Docker Notes

Listing images

```
docker images
```

Inspect Docker Image to find more details about the image

```
docker image inspect ubuntu:16.04
```

Listing currently running containers

```
docker ps
```

Listing all containers including the ones that exited

```
docker ps -a
```

To download a docker image from docker hub

```
docker pull hello-world:latest
```

To create a docker container in foreground mode(interactive)

```
docker run hello-world
docker run -it --name ubuntu1 --hostname ubuntu1 ubuntu:16.04 /bin/bash
```

In the above command

-it stands for interactive terminal
ubuntu1 - docker container name
ubuntu1 - represents hostname
ubuntu:16.04 represents image name with version 16.04
/bin/bash - blocking application that will be launched inside container

Stopping a running container

```
docker stop ubuntu1
```

Starting a exited container

```
docker start ubuntu1
```

Opening a second shell inside a running container

```
docker exec -it ubuntu1 /bin/bash
```

Finding a IP address of a running container

```
docker inspect ubuntu1 | grep IPA
```

Finding IP address of a container from within container shell

```
hostname -i
```

In order to provide internet access to your containers, make sure the below conf is done on CentOS Lab machine

Edit /etc/sysctl.conf and add the below line

```
net.bridge.bridge.nf-call-iptables=1
```

Make sure the below services are restarted

```
systemctl daemon-reload
systemctl restart network
systemctl restart docker
```

```
docker start ubuntu1
```

Get inside the ubuntu1 container using below command

```
docker exec -it ubuntu1 bash
```

```
apt update && apt install -y vim
```

Creating MYSQL Docker container

```
docker run --name mysql-server  --hostname mysql-server -e
MYSQL_ROOT_PASSWORD=root -d mysql:5.6
```

Get inside the mysql-server container with the below command

```
docker exec -it mysql-server  /bin/bash
```

```
mysql -u root -p
```

You need to type root as the password to login to mysql prompt

```
mysql >
```

In the mysql prompt, you may type the below command to display all the existing databases

```
SHOW DATABASES;
```

In case you would like to create a database

```
mysql > CREATE DATABASE tektutor;
```

Before you can create a table, you need to select a database first,

```
mysql > USE tektutor;
```

You may now create a table inside tektutor database as shown below

```
mysql > CREATE TABLE Training(id integer NOT NULL UNIQUE, name varchar(25), duration varchar(10));
```

You may now insert a record as shown below

```
INSERT INTO Training VALUES ( 1, 'DevOps', '5 days' );
```

You may now see the records in the table as shown below

```
SELECT * FROM Training;
```

Volume Mounting

```
docker run --name mysql-server --hostname mysql-server -e MYSQL_ROOT_PASSWORD=root -d -v /home/jegan/tmp:/var/lib/mysql  mysql:5.6
```

Setting up Nginx as a Load Balancer

You need to create 3 nginx web server as shown below

```
docker run -d --name nginx1 --hostname nginx1 nginx:1.16

docker run -d --name nginx2 --hostname nginx2 nginx:1.16

docker run -d --name nginx3 --hostname nginx3 nginx:1.16
```

You need to create a nginx load balancer container as shown below

```
docker run -d --name lb --hostname lb -p 80:80 nginx:1.16
```

In order to configure the lb container to work as a load balancer

we need to first copy the nginx.conf from the container to the local machine

```
docker cp lb:/etc/nginx/nginx.conf  .
```

You need to edit nginx.conf on the centos lab machine with any text editor vim nginx.conf and make sure the file looks as below

```
user  nginx;
worker_processes  1;
error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;
events {
    worker_connections  1024;
}
http  {
    upstream backend {
        server 172.17.0.2:80;
        server 172.17.0.3:80;
        server 172.17.0.4:80;
    }
    server {
        location / {
            proxy_pass http://backend;
        }
    }
}
```

In the above file,

172.17.0.2 is the ip address of nginx1 container

172.17.0.3 is the ip address of nginx2 container

172.17.0.4 is the ip address of nginx3 container

You may need to replace the ip addresses of your containers.

In order to apply the configuration changes in the load balancer container

```
docker restart lb
```

Make sure the lb container is actually running after the config changes

```
docker ps
```

Once you have made sure the lb container is running, then you may find the IP Address of you CentOS Lab machine as shown below

```
ifconfig ens33
```

You may customize the web pages on nginx1, nginx2 and nginx3 respectively as shown below

```
echo “Server 1” > index.html
```

```
docker cp index.html nginx1:/usr/share/nginx/html/index.html
```

```
echo “Server 2” > index.html
```

```
docker cp index.html nginx2:/usr/share/nginx/html/index.html
```

```
echo “Server 3” > index.html
```

```
docker cp index.html nginx3:/usr/share/nginx/html/index.html
```

In my system IP Address of my CentOS machine happens to be 172.16.124.251

From the Alchemy Windows Cloud machine, open the browser with the URL as shown below

http://172.16.124.251

Each time you refresh the web page on the above URL, you may see the output as

Server 1

Server 2

Server 3 in a round robbin fashion.