Docker

Sunday, March 14, 2021

8:43 PM

1. Docker HUB on Google
2. Download Postgres from docker hub page and install on docker, install different version

* $ Docker ps
* $ docker run postgres:9.6
* $ docker run postgres:10.10

**Install Dockers**

Remove dockers

$ sudo yum remove docker \  
 docker-client \  
 docker-client-latest \  
 docker-common \  
 docker-latest \  
 docker-latest-logrotate \  
 docker-logrotate \  
 docker-engine

From <[*https://docs.docker.com/engine/install/centos/*](https://docs.docker.com/engine/install/centos/)>

**Set up the repository**

$ sudo yum install -y yum-utils

$ sudo yum-config-manager \  
 --add-repo \  
 <https://download.docker.com/linux/centos/docker-ce.repo>

From <[*https://docs.docker.com/engine/install/centos/*](https://docs.docker.com/engine/install/centos/)>

**Install Docker Engine**

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

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

doc ker 
doc ker 
docker 
docker 
dccke--ce 
-shc"'du 
3:18.ag.1-3.e17 
-ce.x86 
3:18.ag.a-3.e17 
-ce.x86 
-ce. x86 
-ce .06 
64 
64 
64 
64 
18.06.1 
18.as.a 
. ce-3.e17 
. ce-3.e17 
docker-ce-stable 
docker-ce-stable 
docker-ce-stable 
docker-ce-stable 

$ sudo yum install docker-ce-<VERSION\_STRING> docker-ce-cli-<VERSION\_STRING> containerd.io

**Or you can**

**Force the installation of docker-ce with the --nobest option**

$ sudo dnf install --nobest docker-ce

**Start Docker.**

$ sudo systemctl start docker

**Verify that Docker Engine is installed correctly by running the hello-world image.**

$ sudo docker run hello-world

**If you would like to use Docker as a non-root user, you should now consider adding your user to the “docker” group with something like:**

$ sudo usermod -aG docker <your-user>

**Uninstall Docker Engine**

$ sudo yum remove docker-ce docker-ce-cli containerd.io

**Images, containers, volumes, or customized configuration files on your host are not automatically removed. To delete all images, containers, and volumes:**

$ sudo rm -rf /var/lib/docker

$ sudo rm -rf /var/lib/containerd

**Install Redis Application**

**$ docker pull redis**

**$ docker images ( will display images running on container)**

**$ docker run redis**

**$ docker ps ( list running containers )**

**Ctrl + C (Exit docker)**

**$ docker run -d redis ( start a new container with a command )**

**$ docker ps**

**$ docker stop 838186(container ID) ( stop the container )**

**$ docker ps**

**$ docker start 838186(container ID) ( start the container )**

**$ docker ps**

**$ docker ps -a ( lists all the container which are running and stopped containers)**

**$ docker run redis:4.0 ( pulls image and start container )**

**$ docker ps**

**$ docker run -p6000:6379 ( -p bind the port of your host to the container)**

**$ docker run -p6000:6379 redis**

**$ docker ps**

**$ docker run -p6000:6379 -d redis**

**$ docker ps**

**$ docker run -p6001:6379 -d redis:4.0 ( run in detach mode, with port happed to 6001)**

**$ docker ps**

**How to Run a container will a name change**

**$ docker ps**

**$ docker stop cfec85d7**

**$ docker run -d -p6001:6379 --name redis-older redis:4.0**

**$ docker ps**

**(redis:4.0 image name will be changed to redis-older)**

**$ docker run -d -p6000:6379 --name redis-latest redis**

**$ docker ps**

**(redis image name will be changed to redis-latest)**

**If older version has some issue I can do**

**$ docker logs redis-older**

**To login to the terminal of a container and get log or troubleshoot, we can do**

**$ docker exec -it Ce9032 /bin/bash**

**root@Ce9032 :/data# ls**

**root@Ce9032 :/data# pwd**

**/data**

**root@Ce9032 :/data# cd /**

**root@Ce9032 :/data# ls**

**Bin boot data**

**root@Ce9032 :/# env**

**(displays environment variable)**

**root@Ce9032 :/data# curl**

**( you will have limited number of commands which works under container)**

Ngnix install

docker container create nginx

docker container ls -a

docker container inspect web01 | grep -e "HostPort" -e "IPAddress"

curl 172.17.0.4

docker container run -d --name web02 -dit -p 8080:80 nginx

docker container ls -sa

**Project**

<https://gitlab.com/nanuchi/techworld-js-docker-demo-app>

**Download Mongo DB**

$ docker pull mongo

$ docker pull mongo-express

$ docket images ( to check existing images)

**Creating Docker Network**

$ docker network ls

**$ docker network create mongo-network**

$ docker network ls ( you will see mongo-network

**Mongo DB install**

#commands

**##create docker network**

Docker network create mongo-network

**## start mongodb**

docker run -d \

-p 27017:27017 \

-e MONGO\_INITDB\_ROOT\_USERNAME=admin \

-e MONGO\_INITDB\_ROOT\_PASSWORD=password \

--net mongo-network \

--name mongodb \

mongo

**## start mongo-express**

docker run -d \

-p 8081:8081 \

-e ME\_CONFIG\_MONGODB\_ADMINUSERNAME=admin \

-e ME\_CONFIG\_MONGODB\_ADMINPASSWORD=password \

-e ME\_CONFIG\_MONGODB\_SERVER=mongodb \

--net mongo-network \

--name mongo-express \

mongo-express

16 
18 
19 
20 
21 
23 
create docker netvork 
docker create mnqo-netvork 
1 
start nonqodb 
docker run —d 
-p 27e17:27e17 
—net mnqo-netvork \ 
nongodb 
tart EngO—e.preSS 
docker run —d 
-e "_ccm 
—net 
nongo—expresS 
now-express 

**Docker compose**

$ docker container kill web02 (kill all the application before launching dockers)

$ docker-compose -f mongo.yaml up ( to bring the containers up)

$ docker-compose -f mongo.yaml down ( to bring the containers down)

**Docker build an image**

$ docker build -t my-app:1.0 . (. Is for local install)

$ docker Images

**Remove the container**

$ docker rm 3c5868bi (container name)

$ docker rmi 2e08bi (image deletion)

$ docker ps

$ docker logs 34341FNADLFN (CONTAINER id)

**CONNECT TO A CONTAINER**

$ docker exec -it 542524klksdfjad(container ID) /bin/sh

/# ls

/# env (mongo db username and password are set )

/# ls /home/app/