< MOVE JAVA APPLICATION IN CONTAINER USING DOCKER >

IMPLEMENTATION PLAN

VERSION 1.0 | 17/06/2022

Purpose

Containerization of JAVA App using Docker.

About the Project

- *Multi-Tier web application Stack*
- *Setup on Local PC / VM's*
- *Regular Deployment*
- *Continuous Changes*

Problem

- *High CapEx & OpEx*
- *Human Errors in deployment*
- *Not compatible with microservice architecture*
- *Resource wastage*
- *Not Portable*
- *Environment not in sync*

Solution

- *Containers*
- *Consume Low Resource*
- *Suits very well for microservice design*
- *Deployment via Images*
- *Same Container images across environment*
- *Reusable and repatable*

Tools

- *Docker – Container Runtime Environment*
- *JAVA Stack – NGINX/TOMCAT/MEMCACHED/RABBITMQ*
- *CLI – GIT Bash*
- *IDE – Sublime Text*

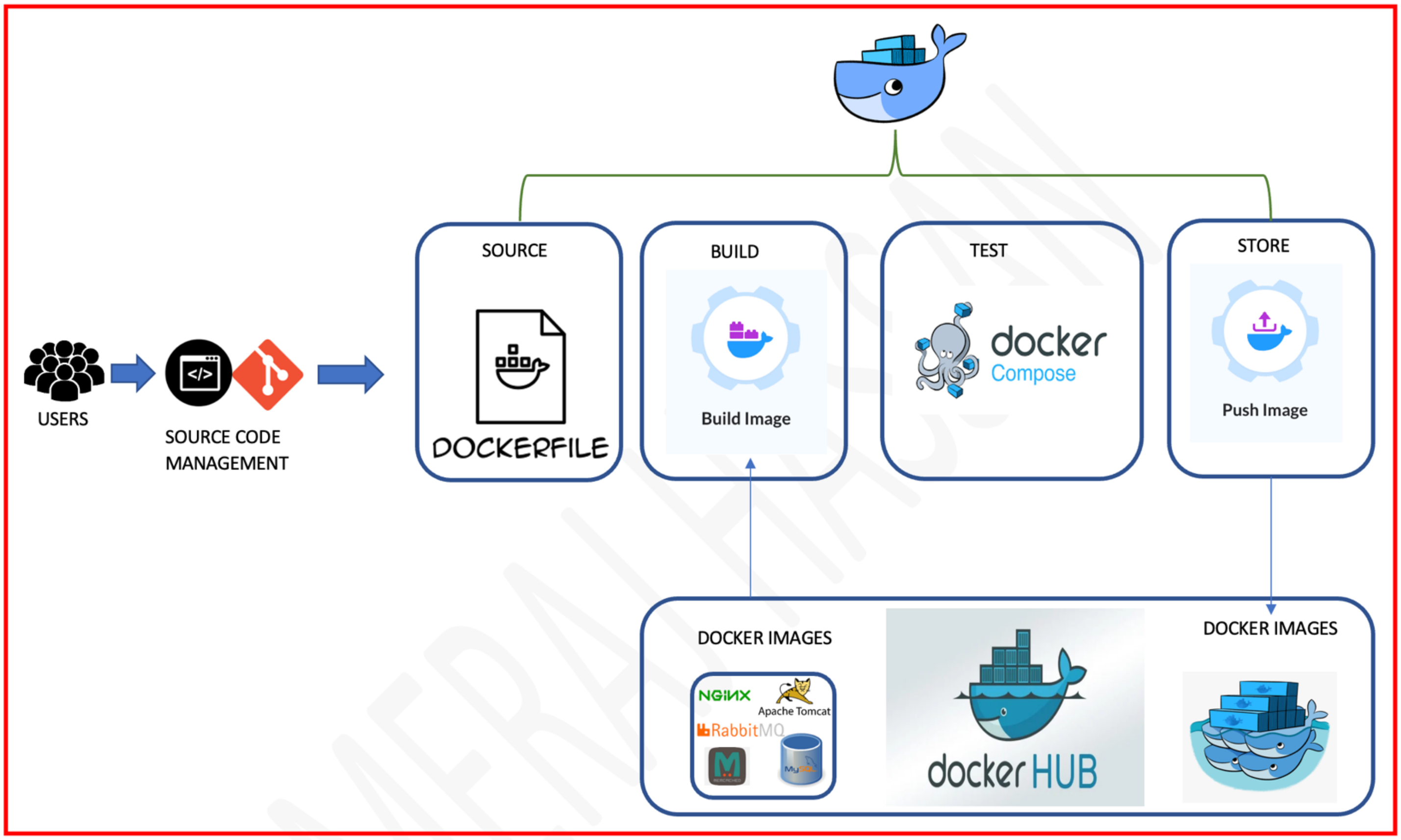
Objective

- *VM automation locally*
- *Baseline for other projects*
- *Real world project set up locally*

Architecture of Project Services

- *NGINX*
- *TOMCAT*
- *RABBITMQ*
- *MEMCACHED*
- *MYSQL*

Architecture of Docker Setup



Flow of Execution

- *Setup EC2 in AWS as a control machine.*
- *Find right base image from DockerHub*
- *Write Dockerfile to customize images*
- *Write docker-compose.yml file to run multi containers*
- *Test it & Host Images on DockerHub*

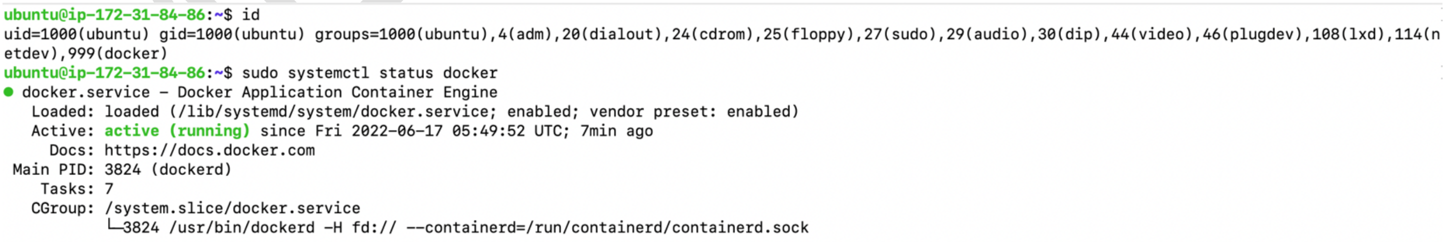
Prerequisite

- *AWS account*
- *DockerHub Account*
- *GITHUB Accounts*

1. Setup Docker Engine

Create EC2 instance (Ubuntu 18 | T2.micro) in AWS and Setup Docker Engine in that.

```
#!/bin/bash
# Install Docker CE
sudo apt-get update
sudo apt-get install -y \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

#Add Docker's official GPG key
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --
dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

#Use the following command to set up the repository
sudo echo \
  "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-
keyring.gpg] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io

# Install docker-compose
sudo curl -L
"https://github.com/docker/compose/releases/download/1.29.2/docker-
compose-$(uname -s)-$(uname -m)" \
-o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose

#Add Ubuntu User into Docker group
sudo usermod -a -G docker ubuntu
```



```
ubuntu@ip-172-31-84-86:~$ id
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),108(lxd),114(n
etdev),999(docker)
ubuntu@ip-172-31-84-86:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-06-17 05:49:52 UTC; 7min ago
     Docs: https://docs.docker.com
 Main PID: 3824 (dockerd)
    Tasks: 7
   CGroup: /system.slice/docker.service
           └─3824 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

2. Dockerhub & Dockerfile reference

URL: https://docs.docker.com/engine/reference/builder/

Docker File for App Image [TOMCAT]

```
FROM tomcat:8-jre11
RUN rm -rf /usr/local/tomcat/webapps/*
COPY target/vprofile-v2.war /usr/local/tomcat/webapps/ROOT.war

EXPOSE 8080
CMD ["catalina.sh","run"]
WORKDIR /usr/local/tomcat/
VOLUME /usr/local/tomcat/webapps
```

Docker File for DB Image [MySQL]

```
FROM mysql:5.7.25

ENV MYSQL_ROOT_PASSWORD:"vpropass"
ENV MYSQL_DATABASE:"accounts"

ADD db_backup.sql docker-entrypoint-init.d/db_backup.sql
```

Docker File for Web Image [NGINX]

```
FROM nginx
LABEL "Project"="Eereeda"
LABEL "Author"="Meraj"

RUN rm -rf /etc/nginx/conf.d/default.conf
COPY NginEereedaApp.conf /etc/nginx/conf.d/EeredaApp.conf
```

NGINX config file

```
upstream eereedaapp {
    server eereedaapp:8080
}
server {
    listen 80
location / {
    proxy_pass http://eereedaapp
}
}
```

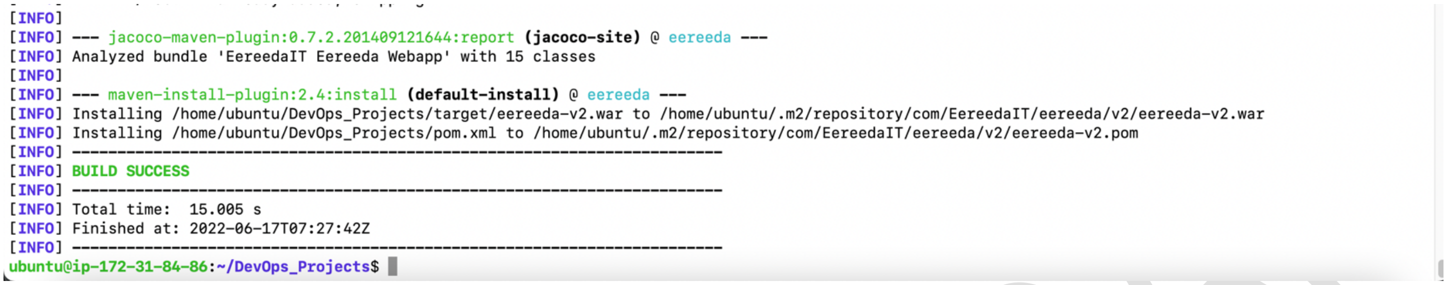
Building Image

a. Check hostname of server, ort number, userid & password in application.properties as per the project src/main/resource/application.properties

```
#JDBC Configutation for Database Connection
jdbc.driverClassName=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://eereedadb:3306/accounts?useUnicode=true&character
Encoding=UTF-8&zeroDateTimeBehavior=convertToNull
jdbc.username=root
jdbc.password=eereedapass

#Memcached Configuration For Active and StandBy Host
#For Active Host
memcached.active.host=eereedacache01
memcached.active.port=11211
#For StandBy Host
memcached.standBy.host=eereedacache02
memcached.standBy.port=11211

#RabbitMq Configuration
rabbitmq.address=eereedamq01
rabbitmq.port=15672
rabbitmq.username=guest
rabbitmq.password=guest

#Elasticsearch Configuration
elasticsearch.host =eereedasearch01
elasticsearch.port =9300
elasticsearch.cluster=vprofile
elasticsearch.node=vprofilenode
```

b. Build artifact using Maven

$ sudo apt install openjdk-8-jdk -y && sudo apt install maven -y
$ mvn install

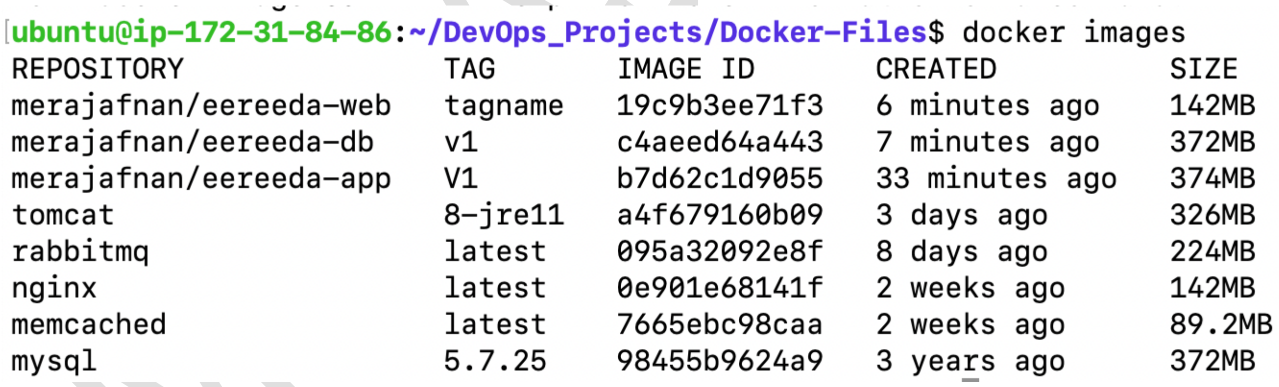


```
[INFO]
[INFO] --- jacoco-maven-plugin:0.7.2.201409121644:report (jacoco-site) @ eereeda ---
[INFO] Analyzed bundle 'EereedaIT Eereeda Webapp' with 15 classes
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ eereeda ---
[INFO] Installing /home/ubuntu/DevOps_Projects/target/eereeda-v2.war to /home/ubuntu/.m2/repository/com/EereedaIT/eereeda/v2/eereeda-v2.war
[INFO] Installing /home/ubuntu/DevOps_Projects/pom.xml to /home/ubuntu/.m2/repository/com/EereedaIT/eereeda/v2/eereeda-v2.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  15.005 s
[INFO] Finished at: 2022-06-17T07:27:42Z
[INFO] ------------------------------------------------------------------------
ubuntu@ip-172-31-84-86:~/DevOps_Projects$
```

$ cp -r target/ Docker-Files/app/

$ cd Docker-Files/app/
$ docker build -t merajafnan/eereeda-app:V1 .
$ cd ../db/
$ docker build -t merajafnan/eereeda-db:v1 .
$ cd ../web/
$ docker build -t merajafnan/eereeda-web:v1 .

$ docker pull Memcached
$ docker pull rabbitmq
$ docker image



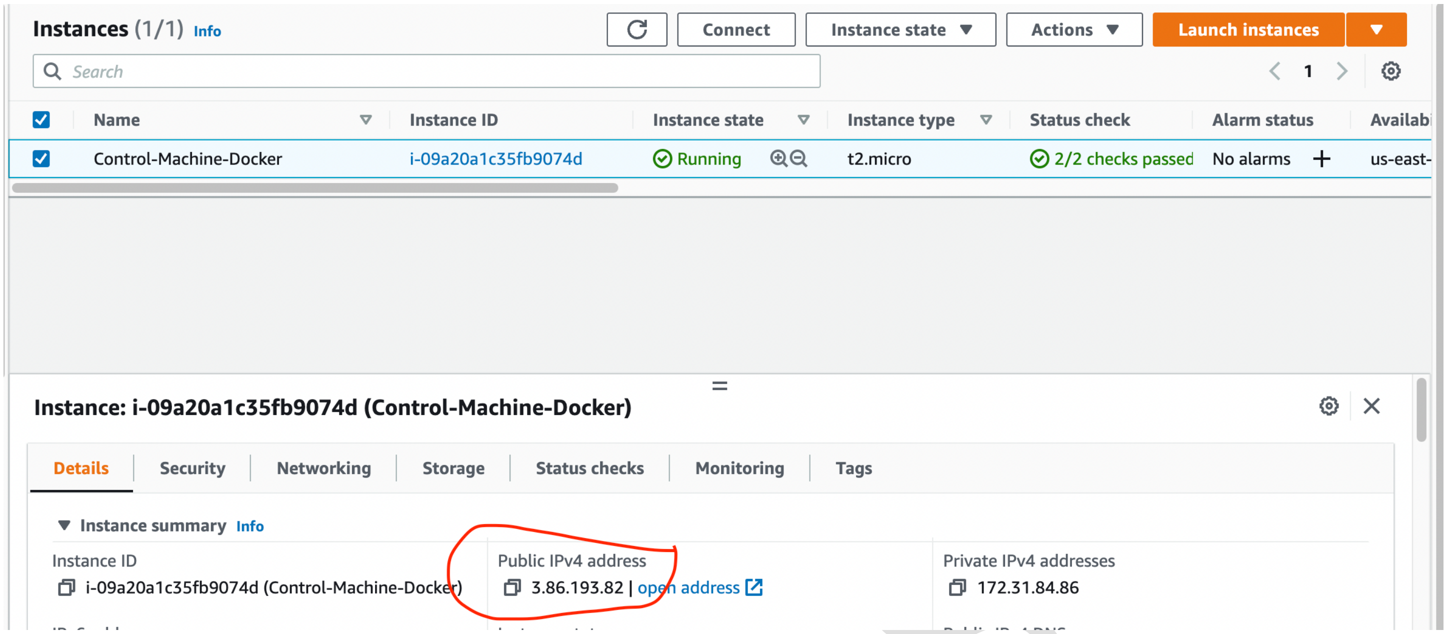
```
ubuntu@ip-172-31-84-86:~/DevOps_Projects/Docker-Files$ docker images
REPOSITORY                TAG       IMAGE ID       CREATED          SIZE
merajafnan/eereeda-web    tagname   19c9b3ee71f3   6 minutes ago    142MB
merajafnan/eereeda-db     v1        c4aeed64a443   7 minutes ago    372MB
merajafnan/eereeda-app    V1        b7d62c1d9055   33 minutes ago   374MB
tomcat                    8-jre11   a4f679160b09   3 days ago       326MB
rabbitmq                  latest    095a32092e8f   8 days ago       224MB
nginx                     latest    0e901e68141f   2 weeks ago      142MB
memcached                 latest    7665ebc98caa   2 weeks ago      89.2MB
mysql                     5.7.25    98455b9624a9   3 years ago      372MB
```

3. Docker Compose

Run all the container and connect them together using Docker Compose {*docker-compose.yml*}

```
version: '3'
services:
  eereedadb:
    image: merajafnan/eereeda-db:v1
    ports:
      - "3306:3306"
    volumes:
      - eereedadbdata:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD="eereedapass"

  eereedacache01:
    image: memcached
    ports:
      - "11211:11211"

  eereedamq01:
    image: rabbitmq
    ports:
      - "15672:15672"
    environment:
      - RABBITMQ_DEFAULT_USER="guest"
      - RABBITMQ_DEFAULT_PASS="guest"

  eereedaapp:
    image: merajafnan/eereeda-app:V1
    ports:
      - "8080:8080"
    volumes:
      - eereedaappdata:/usr/local/tomcat/webapps

  eereedaweb:
    image: merajafnan/eereeda-web:v1
    ports:
      - "80:80"

volumes:
  eereedadbdata: {}
  eereedaappdata: {}
```

Login using Public Ip of EC2 Insatance.



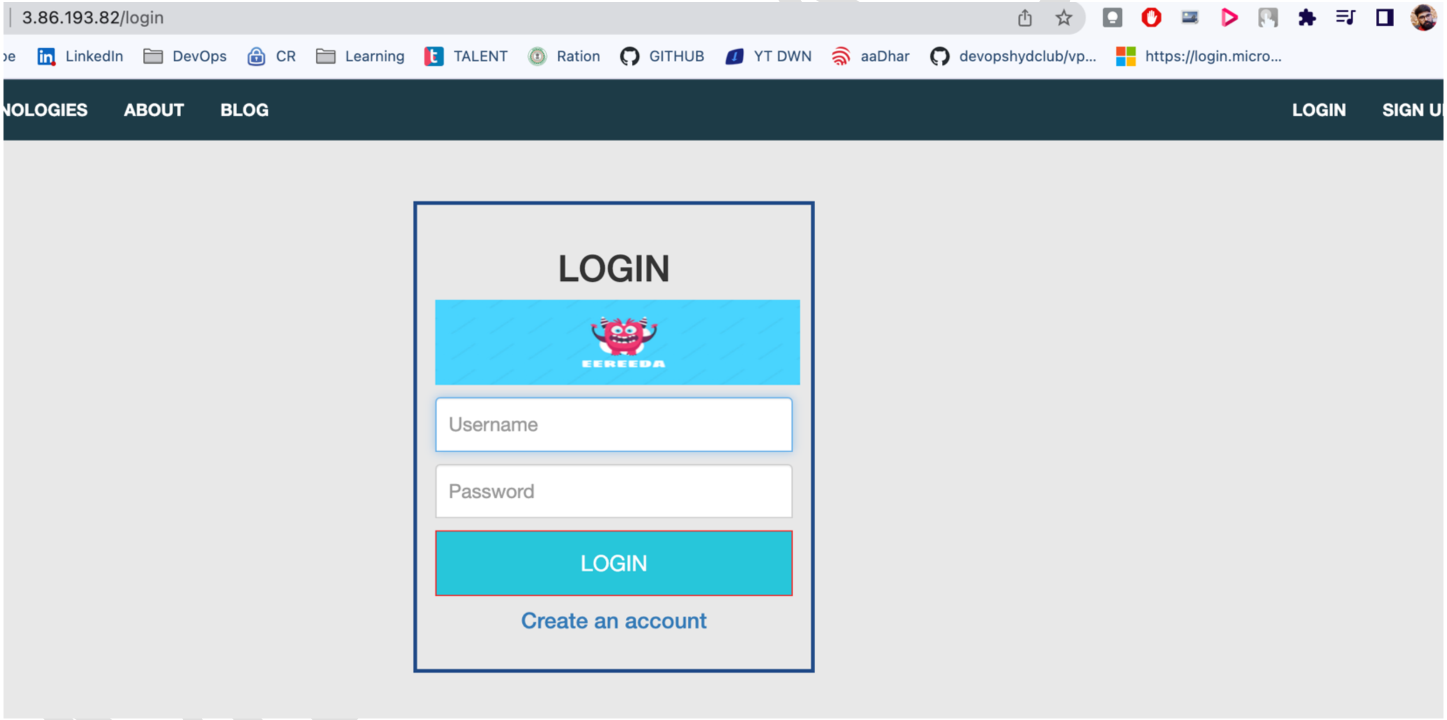
☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabi
☑	Control-Machine-Docker	i-09a20a1c35fb9074d	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-

Instance: i-09a20a1c35fb9074d (Control-Machine-Docker)

Instance ID
i-09a20a1c35fb9074d (Control-Machine-Docker)

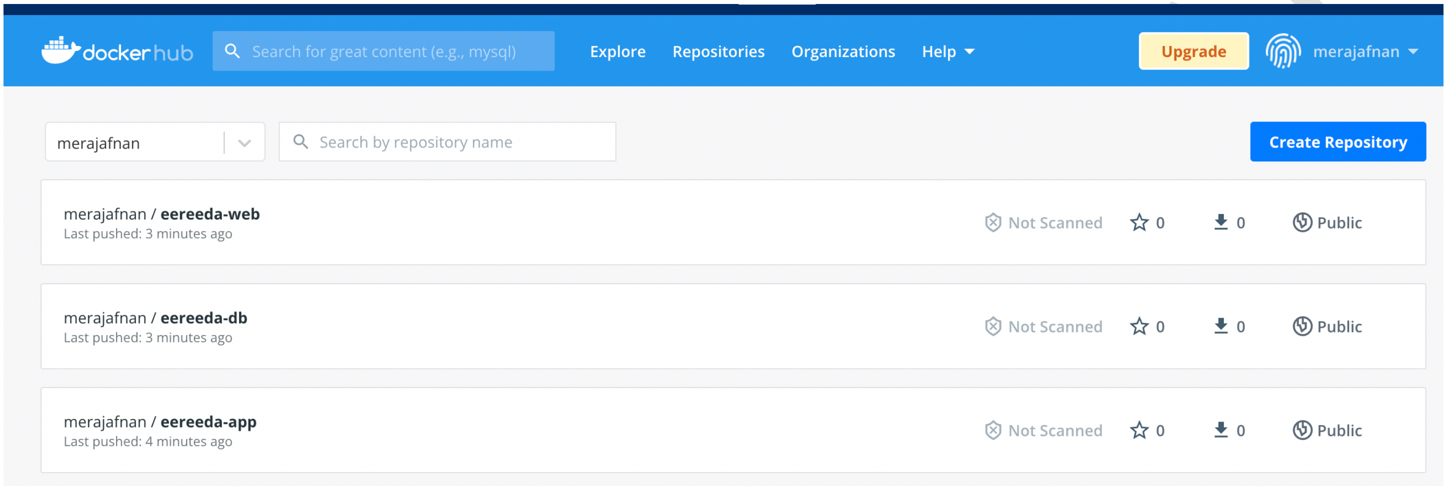
Public IPv4 address
3.86.193.82 | open address

Private IPv4 addresses
172.31.84.86

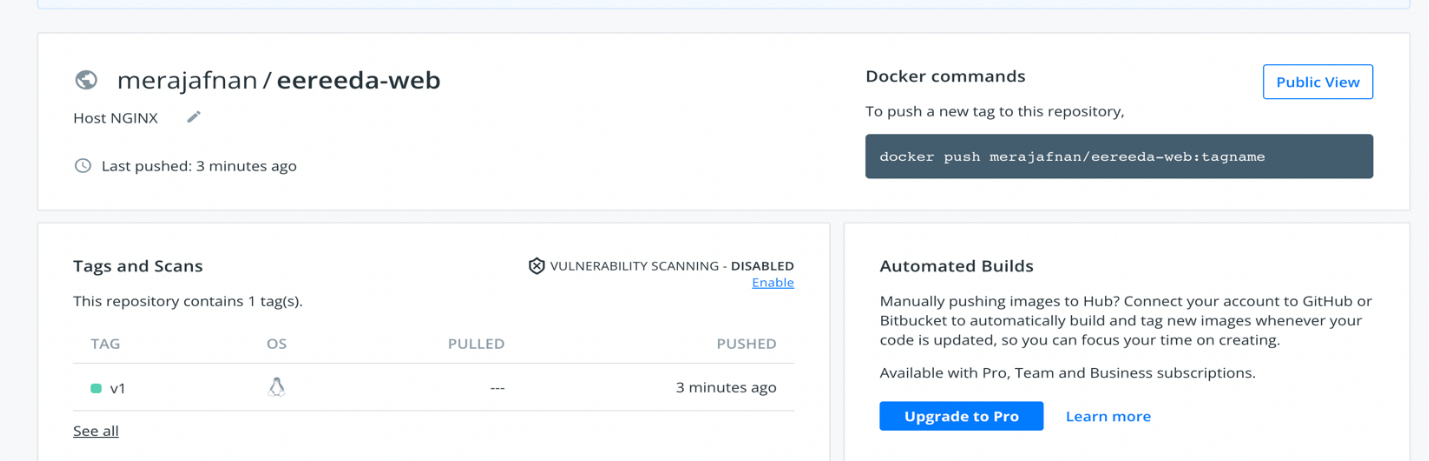


4. Push Image to Docker Hub

```
$ docker login
$ docker push merajafnan/eereeda-app:V1
$ docker push merajafnan/eereeda-db:v1
$ docker push merajafnan/eereeda-web:v1
```



Repository	Scan	Stars	Downloads	Visibility
merajafnan / **eereeda-web** Last pushed: 3 minutes ago	Not Scanned	0	0	Public
merajafnan / **eereeda-db** Last pushed: 3 minutes ago	Not Scanned	0	0	Public
merajafnan / **eereeda-app** Last pushed: 4 minutes ago	Not Scanned	0	0	Public


merajafnan / **eereeda-web**

Host NGINX

Last pushed: 3 minutes ago

Docker commands

To push a new tag to this repository,

```
docker push merajafnan/eereeda-web:tagname
```

Tags and Scans

VULNERABILITY SCANNING - **DISABLED**

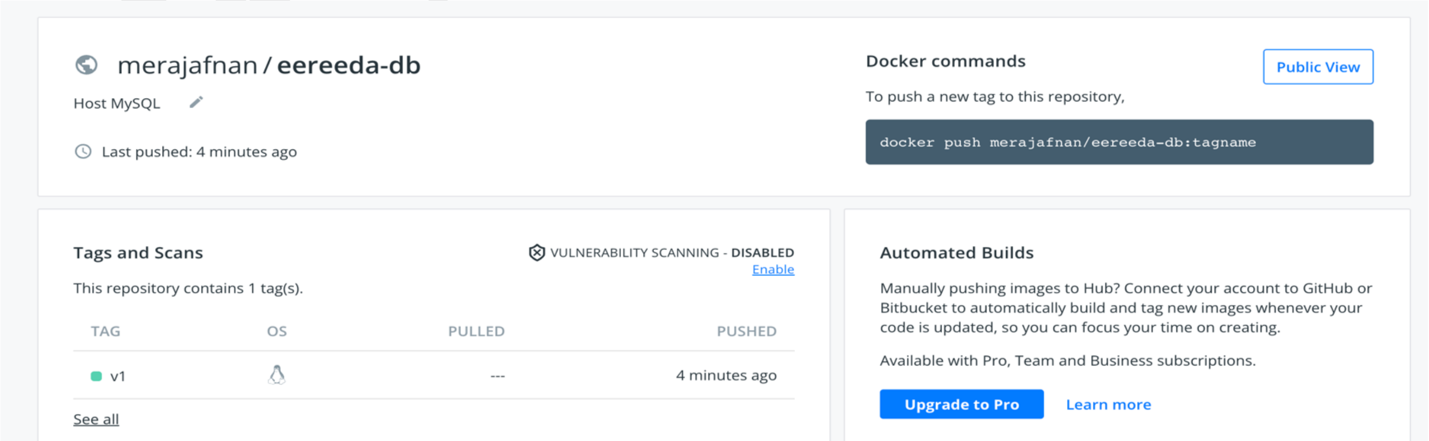
This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
v1		---	3 minutes ago

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions.


merajafnan / **eereeda-db**

Host MySQL

Last pushed: 4 minutes ago

Docker commands

To push a new tag to this repository,

```
docker push merajafnan/eereeda-db:tagname
```

Tags and Scans

VULNERABILITY SCANNING - **DISABLED**

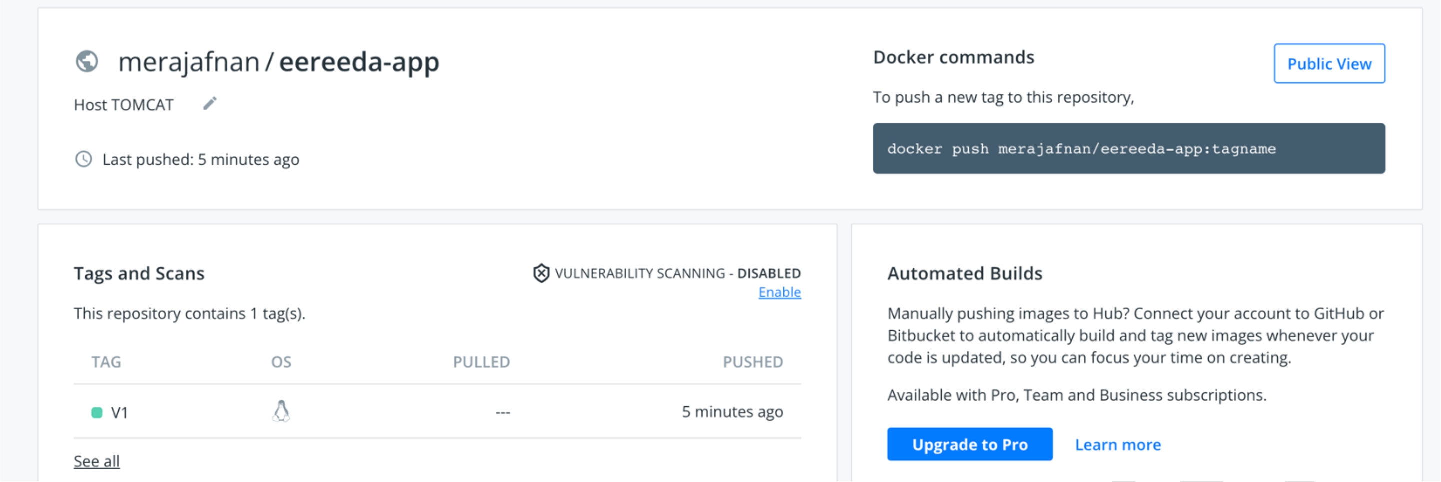
This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
v1		---	4 minutes ago

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions.



merajafnan / eereeda-app

Host TOMCAT

Last pushed: 5 minutes ago

Docker commands

To push a new tag to this repository,

```
docker push merajafnan/eereeda-app:tagname
```

Tags and Scans

VULNERABILITY SCANNING - DISABLED

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
🟩 V1		---	5 minutes ago

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions.

11. References

The following table summarizes the documents referenced in this plan.

DOCUMENT NAME	INSTRUCTOR	LOCATION
DevOps Beginners to Advanced	Imran Teli	https://www.udemy.com/course/decodingdevops/learn/lecture/28273912?start=0#overview