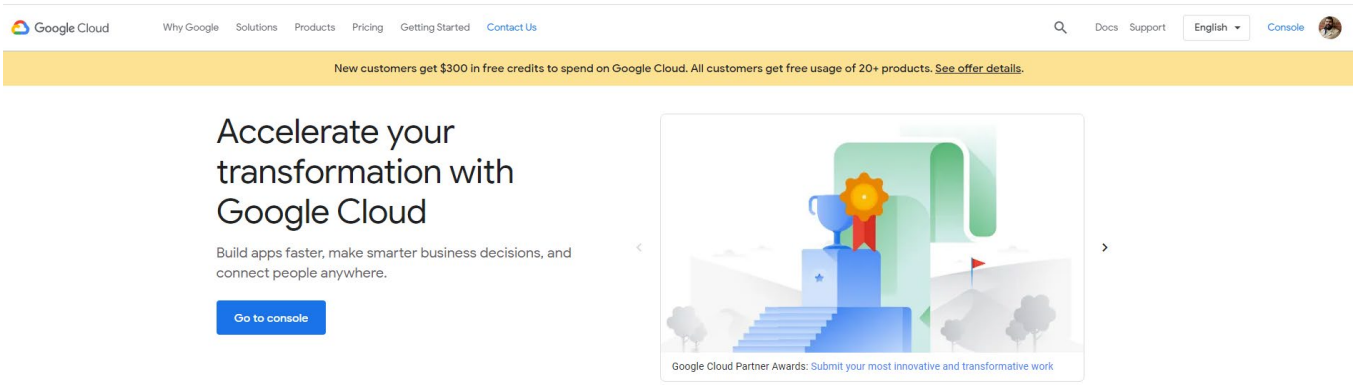
Going live with Google Cloud!

Note: This is a modified version of the community tutorial created in:
https://cloud.google.com/community/tutorials/setting-up-lamp

Step 1: Creating your Google Cloud Account

Sign up for an account at https://cloud.google.com

At the top right, click on the word: **Console**

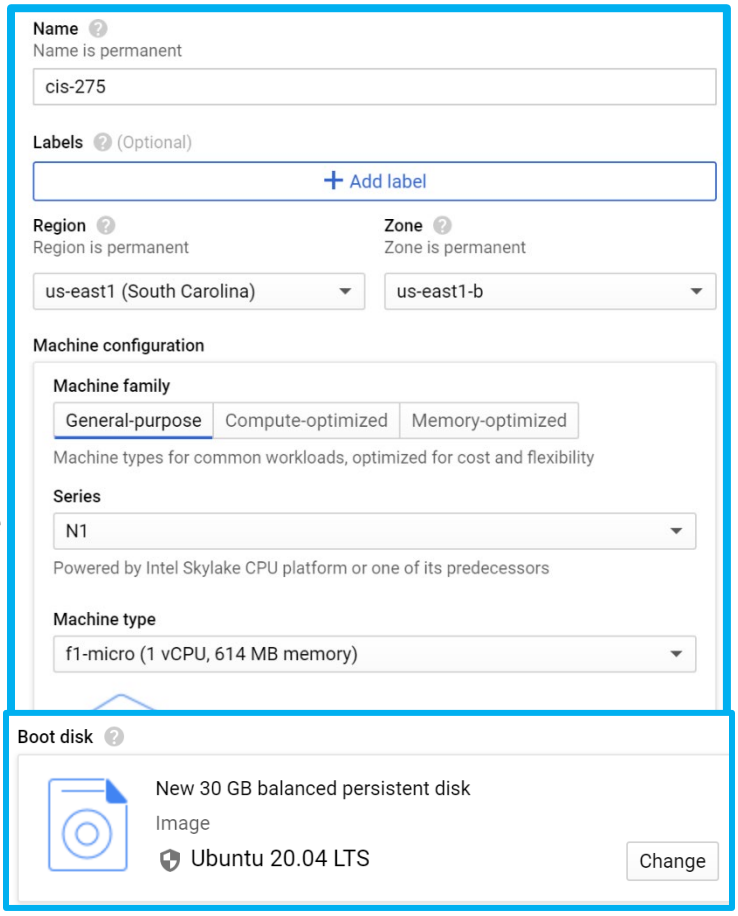


Step 2: Creating a virtual machine instance

You can use these steps to deploy the LAMP stack using the Google Cloud Platform Console:

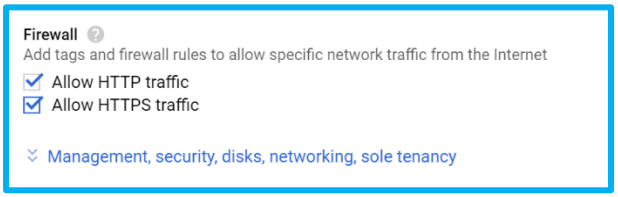
1. In the Cloud Platform, go to the Computer Engine > VM instances
2. Click on Enable API if you don't have access to it yet.
3. Wait for the server to set up,
   then click the Create instance button.
4. Set **Name** to **cis-275**.
5. Set **Region** to us-east1 and **Zone** to us-east1-b
6. Set **Machine Series to N1**
7. **Set Machine Type** to **f1-micro**.

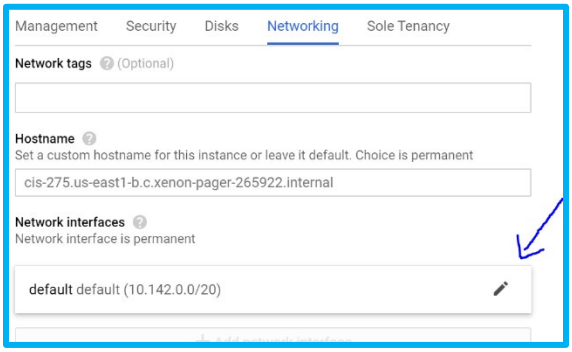
   In the future, once you run out of credits you can use these same settings, if you are interested, machines with these settings are FREE (1 instance) for the month. Additional hours from a similar machine will cost more.

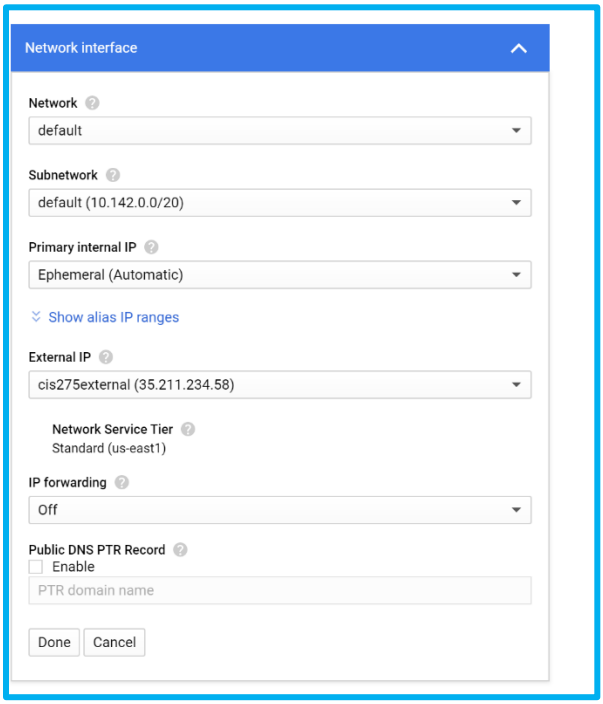


8. In the **Boot disk** section,
   click **Change** to begin configuring your boot disk.
9. In the **Public Images** tab, choose, **Ubuntu** as the OS,
   and **Ubuntu 20.04LTS** version.
10. Choose **Balanced Persistent Disk**
    and set the size(GB) to **30**
11. Click **Select**.
12. In the **Firewall** section,
    select **Allow HTTP traffic** and **Allow HTTPS traffic**.
13. Click on the “Management, security, disks, networking, sole tenancy” link to see more options
14. Click on the **Networking** tab
15. Click on the edit button, under network interfaces.
16. Within the **External IP** options
    Click on **Create IP Address**.
17. Choose a Name (could be anything, for example cis275ip)
18. For the Network Service Tier option, choose **Standard**,
    then click on **Reserve**
19. Click on **Done**
20. Click the **Create** button to create the instance.

    Give the instance a few seconds to start up.







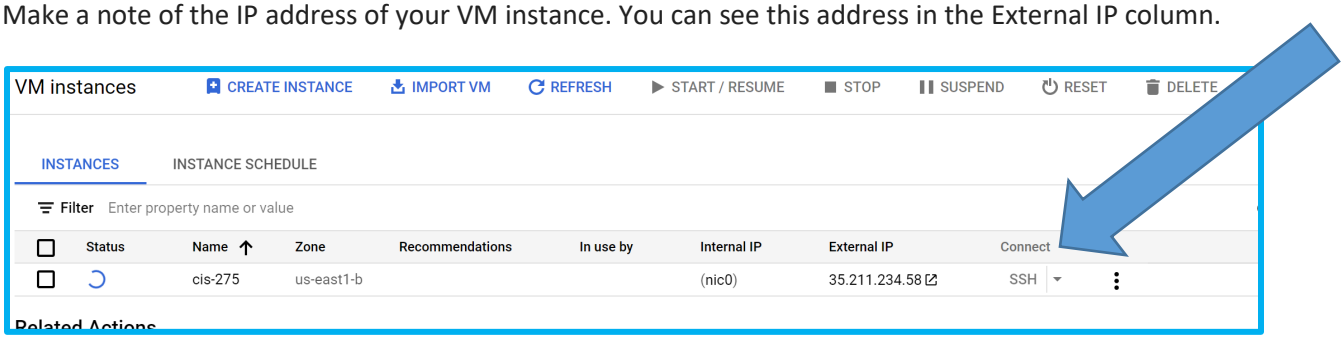
Step 3: Deploying the LAMP stack on Compute Engine

Now that your virtual machine instance is running, configure the LAMP stack.

Connect to your instance

1. In the Cloud Platform Console, go to the **VM Instances** page (You should already be at this page from the previous part).
2. In the list of virtual machine instances, click the *SSH* button in the row of the instance to which you want to connect.

   Make a note of the IP address of your VM instance. You can see this address in the External IP column.



Preparing your instance:

The first step, is to make sure your Linux server has an up to date installer (apt-get). We also will need to make sure we have set up something called a swap file, which allows instances with little memory to better organize the programs that are running in memory.

For each command, you can copy and paste and the instructions below!

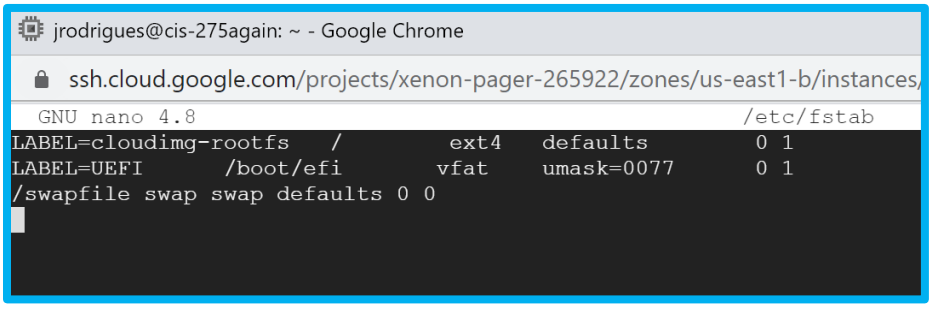
```
sudo apt-get update
```

For the swap file, you need to do a series of commands:

```
sudo fallocate -l 1G /swapfile
sudo dd if=/dev/zero of=/swapfile bs=1024 count=1048576
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile
sudo nano /etc/fstab
```

Upon typing the last command, you will view a window with two lines. Use the arrow key to go to the third line and type (or copy and paste):

```
/swapfile swap swap defaults 0 0
```



Now, **press ctrl+o** (or cmd+o) then press enter (this saves the file).

Then **press ctrl+x** (or cmd+x), this exits the file

Install Apache and PHP on your instance

By creating an instance, you already have the "Linux" part of LAMP. Next, install Apache and PHP by typing the following commands into the shell. During installations you may need to type Y to continue and confirm prompts.

```
sudo apt-get install apache2 php libapache2-mod-php
```

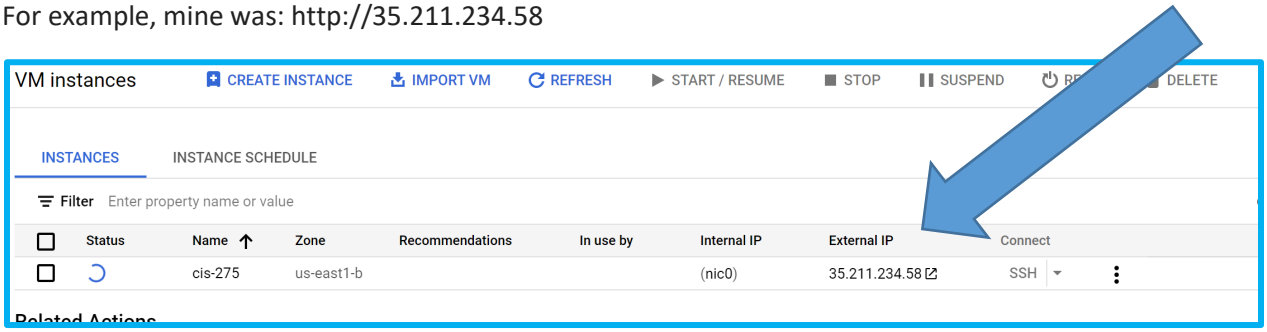
Test Apache and PHP

1. For this step, you need the external IP address of your instance. You can look up the address in the **VM Instances** page in the Cloud Platform Console.

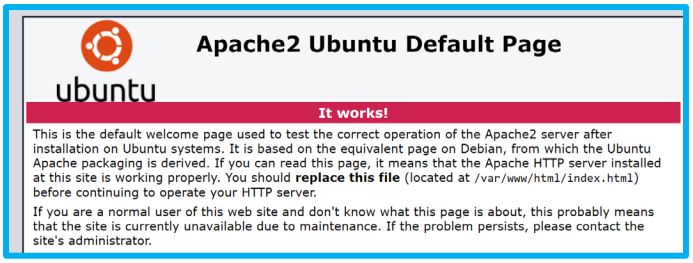
   In a browser, enter your external IP address to verify that Apache is running:

   ```
   http://[YOUR_EXTERNAL_IP_ADDRESS]
   ```

   For example, mine was: http://35.211.234.58



   You should see the Apache test page.



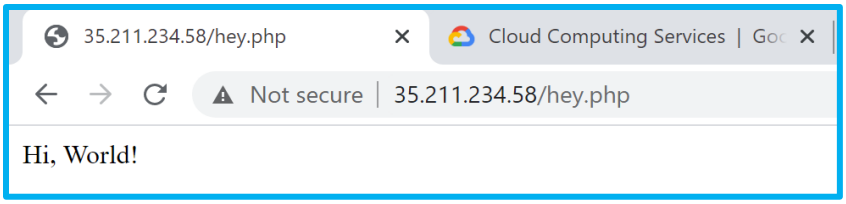
2. You can write the code to the file from the command line by using a statement like the following.

   ```
   sudo sh -c 'echo "<?php echo \"Hi, World!\";  ?>" > /var/www/html/hey.php'
   ```

3. Browse to the test file to verify that Apache and PHP are working together, by typing into your browser:

   ```
   http://[YOUR_EXTERNAL_IP_ADDRESS]/hey.php
   ```

   You should see the message “Hi, World!” on your browser



If the page failed to load (HTTP 404), verify:

- In the Cloud Platform Console, HTTP traffic is allowed for your instance.
- The URL uses the correct IP address and file name.

Install MySQL on your instance

Install MySQL and related PHP components, don't forget to type Y, when prompted.

```
sudo apt-get install mysql-server php7.4-mysql php-pear
```

Configure MySQL

Now that you have MySQL installed, we want to improve the security of your installation. This performs steps such as setting the root user password if it is not yet set, removing the anonymous user, restricting root user access to the local machine, and removing the test database.

When picking the password, note that the cursor will remain invisible (and will not move). I am suggesting using cis275 as the password so that you do not forget it, but this could be something else.

```
sudo mysql_secure_installation
```

Answer as follow:

Would you like to setup VALIDATE PASSWORD component? Press y|Y for Yes, any other key for No: N
Enter password for user root: cis275
Re-enter new password: cis275
Remove anonymous users? (Press y|Y for Yes, any other key for No): Y
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : N
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : Y
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Y

Use phpMyAdmin for database administration

You can use phpMyAdmin to administer your database through a user interface

Install phpMyAdmin

```
sudo apt-get install phpmyadmin
```

- Select **apache2**.
- Select **yes** to use `dbconfig-common` for database setup.
- Enter the database administrator's password that you chose during MySQL configuration. (for example **cis275**)

Testing phpMyAdmin

Now that it is installed, we need to try and make sure that phpMyAdmin is accessible. Type the following command to copy your phpMyAdmin into your website:

```
sudo cp -r /usr/share/phpmyadmin /var/www/html
```

apache must be restarted:

```
sudo service apache2 restart
```

Creating a new Database User and Allowing it to enter information

```
sudo mysql -uroot -p
```

Enter the password (should be cis275)

Create a new user by using the command, replace YOUR_USERNAME and YOUR_PASSWORD with the credentials you provided in your local XAMPP database, the command below is ONE command, type it in the same line!

```
CREATE USER 'YOUR_USERNAME'@'localhost' IDENTIFIED WITH mysql_native_password BY 'YOUR_PASSWORD';
```

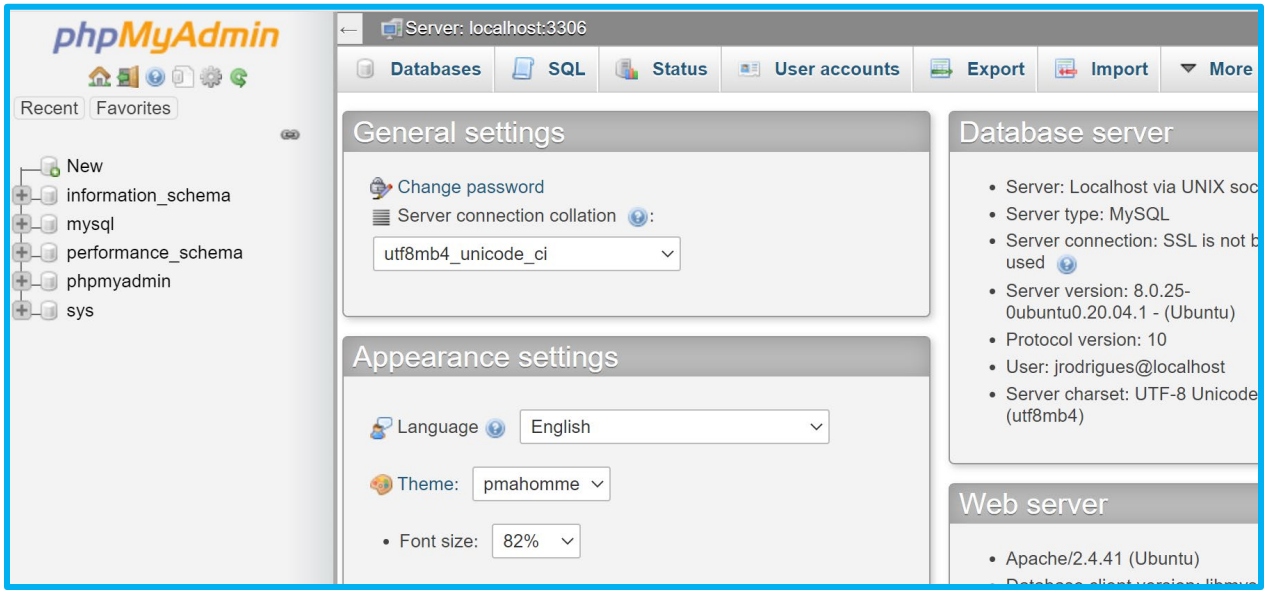
Then grant administrator permission to this use: by typing:

```
GRANT ALL PRIVILEGES ON * . * TO 'YOUR_USERNAME'@'localhost';
```

Type: `exit;`
Go to a new browser window and go to the address:

http://[External IP]/phpmyadmin

Log in with username and password that you just set up. You should our usual phpMyAdmin page:



Step 4: Set Permissions for external use

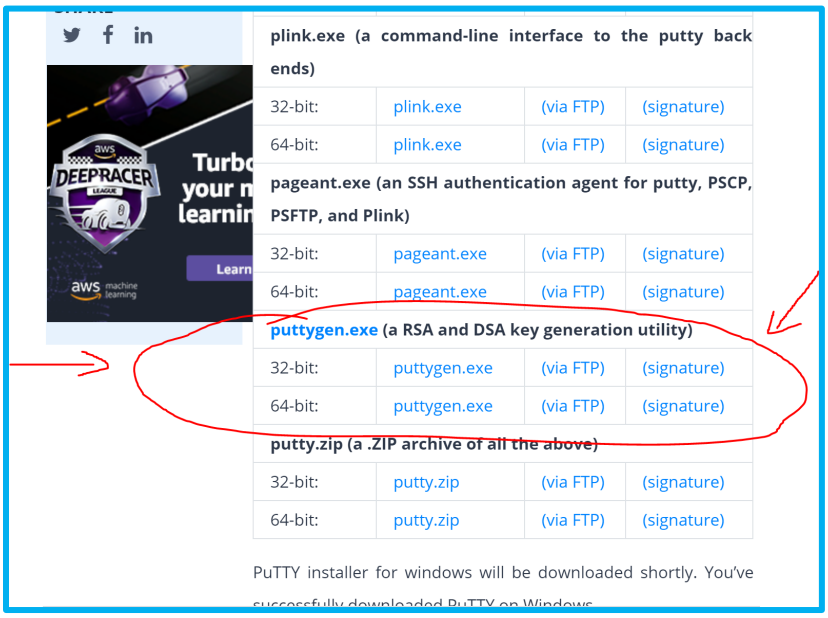
1) Type the following commands in the VM instance (these will give you permissions to edit these files from your own computer, normally this is a very unsafe way of doing this, however, it simplifies the process at this moment… never do this on a real server!)

```
sudo    chmod    777    /var/www/html
```

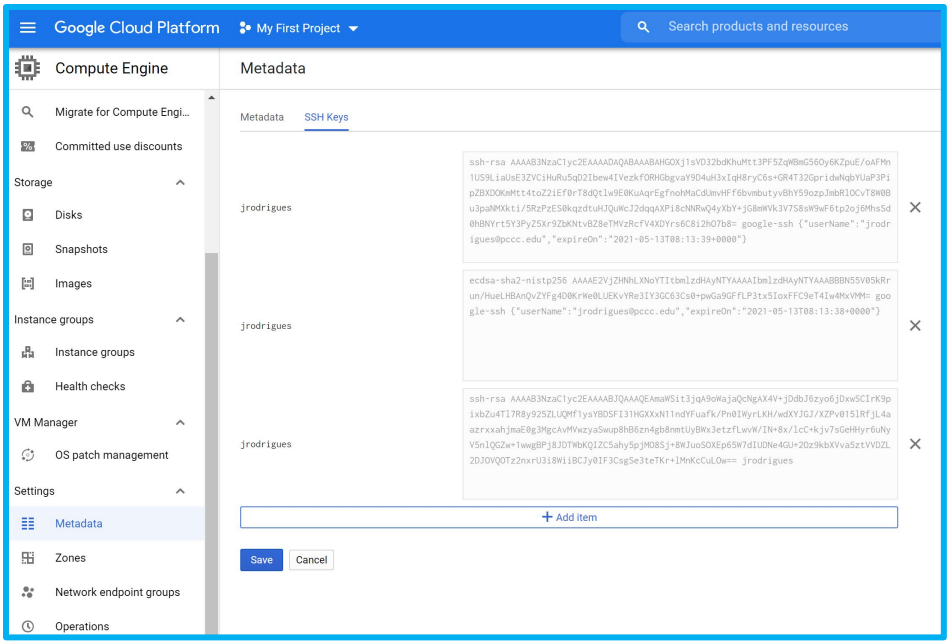
2) **Do not close your VM Instance.**

Step 5: Setting up Filezilla (for file transfer)

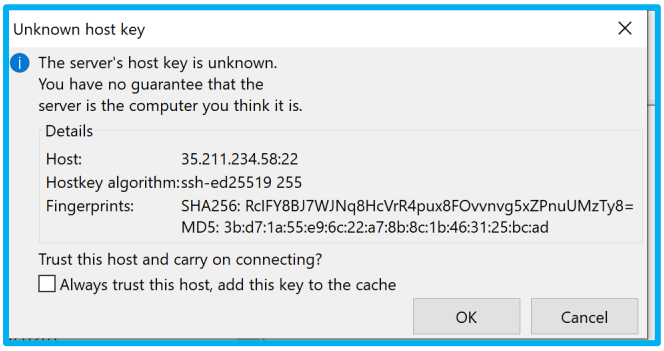
1) Go to the following website, download and install the FileZilla client: https://filezilla-project.org/
2) Go the following website
    a. **Windows:** download and install the **puttygen(**warning, this website is AD-heavy), make sure to scroll down ¼ of the page, and select **puttygen.exe:** **https://www.puttygen.com/download-putty**



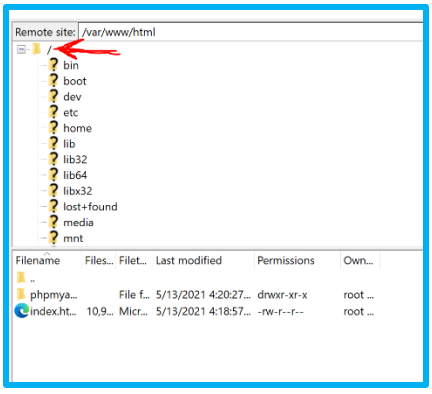
    b. **Mac,** follow the instructions here to create a public and private keys you can then skip to step 4**:** https://docs.digitalocean.com/products/droplets/how-to/add-ssh-keys/create-with-openssh/
3) Open Puttygen and click on **Generate**. You may choose a new "Key Comment"- this will be username, and passphrase if you would like. Once you type them, you do NOT need to generate again. Save your public key, and you private key, somewhere you can find late. Copy the text of the public key that was generated for you.
4) Copy the text for your Public Key, and go to your Google Cloud Console. On the Computer Engine page, scroll down until you see "Metadata"
5) Click on SSH Keys, then choose EDIT, click on "Add Item", and paste



6) Open your FileZilla.
   a. Click on “Edit”> Settings > Click on SFTP
   b. Click on “Add Key File” and browser for your private key, this will have a ppk extension.
   c. Confirm.
   d. Type in your [EXTERNAL IP] for the HOST, your username, and password chosen when the key was generated. For **Port** type 22.
   e. Click on Quickconnect. A prompt should appear. Click ok and re-enter your password if prompted.



   f. On the left side, you will see the folders on your own computer, on the right side, you will see the files on the server.
   g. To find your website files, navigate to /var/www/html by clicking on the / folder at the top.
   h. You can transfer flies by dragging and dropping them into your /var/www/html folder.
   When you access http://[EXTERNAL IP] you are sent directly to index.php (or index.html) any other folders means you have to add that to your website path.



For example, if you create a new folder inside of the /var/www/html/ such as cis275, you would access it by typing: http://[EXTERNAL IP]/cis275/file.html on the browser.

Step 7: Releasing your website.

To make your website work from localhost to the actual server:

1) Use FIlezilla (or WinSCP) to transfer your website files from your computer into the /var/www/html/ folder on the Instance.
2) To transfer your database:
    a. Using localhost, go to your phpMyAdmin page: localhost/phpMyAdmin
    b. Once in phpMyAdmin, click on the database which you would like to recreate.
    c. On the top navigational bar, click on export
    d. Under Export Method, make sure that Quick is selected. Then press Go at the bottom of the page.
    e. A .sql file should have been downloaded into your computer. Open this file in a text editor.
    f. Copy the entire code in this file.
    g. Log in to your instance's phpMyAdmin page, typing on your browser http://[External IP]/phpmyadmin
    h. Log in with your credentials (should be root, and cis275)
    i. On the left menu, click on new, and make sure to create a database with the same name as your localhost database.
    j. On the top menu, click on SQL, paste the code that you copied before, and click on Go.
    k. A series of green symbols should appear on your screen.
    l. Your website should work now and anyone in the world can access it.