

# VPROFILE PROJECT SETUP

## Prerequisite

1. Oracle VM Virtualbox
2. Vagrant
3. Vagrant plugins

Execute below command in your computer to install hostmanager plugin

```
$ vagrant plugin install vagrant-hostmanager
```

4. Git bash or equivalent editor

## VM SETUP

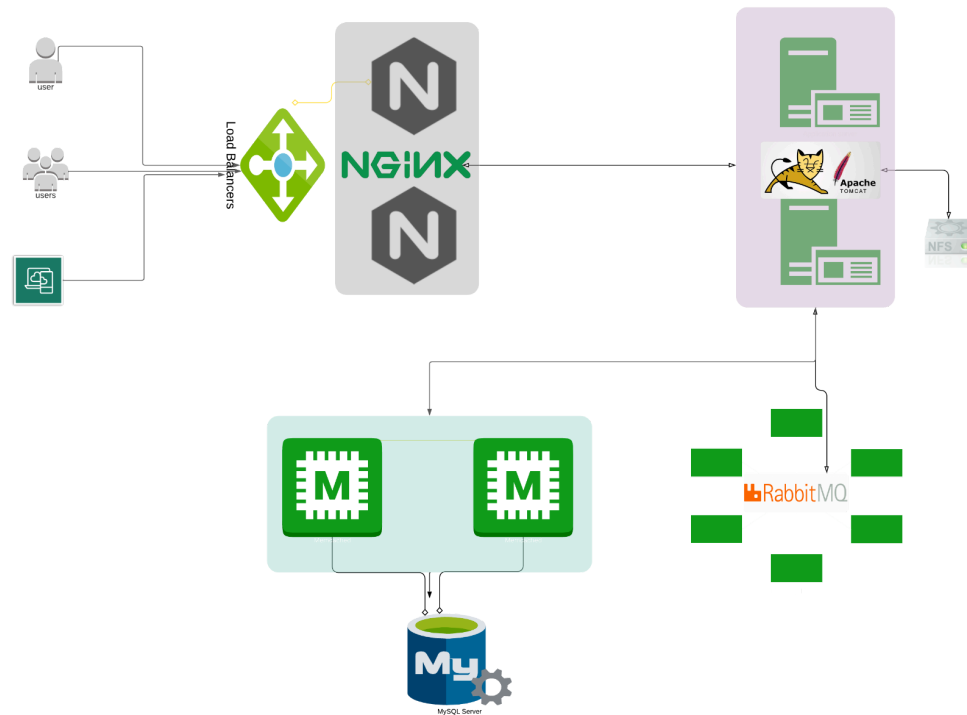
1. Clone source code.
2. cd into the repository.
3. Switch to the local branch.
4. cd into vagrant/Manual\_provisioning

Bring up vm's

```
$ vagrant up
```

**NOTE:** Bringing up all the vm's may take a long time based on various factors.  
If vm setup stops in the middle run "vagrant up" command again.

**INFO:** All the vm's hostname and /etc/hosts file entries will be automatically updated.



## PROVISIONING

### Services

1. Nginx               => Web Service
2. Tomcat             => Application Server
3. RabbitMQ           => Broker/Queueing Agent
4. Memcache          => DB Caching
5. Elasticsearch     => Indexing/Search service
6. MySQL             => SQL Database

Setup should be done in below mentioned order

```
MySQL    (Database SVC)
Memcache (DB Caching SVC)
RabbitMQ (Broker/Queue SVC)
Tomcat   (Application SVC)
Nginx    (Web SVC)
```

# 1. MYSQL Setup

Login to the db vm

```
$ vagrant ssh db01  
$ sudo -i
```

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

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

Update OS with latest patches

```
# dnf update -y
```

Set Repository

```
# dnf install epel-release -y
```

Install Maria DB Package

```
# dnf install git mariadb-server -y
```

Starting & enabling mariadb-server

```
# systemctl start mariadb  
# systemctl enable mariadb
```

RUN mysql secure installation script.

```
# mysql_secure_installation
```

**NOTE:** Set db root password, I will be using **admin123** as password

```
Set root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

By default, a MariaDB installation has an anonymous user, allowing anyone to log into MariaDB without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

```
Remove anonymous users? [Y/n] Y
... Success!
```

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

```
Disallow root login remotely? [Y/n] n
... skipping.
```

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

```
Remove test database and access to it? [Y/n] Y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!
```

Reloading the privilege tables will ensure that all changes made so far

```
will take effect immediately.
```

```
Reload privilege tables now? [Y/n] Y  
... Success!
```

## Set DB name and users.

```
# mysql -u root -padmin123
```

```
mysql> create database accounts;  
mysql> grant all privileges on accounts.* TO 'admin'@'%' identified by 'admin123' ;  
mysql> grant all privileges on accounts.* TO 'admin'@'%'localhost' identified by 'admin123' ;  
mysql> FLUSH PRIVILEGES;  
mysql> exit;
```

## Download Source code & Initialize Database.

```
# cd /tmp/  
# git clone -b local https://github.com/hkhcoder/vprofile-project.git  
# cd vprofile-project  
# mysql -u root -padmin123 accounts < src/main/resources/db_backup.sql  
# mysql -u root -padmin123 accounts
```

```
mysql> show tables;  
mysql> exit;
```

## Restart mariadb-server

```
# systemctl restart mariadb
```

## Starting the firewall and allowing the mariadb to access from port no. 3306

```
# systemctl start firewalld  
# systemctl enable firewalld  
# firewall-cmd --get-active-zones  
# firewall-cmd --zone=public --add-port=3306/tcp --permanent  
# firewall-cmd --reload  
# systemctl restart mariadb
```

## 2. MEMCACHE SETUP

Login to the Memcache vm

```
$ vagrant ssh mc01  
$ sudo -i
```

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

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

Install, start & enable memcache on port 11211

```
# sudo dnf update -y  
# sudo dnf install memcached -y  
# sudo systemctl start memcached  
# sudo systemctl enable memcached  
# sudo systemctl status memcached  
# sed -i 's/127.0.0.1/0.0.0.0/g' /etc/sysconfig/memcached  
# sudo systemctl restart memcached
```

Starting the firewall and allowing the port 11211 to access memcache

```
# systemctl start firewalld  
# systemctl enable firewalld  
# firewall-cmd --add-port=11211/tcp  
# firewall-cmd --runtime-to-permanent  
# firewall-cmd --add-port=11111/udp  
# firewall-cmd --runtime-to-permanent  
# sudo memcached -p 11211 -U 11111 -u memcached -d
```

### 3.RABBITMQ SETUP

Login to the RabbitMQ vm

```
$ vagrant ssh rmq01
```

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

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

Update OS with latest patches

```
# dnf update -y
```

Install Dependencies

```
# sudo dnf install wget -y
# dnf -y install centos-release-rabbitmq-38
# dnf --enablerepo=centos-rabbitmq-38 -y install rabbitmq-server
# systemctl enable --now rabbitmq-server
```

Setup access to user test and make it admin

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

Starting the firewall and allowing the port 5672 to access rabbitmq

```
# sudo systemctl start firewalld
# sudo systemctl enable firewalld
# firewall-cmd --add-port=5672/tcp
# firewall-cmd --runtime-to-permanent
# sudo systemctl start rabbitmq-server
# sudo systemctl enable rabbitmq-server
# sudo systemctl status rabbitmq-server
```

## 4.TOMCAT SETUP

Login to the tomcat vm

```
$ vagrant ssh app01
```

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

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

Update OS with latest patches

```
# dnf update -y
```

Install Dependencies

```
# dnf -y install java-17-openjdk java-17-openjdk-devel
```

```
# dnf install git wget -y
```

Change dir to /tmp

```
# cd /tmp/
```

Download & Tomcat Package

```
# wget  
https://archive.apache.org/dist/tomcat/tomcat-10/v10.1.26/bin/apache-tomcat-10.1.26.tar.gz
```

```
# tar xzvf apache-tomcat-10.1.26.tar.gz
```

Add tomcat user

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



Copy data to tomcat home dir

```
# cp -r /tmp/apache-tomcat-10.1.26/* /usr/local/tomcat/
```

Make tomcat user owner of tomcat home dir

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

Setup systemctl command for tomcat

Create tomcat service file

```
# vi /etc/systemd/system/tomcat.service
```

Update the file with below content

```
[Unit]
Description=Tomcat
After=network.target

[Service]
User=tomcat
Group=tomcat
WorkingDirectory=/usr/local/tomcat
Environment=JAVA_HOME=/usr/lib/jvm/jre
Environment=CATALINA_PID=/var/tomcat/%i/run/tomcat.pid
Environment=CATALINA_HOME=/usr/local/tomcat
Environment=CATALINE_BASE=/usr/local/tomcat
ExecStart=/usr/local/tomcat/bin/catalina.sh run
ExecStop=/usr/local/tomcat/bin/shutdown.sh
RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target
```

## Reload systemd files

```
# systemctl daemon-reload
```

## Start & Enable service

```
# systemctl start tomcat  
# systemctl enable tomcat
```

## Enabling the firewall and allowing port 8080 to access the tomcat

```
# systemctl start firewalld  
# systemctl enable firewalld  
# firewall-cmd --get-active-zones  
# firewall-cmd --zone=public --add-port=8080/tcp --permanent  
# firewall-cmd --reload
```

## CODE BUILD & DEPLOY (app01)

### Maven Setup

```
# cd /tmp/

# wget
https://archive.apache.org/dist/maven/maven-3/3.9.9/binaries/apache-maven-3.9.9-bin.zip

# unzip apache-maven-3.9.9-bin.zip
# cp -r apache-maven-3.9.9 /usr/local/maven3.9
# export MAVEN_OPTS="-Xmx512m"
```

### Download Source code

```
# git clone -b local https://github.com/hkhcoder/vprofile-project.git
```

### Update configuration

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

### Build code

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

```
# /usr/local/maven3.9/bin/mvn install
```

### Deploy artifact

```
# systemctl stop tomcat
```

```
# rm -rf /usr/local/tomcat/webapps/ROOT*
# cp target/vprofile-v2.war /usr/local/tomcat/webapps/ROOT.war
# systemctl start tomcat
# chown tomcat:tomcat /usr/local/tomcat/webapps -R
# systemctl restart tomcat
```

## 5.NGINX SETUP

Login to the Nginx vm

```
$ vagrant ssh web01  
$ sudo -i
```

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

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

Update OS with latest patches

```
# sudo apt update  
# sudo apt upgrade -y
```

Install nginx

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

Create Nginx conf file

```
# vi /etc/nginx/sites-available/vproapp
```

Update with below content

```
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
```

