Day 1 POC:

To check all Docker images

* Docker images

Download and run Ubuntu image

* Docker run –it Ubuntu (image name) /bin/bash

Set new password for image

* # Passwd

Check ip for container

* # Ifconfig

Login to Ubuntu docker container

* 1. Inside container (#) : cd /etc/ssh/
* # Vim sshd\_config
* # In this file in authentication do permitrootlogin to yes
* Open new terminal and run command: ssh root(username)@IP of(hostname) container.
* E.g. ssh root @172.17.0.6

Day 1 POC 2

Create a folder.

Put the dockerfile and your war file inside this folder.

Dockerfile

FROM ubuntu

RUN apt-get update

RUN apt-get install -y --force-yes openjdk-7-jre-headless wget

RUN apt-get install -y tomcat7

ADD jspservletIntro.war /var/lib/tomcat7/webapps/

EXPOSE 8080

CMD service tomcat7 start && tail -f /var/lib/tomcat7/catalina.out

How to build and name dockerfile?

Go the location of that folder in which your dockerfile is there and run the command:

* Docker build –t “name of image” .
* Your image will get created by the name which you will mention in above command.

Start service of tomcat inside the container

* Service start tomcat7.

Check IP address

* Ifconfig

Go to the location inside container /var/lib/tomcat7/webapps

Check whether your war file is there or not.

Hit the URL with ip address of container and port no which you have mentioned in dockerfile

e.g. 172.17.0.6:8080/EmployeeApplication

Port Forwarding:

Docker run –p 172.27.59.200(IP of VM machine):8090(forwarded port):8080(port you want to forward) grp8poc2(image name) /bin/bash

Start tomcat service inside and outside container.

**POC 4:**

**Using Mysql of windows and application in linux.**

Install mysql windows (local machine)

Bind address in my.ini

* Bind address = 0.0.0.0

Container

* Deploy the war in container.
* # Add mysql-connector.jar in /var/lib/tomcat7/lib.
* Run the application in browser IP(linux):port

POC 5:

1. Install tomcat(local machine) and deploy the application in that (windows).
2. Changed bind address in my.cnf file to 0.0.0.0 (etc/mysql/my.cnf)
3. Create mysql container in VM

* Apt-get install mysql-server
* Apt-get install mysql-client

1. Created database test.
2. Created table employee
3. Grant all privileges on database.
4. Grant all privileges on IP of win m/c also. (faced issues.)
5. Do port forward of mysql container.
6. Changed that ip of machine(forwarded port) in jdbc.properties file.
7. Run the application
8. It will store database entries in db of mysql container.

POC 6:

1. Used the above created tomcat container and mysql container.
2. Changed the ip in jdbc.properties file.
3. Mention the ip of mysql container.
4. Grant all the privileges to tomcat container in mysql container.
5. Run the application.

Container linking:

Start the previously created mysql container.

Forward its port to any random port.

Give the ip and forwarded port in previously created application container.

Start the app container by using –link.

Eg. Docker run –link gloomy\_wilson (mysql container name) tomcat\_mysql (application image name) /bin/bash

Docker build –no-cache –t image\_name