DOCKER VOLUMES

Volumes are the preferred mechanism for persisting data generated by and used by Docker containers.

For Example one of the container is deleted Accidentally we lost the total amount of data that is in Docker container. So to resolve this issue we can bind the docker volume to host system.

And there we have other useful thing with this docker volumes concept is that we can mount the new container volume to the existing volume if it is a replica type of the existing container.

==> To Create a new Volume use the following command.

“docker volume create volume-name”

==> To Create a volume from the images.

“docker run -itd -p 8080:8080 -v /home/srikanth/docker/tomlogs:/usr/local/tomcat/logs image-id”

==> We can verify that the volume is present on our system with

```
docker volume inspect Volume-name
```

= = > Docker won't let us remove a volume if it's referenced by a container. Let's see what happens when we try

```
docker volume rm volume-name
```

we will get the o/p as shown below

Output

```
Error response from daemon: unable to remove volume: remove DataVolume2:
volume is in use -
[d0d2233b668eddad4986313c7a4a1bc0d2edaf0c7e1c02a6a6256de27db17a63]
```

= = > With the help of that above id we can remove the container then we can delete that volume.

Sharing Data Between Multiple Containers

= = > First Create a new container with Name and Create new volume by running following command.

docker run -it --name=tom1 -v /home/srikanth/docker/tom1/:/usr/local/tomcat/logs/ -p 8045:8080 tomcat:8-jre8

and just touch a file over in this container like touch guptha.txt

echo “this file is shared between two containers”

Now Create a New Container and mount the volumes from the previously created container.

Then the newly created container have commonly shared volume.

docker run -it --name=tom2 --volumes-from e7f7330e07e4(previous-containerid) tomcat:8-jre8(image name)

== > Mount The Volume In Read-only Mode.

docker run -it --name=tom3 --vloumes-from e7f7330e07e4:ro (previously created containerid) tomcat:8-jre8 (image name).

= = > How to remove the volumes

docker volume rm volume-name

= => How to increase docker container volume size.

The minimum size of docker containers is 10 GB and its not possible to decrease it further. But you can increase the docker container size from 10 GB it to a higher value, say 20 GB, with these steps:

1. Stop the Docker daemon after taking backup of existing containers and images.

2. Reset the Docker default directory.

3. Start Docker service with the parameter ‘dm.basesize’ set for the new value for Docker container size limit.

dockerd --storage-opt dm.basesize=20G

DOCKER-COMPOSE

==> is a tool for defining and running multi-container Docker applications.

==> define the services that make up your app in `docker-compose.yml` so they can be run together in an isolated environment.

==> get an app running in one command by just running `docker-compose up`

Other Definition.

Docker Compose is the toolkit provided by Docker to build, ship and run multi-container applications.

Create a docker-compose file

==> for this docker-compose file we have to write it in yaml language so that the file extension should be like docker-compose.yml

= = > these yaml files can have extension with .yml or .yaml extension.

sudo vi docker-compose.yml

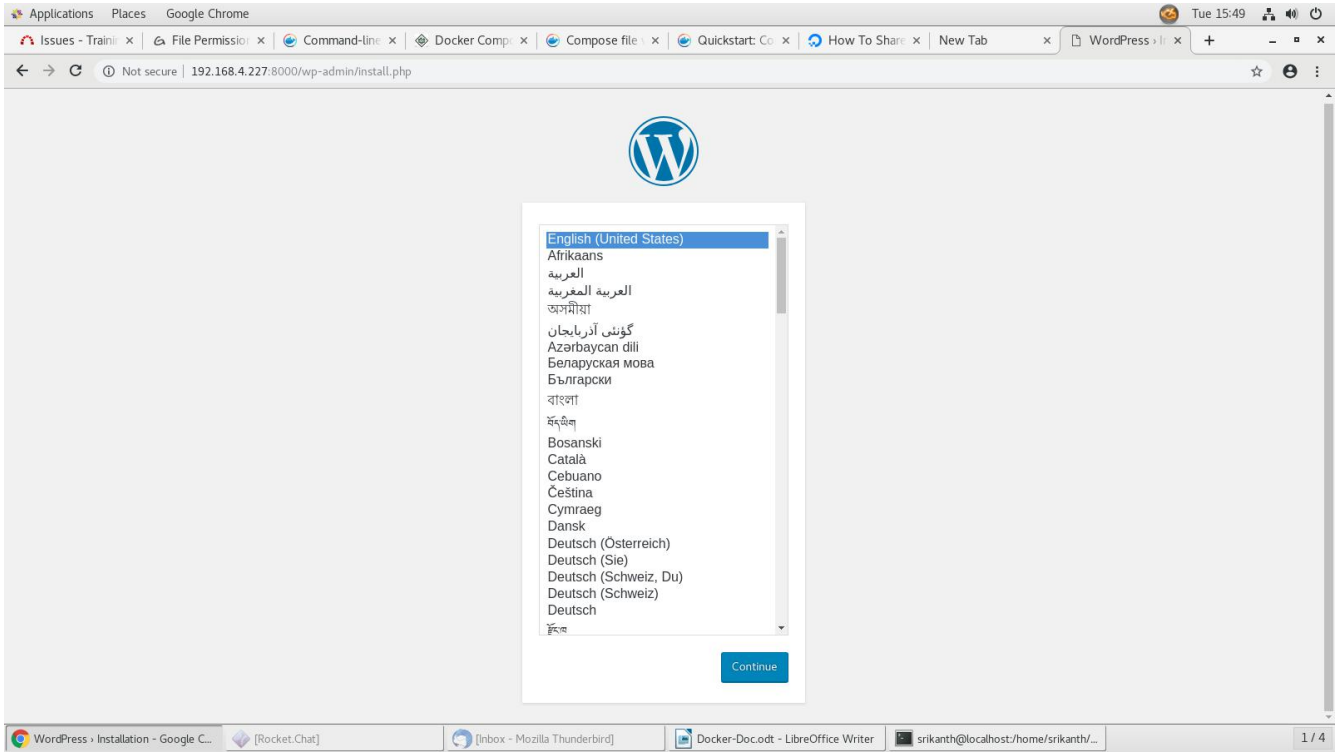
```
version: '3.3'

services:

   db:

     image: mysql:5.7

     volumes:

       - db_data:/var/lib/mysql

     restart: always

     environment:

       MYSQL_ROOT_PASSWORD: somewordpress

       MYSQL_DATABASE: wordpress

       MYSQL_USER: wordpress

       MYSQL_PASSWORD: wordpress


   wordpress:

     depends_on:

       - db

     image: wordpress:latest
```

```
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
volumes:
    db_data: {}
```

= = > Now Bring up the wordpress in browser .

You will get the wordpress page like below

http://192.168.4.227:8000/wp-admin/install.php



= = > To Down The Docker-compose use below command.

docker-compose down

Structure of Docker-compose file.

The compose file reference documentation is layed out following the structure of the Compose file. The main chapters in the docs are related to the top-level entries of the Compose file, these are the following as of today:

== >version - specifies the version of the Compose file reference, you have seen it in the example.

== >services - specifies the services in your application, we used it in the example.

== >networks - you can define the networking set-up of your application here.

==> volumes - you can define the volumes used by your applicaiton here, we'll see an example.

==>secrets - secrets are related to Swarm mode only, you can use them to provide secret information like passwords in a secure way to your application services.

==>configs - configs lets you add external configuration to your containers. Keeping configurations external will make your containers more generic. Config is available both in Compose and in Swarm mode.

Different Directives using for services in a docker compose file.

Image

This will sets the image used to build the container Using this directive assumes that the specified image already exists either on the host or on docker hub.

build

This directive can be used instead of `image`. Specifies the location of the Dockerfile that will be used to build this container.

db

In the case of the example Dockercompose file, `db`is a variable for the container you are about to define.

restart

Tells the container to restart if the system restarts.

Volumes

Mounts a linked path on the host machine that can be used by the container.

environment

Define environment variables to be passed in to the Docker run command.

depends_on

Sets a service as a dependency for the current block-defined container.

port

Maps a port from the container to the host in the following manner:`host:container`

links

Link this service to any other services in the Docker Compose file by specifying their names here.

container_name

directive is used to override the randomly generated container name and replace it with a name that is easier to remember and work with.

entrypoint

is overridden to keep the container running.

Other Example of docker-compose

```
version: '3'
services:
  nginx:
    image: nginx:latest
    container_name: production_nginx
    ports:
      - 89:80
    volumes:
      - /home/srikanth/composefiles/nginxlogs:/var/log/nginx/
  tomcat:
    image: tomcat:latest
    container_name: prod_tom
    ports:
      - "8051:8080"
      - "8052:8080"
      - "8053:8080"
      - "8054:8080"
```

```
      - "8055:8080"
    volumes:
      - /home/srikanth/composefiles/tomlogs:/usr/local/tomcat/logs/
      - /home/srikanth/docker/hello-
world.war:/usr/local/tomcat/webapps/hello-world.war
```

= = > we can run number of servers with a single service which we specified in docker compose file.

= = > You need to add the no.of ports in compose file to run no.of tomcats of a same service.

Different Commands Using with Docker-compose

```
docker-compose up -d
docker-compose down
docker-compose start
docker-compose stop
docker-compose build
docker-compose logs -f db
docker-compose scale db=4
docker-compose events
docker-compose exec db bash
```