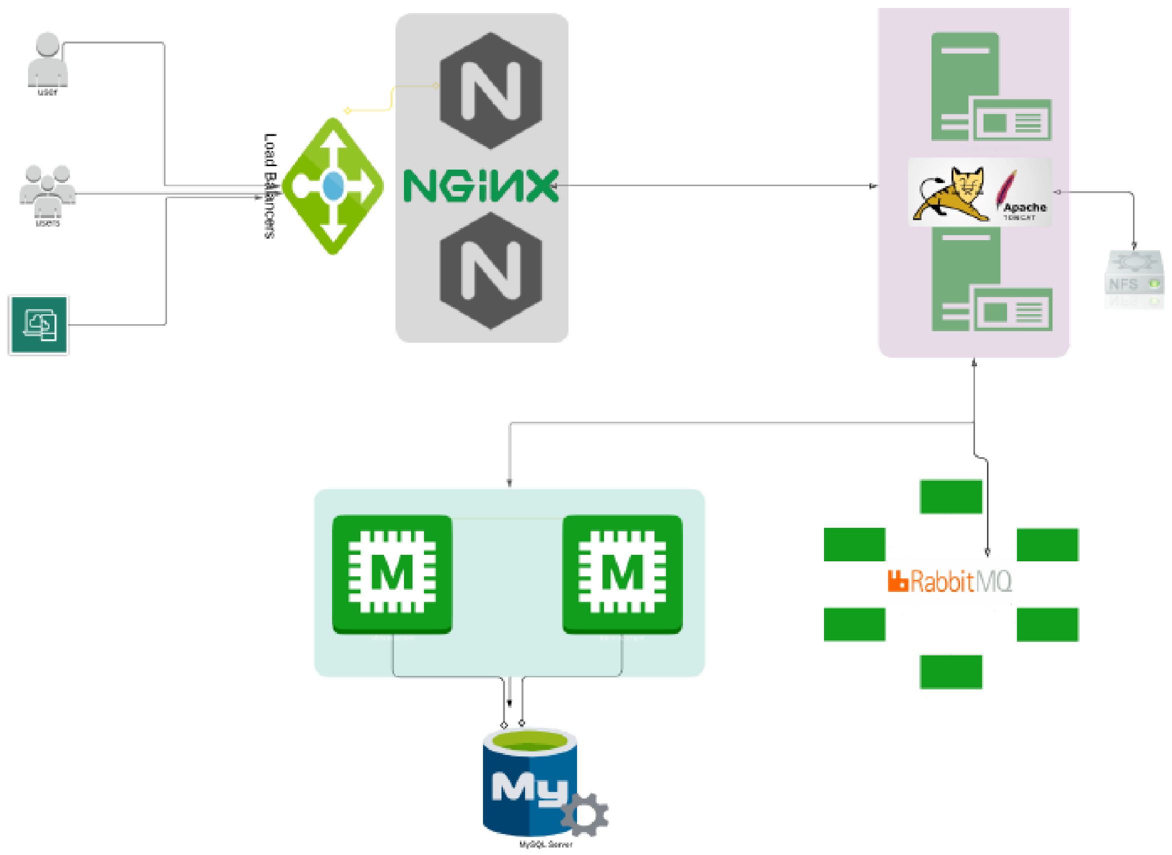
VPROFILE PROJECT

In this Project we have a Java App Source code, and we will build a complete architecture for this App, also we will build "War File" for this App, then we will deploy it on an Application web server (Apache Tomcat).



- **Tools we need for this project.**

1. **Oracle VM Virtualbox:** (Oracle VM VirtualBox is open source virtualization software "Hypervisor" that allows users to run multiple operating system on a single device.)

2. **Vagrant:** (Vagrant is a software product for building and maintaining portable virtual software development environments. For example, for VirtualBox, KVM, Hyper-V, Docker, VMware, and AWS containers. It attempts to simplify software provisioning and configuration management for virtualization in order to increase development productivity. This program gives us the ability to completely build our infrastructure using any number of virtual machines with different types of operating systems through a single file “Vagrantfile”. We can format each machine with its specifications in this file.)

3. **MySQL Database (MariaDB):** (MariaDB is an open source and forever free MySQL alternative that provides greater efficiency, improved database performance, and support for different data types through multiple storage engines.)

4. **Memcached (DB Caching):** (Memcached can serve cached items in less than a millisecond, and enables you to scale easily and cost-effectively to higher loads. Memcached is known for caching database query results, session caching, web page caching, API caching, and caching of objects such as images, files, and metadata.)

5. **RabbitMQ (message-broker):** (The RabbitMQ broker is an open-source message broker. It's a popular open-source message queue software that is written in Erlang. It supports several API protocols such as AMQP, STOMP, MQTT, and HTTP. Additionally, RabbitMQ supports a lot of the common programming languages and can run on different cloud environments and operating systems. It's a message queue tool that connects applications to exchange messages.)

6. **Apache Tomcat (Application Web Server):** (Tomcat is an open source web server and servlet. It was developed by the Apache Software Foundation. It is widely used to host Java-based applications on the web. It is built on Java technologies and implements the Java Servlet and JavaServer Pages (JSP) specifications. Tomcat acts as a bridge between web servers and Java-based applications, facilitating the execution of dynamic content and processing of client requests. Tomcat offers a lightweight and powerful solution for hosting Java web applications. Tomcat is popular for several reasons. One of them is scalability. It has great community support, which is a relief for users. With its modular architecture, Tomcat enables developers to easily create and deploy web applications.)

7. **Nginx (Web Service):** (NGINX is open source software for web serving, reverse proxying, caching, load balancing, media streaming, and more. It started out as a web server designed for maximum performance and stability. In addition to its HTTP server capabilities, NGINX can also function as a proxy server for email (IMAP, POP3, and SMTP) and a reverse proxy and load balancer for HTTP, TCP, and UDP servers.)

   Note: in this project we will using **Nginx** as load balancer to receive requests from users and forward it to the application server (**Apache Tomcat**).

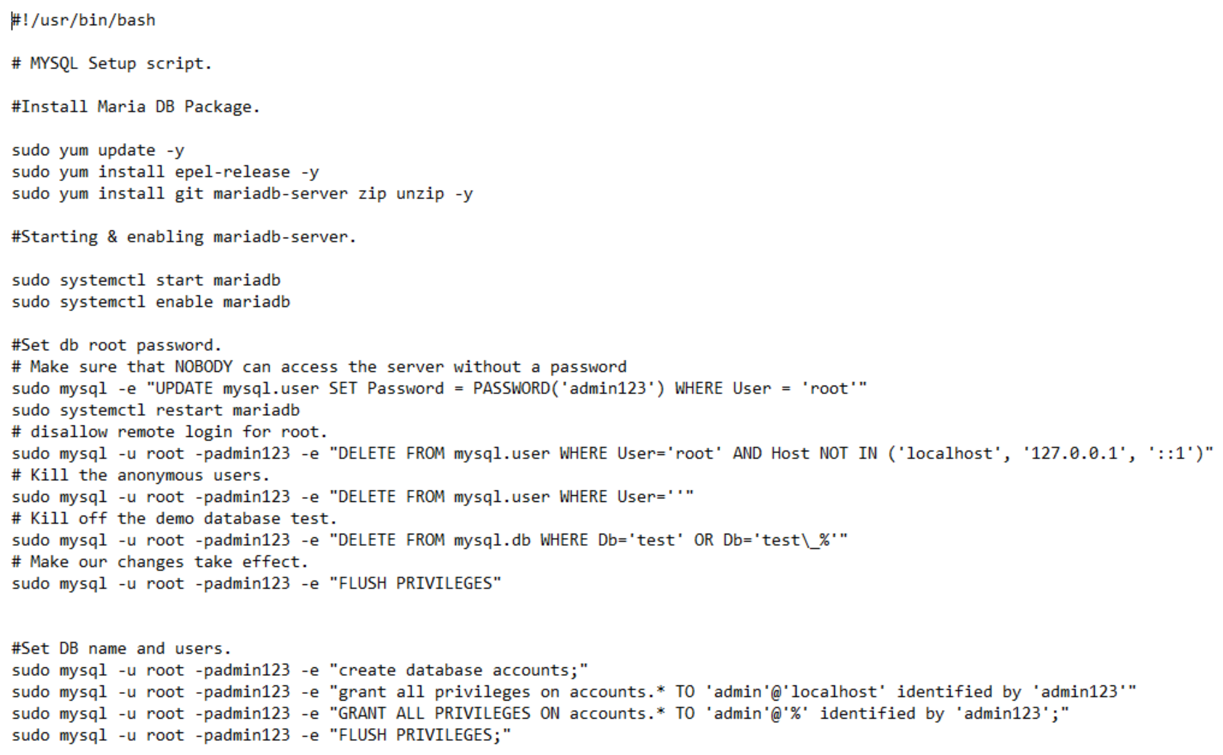
- **Steps**:
  1. First, we need to create 5 Vms (**MYSQL** Vm, **MEMCACHE** Vm, **RABBITMQ** Vm, **TOMCAT** Vm, **NGINX** Vm) using Vagrant Software.
  2. Therefore, we will prepare a Vagrantfile with the necessary specifications and settings for each machine we want to create, such as the version and type of the operating system that each machine needs, the network settings and IP address for each machine, the necessary storage space and memory space.
  3. After that, we will write 5 scripts for the internal settings and configurations necessary for each machine, such as the softwares “packages” needed to

be installed on each machine and the necessary updates for each machine in order to prepare the complete infrastructure of the project.

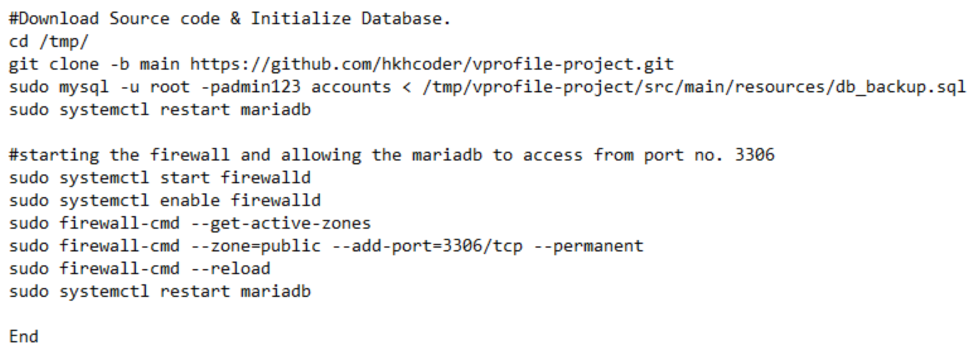
4. Finally, we will add these five scripts to our Vagrantfile, and then we will run this file using the command "Vagrant up".

$~ Lab Setup ~$

➢ MYSQL Setup Script:

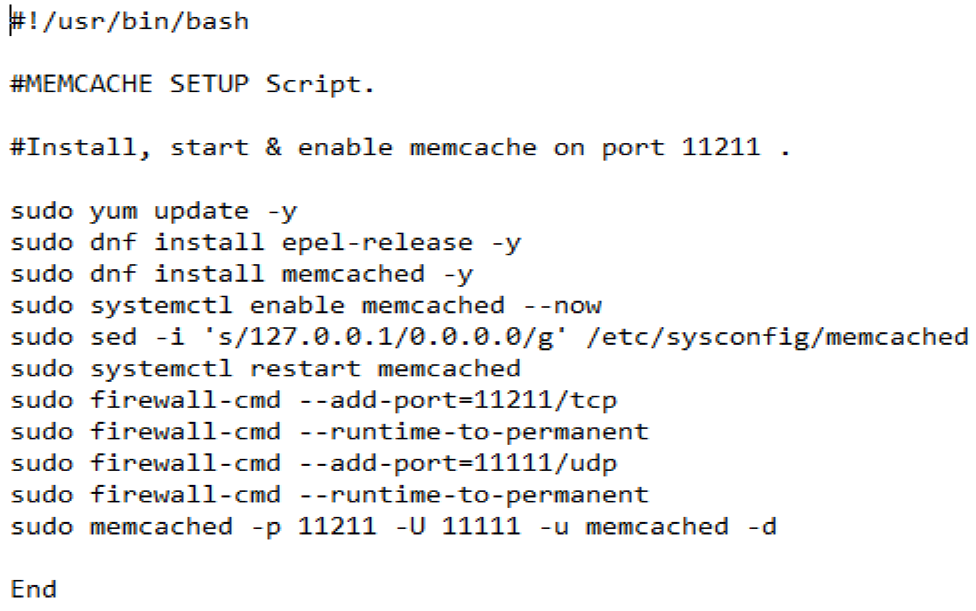


```
#!/usr/bin/bash

# MYSQL Setup script.

#Install Maria DB Package.

sudo yum update -y
sudo yum install epel-release -y
sudo yum install git mariadb-server zip unzip -y

#Starting & enabling mariadb-server.

sudo systemctl start mariadb
sudo systemctl enable mariadb

#Set db root password.
# Make sure that NOBODY can access the server without a password
sudo mysql -e "UPDATE mysql.user SET Password = PASSWORD('admin123') WHERE User = 'root'"
sudo systemctl restart mariadb
# disallow remote login for root.
sudo mysql -u root -padmin123 -e "DELETE FROM mysql.user WHERE User='root' AND Host NOT IN ('localhost', '127.0.0.1', '::1')"
# Kill the anonymous users.
sudo mysql -u root -padmin123 -e "DELETE FROM mysql.user WHERE User=''"
# Kill off the demo database test.
sudo mysql -u root -padmin123 -e "DELETE FROM mysql.db WHERE Db='test' OR Db='test\_%'"
# Make our changes take effect.
sudo mysql -u root -padmin123 -e "FLUSH PRIVILEGES"


#Set DB name and users.
sudo mysql -u root -padmin123 -e "create database accounts;"
sudo mysql -u root -padmin123 -e "grant all privileges on accounts.* TO 'admin'@'localhost' identified by 'admin123'"
sudo mysql -u root -padmin123 -e "GRANT ALL PRIVILEGES ON accounts.* TO 'admin'@'%' identified by 'admin123';"
sudo mysql -u root -padmin123 -e "FLUSH PRIVILEGES;"
```



```
#Download Source code & Initialize Database.
cd /tmp/
git clone -b main https://github.com/hkhcoder/vprofile-project.git
sudo mysql -u root -padmin123 accounts < /tmp/vprofile-project/src/main/resources/db_backup.sql
sudo systemctl restart mariadb

#starting the firewall and allowing the mariadb to access from port no. 3306
sudo systemctl start firewalld
sudo systemctl enable firewalld
sudo firewall-cmd --get-active-zones
sudo firewall-cmd --zone=public --add-port=3306/tcp --permanent
sudo firewall-cmd --reload
sudo systemctl restart mariadb

End
```

➢ MEMCACHE Setup Script :



```
#!/usr/bin/bash

#MEMCACHE SETUP Script.

#Install, start & enable memcache on port 11211 .

sudo yum update -y
sudo dnf install epel-release -y
sudo dnf install memcached -y
sudo systemctl enable memcached --now
sudo sed -i 's/127.0.0.1/0.0.0.0/g' /etc/sysconfig/memcached
sudo systemctl restart memcached
sudo firewall-cmd --add-port=11211/tcp
sudo firewall-cmd --runtime-to-permanent
sudo firewall-cmd --add-port=11111/udp
sudo firewall-cmd --runtime-to-permanent
sudo memcached -p 11211 -U 11111 -u memcached -d

End
```

➢ RABBITMQ Setup Script :

```
#!/usr/bin/bash

# RABBITMQ Setup Script.

#Install Dependencies.
sudo yum update -y
sudo yum install epel-release -y
sudo yum install wget -y

sudo yum -y install rabbitmq-server
sudo yum install erlang -y
sudo 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
sudo systemctl restart rabbitmq-server

# Starting the firewall and allowing the port 5672 to access rabbitmq.
sudo systemctl start firewalld
sudo systemctl enable firewalld
sudo firewall-cmd --add-port=5672/tcp
sudo firewall-cmd --runtime-to-permanent
sudo systemctl start rabbitmq-server
sudo systemctl enable rabbitmq-server
sudo systemctl status rabbitmq-server

End
```

➢ TOMCAT Setup Script :

```
#!/usr/bin/bash

# TOMCAT Setup script.

# Install Dependencies.
sudo yum update -y
sudo dnf install epel-release -y
sudo dnf -y install java-17-openjdk java-17-openjdk-devel
sudo dnf install git maven wget -y

# Download & Tomcat Package.
cd /tmp/
sudo wget https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.75/bin/apache-tomcat-9.0.75.tar.gz
sudo dnf install tar -y
sudo tar xzvf apache-tomcat-9.0.75.tar.gz
sudo useradd --home-dir /usr/local/tomcat --shell /sbin/nologin tomcat
sudo cp -r /tmp/apache-tomcat-9.0.75/* /usr/local/tomcat/
sudo chown -R tomcat.tomcat /usr/local/tomcat
rm -rf /etc/systemd/system/tomcat.service
```

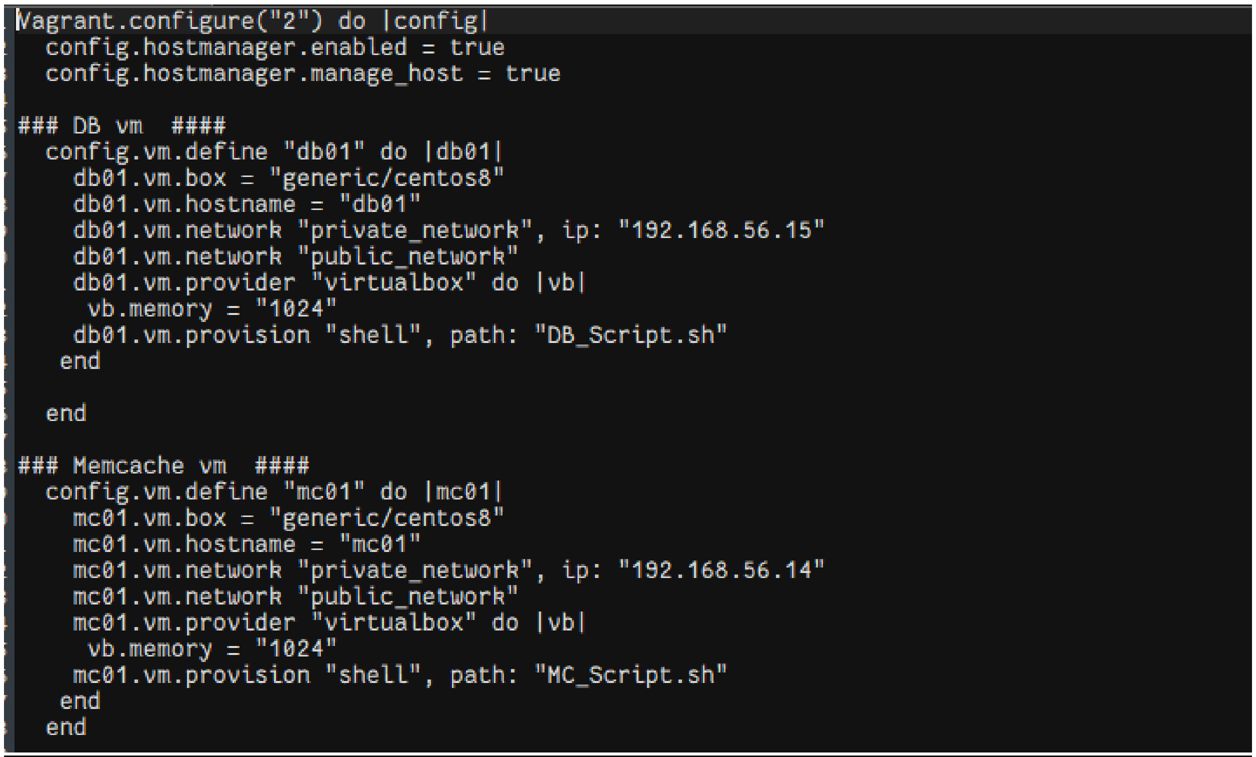
```
# Create tomcat service file.
sudo echo "[Unit]
Description=Tomcat
After=network.target
[Service]
User=tomcat
WorkingDirectory=/usr/local/tomcat
Environment=JRE_HOME=/usr/lib/jvm/jre
Environment=JAVA_HOME=/usr/lib/jvm/jre
Environment=CATALINA_HOME=/usr/local/tomcat
Environment=CATALINE_BASE=/usr/local/tomcat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

sudo systemctl daemon-reload
sudo systemctl start tomcat
sudo systemctl enable tomcat

# CODE BUILD & DEPLOY (app01).

git clone -b main https://github.com/hkhcoder/vprofile-project.git
cd vprofile-project
sudo mvn install
sudo systemctl stop tomcat
sleep 20
sudo rm -rf /usr/local/tomcat/webapps/ROOT*
sudo cp target/vprofile-v2.war /usr/local/tomcat/webapps/ROOT.war
sudo systemctl start tomcat
sleep 20
sudo chown tomcat.tomcat /usr/local/tomcat/webapps -R
sudo systemctl stop firewalld
sudo systemctl disable firewalld
sudo systemctl restart tomcat
```

End

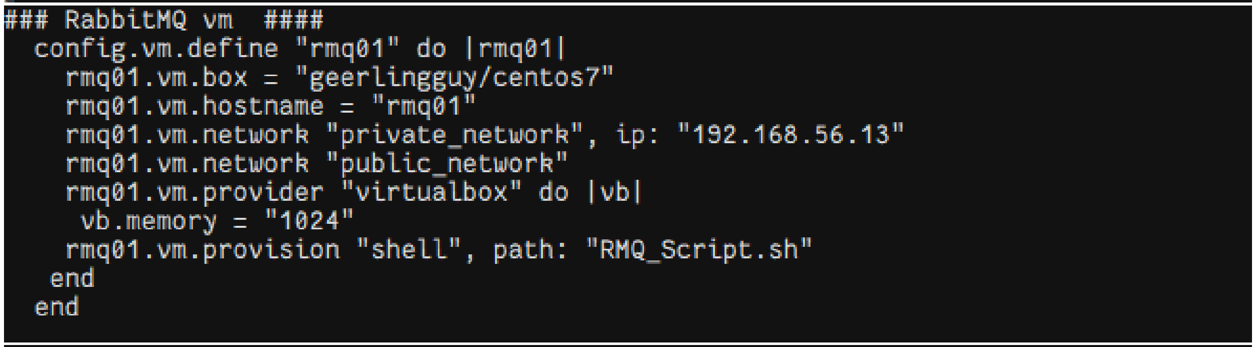
➢ NGINX Setup Script :

```
#!/usr/bin/bash

# NGINX Setup script.

# Install nginx .
sudo apt-get update -y
sudo apt-get upgrade -y
sudo apt-get install nginx -y

# Create Nginx conf file.

sudo echo "upstream vproapp {
server app01:8080;
}
server {
listen 80;
location / {
proxy_pass http://vproapp;
}
}" >> /etc/nginx/sites-available/vproapp
sudo rm -rf /etc/nginx/sites-enabled/default
sudo ln -s /etc/nginx/sites-available/vproapp /etc/nginx/sites-enabled/vproapp

sudo systemctl start nginx
sudo systemctl enable nginx
sudo systemctl restart nginx

End
```

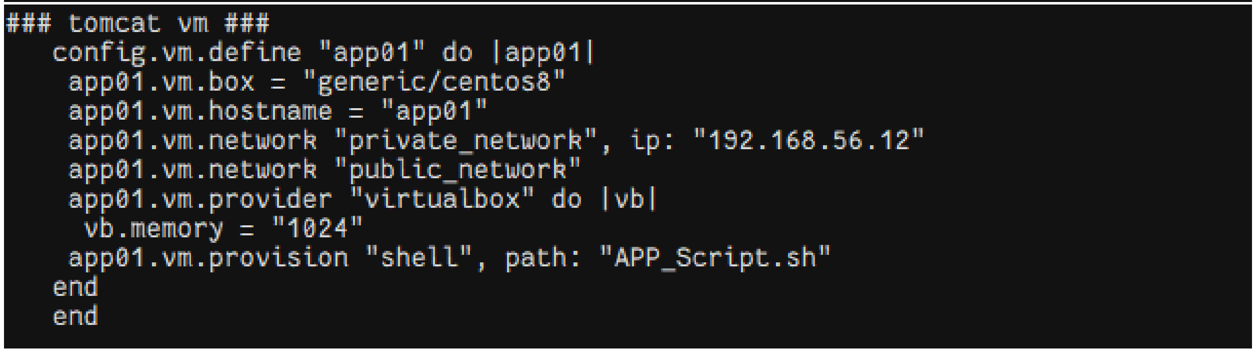
➢ Vagrantfile :



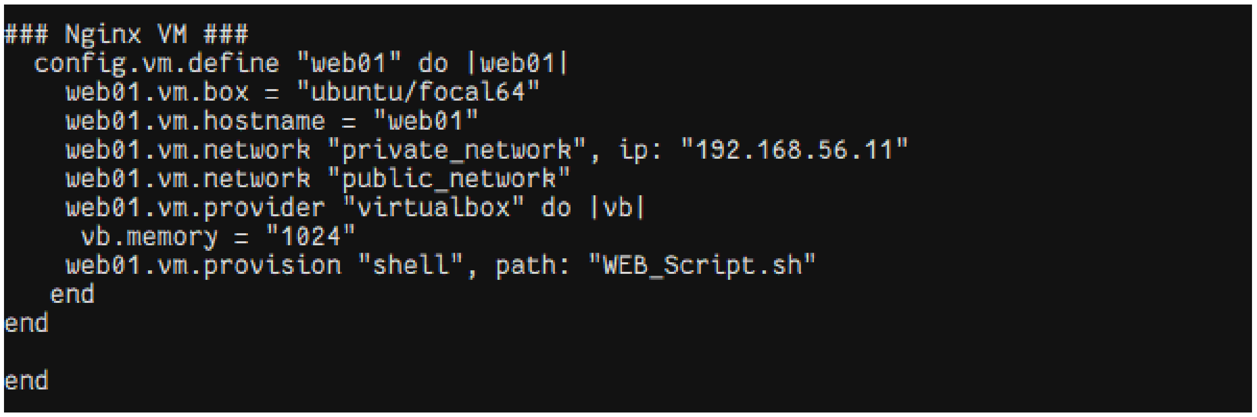
```
Vagrant.configure("2") do |config|
  config.hostmanager.enabled = true
  config.hostmanager.manage_host = true

### DB vm  ####
  config.vm.define "db01" do |db01|
    db01.vm.box = "generic/centos8"
    db01.vm.hostname = "db01"
    db01.vm.network "private_network", ip: "192.168.56.15"
    db01.vm.network "public_network"
    db01.vm.provider "virtualbox" do |vb|
     vb.memory = "1024"
    db01.vm.provision "shell", path: "DB_Script.sh"
   end

  end

### Memcache vm  ####
  config.vm.define "mc01" do |mc01|
    mc01.vm.box = "generic/centos8"
    mc01.vm.hostname = "mc01"
    mc01.vm.network "private_network", ip: "192.168.56.14"
    mc01.vm.network "public_network"
    mc01.vm.provider "virtualbox" do |vb|
     vb.memory = "1024"
    mc01.vm.provision "shell", path: "MC_Script.sh"
   end
  end
```



```
### RabbitMQ vm  ####
  config.vm.define "rmq01" do |rmq01|
    rmq01.vm.box = "geerlingguy/centos7"
    rmq01.vm.hostname = "rmq01"
    rmq01.vm.network "private_network", ip: "192.168.56.13"
    rmq01.vm.network "public_network"
    rmq01.vm.provider "virtualbox" do |vb|
     vb.memory = "1024"
    rmq01.vm.provision "shell", path: "RMQ_Script.sh"
   end
  end
```



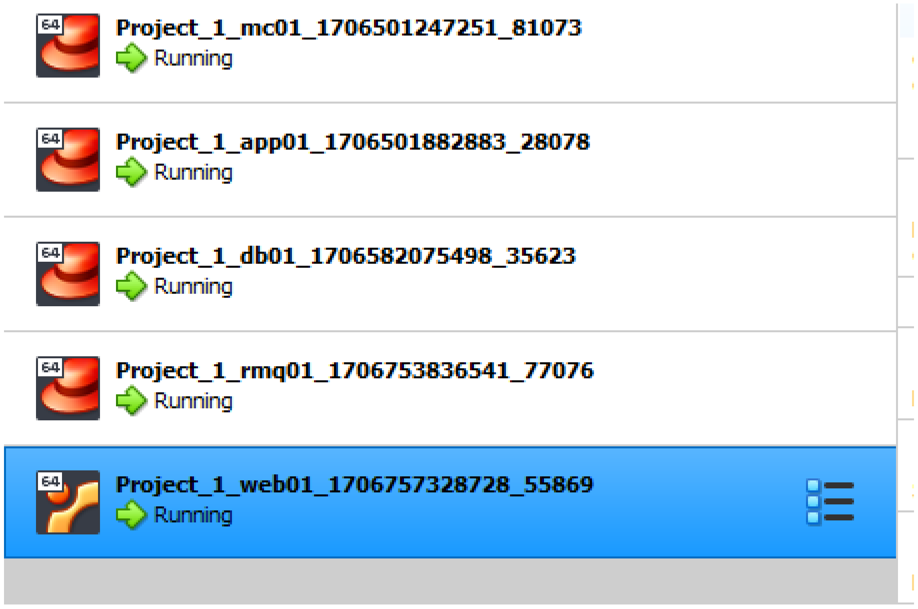
```
### tomcat vm ###
   config.vm.define "app01" do |app01|
    app01.vm.box = "generic/centos8"
    app01.vm.hostname = "app01"
    app01.vm.network "private_network", ip: "192.168.56.12"
    app01.vm.network "public_network"
    app01.vm.provider "virtualbox" do |vb|
     vb.memory = "1024"
    app01.vm.provision "shell", path: "APP_Script.sh"
   end
   end
```



```
### Nginx VM ###
  config.vm.define "web01" do |web01|
    web01.vm.box = "ubuntu/focal64"
    web01.vm.hostname = "web01"
    web01.vm.network "private_network", ip: "192.168.56.11"
    web01.vm.network "public_network"
    web01.vm.provider "virtualbox" do |vb|
     vb.memory = "1024"
    web01.vm.provision "shell", path: "WEB_Script.sh"
   end
end

end
```

- Now we can run the command "**vagrant up**" , but we must be in the same Vagrantfile location.

- After the creation and provisioning process is complete, we will have 5 virtual machines that contain the full specifications and configurations required for each machine, so that the services of each machine can work without any problems.



Project_1_mc01_1706501247251_81073
Running

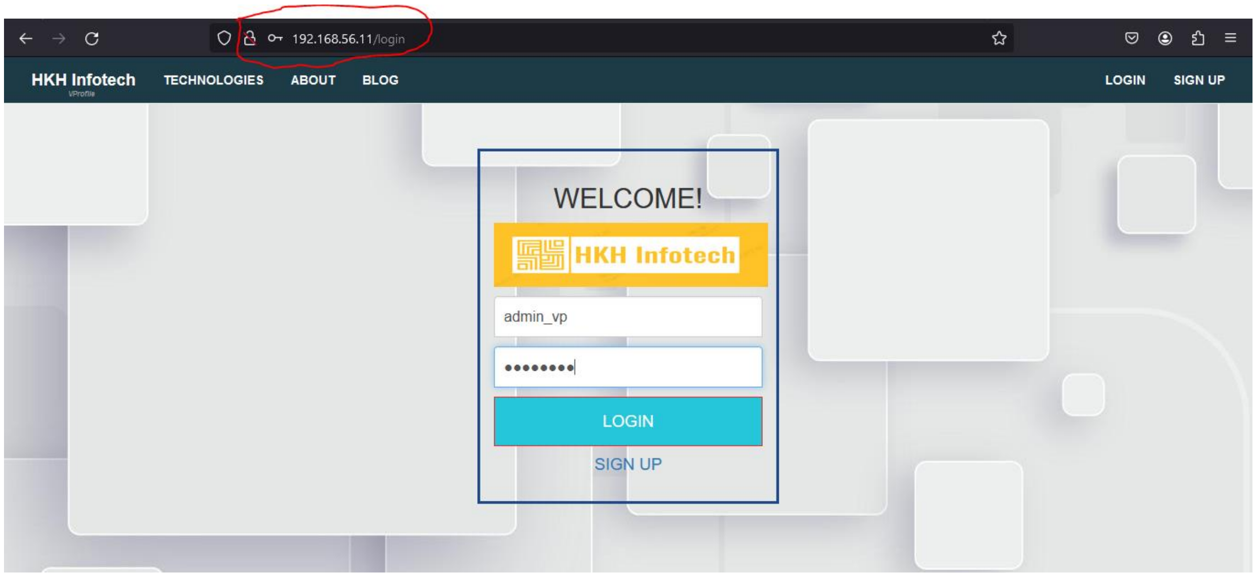
Project_1_app01_1706501882883_28078
Running

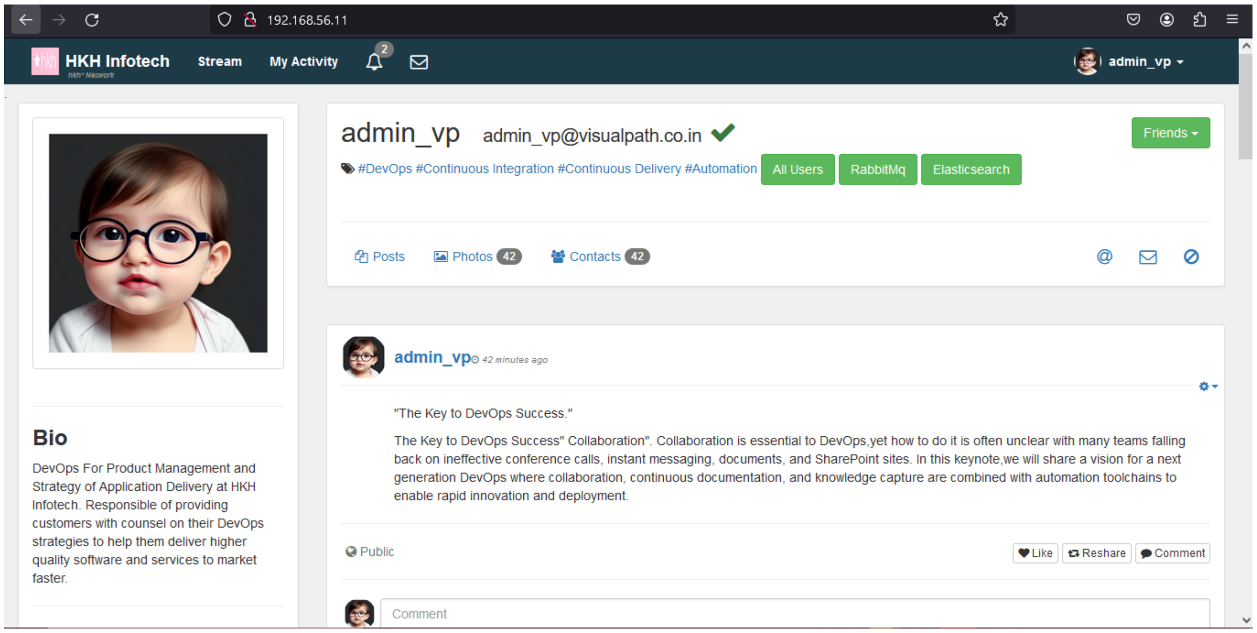
Project_1_db01_1706582075498_35623
Running

Project_1_rmq01_1706753836541_77076
Running

Project_1_web01_1706757328728_55869
Running

- We can access our App from any browser by typing our Web server IP “**Nginx**” (192.168.56.11).





admin_vp admin_vp@visualpath.co.in

#DevOps #Continuous Integration #Continuous Delivery #Automation

Bio

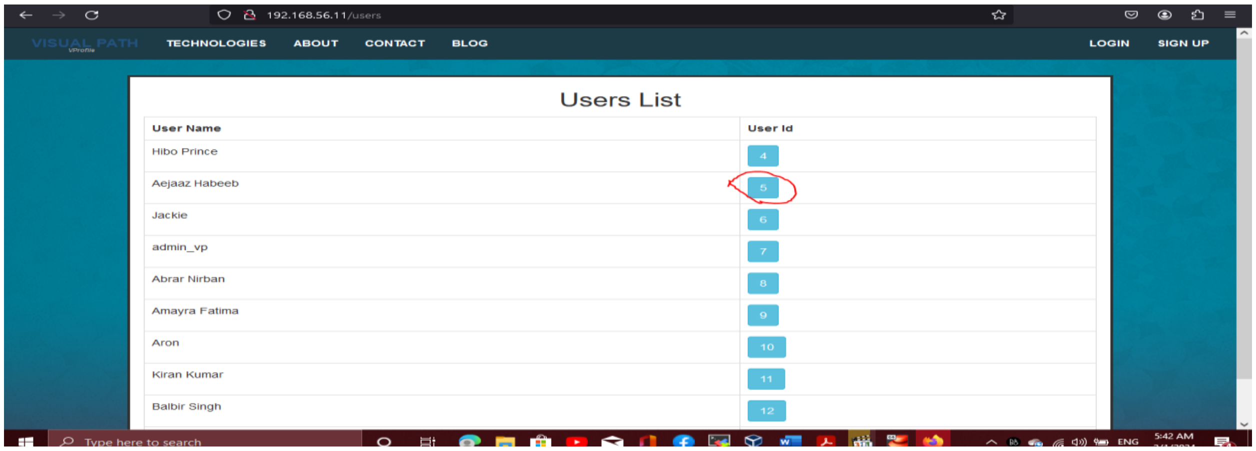
DevOps For Product Management and Strategy of Application Delivery at HKH Infotech. Responsible of providing customers with counsel on their DevOps strategies to help them deliver higher quality software and services to market faster.

admin_vp 42 minutes ago

"The Key to DevOps Success."

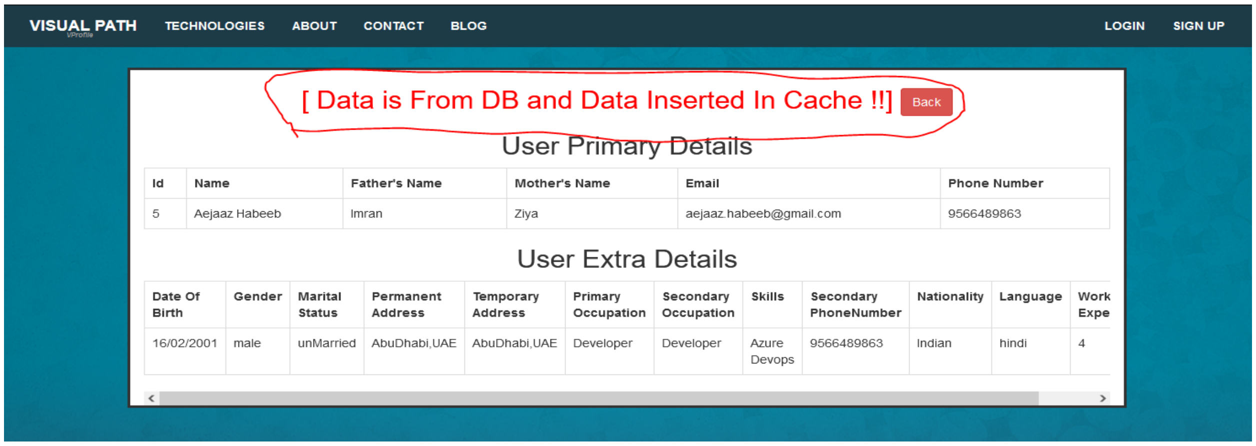
The Key to DevOps Success" Collaboration". Collaboration is essential to DevOps,yet how to do it is often unclear with many teams falling back on ineffective conference calls, instant messaging, documents, and SharePoint sites. In this keynote,we will share a vision for a next generation DevOps where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

- We will examine our **Memcached** service by access any user .



Users List

User Name	User Id
Hibo Prince	4
Aejaaz Habeeb	5
Jackie	6
admin_vp	7
Abrar Nirban	8
Amayra Fatima	9
Aron	10
Kiran Kumar	11
Balbir Singh	12



[Data is From DB and Data Inserted In Cache !!]

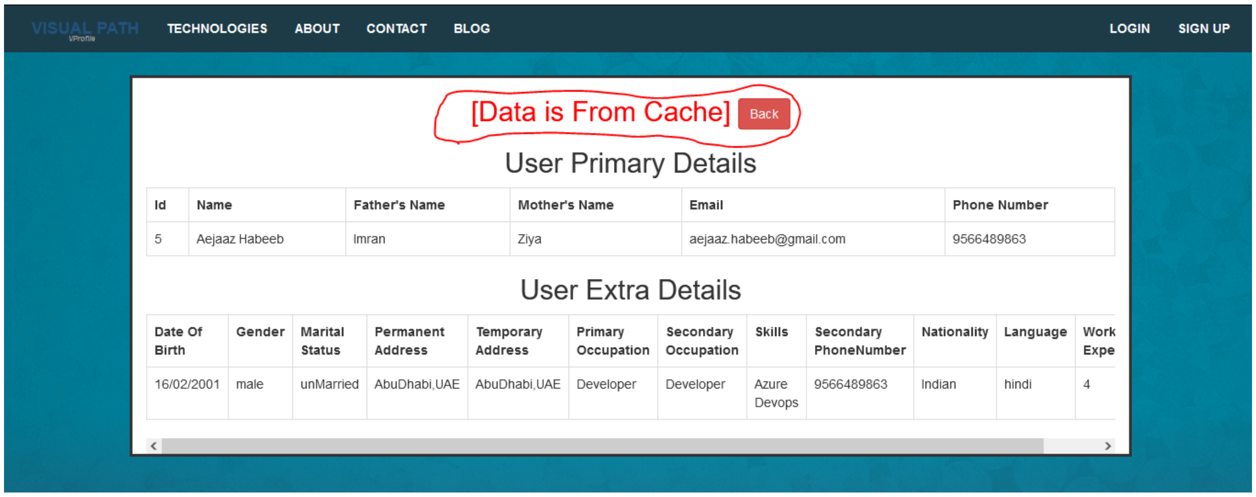
User Primary Details

Id	Name	Father's Name	Mother's Name	Email	Phone Number
5	Aejaaz Habeeb	Imran	Ziya	aejaaz.habeeb@gmail.com	9566489863

User Extra Details

Date Of Birth	Gender	Marital Status	Permanent Address	Temporary Address	Primary Occupation	Secondary Occupation	Skills	Secondary PhoneNumber	Nationality	Language	Work Expe
16/02/2001	male	unMarried	AbuDhabi,UAE	AbuDhabi,UAE	Developer	Developer	Azure Devops	9566489863	Indian	hindi	4

- Click on Back, then access this user again.



[Data is From Cache]

User Primary Details

Id	Name	Father's Name	Mother's Name	Email	Phone Number
5	Aejaaz Habeeb	Imran	Ziya	aejaaz.habeeb@gmail.com	9566489863

User Extra Details

Date Of Birth	Gender	Marital Status	Permanent Address	Temporary Address	Primary Occupation	Secondary Occupation	Skills	Secondary PhoneNumber	Nationality	Language	Work Expe
16/02/2001	male	unMarried	AbuDhabi,UAE	AbuDhabi,UAE	Developer	Developer	Azure Devops	9566489863	Indian	hindi	4

➢ Conclusion:

# In this project we have covered:

1. What is (**Nginx, Memcached, RabbitMQ, Tomcat, MariaDB, Vagrant, VirtualBox**).
2. How to use **Vagrant** tool to build, configure and provision a complete infrastructure for any Web App.
3. How to write a **Vagrantfile** to create any number of VM’S with them specs, and how to integrate a shell script into this **Vagrantfile** to automate the provisioning process.
4. How to install and configure (**MariaDB, Memcached, RabbitMQ, Tomcat, Nginx**) services.