# < SETUP MULTI TIER WEB APPLICATION LOCALLY >

## IMPLEMENTATION PLAN

VERSION 1.0 | 18/05/2022

### Purpose

*In this section, describe the purpose to setup multi-tier web application locally in pc.*

### About the Project

- *Multi-Tier web application Stack*
- *Setup on Local PC*
- *Baseline for upcoming projects.*
- *Helps to setup any project locally.*

### Problem

- *Not comfortable in making changes in real server*
- *Local setup is complex*
- *Time consuming*
- *Not repeatable*

## Solution

- *Automated*
- *Repeatable*
- *Code ( IAAS )*
- *R&D in own Machine*

### Tools

- *Hypervisor – Oracle VM*
- *Automation – Vagrant*
- *CLI – GIT Bash*
- *IDE – Sublime Text*

### Objective

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

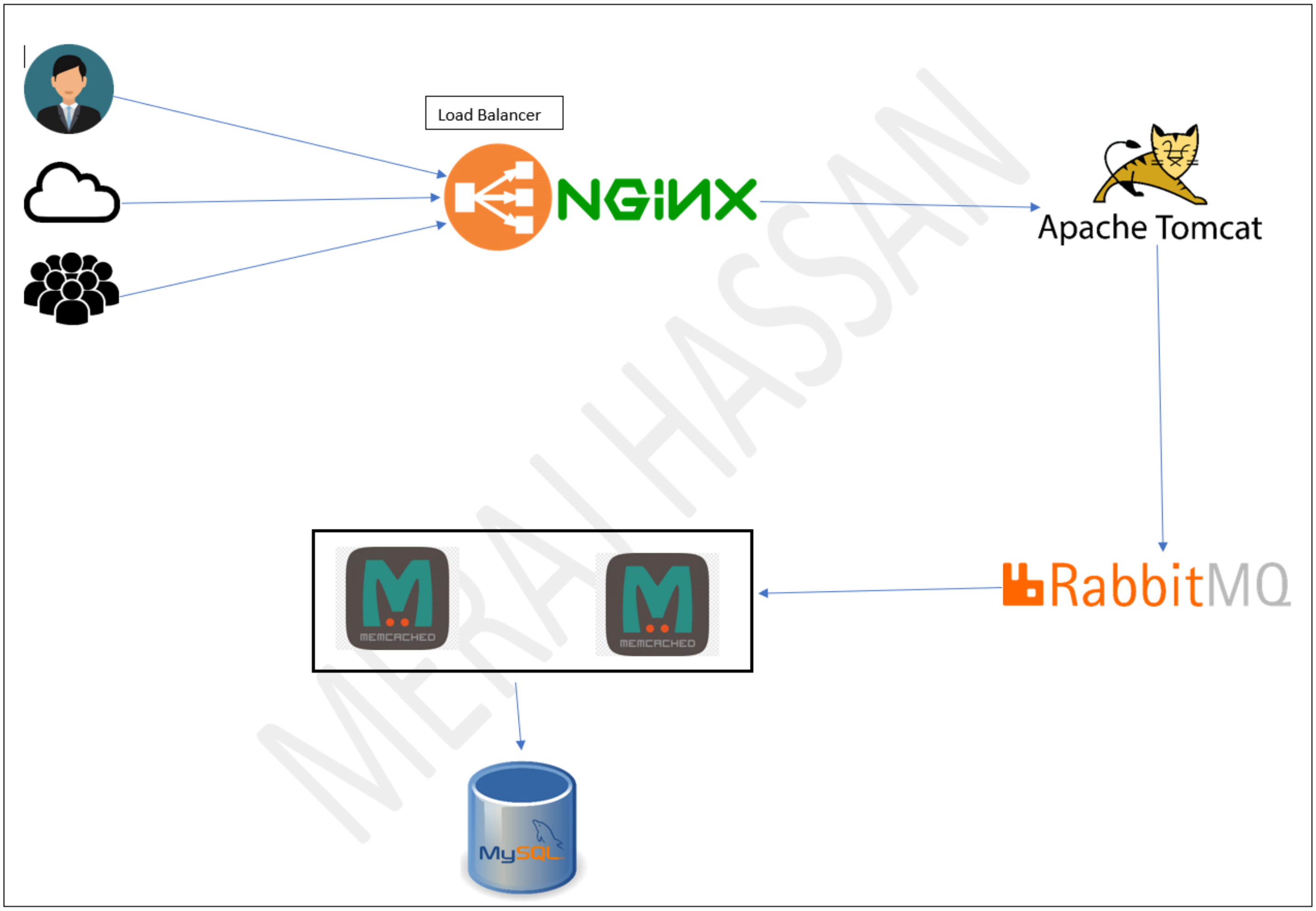
### Architecture of Project Services

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

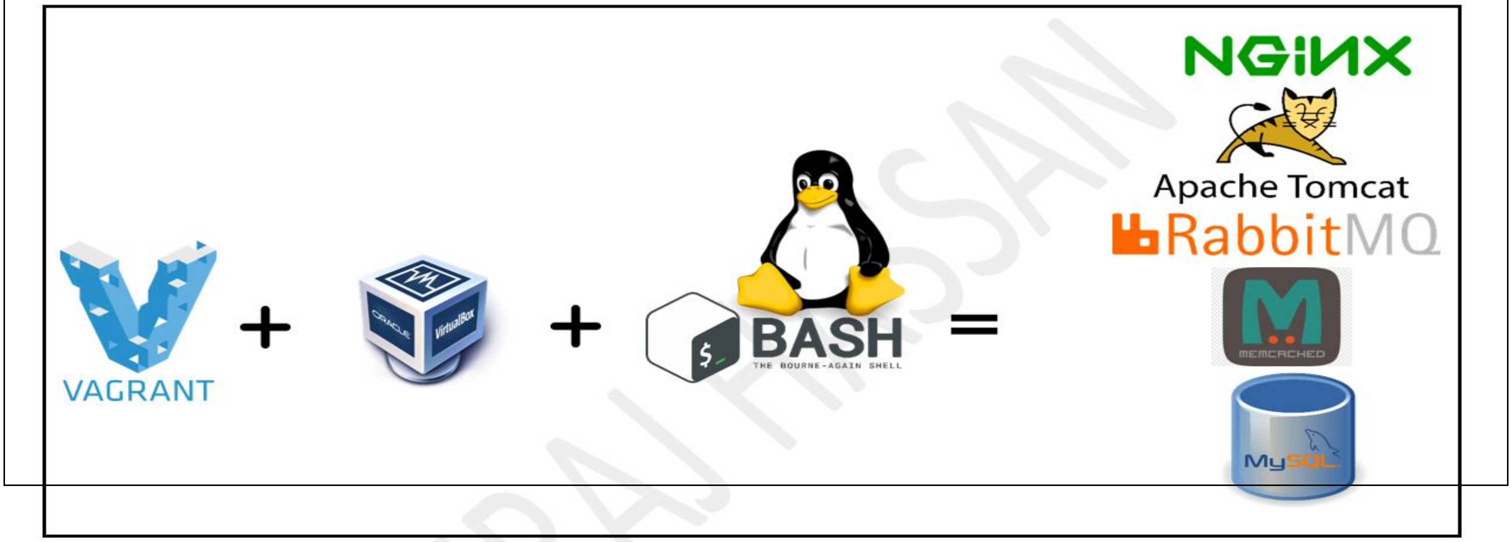
### Architecture of Automated Setup

- *VAGRANT*
- *VIRTUALBOX*
- *GITBASH*

### Project Services Overview



### Automated Setup Overview



*Add or delete additional lines as needed to the table. If the applicable team members are listed in the Project Management Plan, reference the appropriate section within that document.*

**Flow of Execution**

- *Setup Tools –*
- *Clone Source Code – Branch { local-setup }*
- *cd into vagrant dir*
- *Bring VM up*
- *Validate*
- *Setup All services*
  - *MySql*
  - *Memcached*
  - *RabbitMQ*
  - *Tomcat*
  - *Nginx*
  - *App Build & Deploy*

## Prerequisite

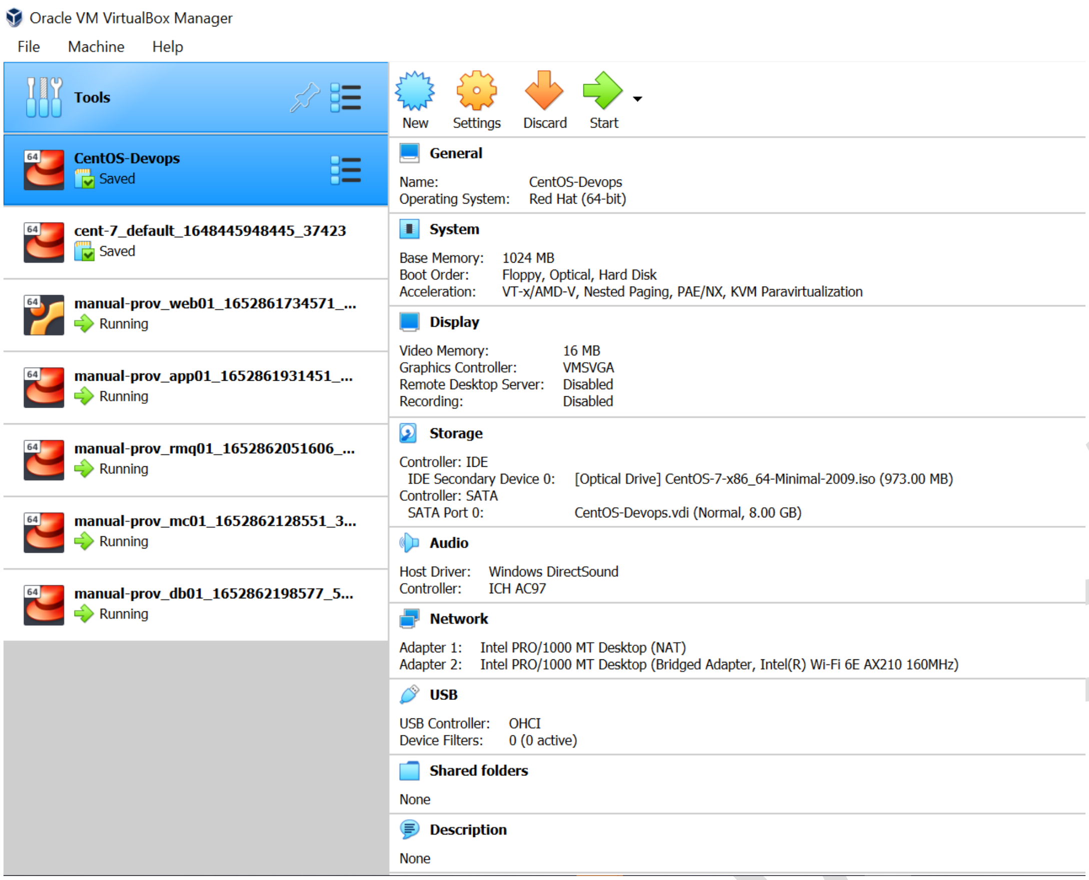
1. *Oracle VM Virtualbox*
2. *Vagrant*
3. *Vagrant plugins*
   - *vagrant plugin install vagrant-hostmanager*
   - *vagrant plugin install vagrant-vbguest*
4. *Git bash or equivalent editor*

## VM Setup

1. *Clone source code.*
2. *Cd into the repository - DevOps_Projects*
3. *Switch to the branch - local-setup*
4. *cd - vagrant/manual-prov*
5. *Major Tasks*

## Bring up VM's

```
$ vagrant up
```



CentOS-Devops
Saved

cent-7_default_1648445948445_37423
Saved

manual-prov_web01_1652861734571_...
Running

manual-prov_app01_1652861931451_...
Running

manual-prov_rmq01_1652862051606_...
Running

manual-prov_mc01_1652862128551_3...
Running

manual-prov_db01_1652862198577_5...
Running

**General**

Name: CentOS-Devops
Operating System: Red Hat (64-bit)

**System**

Base Memory: 1024 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization

**Display**

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

**Storage**

Controller: IDE
IDE Secondary Device 0: [Optical Drive] CentOS-7-x86_64-Minimal-2009.iso (973.00 MB)
Controller: SATA
SATA Port 0: CentOS-Devops.vdi (Normal, 8.00 GB)

**Audio**

Host Driver: Windows DirectSound
Controller: ICH AC97

**Network**

Adapter 1: Intel PRO/1000 MT Desktop (NAT)
Adapter 2: Intel PRO/1000 MT Desktop (Bridged Adapter, Intel(R) Wi-Fi 6E AX210 160MHz)

**USB**

USB Controller: OHCI
Device Filters: 0 (0 active)

**Shared folders**

None

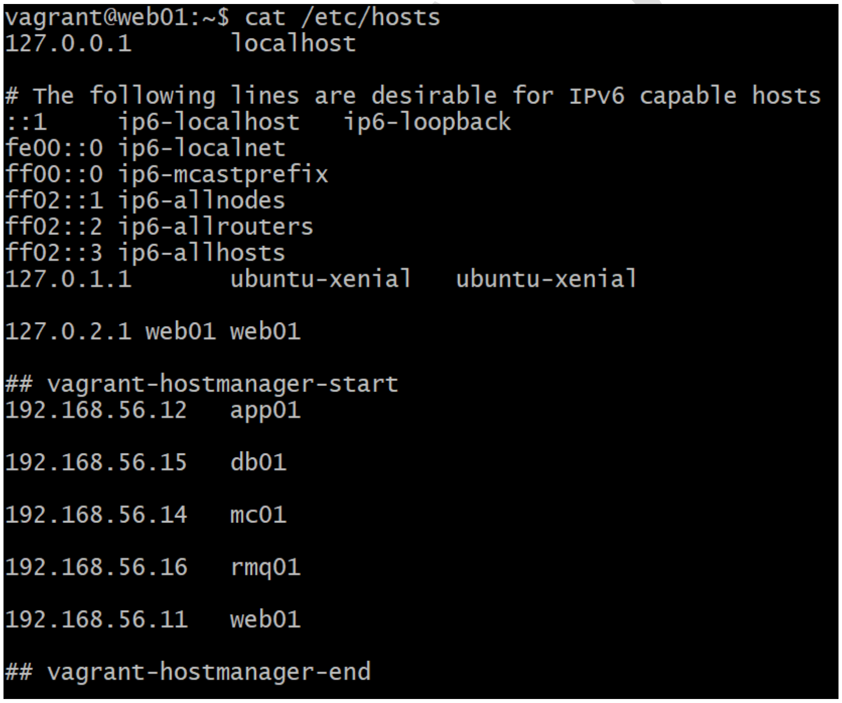
**Description**

None

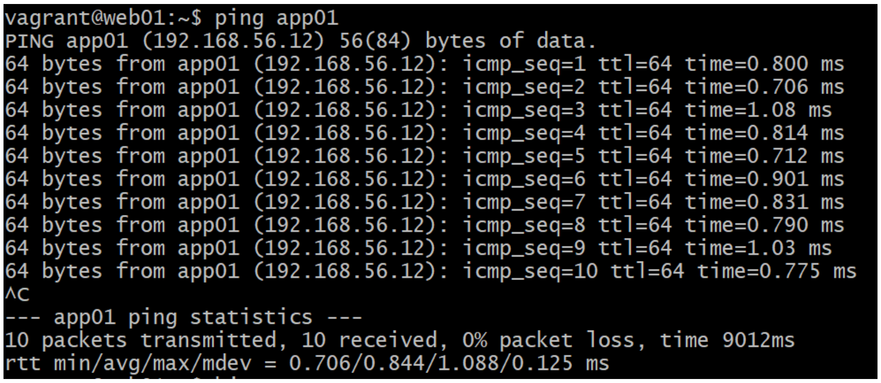
$ ssh vagrant web01

$ cat /etc/hosts

$ ping app01



```
vagrant@web01:~$ cat /etc/hosts
127.0.0.1       localhost

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost   ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
127.0.1.1       ubuntu-xenial   ubuntu-xenial

127.0.2.1 web01 web01

## vagrant-hostmanager-start
192.168.56.12   app01

192.168.56.15   db01

192.168.56.14   mc01

192.168.56.16   rmq01

192.168.56.11   web01

## vagrant-hostmanager-end
```



```
vagrant@web01:~$ ping app01
PING app01 (192.168.56.12) 56(84) bytes of data.
64 bytes from app01 (192.168.56.12): icmp_seq=1 ttl=64 time=0.800 ms
64 bytes from app01 (192.168.56.12): icmp_seq=2 ttl=64 time=0.706 ms
64 bytes from app01 (192.168.56.12): icmp_seq=3 ttl=64 time=1.08 ms
64 bytes from app01 (192.168.56.12): icmp_seq=4 ttl=64 time=0.814 ms
64 bytes from app01 (192.168.56.12): icmp_seq=5 ttl=64 time=0.712 ms
64 bytes from app01 (192.168.56.12): icmp_seq=6 ttl=64 time=0.901 ms
64 bytes from app01 (192.168.56.12): icmp_seq=7 ttl=64 time=0.831 ms
64 bytes from app01 (192.168.56.12): icmp_seq=8 ttl=64 time=0.790 ms
64 bytes from app01 (192.168.56.12): icmp_seq=9 ttl=64 time=1.03 ms
64 bytes from app01 (192.168.56.12): icmp_seq=10 ttl=64 time=0.775 ms
^C
--- app01 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9012ms
rtt min/avg/max/mdev = 0.706/0.844/1.088/0.125 ms
```

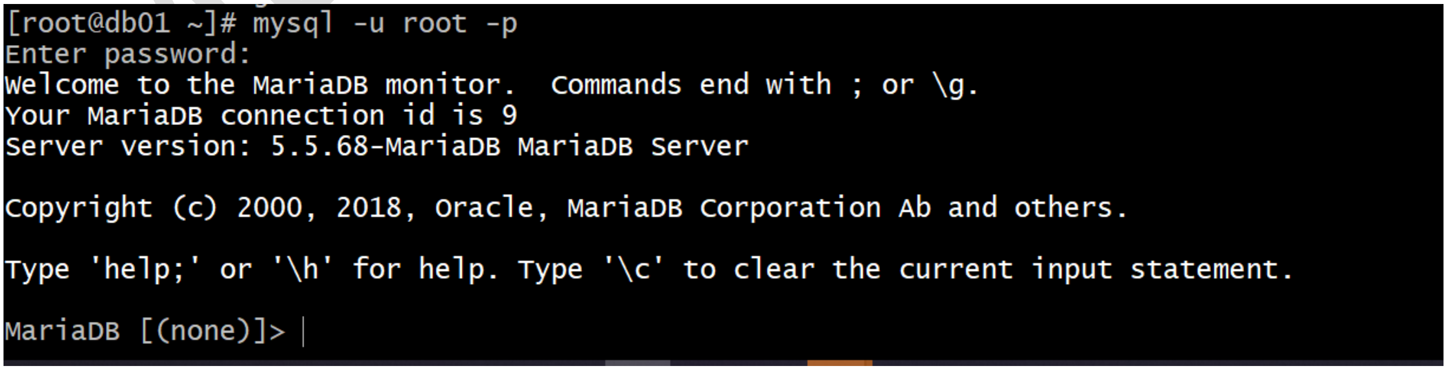
Similarly check from all other VM’s

## Provisioning

1. *MySQL (Database SVC) - SQL Database*
2. *Memcache (DB Caching SVC) - DB Caching*
3. *RabbitMQ (Broker/Queue SVC) - Broker/Queuing Agent*
4. *Tomcat (Application SVC) -Application Server*
5. *Nginx (Web SVC) - Web Service*

### 1. MySQL (Database SVC) - SQL Database

```
$ vagrant ssh db01
yum update -y
DATABASE_PASS='admin123'
vi /etc/profile
        DATABASE_PASS='admin123'
yum install epel-release -y                # Set Repository
yum install git mariadb-server -y
systemctl start mariadb
systemctl status mariadb
mysql_secure_installation                  # RUN mysql secure installation script.
mysql -u root -p                           # Login MySQL
```



```
[root@db01 ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 5.5.68-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> |
```

**Set DB name and users:**

```
mysql -u root -padmin123
        create database accounts;
        CREATE USER 'admin'@'%' IDENTIFIED BY 'admin123'
        GRANT ALL PRIVILEGES ON accounts.* TO 'admin'@'%' WITH GRANT OPTION;
```

**Download Source code & Initialize Database:**

```
git clone -b local-setup https://github.com/merajafnan/DevOps_Projects.git
cd DevOps_Projects/
mysql -u root -padmin123 accounts < src/main/resources/db_backup.sql
mysql -u root -padmin123 -e "FLUSH PRIVILEGES"
mysql -u root -padmin123 accounts
        show databases;
        use accounts;
        show tables;
```

## 2. MEMCACHE SETUP

```
yum install epel-release -y
yum install memcached -y
systemctl start memcached
systemctl enable memcached
systemctl status memcached
memcached -p 11211 -U 11111 -u memcached -d
ss -tunlp | grep 11211
```

## 3. RABBITMQ SETUP

**Set EPEL Repository**

```
# yum install epel-release -y
```

**Install Dependencies**

```
#sudo yum install wget -y
#cd /tmp/
#wget http://packages.erlang-solutions.com/erlang-solutions-2.0-1.noarch.rpm
#sudo rpm -Uvh erlang-solutions-2.0-1.noarch.rpm
#sudo yum -y install erlang socat
```

**Install Rabbitmq Server**

```
#curl -s https://packagecloud.io/install/repositories/rabbitmq/rabbitmq-server/script.rpm.sh | sudo bash
#sudo yum install rabbitmq-server -y
```

**Start & Enable RabbitMQ**

```
#sudo systemctl start rabbitmq-server
#sudo systemctl enable rabbitmq-server
#sudo systemctl status rabbitmq-server
```

**Config Change**

```
#sudo sh -c 'echo "[{rabbit, [{loopback_users, []}]}]." > /etc/rabbitmq/rabbitmq.config'
#sudo rabbitmqctl add_user test test
#sudo rabbitmqctl set_user_tags test administrator
```

**Restart RabbitMQ service**

```
# systemctl restart rabbitmq-server
```

## 4. TOMCAT SETUP

```
vagrant ssh app01
yum update -y
yum install epel-release -y
```

**Install Dependencies**

```
# yum install java-1.8.0-openjdk -y
# yum install git maven wget -y
```

**Change dir to /tmp**

```
# cd /tmp/
```

**Download & Tomcat Package**

```
# wget https://archive.apache.org/dist/tomcat/tomcat-8/v8.5.37/bin/apache-tomcat-8.5.37.tar.gz
# tar xzvf apache-tomcat-8.5.37.tar.gz
```

**Add tomcat user**

```
# useradd --home-dir /usr/local/tomcat8 --shell /sbin/nologin tomcat
```

**Copy data to tomcat home dir**

```
# cp -r /tmp/apache-tomcat-8.5.37/* /usr/local/tomcat8/
```

**Make tomcat user owner of tomcat home dir**

```
# chown -R tomcat.tomcat /usr/local/tomcat8
```

**Setup systemd for tomcat**

**Update file with following content.**

```
vi /etc/systemd/system/tomcat.service
        [Unit]
        Description=Tomcat
        After=network.target
        [Service]
        User=tomcat
        WorkingDirectory=/usr/local/tomcat8
        Environment=JRE_HOME=/usr/lib/jvm/jre
        Environment=JAVA_HOME=/usr/lib/jvm/jre
        Environment=CATALINA_HOME=/usr/local/tomcat8
        Environment=CATALINE_BASE=/usr/local/tomcat8
        ExecStart=/usr/local/tomcat8/bin/catalina.sh run
        ExecStop=/usr/local/tomcat8/bin/shutdown.sh
        SyslogIdentifier=tomcat-%i
        [Install]
        WantedBy=multi-user.target
```

```
# systemctl daemon-reload
# systemctl start tomcat
# systemctl enable tomcat
```

## 5. CODE BUILD & DEPLOY (app01)

**Download Source code**

```
# git clone -b local-setup https://github.com/merajafnan/DevOps_Projects.git
```

**Update configuration**

```
# cd DevOps_Projects
# vim src/main/resources/application.properties
# Update file with backend server details
```

**Build code**

**Run below command inside the repository (vprofile-project)**

```
# mvn install
```

**Deploy artifact**

```
# systemctl stop tomcat
# sleep 120
# rm -rf /usr/local/tomcat8/webapps/ROOT*
# cp target/vprofile-v2.war /usr/local/tomcat8/webapps/ROOT.war
# systemctl start tomcat
# sleep 300
# chown tomcat.tomcat /usr/local/tomcat8/webapps -R
# systemctl restart tomcat
```

## 6. NGINX SETUP

**Login to the Nginx vm**

```
$ vagrant ssh web01
```

**Verify Hosts entry, if entries missing update the it with IP and hostnames**

```
# cat /etc/hosts
```

**Update OS with latest patches**

```
# apt update
# apt upgrade
```

**Install nginx**

```
# apt install nginx -y
```

**Create Nginx conf file with below content**

```
# vi /etc/nginx/sites-available/vproapp
          upstream vproapp {
           server app01:8080;
          }
          server {
           listen 80;
          location / {
           proxy_pass http://vproapp;
          }
          }
```

**Remove default nginx conf**

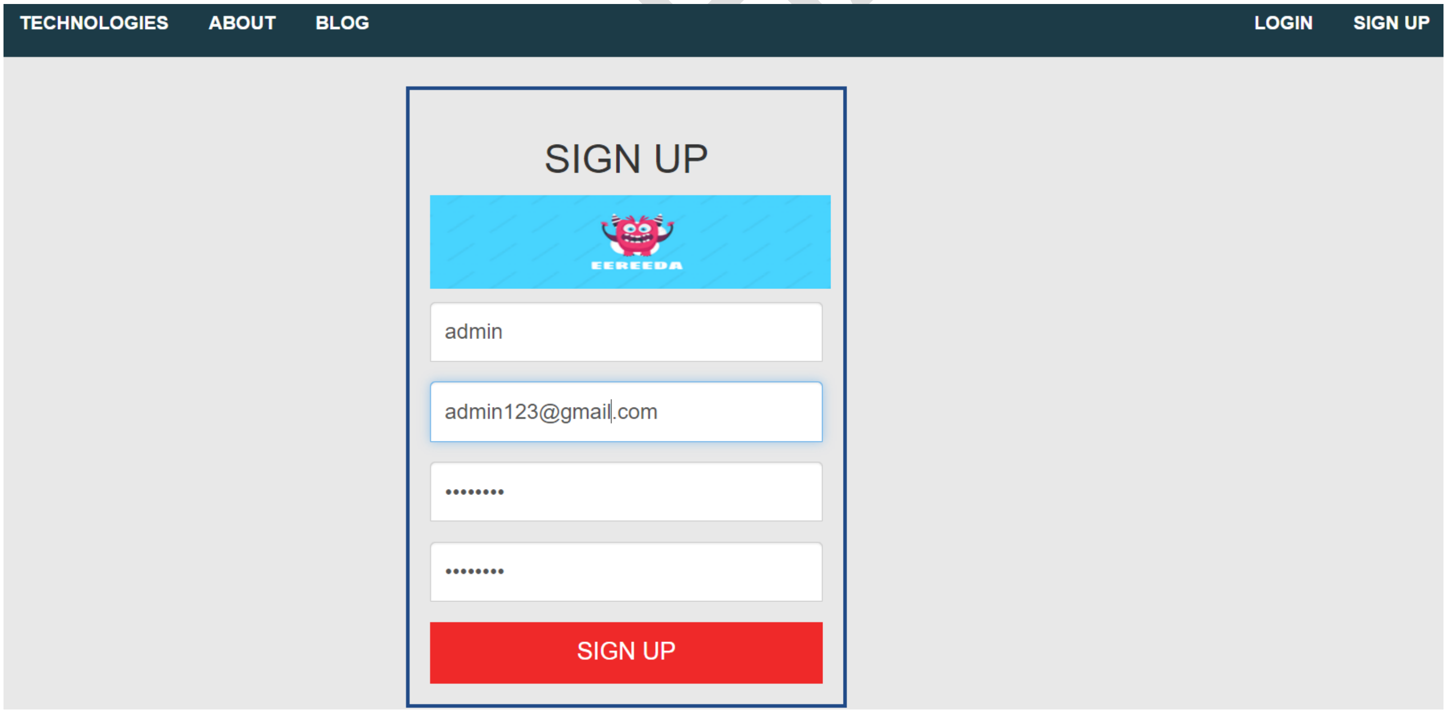
```
# rm -rf /etc/nginx/sites-enabled/default
```

**Create link to activate website**

```
# ln -s /etc/nginx/sites-available/vproapp /etc/nginx/sites-enabled/vproapp
```

**Restart Nginx**

```
# systemctl restart nginx
```



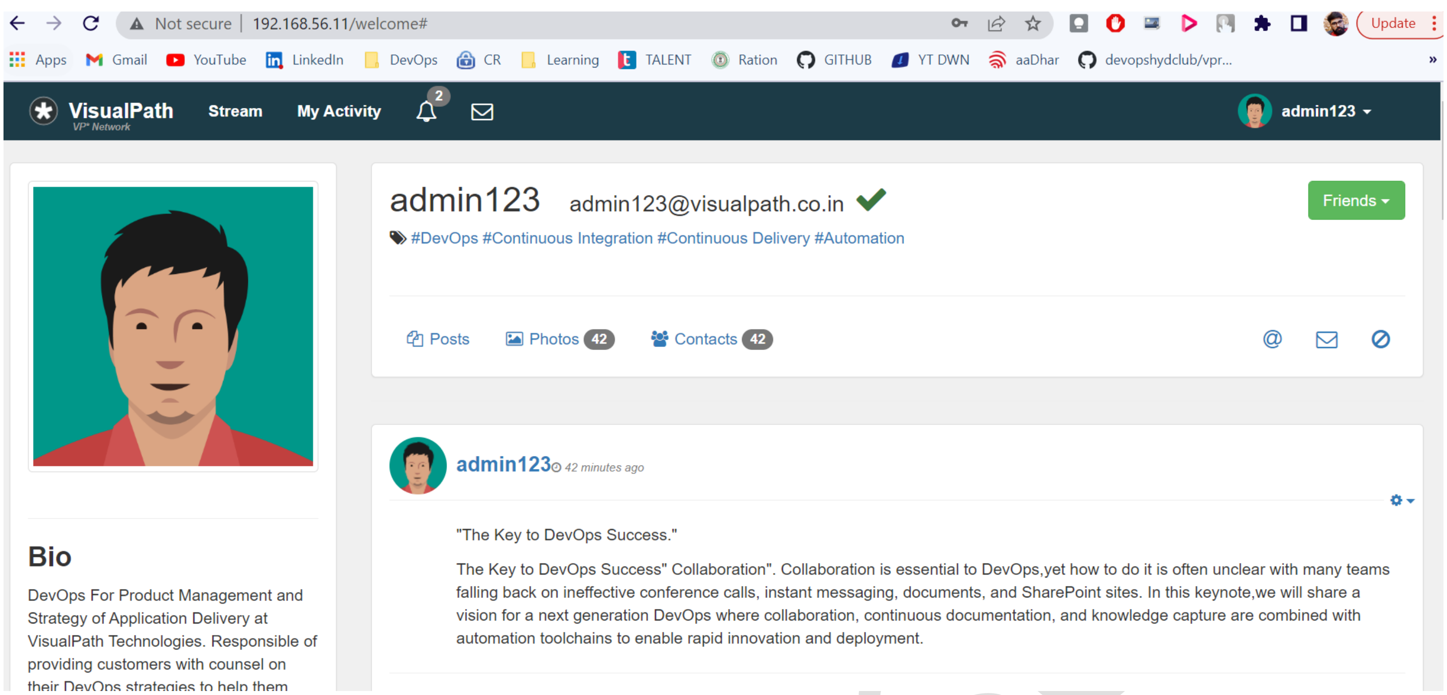
SIGN UP

admin

admin123@gmail.com

••••••••

••••••••



admin123 admin123@visualpath.co.in ✓

#DevOps #Continuous Integration #Continuous Delivery #Automation

Bio

DevOps For Product Management and Strategy of Application Delivery at VisualPath Technologies. Responsible of providing customers with counsel on their DevOps strategies to help them

admin123 42 minutes ago

"The Key to DevOps Success."

The Key to DevOps Success" Collaboration". Collaboration is essential to DevOps,yet how to do it is often unclear with many teams falling back on ineffective conference calls, instant messaging, documents, and SharePoint sites. In this keynote,we will share a vision for a next generation DevOps where collaboration, continuous documentation, and knowledge capture are combined with automation toolchains to enable rapid innovation and deployment.

## 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 |
| | | |
| | | |