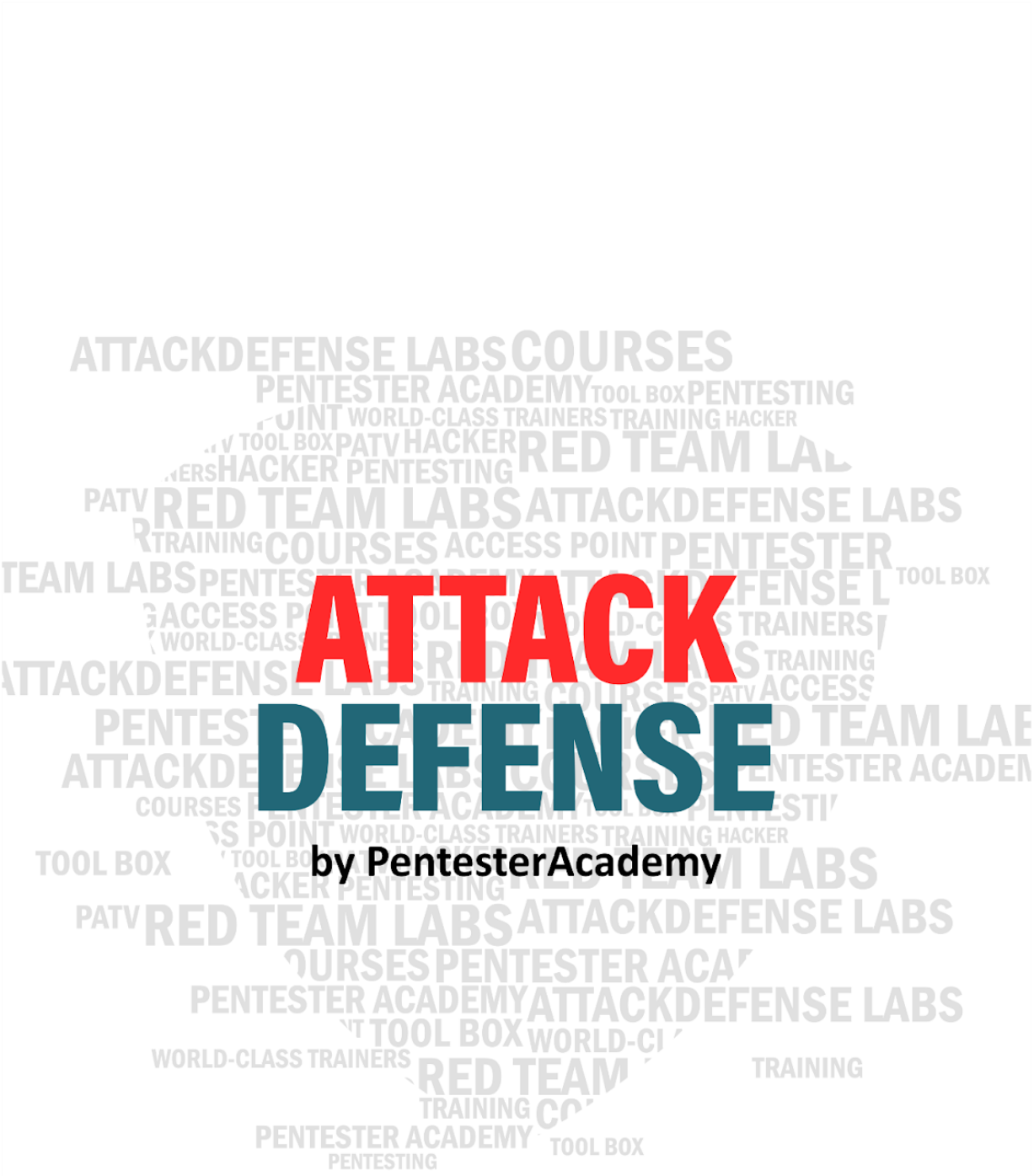
ATTACK
DEFENSE
by PentesterAcademy

Name	Docker Basics Lab
URL	https://attackdefense.com/challengedetails?cid=1342
Type	Docker Security : Container Basics

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective: Learn basic Docker commands.

1. Check Docker version

Command: docker version

```
root@localhost:~# docker version
Client: Docker Engine - Community
 Version:           19.03.1
 API version:       1.40
 Go version:        go1.12.5
 Git commit:        74b1e89
 Built:             Thu Jul 25 21:21:05 2019
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.1
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.12.5
  Git commit:       74b1e89
  Built:            Thu Jul 25 21:19:41 2019
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.2.6
  GitCommit:        894b81a4b802e4eb2a91d1ce216b8817763c29fb
```

```
 runc:
  Version:          1.0.0-rc8
  GitCommit:        425e105d5a03fabd737a126ad93d62a9eeede87f
 docker-init:
  Version:          0.18.0
  GitCommit:        fec3683
root@localhost:~#
```

2. Check host Information

Command: docker info

```
root@localhost:~# docker info
Client:
 Debug Mode: false

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 19.03.1
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 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: runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 894b81a4b802e4eb2a91d1ce216b8817763c29fb
 runc version: 425e105d5a03fabd737a126ad93d62a9eeede87f
 init version: fec3683
```

```
 Security Options:
  apparmor
  seccomp
   Profile: default
 Kernel Version: 5.0.0-20-generic
 Operating System: Ubuntu 18.04 LTS
 OSType: linux
 Architecture: x86_64
 CPUs: 2
 Total Memory: 2.929GiB
 Name: localhost
 ID: SCFL:4U3U:N4W7:MKLC:BLLZ:56YT:25GA:GYWU:7YGE:2PU3:YYVI:25Z7
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  registry:5000
  127.0.0.0/8
 Live Restore Enabled: false

WARNING: No swap limit support
```

3. Pull image

Command: docker pull registry:5000/alpine

```
root@localhost:~# docker pull registry:5000/alpine
Using default tag: latest
latest: Pulling from alpine
89d9c30c1d48: Pull complete
4dc2274c3171: Pull complete
Digest: sha256:4963a02ddb4f256659e54b6aeb85303cab638477061bb9978782dd2a5ace957f
Status: Downloaded newer image for registry:5000/alpine:latest
registry:5000/alpine:latest
root@localhost:~#
```

4. List images

Command: docker images

```
root@localhost:~# docker images
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
registry:5000/alpine    latest              84c5dcc06300        2 weeks ago         113MB
root@localhost:~#
```

Alternatively,

Command: docker image ls

```
root@localhost:~# docker image ls
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
registry:5000/alpine    latest              84c5dcc06300        2 weeks ago         113MB
root@localhost:~#
```

5. Run image in background mode

Command: docker run -dt registry:5000/alpine

```
root@localhost:~# docker run -dt registry:5000/alpine
ec8647e762aff625f701e1e38c8f8fdc31f76d9daa51d2d830a2050f73cd6951
root@localhost:~#
```

6. Run image in interactive mode

Command: docker run -it registry:5000/alpine

```
root@localhost:~# docker run -it registry:5000/alpine
/ #
/ #
```

7. List running containers

Command: docker ps

```
root@localhost:~# docker ps
CONTAINER ID        IMAGE                COMMAND             CREATED             STATUS              PORTS               NAMES
458b83284a43        registry:5000/alpine   "/bin/sh"           15 seconds ago      Up 5 seconds                            inspiring_nightingale
root@localhost:~#
```

Alternatively

Command: docker container ls

```
root@localhost:~# docker container ls
CONTAINER ID        IMAGE                COMMAND             CREATED             STATUS              PORTS               NAMES
458b83284a43        registry:5000/alpine   "/bin/sh"           2 minutes ago       Up 2 minutes                            inspiring_nightingale
root@localhost:~#
```

8. Inspect a container

Command: docker inspect 458b83284a43

```
root@localhost:~# docker inspect 458b83284a43
[
    {
        "Id": "458b83284a4337ca1ba5bfaed3328109c4855bde58dd9a74cf6be1507bdf264a",
        "Created": "2019-11-28T06:20:47.90865984Z",
        "Path": "/bin/sh",
        "Args": [],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 1061,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2019-11-28T06:20:56.54465156Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
```

```
"Config": {
    "Hostname": "458b83284a43",
    "Domainname": "",
    "User": "",
    "AttachStdin": true,
    "AttachStdout": true,
    "AttachStderr": true,
    "Tty": true,
    "OpenStdin": true,
    "StdinOnce": true,
    "Env": [
        "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
    ],
    "Cmd": [
        "/bin/sh"
    ],
    "Image": "registry:5000/alpine",
    "Volumes": null,
```

```
"NetworkSettings": {
    "Bridge": "",
    "SandboxID": "82b845a2732d010d6c9d6a675e6455d8c0516d582a979c2226a62b7cfcb4451f",
    "HairpinMode": false,
    "LinkLocalIPv6Address": "",
    "LinkLocalIPv6PrefixLen": 0,
    "Ports": {},
    "SandboxKey": "/var/run/docker/netns/82b845a2732d",
    "SecondaryIPAddresses": null,
    "SecondaryIPv6Addresses": null,
    "EndpointID": "ceeab1431ce18fb56e4dbda3915384c72cc060c1301df24459e08a2be0c0304e",
    "Gateway": "172.17.0.1",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "IPAddress": "172.17.0.2",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "MacAddress": "02:42:ac:11:00:02",
    "Networks": {
        "bridge": {
            "IPAMConfig": null,
            "Links": null,
            "Aliases": null,
            "NetworkID": "11ba47e08edf54f2ddad339b01ece62c264d7bee74ba89481d5088bafe37f50d",
            "EndpointID": "ceeab1431ce18fb56e4dbda3915384c72cc060c1301df24459e08a2be0c0304e",
            "Gateway": "172.17.0.1",
            "IPAddress": "172.17.0.2",
```

9. Manage containers

Command: docker container

```
root@localhost:~# docker container

Usage:  docker container COMMAND

Manage containers

Commands:
  attach      Attach local standard input, output, and error streams to a running container
  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
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  inspect     Display detailed information on one or more containers
  kill        Kill one or more running containers
  logs        Fetch the logs of a container
  ls          List containers
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  prune       Remove all stopped containers
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  run         Run a command in a new container
  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
```

10. Manage images

Command: docker image

```
root@localhost:~# docker image

Usage:  docker image COMMAND

Manage images

Commands:
  build       Build an image from a Dockerfile
  history     Show the history of an image
  import      Import the contents from a tarball to create a filesystem image
  inspect     Display detailed information on one or more images
  load        Load an image from a tar archive or STDIN
  ls          List images
  prune       Remove unused images
  pull        Pull an image or a repository from a registry
  push        Push an image or a repository to a registry
  rm          Remove one or more images
  save        Save one or more images to a tar archive (streamed to STDOUT by default)
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

Run 'docker image COMMAND --help' for more information on a command.
root@localhost:~#
```

11. List networks

Command: docker network ls

```
root@localhost:~# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
11ba47e08edf        bridge              bridge              local
2371877d0b47        host                host                local
282213cc5959        none                null                local
root@localhost:~#
```

12. Manage networks

Command: docker network

```
root@localhost:~# docker network

Usage:  docker network COMMAND

Manage networks

Commands:
  connect     Connect a container to a network
  create      Create a network
  disconnect  Disconnect a container from a network
  inspect     Display detailed information on one or more networks
  ls          List networks
  prune       Remove all unused networks
  rm          Remove one or more networks

Run 'docker network COMMAND --help' for more information on a command.
root@localhost:~#
```

13. Attach to a running container

Command: docker attach 458b83284a43

```
root@localhost:~# docker ps
CONTAINER ID        IMAGE                  COMMAND             CREATED             STATUS              PORTS               NAMES
458b83284a43        registry:5000/alpine   "/bin/sh"           4 minutes ago       Up 4 minutes                            inspiring_nightingale
root@localhost:~#
root@localhost:~# docker attach 458b83284a43
/ #
/ #
/ #
```

14. Execute a command in a running container

Command: docker exec -it 9b44e6a619ae /bin/sh

```
root@localhost:~# docker ps
CONTAINER ID        IMAGE                  COMMAND             CREATED             STATUS              PORTS               NAMES
9b44e6a619ae        registry:5000/alpine   "/bin/sh"           9 seconds ago       Up 1 second                             distracted_joliot
root@localhost:~#
root@localhost:~#
root@localhost:~# docker exec -it 9b44e6a619ae /bin/sh
/ #
/ #
```

15. Stop a running container

Command: docker stop 9b44e6a619ae

```
root@localhost:~# docker ps
CONTAINER ID        IMAGE                  COMMAND             CREATED             STATUS              PORTS               NAMES
9b44e6a619ae        registry:5000/alpine   "/bin/sh"           About a minute ago  Up 56 seconds                           distracted_joliot
root@localhost:~#
root@localhost:~#
root@localhost:~# docker stop 9b44e6a619ae
9b44e6a619ae
root@localhost:~#
```

16. Start a stopped container

Command: docker start 9b44e6a619ae

```
root@localhost:~# docker start 9b44e6a619ae
9b44e6a619ae
root@localhost:~#
root@localhost:~# docker ps
CONTAINER ID        IMAGE                  COMMAND             CREATED             STATUS              PORTS               NAMES
9b44e6a619ae        registry:5000/alpine   "/bin/sh"           2 minutes ago       Up 7 seconds                            distracted_joliot
root@localhost:~#
```

17. Kill a running container

Command: docker kill 9b44e6a619ae

```
root@localhost:~# docker kill 9b44e6a619ae
9b44e6a619ae
root@localhost:~#
```

18. Build Docker image

Create a Dockerfile.

Note: The lab is NOT connected to the internet. Hence, installations won't work.

```
root@localhost:~# cat Dockerfile
FROM registry:5000/alpine

COPY script /root/

RUN chmod +x /root/script
root@localhost:~#
```

This dockerfile takes locally present alpine image as base and copies a script in it.

Create a dummy script.

```
root@localhost:~# cat script
Dummy script
root@localhost:~#
```

Command: docker build -t registry:5000/alpine-mod

```
root@localhost:~# docker build -t registry:5000/alpine-mod .
Sending build context to Docker daemon  22.53kB
Step 1/3 : FROM registry:5000/alpine
 ---> 84c5dcc06300
Step 2/3 : COPY script /root/
 ---> c65904461f44
Step 3/3 : RUN chmod +x /root/script
 ---> Running in 469b9702c3d3
Removing intermediate container 469b9702c3d3
 ---> 896f0e5e18af
Successfully built 896f0e5e18af
Successfully tagged registry:5000/alpine-mod:latest
root@localhost:~#
root@localhost:~# docker images
REPOSITORY                 TAG                 IMAGE ID            CREATED             SIZE
registry:5000/alpine-mod   latest              896f0e5e18af        23 seconds ago      113MB
registry:5000/alpine       latest              84c5dcc06300        2 weeks ago         113MB
root@localhost:~#
```

19. Push the image

Command: docker push registry:5000/alpine-mod

```
root@localhost:~# docker push registry:5000/alpine-mod
The push refers to repository [registry:5000/alpine-mod]
86b30fb13a33: Pushed
31caf3adb26f: Pushed
2fbae97c021e: Mounted from alpine
77cae8ab23bf: Mounted from alpine
latest: digest: sha256:ba8e142aa446bb2b2bc60859e77f6bd5ceaac75ae22a0130c95fb149fe6a3c63 size: 1154
root@localhost:~#
```

20. Commit container as an image

Create a container by running an image. Make some changes to it.

```
root@localhost:~# docker run -it registry:5000/alpine sh
/ #
/ # cd /root
~ # echo "Want to save this change" > note.txt
~ #
~ # cat note.txt
Want to save this change
~ #
~ #
```

Here, a text file is created in it. Now, commit it.

Command: docker commit 182329d8e111 registry:5000/alpine-note

```
root@localhost:~# docker ps
CONTAINER ID        IMAGE                  COMMAND             CREATED             STATUS              PORTS               NAMES
182329d8e111        registry:5000/alpine   "sh"                53 seconds ago      Up 44 seconds                           busy_pike
root@localhost:~#
root@localhost:~# docker commit 182329d8e111 registry:5000/alpine-note
sha256:f33b92a8890af915633c3ac213ace240d7265d7b0bf483446c0ca53cf5b8d21a
root@localhost:~#
root@localhost:~# docker images
REPOSITORY                  TAG                 IMAGE ID            CREATED             SIZE
registry:5000/alpine-note   latest              f33b92a8890a        7 seconds ago       113MB
registry:5000/alpine-mod    latest              896f0e5e18af        3 minutes ago       113MB
registry:5000/alpine        latest              84c5dcc06300        2 weeks ago         113MB
root@localhost:~#
```

21. Export an image as tar archive

Command: docker export -o alpine.tar 182329d8e111

```
root@localhost:~# docker ps
CONTAINER ID        IMAGE                  COMMAND             CREATED             STATUS              PORTS               NAMES
182329d8e111        registry:5000/alpine   "sh"                19 minutes ago      Up 19 minutes                           busy_pike
root@localhost:~#
root@localhost:~# docker export -o alpine.tar 182329d8e111
root@localhost:~#
root@localhost:~# ls -l
total 116448
-rw-r--r-- 1 root root        73 Nov 28 06:32 Dockerfile
-rw------- 1 root root 119231488 Nov 28 06:55 alpine.tar
-rw-r--r-- 1 root root        13 Nov 28 06:32 script
root@localhost:~#
```

22. Remove stopped containers

List all running and stopped containers

Command: docker ps -a

```
root@localhost:~# docker ps -a
CONTAINER ID        IMAGE                  COMMAND             CREATED             STATUS                      PORTS               NAMES
182329d8e111        registry:5000/alpine   "sh"                26 minutes ago      Up 26 minutes                                   busy_pike
1fa8ae5d1556        registry:5000/alpine   "/bin/sh"           32 minutes ago      Exited (130) 32 minutes ago                     naughty_wi
les
9b44e6a619ae        registry:5000/alpine   "/bin/sh"           34 minutes ago      Exited (137) 31 minutes ago                     distracted
_joliot
458b83284a43        registry:5000/alpine   "/bin/sh"           40 minutes ago      Exited (0) 35 minutes ago                       inspiring_
nightingale
d80519f171d8        registry:5000/alpine   "/bin/sh"           41 minutes ago      Exited (0) 41 minutes ago                       busy_allen
7961ad7bd1bd        registry:5000/alpine   "/bin/sh"           42 minutes ago      Exited (0) 42 minutes ago                       dazzling_k
ilby
root@localhost:~#
```

Take container IDs of desired containers and delete them.

Command: docker docker rm 9b44e6a619ae 458b83284a43 d80519f171d8 7961ad7bd1bd

```
root@localhost:~# docker rm 9b44e6a619ae 458b83284a43 d80519f171d8 7961ad7bd1bd
9b44e6a619ae
458b83284a43
d80519f171d8
7961ad7bd1bd
root@localhost:~# docker ps -a
CONTAINER ID        IMAGE                  COMMAND             CREATED             STATUS                          PORTS               NAMES
182329d8e111        registry:5000/alpine   "sh"                26 minutes ago      Up 26 minutes                                       busy_pike
1fa8ae5d1556        registry:5000/alpine   "/bin/sh"           33 minutes ago      Exited (130) 33 minutes ago                         naughty_wi
les
root@localhost:~#
```

23. Remove an image

List all images present in the local storage.

```
root@localhost:~# docker images
REPOSITORY                   TAG                 IMAGE ID            CREATED             SIZE
registry:5000/alpine-note    latest              f33b92a8890a        25 minutes ago      113MB
registry:5000/alpine-mod     latest              896f0e5e18af        29 minutes ago      113MB
registry:5000/alpine         latest              84c5dcc06300        2 weeks ago         113MB
root@localhost:~#
```

Remove image by name

Command: docker rmi -f registry:5000/alpine-note

```
root@localhost:~# docker rmi -f registry:5000/alpine-note
Untagged: registry:5000/alpine-note:latest
Deleted: sha256:f33b92a8890af915633c3ac213ace240d7265d7b0bf483446c0ca53cf5b8d21a
Deleted: sha256:856b7da3f597d1a344f3d0a4d35b4a6ddaf6fd8b9b32c0d6ee93390468ae510c
root@localhost:~#
```

Remove image by Image ID

Command: docker rmi -f 896f0e5e18af

```
root@localhost:~# docker rmi -f 896f0e5e18af
Untagged: registry:5000/alpine-mod:latest
Untagged: registry:5000/alpine-mod@sha256:ba8e142aa446bb2b2bc60859e77f6bd5ceaac75ae
Deleted: sha256:896f0e5e18af0509308a74d899139845daeda5b7a7e2b39be388259557b91771
Deleted: sha256:ca770f4bf1ea1199d3ec0439e70555f1c34eaf4cff38a7df14e8088a88f0bbd1
Deleted: sha256:c65904461f447eaa2b6bb07e25b42df1b31505023d3a08b29ae09f19fdfe749c
Deleted: sha256:9c9e8cae517db8d6fa2170b149faba2f04b94369599f879ac25ad507bee30649
root@localhost:~#
```

Check the image list to verify the deletion.

```
root@localhost:~# docker images
REPOSITORY             TAG                 IMAGE ID            CREATED             SIZE
registry:5000/alpine   latest              84c5dcc06300        2 weeks ago         113MB
root@localhost:~#
```

24. Remove stopped container, unused images/networks

Command: docker system prune -a

```
root@localhost:~# docker system prune -a
WARNING! This will remove:
  - all stopped containers
  - all networks not used by at least one container
  - all images without at least one container associated to them
  - all build cache

Are you sure you want to continue? [y/N] y
Deleted Containers:
1fa8ae5d1556725275553b4ed7af14416c556ac6195fb44fdac88fb2722f4b0a

Total reclaimed space: 5B
root@localhost:~#
```

25. Building image

The lab is not connected to the internet so only a limited number of packages present in apt-cache server can be used while building the image.

Following packages are available:

man-db manpages apt-transport-https ca-certificates curl gnupg-agent software-properties-common net-tools ssh netcat-traditional libcap2-bin kmod linux-headers-5.0.0-20-generic less file iproute2 gdb nasm supervisor openssh-server vim gcc g++ ltrace strace make

Create a dockerfile and use the “registry:5000/ubuntu-base:18.04” image as base.

Dockerfile:

```
FROM registry:5000/ubuntu-base:18.04

RUN apt-get update \
    && apt-get install -y sudo man-db manpages\
    curl \
    net-tools \
    ssh
```

```
root@localhost:~# cat Dockerfile
FROM registry:5000/ubuntu-base:18.04

RUN apt-get update \
    && apt-get install -y sudo man-db manpages\
    curl \
    net-tools \
    ssh
```

The image can also be pulled in advance.

Command: docker pull registry:5000/ubuntu-base:18.04

```
root@localhost:~# docker pull registry:5000/ubuntu-base:18.04
18.04: Pulling from ubuntu-base
7ddbc47eeb70: Pull complete
c1bbdc448b72: Pull complete
8c3b70e39044: Pull complete
45d437916d57: Pull complete
1368264894a9: Pull complete
adc817f2db40: Pull complete
Digest: sha256:0fe1936e472959e093d152c6b000c4c8c390a77f061c396a16e42bac95da4955
Status: Downloaded newer image for registry:5000/ubuntu-base:18.04
registry:5000/ubuntu-base:18.04
root@localhost:~#
```

Now, start the build process, --network=host is make sure that /etc/hosts and /etc/resolv.conf are used from host machine.

Command: docker build --network=host -t test-ubuntu .

```
root@localhost:~# docker build --network=host -t test-ubuntu .
Sending build context to Docker daemon  22.02kB
Step 1/2 : FROM registry:5000/ubuntu-base:18.04
 ---> 7e4ccdb00303
Step 2/2 : RUN apt-get update     && apt-get install -y sudo man-db manpages
 ---> Running in 505efccb5208
Err:1 http://archive.ubuntu.com/ubuntu bionic InRelease
  503  Unable to download in offline mode [IP: 192.27.164.5 3142]
Get:2 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
```

```
debconf: falling back to frontend: Readline
Building database of manual pages ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Removing intermediate container 505efccb5208
 ---> 7f9feb22c633
Successfully built 7f9feb22c633
Successfully tagged test-ubuntu:latest
root@localhost:~#
```

The build process is finished and the image is ready.

```
root@localhost:~# docker images
REPOSITORY                  TAG                 IMAGE ID            CREATED             SIZE
test-ubuntu                 latest              7f9feb22c633        19 minutes ago      366MB
registry:5000/ubuntu-base   18.04               7e4ccdb00303        35 minutes ago      351MB
root@localhost:~#
```

The “test-ubuntu” image is ready.

References:

1. Docker (https://www.docker.com/)