link:

https://vagrantup.com

```
config.vm.network "forwared_port",guest: 80, host:8080
```

```
docker search
docker pull
docker run //
docker create
docker start //
docker attach
docker detach // shell run on background
docker ps -a //verbo string id
docker ps -a -q // list id of container
docker ps -a -q -l // last run id
docker ps
docker exec container-id echo "HELLO" // to execute commd in container
docker du // disk use it will show how much space allocated for directoies
docker container-id/name hostname
docker log test
```

-- can start container with container_id,name
-- docker run -it --name myubuntu /bin/bash

busybox // compressed ubuntu use for embadded system

Set environment
```
docker run -e FOO=bar busybox env
```

Docker Volumes
mount directories
pvcscan

```
docker run -v /dbdata --name dbstore2 ubuntu /bin/bash
docker volume ls
docker volume ls -a
docker run -it -v /john1 busybox
```

```
docker inspect container-id
```
hostpath -- to check volumes what you created

to share the data b/w the containers we use volumes

Create Volume:
Task:

Create volume
attach to diff containers

```
docker volume create --name myvalume
docker  run -d --name test -v my-vol:/data ubuntu  // mapping volume inside data dir in container
docker run -it -v /vagrant/john3:/john3 // what ever in john3 in the container will availble in
/vagrant/john3
```

share b/w container

```
docker run --name ctr1 -it -v /mydata ubuntu /bin/bash
```

create 2 container

```
docker run --name ctr1 -it -v /mydata ubuntu /bin/bash
docker run --name ctr2 -it -v /mydata ubuntu /bin/bash

docker run -it --name john2 --v
```

commit on changes:

Task:

Run an image of ubuntu
Install apache on it
Port forwarding
Stop the container
Restart it
You apache should be there

```
config.vm.define "server2" do |server2|
    server2.vm.box = "bento/centos-7.2"
    server2.vm.network "private_network", ip: "192.168.50.71"
    server2.vm.network "forwared_port",guest: 80, host:8080
  End
```

Docker File:

Sudo su
Step 2: create a file
Vi dockerfile
Content:
FROM ubuntu:14.04
MAINT
Step 3: run the build command

Docker build -t newubuntu:14.04 .
Docker build alway expect dockerfile
Docker images
Docker build .
Docker build . myimages
Docker build . -t myimages:latest
Docker build -f myimages

Nginx from dockerfile

```
FROM ubuntu
RUN apt-get update
RUN apt-get install -y nginx
ENTRYPOINT [“/usr/sbin/nginx”,”-g”,”daemon off;”]
EXPOSE 80
```

Run cmd:
Docker build -f dockerfile-nginx -t test:1.0 .
Docker run -d -p 80:80 --name webserver nginx-ubuntu
You can also run as
Docker run -d -P --name webserver nginx-ubuntu
Curl localhost:<portid>

Task :
Create 2 volumes attach to ubuntu container and add some files to volumes

Step 1:
```
docker volume create --name volume-1
docker volume create --name volume-2
```

Step 2:
attach this volumes to container
```
docker run -d -it -v volume-1:/volume-data-1 -v volume-2:/volume-data-2 ubuntu /bin/bash
```

attach container
```
docker attach 3bcfb18
```

```
root@3bcfb18c1c9c:/# ls
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var        volume-data-2
boot  etc  lib  media  opt  root  sbin  sys  usr  volume-data-1
```

Step 3:

Just create 2 files inside the volume and those files should in hosted directory called test
```
  mkdir test
  docker run -v /tmp/test:/test -it -d ubuntu
  docker attach container id
  cd test
  touch one.txt
  touch two.txt
  exit
  cd /tmp/test
  cs
```

Practice :Nginx with docker volume

Step 1:
```
docker run --name nginx-container -p -d nginx
docker run --name nginx-container -P -d nginx
```
-p -
-P - grab any open port dynamically
Nginx by default on 80 port
Reverse proxy : get the request with one port and redirect to another port
```
Curl localhost:32768
Ifup enp0s3
```

```
[root@tomcat /]# docker network ls

NETWORK ID          NAME          DRIVER                SCOPE
7ad2f83c9b47        bridge        bridge        local
b4be2e2da55e        host          host          local
515a0170027f        none          null          local
```

Network start with
172.17.0.6/16

By default all container in the bridge n/w

Add Proxy Details in /etc/apt/apt.conf

```
acquire::http::proxy "http://Guduru.Reddy:<password>@btpproxy.mphasis.com:8080";
acquire::https::proxy "https://Guduru.Reddy:<password>@btpproxy.mphasis.com:8080";
```

```
sudo apt-key adv --keyserver-options
http-proxy=http://Guduru.Reddy:sras%402017@btpproxy.mphasis.com:8080. --keyserver
hkp://p80.pool.sks-keyservers.net:80 --recv-keys
58118E89F3A912897C070ADBF76221572C52609D
```

then sudo apt-key adv --keyserver-options
http-proxy=http://Guduru.Reddy:sras%402017@btpproxy.mphasis.com:8080. --keyserver
hkp://p80.pool.sks-keyservers.net:80 --recv-keys
58118E89F3A912897C070ADBF76221572C52609D
then create - /etc/apt/sources.list.d/docker.list and add deb https://apt.dockerproject.org/repo
ubuntu-trusty main
sudo apt-get install docker-engine
sudo service docker start
Run below command to enable the proxy to pull images(Update passwords)-----

```
cat <<EOF | sudo tee -a /etc/default/docker
export http_proxy="http://jayashree.h:<pwd>@gtpproxy.mphasis.com:8080" - ! is replaced with
%21
export https_proxy="https://jayashree.h:<password>@gtpproxy.mphasis.com:8080"
export no_proxy=<REGISTRY_IP>
EOF
```

sudo restart docker