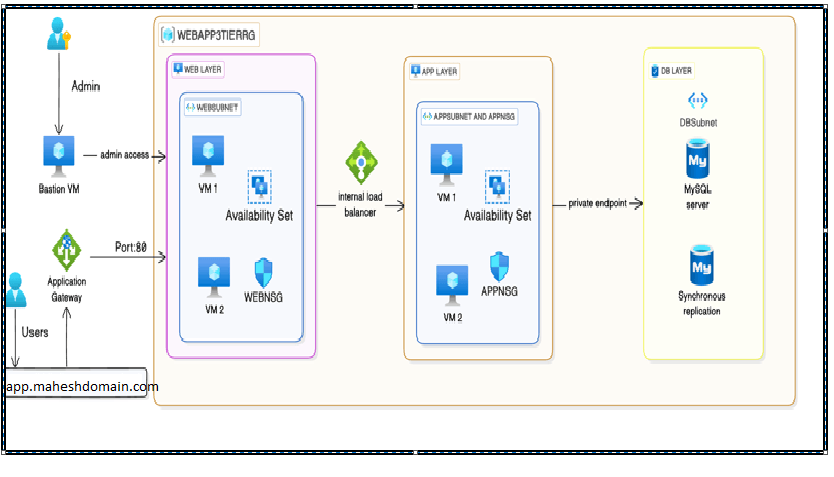
# 3-Tier Architecture in Azure Cloud



Web Tier (Front End): This is the user interface (UI) of the application. The web tier is responsible for interacting with the users and presenting the data retrieved from the app tier. In this case, it is built using React, a JavaScript library for building dynamic user interfaces.

Application Tier (Back End): The application tier handles business logic and data processing. It receives requests from the frontend and communicates with the database to retrieve or modify data.

Database Tier (Data Base): The database tier is responsible for storing and retrieving the data required by the application. In this project, a MySQL database is used to store user data, application configurations, and other necessary information.

 I have Deployed Tier 3 Web Application in Azure Cloud using :

* **Virtual Machine Scale Sets** [VMSS] for running web and app server.
* **Load Balancer** for internal traffic between web-server and app-server.
* **Azure Database for MySQL servers** for MySQL Database.
* **Application Gateway** for internet access to web-server port 80.
* **Virtual Network**, **Subnets**, **Network Security Group**.

**Web Application Tech stack**

* React
* Nodejs
* MySQL

**Frontend Installation**

**NOTE**: Install Nodejs and nginx on your System

1. Install dependency to run react application

cd application-code/web-tier

npm install

1. Make Changes in nginx.conf file for backend application

cd ..

vi nginx.conf

#proxy for internal lb

location /api/{

proxy\_pass http://[REPLACE-WITH-INTERNAL-LB-DNS]:80/;

}

1. Now let's Create Build Folder

npm run build

Now move content of build/\* to Nginx root location /var/www/html. Now start your nginx server with sudo systemctl start nginx, review you webpage on :80.

## Backend Installation

**NOTE** : Install Nodejs and mysql on your System

1. Install dependency to run Nodejs API

cd application-code/app-tier

npm install

1. Make Changes in DbConfig.js file

vi application-code/app-tier/DbConfig.js

module.exports = Object.freeze({

DB\_HOST : '<Enter hostname>',

DB\_USER : '<Enter Admin user>',

DB\_PWD : '<Enter Admin Passwd>',

DB\_DATABASE : 'webappdb'

});

1. Create webappdb database and create Transactions table on MySQL server

mysql -h <Database Hostname> -u <Admin User> -p<Admin Passwd> < db\_setup.sql

1. Run you Application

npm install -g pm2

pm2 start index.js

startup\_as\_process=$(pm2 startup | grep -o 'sudo env.\*')

eval "$startup\_as\_process"

pm2 save

Above Command will start Node Server on :4000. You can can health of your server by running

* curl http://localhost:4000/health for application health
* curl http://localhost:4000/transaction for database health

Now our application is ready !!

**Azure Cloud Services Used**

Resource Group: Logical container for Storing Tier 3 Web-Application resources.

Virtual Network: Isolated network for private communication between different tiers in different subnets.

Virtual Machine: Computing instance to demo run tier3 application. [Create an Image out of it]

Virtual Machine Scale Set: Deploy and manage a group of identical VM, providing high availability and scalability. [Used for Deploying Scale web and application server]

Azure Load Balancer: Layer 4 Load Balancer Distributes incoming network traffic across multiple virtual machines or instances, ensuring even distribution. Support TCP and UDP Protocols. [Used as Internal load balancer to allow Traffic from Web to App]

Azure Application Gateway: Layer 7 Load Balancer and web traffic management service, used to distribute traffic to web applications .Support HTTP, HTTPS, and HTTP2 Protocols. [Used as Internet-facing load balancer allowing access to Web Page]

Azure Database for MySQL Server: Fully managed service [PASS] that allows you to run MySQL databases in Azure without worrying about maintenance tasks. It provides automatic backups, security, monitoring, and easy scalability, making it ideal for building web apps, mobile app backend, and more.