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

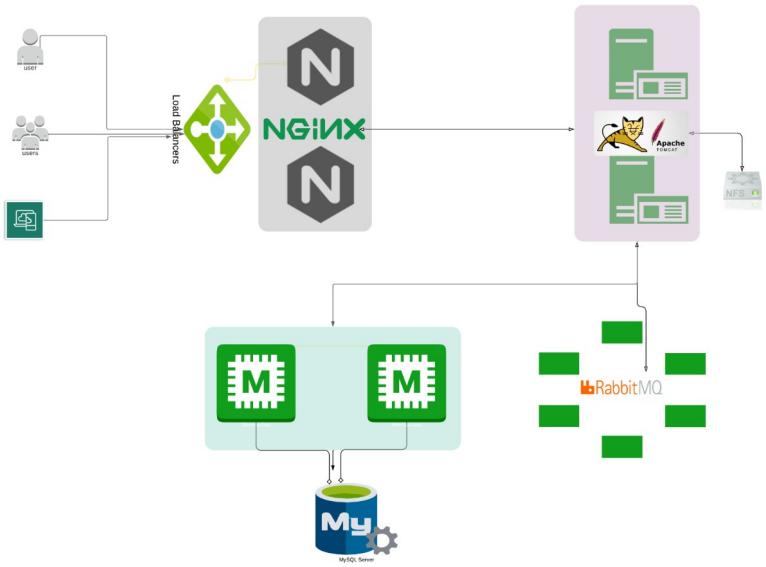
1. Clone source code.
2. Cd into the repository.
3. Switch to the main branch.
4. cd into vagrant/Manual_provisioning

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/Queuing 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
```

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

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

Update OS with latest patches

```
# yum update -y
```

Set Repository

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

Install Maria DB Package

```
# yum 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> FLUSH PRIVILEGES;
mysql> exit;
```

Download Source code & Initialize Database.

```
# git clone -b main 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
```