Docker

- Docker Volume
- Docker backup & restore
    - Reference link - https://bobcares.com/blog/docker-backup/2/
- To push container as a image on docker-hub
- Docker Commands.....

Docker Volume

Data volumes are directories that can store data outside Docker containers. This data can be shared among containers and is persistent.

The data volumes are usually stored in the folder '/var/lib/docker/volumes' and can be listed using the command 'docker volume ls'.

Docker image ?

A Docker image is a file, comprised of multiple layers, that is used to execute code in a Docker container.

What is Docker ? – Docker is a containerization platform that packages your application and all its dependencies together in the form of a docker container to ensure that your application works seamlessly in any environment.

What is Container ? – Docker Container is a standardized unit which can be created on the fly to deploy a particular application or environment. It could be an Ubuntu container, CentOs container, etc. to full-fill the requirement from an operating system point of view. Also, it could be an application oriented container like CakePHP container or a Tomcat-Ubuntu container etc.

To create volume

docker volume create --name DataVolume1

To run container using create voluem

docker run -ti --rm -v DataVolume1(host volume):/datavolume1(directory_name into docker) ubuntu(name of image)

To inspect volume

docker volume inspect DataVolume1

To create volume with container name

docker run -ti --name=Container2 -v DataVolume2:/datavolume2 ubuntu

To create container from cotainer2 volume

docker run -ti --name=Container3 --volumes-from Container2 ubuntu

Docker backup & restore

Reference link - https://bobcares.com/blog/docker-backup/2/

How to restore Docker containers

docker load -i backup.tar

backup.tar name of backup container file name

To backup Docker data volumes

first check docker volume ls

docker inspect -f '{{ .Mounts }}' 2bcfbeb21853

2bcfbeb21853 - container id

Output ->

[{volume 90c1ef308199093e9b9e5806b53abe15c73dee416951cd9b41445a10fd828965 /var/lib/docker/volumes/90c1ef308199093e9b9e5806b53abe15c73dee416951cd9b41445a10fd828965/_data /data/configdb local true } {volume 2f779c0db084b3cbedfa3a97e15e7bf703a9445afe30d31bdd25b3153064f631 /var/lib/docker/volumes/2f779c0db084b3cbedfa3a97e15e7bf703a9445afe30d31bdd25b3153064f631/_data /data/db local true }]

docker run --rm --volumes-from mongodb -v $(pwd):/backup e43a2492d00f tar cvf /backup/mongodb.tar /data/db

How to restore Docker data volumes

To restore a container using the backup of data volumes taken, first create a new container by providing data volume and container names:

docker run -p 27017:27017 --name mongodb e43a2492d00f

e43a2492d00f = image-id

docker run -p 27017:27017 --rm --volumes-from mongodb -v $(pwd):/backup e43a2492d00f tar xvf /backup/mongodb.tar

docker restart mongodb

mongodb = name of stop container

To push container as a image on docker-hub

To login on Docke-Hub

https://hub.docker.com/

docker login

1. Commit the required container as an image

docker commit -p [container-id] backup01

backup01 = it can be any name

1

sudo docker commit [CONTAINER_ID] [new_image_name]

2. You can save the image backup01 to tar file using the following command:

docker save -o backup01.tar backup01

To check backup .tar file

ls -al | grep back

output:

-rw------- 1 root root 178697728 Mar 31 23:35 backup01.tar

3. To add tag to image for push on Docker hub

docker tag backup01 localhost:5000/backup-image:v1

OR

docker tag mariadbforbootstack siyatechventures53/mariadbforbookstack-img:stv

4. To push image on docker-hub

docker push backup-image:v1

OR

docker push siyatechventures53/mariadbforbookstack-img:stv

Restoring a Docker Container

docker load -i /tmp/backup01.tar

ff91b8b5abb1: Loading layer [==>] 2.56 kB/2.56 kB

Loaded image: backup01:latest

Docker Commands.....

Stop all container

docker container stop $(docker container ls -aq)

Remove all container

docker container rm $(docker container ls -aq)

Remove all images

docker image prune -a

Purging All Unused or Dangling Images, Containers, Volumes, and Networks

docker system prune

To additionally remove any stopped containers and all unused images (not just dangling images), add the -a flag to the command:

docker system prune -a

To see docker images list

docker images -a

Removing Docker Images

docker rmi Image Image

Remove all images in one command

docker rmi $(docker images -a -q)

Remove a container upon exit

docker run --rm image_name

List of all exited container

docker ps -a -f status=exited

To all exited container

docker rm $(docker ps -a -f status=exited -q)

To stop all container

docker stop $(docker ps -a -q)

To remove unused all images

docker rmi $(docker images -f dangling=true)

Delete all volumes using the following command:

docker volume rm $(docker volume ls -q)