**<https://talentnext.mphasism4l.cloud/#>!**

**Day 2**

**Chapter 1: Installation and configuration of docker on CentOS7 vm**

http://docs.docker.com/installation

# sudo yum install -y yum-utils \

device-mapper-persistent-data \

lvm2

# sudo vi /etc/yum.conf

sslverify=false

# yum-config-manager \

--add-repo \

https://download.docker.com/linux/centos/docker-ce.repo

# yum-config-manager --enable docker-ce-nightly

# yum-config-manager --enable docker-ce-test

# sudo yum install docker-ce docker-ce-cli containerd.io

# yum list docker-ce --showduplicates | sort -r

# systemctl enable docker

# systemctl start docker

# docker version

#docker info

**Chapter 2: Installation and configuration of docker on Ubuntu vm**

Tast 1 installing old version:

## For Debian & Ubuntu systems

#apt update or #apt-get update

#apt install docker.io

#docker info

Tast 2 Install vs Removing old version: io / docker-engine

$ uname -r

$ sudo apt-get remove docker docker-engine docker.io

$ sudo apt-get update

$ sudo apt-get install docker-engine

Task 3 Install and remove Docker community Edition (Docker CE)

$ apt-get remove docker docker-engine docker.io containerd runc

$ apt-get update

$ sudo apt-get install \

apt-transport-https \

ca-certificates \

curl \

gnupg-agent \

software-properties-common

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

$ apt-key fingerprint 0EBFCD88

$ add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) \

stable"

$ apt-get update

$ apt-get install docker-ce docker-ce-cli containerd.io

$ apt-cache madison docker-ce

$ apt-get install docker-ce=<VERSION\_STRING> docker-ce-cli=<VERSION\_STRING> containerd.io

$ docker run hello-world

Task 5 Removing Community Edition

$ apt-get purge docker-ce

$ rm -rf /var/lib/docker

**Working with Windows container**

docker pull windows/nanoserver

docker pull nanoserver/iis

docker images

docker run -it windows/nanoserver cmd.exe

dir

echo "hello world" > hello.txt

powershell

cat hello.txt

type hello.txt

ls

exit

docker ps -a

docker logs eec

docker commit eec helloworld

docker images

docker run --rm helloworld cmd.exe

docker run --rm helloworld cmd.exe /s /c type Hello.txt

**Eg.1**

# vi Dockerfile

FROM busybox:latest

MAINTAINER Chinnajee Rao (Chinnajee.rao@mphasis.com)

:wq

#docker build -t myimage:latest .

# docker images

# docker run -it myimage

# docker rmi myimage

**Eg.2**

# vi Dockerfile

FROM busybox:latest

MAINTAINER Chinnajee Rao (Chinnajee.rao@mphasis.com)

CMD ["date"]

:wq

#docker build -t myimage:latest .

# docker run -it myimage

**Chapter 12: SHIP**

**Option 1**

docker ps

docker images

docker pull ubuntu

docker images

docker pull alphine:3.4

docker tag alpine:3.4 chinnajee/myalpine:3.4

docker images

docker login

docker push chinnajee/myalpine:3.4

docker ps

docker ps -a

docker images

docker search chinnajee/myalpine:3.4

docker pull chinnajee/myalpine:3.4

docker rmi chinnajee/myalpine

docker images

docker rmi chinnajee/myalpine:3.4

docker images

docker pull chinnajee/myalpine:3.4

docker images

docker pull chinnajee/myalpine:3.4

docker tag ubuntu:latest chinnajee/ubuntu:1.0

docker push chinnajee/ubuntu:1.0

docker push chinnajee/ubuntu:2.0

docker tag ubuntu:latest chinnajee/ubuntu:2.0

docker push chinnajee/ubuntu:2.0

docker tag ubuntu:latest chinnajee/ubuntu

docker push chinnajee/ubuntu

docker images

docker tag busybox:latest chinnajee/mybusy:1.0

docker push chinnajee/mybusy:1.0

docker login

docker push chinnajee/mybusy:1.0

docker push chinnajee/ubuntu

docker push chinnajee/ubuntu1.0

docker images

docker push chinnajee/ubuntu:1.0

docker push chinnajee/ubuntu:2.0

docker run hello-world

docker imaegs

docker images

docker tag hello-world chinnajee/hello-world:1.0

docker push chinnajee/hello-world:1.0

docker ps

**Option 2**

docker pull ubuntu

docker run -it -d ubuntu

docker exec -it ab2189 bash

apt-get update

apt-get install nginx

apt-get install elinks

elinks <http://hostname>

exit

docker commit ab2189 chinnajee/nginx1

docker push Chinnajee/nginx1

on other machine

docker pull Chinnajee/nginx1

docker run -it -d Chinnajee/nginx1

open browser

<http://hostname>

**Chapter 13: RUN**

$ docker run --rm -it --name web -p 5000:80 -v ~/dev:/code alpine:3.4 /bin/sh

$ docker stop web

$ docker kill web

$ docker ps

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

$ docker exec -it web bash

$ docker logs --tail 100 web

**Chapter 6: Name container and link two containers**

$ sudo docker run –name mycont1 -d afakharany/server

$sudo docker ps

$sudo docker run -it –name checker afakharany/checker // return error bse it is not linked with server

Container

$sudo docker rm checker

$sudo docker run -it --link mycont1:server --name checker afakharany/checker

$ sudo docker run -it --link mycont1:server --name checker1 afakharany/checker

$ sudo docker run -d --link mycont1:server --name checker2 afakharany/checker //needs to delete the 2

Pervious containers to use the

Same name

$sudo docker ps

**Chapter 7: Create a service monitor**

**Part 1**

$ sudo docker stop mycont1

$sudo docker ps

$ sudo docker start mycont1

$sudo docker start checker2

$ sudo docker logs checker2

**Part 2**

$ sudo docker stop mycont1

$sudo docker ps

$ sudo docker start mycont1

$sudo docker start checker2

$ sudo docker logs checker2

$ sudo docker logs -f checker2 //this command acts like tail -f command

$ sudo docker stop mycont1

$ sudo docker logs -f checker2 // now it acts like without -f bse containers are stoped

$ sudo docker restart mycont1

$ sudo docker logs -f checker2