Tutorial/Laboratory 07 (SDL)

Installing LAMP Stack on Ubuntu Server 18.04

1. LEARNING OUTCOMES

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

- Create an Ubuntu Server 18.04 virtual machine in the Amazon Web Services (AWS) cloud.
- Install the standard LAMP (Linux, Apache, MySQL, PHP) stack on Ubuntu Server.
- Deploy a PHP application to the server using NetBeans IDE.

2. REQUIRED SOFTWARE

- Apache NetBeans 11.2 (or later): https://netbeans.apache.org/download/index.html
- FireFox (https://www.mozilla.org/en-US/firefox/new/) or Chrome (https://www.google.com/chrome/) web browser.
- Activated AWS Educate account (https://www.awseducate.com/signin/SiteLogin).

3. INSTRUCTIONS

3.1 This document combines the Lecture, Tutorial and Laboratory for Week 8. You are to complete the exercises on your own via self-directed learning (SDL). Note that this material will be tested on Quiz 2 and the Exam.

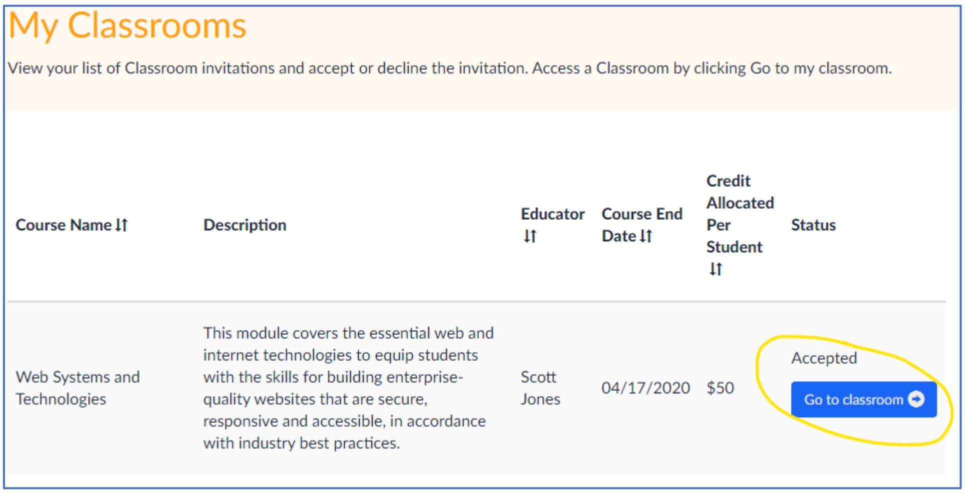
4. EXERCISE 1: LAUNCHING AN EC2 INSTANCE FROM A SHARED AMI

In this exercise, you will launch an Amazon EC2 instance (virtual machine) of Ubuntu 18.04 Server from a shared Amazon Machine Image (AMI).

You should have received an invitation by email to join the ICT1004 class on AWS Educate (check your junk/spam folder if you did not receive).

You will need to complete the AWS Educate sign-up process before proceeding with this exercise.

4.1 Log into AWS Educate and click “My Classrooms” in the menu at the top, then click the “Go to classroom” button:



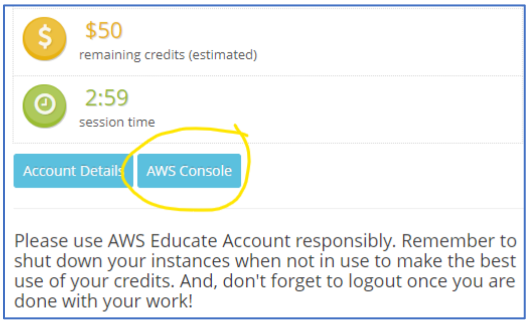
My Classrooms

View your list of Classroom invitations and accept or decline the invitation. Access a Classroom by clicking Go to my classroom.

Course Name ↓↑	Description	Educator ↓↑	Course End Date ↓↑	Credit Allocated Per Student ↓↑	Status
Web Systems and Technologies	This module covers the essential web and internet technologies to equip students with the skills for building enterprise-quality websites that are secure, responsive and accessible, in accordance with industry best practices.	Scott Jones	04/17/2020	$50	Accepted Go to classroom ➔

If this is the first time you've accessed your account, you'll need to accept the terms & conditions before proceeding.

4.2 Click on the "AWS Console" button to start the AWS Management Console:

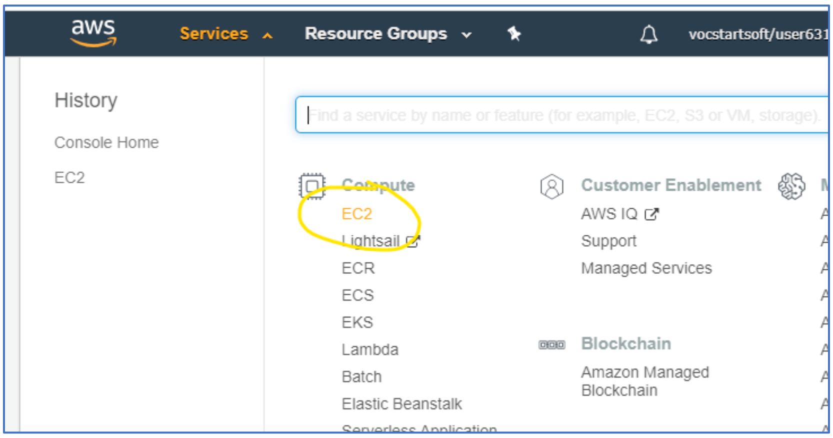


$50
remaining credits (estimated)

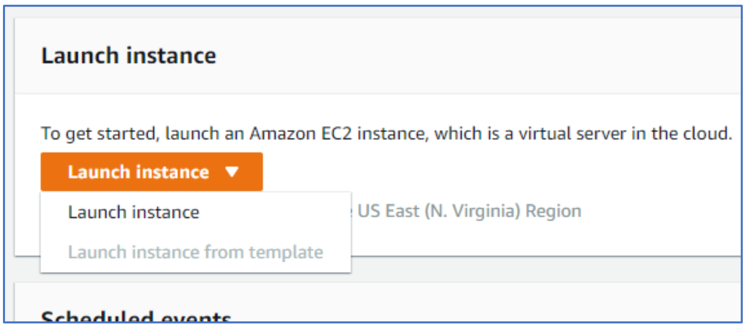
2:59
session time

Please use AWS Educate Account responsibly. Remember to shut down your instances when not in use to make the best use of your credits. And, don't forget to logout once you are done with your work!

4.3 Expand the "Services" menu at the top of the page and select "EC2" from the "Compute" category:



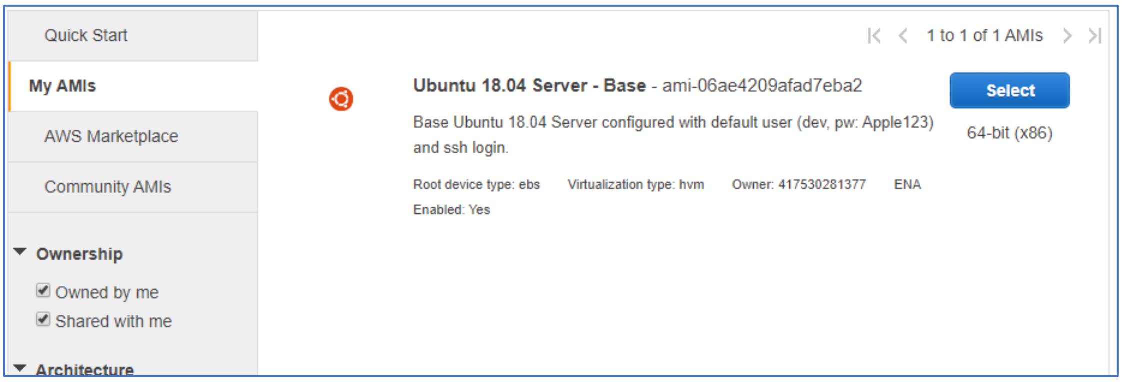
4.4 Click the "Launch instance" button and select "Launch instance" from the drop-down menu:



Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

4.5 From the tabs on the left, select “My AMIs” to expand the AMI panel, then make sure the “Shared with me” box is checked. You should see an Ubuntu 18.04 Server AMI image that has been shared with the ICT1004 class:



Ownership

☑ Owned by me

☑ Shared with me

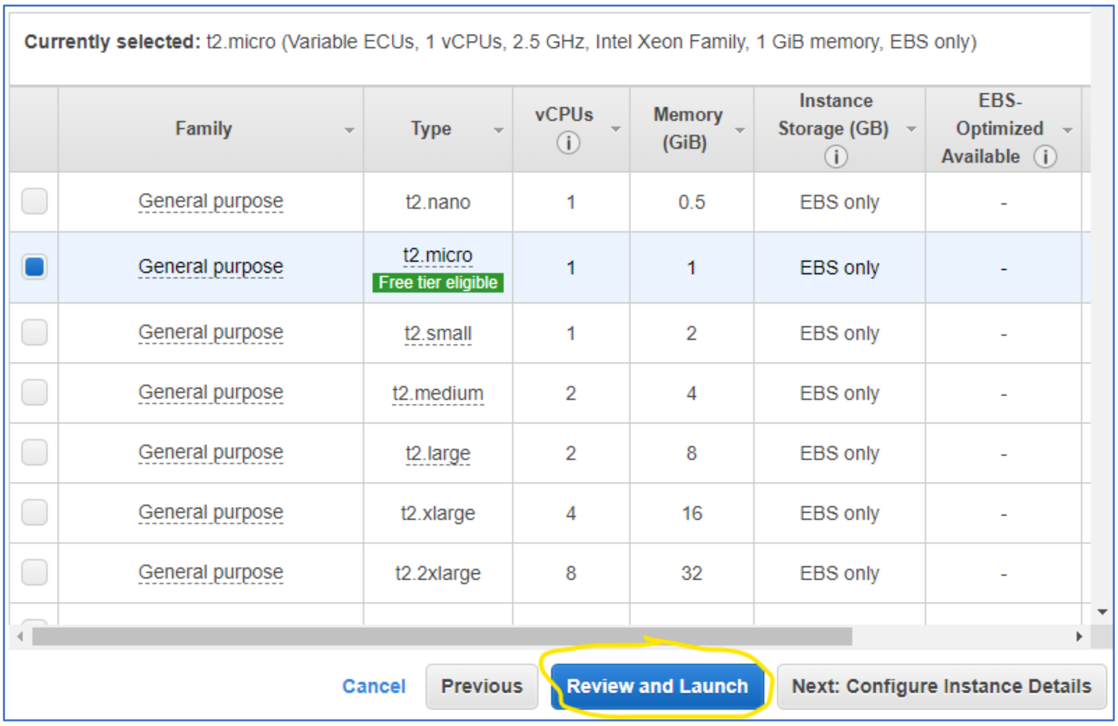
Ubuntu 18.04 Server - Base - ami-06ae4209afad7eba2

Base Ubuntu 18.04 Server configured with default user (dev, pw: Apple123) and ssh login.

64-bit (x86)

Root device type: ebs Virtualization type: hvm Owner: 417530281377 ENA Enabled: Yes

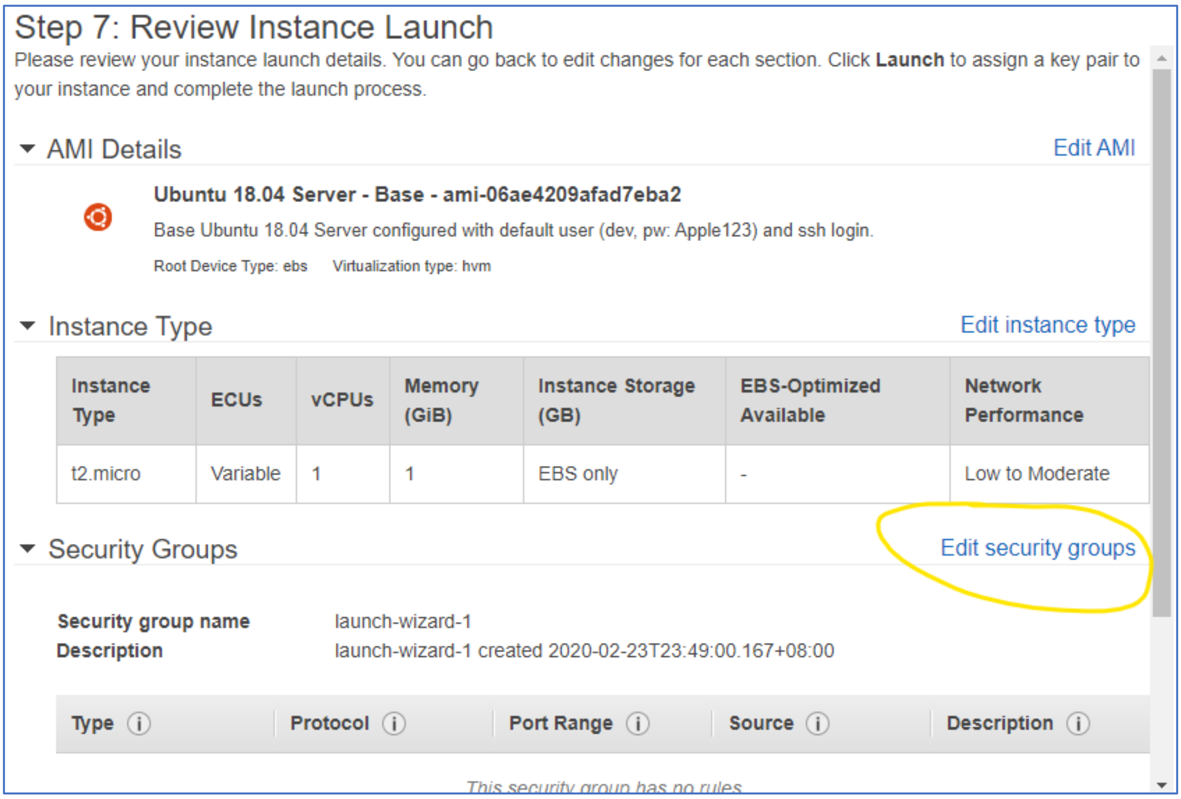
4.6 Click the “Select” button to configure your copy of the instance. You can keep the default settings and click the “Review and Launch” button:



Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available
□	General purpose	t2.nano	1	0.5	EBS only	-
☑	General purpose	t2.micro Free tier eligible	1	1	EBS only	-
□	General purpose	t2.small	1	2	EBS only	-
□	General purpose	t2.medium	2	4	EBS only	-
□	General purpose	t2.large	2	8	EBS only	-
□	General purpose	t2.xlarge	4	16	EBS only	-
□	General purpose	t2.2xlarge	8	32	EBS only	-

4.7 On the next page, click “Edit security groups” to configure firewall settings:



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Ubuntu 18.04 Server - Base - ami-06ae4209afad7eba2

Base Ubuntu 18.04 Server configured with default user (dev, pw: Apple123) and ssh login.

Root Device Type: ebs   Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

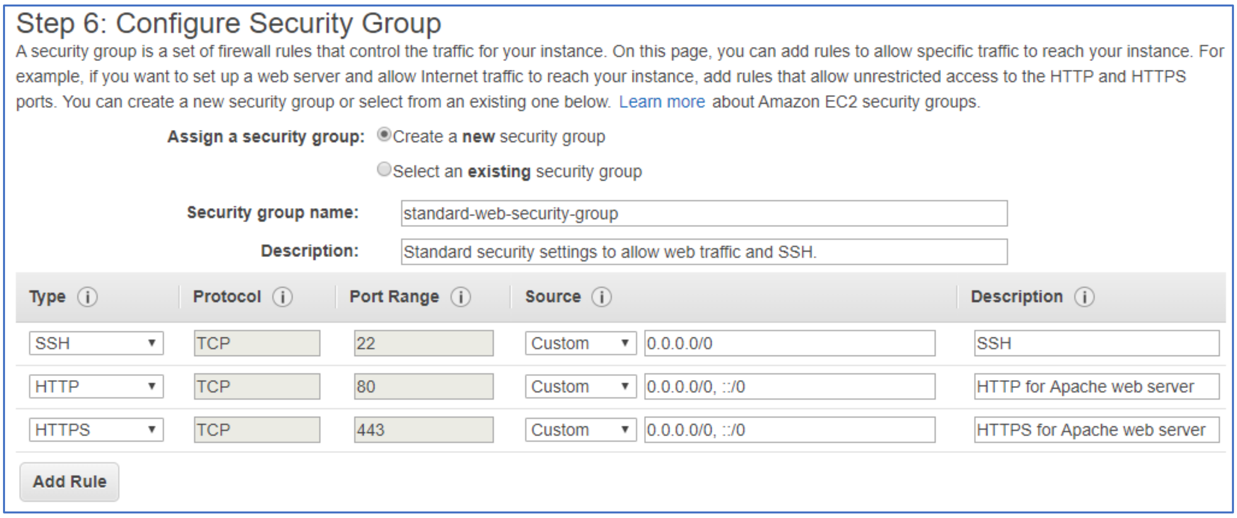
Security Groups

Security group name launch-wizard-1
Description launch-wizard-1 created 2020-02-23T23:49:00.167+08:00

Type	Protocol	Port Range	Source	Description

This security group has no rules

4.8 By default, the SSH port should already be configured. We also need to open ports for HTTP and HTTPS to allow web traffic. Click the “Add Rule” button to add these, then give your security group a name and description of your choosing. When finished, your settings should look similar to this:



Step 6: Configure Security Group

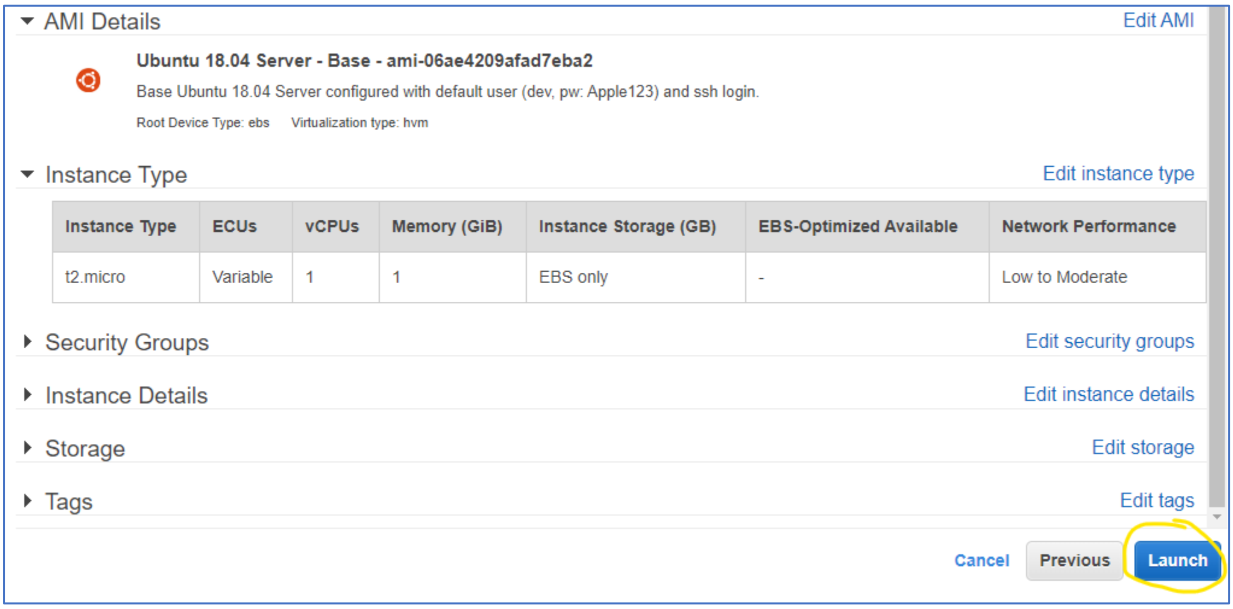
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: standard-web-security-group
Description: Standard security settings to allow web traffic and SSH.

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	SSH
HTTP	TCP	80	Custom	0.0.0.0/0, ::/0	HTTP for Apache web server
HTTPS	TCP	443	Custom	0.0.0.0/0, ::/0	HTTPS for Apache web server

4.9 Click the “Review and Launch” button. After reviewing your settings, click the “Launch” button:



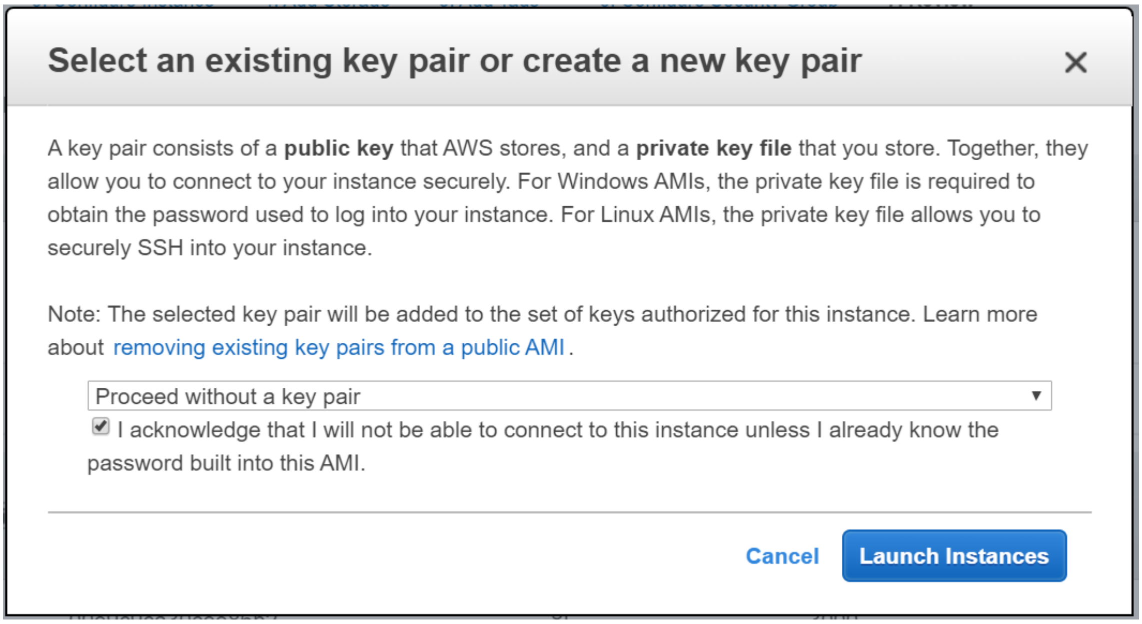
Ubuntu 18.04 Server - Base - ami-06ae4209afad7eba2

Base Ubuntu 18.04 Server configured with default user (dev, pw: Apple123) and ssh login.

Root Device Type: ebs Virtualization type: hvm

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

4.10 On the next screen, you'll be prompted to create a key pair, which provides a secure way of logging into new instances. However, the AMI that we're cloning already has a pre-configured user/password that we can use to log in, so you can select "Proceed without a key pair" from the drop-down menu and check the acknowledgement box:



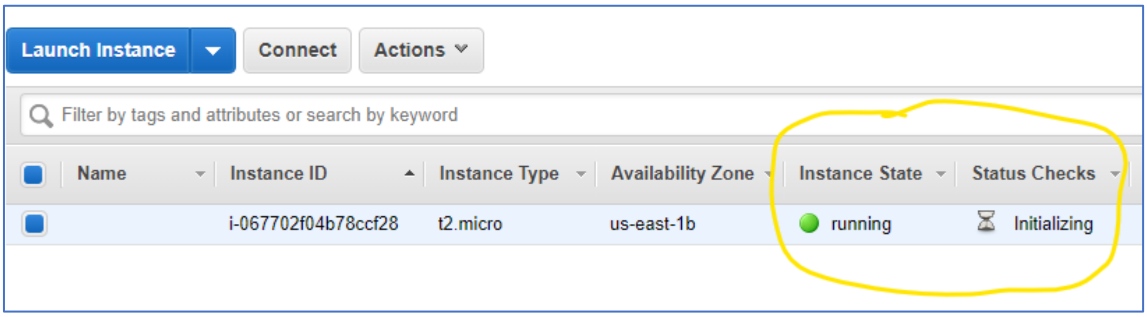
A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Proceed without a key pair ▾

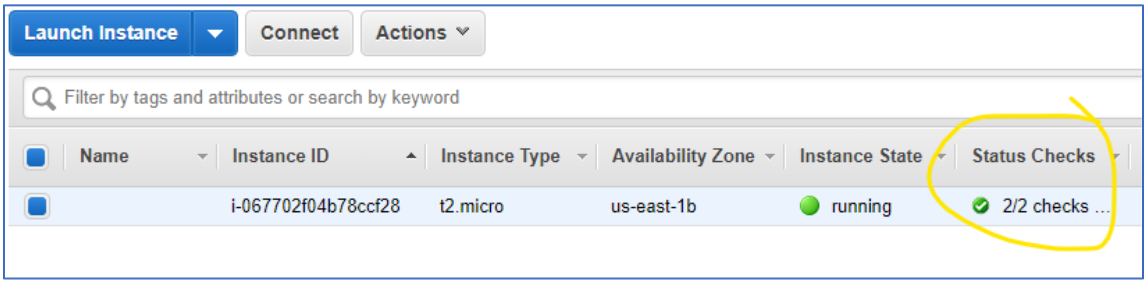
☑ I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI.

4.11 Finally, click the "Launch Instances" button to generate your new Ubuntu 18.04 Server virtual machine. This process may take several minutes. Click the "View Instances" button and observe the "Instance State" and "Status Checks":



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
	i-067702f04b78ccf28	t2.micro	us-east-1b	● running	⌛ Initializing

Initializing



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
	i-067702f04b78ccf28	t2.micro	us-east-1b	● running	✓ 2/2 checks ...

Status Checks Complete

4.12 Once the status checks are complete and the state shows the green running indicator, your instance is ready to use.

5. EXERCISE 2: LOGGING INTO THE UBUNTU SERVER INSTANCE

5.1 With our server instance up and running, we can now use SSH to log into the virtual machine remotely. You may use any SSH client you wish, including:

a. Windows PowerShell (Windows users)

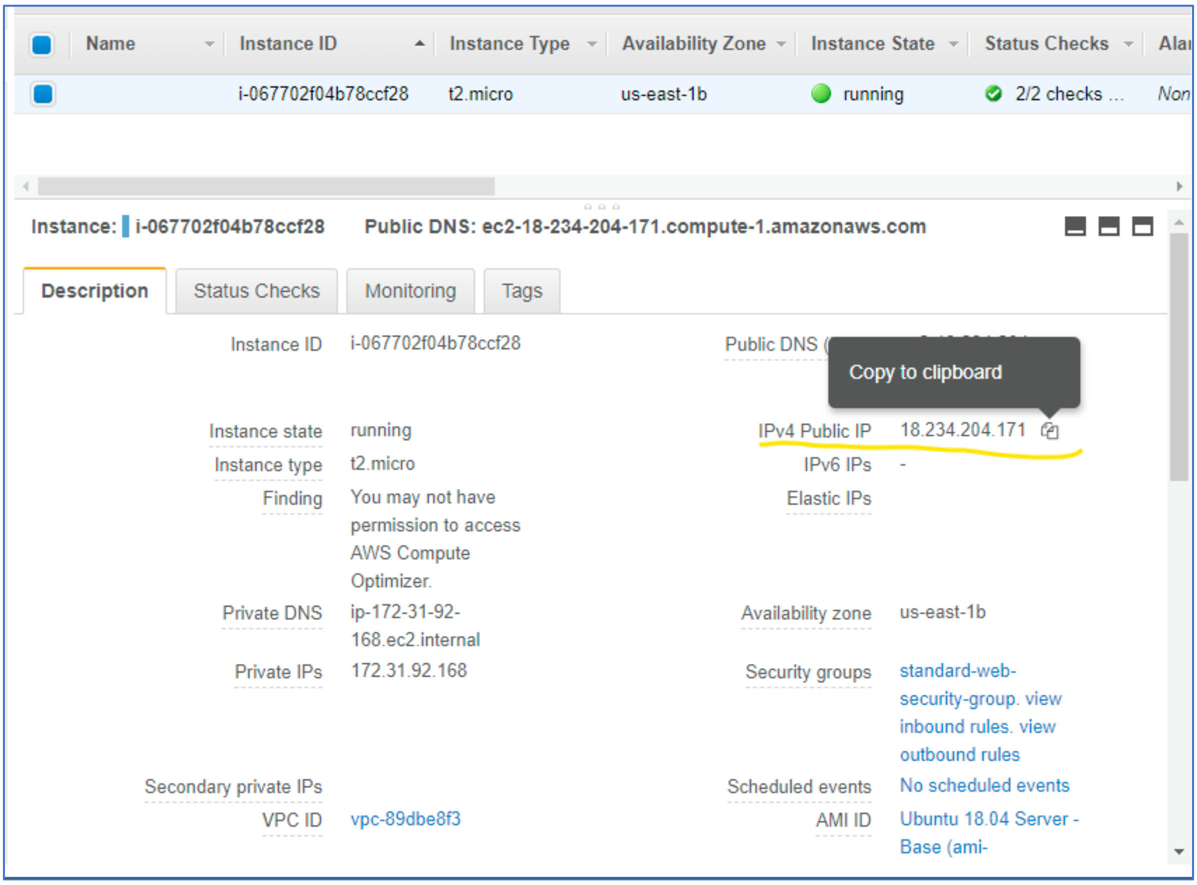
b. Putty (Windows & Linux)

c. Built-in terminal program (Mac and Linux)

We won’t be doing anything fancy so any of these will do.

Important note: SSH is blocked by the firewall on the school’s SIT Wi-fi, so you’ll have to either connect to the ICT Wi-fi or connect from outside the school network (e.g. your mobile hotspot).

5.2 In order to connect via SSH, you need to first determine the IP address of your Ubuntu instance. To do this, click on the “Description” tab in the console and copy the “IPv4 Public IP”:



	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alaı
■		i-067702f04b78ccf28	t2.micro	us-east-1b	● running	✓ 2/2 checks ...	Non

Instance: i-067702f04b78ccf28 Public DNS: ec2-18-234-204-171.compute-1.amazonaws.com

Instance ID	i-067702f04b78ccf28	Public DNS (	
Instance state	running	IPv4 Public IP	18.234.204.171
Instance type	t2.micro	IPv6 IPs	-
Finding	You may not have permission to access AWS Compute Optimizer.	Elastic IPs	
Private DNS	ip-172-31-92-168.ec2.internal	Availability zone	us-east-1b
Private IPs	172.31.92.168	Security groups	standard-web-security-group. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-89dbe8f3	AMI ID	Ubuntu 18.04 Server - Base (ami-

Note: the IP address is dynamically assigned and will likely change each time you stop and restart the instance. Therefore be sure to check and use the current IP address when connecting in the future.

You may also utilize the AWS “Elastic IP” option, which allows you to configure a static (permanent) IP address.

5.3 With your SSH client open, enter the following command - **be sure to replace the IP address with the address of your instance**:

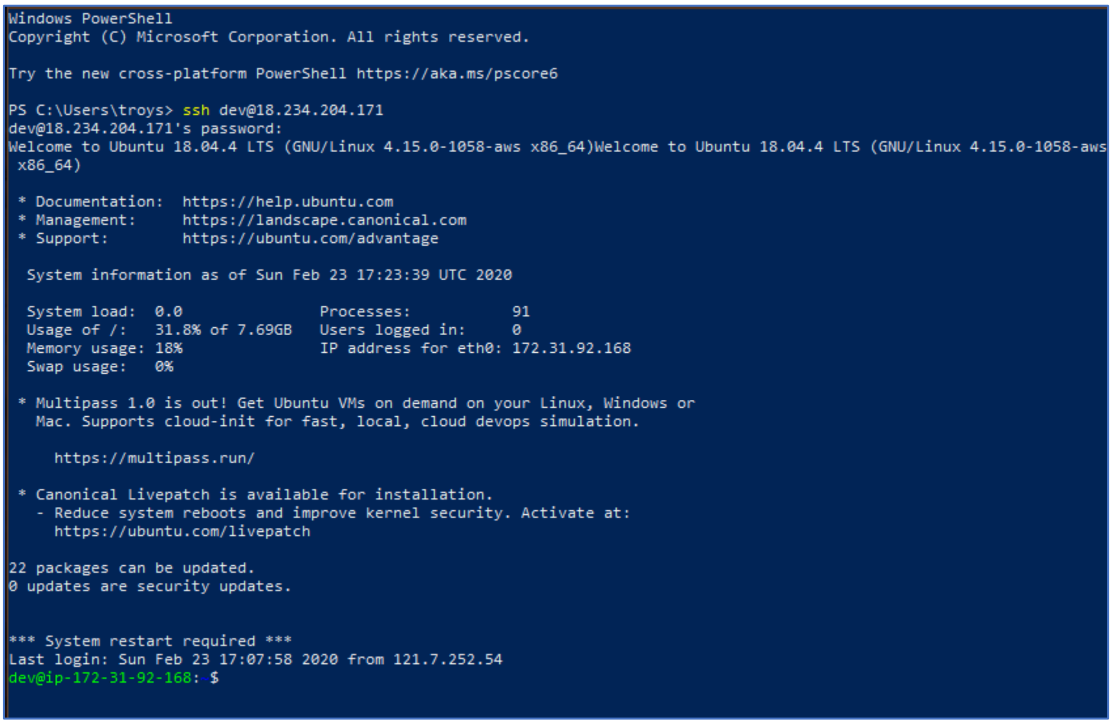
```
ssh dev@18.234.204.171
```

Answer ‘yes’ to any security prompts, then enter the following password:

Apple123

This logs you into the server as username **dev**, which was pre-configured in the AMI that you cloned. The dev account has administrative privileges and can be used to add more users or perform any other tasks.

5.4 If your login was successful, you should now be at the Ubuntu terminal prompt:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\troys> ssh dev@18.234.204.171
dev@18.234.204.171's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1058-aws x86_64)Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1058-aws
 x86_64)

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

  System information as of Sun Feb 23 17:23:39 UTC 2020

  System load:  0.0               Processes:           91
  Usage of /:   31.8% of 7.69GB   Users logged in:     0
  Memory usage: 18%               IP address for eth0: 172.31.92.168
  Swap usage:   0%

 * Multipass 1.0 is out! Get Ubuntu VMs on demand on your Linux, Windows or
   Mac. Supports cloud-init for fast, local, cloud devops simulation.

     https://multipass.run/

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

22 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Sun Feb 23 17:07:58 2020 from 121.7.252.54
dev@ip-172-31-92-168:~$
```

5.5 Now would be a good idea to change the password! To do that, use the following command:

```
sudo passwd dev
```

6. EXERCISE 3: INSTALLING APACHE WEB SERVER

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

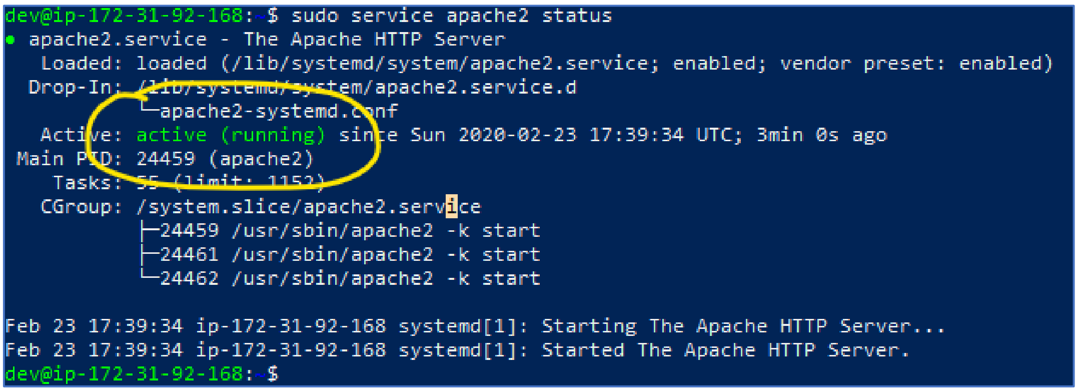
Enter the following commands one at a time at the terminal prompt (you will need to enter your password the first time you issue a sudo command):

```
$ sudo apt update
$ sudo apt upgrade
$ sudo apt install apache2
```

6.2 Check that the Apache installation was successful:

```
$ sudo service apache2 status
```

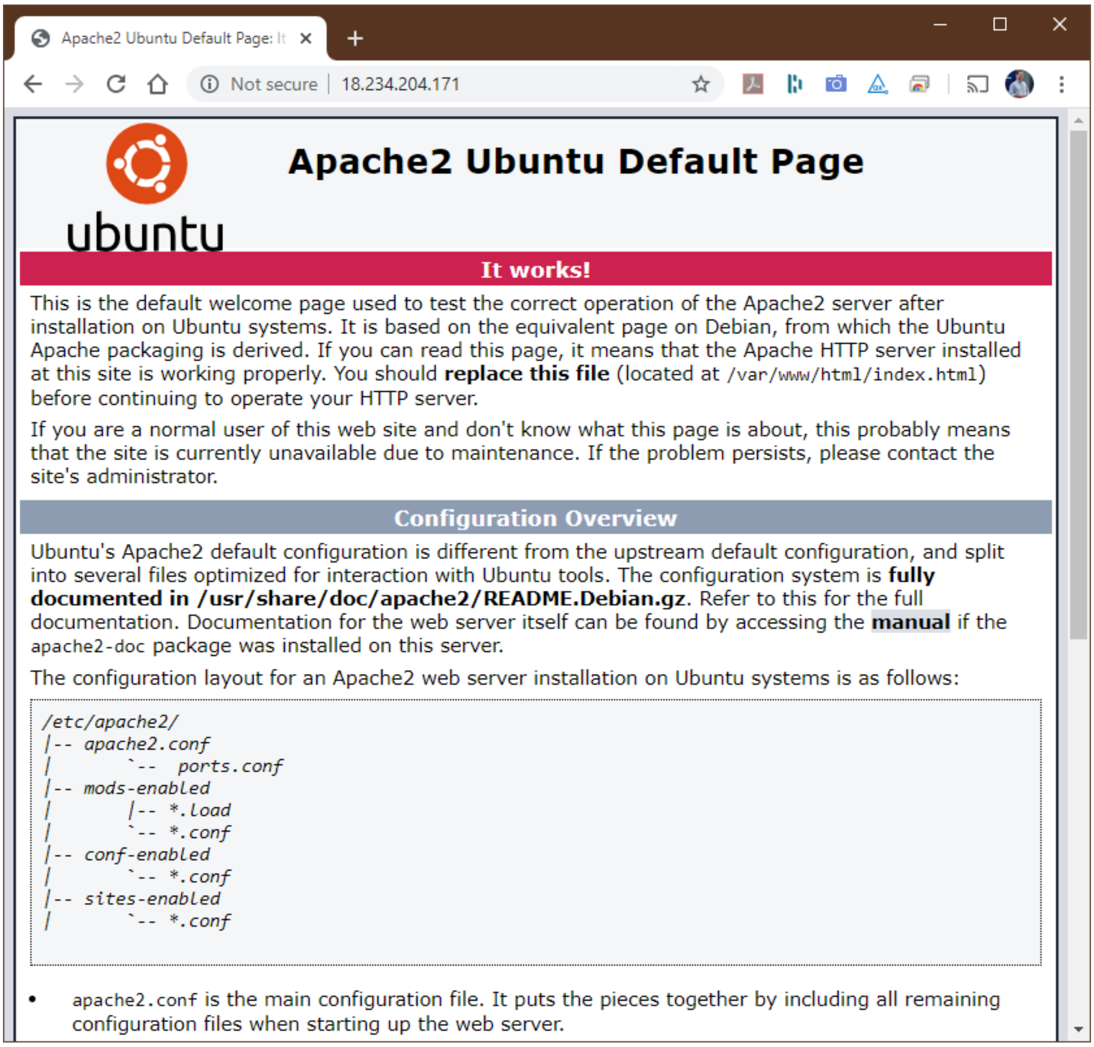
The state should show as **active**:



```
dev@ip-172-31-92-168:~$ sudo service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Sun 2020-02-23 17:39:34 UTC; 3min 0s ago
 Main PID: 24459 (apache2)
    Tasks: 55 (limit: 1152)
   CGroup: /system.slice/apache2.service
           ├─24459 /usr/sbin/apache2 -k start
           ├─24461 /usr/sbin/apache2 -k start
           └─24462 /usr/sbin/apache2 -k start

Feb 23 17:39:34 ip-172-31-92-168 systemd[1]: Starting The Apache HTTP Server...
Feb 23 17:39:34 ip-172-31-92-168 systemd[1]: Started The Apache HTTP Server.
dev@ip-172-31-92-168:~$
```

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:



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

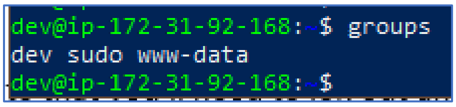
```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.

6.3 Lastly, we'll add the **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 dev
```

Note that you'll need to log out and back in for this change to take effect. After re-logging, you should see the group membership using the **groups** command:



```
dev@ip-172-31-92-168: $ groups
dev sudo www-data
dev@ip-172-31-92-168: $
```

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

```
$ sudo chown -R 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 {} \;
```

7. EXERCISE 4: INSTALLING MYSQL

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

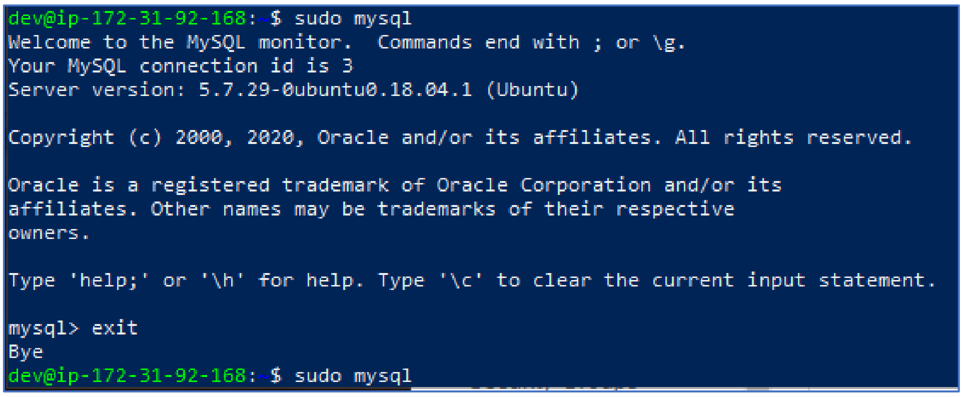
Enter the following command to install MySQL:

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

7.2 Verify that MySQL was installed by typing:

```
$ sudo mysql
```

You should see the MySQL server version information (type "exit" when done):



```
dev@ip-172-31-92-168: $ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.29-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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> exit
Bye
dev@ip-172-31-92-168: $ sudo mysql
```

7.3 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. Follow the example below to run the script and answer the prompts (responses are highlighted in green).

When asked to enter the MySQL root password, enter any password of your choosing, but it should be strong/unique. Do NOT leave it blank.

```
dev@ip-172-31-92-168:~$ sudo mysql_secure_installation
[sudo] password for dev:

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?

Press y|Y for Yes, any other key for No: n
Please set the password for root here.

New password:

Re-enter new password:
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.


Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.


Remove test database and access to it? (Press y|Y for Yes, any other key for No) :
y
 - Dropping test database...
Success.

 - Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
dev@ip-172-31-92-168:~$
```

8. EXERCISE 5: INSTALLING PHP

8.1 In this exercise, we'll install PHP.

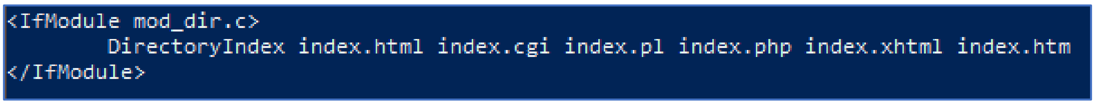
In our earlier Labs on PHP, we installed it locally and ran our websites using PHP's embedded web server. This is suitable for development and testing, but in a production environment, you'll need to publish your PHP applications to a proper web server and run them from there.

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

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

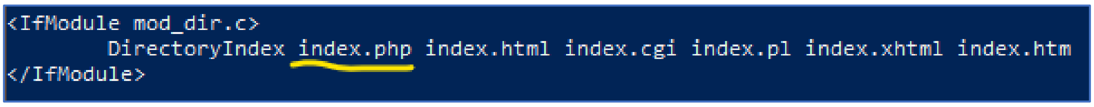
8.3 Next, we'll edit the Apache configuration file so that the server looks for **index.php** before **index.html**. Enter the command below to open the configuration file in the nano text editor, then move **index.php** to the head of the list of default files. When done editing, hit CTRL-X to save and close the editor (hit 'y' and ENTER to confirm).

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



```
<IfModule mod_dir.c>
        DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>
```

Before



```
<IfModule mod_dir.c>
        DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
```

After

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

```
$ sudo systemctl restart apache2
```

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

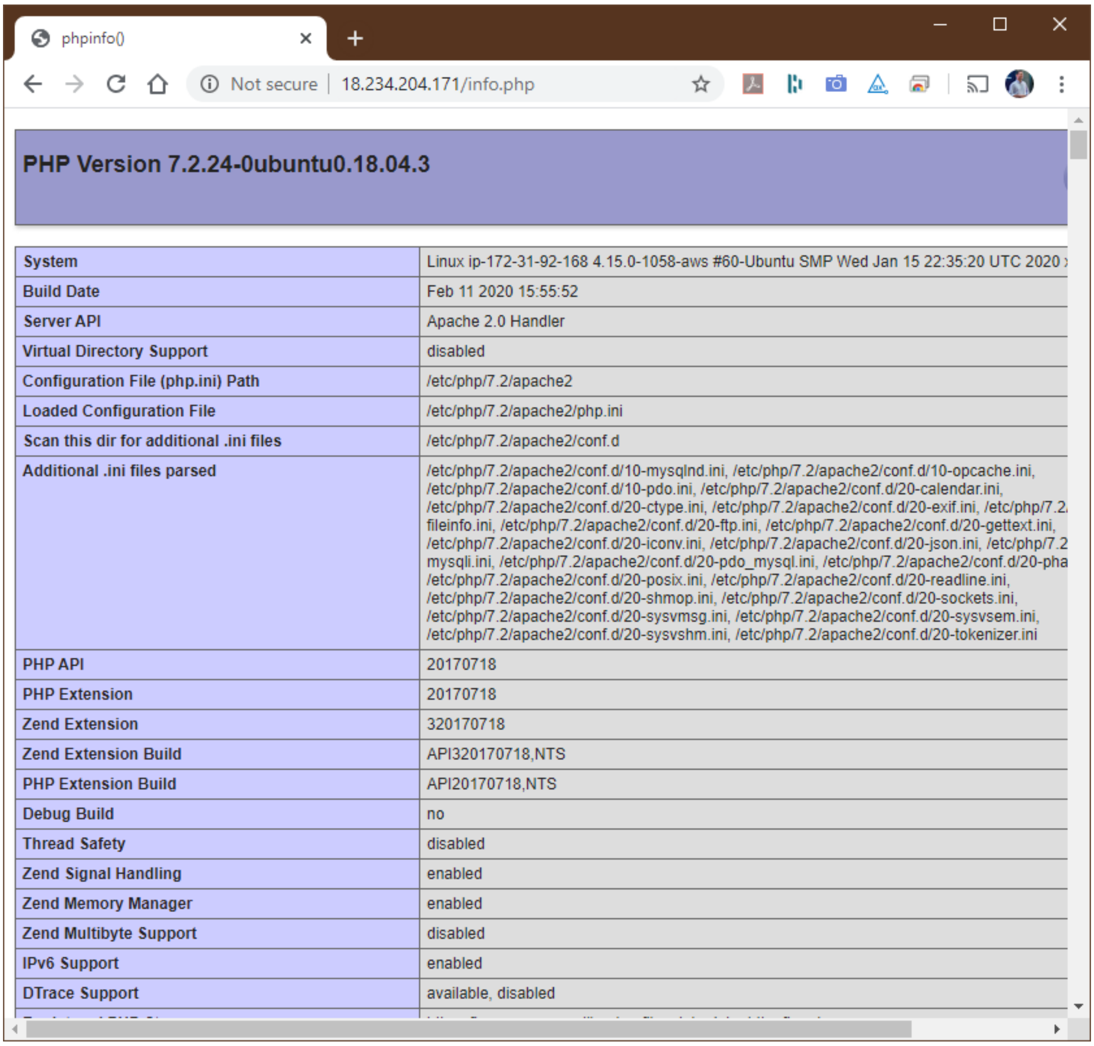
Open the nano editor again but this time create a new file called **info.php**:

```
$ 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:



PHP Version 7.2.24-0ubuntu0.18.04.3

System	Linux ip-172-31-92-168 4.15.0-1058-aws #60-Ubuntu SMP Wed Jan 15 22:35:20 UTC 2020
Build Date	Feb 11 2020 15:55:52
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2. fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2 mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-pha /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled

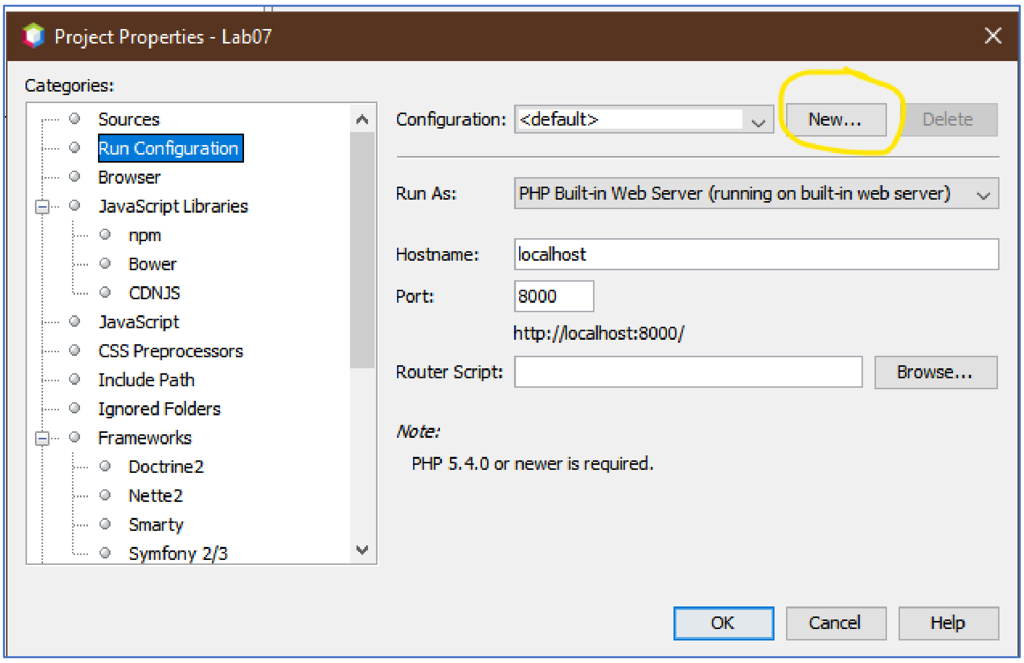
9. EXERCISE 6: DEPLOYING A PHP APPLICATION TO THE SERVER

9.1 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 using NetBeans IDE.

9.2 Open NetBeans and make a copy of your latest World of Pets website from the previous Labs.

9.3 Open the **Project Properties** and select “Run Configuration” from the Categories. The default configuration that we’ve been using will run the PHP application on the local built-in server, so we will create a new configuration for deploying/running on the remote server.

a. Click the “New” button next to configuration:



Project Properties - Lab07

Categories:

- Sources
- Run Configuration
- Browser
- JavaScript Libraries
  - npm
  - Bower
  - CDNJS
- JavaScript
- CSS Preprocessors
- Include Path
- Ignored Folders
- Frameworks
  - Doctrine2
  - Nette2
  - Smarty
  - Symfony 2/3

Configuration: <default>

Run As: PHP Built-in Web Server (running on built-in web server)

Hostname: localhost

Port: 8000

http://localhost:8000/

Router Script:

Note:

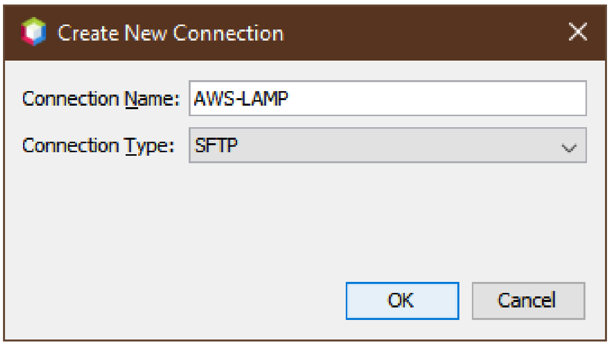
PHP 5.4.0 or newer is required.

b. Enter a name for the configuration (e.g. “Remote SFTP”) and click “OK”.

c. Change the **Run As:** option to “Remote Web Site (FTP, SFTP)”.

d. Next to **Remote Connections**, click the “Manage…” button.

e. Click “Add…” to create a new connection. Enter the name (e.g. “AWS-LAMP”) and select SFTP as the Connection Type.

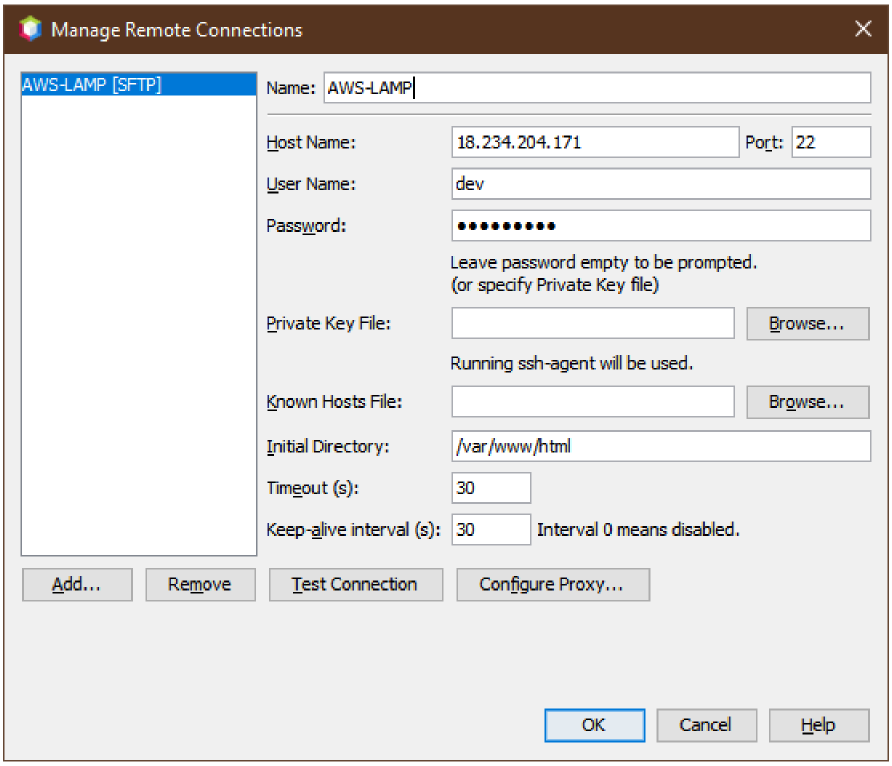


Create New Connection

Connection Name: AWS-LAMP

Connection Type: SFTP

f. For **Host Name**, enter the IP address of your Ubuntu server, for **User Name** enter the **dev** user and corresponding password, and for **Initial Directory** enter **/var/www/html** (Apache web server main directory). Leave the default settings for the rest:



Manage Remote Connections

AWS-LAMP [SFTP]

Name: AWS-LAMP

Host Name: 18.234.204.171 Port: 22

User Name: dev

Password: •••••••••

Leave password empty to be prompted.
(or specify Private Key file)

Private Key File:

Running ssh-agent will be used.

Known Hosts File:

Initial Directory: /var/www/html

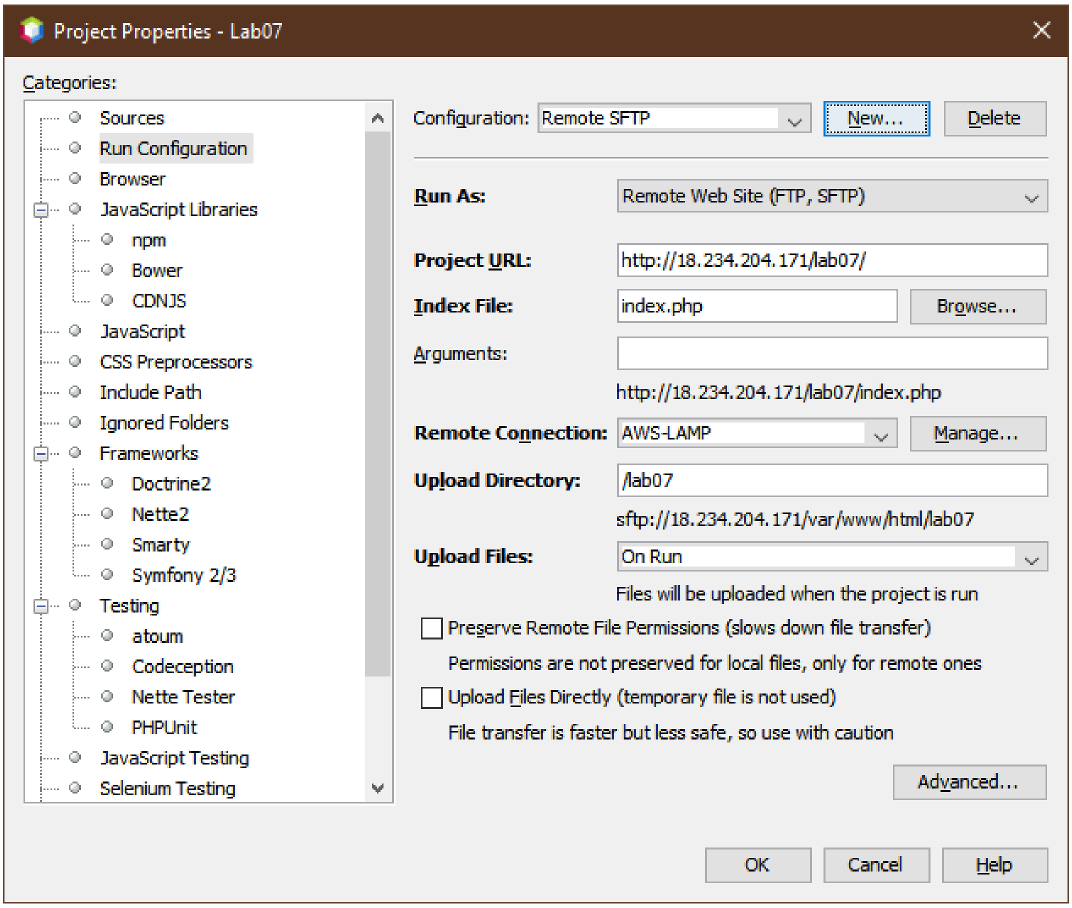
Timeout (s): 30

Keep-alive interval (s): 30 Interval 0 means disabled.

g. Click the "Test Connection" button to make sure the connection works (answer "Yes" to any authentication prompts).

h. Click "OK" to save the configuration and return to the Project Properties dialog.

i. Fill in the remaining settings as indicated below. Note that **Upload Directory** is optional, but it's a good idea to create a separate sub-directory for your PHP project in case you wish to deploy other projects to the server later.



Configuration: Remote SFTP

Run As: Remote Web Site (FTP, SFTP)

Project URL: http://18.234.204.171/lab07/

Index File: index.php

Arguments:

http://18.234.204.171/lab07/index.php

Remote Connection: AWS-LAMP

Upload Directory: /lab07

sftp://18.234.204.171/var/www/html/lab07

Upload Files: On Run

Files will be uploaded when the project is run

□ Preserve Remote File Permissions (slows down file transfer)

Permissions are not preserved for local files, only for remote ones

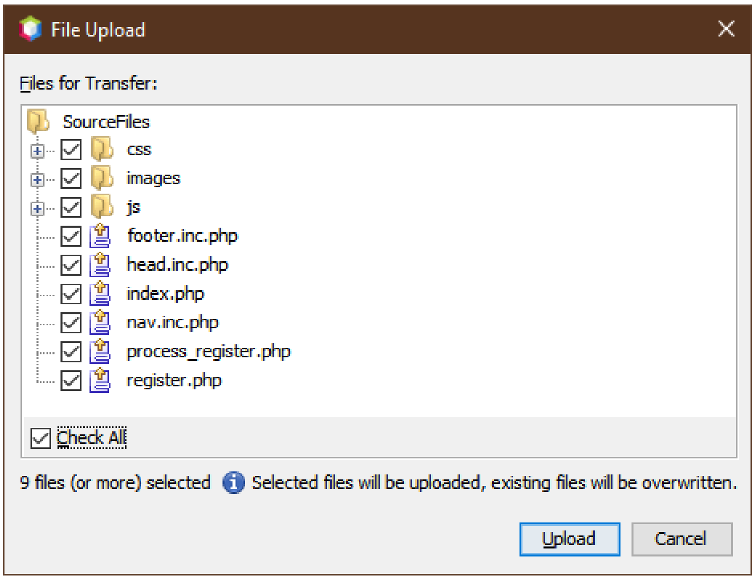
□ Upload Files Directly (temporary file is not used)

File transfer is faster but less safe, so use with caution

j. Click "OK" to save the run configuration settings.

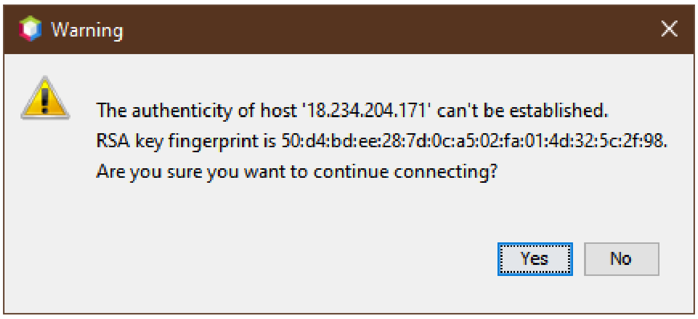
Tip: to avoid having to change the IP address every time you restart your AWS instance, you can configure the Elastic IP in AWS. But bear in mind that Elastic IPs will consume a small amount of credits whenever the instance is shut down. Refer to the following guide: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html.

9.4 From the NetBeans IDE, click the green "Run Project" button. In the File Upload dialog, check the "Check All" box to select all files, then click "Upload":



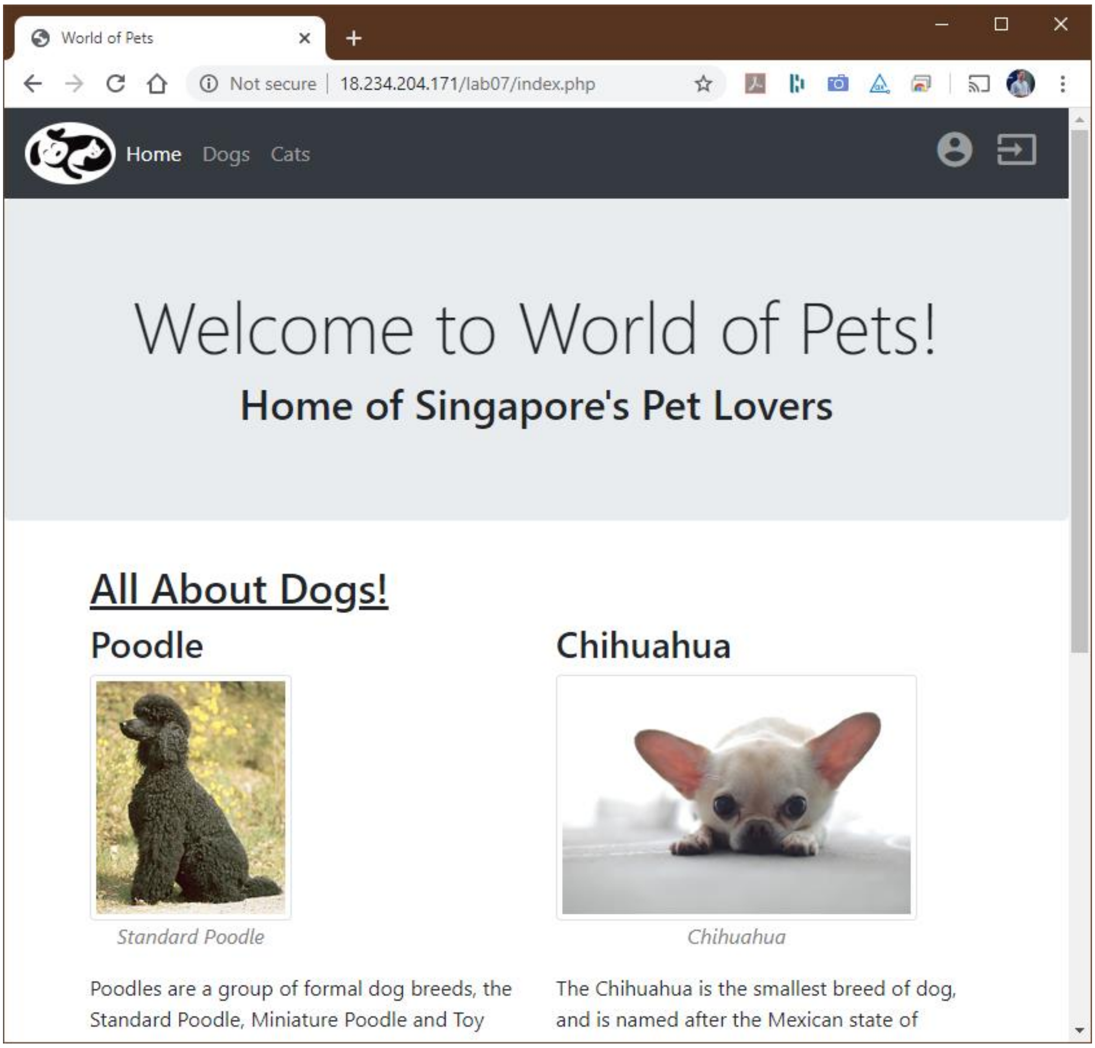
9 files (or more) selected Selected files will be uploaded, existing files will be overwritten.

Click 'Yes' on the authenticity warning dialog:



The authenticity of host '18.234.204.171' can't be established.
RSA key fingerprint is 50:d4:bd:ee:28:7d:0c:a5:02:fa:01:4d:32:5c:2f:98.
Are you sure you want to continue connecting?

9.5 The Output window will show the files being transferred to the server. When completed, you should see the World of Pets website running on the Ubuntu server:



Welcome to World of Pets!

Home of Singapore's Pet Lovers

All About Dogs!

Poodle

Standard Poodle

Poodles are a group of formal dog breeds, the Standard Poodle, Miniature Poodle and Toy

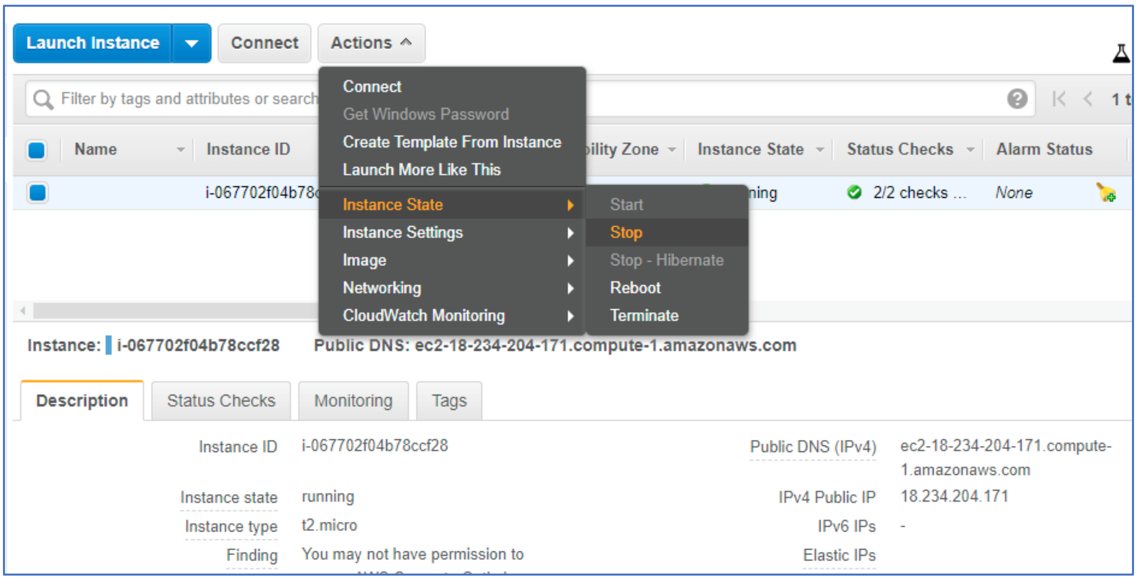
Chihuahua

Chihuahua

The Chihuahua is the smallest breed of dog, and is named after the Mexican state of

10. STOPPING THE AWS INSTANCE

10.1 It’s a good idea to shut down any instances if you’re not using them in order to avoid wasting your AWS credits. To do this, simply select the instance in the AWS console and from the “Actions” menu, choose “Instance State->Stop”:



Name	Instance ID	...ility Zone	Instance State	Status Checks	Alarm Status
	i-067702f04b78c...		...ning	2/2 checks ...	None

Instance: i-067702f04b78ccf28 Public DNS: ec2-18-234-204-171.compute-1.amazonaws.com

Instance ID	i-067702f04b78ccf28	Public DNS (IPv4)	ec2-18-234-204-171.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	18.234.204.171
Instance type	t2.micro	IPv6 IPs	-
Finding	You may not have permission to	Elastic IPs	

This is equivalent to shutting down the computer and will preserve all of your settings and files. When you're ready to use the instance again, select the "Start" menu option.

As mentioned, your IP address will change the next time you restart the instance, unless you've configured Elastic IP.

11. REFERENCES

11.1 For more details on installing LAMP stack, refer to:

a. How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 18.04 (https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04).

b. Install a LAMP Web Server with the Amazon Linux AMI (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-LAMP.html).

c. Deploying a PHP Application on a Remote Web Server Using the NetBeans IDE (https://netbeans.org/kb/docs/php/remote-hosting-and-ftp-account.html).