Module 2 - Apache and Postgresql server modernization

Overview

In this workshop you will build a traditional two-tier Web service that uses Apache and Postgresql services which can be modernized using PaaS.

NOTE Ensure that you have updated the VM's using yum before running the following steps.

Setup Postgresql server and import database

1. Log into server created in Module 1, LinuxLabVM-CentOS-7-PostgreSQL, using putty as root. Type in the command to set the hostname of the server. Then reboot the server.

```
hostnamectl set-hostname postgresql
```

```
reboot
```

NOTE Changing the host name will make it easier to identify if you open several SSH connections with Putty.

1. Log into the postgresql server using putty as the root user. Download the postgresql setup script using curl

```
curl -o /root/setuppostgresql.sh https://raw.githubusercontent.com/Azure/linux-on-azure-levelup/main/Module%202%20-%20Modernize/setuppostgresql.sh
```

2. Now run the script to install postgresql server

```
bash setuppostgresql.sh
```

3. Log into the postgresql server

```
sudo -u postgres psql
```

4. Run the SQL script to create the sample database

```
\i /northwind_postgresql.sql;
```

5. Connect to the database

```
\c northwind;
```

Then quit

```
\q
```

7. We need to modify the PostgreSQL configuration to allow remote connections. The two files that we need to modify are:

- /var/lib/pgsql/data/pg_hba.conf
- /var/lib/pgsql/data/postgresql.conf

8. Using vi, add at the bottom of pg_hba.conf file the following two lines.

```
host   all   all  0.0.0.0/0   md5
host   all   all  ::/0   md5
```

NOTE This last action is not a secure or best practice. For the sake of troubleshooting within the lab, this will allow connections from any IP address.

9. Using vi, modify postgresql.conf file so that the server will listen on any ip address. Scroll to the CONNECTIONS AND AUTHENTICATION section of the file. For this lab, add the below information above the current entry and using the '#' to comment out the entry below.

```
listen_addresses = '*'
```

NOTE This last action is not a secure or best practice. For the sake of troubleshooting within the lab, this will allow the IP address to change on reboots for the lab.

10. Last step is to set the password for postgres database user

```
sudo -u postgres psql
```

Then

```
ALTER USER postgres PASSWORD 'yourcomplexpassword';
```

11. You should be able to use a database client such as pgAdmin to connect and view the database.

Setup Apache web server

1. Log into server created in Module 1, LinuxLabVM-CentOS-7-Apache, using putty as root. Type in the command to set the hostname of the server. Then reboot the server.

```
hostnamectl set-hostname apache
```

```
reboot
```

NOTE Changing the host name will make it easier to identify if you open several SSH connections with Putty.

1. Log into the server using putty as the root user. Download the apache setup script using curl

```
curl -o /root/setupapache.sh https://raw.githubusercontent.com/Azure/linux-on-azure-levelup/main/Module%202%20-%20Modernize/setupapache.sh
```

2. Now run the script to install apache web server

```
bash setupapache.sh
```

3. Now download the sample php file into the web directory

```
curl -o /var/www/html/index.php https://raw.githubusercontent.com/Azure/linux-on-azure-levelup/main/Module%202%20-%20Modernize/index.php
```

4. The sample php file has a postgresql database connection string that needs to be modified. Using vi as an editor open the file and change the IP Address to the PostgreSQL Server and Password you set for postgres user.

```
vi /var/www/html/index.php
```

Migration steps

For PostgreSQL Database: https://learn.microsoft.com/en-us/azure/dms/tutorial-postgresql-azure-postgresql-online-portal