ATTACK
DEFENSE
by PentesterAcademy

Name	Podman Basics Lab
URL	https://attackdefense.com/challengedetails?cid=1451
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 Podman commands.

1. Check Podman version

Command: podman --version

```
root@localhost:~# podman --version
podman version 1.6.2
root@localhost:~#
```

2. Check host Information

Command: podman info

```
root@localhost:~# podman info
host:
  BuildahVersion: 1.11.3
  CgroupVersion: v1
  Conmon:
    package: 'conmon: /usr/bin/conmon'
    path: /usr/bin/conmon
    version: 'conmon version 2.0.3, commit: unknown'
  Distribution:
    distribution: ubuntu
    version: "18.04"
  MemFree: 2974388224
```

```
OCIRuntime:
  name: runc
  package: 'containerd.io: /usr/bin/runc'
  path: /usr/bin/runc
  version: |-
    runc version 1.0.0-rc8
    commit: 425e105d5a03fabd737a126ad93d62a9eeede87f
    spec: 1.0.1-dev
```

```
  arch: amd64
  cpus: 2
  eventlogger: journald
  hostname: localhost
  kernel: 5.0.0-20-generic
  os: linux
  rootless: false
  uptime: 10m 50.55s
registries:
  blocked: null
  insecure:
  - registry:5000
  search:
  - registry:5000
```

3. Pull image

Command: podman pull registry:5000/alpine

```
root@localhost:~# podman pull registry:5000/alpine
Trying to pull registry:5000/alpine...
Getting image source signatures
Copying blob 89d9c30c1d48 done
Copying blob 4dc2274c3171 done
Copying config 84c5dcc063 done
Writing manifest to image destination
Storing signatures
84c5dcc06300eaba56eed7a7f94f54cf9ad2658f045fe86c42825296914a9857
root@localhost:~#
```

4. List images

Command: podman images

```
root@localhost:~# podman images
REPOSITORY            TAG       IMAGE ID        CREATED       SIZE
registry:5000/alpine  latest    84c5dcc06300    2 weeks ago   119 MB
root@localhost:~#
```

5. Run image in background mode

Command: podman run -dt registry:5000/alpine

```
root@localhost:~# podman run -dt registry:5000/alpine sh
b42cb8f41a887b5631d8fcd3a8ba2bacbf19045126471d382ad39bef1316d777
root@localhost:~#
```

6. Run image in interactive mode

Command: podman run -it registry:5000/alpine sh

```
root@localhost:~# podman run -it registry:5000/alpine sh
/ #
/ # pwd
/
/ # whoami
root
```

7. List running containers

Command: podman ps

```
root@localhost:~# podman ps
CONTAINER ID  IMAGE                        COMMAND  CREATED            STATUS             PORTS  NAMES
b42cb8f41a88  registry:5000/alpine:latest  sh       About a minute ago Up 57 seconds ago         lucid_moore
root@localhost:~#
```

8. Inspect a container

Command: podman inspect b42cb8f41a88

```
root@localhost:~# podman inspect b42cb8f41a88
[
    {
        "Id": "b42cb8f41a887b5631d8fcd3a8ba2bacbf19045126471d382ad39bef1316d777",
        "Created": "2019-11-29T05:40:44.16354436Z",
        "Path": "sh",
        "Args": [
            "sh"
        ],
        "State": {
            "OciVersion": "1.0.1-dev",
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 701,
```

```
            "EffectiveCaps": [
                "CAP_CHOWN",
                "CAP_DAC_OVERRIDE",
                "CAP_FSETID",
                "CAP_FOWNER",
                "CAP_MKNOD",
                "CAP_NET_RAW",
                "CAP_SETGID",
                "CAP_SETUID",
                "CAP_SETFCAP",
                "CAP_SETPCAP",
                "CAP_NET_BIND_SERVICE",
                "CAP_SYS_CHROOT",
                "CAP_KILL",
                "CAP_AUDIT_WRITE"
            ],
```

```
"NetworkSettings": {
    "Bridge": "",
    "SandboxID": "",
    "HairpinMode": false,
    "LinkLocalIPv6Address": "",
    "LinkLocalIPv6PrefixLen": 0,
    "Ports": [],
    "SandboxKey": "/var/run/netns/cni-84f79dd9-7580
    "SecondaryIPAddresses": null,
    "SecondaryIPv6Addresses": null,
    "EndpointID": "",
    "Gateway": "10.88.0.1",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "IPAddress": "10.88.0.4",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "MacAddress": "8a:51:59:f2:c4:58"
},
```

9. Manage containers

Command: podman container --help

```
root@localhost:~# podman container --help
Manage containers

Usage:
  podman container [command]

Available Commands:
  attach      Attach to a running container
  checkpoint  Checkpoints one or more containers
  cleanup     Cleanup network and mountpoints of one or more containers
  commit      Create new image based on the changed container
  cp          Copy files/folders between a container and the local filesystem
  create      Create but do not start a container
  diff        Inspect changes on container's file systems
  exec        Run a process in a running container
  exists      Check if a container exists in local storage
```

```
export      Export container's filesystem contents as a tar archive
init        Initialize one or more containers
inspect     Display the configuration of a container
kill        Kill one or more running containers with a specific signal
list        List containers
logs        Fetch the logs of a container
mount       Mount a working container's root filesystem
pause       Pause all the processes in one or more containers
port        List port mappings or a specific mapping for the container
prune       Remove all stopped containers
restart     Restart one or more containers
restore     Restores one or more containers from a checkpoint
```

10. Manage images

Command: podman image --help

```
root@localhost:~# podman image --help
Manage images

Usage:
  podman image [command]

Available Commands:
  build      Build an image using instructions from Containerfiles
  exists     Check if an image exists in local storage
  history    Show history of a specified image
  import     Import a tarball to create a filesystem image
  inspect    Display the configuration of an image
  list       List images in local storage
  load       Load an image from container archive
  prune      Remove unused images
  pull       Pull an image from a registry
  push       Push an image to a specified destination
  rm         Removes one or more images from local storage
  save       Save image to an archive
  sign       Sign an image
  tag        Add an additional name to a local image
  tree       Prints layer hierarchy of an image in a tree format
```

11. List networks

Command: podman network ls

```
root@localhost:~# podman network ls
NAME       VERSION    PLUGINS
podman     0.4.0      bridge,portmap,firewall
root@localhost:~#
```

12. Manage networks

Command: podman network --help

```
root@localhost:~# podman network --help
Manage networks

Usage:
  podman network [command]

Available Commands:
  create      network create
  inspect     network inspect
  ls          network list
  rm          network rm
```

13. Attach to a running container

Command: podman attach b42cb8f41a88

```
root@localhost:~# podman ps
CONTAINER ID   IMAGE                        COMMAND   CREATED         STATUS          PORTS   NAMES
b42cb8f41a88   registry:5000/alpine:latest  sh        8 minutes ago   Up 8 minutes ago        lucid_moore
root@localhost:~#
root@localhost:~#
root@localhost:~# podman attach b42cb8f41a88

/ # whoami
root
```

14. Execute a command in a running container

Command: podman exec -it b42cb8f41a88 /bin/sh

```
root@localhost:~# podman exec -it b42cb8f41a88 /bin/sh
/ #
/ # whoami
root
/ #
```

15. Stop a running container

Command: podman stop b42cb8f41a88

```
root@localhost:~# podman ps
CONTAINER ID  IMAGE                        COMMAND   CREATED        STATUS             PORTS   NAMES
b42cb8f41a88  registry:5000/alpine:latest  sh        11 minutes ago Up 10 minutes ago          lucid_moore
root@localhost:~#
root@localhost:~#
root@localhost:~# podman stop b42cb8f41a88
b42cb8f41a887b5631d8fcd3a8ba2bacbf19045126471d382ad39bef1316d777
root@localhost:~#
root@localhost:~# podman ps
CONTAINER ID  IMAGE  COMMAND  CREATED  STATUS  PORTS  NAMES
root@localhost:~#
```

16. Kill a running container

Command: podman kill d063334743c0

```
root@localhost:~# podman ps
CONTAINER ID  IMAGE                        COMMAND   CREATED        STATUS            PORTS   NAMES
d063334743c0  registry:5000/alpine:latest  sh        10 seconds ago Up 5 seconds ago          fervent_goldwasser
root@localhost:~#
root@localhost:~# podman kill d063334743c0
d063334743c0e10401b7a9a79457ee462ceac823d37dea853ed75dc2ae24d942
root@localhost:~#
root@localhost:~# podman ps
CONTAINER ID  IMAGE  COMMAND  CREATED  STATUS  PORTS  NAMES
root@localhost:~#
```

17. Build Podman image

Create a containerfile.

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

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

COPY script /root/

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

This containerfile 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: podman build -t registry:5000/alpine-mod -f containerfile .

```
root@localhost:~# podman build -t registry:5000/alpine-mod -f containerfile .
STEP 1: FROM registry:5000/alpine:latest
STEP 2: COPY script /root/
6818042d0ca6b5dfd3133828819272d1bf2ce00bc474827165d8301fc7419aa3
STEP 3: RUn chmod +x /root/script
STEP 4: COMMIT registry:5000/alpine-mod
6f2fad5fa68f8bdb45be9b42cb7f16fa1afbd807db7849e557eae0bc6b1db741
root@localhost:~#
root@localhost:~#
root@localhost:~# podman images
REPOSITORY              TAG      IMAGE ID       CREATED          SIZE
registry:5000/alpine-mod    latest   6f2fad5fa68f   52 seconds ago   119 MB
registry:5000/alpine        latest   84c5dcc06300   2 weeks ago      119 MB
root@localhost:~#
```

19. Push the image

Command: podman push registry:5000/alpine-mod

```
root@localhost:~# podman push registry:5000/alpine-mod
Getting image source signatures
Copying blob 96eba573f652 done
Copying blob e5f6aea9e83f done
Copying blob 2fbae97c021e skipped: already exists
Copying blob 77cae8ab23bf skipped: already exists
Copying config 6f2fad5fa6 done
Writing manifest to image destination
Storing signatures
root@localhost:~#
```

20. Commit container as an image

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

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

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

Command: podman commit 879eba3ce455 registry:5000/alpine-note

```
root@localhost:~# podman ps
CONTAINER ID   IMAGE                        COMMAND   CREATED         STATUS             PORTS   NAMES
879eba3ce455   registry:5000/alpine:latest  sh        45 seconds ago  Up 40 seconds ago          distracted_mayer
root@localhost:~#
root@localhost:~# podman commit 879eba3ce455 registry:5000/alpine-note
Getting image source signatures
Copying blob 77cae8ab23bf skipped: already exists
Copying blob 2fbae97c021e skipped: already exists
Copying blob bb849cbea819 done
Copying config 1e760bf8c2 done
Writing manifest to image destination
Storing signatures
1e760bf8c2568378b50fd8b0955615b9253d526c521e68d7f392c29437a1d338
root@localhost:~#
root@localhost:~# podman images
REPOSITORY                TAG      IMAGE ID       CREATED         SIZE
registry:5000/alpine-note latest   1e760bf8c256   14 seconds ago  119 MB
registry:5000/alpine-mod  latest   6f2fad5fa68f   5 minutes ago   119 MB
registry:5000/alpine      latest   84c5dcc06300   2 weeks ago     119 MB
root@localhost:~#
```

21. Export an image as tar archive

Command: podman export -o alpine.tar f85a6425bacd

```
root@localhost:~# podman ps
CONTAINER ID   IMAGE                         COMMAND   CREATED         STATUS            PORTS   NAMES
f85a6425bacd   registry:5000/alpine:latest   sh        11 seconds ago  Up 5 seconds ago          angry_satoshi
root@localhost:~#
root@localhost:~# podman export -o alpine.tar f85a6425bacd
root@localhost:~#
root@localhost:~# ls -l
total 116444
-rw-r--r-- 1 root root 119228928 Nov 29 06:07 alpine.tar
-rw-r--r-- 1 root root        80 Nov 29 05:58 containerfile
-rw-r--r-- 1 root root        13 Nov 29 05:58 script
root@localhost:~#
```

22. Remove stopped containers

List all running and stopped containers

Command: podman ps -a

```
root@localhost:~# podman ps -a
CONTAINER ID  IMAGE                      COMMAND  CREATED             STATUS                     PORTS  NAMES
f85a6425bacd  registry:5000/alpine:latest  sh       About a minute ago  Up About a minute ago             angry_satoshi
879eba3ce455  registry:5000/alpine:latest  sh       4 minutes ago       Exited (0) About a minute ago     distracted_mayer
d063334743c0  registry:5000/alpine:latest  sh       12 minutes ago      Exited (137) 11 minutes ago       fervent_goldwasser
f0e3bd8849e9  registry:5000/alpine:latest  sh       12 minutes ago      Exited (0) 12 minutes ago         heuristic_lehmann
b42cb8f41a88  registry:5000/alpine:latest  sh       27 minutes ago      Exited (137) 16 minutes ago       lucid_moore
7ee012217b3c  registry:5000/alpine:latest  sh       28 minutes ago      Exited (130) 26 minutes ago       zealous_goodall
```

Take container IDs of desired containers and delete them.

Command: podman rm 879eba3ce455 d063334743c0 b42cb8f41a88 7ee012217b3c f0e3bd8849e9

```
root@localhost:~# podman rm 879eba3ce455 d063334743c0 b42cb8f41a88 7ee012217b3c f0e3bd8849e9
879eba3ce45587d045a6e955d55d3680d27ab747eaf3c3cc84708229e456ae44
d063334743c0e10401b7a9a79457ee462ceac823d37dea853ed75dc2ae24d942
b42cb8f41a887b5631d8fcd3a8ba2bacbf19045126471d382ad39bef1316d777
7ee012217b3c0d19a77be6f5e51f5134fad6c95534d5a0c27ea5033cd67a8bbb
f0e3bd8849e92b743b3200044818bdf9baabe1f4ba0b728d2d1bcee1c8eceb4a
root@localhost:~#
root@localhost:~# podman ps -a
CONTAINER ID  IMAGE                      COMMAND  CREATED        STATUS         PORTS  NAMES
f85a6425bacd  registry:5000/alpine:latest  sh       2 minutes ago  Up 2 minutes ago      angry_satoshi
root@localhost:~#
```

23. Remove an image

List all images present in the local storage.

```
root@localhost:~# podman images
REPOSITORY                 TAG     IMAGE ID      CREATED         SIZE
registry:5000/alpine-note  latest  1e760bf8c256  5 minutes ago   119 MB
registry:5000/alpine-mod   latest  6f2fad5fa68f  10 minutes ago  119 MB
registry:5000/alpine       latest  84c5dcc06300  2 weeks ago     119 MB
root@localhost:~#
```

Remove image by name

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

```
root@localhost:~# podman rmi -f registry:5000/alpine-note
Untagged: registry:5000/alpine-note:latest
Deleted: 1e760bf8c2568378b50fd8b0955615b9253d526c521e68d7f392c29437a1d338
root@localhost:~#
```

Remove image by Image ID

Command: podman rmi -f 896f0e5e18af

```
root@localhost:~# podman rmi -f 6f2fad5fa68f
6818042d0ca6b5dfd3133828819272d1bf2ce00bc474827165d8301fc7419aa3
Untagged: registry:5000/alpine-mod:latest
Deleted: 6f2fad5fa68f8bdb45be9b42cb7f16fa1afbd807db7849e557eae0bc6b1db741
root@localhost:~#
```

Check the image list to verify the deletion.

```
root@localhost:~# podman images
REPOSITORY             TAG      IMAGE ID      CREATED       SIZE
registry:5000/alpine   latest   84c5dcc06300  2 weeks ago   119 MB
root@localhost:~#
```

24. Remove stopped container, unused images/networks

Command: podman system prune -a

```
root@localhost:~# podman system prune -a

WARNING! This will remove:
        - all stopped containers
        - all stopped pods
        - all dangling images
        - all build cache
Are you sure you want to continue? [y/N] y
Deleted Pods
Deleted Containers
f85a6425bacdaf1f2eede84b6e5ad534f89ceaed3ec7080aa4de66cce1995d3c
Deleted Images
84c5dcc06300eaba56eed7a7f94f54cf9ad2658f045fe86c42825296914a9857
```

References:

1. Podman (https://podman.io/)