**Application Stack Assignment - Cloud Stack LAPP Group 3**

Setting up a LAPP (Linux, Apache, PostgreSQL, PHP) VM instance on Google Cloud Platform.

Step-by-step tutorial to get a LAPP stack up and running:

### **Step 1: Create a Google Cloud Account**

1. Go to the [Google Cloud Platform](https://cloud.google.com/) website.
2. Click on "Get started for free" and follow the prompts to create a new account. You may need to provide billing information, but you won't be charged until your free trial ends or you upgrade to a paid account.
3. Log in to your Google Cloud

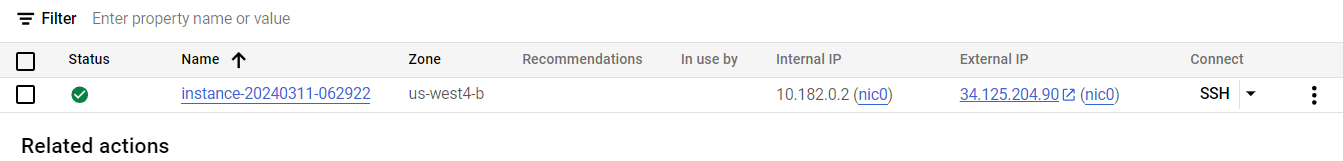
### **Step 2: Create a New Project**

1. Once logged in, go to the Google Cloud Console.
2. Click on the project drop-down menu at the top of the page and select "New Project."
3. Enter a project name, select a billing account, and click "Create."

### **Step 3: Create a Virtual Machine Instance**

1. In the Google Cloud Console, go to the Navigation Menu on the left (☰) > **Compute Engine** > **VM Instances**.
2. Click **Enable**
3. Click "**Create Instance.**"
4. Enter a **name** for your instance. (I used cloudstack3)
5. Choose a region and zone for your instance. (Default or choose close or local region)
6. Change series to N1
7. Click on the arrow to the right of N1
8. Click shared-core
9. Select a machine type (e.g., "e2-medium"). or f1-micro, **g1-small;** etc.
10. In the "Boot disk" section, click "Change," choose "**Ubuntu**" as the OS, and select a version (e.g., **Ubuntu 20.04 LTS**). select size **10 or 20 GB** etc. Click "**Select.**"
11. In the "Firewall" section, check the boxes to allow HTTP and HTTPS traffic.
12. Click "**Create**" to create your VM instance.

it should look like this or similar



### **Step 4: Connect to Your Instance**

1. Once your instance is created, click on the "SSH" button in the VM instances page to open a terminal window in your browser.
2. You are now connected to your virtual machine.

### **Step 5: Install Apache, PostgreSQL, and PHP (in SSH shell)**

1. Update your package list: **sudo apt-get update**
2. Install Apache and PHP: **sudo apt-get install apache2 php libapache2-mod-php**
3. Check PHP Installation: Check if PHP was installed correctly by running:  **php -v**
4. Create a Test PHP Page:
5. Navigate to the web folder**: cd /var/www/html/**
6. Open nano to create a page named info.php**: sudo nano info.php**
7. Add the following code to the file:

**<?php**

**phpinfo();**

* Save and exit nano. CTRL+C, then CTRL+X
* Find the external IP to your VM instance, to the left of the SSH button.
* You can now visit your instance at http://YourExternalIpGoesHere/ to see the PHP configuration.

1. **Restart Apache**: To apply the changes, restart Apache: **sudo service apache2 restart**
2. **Install PostgreSQL:**
   1. Run: **sudo apt install postgresql postgresql-contrib**
   2. During installation, you might be prompted to set a password for the PostgreSQL user.
3. **Create a Database User:**
   1. Switch to the PostgreSQL user: **sudo -i -u postgres**
   2. Access the PostgreSQL prompt:  **psql**
   3. Create a new user with username and password: **CREATE USER stack3 WITH PASSWORD 'YourPassword';**
   4. Exit the PostgreSQL prompt:  **\q**
   5. Return to your regular user: **exit**

**sudo -i -u postgres**

**psql**

**CREATE USER stack3 WITH PASSWORD 'YourPassword';**

**\q**

**exit**

1. **Create a Database:**

Create a database with the name and grant privileges to the new user:

**sudo -i -u postgres**

**psql**

**CREATE DATABASE cloudstack3;**

**GRANT ALL PRIVILEGES ON DATABASE cloudstack3 TO stack3;**

**\q**

**exit**

**ck3;**

1. **Login to the Database:**

**Before we can log onto the database we need to modify the file - "pg\_hba.conf" it is usually located in the PostgreSQL data directory**

* 1. Open the file with a text editor, such as nano:

**sudo nano /etc/postgresql/12/main/pg\_hba.conf**

* 1. Modify the Authentication Method:
  2. Find the lines that looks like this:

**local all all peer**

**local replication all peer**

* 1. Change **peer** to **md5** so that it looks like this: **local all all md5**
  2. Save and close the file. ^c and ^x
  3. Restart PostgreSQL:
  4. Restart the PostgreSQL service to apply the changes:

**sudo service postgresql restart**

**Now that we have modified the config file, we can use the standard database login process that will work going forward.**

1. Exit the PostgreSQL prompt by typing **\q** and then **exit** to return to your regular user.
2. Log in to the database with the new user: **psql -U stack3 -d cloudstack3 -W** (you will be prompted for the password).

Enter the password **'YourPassword'** when prompted.

1. **Create a Table and Insert Data:**

Create a table and insert some data: copied from (Kyrrah's Class)

**CREATE TABLE test\_customers (**

**customer\_id SERIAL PRIMARY KEY,**

**last\_name VARCHAR(50),**

**first\_name VARCHAR(50),**

**email VARCHAR(80)**

**);**

**INSERT INTO test\_customers (last\_name, first\_name, email) VALUES**

**('Smith', 'Bob', 'bob@example.com'),**

**('Jones', 'Bill', 'bill@example.com'),**

**('Doe', 'John', 'john@example.com'),**

**('Rules', 'Ann', 'ann@example.com');**

1. View the Data: To view the data in the table, use the command:

**SELECT \* FROM test\_customers;**

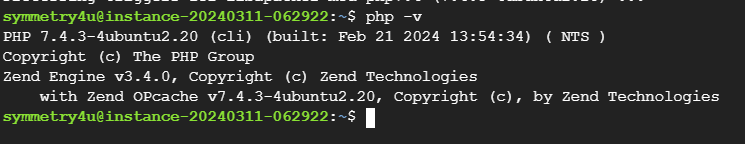
1. Exit PostgreSQL:

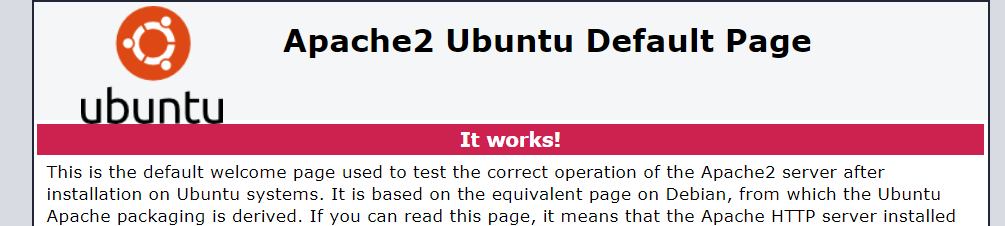
Type:  **\q** to exit the PostgreSQL prompt.

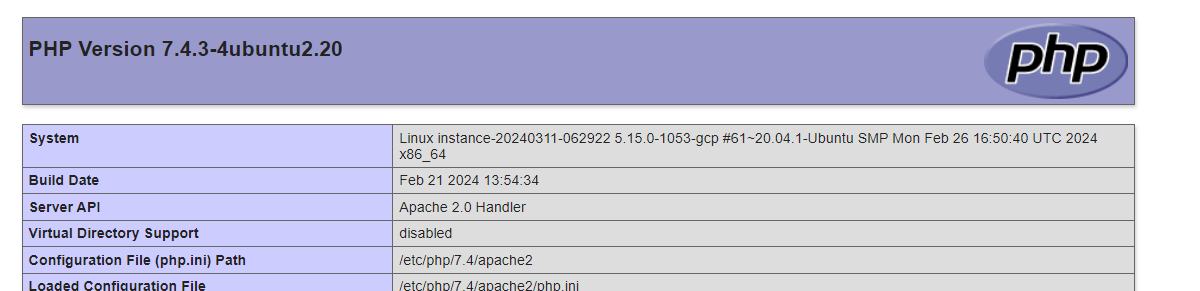
**Install Bill Newman's stuff to test**

1. To navigate to the web folder, we typed the following: **cd /var/www/html/**
2. To view the default web page, we went back to the console and clicked the External IP link, and changed the URL to be insecure (http) **http://34.125.204.90/**
3. Next we installed git: **sudo apt install git**
4. Next we associated our git user.name with this installation: **git config --global user.name "[Name]"**
5. Next we associated our git user.**email** with this installation: **git config --global user.email "Your email"**
6. As a sample, we used git clone to copy/paste a repo in our web space: **sudo git clone https://github.com/newmanix/js-cards.git**
7. We were able to view the web page here: **http://34.125.204.90**[**/js-cards/**](http://34.82.73.108/js-cards/)

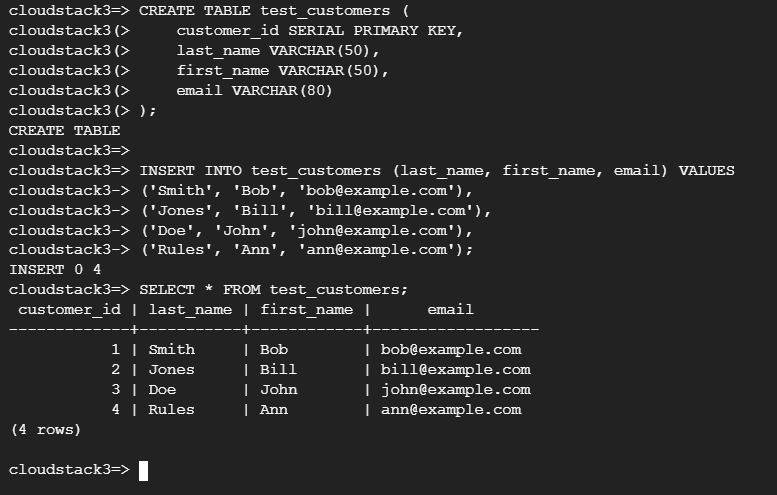
**Screenshots of the Successful LAPP server instance install**







**Functioning DataBase Example**



**Website pages served on LAPP server instance**

