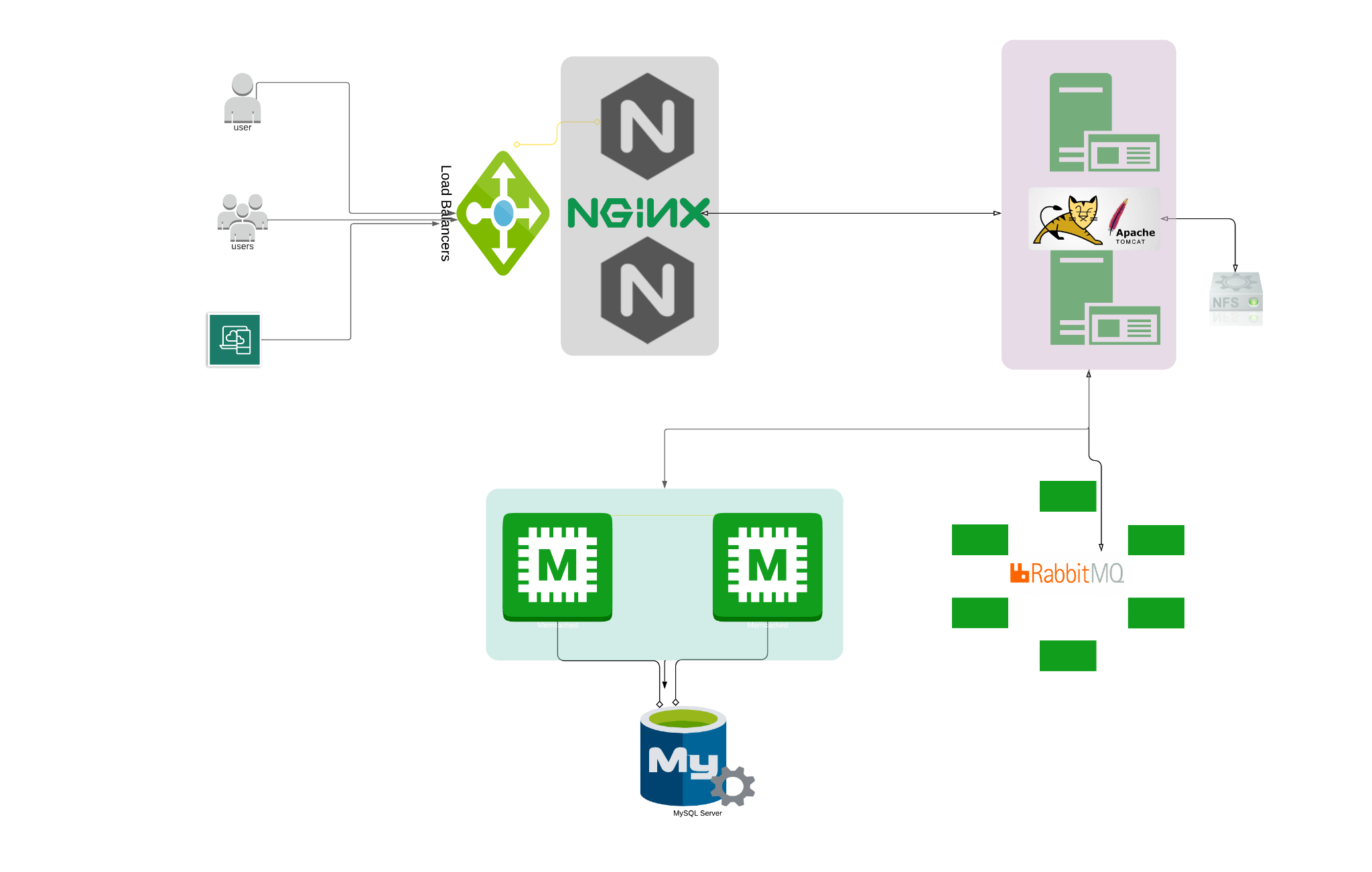
**1. Multi Tier Web Application Setup - Locally**

The agenda of this Project is to host a multitier web application stack on my local machine.

* Project Architecture (Services) :

**

* Project Execution:

1. VM Setup
2. Application Setup

**1.VM Setup**

Vagrant Setup :

1. Install Vagrant : <https://www.vagrantup.com/downloads>

Check if vagrant is installed

$ vagrant --version

1. Create a new Folder named as Vagrant

On terminal go inside the Vagrant folder and execute below command to initialize

$ vagrant init

Oracle VirtualBox Setup:

1. Install VirtualBox : <https://www.virtualbox.org/wiki/Downloads>

*NOTE:* A box is a pre-configured virtual machine image that you can use as a starting point for your virtual machine

Choose a box to use <https://app.vagrantup.com/boxes/search>

*NOTE:* Refer to Vagrant word Doc inside the Notes/Vagrant for deep dive into Vagrant

# VM SETUP

1. Write & Place the ‘vagrantfile’ inside the initialized Vagrant directory
2. Execute below command to install Vagrant Plugin

$ vagrant plugin install vagrant-hostmanager

1. 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.

1. **Application Setup**

**Services**

1. Nginx
2. Tomcat
3. RabbitMQ
4. Memcache

=> Web Service

=> Application Server

=> Broker/Queuing Agent

=> DB Caching

1. ElasticSearch => Indexing/Search service
2. 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)**

**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

Switch to root user

# sudo su

Update OS with latest patches

# yum update -y

Set DB password

# DATABASE\_PASS=admin123

Check DB password

# echo $DATABASE\_PASS

Set Repository

# yum install epel-release -y

Install Maria DB Package

# yum install git mariadb-server -y

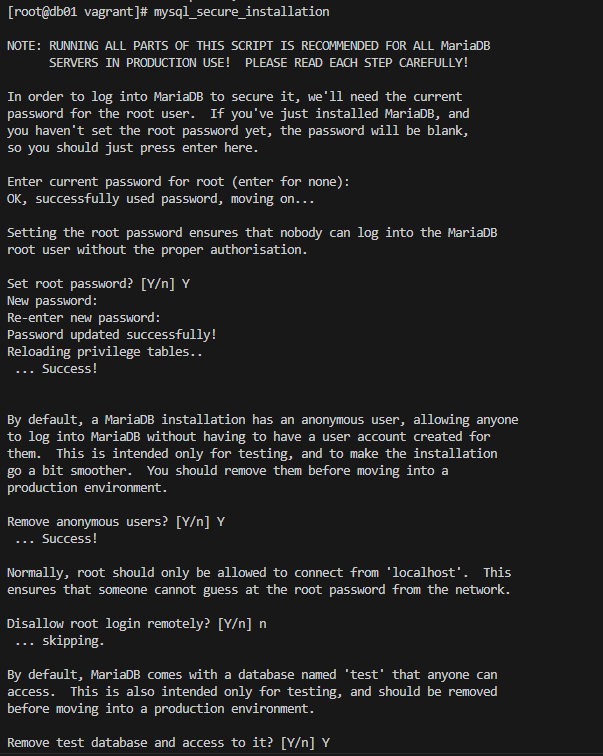
Starting & enabling mariadb-server

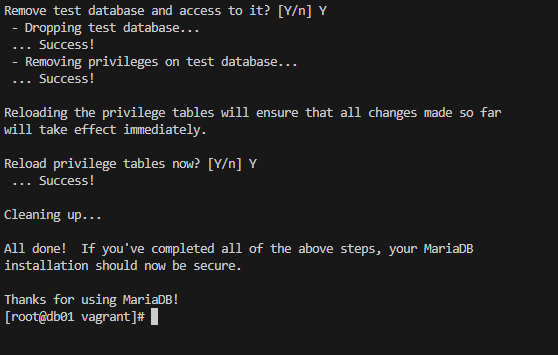
# systemctl start mariadb # systemctl enable mariadb

RUN mysql secure installation script.

# mysql\_secure\_installation

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





Set 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

**MEMCACHE SETUP**

Login to Memcache vm

$ vagrant ssh mc01

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

# cat /etc/hosts

Update OS with latest patches

# yum update -y

Install, start & enable memcache on port 11211

# sudo dnf install epel-release -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

# 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