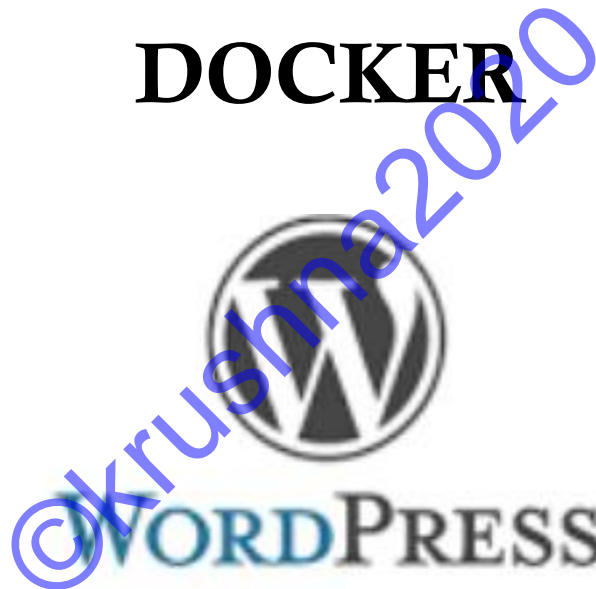


A PROJECT REPORT ON

**A MULTI-TIER
ARCHITECTURE FOR
LOCAL WORDPRESS
HOSTING USING
DOCKER**



ACKNOWLEDGEMENT

I am profoundly grateful to the World Record Holder , TEDx speaker , Philanthropist , Senior IT consultant , Entrepreneur, Founder of Linux World Informatics Pvt. Ltd. & IIEC RISE and Expert Corporate Trainer in almost all Latest & High-End Technologies Mr. VIMAL DAGA for his expert guidance and encouragement throughout to see that this project rights its target since its commencement to its completion.

I would like to express my gratitude again to such a huge personality Mr. VIMAL Sir who gave me this golden opportunity to do this wonderful project , which helped me in doing a lot of research and I learned so many new things.

I would like to extend my special thanks to all the volunteers and staffs of IIEC (Indian Innovation and Entrepreneurship Community) as well as Linux World for giving me such a chance to join them. This project would not have completed without their enormous help and worthy experience. Whenever I was in need, they were there behind me .

Although this report has been prepared with utmost care and deep routed interest. Even then I would like to accept respondents and imperfections.

Thank You

KRUSHNA PRASAD SAHOO
RISE 2020.11.50.3
IIEC RISE 1.0

CERTIFICATION

IIEC-RISE

An Initiative for empowering engineering youth of India

This certificate is hereby bestowed upon

Krushna Prasad Sahoo

for successfully completing the training on

EXPERTISE IN DOCKER

a 24 hour training that covers Orchestration of Multi Container Application Using Docker Compose under the Mentorship of Mr. Vimal Daga

Certificate No: IIEC-RISE-2020- 1653
Date: 9th May, 2020

Authorised Signatory
Indian Innovation & Entrepreneurship
Community (IIEC)



**CERTIFICATE
OF COMPLETION**

CONTENTS

Introduction.....	02
Objective.....	02
Project Ideation & Technical Aspects.....	02
Project Implementation.....	03
Results & Discussions.....	17
Future Scope.....	18
Conclusion.....	18

©krushna2020

■ INTRODUCTION

In today's digital world, the place where most people in most markets go is to the internet, specifically a search engine like Google, Yahoo, Firefox. With 93% of business decisions starting with a search engine search that means if you don't have a website, you are only selling to 7% of your market. Having your website will give you a chance to introduce people to know of your products and services and a way to find you. Not only it gives you online identity but also a platform to tell them who you are and why you do it. One of its way is blogging .

Blogging is an easy way to express yourself. It's is an amazing way to drive traffic to your website and increase your SEO. A blog with useful content shows your audience and customers that you are a trusted source.

■ OBJECTIVE

My aim in this project is to host a WordPress (a blogging tool) site in my local machine for learning purpose. Moreover this hosting will be done using the most demanding containerization technology(Docker) for more reliability , flexibility & also for safety and security .

The purpose of this document is to provide information about the technology and technical aspects of a real industry use case. It covers the technology used for this project , its planning and implementation process , results & its future scope .

■ PROJECT IDEATION & TECHNICAL ASPECTS

In this project I'm going create a multi-tier architecture for hosting my blog using the WordPress framework on Docker container . For testing & leaning purpose it will be hosted in my local machine . I will be also able to access my site from the outside network .

WordPress is a open-source content management system (CMS) written in PHP and paired with database. To function, WordPress has to

be installed on a web server . For this I can use Apache web server . I am using Windows 10 as bare-metal setup on which I have a virtual machine of Red Hat Enterprise Linux 8 with the help of Oracle Virtual Box . I'll perform all implementation part of this project on RHEL8 only .

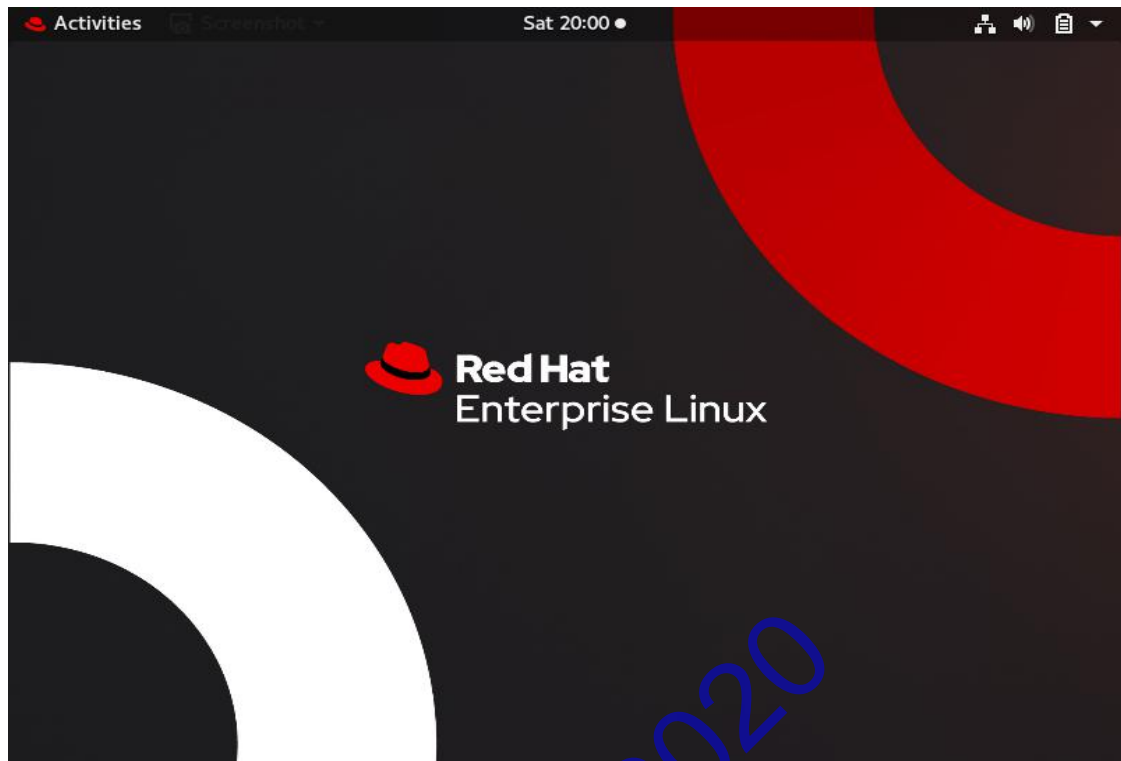
Wordpress will provide a dynamic website so I need to store the data what I will get from user while interacting with my website. Hence a dedicated database server is needed . Here I'm using MySQL server for this and to access & manage data I'm using MySQL client software .

For running any services or programs some RAM , CPU & storage are required or typically an well fitted environment is needed . To achieve greater performance & for safety ,security I would like to use two dedicated servers i.e one as web server another one database server. For this I may use virtualization technology where two dedicated virtual machines will work for me as servers . However running a single service won't consume much resources like RAM in GBs or higher CPUs . They need only some RAM in MBs. Hence not to waste my resources I am using Docker tool which is based on containerization technology in this project implementation . Some more such tools are Buildah , Podman etc.

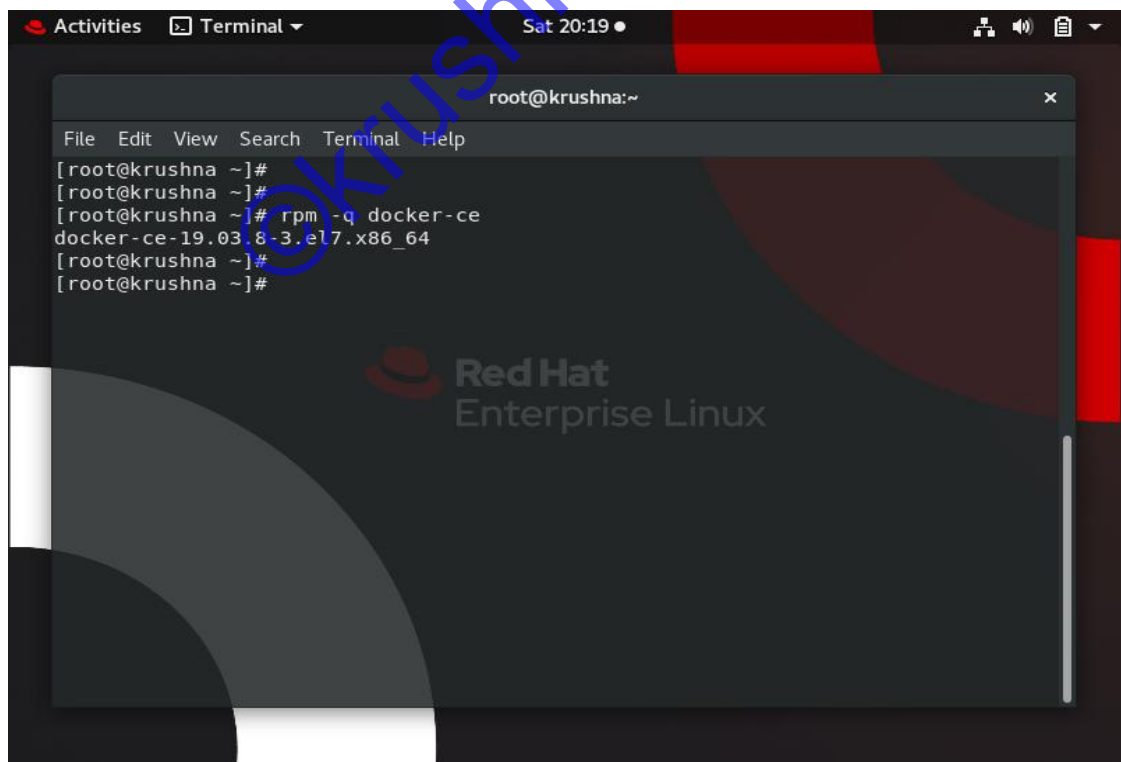
Docker is designed to make is easier to create , deploy and run applications using containers . Containers allow a developer to package up an application with all parts it needs, such as libraries and other dependencies, and deploy it as one package . And unlike a virtual machine, rather than creating a whole virtual operating system, Docker allows applications to use the same Linux kernel as the system that they're running on and only requires applications be shipped with things not already running on the host computer. This gives a significant performance boost and reduces the size of application. In this project Red Hat Enterprise Linux 8 is my docker host .

■ **PROJECT IMPLEMENTATION**

- a. I have Windows 10 as bare metal setup in my laptop . I'm running Red Hat Enterprise Linux 8 on it using virtualization . Hence my Docker host will be RHEL8 .

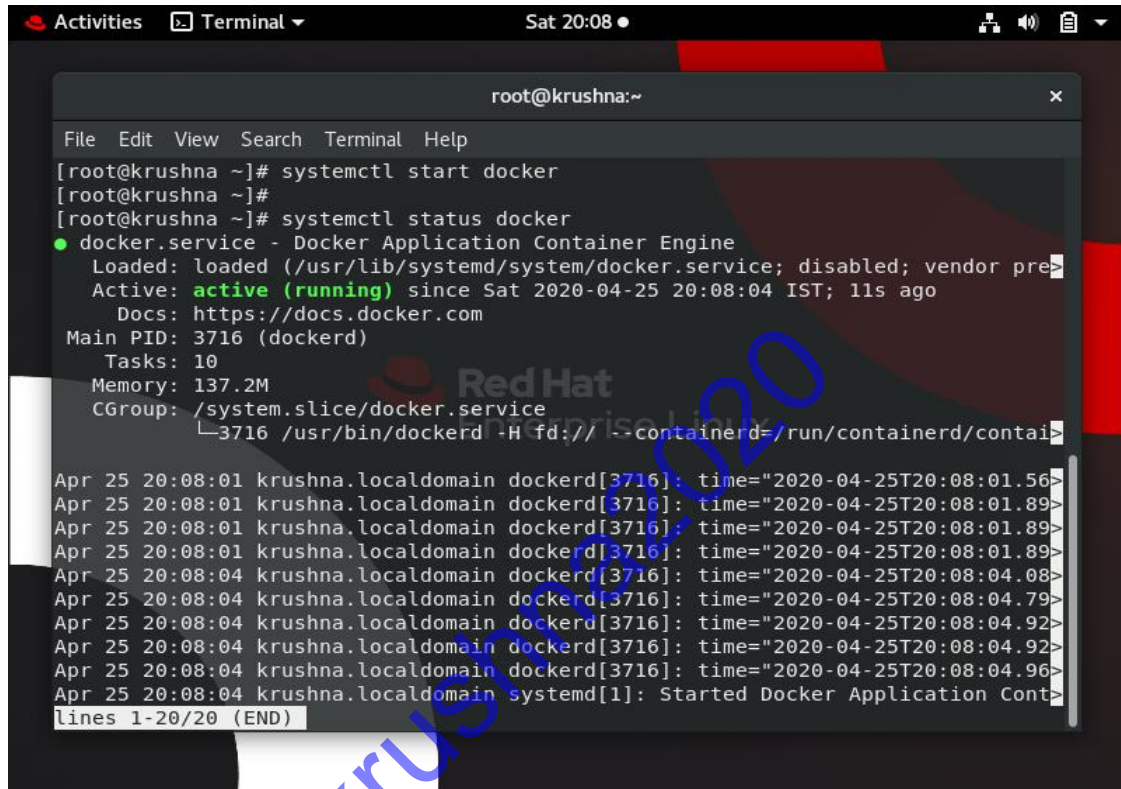


- b. I have installed Docker community edition in it. We can check if this package is installed or not as following :



- c. When Linux boots and starts the OS , the first program loaded in RAM or the first process starts that is "systemd" . Rest all processes

are child process of it. It has process id 1. Docker is a program which will run in the background to provide docker service. Normally these are known as daemon. To start docker service we use systemctl command which comes from systemd. We can use this command to see the status of daemon if it is active or dead.

A screenshot of a terminal window titled 'root@krushna:~'. The terminal shows the output of the 'systemctl status docker' command. The output indicates that the 'docker.service' is loaded, active (running), and has been running since Saturday, 2020-04-25 at 20:08:04 IST. It also shows the main PID as 3716 (dockerd) and lists various system resources like tasks, memory, and cgroup. At the bottom, there is a log of system messages showing the service starting successfully.

```
root@krushna:~  
File Edit View Search Terminal Help  
[root@krushna ~]# systemctl start docker  
[root@krushna ~]#  
[root@krushna ~]# systemctl status docker  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor pre  
   Active: active (running) since Sat 2020-04-25 20:08:04 IST; 11s ago  
     Docs: https://docs.docker.com  
   Main PID: 3716 (dockerd)  
     Tasks: 10  
    Memory: 137.2M  
    CGroup: /system.slice/docker.service  
           └─3716 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai  
  
Apr 25 20:08:01 krushna.localdomain dockerd[3716]: time="2020-04-25T20:08:01.56>  
Apr 25 20:08:01 krushna.localdomain dockerd[3716]: time="2020-04-25T20:08:01.89>  
Apr 25 20:08:01 krushna.localdomain dockerd[3716]: time="2020-04-25T20:08:01.89>  
Apr 25 20:08:01 krushna.localdomain dockerd[3716]: time="2020-04-25T20:08:01.89>  
Apr 25 20:08:04 krushna.localdomain dockerd[3716]: time="2020-04-25T20:08:04.08>  
Apr 25 20:08:04 krushna.localdomain dockerd[3716]: time="2020-04-25T20:08:04.79>  
Apr 25 20:08:04 krushna.localdomain dockerd[3716]: time="2020-04-25T20:08:04.92>  
Apr 25 20:08:04 krushna.localdomain dockerd[3716]: time="2020-04-25T20:08:04.92>  
Apr 25 20:08:04 krushna.localdomain dockerd[3716]: time="2020-04-25T20:08:04.96>  
Apr 25 20:08:04 krushna.localdomain systemd[1]: Started Docker Application Cont  
lines 1-20/20 (END)
```

- d. In general when we install an Operating system, we require CD/DVD or iso image file. Similarly, for creating a docker container or typically a light weight OS we need docker images. From a single image multiple container can be created. We can customize an image according to our need and commit it.

In this project I need two dedicated servers for which I have to get their corresponding images. For web server and database server I can create my own images. However these are popular, so I can get them from docker hub.

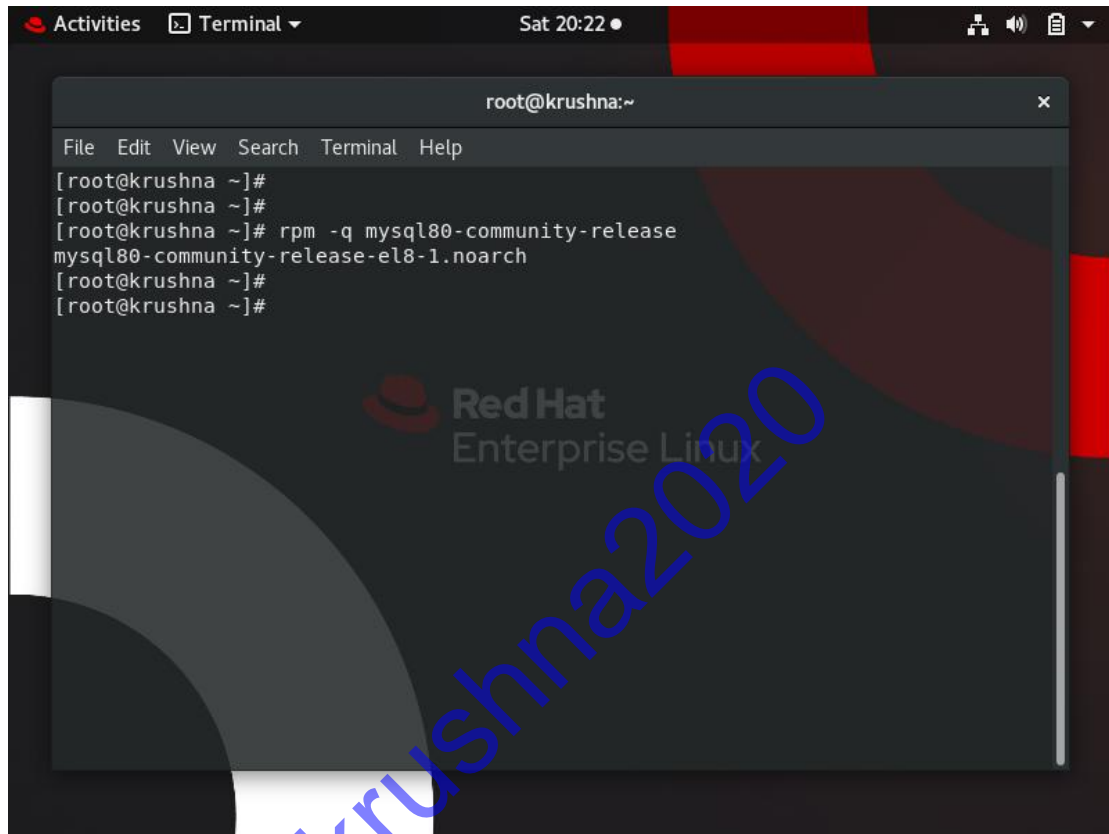
For database server I will be using mysql : 5.7 and for web server wordpress : 5.5.1-php7.3-apache. Both of them contains all my required packages. Docker pull command is used for this.


```
Activities Terminal Sat 20:09
root@krushna:~
File Edit View Search Terminal Help
[root@krushna ~]#
[root@krushna ~]# docker pull wordpress:5.1.1-php7.3-apache
5.1.1-php7.3-apache: Pulling from library/wordpress
Digest: sha256:e52dab89ab4c24e21ec490f606d6d415fb15fb96008a5dc4b1d9956e2e5fcfc3
Status: Image is up to date for wordpress:5.1.1-php7.3-apache
docker.io/library/wordpress:5.1.1-php7.3-apache
[root@krushna ~]#
[root@krushna ~]#
[root@krushna ~]#
[root@krushna ~]# docker pull mysql:5.7
5.7: Pulling from library/mysql
54fec2fa59d0: Pulling fs layer
bcc6c6145912: Download complete
951c3d959c9d: Downloading 425.1kB/4.178MB
05de4d0e206e: Waiting
319f0394ef42: Waiting
d9185034607b: Waiting
013a9c64dadc: Waiting
58b7b840ebff: Waiting
9b85c0abc43d: Waiting
bdf022f63e85: Waiting
35f7f707ce83: Waiting
```

e. After downloading the images we can check our image list as follows :

```
Activities Terminal Sat 20:12
root@krushna:~
File Edit View Search Terminal Help
[root@krushna ~]#
[root@krushna ~]#
[root@krushna ~]# docker image ls
REPOSITORY          TAG          IMAGE ID          CREATED
mywebserver          v1           ce37f4bf5001      43 hours ago
434MB
mysql                5.7          413be204e9c3      3 weeks ago
456MB
centos                latest       470671670cac      3 months ago
237MB
centos                7           5e35e350aded      5 months ago
203MB
wordpress            5.1.1-php7.3-apache a69f6702fdda      11 months ago
422MB
[root@krushna ~]#
[root@krushna ~]#
```

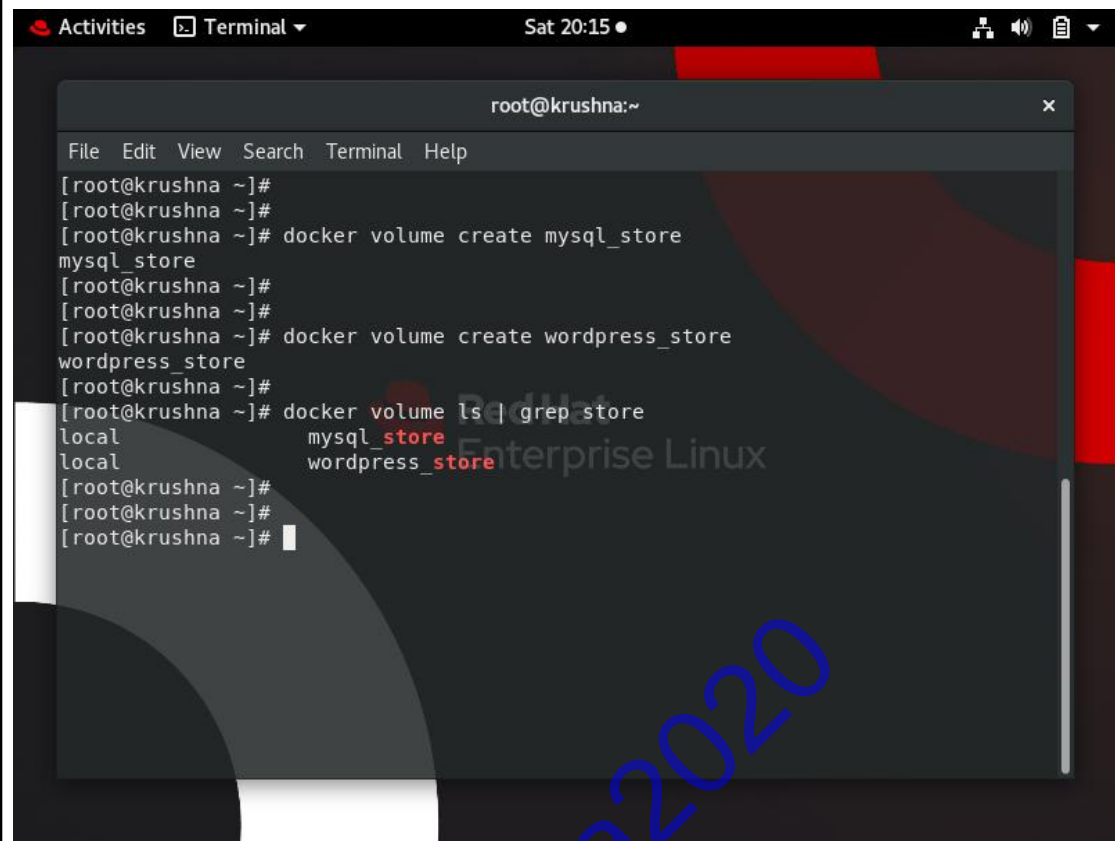
- f. From base OS if I want some database management work , I need a client for accessing the server. Many clients are available for accessing mysql server. They can be installed using yum command. I've already one client program for it.



```
root@krushna:~  
File Edit View Search Terminal Help  
[root@krushna ~]#  
[root@krushna ~]#  
[root@krushna ~]# rpm -q mysql80-community-release  
mysql80-community-release-el8-1.noarch  
[root@krushna ~]#  
[root@krushna ~]#
```

- g. The data which will be stored in database server & web server are ephemeral in nature, because if due to any reason my containers terminated I'll loose my servers as well as data. However later I can launch new server containers again from those images but I won't be able to get data back. So I want to store data persistent.

For this I'll create two separate docker volumes and attach them to the servers' document root where original data are stored. This docker volume takes storage from docker host. In my case it will consume space from RHEL8 . Docker volumes can be created as follows :

A screenshot of a Linux terminal window titled 'root@krushna:~'. The terminal shows the following commands and output:

```
File Edit View Search Terminal Help
[root@krushna ~]#
[root@krushna ~]#
[root@krushna ~]# docker volume create mysql_store
mysql_store
[root@krushna ~]#
[root@krushna ~]#
[root@krushna ~]# docker volume create wordpress_store
wordpress_store
[root@krushna ~]#
[root@krushna ~]# docker volume ls | grep store
local                mysql_store
local                wordpress_store
[root@krushna ~]#
[root@krushna ~]#
[root@krushna ~]#
```

A large blue diagonal watermark reading '©krushna2020' is overlaid across the terminal output.

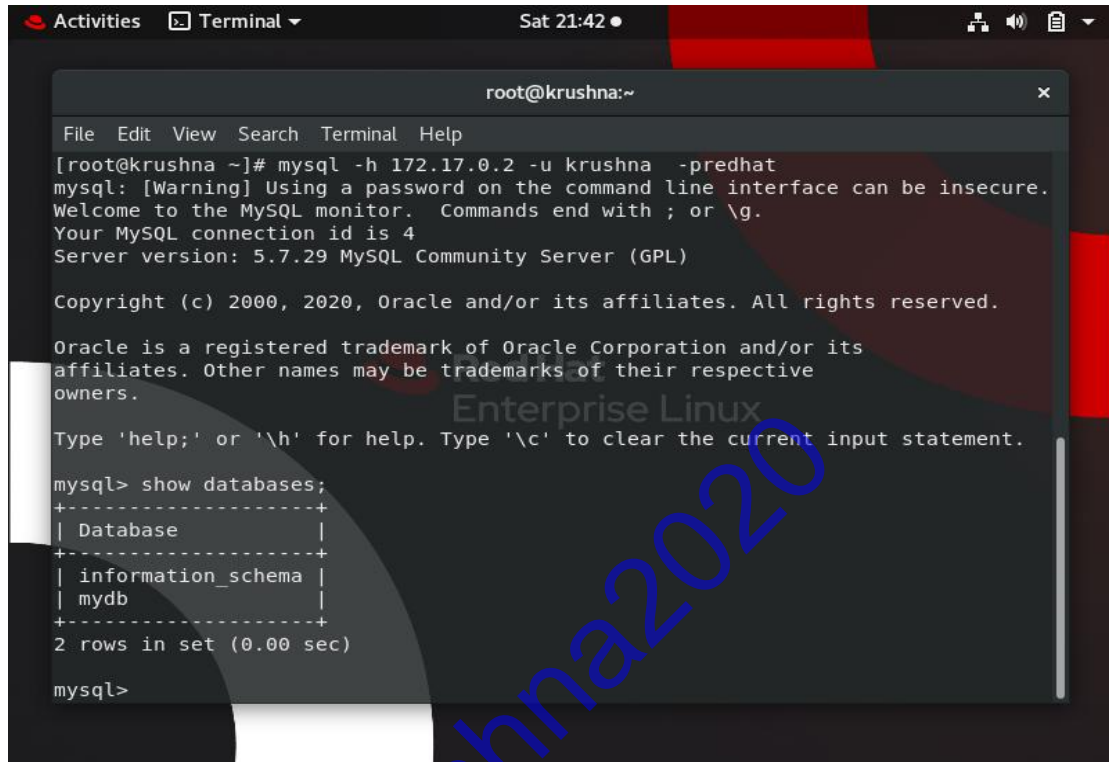
- h. Now it's time to launch the servers . Here the database server I'm using has entry point which requires some environmental variables to be passed during container launch. This details are provided in the images documentation in docker hub. This DB server basically needs MYSQL_ROOT_PASSWORD (for root login), MYSQL_USER (create another user), MYSQL_PASSWORD(password for previous user created), MYSQL_DATABASE(creates database). In my context I've given rootpass, krushna , redhat, mydb respectively for above entry points. '-e' is used for environmental variables. '-v ' is used for mounting my created volume to the document root directory '/var/lib/mysql'. I have given the name of this server as dbos . Docker run command is used for launching the container from an image as shown below . The option '-it ' is used for getting an interactive terminal of the container . At last I need to mention the image name with version from which the container will be launched.

```
Activities Terminal Sat 21:22
root@krushna:~
File Edit View Search Terminal Help
[root@krushna ~]#
[root@krushna ~]# docker run -it -e MYSQL_ROOT_PASSWORD=rootpass -e
MYSQL_USER=krushna -e MYSQL_PASSWORD=redhat -e
MYSQL_DATABASE=mydb -v mysql_store:/var/lib/mysql
--name dbos mysql:5.7
2020-04-25 15:51:38+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Serve
r 5.7.29-1debian10 started.
2020-04-25 15:51:38+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysq
l'
2020-04-25 15:51:38+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Serve
r 5.7.29-1debian10 started.
2020-04-25 15:51:39+00:00 [Note] [Entrypoint]: Initializing database files
2020-04-25T15:51:39.022999Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is
deprecated. Please use --explicit_defaults_for_timestamp server option (see doc
umentation for more details).
2020-04-25T15:51:39.916302Z 0 [Warning] InnoDB: New log files created, LSN=45790
2020-04-25T15:51:40.325667Z 0 [Warning] InnoDB: Creating foreign key constraint
system tables.
2020-04-25T15:51:40.497537Z 0 [Warning] No existing UUID has been found, so we a
ssume that this is the first time that this server has been started. Generating
a new UUID: a755761d-870c-11ea-bccd-0242ac110002.
2020-04-25T15:51:40.534265Z 0 [Warning] Gtid table is not ready to be used. Tabl
e 'mysql.gtid_executed' cannot be opened.
2020-04-25T15:51:42.754906Z 0 [Warning] CA certificate ca.pem is self signed.
```

- i. Now to access database from my client program first I need the IP address of the server. Docker inspect command gives the all details of the respective container. For IP address :

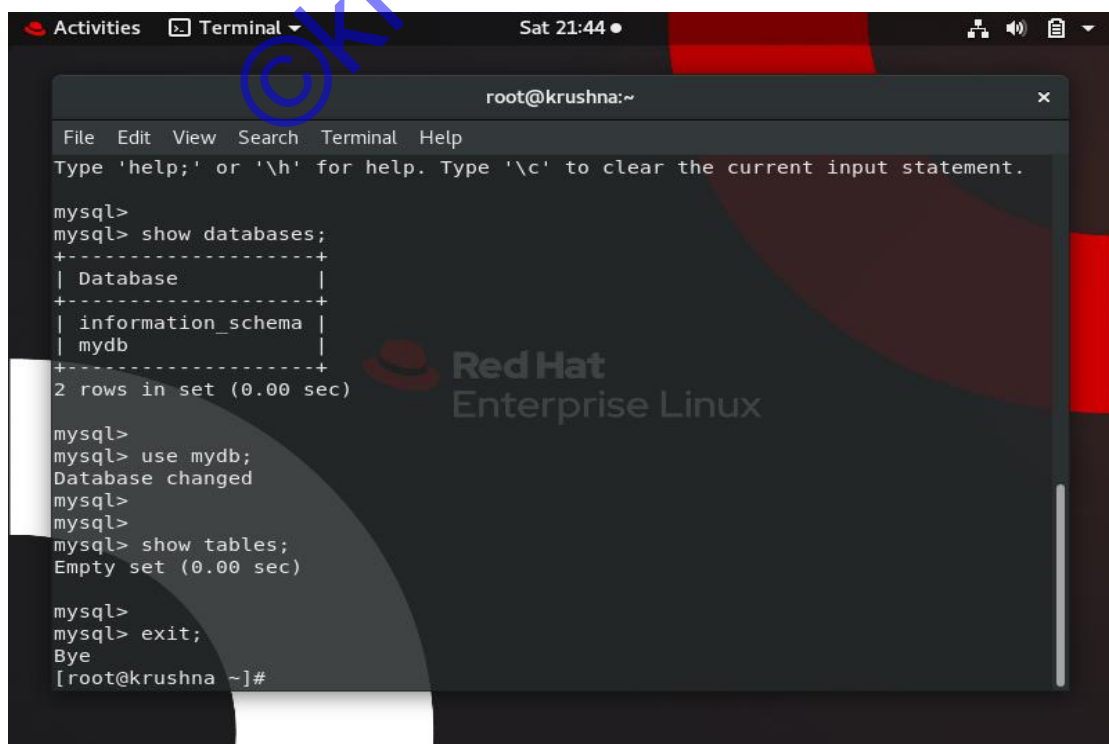
```
Activities Terminal Sat 21:38
root@krushna:~
File Edit View Search Terminal Help
[root@krushna ~]#
[root@krushna ~]# docker inspect dbos | grep IP
"LinkLocalIPv6Address": "",
"LinkLocalIPv6PrefixLen": 0,
"SecondaryIPAddresses": null,
"SecondaryIPv6Addresses": null,
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"IPAddress": "172.17.0.2",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"IPAMConfig": null,
"IPAddress": "172.17.0.2",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
[root@krushna ~]#
```

- j. Now after getting IP , I can access to the database , manage data and everything by providing the user name(krushna) & password(redhat).

A terminal window titled 'root@krushna:~' showing a MySQL command-line interface. The user has connected to the MySQL server at 172.17.0.2 as user 'krushna' with password 'predhat'. The terminal displays the MySQL welcome message and the output of the 'show databases;' command, which lists 'information_schema' and 'mydb'.

```
root@krushna:~  
File Edit View Search Terminal Help  
[root@krushna ~]# mysql -h 172.17.0.2 -u krushna -predhat  
mysql: [Warning] Using a password on the command line interface can be insecure.  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 4  
Server version: 5.7.29 MySQL Community Server (GPL)  
  
Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mydb |  
+-----+  
2 rows in set (0.00 sec)  
  
mysql>
```

- k. Now I can see that 'mydb' database is created. But it is empty.

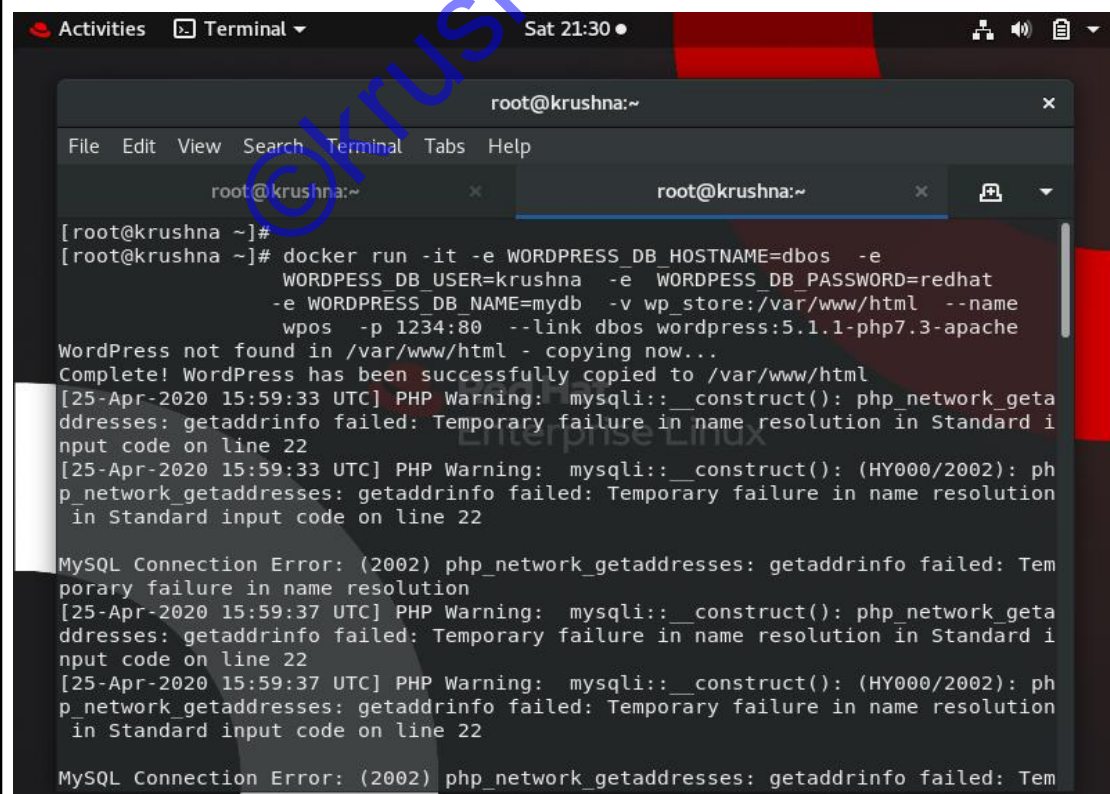
A terminal window titled 'root@krushna:~' showing the MySQL command-line interface. The user has entered 'show databases;' which shows 'information_schema' and 'mydb'. Then they entered 'use mydb;' and 'show tables;', which shows an empty set of tables. Finally, they entered 'exit;' to return to the shell.

```
root@krushna:~  
File Edit View Search Terminal Help  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql>  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mydb |  
+-----+  
2 rows in set (0.00 sec)  
  
mysql>  
mysql> use mydb;  
Database changed  
mysql>  
mysql> show tables;  
Empty set (0.00 sec)  
  
mysql>  
mysql> exit;  
Bye  
[root@krushna ~]#
```


- I. Now web server will be launched. It also requires some environmental variables like WORDPRESS_DB_HOSTNAME (database host name), WORDPRESS_DB_USER (database user), WORDPRESS_DB_PASSWORD (password of previous user), WORDPRESS_DB_NAME (the database on which WordPress depends). I've provided the details as dbos, krushna, redhat, mydb respectively. I'll also bind my previously created docker volume to its document root where my webpages will be stored. I've given name of this container as wpos

In a machine processes or services run on port numbers. There are 65,536 logical ports are there. Web server basically uses port no. 80. As my server will be in internal network so from outside world having public IPs can't connect to it. For achieving this I'll use the Port Address Translation concept. By this my web server container will be exposed to outside network.

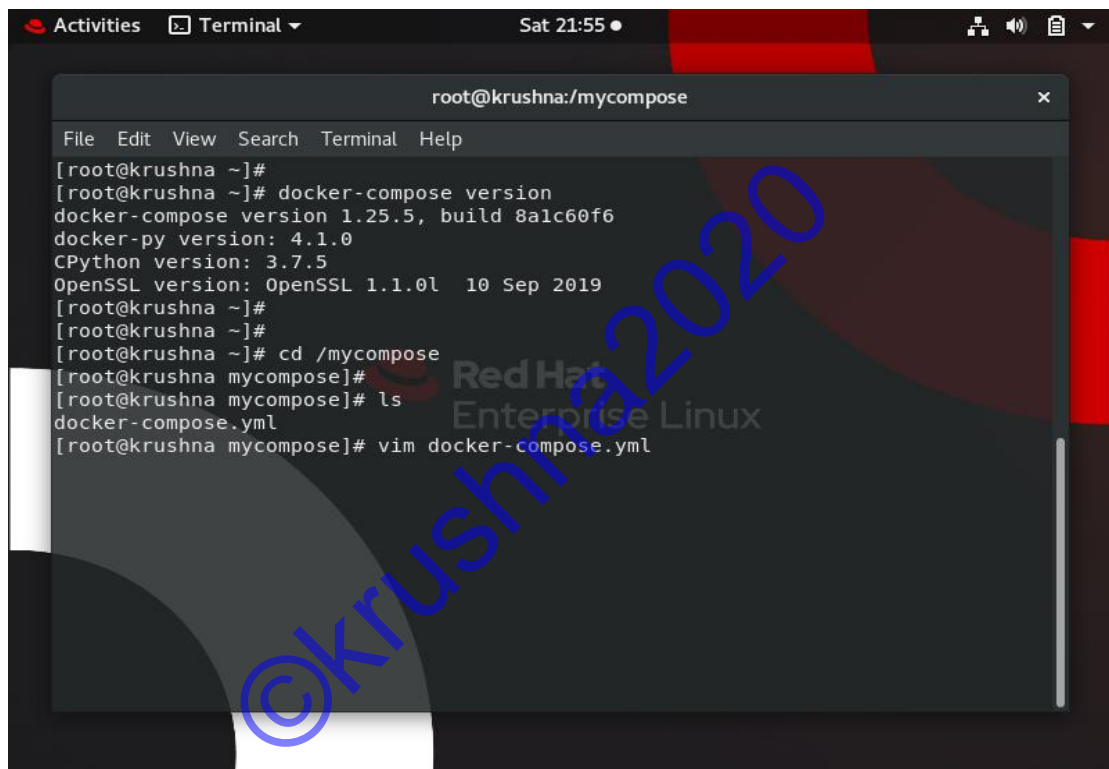
One more thing, technically this server depends on DB server. Till now I've just given the DB host name, but I need to update the DNS record too. '--link' is used for it.



```
root@krushna:~  
[root@krushna ~]# docker run -it -e WORDPRESS_DB_HOSTNAME=dbos -e  
WORDPRESS_DB_USER=krushna -e WORDPRESS_DB_PASSWORD=redhat  
-e WORDPRESS_DB_NAME=mydb -v wp_store:/var/www/html --name  
wpos -p 1234:80 --link dbos wordpress:5.1.1-php7.3-apache  
WordPress not found in /var/www/html - copying now...  
Complete! WordPress has been successfully copied to /var/www/html  
[25-Apr-2020 15:59:33 UTC] PHP Warning: mysqli::__construct(): php_network_geta  
ddresses: getaddrinfo failed: Temporary failure in name resolution in Standard i  
nput code on line 22  
[25-Apr-2020 15:59:33 UTC] PHP Warning: mysqli::__construct(): (HY000/2002): ph  
p_network_getaddresses: getaddrinfo failed: Temporary failure in name resolution  
in Standard input code on line 22  
MySQL Connection Error: (2002) php_network_getaddresses: getaddrinfo failed: Tem  
porary failure in name resolution  
[25-Apr-2020 15:59:37 UTC] PHP Warning: mysqli::__construct(): php_network_geta  
ddresses: getaddrinfo failed: Temporary failure in name resolution in Standard i  
nput code on line 22  
[25-Apr-2020 15:59:37 UTC] PHP Warning: mysqli::__construct(): (HY000/2002): ph  
p_network_getaddresses: getaddrinfo failed: Temporary failure in name resolution  
in Standard input code on line 22  
MySQL Connection Error: (2002) php_network_getaddresses: getaddrinfo failed: Tem
```


m. Now up to this I've made this infrastructure step by step by commands. So if in any other environment I want to make this set up again I have follow the whole steps again . There is a chance of mistakes too. Thus I can write a code which will automate the things for me.

To achieve this I'll use the concept of 'Infrastructure As Code'. This is possible using Docker-compose . It does not come with docker-community edition by default. I've installed it separately .

A screenshot of a terminal window titled 'root@krushna:/mycompose'. The terminal shows the following commands and output:

```
File Edit View Search Terminal Help
[root@krushna ~]#
[root@krushna ~]# docker-compose version
docker-compose version 1.25.5, build 8alc60f6
docker-py version: 4.1.0
CPython version: 3.7.5
OpenSSL version: OpenSSL 1.1.0l 10 Sep 2019
[root@krushna ~]#
[root@krushna ~]#
[root@krushna ~]# cd /mycompose
[root@krushna mycompose]#
[root@krushna mycompose]# ls
docker-compose.yml
[root@krushna mycompose]# vim docker-compose.yml
```

A large blue watermark '©krushna2020' is diagonally across the terminal output. The background of the terminal window shows a Red Hat Enterprise Linux logo.

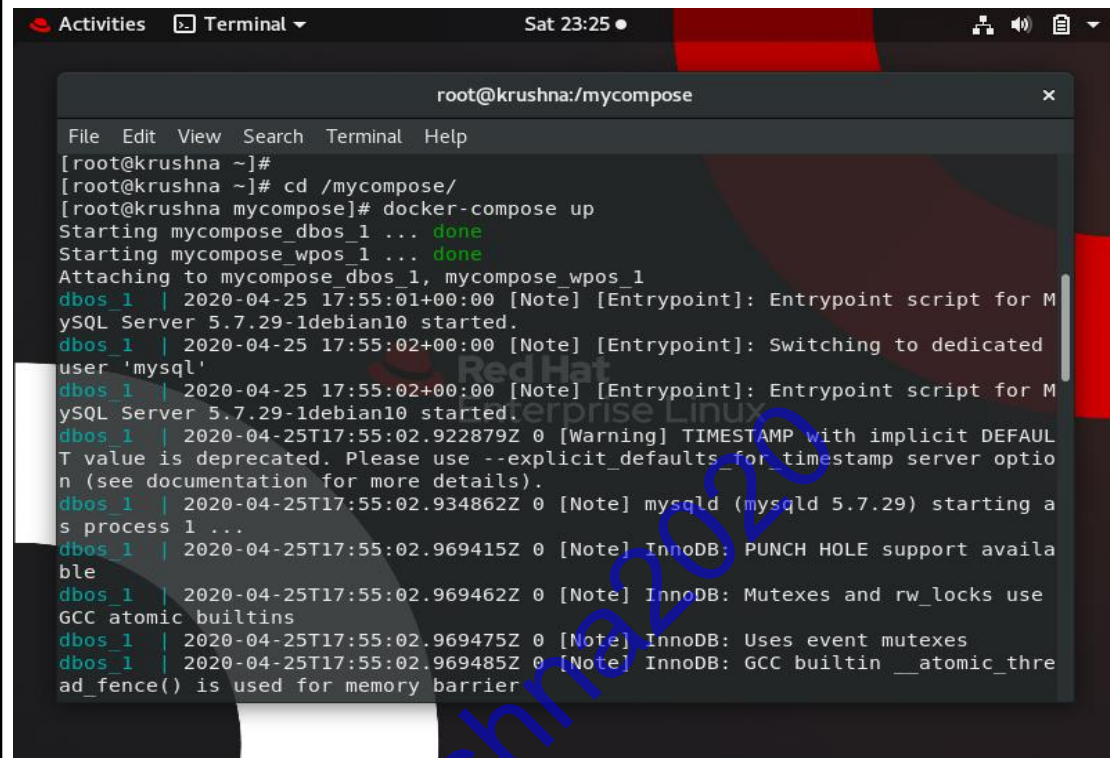
n. Now I created a directory 'mycompose' , inside which the main program is written in yml format and saved as 'docker-compose.yml' .

As shown following is the code in vim editor of redhat. It is written in yml format with proper indentation. In the following code I've just created two new volumes data_store for mysql database & word_store for webpages hosting.

```
root@krushna:/mycompose
File Edit View Search Terminal Tabs Help
root@krushna:/mycompose root@krushna:/mycompose
version: '3'
services:
  dbos:
    image: mysql:5.7
    volumes:
      - data_store:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: rootpass
      MYSQL_USER: krushna
      MYSQL_PASSWORD: redhat
      MYSQL_DATABASE: mydb
  wpos:
    image: wordpress:5.1.1-php7.3-apache
    restart: always
    depends_on:
      - dbos
    ports:
      - 8081:80
    environment:
      WORDPRESS_DB_HOST: dbos
-- INSERT -- 2,1 Top
```

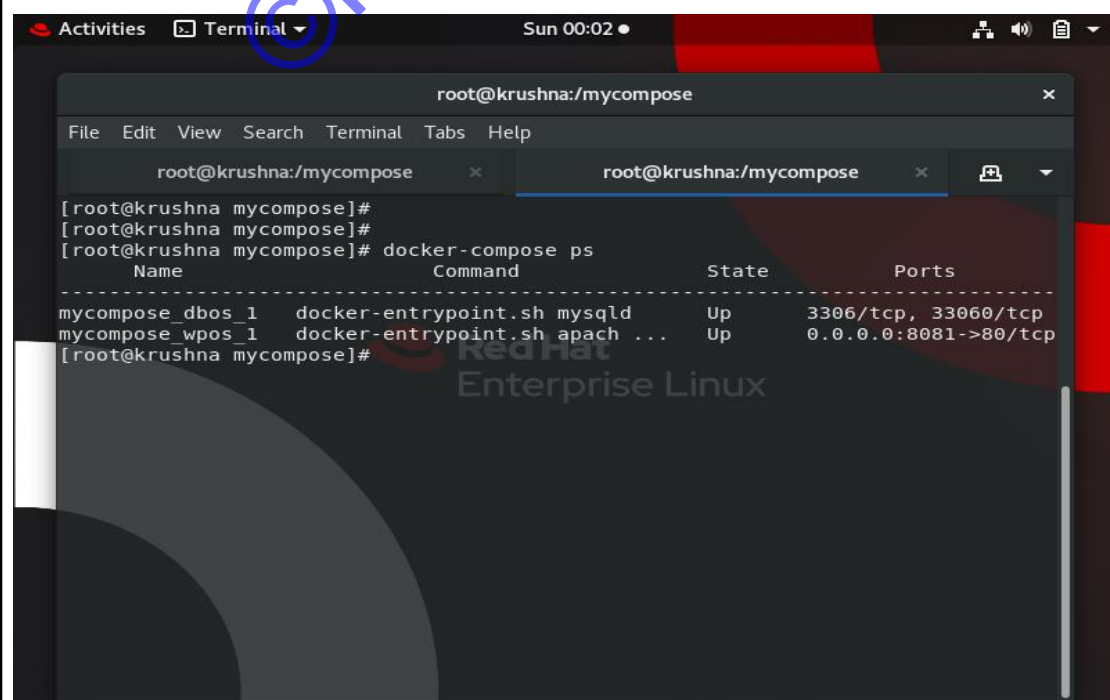
```
root@krushna:/mycompose
File Edit View Search Terminal Tabs Help
root@krushna:/mycompose root@krushna:/mycompose
  MYSQL_PASSWORD: redhat
  MYSQL_DATABASE: mydb
  wpos:
    image: wordpress:5.1.1-php7.3-apache
    restart: always
    depends_on:
      - dbos
    ports:
      - 8081:80
    environment:
      WORDPRESS_DB_HOST: dbos
      WORDPRESS_DB_USER: krushna
      WORDPRESS_DB_PASSWORD: redhat
      WORDPRESS_DB_NAME: mydb
    volumes:
      - word_store:/var/www/html
  volumes:
    data_store:
    word_store:
-- INSERT -- 33,1 Bot
```

- o. Now after this from the same directory I'll execute my program using docker-compose up command. It will do all the manual steps automatically for me.



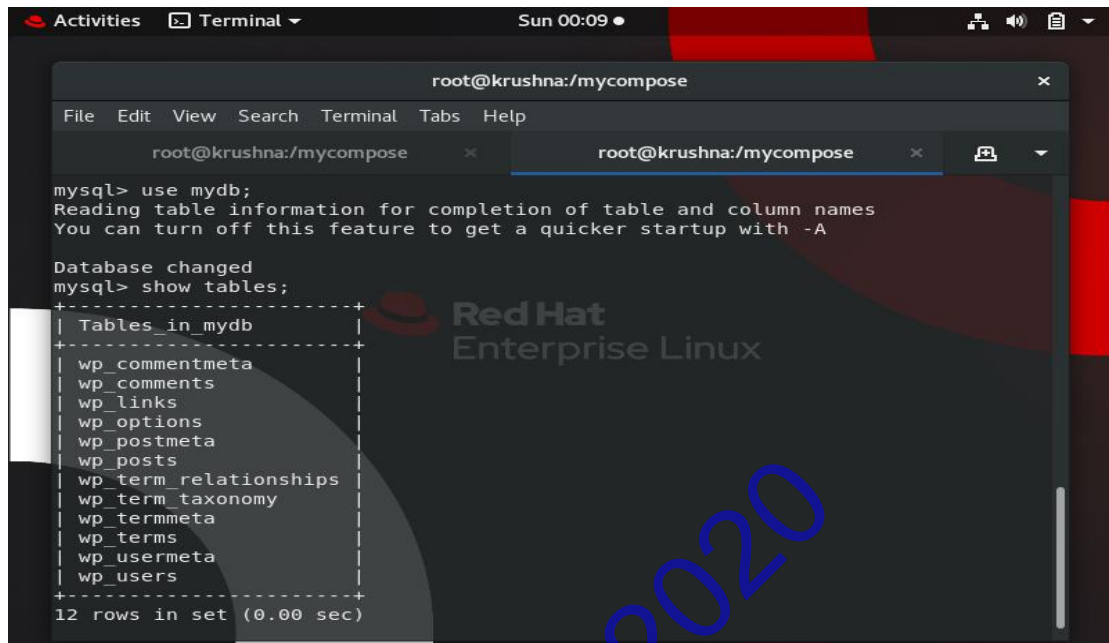
```
root@krushna:/mycompose
File Edit View Search Terminal Help
[root@krushna ~]#
[root@krushna ~]# cd /mycompose/
[root@krushna mycompose]# docker-compose up
Starting mycompose_dbos_1 ... done
Starting mycompose_wpos_1 ... done
Attaching to mycompose_dbos_1, mycompose_wpos_1
dbos_1 | 2020-04-25 17:55:01+00:00 [Note] [Entrypoint]: Entrypoint script for M
ySQL Server 5.7.29-1debian10 started.
dbos_1 | 2020-04-25 17:55:02+00:00 [Note] [Entrypoint]: Switching to dedicated
user 'mysql'
dbos_1 | 2020-04-25 17:55:02+00:00 [Note] [Entrypoint]: Entrypoint script for M
ySQL Server 5.7.29-1debian10 started.
dbos_1 | 2020-04-25T17:55:02.922879Z 0 [Warning] TIMESTAMP with implicit DEFAULT
T value is deprecated. Please use --explicit_defaults_for_timestamp server optio
n (see documentation for more details).
dbos_1 | 2020-04-25T17:55:02.934862Z 0 [Note] mysqld (mysqld 5.7.29) starting a
s process 1 ...
dbos_1 | 2020-04-25T17:55:02.969415Z 0 [Note] InnoDB: PUNCH HOLE support availa
ble
dbos_1 | 2020-04-25T17:55:02.969462Z 0 [Note] InnoDB: Mutexes and rw_locks use
GCC atomic builtins
dbos_1 | 2020-04-25T17:55:02.969475Z 0 [Note] InnoDB: Uses event mutexes
dbos_1 | 2020-04-25T17:55:02.969485Z 0 [Note] InnoDB: GCC builtin __atomic_thre
ad_fence() is used for memory barrier
```

- p. Now using docker-compose ps command I can see the running containers .



```
root@krushna:/mycompose
File Edit View Search Terminal Tabs Help
root@krushna:/mycompose x root@krushna:/mycompose x
[root@krushna mycompose]#
[root@krushna mycompose]#
[root@krushna mycompose]# docker-compose ps
Name Command State Ports
-----
mycompose_dbos_1 docker-entrypoint.sh mysqld Up 3306/tcp, 33060/tcp
mycompose_wpos_1 docker-entrypoint.sh apach ... Up 0.0.0.0:8081->80/tcp
[root@krushna mycompose]#
```

- q. Now if I check my database , I can see the inside 'mydb' some tables are created.

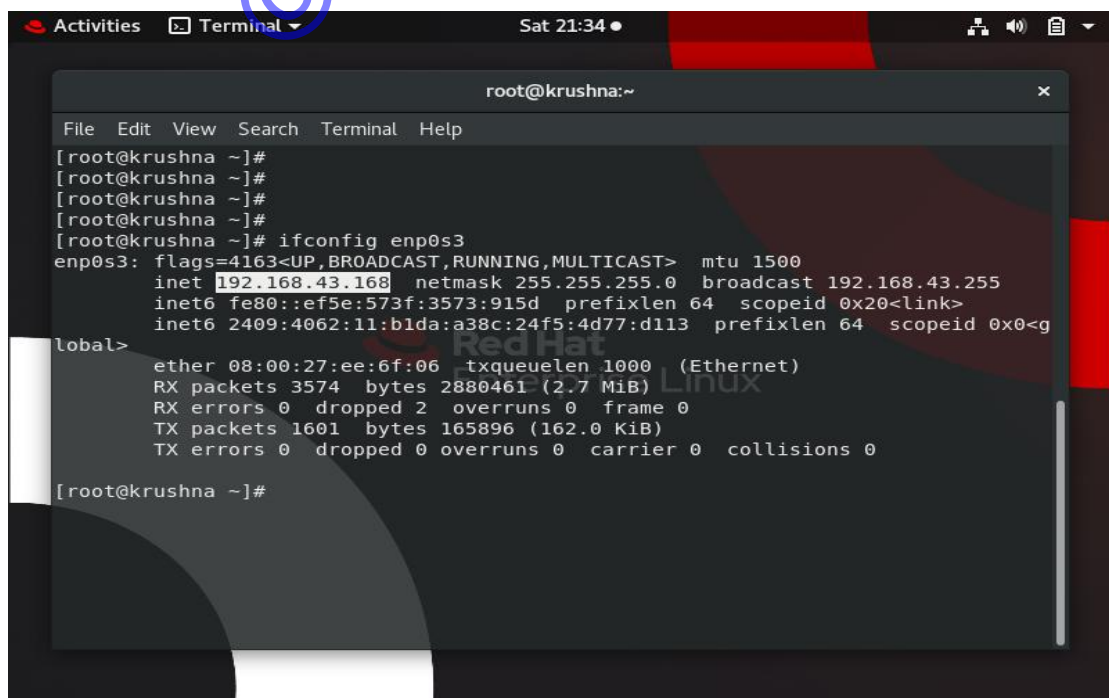


A terminal window titled 'root@krushna:/mycompose' showing the execution of MySQL commands. The user enters 'mysql> use mydb;' and receives a message about table information completion. Then, they enter 'mysql> show tables;' and see a list of 12 tables in the 'mydb' database. The tables are: wp_commentmeta, wp_comments, wp_links, wp_options, wp_postmeta, wp_posts, wp_term_relationships, wp_term_taxonomy, wp_termmeta, wp_terms, wp_usermeta, and wp_users. The output is formatted as a table with a header row 'Tables_in_mydb'.

```
mysql> use mydb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_mydb |
+-----+
| wp_commentmeta |
| wp_comments    |
| wp_links       |
| wp_options     |
| wp_postmeta    |
| wp_posts       |
| wp_term_relationships |
| wp_term_taxonomy |
| wp_termmeta    |
| wp_terms       |
| wp_usermeta    |
| wp_users       |
+-----+
12 rows in set (0.00 sec)
```

- r. Now my back end architecture is ready. Let's check go to the front end part. I'll access front end from Windows 10 browser and design there my blog. For this I need my IP address of Docker host as I've enabled port forwarding. So when any client request come to IP '192.168.43.168' on port no. 8081, router will forward it to the webserver on port 80.

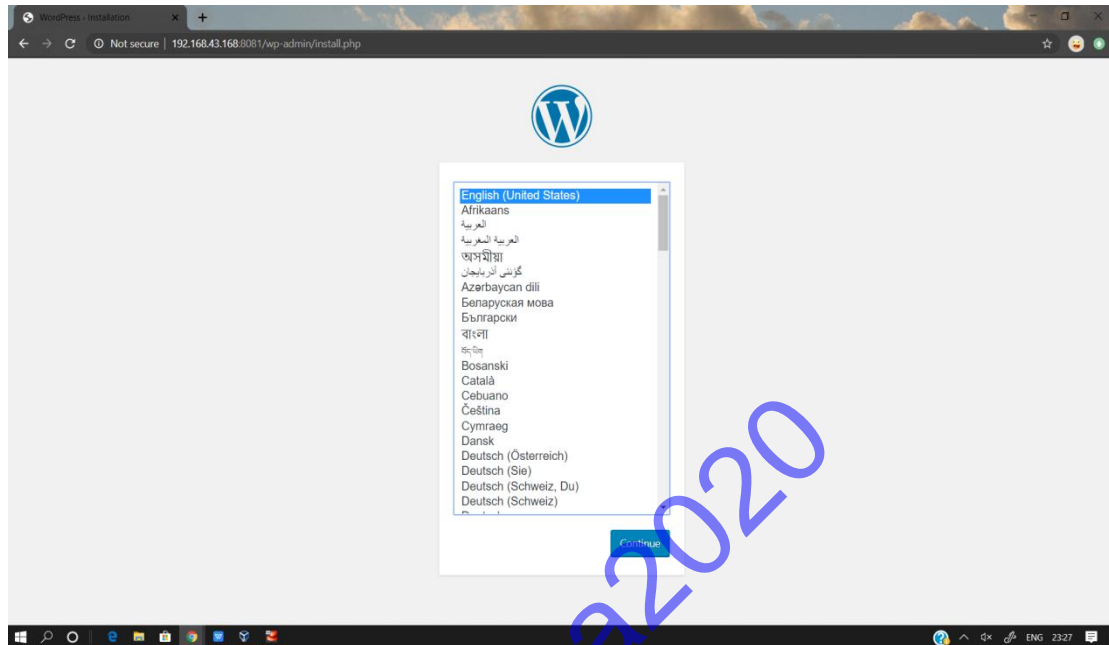


A terminal window titled 'root@krushna:~' showing the output of the 'ifconfig enp0s3' command. The output displays various network parameters for the 'enp0s3' interface, including flags, MTU, IP address (192.168.43.168), netmask, broadcast address, and MAC address. It also shows statistics for RX and TX packets, bytes, and errors.

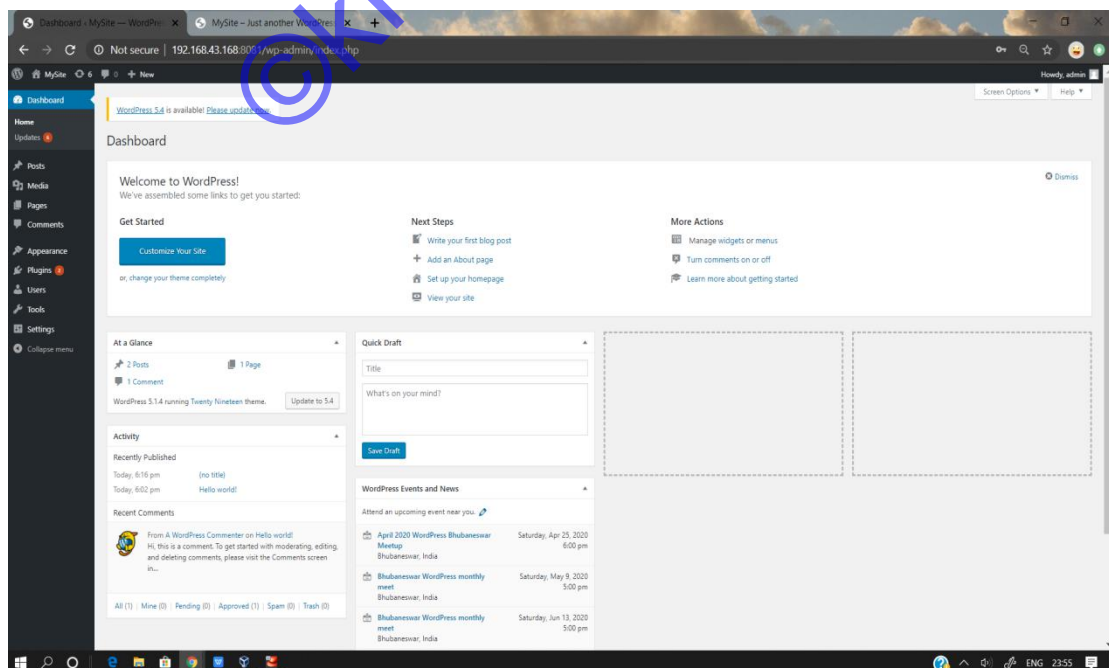
```
[root@krushna ~]# ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.43.168 netmask 255.255.255.0 broadcast 192.168.43.255
    inet6 fe80::ef5e:573f:3573:915d prefixlen 64 scopeid 0x20<link>
    inet6 2409:4062:11:b1da:a38c:24f5:4d77:d113 prefixlen 64 scopeid 0x0<global>
    ether 08:00:27:ee:6f:06 txqueuelen 1000 (Ethernet)
    RX packets 3574 bytes 2880461 (2.7 MiB)
    RX errors 0 dropped 2 overruns 0 frame 0
    TX packets 1601 bytes 165896 (162.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@krushna ~]#
```

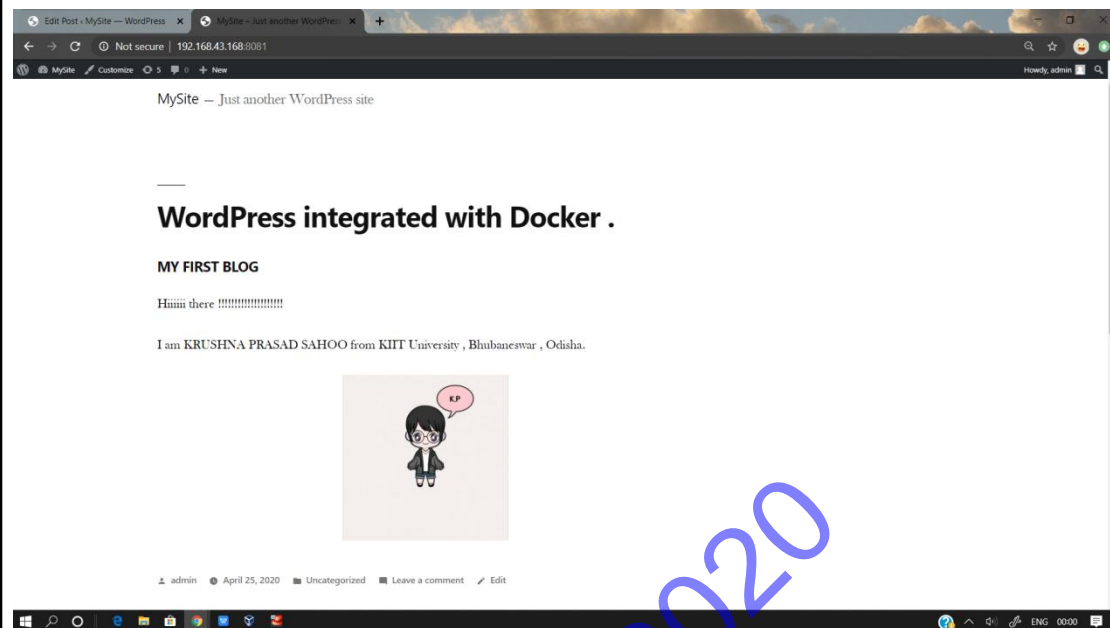
- s. Now in my Windows Chrome browser I browse for address ' 192.168.43.168:8081 '. Then I can access my WordPress installation page.



- t. After installation & following subsequent steps I can find the administrator dashboard. There I can create my own blog post using various GUI features.



u. After creating my page and all , I published the page. Now I can successfully access to my web page.



■ RESULTS AND DISCUSSIONS

The purpose of this project is achieved after these much steps. This project surely helped me a lot to gain knowledge in the fields which I've had only heard about and never researched on it.

Thanks to this project, I covered from basics to intermediate level of Docker technology, which is used in DevOps to manage software parts as isolated , self-sufficient containers, and can be deployed and run in any environment. Docker reduces back and forth between Dev and Ops in continuous deployment, which eliminates overheads and cuts operational costs . It also motivated me to learn Kubernetes, Jenkins, Openshift kind of High-End technologies.

Also , I got to know about the WordPress CMS tool, using which simple, attractive and user-friendly websites can be made very easily . It provides GUIs and variety plugins which can be used for web development.

■ **FUTURE SCOPE**

In this project I've integrated WordPress with Docker tool in small scale. This idea can be implemented in large scale for real industry use cases. For any small business organisation or companies, they can develop a full-fledged website on the top of Docker. Kubernetes can also be used in this case.

Automation in the industrial workplace provides the advantages of improving productivity and quality while reducing errors and resource wastage, increasing safety, and adding flexibility to the manufacturing process. Achieving automation across process flows is not a easy task, but DevOps by improving collaboration between development and operations team, helps to reach a very high limit of productivity by automating workflows. This is the future demand in fact .

■ **CONCLUSION**

At the end I would like to say that Technology is always created by human and in turn re-defining what we can and will do. Every single technological change is now impacting humanity in much deeper way than ever before because it'll soon impact our own biology , primarily via the rise of genome editing and AI .Technology is no longer just a tool we use to achieve something , the merging of machine capability and human consciousness is already happening.

However, Mankind's greatest fear is unknown. So a continually co-evolving well dynamic relationship should exit between Technology and Human Culture .Technological developments should be done for a good cause of our country & our world .