

Session: 159

DO/BO

```
# podman run -itd --name os1 httpd
```

```
# runs /etc/container
```

↳ ~~uses~~ ~~pulls~~ the from which
registry to download the images/
containers

```
# podman inspect os1 -f
```

"{{.NetworkSettings.IPAddress}}"

↳ bcz it is in root block

```
# podman inspect os1 -f "{{.State.Status}}"
```

```
# podman ps --format="{{.ID}}{{.Names}}
```

```
# podman pull mysql
```

● /var/lib/mysql

↳ all the databases are stored in
this dir.

```
# setenforce 1
```

↳ Enforcing

```
# podman run -dit --name mydb  
-e MYSQL_USER=vimel -e MYSQL_PASSWORD=  
redhat -e MYSQL_DBNAME=wpdb  
-e MYSQL_ROOT_PASSWORD = redhat  
-v /dbdata:/var/lib/mysql mysql  
nospecy
```

↳ They will start and exit
why? bcc of selinux

```
# cd /dbdata/  
↳ Nothing here
```

```
# setenforce 0
```

```
# podman run -dit --name mydb2 mysql
```

```
# ls /dbdata/
```

↳ data comes up

• The data in /dbdata will be
~~per~~ permanent. we won't lose it
when we delete the container/db.

• SELinux doesn't allow mysql / containers
to work on mandatory Access Control

```
# ls -ld /dbdata
```

↳ also lists the selinux status

↳ -- : default_t: so
containers can't access it

```
# chcon dbdata -t container_type_t  
↳ changes selinux permission / dbdata
```

command ~~exit~~ or `!/dbdata/exit`

- New `/dbdata` file will provide all the data of the newly launched database (`mysql` container)



IMPORTANT CHAPTERS/TOPICS

Ch: 2.3

Ch: 3.4

Ch: 5.5

Ch: 6.4 / 6.6 / 6.10 / 6.11

Ch: 6.2 / 7.4 / 7.5

EX-180 BOOTCAMP DAY-1

duration : 2 hrs

- Login to the portal 30 min prior to the exam.

Activities → Examview (icon)

↳ Browser opens & exam paper.

- DONT READ INSTRUCTIONS

- Questions are asked

↳ (4 predominant + 3 QC)

- Q1 is the biggest & most complex question

- all the questions to be attempted on the terminal.

- we have to work with a user. root won't be provided.

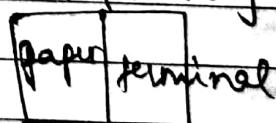
- Exam will be held live so the ^{user} experience won't be smooth.

- Decrease the font so we won't have to scroll much.

- split both the windows on left & right

ctrl + shift + +/-

ctrl + +/-



- check for the workspace and attempt the question in it only.

After attempting a question TEST it &
 make a BACKUP of it

cp -rf workspace /temp/b1

~~cp -rf workspace /temp/b2~~

Q1. Create our own custom img using Dockerfile (Config app server)

↳ They give pre-created dockerfile with all the keywords. We only have to fill the blanks.

APPServer

JBoss → install (tar.gz)

tar -zvxf jboss-6.0.0.Final.tar.gz

• we find the tar.gz in cd /home/cusrer

• DONT PLAY WITH COMMENTS. They are there to confuse/guide.

^{pre written} # Dockerfile

^{keywords} → FROM jboss/base-jdk:11

COPY/wildfly-23.0.1.Final.tar.gz .

pedman build -t zl:v1 .

pedman run -u zl:v1 .

↳ ls

If they give software in .zip

unzip <software>.zip

* Dockerfile

```
RUN tar -xzf wildfly-23-.tar.gz
```

* predominant build -t 7f:vb

RUN ONE THING AND TEST

- COPY

↳ only copies

- ADD

↳ a copies & extracts

↳ bug in DO180 don't work as expected

* Dockerfile

ENV JBOSS_HOME /opt/jboss/wild-final

ENV ~~JBOSS~~ LAUNCH-JBOSS-IN-BACKGROUND true

USER jboss

8080 ← apps run on this port
 jboss ← 9990 ← config port
 ↘ 9999

EXPOSE 8080

EXPOSE 9990

EXPOSE 9999

confirm

<space>

RUN ~~/opt/jboss/wildfly/bin/add-user.sh~~

admin Admin#70365 -silent

no extra space before options

```
CMD ["${GOPATH}/bin/wildfly/bin/standalone.sh",
      "-b", "0.0.0.0", "-bmanagement",
      "0.0.0.0"]
```

- Don't change any pre-created cmd.

* ~~podman build -t webapp:latest~~

image name will be given in the question

* ~~podman run -it -p 8080:8080~~
webapp:latest

* Firefox : 127.0.0.1:8080
: 127.0.0.1:9990

~~podman~~

~~Z~~

- Print last 10 lines of log file

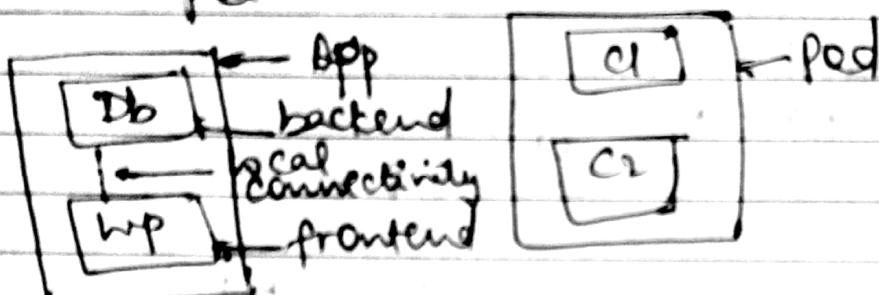
podman logs --tail 2 --since

EX-180 BOOTCAMP DAY - 2

podman ^{pod} create -name wped

podman run -dit --name w0l vimall3/ --pod wped

↳ Creates the container inside the wped pod.



↳ Local connectivity is provided by the loopback ip (127.0.0.1)

podman pod create wapp

c1 # podman run -d --pod wapp --name=wptest

~~--label WORDPRESS_DB_HOST="127.0.0.1" --name=wptest-wp~~

-v /opt/dbdata:/var/lib/mysql
↳ mount dir

c2 #

-e WORDPRESS_DB_HOST="127.0.0.1" --name=wptest-wp
-p 8080:81

127.0.0.1:8080 ↳ firefox

- To test run the pod -f then again install it and check if the blog is still accessible.

- BEFORE IT create an account & write a test blog.

pedman tag workstation/webapp:latest
workstation.example.com:5000/webapp:
↳ tags the img

pedman push workstation.example.com:
5000/webapp:7.2.3
↳ pushes to local priv. registry

pedman images | grep webapp
↳ → → dir/path

pedman save, webapp:latest

↳ creates a backup of the img

ex: /home/nimrat/w.tar

- UH

username } of openshift will be in
password } the instructions

oc login -u <username>

-p <password>

Sol 6:

oc new-project <prname> --description='my test proj'

newapp = pod

(1)

DB

↳ postgres

(2)

Nodejs

↳ fm4UH

↳ Create fm template

#

oc create -f <url>

↳ downloads the template

oc get templates in openshift kgroup

↳ also avail in naming spec

```

    graph LR
      PG[PostgreSQL Pod] ---|> FM4UH[fm4UH Pod]
      PG ---|> FM4UH
      subgraph SharedVolume [Shared Volume]
        PG
        FM4UH
      end
  
```

oc get templates postgresql-ephemeral
in openshift

oc describe templates postgresql-ephemeral
in openshift

parameters
oc new-app postgresql-eph.
→ p POSTGRESQL_USER="vimal" → p = mydb
POSTGRESQL_PWD="redhat" → p
p → DATABASE="lwdb" → p DDBBNAME SERVICE NAME

-1 "app=mysqldb"

oc get pods

↳ pod created

oc get svc

↳ mysqls

oc expose svc mysqls-ex

↳ new svc

oc get route

↳ give the url automatically

* for custom url

oc expose svc mysqls-ex --hostname=
custom

oc get pods -o wide | grep Running |

grep mysql | awk '{print \$6}'

vim ip.sh

* bash
↳ oc get pods -n kube-system |

awk '{print \$1}' > ip.sh

- Complete paper within 1:00hr — 1:15hr

- Create backup of q1 and read

- the question in every detail

- (in case we miss something) and again run it.