Create EC2 instance Ubuntu :

Enable security as per your requirement or just enable for all traffic and says from anywhere.

Login with Ubuntu

Login with root --- sudo –i

Installing Java:

sudo add-apt-repository ppa:openjdk-r/ppa

sudo apt-get update

sudo apt-get install openjdk-8-jdk

Installing Maven :

sudo apt-get install maven

sudo apt-get install git

git clone https://github.com/sivakethineni/CI-CD-project.git

cd CI-CD-project

git branch –a

git checkout vp-docker

ls

mvn install

ls

cd target

ls

cd ..

cp target/vprofile-v1.war docker-web

cd docker-web

ls

vi Dockerfile

```
root@ip-172-31-2-217:~/CI-CD-project# cd Docker-web
root@ip-172-31-2-217:~/CI-CD-project/Docker-web# cat Dockerfile
FROM tomcat:8-jre8

RUN rm -rf /usr/local/tomcat/webapps/*

COPY vprofile-v1.war /usr/local/tomcat/webapps/ROOT.war

EXPOSE 8080
CMD ["catalina.sh", "run"]

root@ip-172-31-2-217:~/CI-CD-project/Docker-web#
```

Just remove target before vprofile-v1.war

:wq

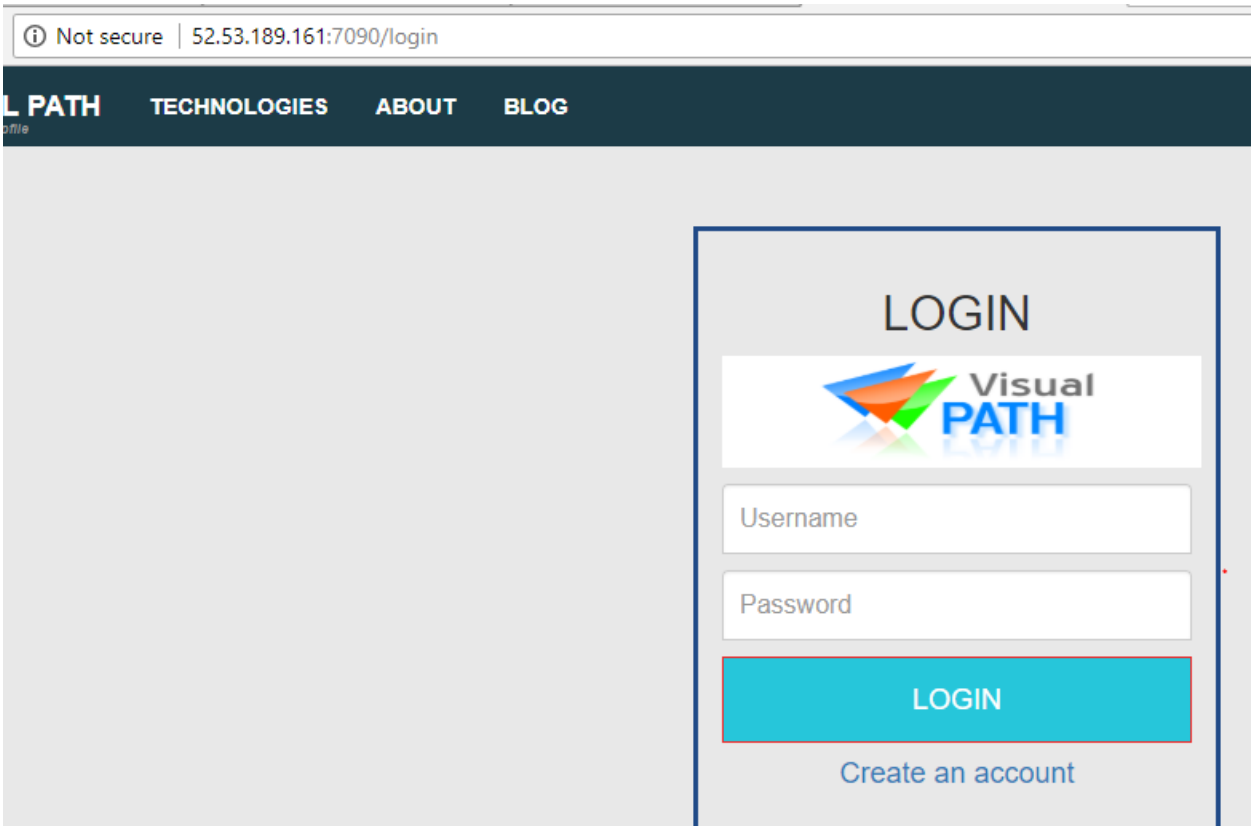
docker build –t goldentech/vprofileapp:v1 .

docker images

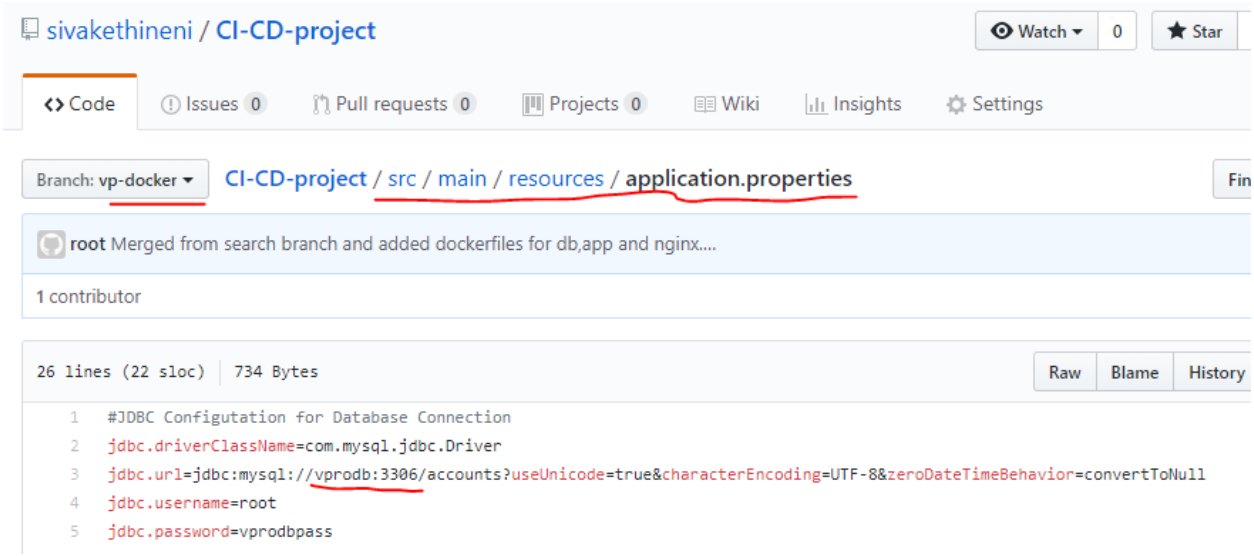
docker run –d –name visualpathapp –p 7090:8080 goldentech/vprofileapp:v1

docker ps

goto browser type <ipaddress:7090>



cd Docker-db



docker build -t goldentech/visualdbvpro:v1 .

docker images

docker run -d --name vprodb -p 3306:3306 -e MYSQL_ROOT_PASSWORD=vprodbpass goldentech/visualdbvpro:v1

docker exec –it <containerid> /bin/bash

mysql –u root –p

show databases;

use accounts;

show tables;

quit

exit

docker ps

ls

docker run -d --name vprodb -p 3306:3306 -e MYSQL_ROOT_PASSWORD=vprodbpass goldentech/visualdbvpro:v1

docker ps

My DB is ready now

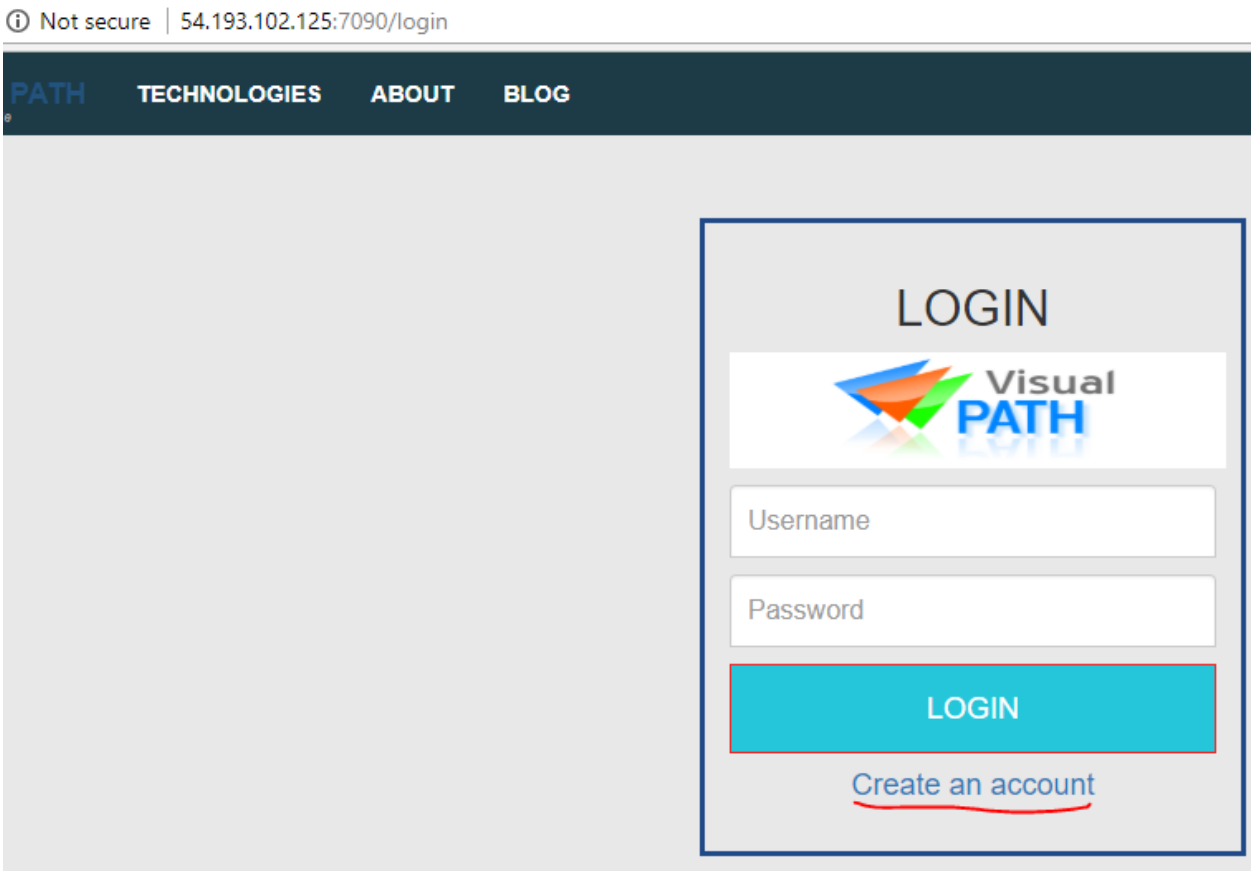
We have to link application with database now

docker run –name vprofileapp –link vprodb:mysql –d –p 7070:8080 goldentech/ vprofileapp:v1

docker ps

```
root@ip-172-31-2-217:~/CI-CD-project/Docker-db# docker run --name vprofileapp --link vprodb:mysql -d -p
 7070:8080 goldentech/vprofileapp:v1
d7e07f1d441348392f864bf48e0e31c3d39799dc0cd987e45a6428efe048e107
root@ip-172-31-2-217:~/CI-CD-project/Docker-db# docker ps
CONTAINER ID        IMAGE                              COMMAND                  CREATED
    STATUS              PORTS                      NAMES
d7e07f1d4413        goldentech/vprofileapp:v1          "catalina.sh run"        6 seconds ago
    Up 4 seconds        0.0.0.0:7070->8080/tcp     vprofileapp
```

```
root@ip-172-31-41-156:~/VProfile/Docker-db# docker ps
CONTAINER ID        IMAGE                      COMMAND                  CREATED             STATUS              PORTS                    NAMES
b9396f135ed5        visualpath/visualdbvpro:V1   "docker-entrypoint.s…"   About a minute ago   Up About a minute   0.0.0.0:3306->3306/tcp   vprodb
root@ip-172-31-41-156:~/VProfile/Docker-db# docker run --name vprodb -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=vprodbpass visualpath/visualdbvpro:V1^C
root@ip-172-31-41-156:~/VProfile/Docker-db# docker ps
CONTAINER ID        IMAGE                      COMMAND                  CREATED             STATUS              PORTS                    NAMES
b9396f135ed5        visualpath/visualdbvpro:V1   "docker-entrypoint.s…"   3 minutes ago       Up 3 minutes        0.0.0.0:3306->3306/tcp   vprodb
root@ip-172-31-41-156:~/VProfile/Docker-db# docker run --name vprofileapp --link vprodb:mysql -d -p 7070:8080 visualpath/visualvprofile:v1
```



Create an account and try to login with those credentials that’s it we successfully deployed our project on containers.

Second use case :

It is a ruby application which is available on public repository

git clone https://github.com/jpetazzo/namer

ls

cd namer

ls

docker run –d –v $(pwd):/src –p 7080:9292 jpetazzo/namer

$(pwd) ---- Means it is pointing to a current working directory.

Now go to browser and type <ipaddress:7080>

ls

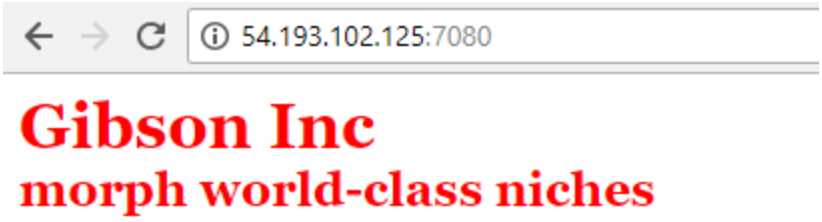
vi company_name_generator.rb

Here we are going to test our volumes here I just mapped with my working directory with /src directory in container so we can do some modifications here.

```
root@ip-172-31-2-217:~/namer# cat company_name_generator.rb
class CompanyNameGenerator < Sinatra::Base
  register Sinatra::Reloader

  get '/' do
    name = Faker::Company.name
    bs = Faker::Company.bs
    <<-HTML
      <html>
        <style>
          h1, h2 {
            font-family: Georgia, Times New Roman, Times, serif;
            color: red;
            margin: 0;
          }
        </style>
        <title>Company name generator</title>
        <body>
```

Red is replaced with royalblue colour and go to browser and just refresh it



Became red colour means files are in sync.

docker ps

docker exec -it <container id> /bin/bash

```
root@ip-172-31-2-217:~/namer# docker exec -it bfb59aa4e942 /bin/bash
root@bfb59aa4e942:/src# ls
Dockerfile  Gemfile.lock  company_name_generator.rb  docker-compose.yml
Gemfile     README.md     config.ru
root@bfb59aa4e942:/src#
```

exit

docker stop <container id>

docker rm <container id>

docker run –name newrubyapp –d –v $(pwd):/src –p 7080:9292 jpetazzo/namer

Here out current working directory changes what we done those changes again map with new container so that text red colour means we just shared same volume with multiple containers that’s the advantage of volumes.