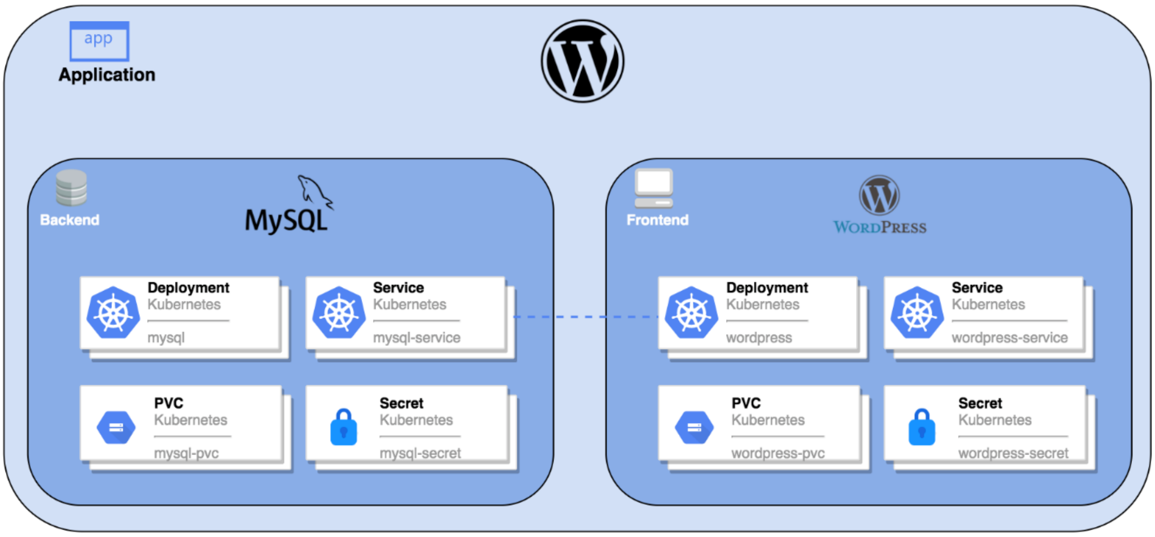
Deploying WordPress Application with MySQL DB through Kubernetes

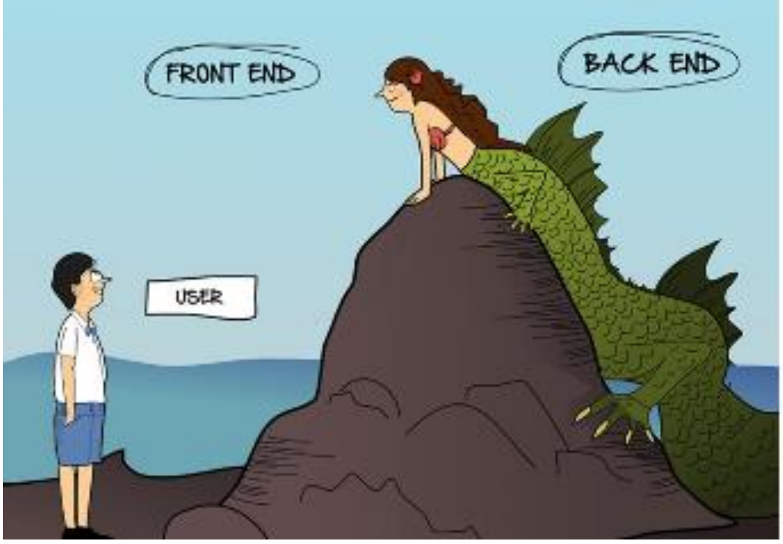


Applications like Facebook and many more gives services of connections and all other database related services. But nobody knows or wants to know what's really behind them. And can we target to the database of those Applications.??

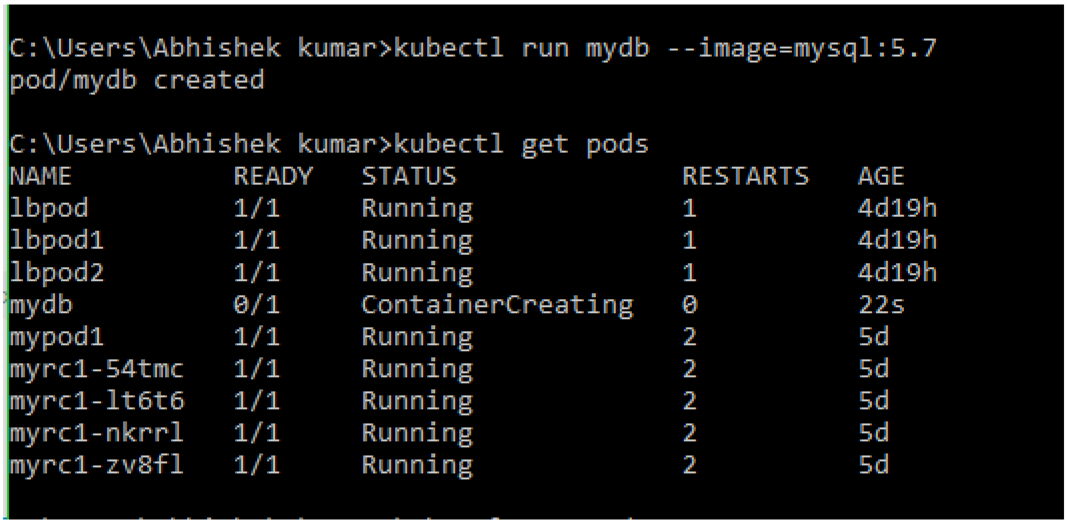
Answer is No.

As They work as a Multi-tier Application.
Means they are divided into two halves, first one is Front end which can be Accessed by the clients, and all the admins, it is open to all. The main thing is the back-end where the main data is stored.
Similar like thing we will do today,
We will launch a PHP Application ->Wordpress [Front-end]-> it's a webapp.
And the other one is MYSQL database -> back-end.
Both ends needs to be connected good. But the backend should only have connection to the front-end and not to the clients.[isolated from public world]

Actually the scene is this. We{clients}
Are only looking to the Front end.
No one can see the back end.

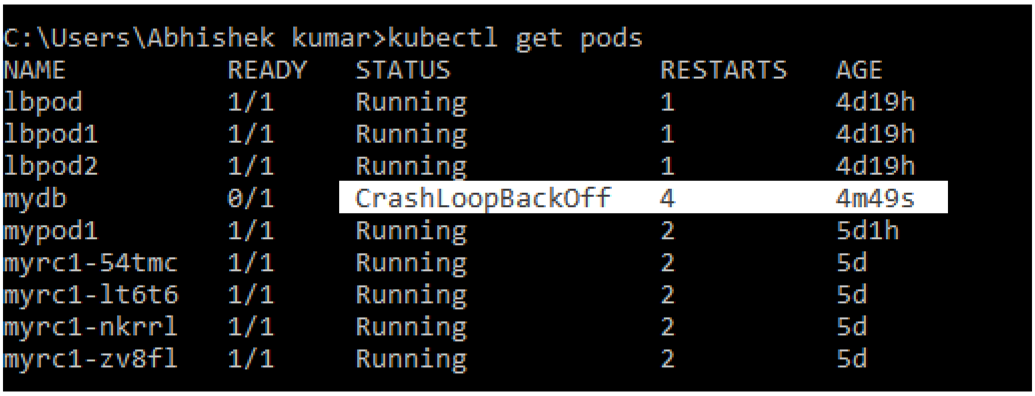


Let’s Start with Mysql(back-end)
With supported version.
Kubectl run mydb –image=mysql:5.7



```
C:\Users\Abhishek kumar>kubectl run mydb --image=mysql:5.7
pod/mydb created

C:\Users\Abhishek kumar>kubectl get pods
NAME           READY   STATUS              RESTARTS   AGE
lbpod          1/1     Running             1          4d19h
lbpod1         1/1     Running             1          4d19h
lbpod2         1/1     Running             1          4d19h
mydb           0/1     ContainerCreating   0          22s
mypod1         1/1     Running             2          5d
myrc1-54tmc    1/1     Running             2          5d
myrc1-lt6t6    1/1     Running             2          5d
myrc1-nkrrl    1/1     Running             2          5d
myrc1-zv8fl    1/1     Running             2          5d
```

Now,
Let’s check for the status of updated status
Kubectl get pods



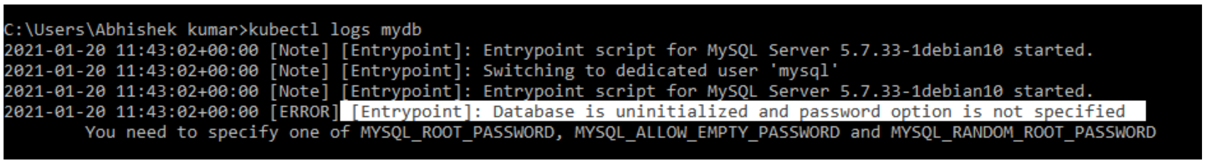
```
C:\Users\Abhishek kumar>kubectl get pods
NAME           READY   STATUS             RESTARTS   AGE
lbpod          1/1     Running            1          4d19h
lbpod1         1/1     Running            1          4d19h
lbpod2         1/1     Running            1          4d19h
mydb           0/1     CrashLoopBackOff   4          4m49s
mypod1         1/1     Running            2          5d1h
myrc1-54tmc    1/1     Running            2          5d
myrc1-lt6t6    1/1     Running            2          5d
myrc1-nkrrl    1/1     Running            2          5d
myrc1-zv8fl    1/1     Running            2          5d
```

Status=> CrashLoopBackOff
What is this??
It completely shows it failed. But not due to errors, it says that it needed some more info.
Why pod didn’t run??
May be image needs something and we are not providing that?
As it is created by some other developer and he might have set something that, hey program if you received a like from the viewer then treat him as a good user and allow him to use that pulled imaged which is created by me.
Let’s troubleshoot the problem.
First of all, let’s know the problem.
As, we have Downloaded image, Launched Container, started Container but failed. It must be asking something more from us.
There may be 2 possibilities:-

- May be image has internal bug.
- May be image needs something more?

But internal bug is not possible as this image is downloaded multiple times and used my millions of users. Means this image needs something more such as Credentials from us.

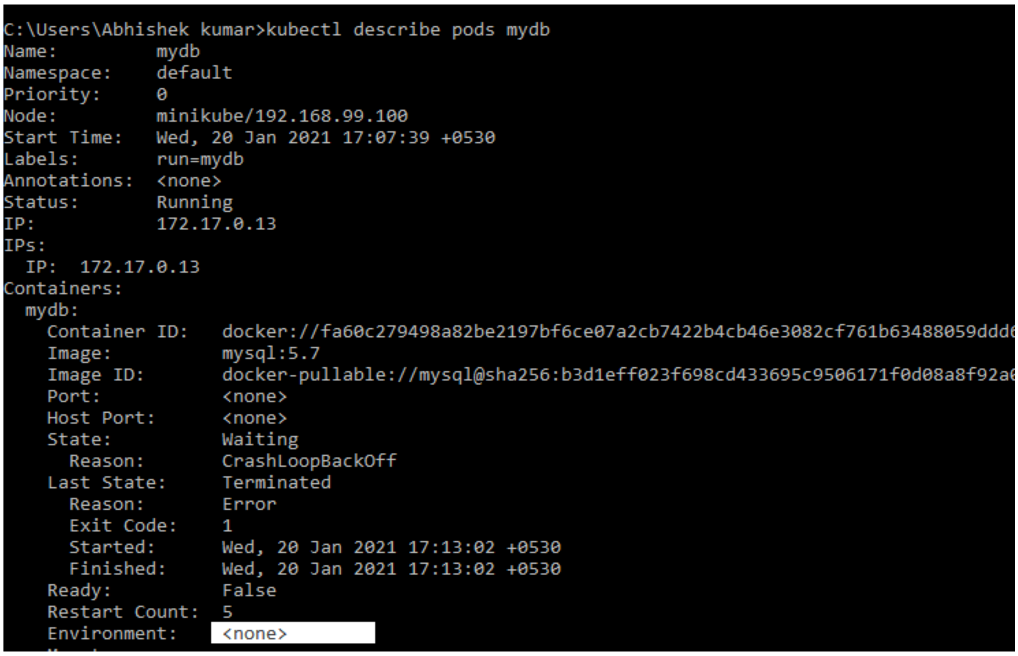
We have to go inside the pod and ask can you help me why my pod failed?
Programmers usually makes a program that runs when you launch the container.
So this program says as soon as you launch me [Normally in DB-> needed username and password] I need some credentials so that I can allow you to access to it.
To see error inside pod.
We will check logs of mydb
Kubectl logs mydb



```
C:\Users\Abhishek kumar>kubectl logs mydb
2021-01-20 11:43:02+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.33-1debian10 started.
2021-01-20 11:43:02+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2021-01-20 11:43:02+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.33-1debian10 started.
2021-01-20 11:43:02+00:00 [ERROR] [Entrypoint]: Database is uninitialized and password option is not specified
        You need to specify one of MYSQL_ROOT_PASSWORD, MYSQL_ALLOW_EMPTY_PASSWORD and MYSQL_RANDOM_ROOT_PASSWORD
```

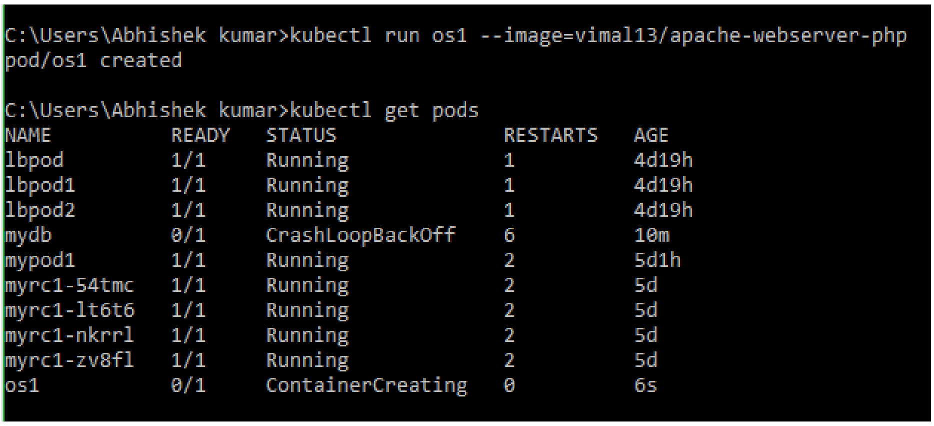
This says error because of container internal program.
Asking for username and password.
You can only access after providing information called Shell Variable.
And this only is called, **Environmental Variable.**

Use kubectl describe pods mydb to see some detailed info about Environmental Variables.



```
C:\Users\Abhishek kumar>kubectl describe pods mydb
Name:         mydb
Namespace:    default
Priority:     0
Node:         minikube/192.168.99.100
Start Time:   Wed, 20 Jan 2021 17:07:39 +0530
Labels:       run=mydb
Annotations:  <none>
Status:       Running
IP:           172.17.0.13
IPs:
  IP:  172.17.0.13
Containers:
  mydb:
    Container ID:   docker://fa60c279498a82be2197bf6ce07a2cb7422b4cb46e3082cf761b63488059ddd
    Image:          mysql:5.7
    Image ID:       docker-pullable://mysql@sha256:b3d1eff023f698cd433695c9506171f0d08a8f92a
    Port:           <none>
    Host Port:      <none>
    State:          Waiting
      Reason:       CrashLoopBackOff
    Last State:     Terminated
      Reason:       Error
      Exit Code:    1
      Started:      Wed, 20 Jan 2021 17:13:02 +0530
      Finished:     Wed, 20 Jan 2021 17:13:02 +0530
    Ready:          False
    Restart Count:  5
    Environment:    <none>
```

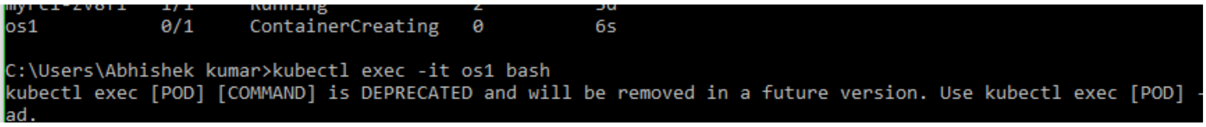
See here none ^ means it is showing nothing set for this pod.
Let's have a demo over docker images.
Kubectl run os1 - -image=vimal13/apache-webserver-php



```
C:\Users\Abhishek kumar>kubectl run os1 --image=vimal13/apache-webserver-php
pod/os1 created

C:\Users\Abhishek kumar>kubectl get pods
NAME           READY   STATUS              RESTARTS   AGE
lbpod          1/1     Running             1          4d19h
lbpod1         1/1     Running             1          4d19h
lbpod2         1/1     Running             1          4d19h
mydb           0/1     CrashLoopBackOff    6          10m
mypod1         1/1     Running             2          5d1h
myrc1-54tmc    1/1     Running             2          5d
myrc1-lt6t6    1/1     Running             2          5d
myrc1-nkrrl    1/1     Running             2          5d
myrc1-zv8fl    1/1     Running             2          5d
os1            0/1     ContainerCreating   0          6s
```

Container Creating. Now, let’s go inside that. Use

Kubectl exec –it os1 bash

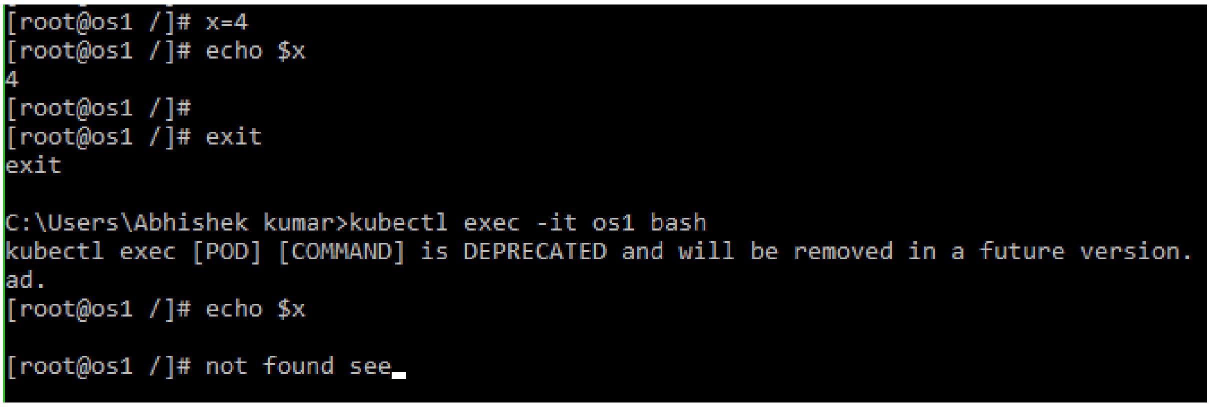


```
myrc1-zv8r1   1/1     Running             2          3d
os1           0/1     ContainerCreating   0          6s

C:\Users\Abhishek kumar>kubectl exec -it os1 bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -
ad.
```

Now, see when I am inside that container, I defined variable x=4 and after that when I use echo $x shows the value 4.

But after that when I exit and when came back to container os1

When I search for value of x. It disappeared as it is saved as temporary and till the time of OS is running.

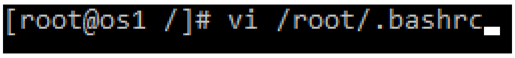


```
[root@os1 /]# x=4
[root@os1 /]# echo $x
4
[root@os1 /]#
[root@os1 /]# exit
exit

C:\Users\Abhishek kumar>kubectl exec -it os1 bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version.
ad.
[root@os1 /]# echo $x

[root@os1 /]# not found see
```

To save permanently even after exited or shut down or failure.

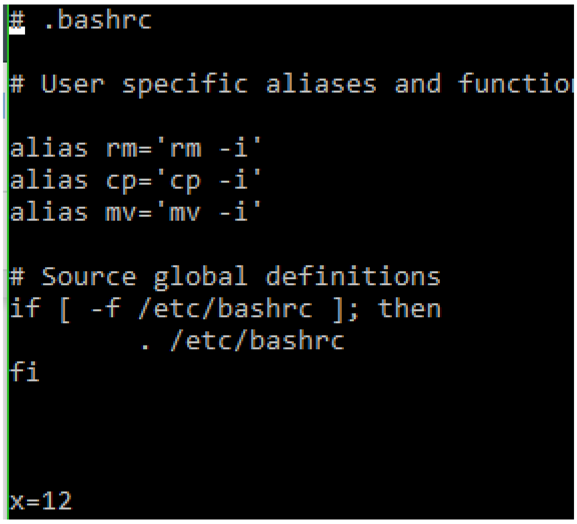
We need to use .bashrc

Vi /root/.bashrc



```
[root@os1 /]# vi /root/.bashrc
```

Inside that write down the value of x(here), we can set any value permanently here.

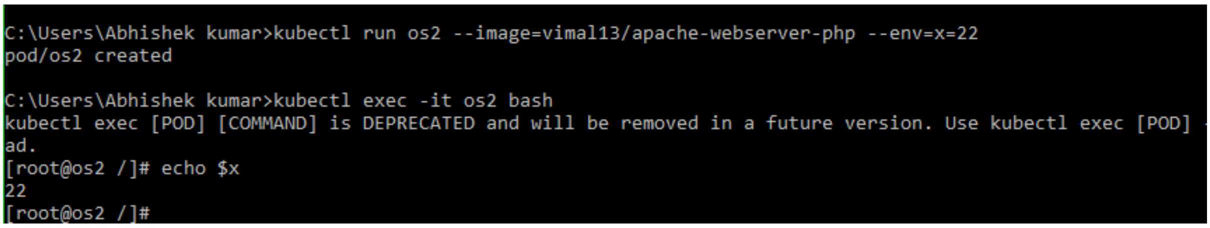


```
# .bashrc

# User specific aliases and functio

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi



x=12
```

Here, I set x=12.

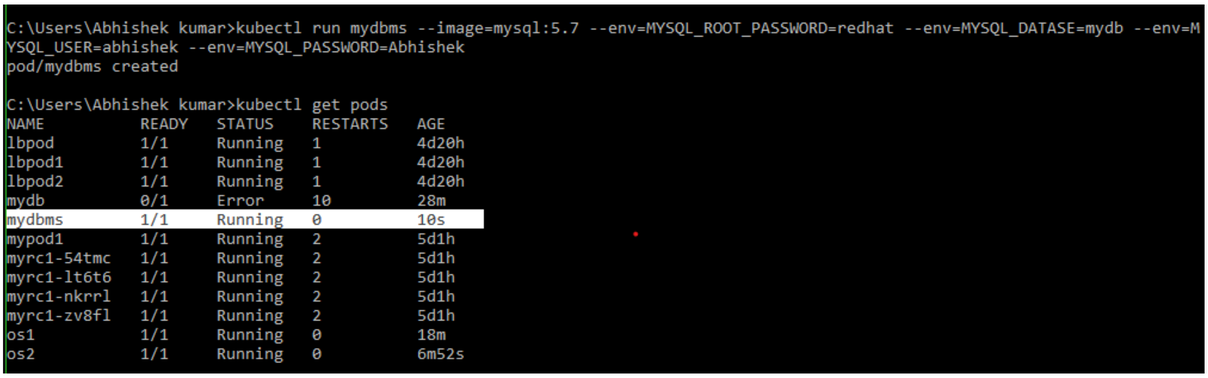
But this time even after exiting when I am asking for x it will show 12.

Similarly same thing we can do by cli.



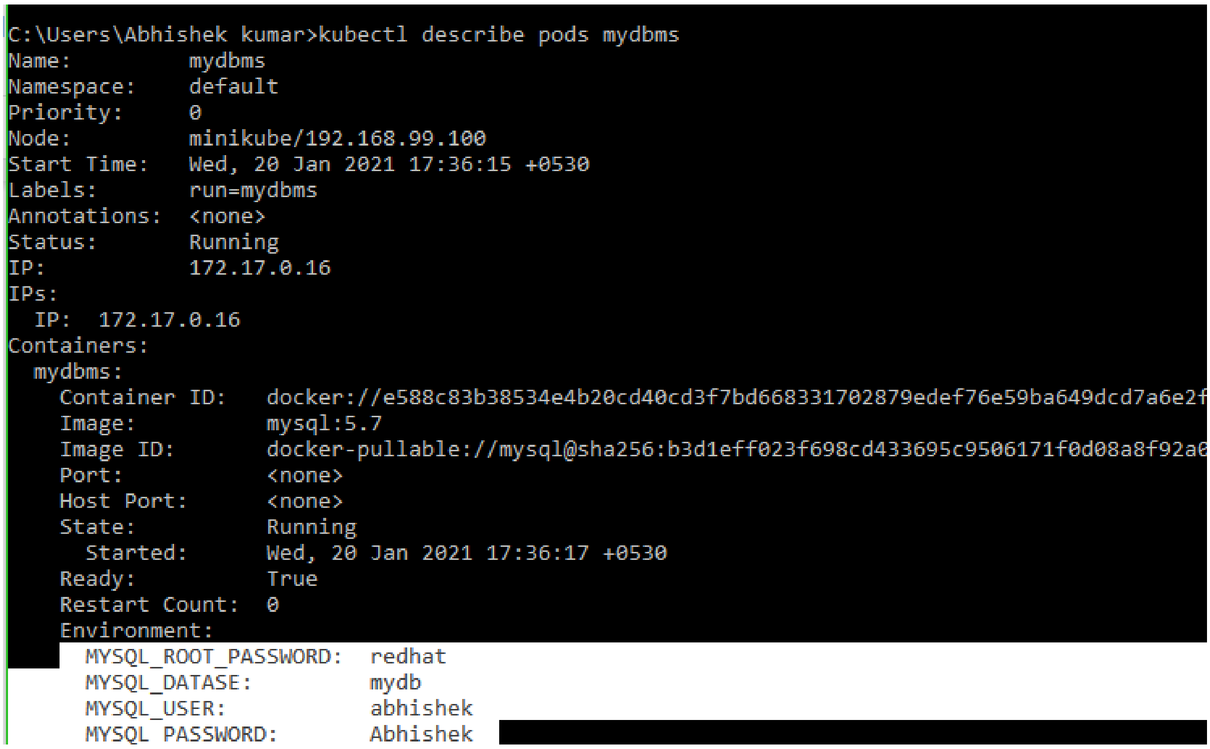
```
C:\Users\Abhishek kumar>kubectl run os2 --image=vimal13/apache-webserver-php --env=x=22
pod/os2 created

C:\Users\Abhishek kumar>kubectl exec -it os2 bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -
ad.
[root@os2 /]# echo $x
22
[root@os2 /]#
```

--env=x=12 sets it as permanent file. See in the above picture.
So from this demo,
We can conclude that using ./bashrc system we can save our environmental variables that will be saved permanently.
So, I used
Kubectl run mydbms –image=MYSQL_ROOT_PASSWORD=redhat –env=MYSQL_DATABASE=mydb –env=MYSQL_USER=Abhishek –env=MYSQL_PASSWORD=Abhishek

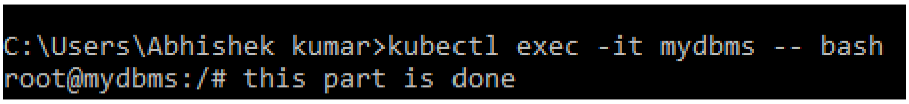


```
C:\Users\Abhishek kumar>kubectl run mydbms --image=mysql:5.7 --env=MYSQL_ROOT_PASSWORD=redhat --env=MYSQL_DATASE=mydb --env=M
YSQL_USER=abhishek --env=MYSQL_PASSWORD=Abhishek
pod/mydbms created

C:\Users\Abhishek kumar>kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
lbpod         1/1     Running   1          4d20h
lbpod1        1/1     Running   1          4d20h
lbpod2        1/1     Running   1          4d20h
mydb          0/1     Error     10         28m
mydbms        1/1     Running   0          10s
mypod1        1/1     Running   2          5d1h
myrc1-54tmc   1/1     Running   2          5d1h
myrc1-lt6t6   1/1     Running   2          5d1h
myrc1-nkrrl   1/1     Running   2          5d1h
myrc1-zv8fl   1/1     Running   2          5d1h
os1           1/1     Running   0          18m
os2           1/1     Running   0          6m52s
```

Now, see it’s running.
Kubectl describe pods mydbms



```
C:\Users\Abhishek kumar>kubectl describe pods mydbms
Name:         mydbms
Namespace:    default
Priority:     0
Node:         minikube/192.168.99.100
Start Time:   Wed, 20 Jan 2021 17:36:15 +0530
Labels:       run=mydbms
Annotations:  <none>
Status:       Running
IP:           172.17.0.16
IPs:
  IP:  172.17.0.16
Containers:
  mydbms:
    Container ID:   docker://e588c83b38534e4b20cd40cd3f7bd668331702879edef76e59ba649dcd7a6e2f
    Image:          mysql:5.7
    Image ID:       docker-pullable://mysql@sha256:b3d1eff023f698cd433695c9506171f0d08a8f92a6
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Wed, 20 Jan 2021 17:36:17 +0530
    Ready:          True
    Restart Count:  0
    Environment:
      MYSQL_ROOT_PASSWORD:  redhat
      MYSQL_DATASE:         mydb
      MYSQL_USER:           abhishek
      MYSQL_PASSWORD:       Abhishek
```



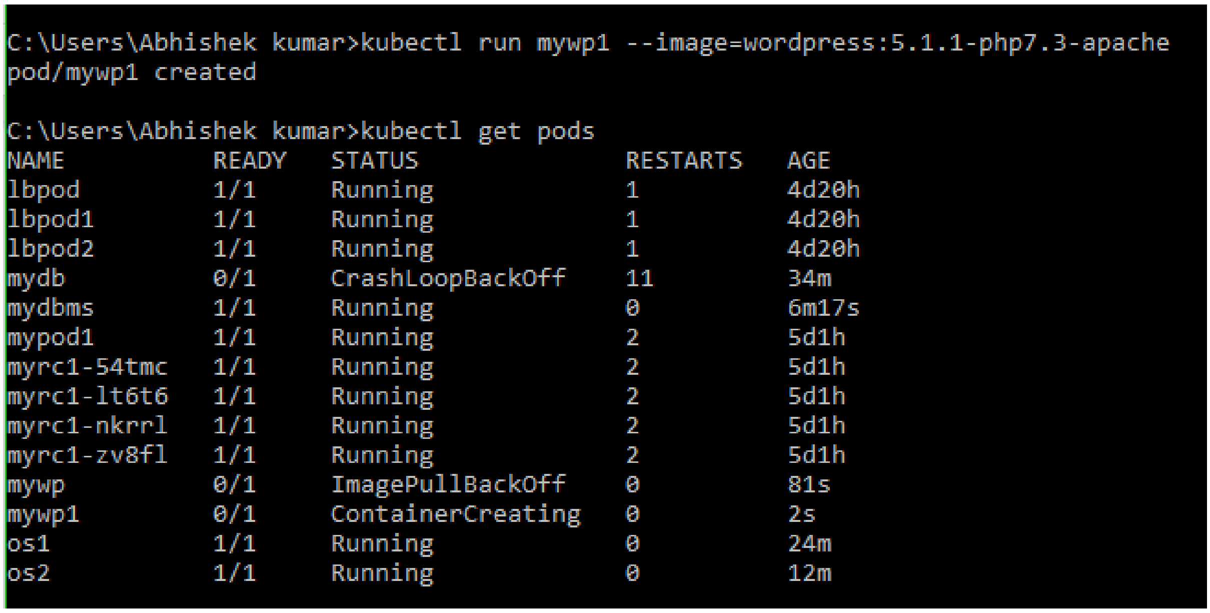
```
C:\Users\Abhishek kumar>kubectl exec -it mydbms -- bash
root@mydbms:/# this part is done
```

See, the credentials are set now and set permanently in that container.
Now, our backend part is done after this. All databases are handled by MYSQL here.
Now, it’s turn for Front End.

Front end Development

Directly use php apche webserver.

Kubectl run mywp1 –iamge=wordpress:5.1.1-php7.3-apache

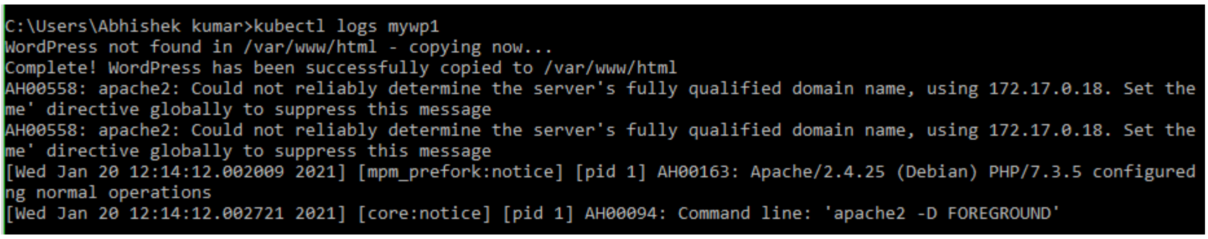


```
C:\Users\Abhishek kumar>kubectl run mywp1 --image=wordpress:5.1.1-php7.3-apache
pod/mywp1 created

C:\Users\Abhishek kumar>kubectl get pods
NAME           READY   STATUS              RESTARTS   AGE
lbpod          1/1     Running             1          4d20h
lbpod1         1/1     Running             1          4d20h
lbpod2         1/1     Running             1          4d20h
mydb           0/1     CrashLoopBackOff    11         34m
mydbms         1/1     Running             0          6m17s
mypod1         1/1     Running             2          5d1h
myrc1-54tmc    1/1     Running             2          5d1h
myrc1-lt6t6    1/1     Running             2          5d1h
myrc1-nkrrl    1/1     Running             2          5d1h
myrc1-zv8fl    1/1     Running             2          5d1h
mywp           0/1     ImagePullBackOff    0          81s
mywp1          0/1     ContainerCreating   0          2s
os1            1/1     Running             0          24m
os2            1/1     Running             0          12m
```

See the Third last line ^ shows that container Creating.

Check logs by

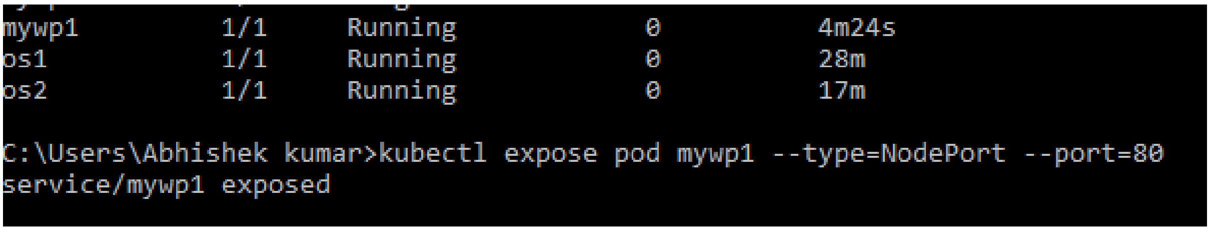
Kubectl logs mywp1



```
C:\Users\Abhishek kumar>kubectl logs mywp1
WordPress not found in /var/www/html - copying now...
Complete! WordPress has been successfully copied to /var/www/html
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.18. Set the
me' directive globally to suppress this message
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.18. Set the
me' directive globally to suppress this message
[Wed Jan 20 12:14:12.002009 2021] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.25 (Debian) PHP/7.3.5 configured
ng normal operations
[Wed Jan 20 12:14:12.002721 2021] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
```

Now, it is all set, but we need to expose it to the public. Like we use to do in the day 5, where we learnt that how we set the 2 Load balancer called NodePort so that the conatiners inside minikube can be accessed by the public.

Kubectl expose pod mywp1 –type=NodePort –port=80

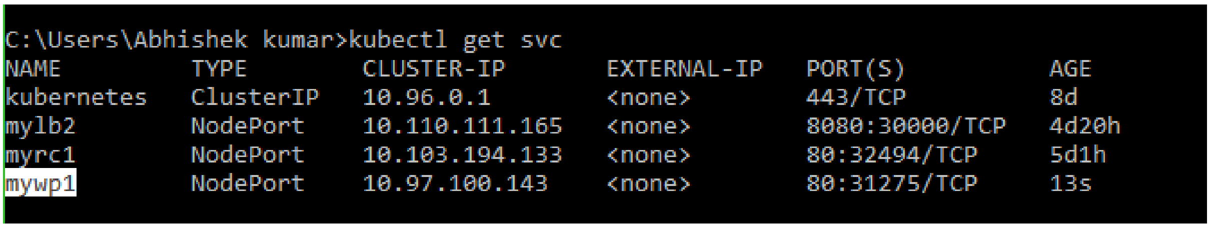


```
mywp1          1/1     Running             0          4m24s
os1            1/1     Running             0          28m
os2            1/1     Running             0          17m

C:\Users\Abhishek kumar>kubectl expose pod mywp1 --type=NodePort --port=80
service/mywp1 exposed
```

Now, it is exposed to public.

Let’s see that

Run kubectl get svc



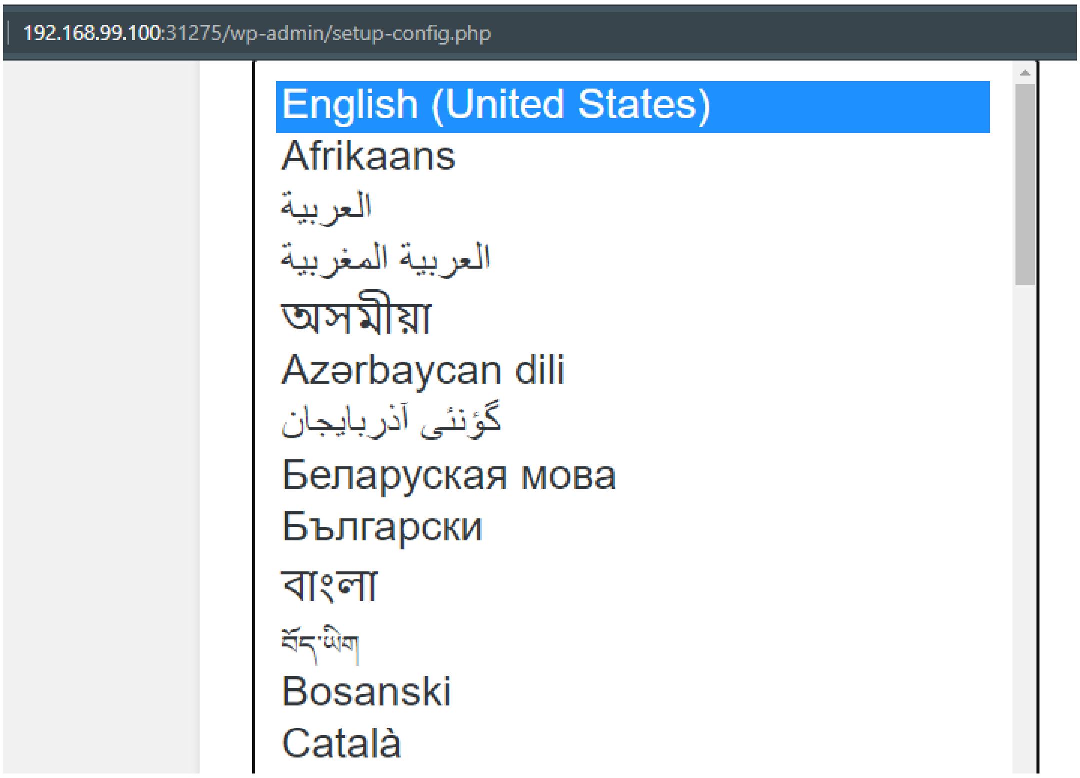
```
C:\Users\Abhishek kumar>kubectl get svc
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP          8d
mylb2        NodePort    10.110.111.165   <none>        8080:30000/TCP   4d20h
myrc1        NodePort    10.103.194.133   <none>        80:32494/TCP     5d1h
mywp1        NodePort    10.97.100.143    <none>        80:31275/TCP     13s
```

Note down the port number of mywp1 => **31275**

Now we know the IP of minikube. In my case it is

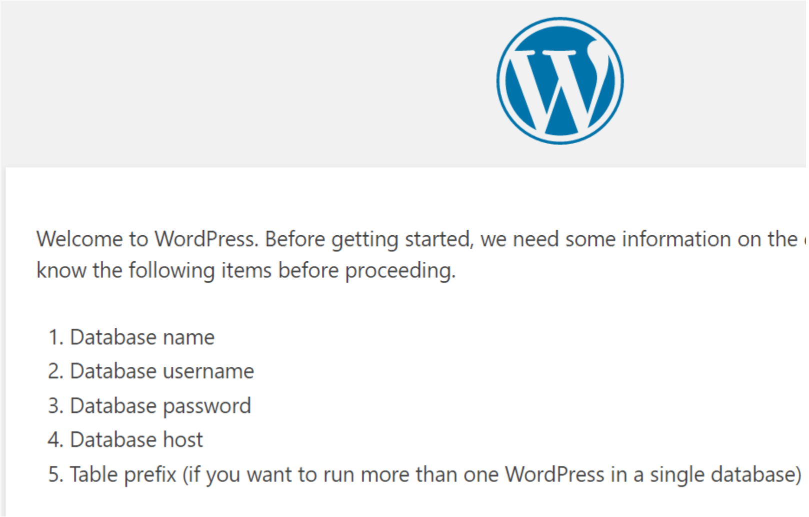
192.168.99.100

If you don't know then open vm
With username: docker
And password : tcuser
Use ifconfig | less and note down the IP.
Using IP and port like IP:port => browse it.
192.168.99.100:31275



192.168.99.100:31275/wp-admin/setup-config.php

Next

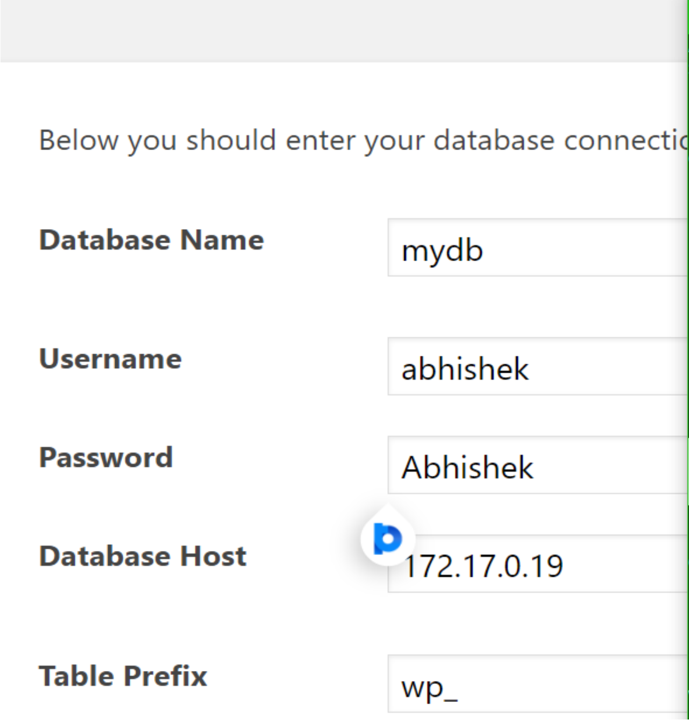


Welcome to WordPress. Before getting started, we need some information on the
know the following items before proceeding.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

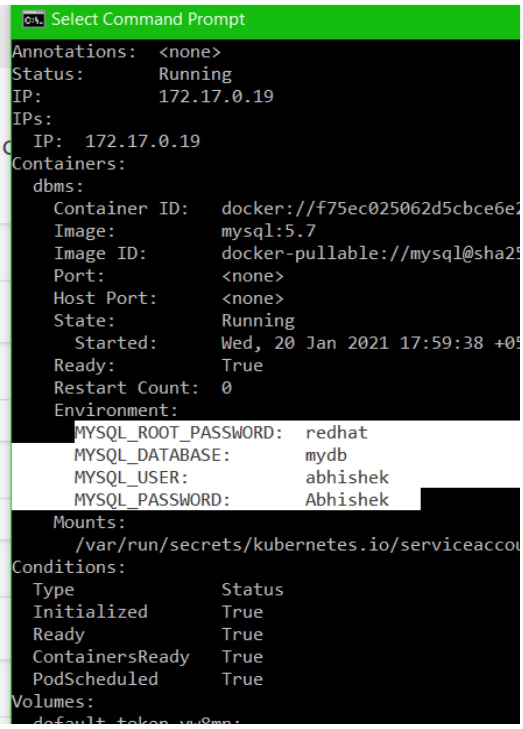
Again next, you will arrive to the page where you will assign the Front end to the backend.
So, let's first check the IP of the Databse pod which can be used as back end.
Kubectl describe pods mydb

After that ip 172.17.0.19
Give that in the credentials of Database Host.
Create username and Password
Also use table prefix which will be added in front of the database table.
Here, I used wp_



Below you should enter your database connectio

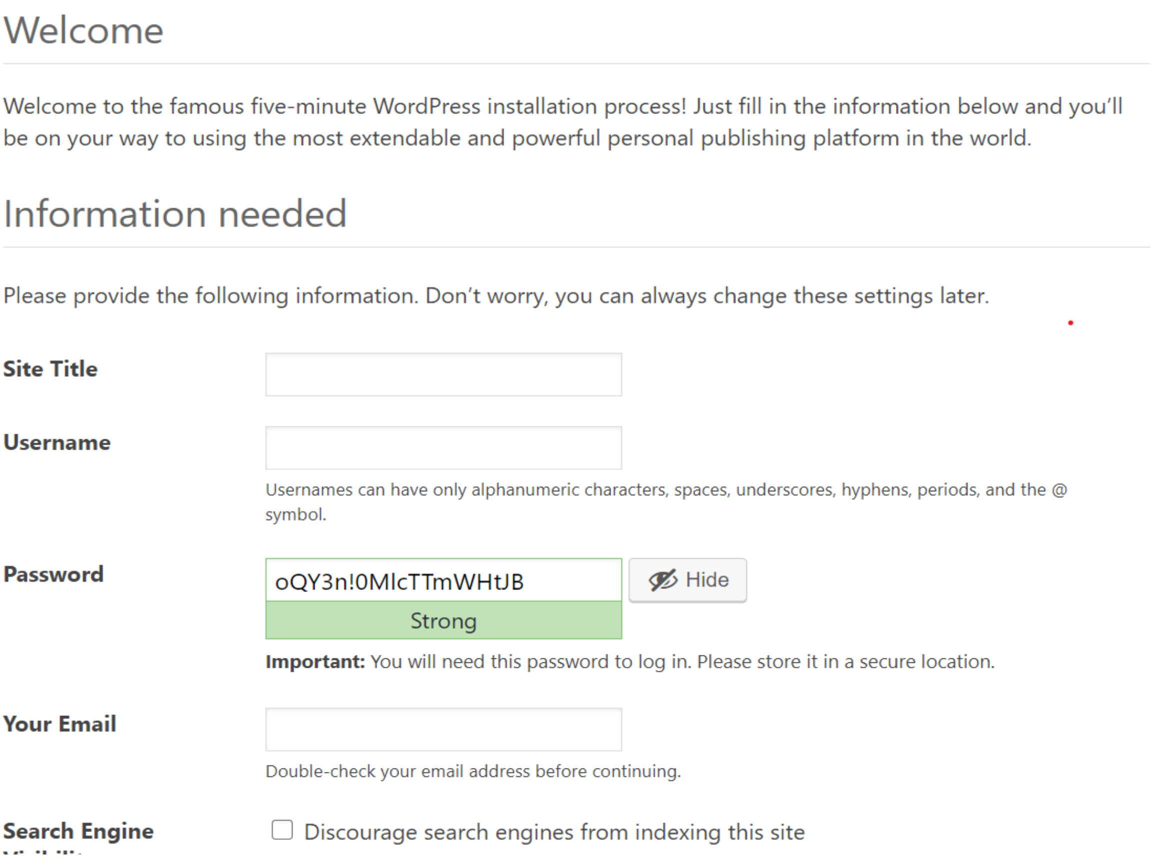
Database Name	mydb
Username	abhishek
Password	Abhishek
Database Host	172.17.0.19
Table Prefix	wp_



Select Command Prompt

```
Annotations:  <none>
Status:       Running
IP:           172.17.0.19
IPs:
  IP:  172.17.0.19
Containers:
  dbms:
    Container ID:   docker://f75ec025062d5cbce6e
    Image:          mysql:5.7
    Image ID:       docker-pullable://mysql@sha2
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Wed, 20 Jan 2021 17:59:38 +0
    Ready:          True
    Restart Count:  0
    Environment:
      MYSQL_ROOT_PASSWORD:  redhat
      MYSQL_DATABASE:       mydb
      MYSQL_USER:           abhishek
      MYSQL_PASSWORD:       Abhishek
    Mounts:
      /var/run/secrets/kubernetes.io/serviceacco
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  default-token-vw8mn:
```

After this a next page will come like this.



Welcome

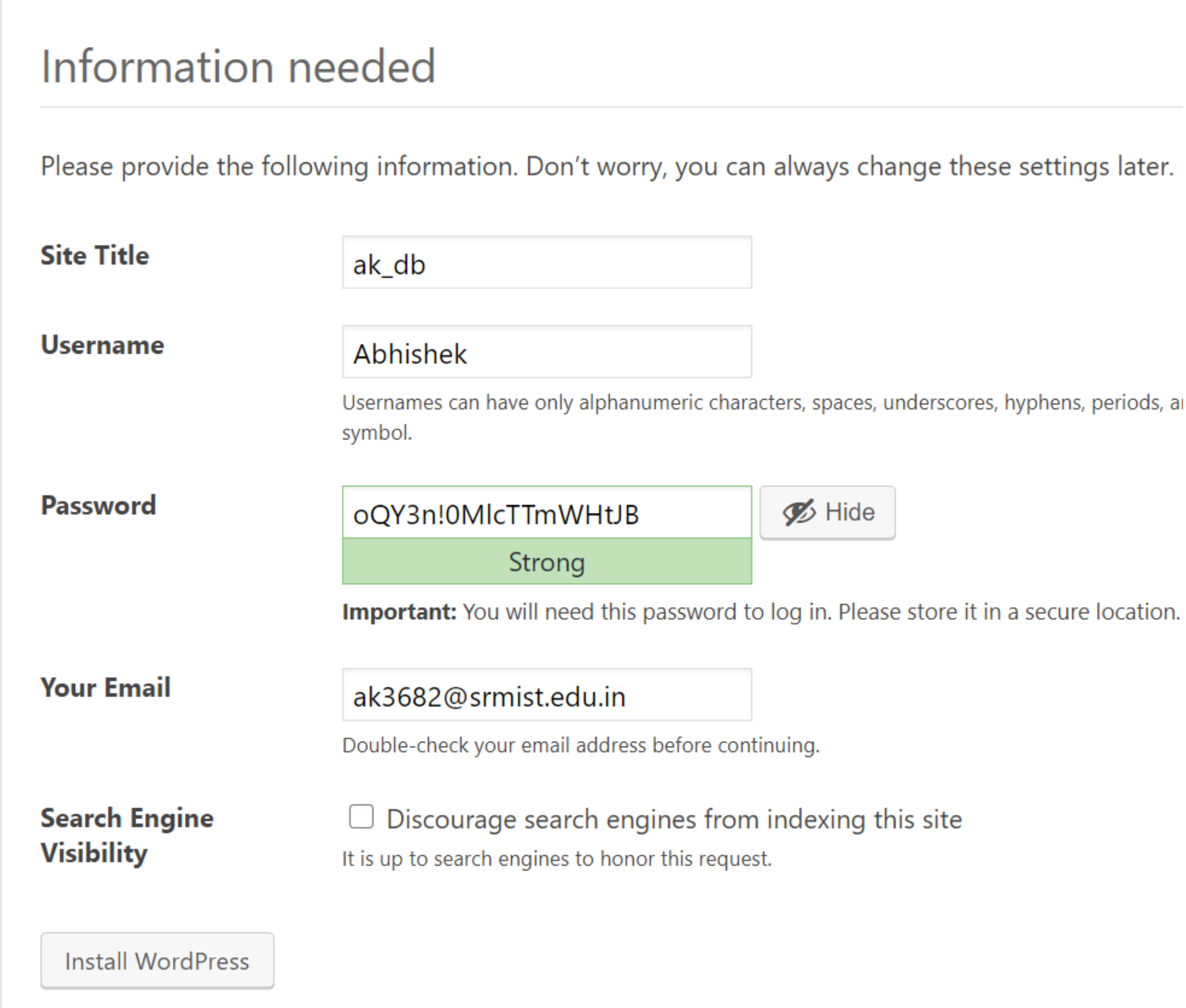
Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title	
Username	Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.
Password	oQY3n!0MlcTTmWHtJB Strong **Important:** You will need this password to log in. Please store it in a secure location.
Your Email	Double-check your email address before continuing.
Search Engine Visibility	□ Discourage search engines from indexing this site

Fill the Information's needed as a Admin of these pods.



Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title ak_db

Username Abhishek

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, a
symbol.

Password oQY3n!0MlcTTmWHtJB

Important: You will need this password to log in. Please store it in a secure location.

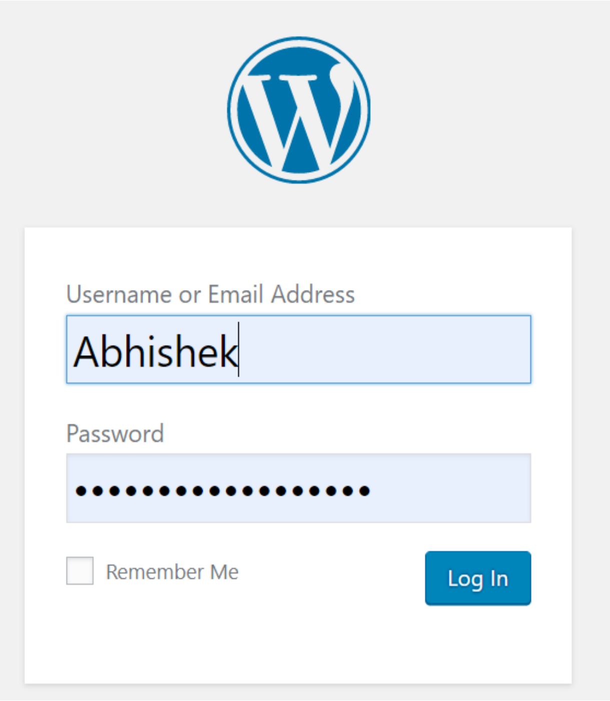
Your Email ak3682@srmist.edu.in

Double-check your email address before continuing.

Search Engine Visibility □ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Click on Install WordPress and boom.
You are all set. Login back to the WordPress.

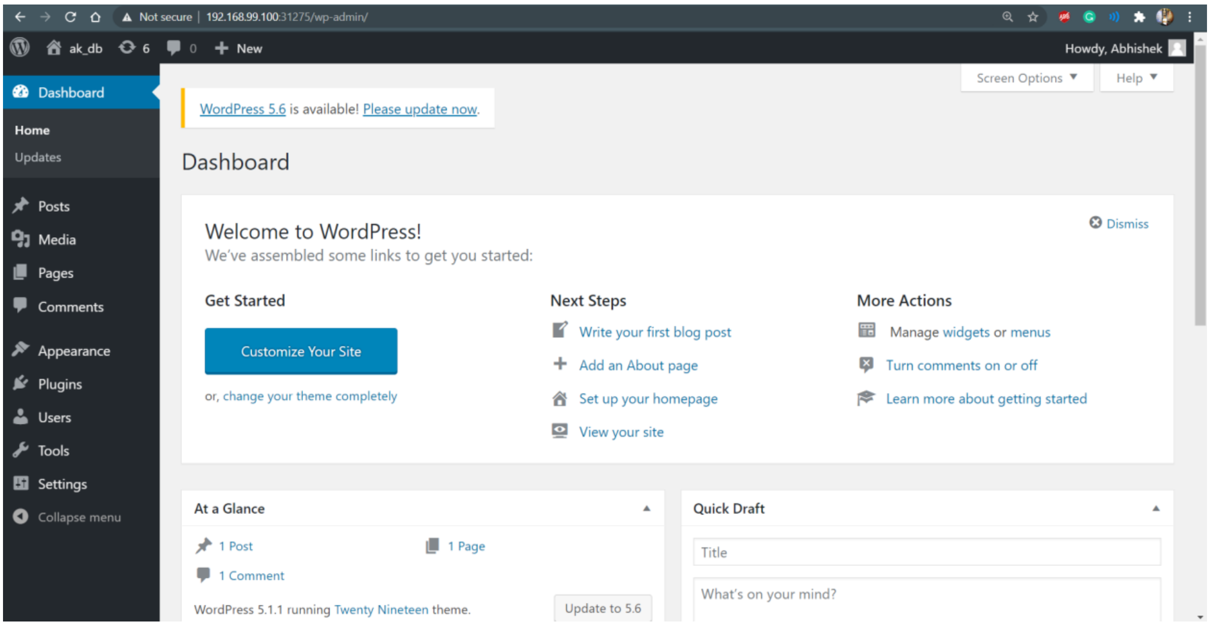


Username or Email Address: Abhishek

Password: ••••••••••••••••••

□ Remember Me

Front end will come with complete Dashboard and attached with a Database.



Not secure | 192.168.99.100:31275/wp-admin/

ak_db 6 0 + New Howdy, Abhishek

Dashboard
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu

Screen Options ▼ Help ▼

WordPress 5.6 is available! Please update now.

Dashboard

Welcome to WordPress!
We've assembled some links to get you started:

Dismiss

Get Started
Customize Your Site
or, change your theme completely

Next Steps
Write your first blog post
Add an About page
Set up your homepage
View your site

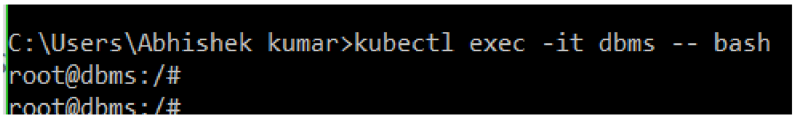
More Actions
Manage widgets or menus
Turn comments on or off
Learn more about getting started

At a Glance
1 Post
1 Page
1 Comment
WordPress 5.1.1 running Twenty Nineteen theme.
Update to 5.6

Quick Draft
Title
What's on your mind?

Now, one last thing, we need to check the table in the database pod. So come inside that container and check.

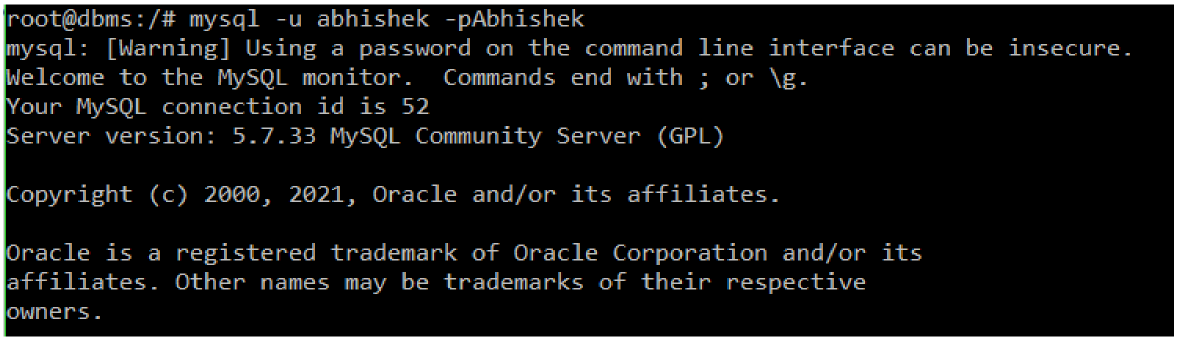
Kubectl exec –it dbms --bash



```
C:\Users\Abhishek kumar>kubectl exec -it dbms -- bash
root@dbms:/#
root@dbms:/#
```

Mysql –u Abhishek –pAbhishek

These are my username and passoword which I set.



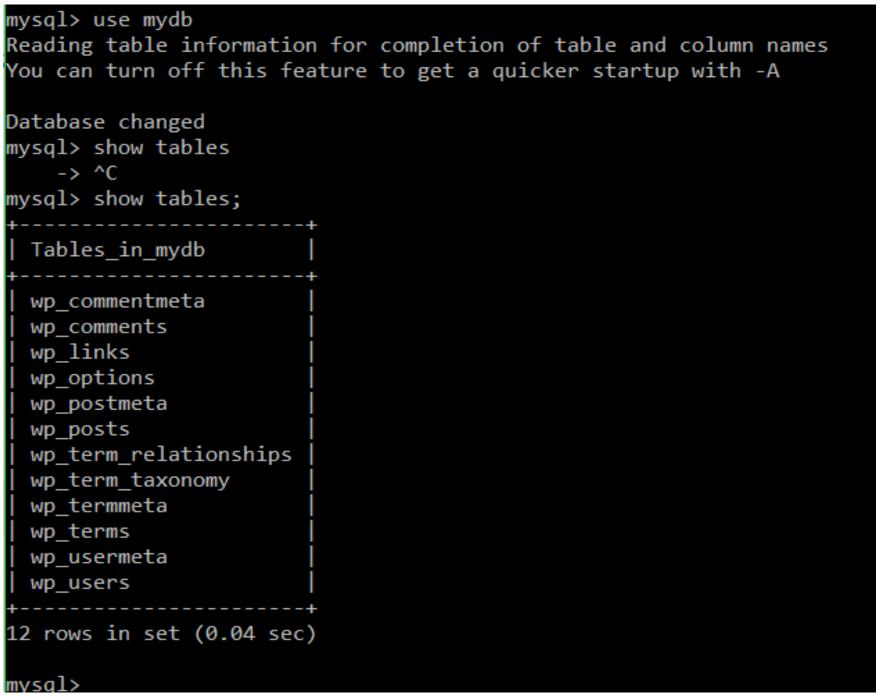
```
root@dbms:/# mysql -u abhishek -pAbhishek
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 52
Server version: 5.7.33 MySQL Community Server (GPL)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

Now, I am inside that container.

Use mydb

Show tables;

See, in the table database,
The records have prefix
Wp_ which I set earlier.



```
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
    -> ^C
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 (0.04 sec)

mysql>
```