Tutorial/Laboratory 08

LAMP Stack Setup and Website Deployment

1. LEARNING OUTCOMES

Upon completion of these laboratory exercises, you should be able to:

- Create an Ubuntu virtual machine hosted in Google Cloud.
- Install and configure the standard LAMP (Linux, Apache, MySQL, PHP) stack.
- Deploy a PHP application to the server.

2. SOFTWARE

2.1 Required:

- Visual Studio Code (https://code.visualstudio.com/)
- Activated Google Cloud account (https://console.cloud.google.com/)
- SFTP extension for Visual Studio Code (https://marketplace.visualstudio.com/items?itemName=Natizyskunk.sftp)

2.2 Optional:

- Google Cloud Code extension for VS Code (https://marketplace.visualstudio.com/items?itemName=GoogleCloudTools.cloudcode)

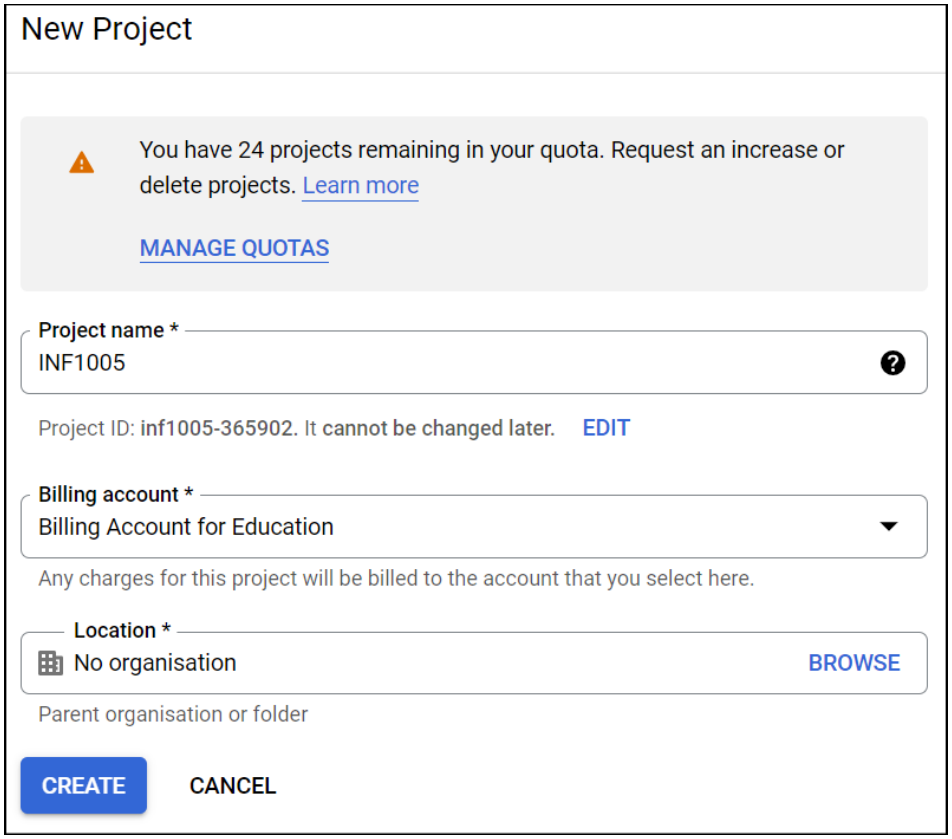
NOTE: we will be using SSH (Secure Shell Protocol) in these exercises, but for security reasons, the SSH port (22) is sometimes blocked by network administrators. If this is the case, you can try alternate networks (e.g. ICT-WIFI) or connect using your mobile phone hotspot.

3. CREATING A VIRTUAL MACHINE INSTANCE IN GOOGLE CLOUD

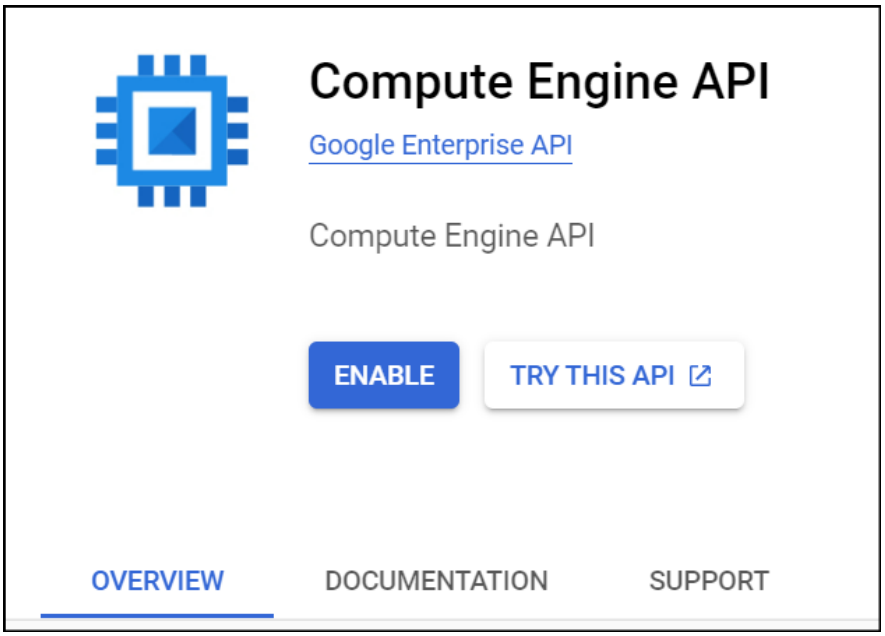
In this exercise, you will create and launch a virtual machine (VM) instance of Ubuntu using Google Cloud. **You will need a Google Cloud account in order to proceed with this exercise.**

You should have received an email with instructions for claiming your $50 in Google Cloud credits (check your junk/spam folder if you did not receive).

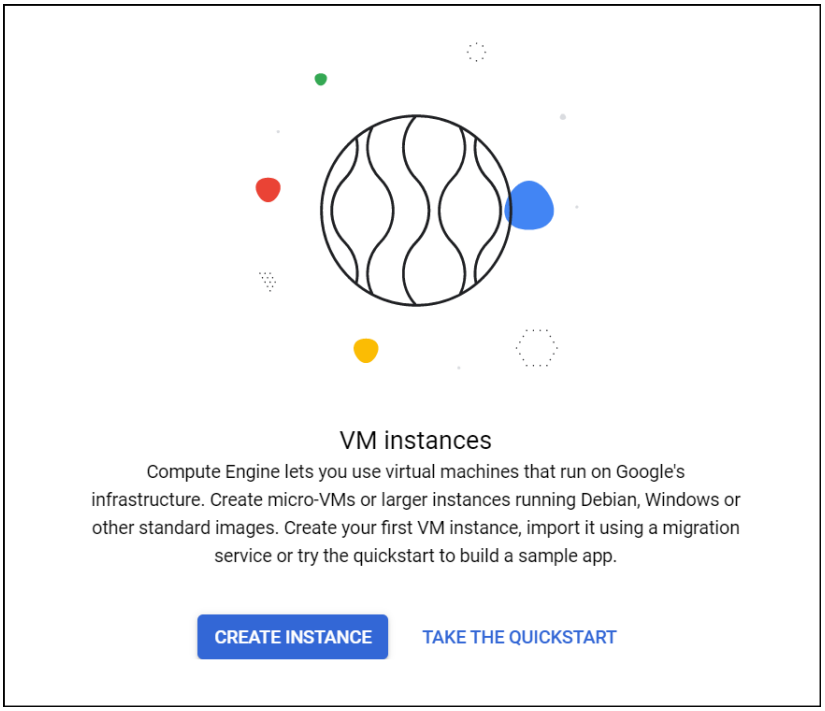
3.1 Go to your Dashboard in Google Cloud and either select an existing Google Cloud Project or create a new one:



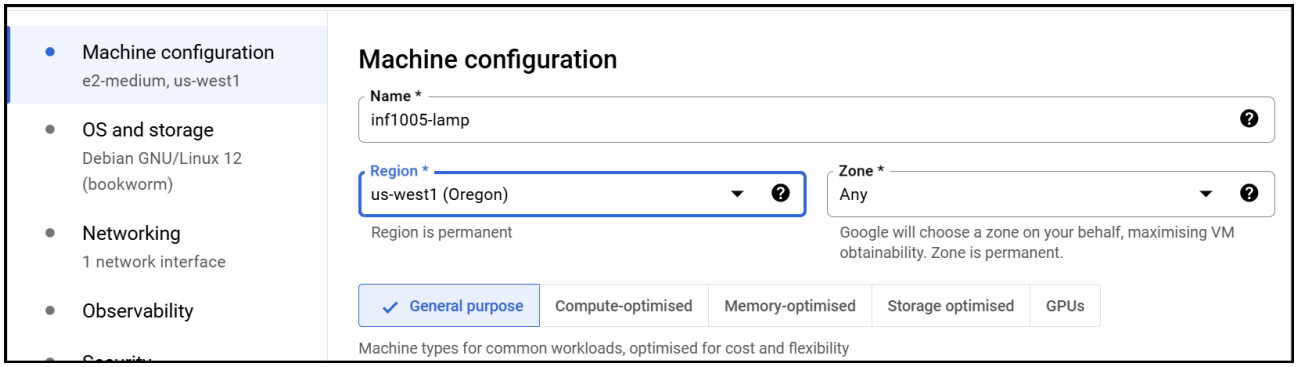
3.2 Go to the VM Instances page - you may be prompted to enable the Compute Engine API if you haven't previously:



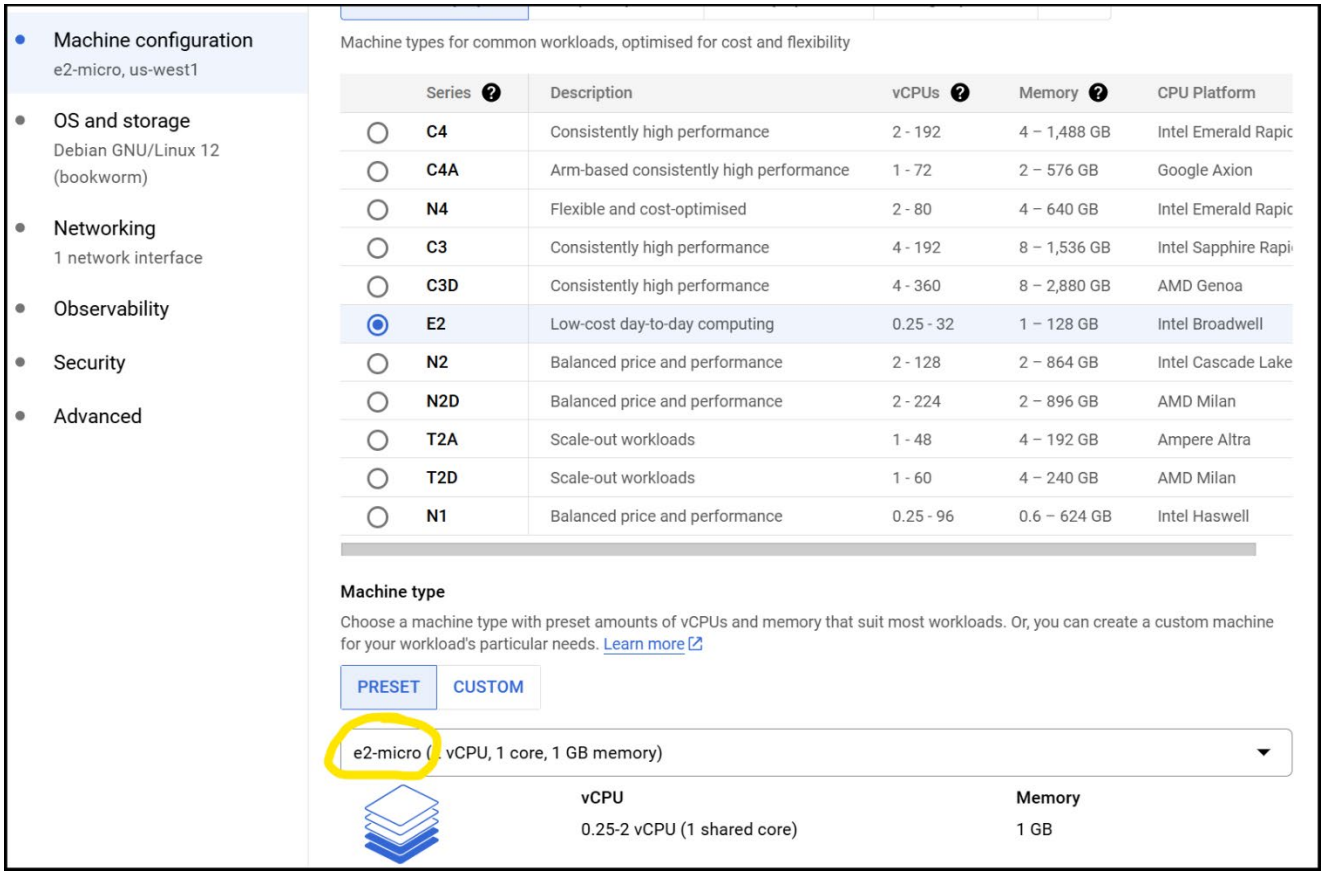
3.3 Click **Create Instance** to initiate the virtual machine configuration:



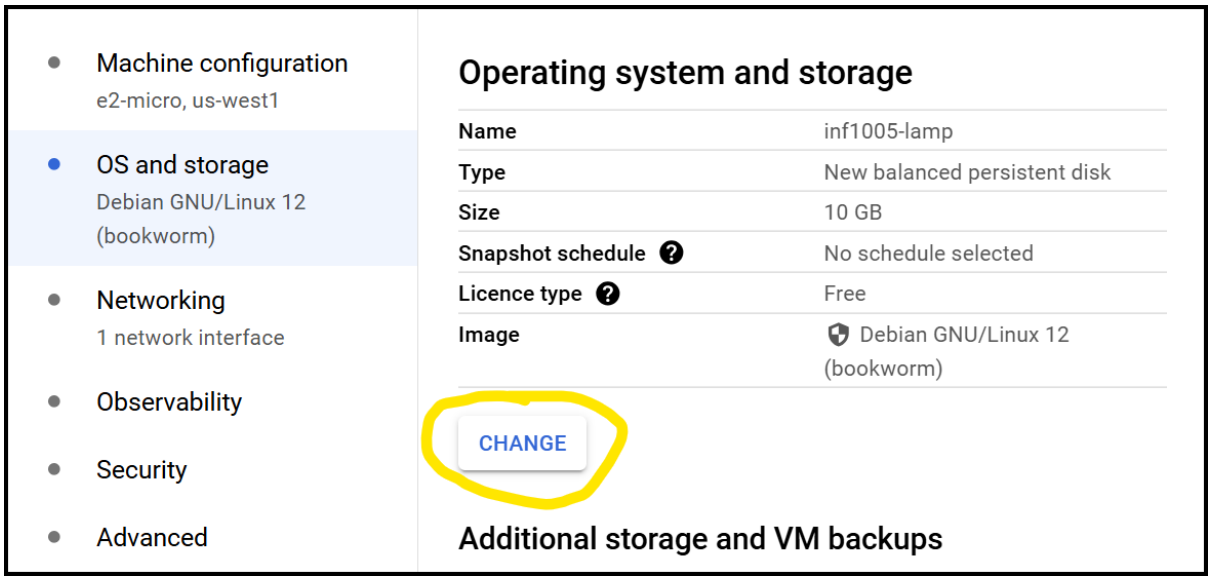
3.4 With the **Machine configuration** section selected, enter an appropriate name such as 'inf1005-lamp' in the **Name** field. Select 'us-west1 (Oregon)' for the **Region** (other regions will work, but us-west1 allows you to run the free-tier services).



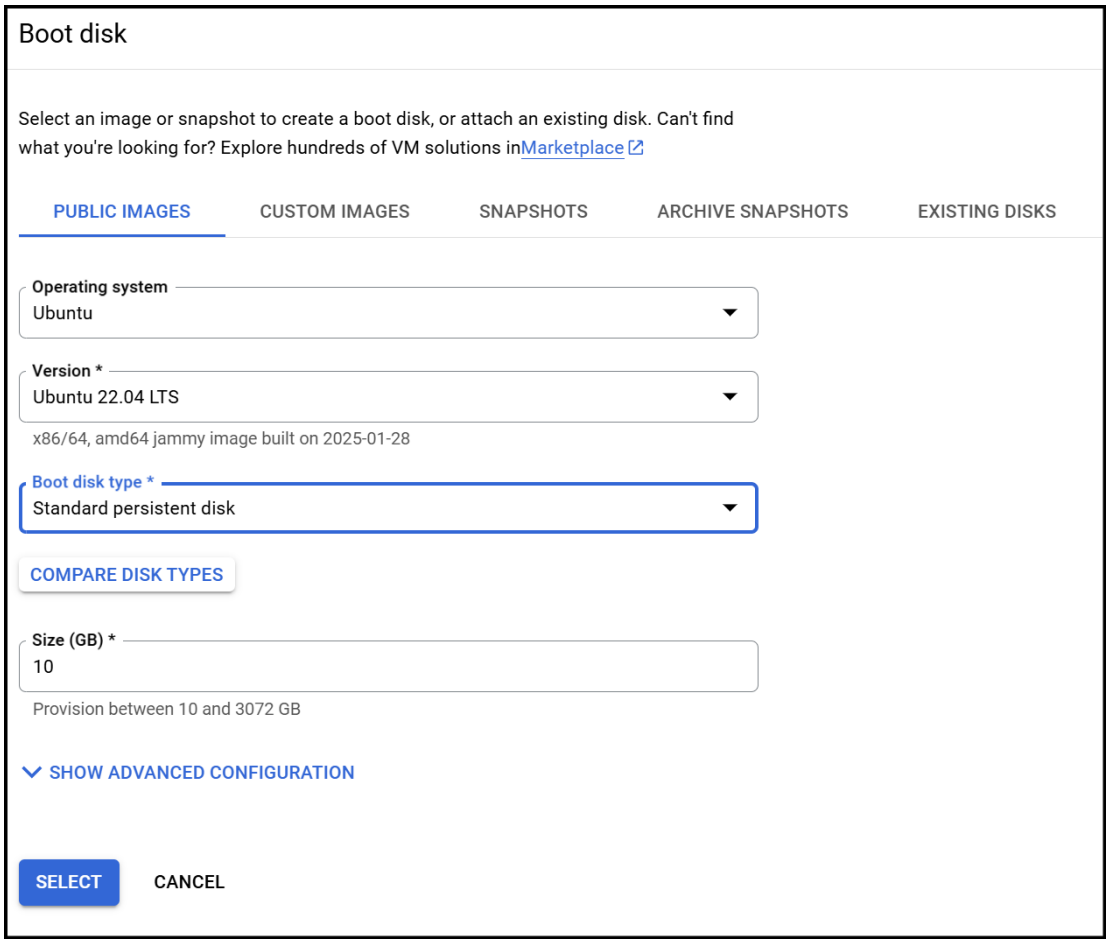
3.5 Select 'E2' for the **Series** and 'e2-micro' for **Machine type**:



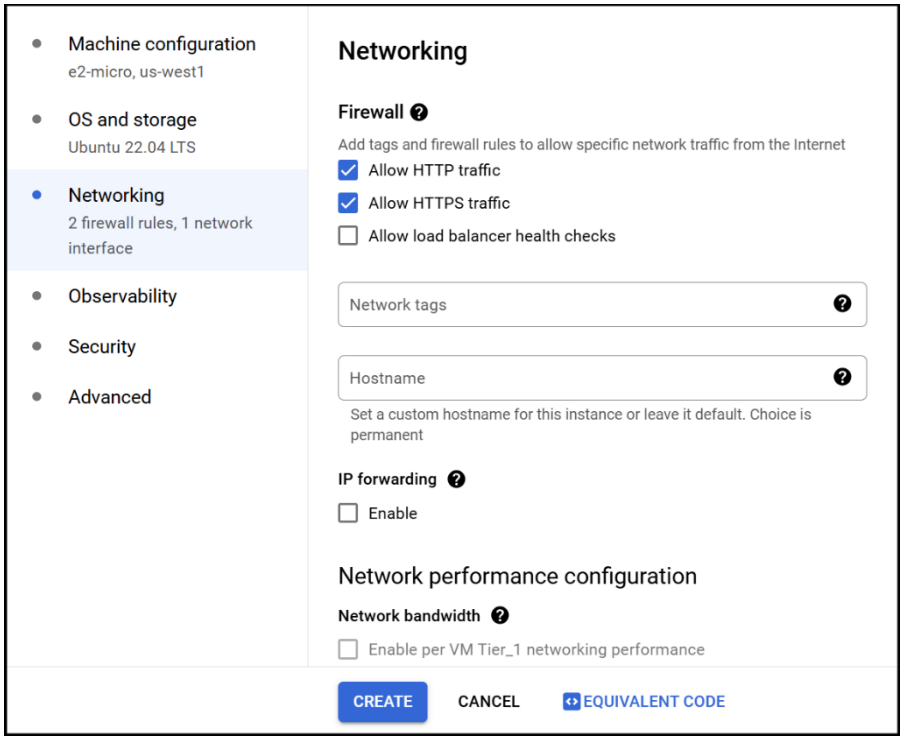
3.6 With the **OS and storage** section selected, click the **CHANGE** button:



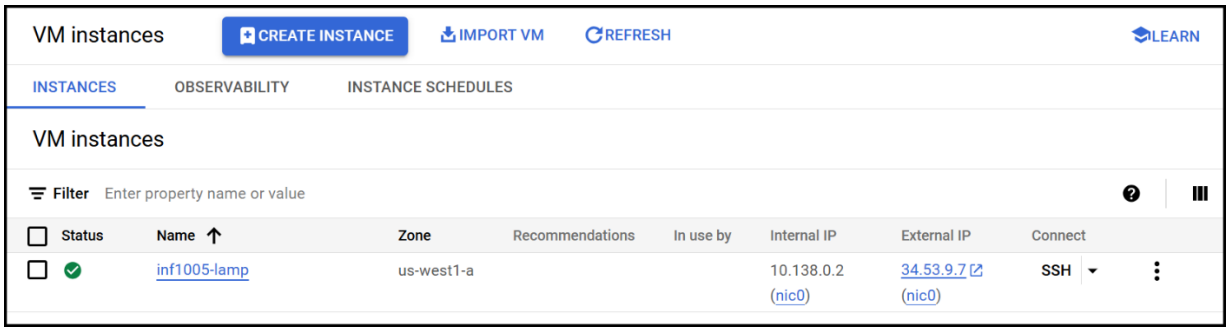
3.7 Select ‘Ubuntu’ for the **Operating system**, then ‘Ubuntu 22.04 LTS x86/64’ for the **Version**. Change the **Boot disk type** to ‘Standard persistent disk’ and set the **Size** to 10 GB.



3.8 Click **SELECT** to save the changes and return to the instance creation page. Then, with the **Networking** section selected, check the boxes under **Firewall** to allow HTTP and HTTPS traffic:

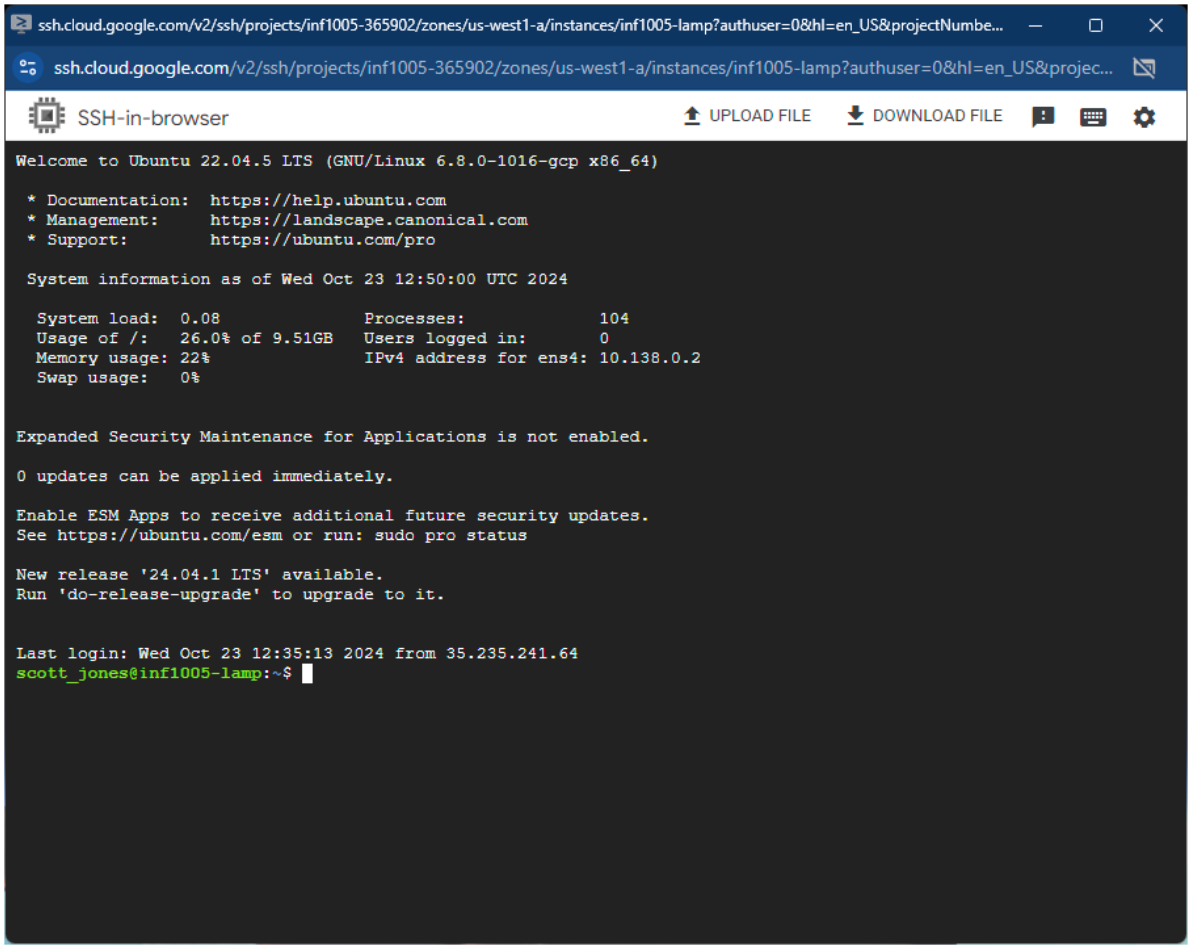


3.9 Finally, click the **CREATE** button at the bottom to create the new instance. You will be returned to the VM Instances screen while the instance is being created. After a minute or so, you should see a green checkmark under the **Status** column, indicating that the instance was successfully created and is running:



4. CONFIGURING THE UBUNTU INSTANCE

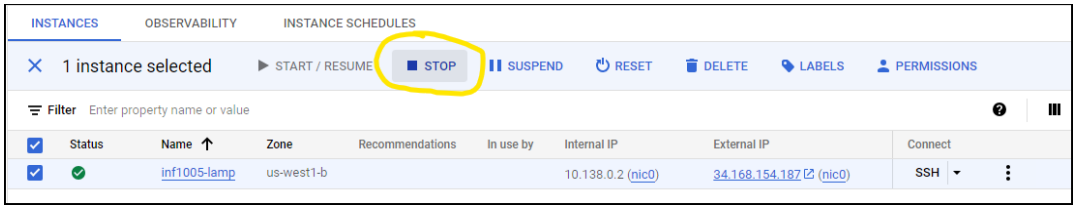
4.1 With our server instance up and running, we can use Google Cloud's **SSH-in-browser** feature to connect to the virtual machine remotely. Click on **SSH** under the **Connect** column to establish the connection (you may see a pop-up window prompting you for authorization the first time you connect). This will open a secure connection using a browser window:



4.2 Now we will perform some standard updates to the base software to be sure we have the latest security patches. Enter the following commands in the terminal window, one at a time (omit the '$' character):

```
$ sudo apt update
$ sudo apt upgrade -y
```

This may take several minutes to complete. When the updates are finished, it's a good idea to restart your instance. To do this, exit the **SSH-in-browser** window (type `exit` at the prompt or close the window), then select the instance in the VM Instances console and click on the **STOP** button at the top:



Wait for the instance to finish shutting down, then click **START** to restart it. Reopen the **SSH-in-browser** connection to complete the remaining exercises.

4.3 Next, we will add a user account for remote login. This will also allow us to connect from any SSH client without having to go through the Google Cloud Console, as well as connect with other tools such as the SFTP extension and MySQL Workbench.

a. Enter the following command to add a new user called `inf1005-dev`:

```
$ sudo adduser inf1005-dev
```

You'll be prompted for a password. Be sure to remember this password – if you should forget it, you'll have to log in via Google Cloud Console and repeat the above steps. Also, **make sure you use a strong password** – we've had incidents in the past where hackers gained access to students' VMs using simple dictionary attacks!

In addition to the password, you'll be prompted to enter some optional information about the user – you can just hit ENTER to accept the defaults:

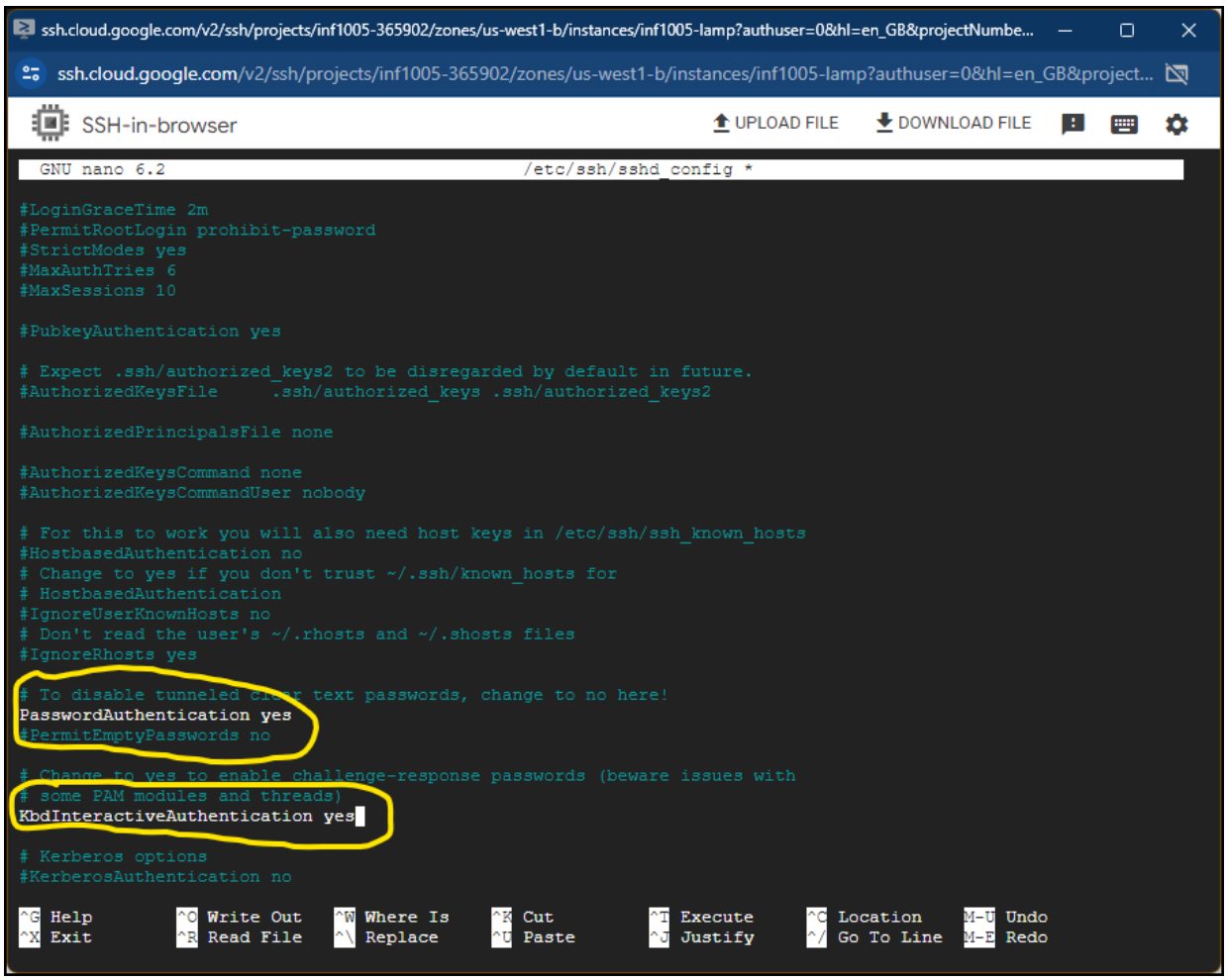
```
scott_jones@inf1005-lamp:~$ sudo adduser inf1005-dev
Adding user `inf1005-dev' ...
Adding new group `inf1005-dev' (1003) ...
Adding new user `inf1005-dev' (1002) with group `inf1005-dev' ...
Creating home directory `/home/inf1005-dev' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for inf1005-dev
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
scott_jones@inf1005-lamp:~$
```

b. Next, we will grant administrative privileges to the `inf1005-dev` user. Enter the following command to add the `inf1005-dev` user to the `sudo` group:

```
$ sudo usermod -aG sudo inf1005-dev
```

c. We also need to enable password authentication for SSH so that we can log in with the `inf1005-dev` user externally. To do this, edit the `sshd_config` file and change both **PasswordAuthentication** and **KbdInteractiveAuthentication** to '**yes**' (after removing the '#' comment symbol at the start). In this example, we'll use the nano editor, but you can use vi or any editor of your choice:

```
$ sudo nano /etc/ssh/sshd_config
```



Hit Ctrl-O and ENTER to save the file, then Ctrl-X to exit.

For the above changes to take effect, **you must enter the following command to restart the SSH service**:

```
$ sudo service ssh restart
```

You may now exit the SSH-in-browser connection.

4.4 To verify that the above steps were completed successfully, let's try connecting using our own SSH client. You may use any SSH client you wish, including:

- Terminal window provided by Visual Studio Code
- Windows PowerShell (Windows users)
- Putty (Windows & Linux)
- Built-in terminal program (Mac and Linux)

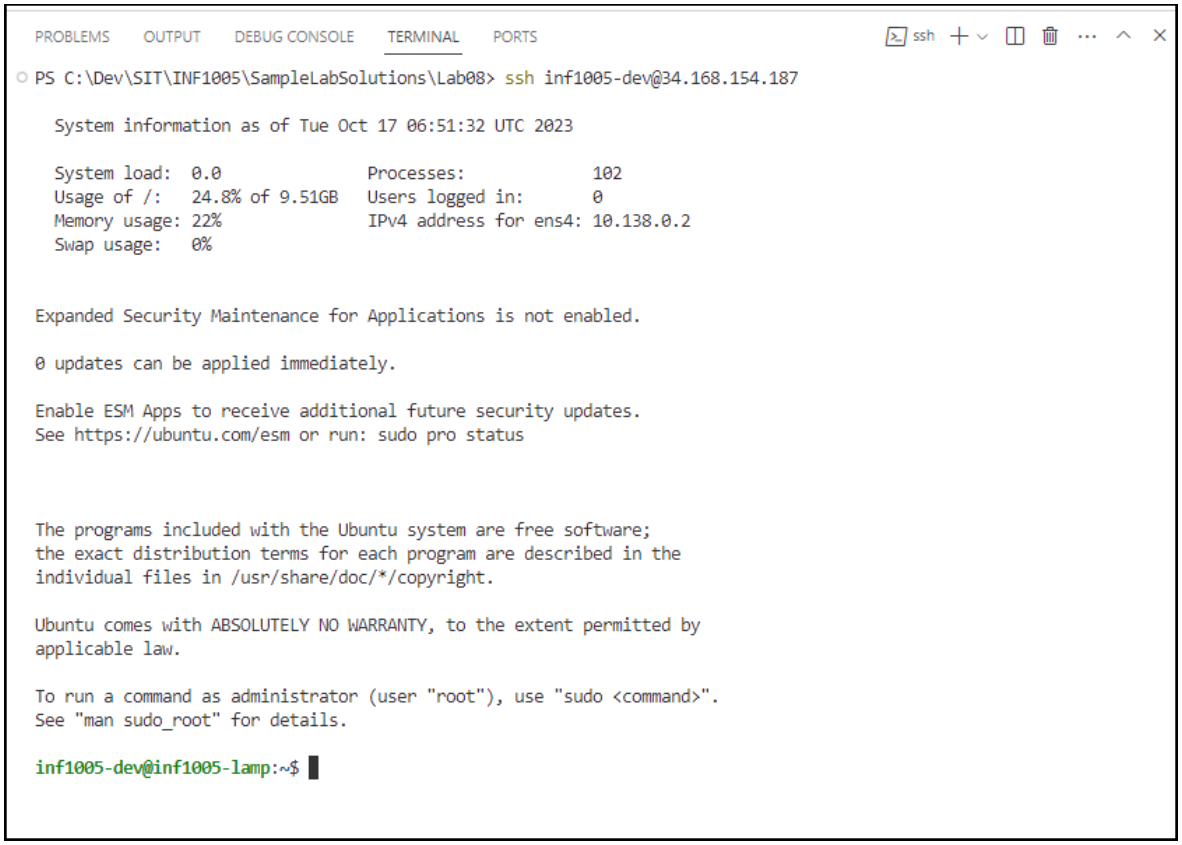
a. First, determine the public IP address of your instance. This is shown next to your instance under the **External IP** column. You may click the icon there to copy the IP address to the clipboard:

b. With your SSH client open, enter the following command, replacing <IP-ADDRESS> address with the address of your instance:

```
ssh inf1005-dev@<IP-ADDRESS>
```

Answer 'yes' to any security prompts, then enter the password you set for the `inf1005-dev` user earlier.

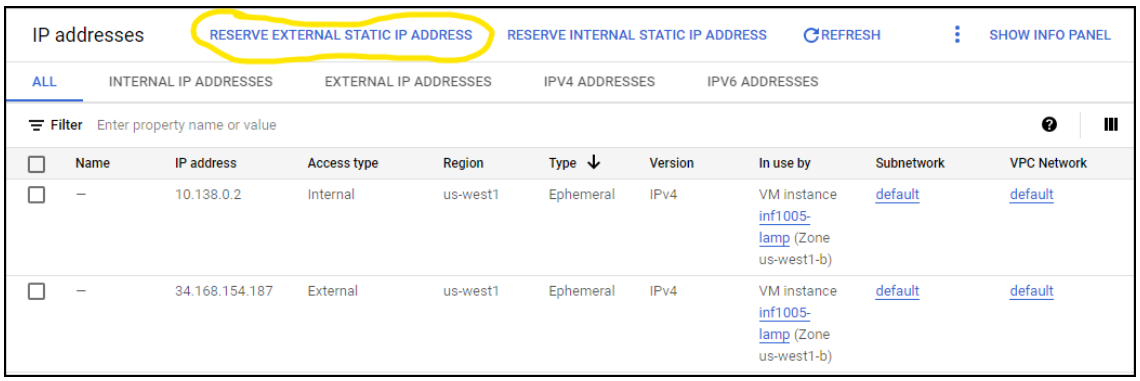
c. If your login was successful, you should now be at the Ubuntu terminal prompt. This example uses the Terminal from VS Code:



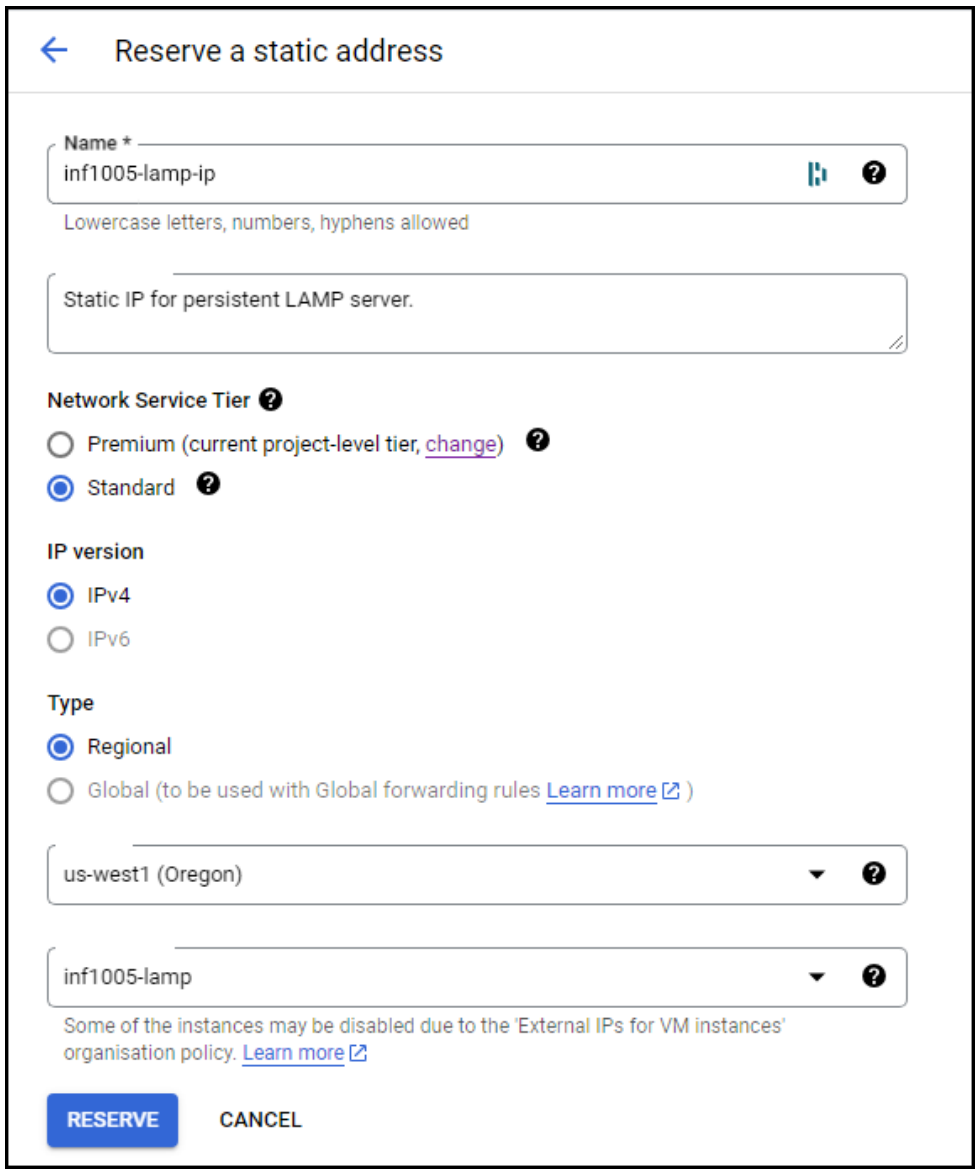
Type `exit` at the prompt to close the connection.

4.5 The final step in setting up our Ubuntu virtual machine is to reserve a static IP address so that we don't have to keep typing a different IP address every time the VM is restarted. To do this:

a. Go to the External IP addresses page and click **RESERVE EXTERNAL STATIC ADDRESS** at the top:



b. On the next screen, enter a name to help remember what this IP address is used for and an optional description. **Network Service Tier** should be 'Standard' and **Region** should be the same as your VM (e.g., `us-west1`). **Attached to** should be set to the instance you created earlier:



Click **Reserve** to create the static IP address. Be sure to write down this address as you will be referring to it often.

5. INSTALLING APACHE WEB SERVER

In this exercise, we will install Apache, the world's most popular open source web server.

5.1 Using the static IP address you created earlier, open a SSH connection to your VM instance and enter the following command in the terminal window:

```
$ sudo apt install apache2
```

5.2 Check that the Apache installation was successful:

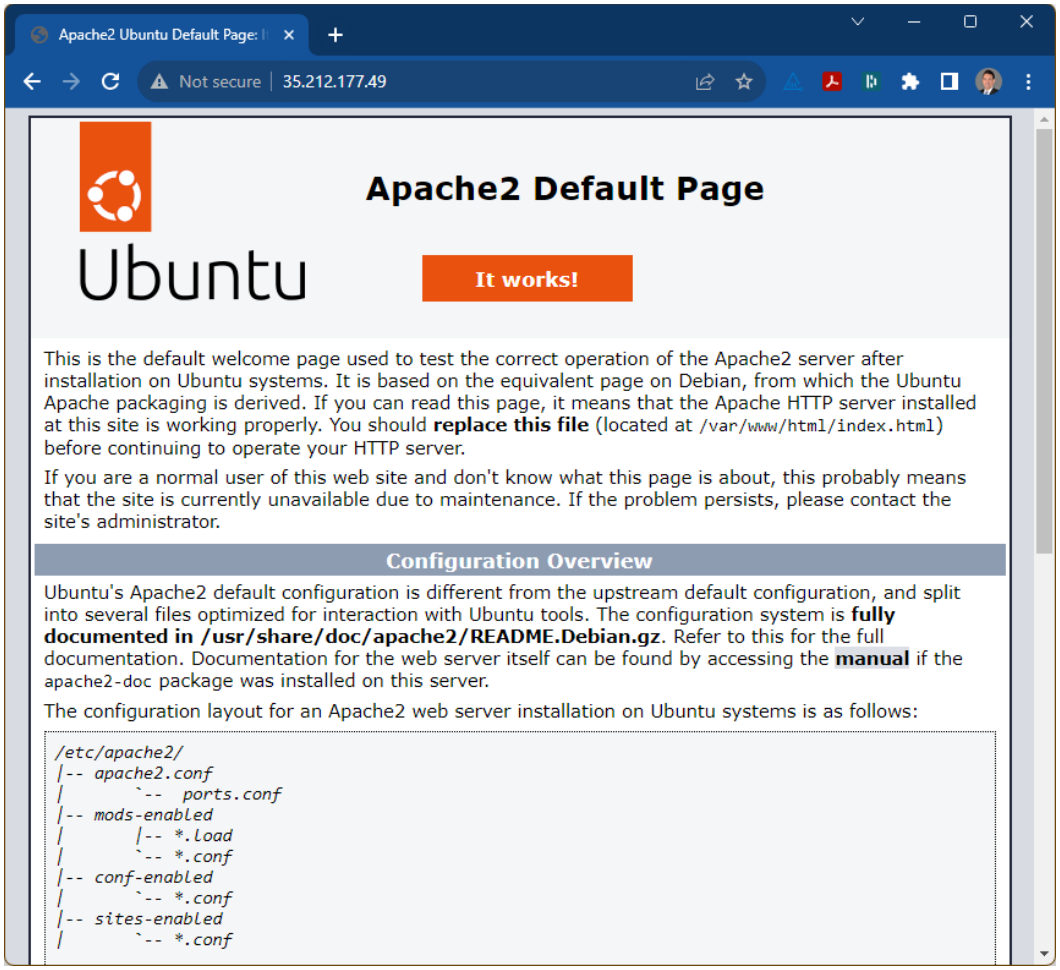
```
$ sudo service apache2 status
```

The state should show as **active (running)**:

```
inf1005-dev@inf1005-lamp:~$ sudo service apache2 status
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2023-10-17 07:16:59 UTC; 50s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 1943 (apache2)
      Tasks: 55 (limit: 1129)
     Memory: 5.1M
        CPU: 41ms
     CGroup: /system.slice/apache2.service
             ├─1943 /usr/sbin/apache2 -k start
             ├─1945 /usr/sbin/apache2 -k start
             └─1946 /usr/sbin/apache2 -k start

Oct 17 07:16:59 inf1005-lamp systemd[1]: Starting The Apache HTTP Server...
Oct 17 07:16:59 inf1005-lamp systemd[1]: Started The Apache HTTP Server.
inf1005-dev@inf1005-lamp:~$
```

You can also enter the server's IP address (the same one used for SSH) in your browser address bar to confirm that the Apache web server is running:



5.3 Lastly, we'll add the `inf1005-dev` user to the `www-data` group so that we can use this account for SFTP later when we wish to deploy our PHP application to the server. Execute the following command:

```
$ sudo usermod -a -G www-data inf1005-dev
```

Important: you'll need to log out (type `exit` at the prompt) and back in for this change to take effect. After re-logging, you should see the group membership using the `groups` command:

```
inf1005-dev@inf1005-lamp:~$ sudo usermod -a -G www-data inf1005-dev
inf1005-dev@inf1005-lamp:~$ exit
logout
Connection to 35.212.177.49 closed.
PS C:\Dev\SIT\INF1005\SampleLabSolutions\Lab08> ssh inf1005-dev@35.212.177.49
inf1005-dev@35.212.177.49's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1016-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Oct 17 07:32:09 UTC 2023

  System load:  0.02734375        Processes:             105
  Usage of /:   24.9% of 9.51GB   Users logged in:       0
  Memory usage: 24%               IPv4 address for ens4: 10.138.0.2
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Tue Oct 17 07:15:51 2023 from 116.87.33.190
inf1005-dev@inf1005-lamp:~$ groups
inf1005-dev sudo www-data
inf1005-dev@inf1005-lamp:~$
```

5.4 Once user `inf1005-dev` is in the `www-data` group, execute the following commands one at a time to set the necessary group permissions in the web directories:

```
$ sudo chown -R inf1005-dev:www-data /var/www/html
$ sudo chmod 2775 /var/www/html
$ find /var/www/html -type d -exec sudo chmod 2775 {} \;
$ find /var/www/html -type f -exec sudo chmod 0664 {} \;
```

6. INSTALLING MYSQL

In this exercise, you'll install MySQL, a very popular open source database management system.

6.1 Enter the following command to install MySQL:

```
$ sudo apt install mysql-server
```

6.2 Verify that MySQL was installed by typing the following command to enter the MySQL monitor:

```
$ sudo mysql
```

You should see the MySQL server version information:

```
inf1005-dev@inf1005-lamp:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.34-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

6.3 With the MySQL monitor still running, enter the following SQL query to configure the root MySQL password (replace YourRootPassword with a strong password of your choosing). Type `exit` when done.

```
mysql>ALTER USER 'root'@'localhost' IDENTIFIED WITH
mysql_native_password BY 'YourRootPassword';
```

e.g.:

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '[redacted]';
Query OK, 0 rows affected (0.18 sec)

mysql> exit
Bye
inf1005-dev@inf1005-lamp:~$
```

6.4 Next, we'll run a script to secure the MySQL installation, which will remove some dangerous defaults and lock down access to your database system:

```
$ sudo mysql_secure_installation
```

When this script runs, you will be prompted to answer **yes (y) or no (n)** to several questions. Following are the prompts with suggested responses:

- Enter password for user root: (**enter the password you set in 6.3 above**)

- Would you like to setup VALIDATE PASSWORD component? **n**
- Change the password for root? **n**
- Remove anonymous users? **y**
- Disallow root login remotely? **y**
- Remove test database and access to it? **y**
- Reload privilege tables now? **y**

6.5 This is all we need to do for MySQL for this Lab, but in the next Lab on database operations, we'll continue by adding a non-root MySQL user with required privileges.

7. INSTALLING PHP

In this exercise, we'll install PHP. In our earlier Labs on PHP, we installed it locally and ran our websites using PHP's built-in web server. This is suitable for development and testing, but in a production environment you'll need to publish your PHP applications to a proper web server and run them from there.

7.1 Run the following commands to install PHP along with some helper packages:

```
$ sudo apt install php libapache2-mod-php php-mysql
```

7.2 Once the installation completes successfully, you can confirm your PHP version:

```
$ php -v
```

```
inf1005-dev@inf1005-lamp:~$ php -v
PHP 8.1.2-1ubuntu2.14 (cli) (built: Aug 18 2023 11:41:11) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.14, Copyright (c), by Zend Technologies
inf1005-dev@inf1005-lamp:~$
```

7.3 Next, we'll edit the Apache configuration file so that the server looks for `index.php` before `index.html`. Enter the command below to open the configuration file in the nano text editor:

```
$ sudo nano /etc/apache2/mods-enabled/dir.conf
```

Then move `index.php` to the head of the list of default files. When done editing, hit Ctrl-X to save and close the editor (hit `y` and ENTER to confirm).

```
 GNU nano 6.2                  /etc/apache2/mods-enabled/dir.conf *
<IfModule mod_dir.c>
        DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
```

7.4 Restart the Apache server for the above to take effect:

```
$ sudo systemctl restart apache2
```

7.5 Lastly, we'll use the nano editor again to create a PHP file on the server so we can test our installation.

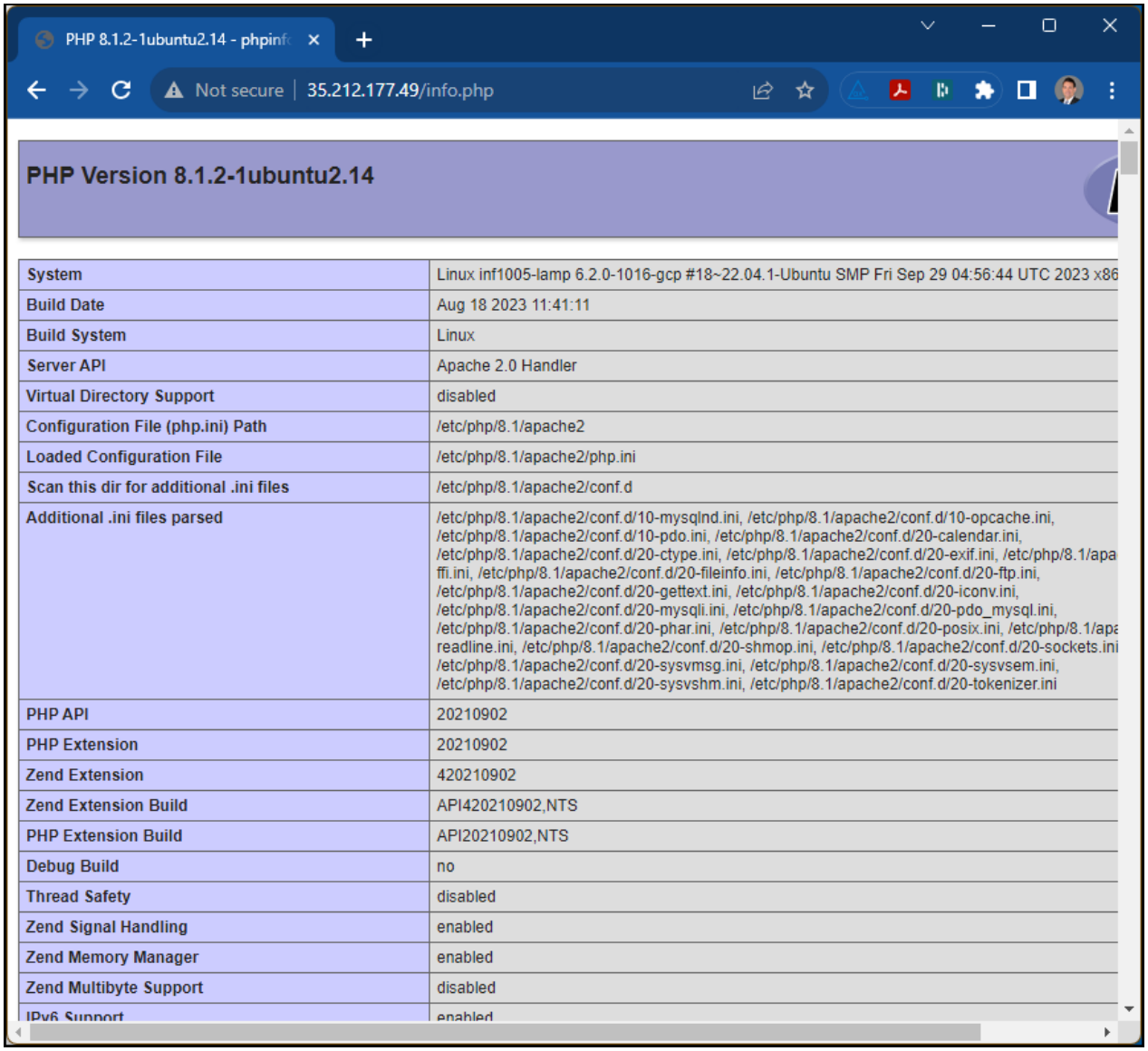
Open the nano editor and create a new file called `info.php`:

```
$ sudo nano /var/www/html/info.php
```

Enter the following PHP code in the file:

```
<?php
   phpinfo();
?>
```

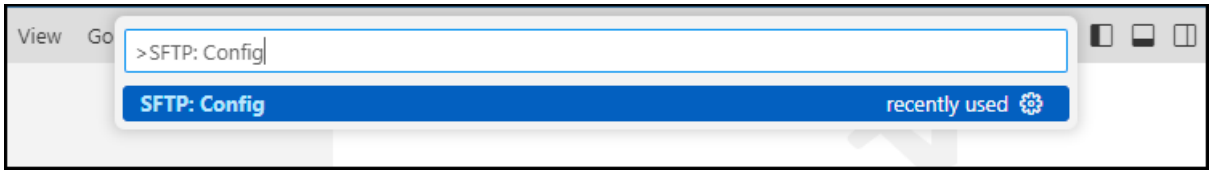
Hit Ctrl-X to save and close the file, then open your web browser and enter the IP address again, followed by `/info.php`. You should see the familiar PHP info page:



8. DEPLOYING A PHP APPLICATION TO THE SERVER

8.1 Now that our LAMP stack is installed and our web server is ready for business, we will learn how to deploy (publish) a PHP application to the server. First, make a copy of your latest World of Pets website from the previous Lab and open the folder in Visual Studio Code.

8.2 Ensure that you have installed the SFTP extension for Visual Studio Code, then type Ctrl+Shift+P to open the Command Palette. Enter `SFTP: Config` and hit ENTER.



8.3 This will open the SFTP configuration file in the VS Code editor. Make the following changes to this file:

a. Change the `name` to something more appropriate.

b. Set `host` to the static IP address of your LAMP server.

c. Set `username` to the user you created in section 4.3 (e.g., `inf1005-dev`).

d. Change `remotePath` to `/var/www/html`, which is the Apache web server folder.

e. Add `"interactiveAuth": true,` - this fixes the "All configured authentication methods failed" error.

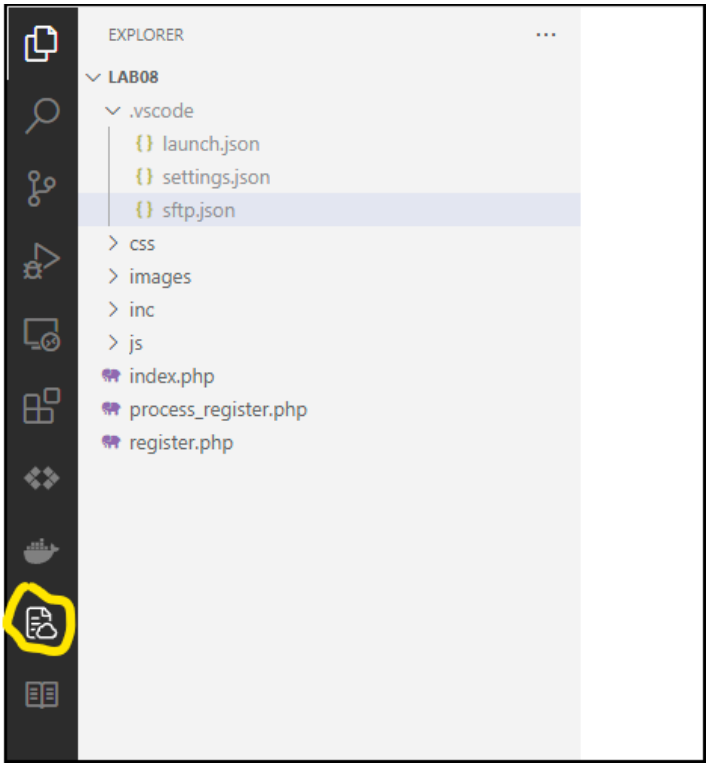
f. Add an `ignore` setting so that SFTP will not transfer local configuration files to the server.

Example:

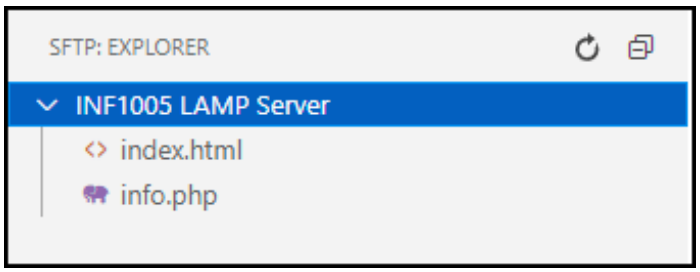
```
{
    "name": "INF1005 LAMP Server",
    "host": "35.212.177.49",
    "protocol": "sftp",
    "port": 22,
    "username": "inf1005-dev",
    "remotePath": "/var/www/html",
    "uploadOnSave": false,
    "useTempFile": false,
    "openSsh": false,
    "interactiveAuth": true,
    "ignore": [".vscode", ".git"]
}
```

*Change to your IP address.

8.4 Save the config file and you should now see the SFTP icon added to the left:

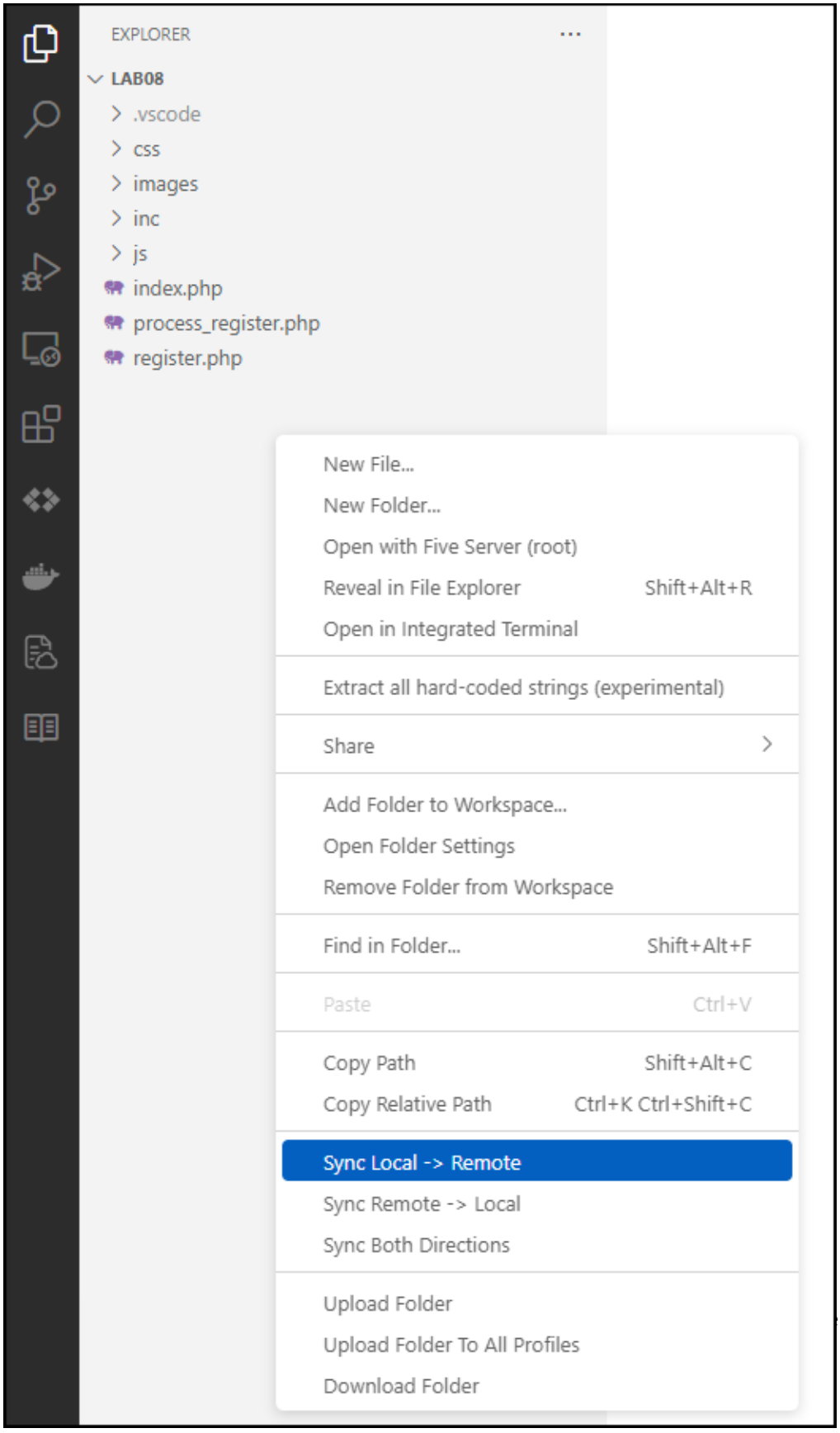


8.5 Click on the **SFTP** icon to open the **SFTP: EXPLORER**, then click the arrow button next to **INF1005 LAMP Server** (or whatever name you used in the previous step) to initiate a secure connection to your server. Enter your password at the prompt and you should see a directory listing from the server:

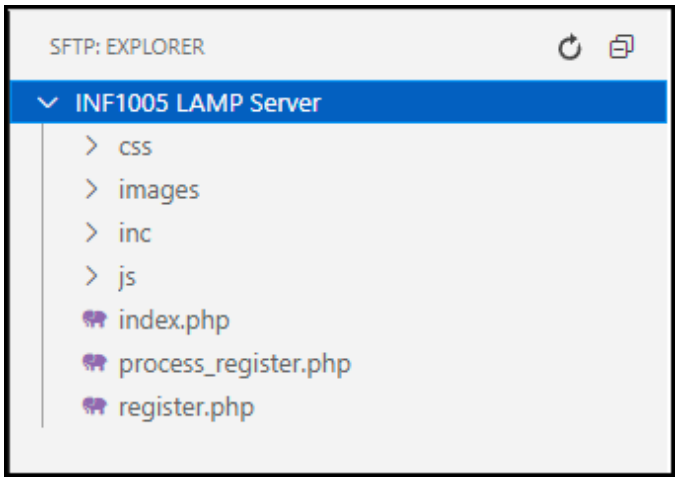


8.6 This corresponds to the `var/www/html` folder on the remote server, which we configured in the SFTP settings file with the `remotePath` setting. The two files there were created earlier when we installed Apache and PHP. You may proceed to delete both of these files (Right-Click on the file and choose **Delete** from the popup menu). We will replace them with the files from our World of Pets application in the next step.

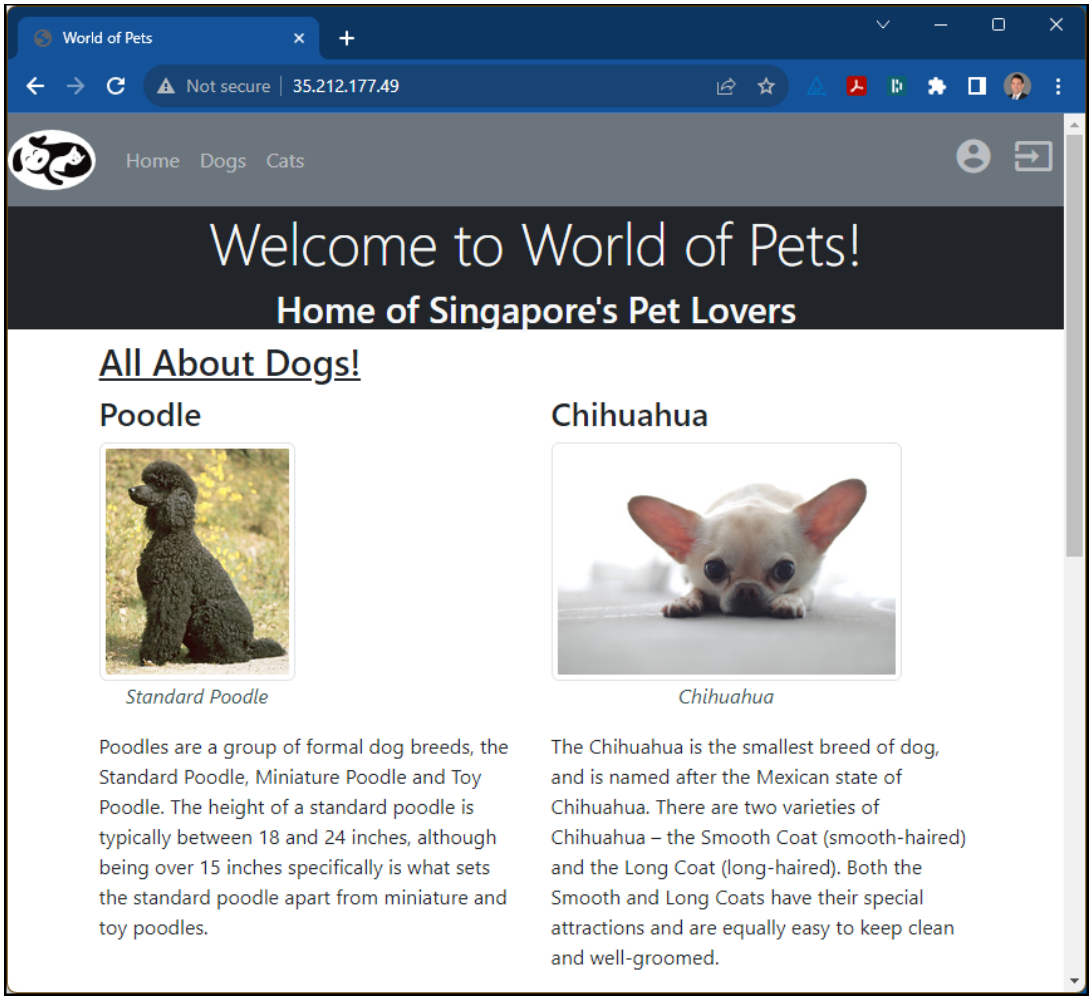
8.7 Go to the **Explorer** view in VS Code and Right-Click below the file list in the **Explorer** pane, and select **Sync Local -> Remote** to transfer the local files to the server:


EXPLORER
LAB08
.vscode
css
images
inc
js
index.php
process_register.php
register.php
New File...
New Folder...
Open with Five Server (root)
Reveal in File Explorer Shift+Alt+R
Open in Integrated Terminal
Extract all hard-coded strings (experimental)
Share
Add Folder to Workspace...
Open Folder Settings
Remove Folder from Workspace
Find in Folder... Shift+Alt+F
Paste Ctrl+V
Copy Path Shift+Alt+C
Copy Relative Path Ctrl+K Ctrl+Shift+C
Sync Local -> Remote
Sync Remote -> Local
Sync Both Directions
Upload Folder
Upload Folder To All Profiles
Download Folder

8.8 You can monitor the file upload progress in VS Code's **OUTPUT** window. When completed, return to the **SFTP EXPLORER** view, click the Refresh icon at the top, and you should see all of your project files on the remote server:



8.9 Finally, enter your static IP address in your browser and you should see your World of Pets application running on the remote server. Congratulations!



9. SUBMISSION OF LAB ASSIGNMENT

9.1 In order to receive credit for this Lab assignment, you must submit the following to xSiTe LMS (**Assessments->DropBox**) before the end of the Lab session:

- Screenshot of your Google Console showing the Ubuntu VM instance that you created in the exercise. This should show the instance running with status green, and the IP address should be visible.
- Screenshot of your World of Pets application running from your LAMP server.

10. REFERENCES

10.1 For more references on installing LAMP stack, refer to:

a. Setting Up LAMP on Compute Engine (https://cloud.google.com/community/tutorials/setting-up-lamp)

b. Initial Server Setup with Ubuntu 22.04 (https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-22-04)

c. How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 22.04 (https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-22-04).