**Containerized v-profile web application in multistage docker compose**

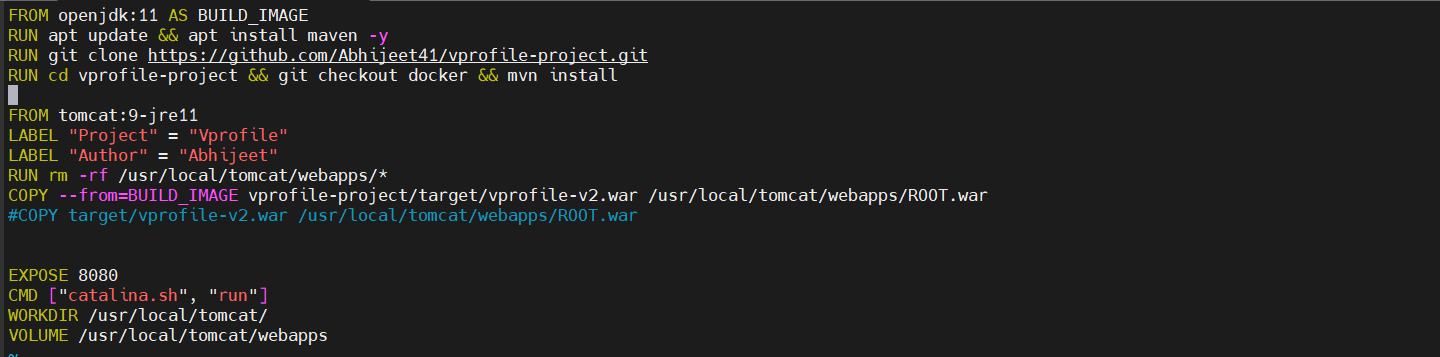
First clone vprofile project from git using below command

git clone <https://github.com/Abhijeet41/vprofile-project.git> -b containers

now navigate to dockerfiles and write docker command for app

cd /Docker-files/app

vim Dockerfile



Note:

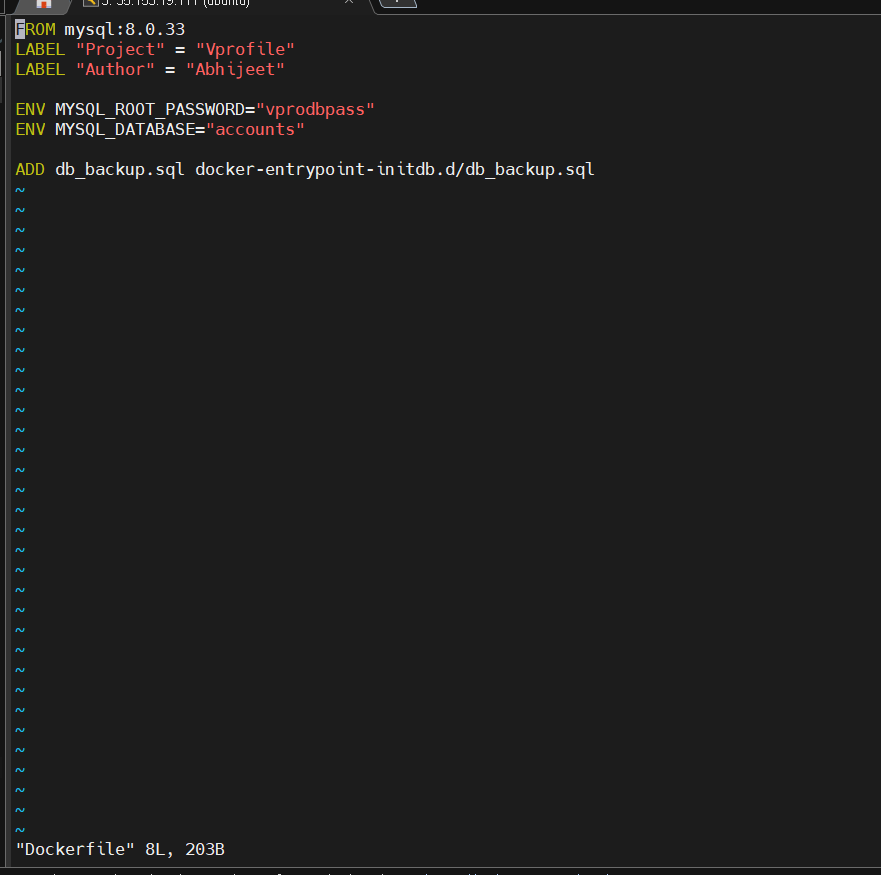
In this OpenJdk Image first we install open jdk and install app update then we clone source code from docker branch and using cd we navigate to vprofile-project and run mvn install so basically in this image we are generating artifact

In this TomCat Image you will be having default artifacte so far we need to remove it then copy .war file from above image path location. Expose port 8080 because tomcat run process on port

8080.

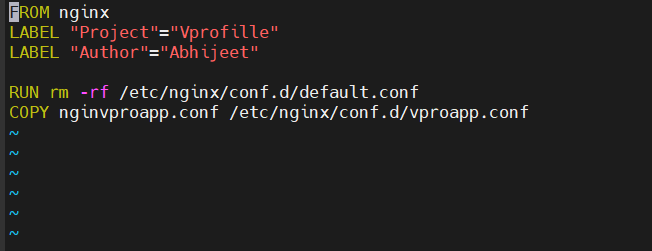
Setup Database

Now navigate to db folder and write docker commands to create mysql database



Now create Nginx image for web

Navigate to the web folder and edit Dockerfile

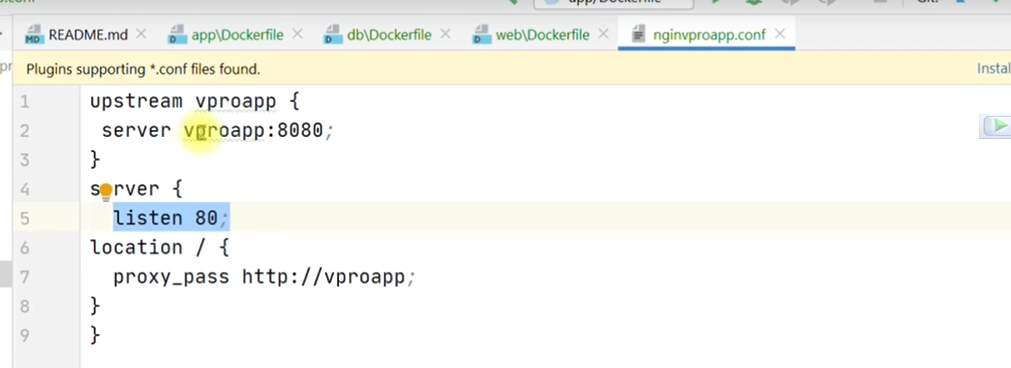


From nginx is default image name

give label as project and auther its optional

Note

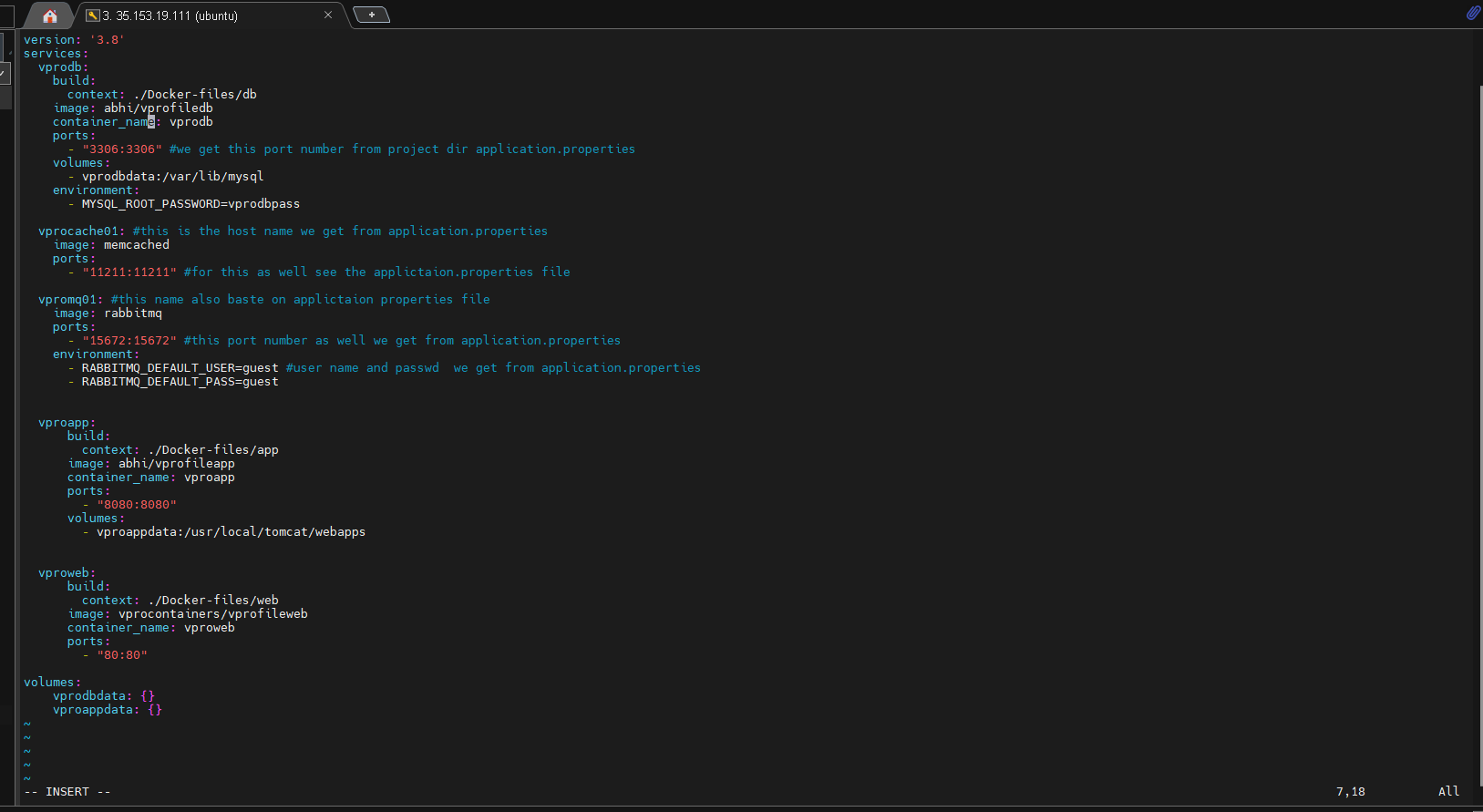
We are building our own conf file i.e nginvproapp.conf



remove existing default.conf file then copy nginvproapp.confg // this info we will get from

official nginx image read the instruction carefully.

Now create and edit docker compose file



Above docker compose file we create from official docker compose references and the port we are using in this project itself application.properies

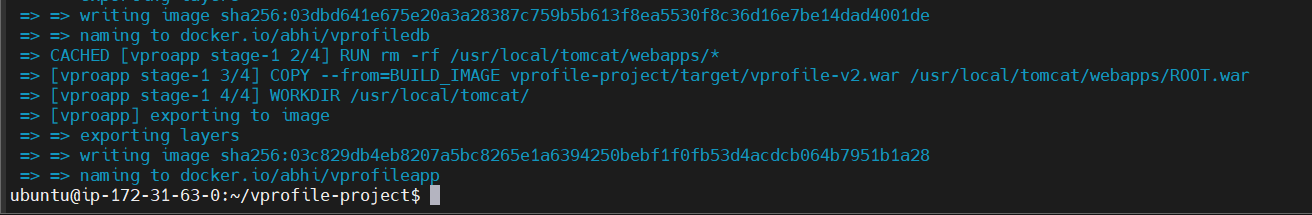
We are also getting environment username and password from application.properties

**Now Build & Run Our Docker compose file using below command**

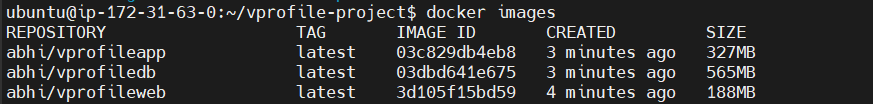
First navigate to directory were compose.yaml file is present and type this command

docker compose build

o/p:- successfully build our image

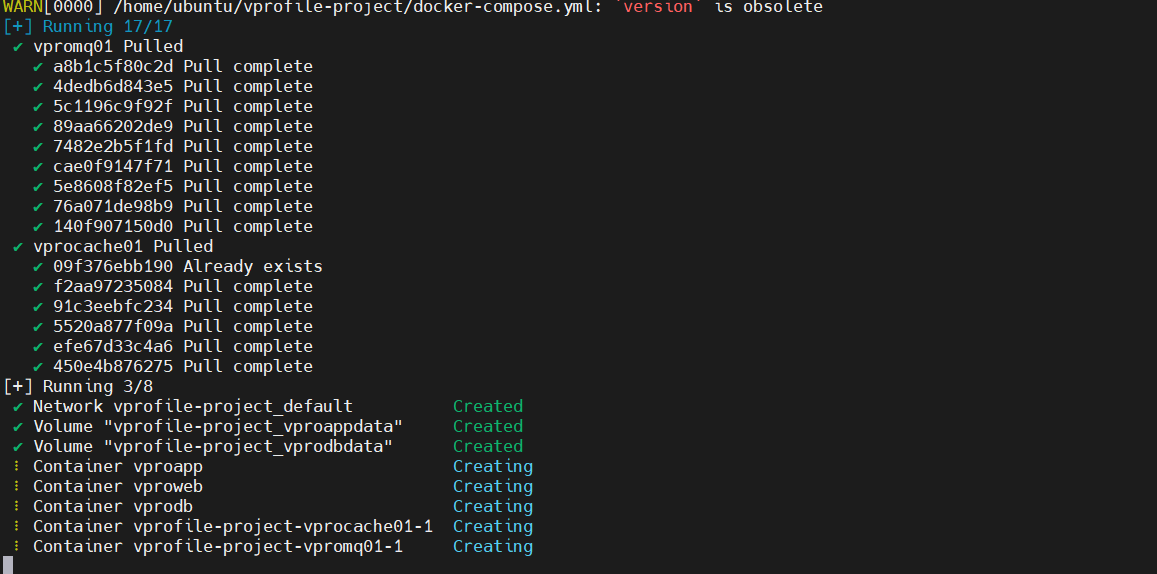


Make sure this 3 images get created



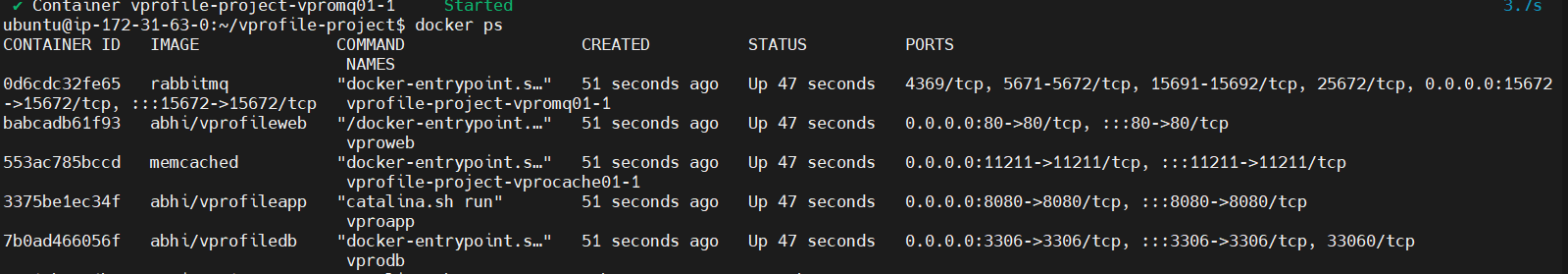
To start docker compose container use below command

docker compose up -d



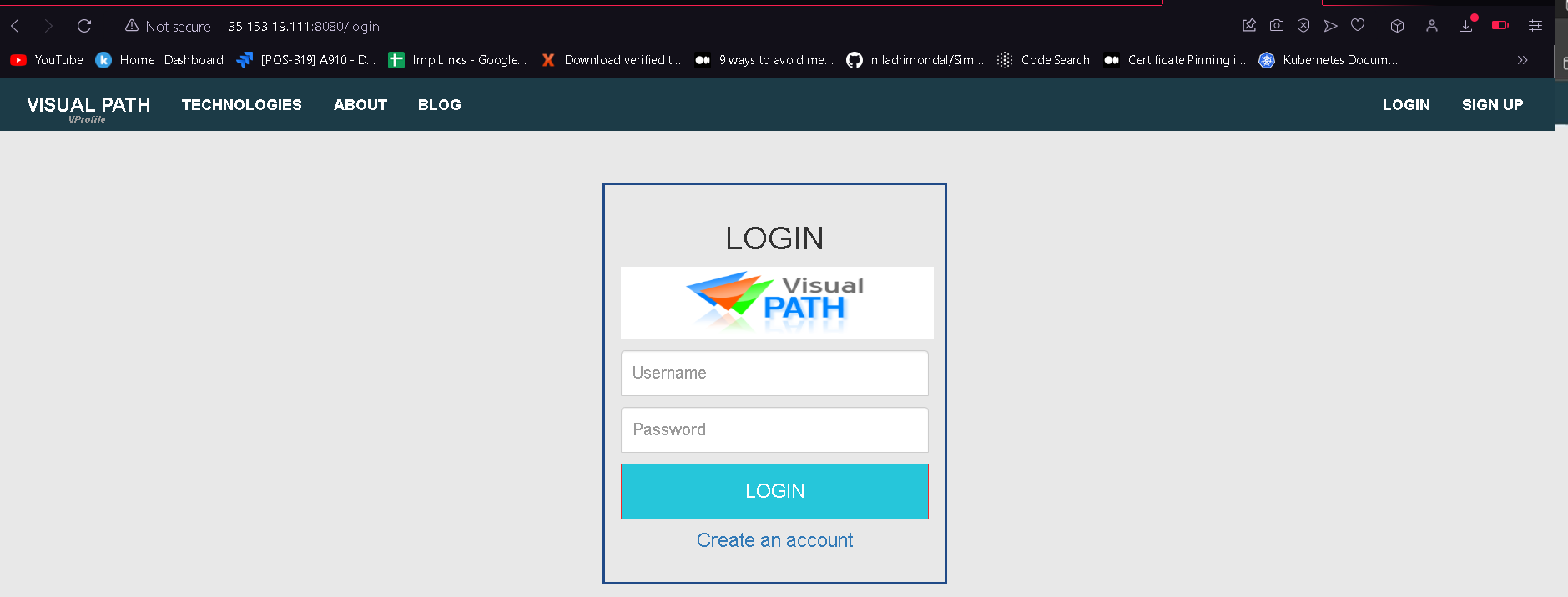
Use docker ps command to check all containers are working

docker ps -a



Now test on borwser weater its working or not

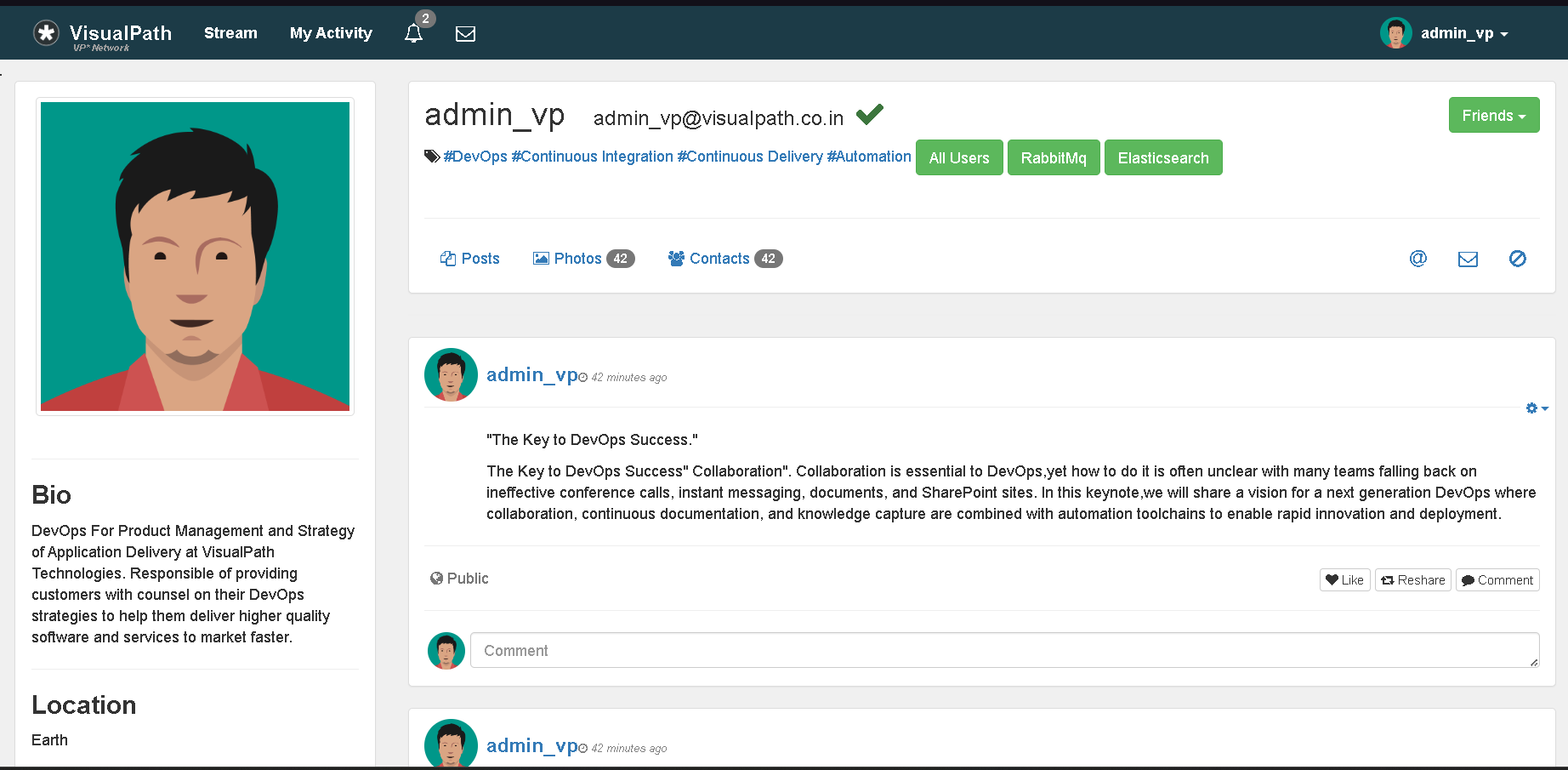
Now in browser type your\_ip:8080 you get login page



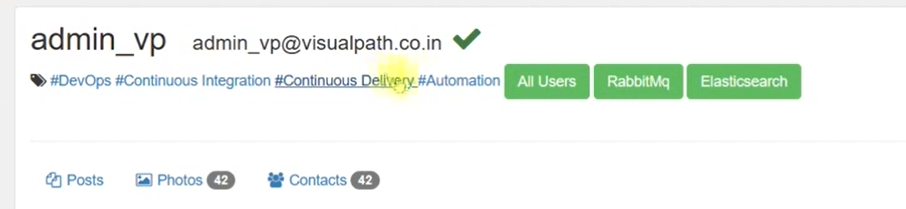
Loginid and password is same

Login id = admin\_vp and password = admin\_vp

If we able to login the consider your database container also working fine



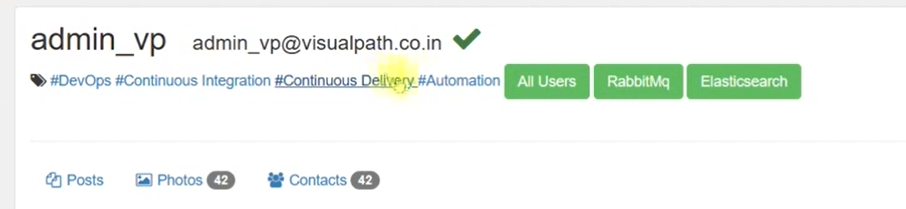
Now click on rabbit\_mq button in website



o/p



Check memcache workig or not by clicking all users button



o/p

