**MULTI-CONTAINER ARCHITECTURE**

**We can create multi-container architecture in 2 ways**

1. --link

2. docker compose

Create two busy box containers both the container will be communicate each other

**Busy box:**

Busybox is a small tiny unix operating system

Start two busybox containers and create link between them

**Create 1st busy box container**

**# docker run --name c10 -it busybox**

**/ #**

**Note:**

Control will enter into busybox container and will see /# in console

We can come out from container 2 ways

1. Using exit and ctrl+c

Container will be stopped and control will come out to docker host

2. Using **(ctrl + p + q)**

Container will not be stop and container will be running background

**How to come out of the container without exit**

**( ctrl + p + q)**

**Create 2nd busy box container and establish link to c10 container**

**#docker run --name c20 --link c10:c10-alies -it busybox (c10-alias is alias name)**

**/ #**

**How to check link is established for not?**

/ # ping c1

Ctrl +c ( to come out from ping )

( ctrl + p + q)

**Note:**

**No concep of switching one container to another container**

**Creating development environment using docker**

**Start mysql as container and link it with wordpress container.**

**Developer should be able to create wordpress website**

**1) TO start mysql as container**

**# docker run --name mysqldb1 -d -e MYSQL\_ROOT\_PASSWORD=root mysql:5**

**( if container is already in use , remove it**

**# docker rm -f mydb )**

**Check whether the container is running or not**

**# docker container ls**

**2) TO start wordpress container**

**# docker run --name mysite -d -p 5050:80 --link mydb:mysql wordpress**

**Check wordpress installed or not**

**Open browser**

**public\_ip:5050**

**18.138.58.3:5050**

**CRETE LAMP ARCHITECTURE USING DOCKER:**

**L -- linux**

**A -- apache tomcat**

**M -- mysql**

**P -- php**

**( Linux os we already have )**

**Beauty of LAMP is all are open source technology**

**here, we can see one to many container linking**

**Lets remove all the docker containers**

**# docker rm -f $(docker ps -aq)**

**# docker container ls ( we have no containers now )**

**1) TO start mysql as container**

**# docker run --name mydb -d -e MYSQL\_ROOT\_PASSWORD=sunil mysql:5**

**2) TO start tomcat as container**

**# docker run --name apache -d -p 6060:8080 --link mydb:mysql tomcat**

**TO see the list of containers**

**# docker container ls**

**To check if tomcat is linked with mysql**

**# docker inspect apache ( apache is the name of the container )**

**3) TO start php as container**

**# docker run --name php -d --link apache:tomcat --link mydb:mysql php**

**Create CI-CD environment, where jenkins container is linked with two tomcat containers.**

**Lets delete all the container**

**# docker rm -f $(docker ps -aq)**

**To start jenkins as a container**

**# docker run --name devserver -d -p 7070:8080 jenkins/jenkins**

**Can we remove the link?**

**to check jenkins is running or not?**

**Open browser**

**public\_ip:7070**

**http://18.138.58.3:7070**

**We need two tomcat containers ( qa server and prod server )**

**# docker run --name qaserver -d -p 8080:8080 --link devserver:jenkins tomee**

**To check the tomcat use public\_ip but port number will be 8080**

**http://18.138.58.3:8080**

**# docker run --name prodserver -d -p 9090:8080 --link devserver:jenkins tomcat**

**to check the tomcat of prodserver**

**http://18.138.58.3:9090**

**Creating testing environment using docker**

**Create selenium hub container, and link it with two node containers.**

**One node with firefox installed, another node with chrome installed.**

**Tester should be able to run selenuim automation programs for testing the application on multiple browsers.**

**To delete all the running containers**

**#**

**In Browser -- open - hub.docker.com**

**Search for selenium**

**We have a image - selenium/hub**

**To start selenium/hub as container**

**# docker run --name hub -d -p 4444:4444 selenium/hub**

**In hub.docker.com**

**we also have- selenium/node-chrome-debug ( It is ubuntu container with chrome)**

**To start it as a container and link to hub ( previous container)**

**# docker run --name chrome -d -p 5901:5900 --link hub:selenium selenium/node-chrome-debug**

**In hub.docker.com**

**we also have- selenium/node-firefox-debug**

**To start it as a container and link to hub ( It is ubuntu container with firefox)**

**# docker run --name firefox -d -p 5902:5900 --link hub:selenium selenium/node-firefox-debug**

**To see the list of container**

**# docker container ls**

**Note: firefox and chrome containers are GUI containers.**

**To see the GUI interface to chrome / Firefox container**

**Download and install vnc viewer**

**In VNC viewer search bar**

**public\_ip\_dockerhost:5901**

**18.136.211.65:5901**

**Password - secret**