# Workshop 7 - Week 10 - CSY2085 – Server Administration and Security

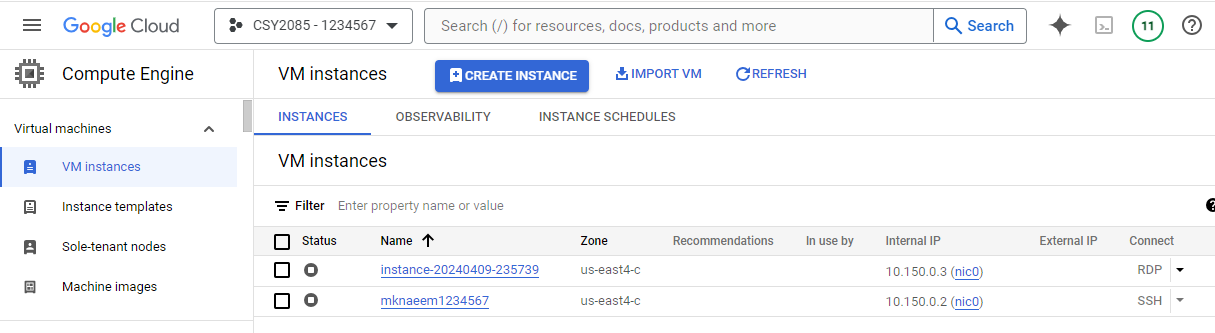
## Workshop: Linux DNS and Web Server

## Objective

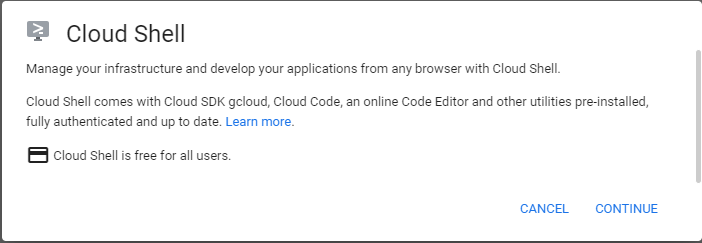
The main objective is to set up your Linux server a web server. For websites and domain names such as www.google.com and www.yahoo.com to work on the internet, the names need to be translated into IP addresses. This is the job of DNS or Domain Name Servers. Thus, you will be setting up your Ubuntu Linux server to act as a Domain Name Service (DNS) server and setting up a website with a specified domain name. You will also be learning how to manage the Linux server via the Webmin Web Control.

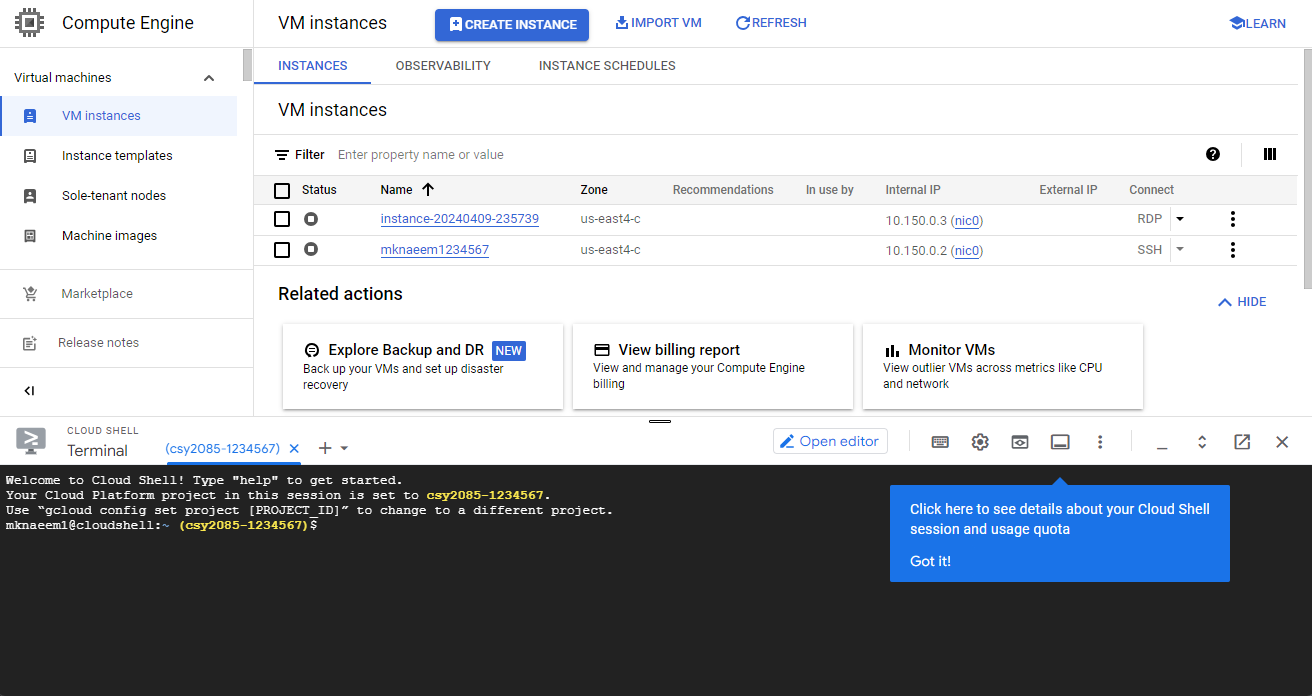
## Task 1 - Preparing the Ubuntu Linux Server

1. Login to your google cloud environment.
2. Locate and start the Ubuntu Linux VM that you created from the previous workshop.
3. First, we need to modify the firewall rules on the VM Instance to enable us to communicate with the server.
4. In your VM instance page, click on the Google Cloud Shell button at the top of the screen:



1. That will open the Google Cloud Shell at the bottom of the screen – Press Continue:





1. Type in the following commands to modify the firewall rules to allow the Webmin and DNS services:

**gcloud compute firewall-rules create webmin --allow tcp:10000**

**gcloud compute firewall-rules create dns --allow udp:53**

1. Type in the following commands to check if the firewall rules have been applied correctly

**gcloud compute firewall-rules list**

**[Capture your screen and paste it here]**

1. Open your VM in a separate window by clicking on the SSH button
2. Login to the Ubuntu Linux as root (**Important to logged in as a Root User**)

## Task 2 - Install Webmin on your server

1. Issue the following command in Terminal to import the Webmin repository key:

**wget http://www.webmin.com/jcameron-key.asc**

1. Next, issue the following command in Terminal to install the key:

**sudo apt-key add jcameron-key.asc**

1. You will add the Webmin repository in the */etc/apt/sources.list*file. In this way, you will be able to install Webmin via the APT. Use any text editor to edit the */etc/apt/sources.list* file:

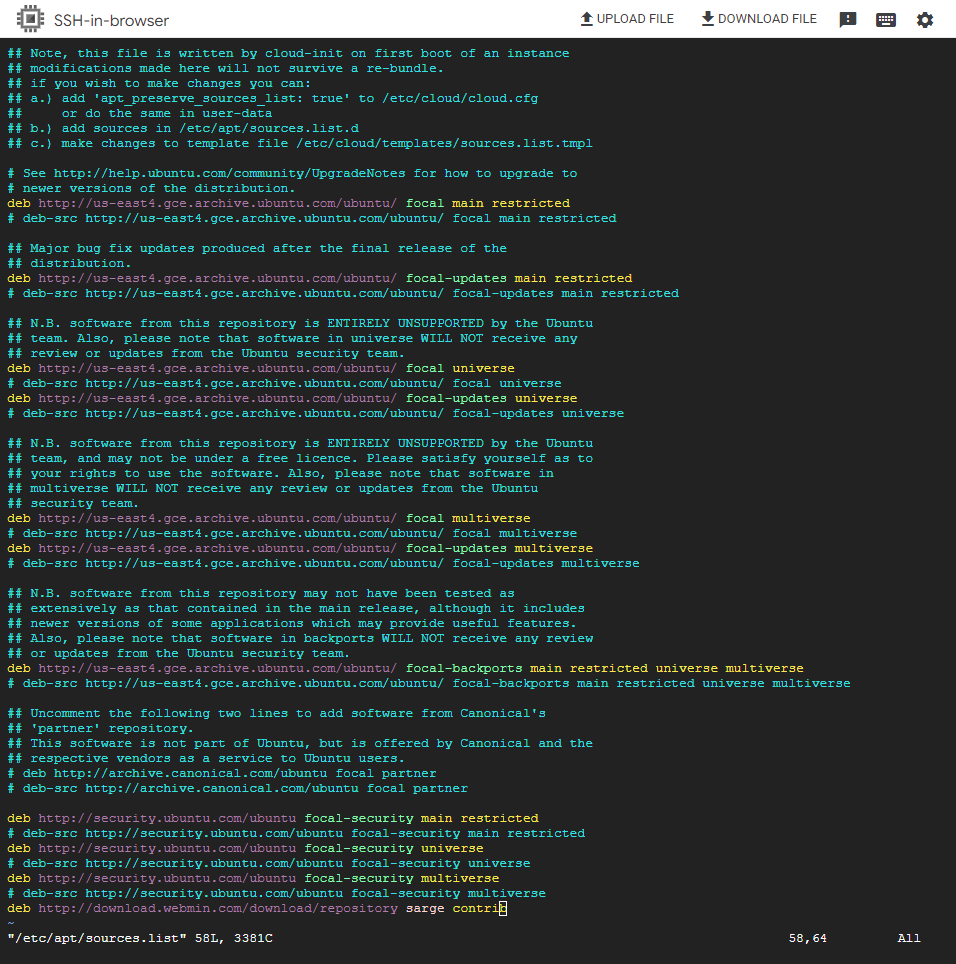
**vi /etc/apt/sources.list**

1. Then, add the Webmin repository to the */etc/apt/sources.list* file by adding the following line in the editor:

**deb http://download.webmin.com/download/repository sarge contrib**

Once in **vi editor**

* Press the"**i**" key to enter insert mode**.**
* Once you have finished writing, press the "**Esc**" key to exit insert mode and return to command mode.
* To save your changes and exit vi, type **:wq** and then press Enter. This command writes the changes to the file and quits vi.



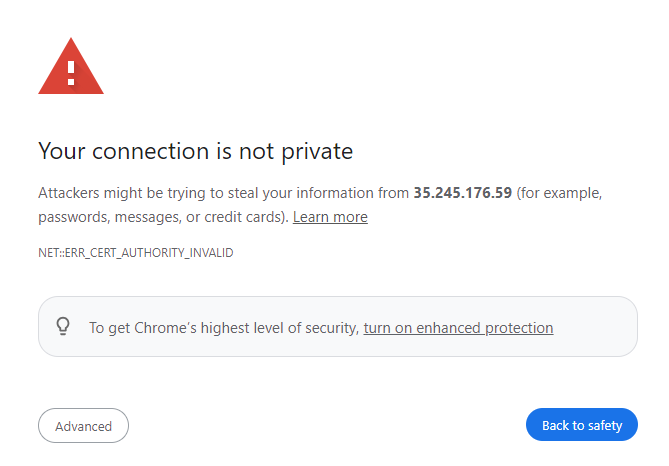
1. Next, issue the following command in Terminal to update the repository index with that of the newly added Webmin repository.

**apt update**

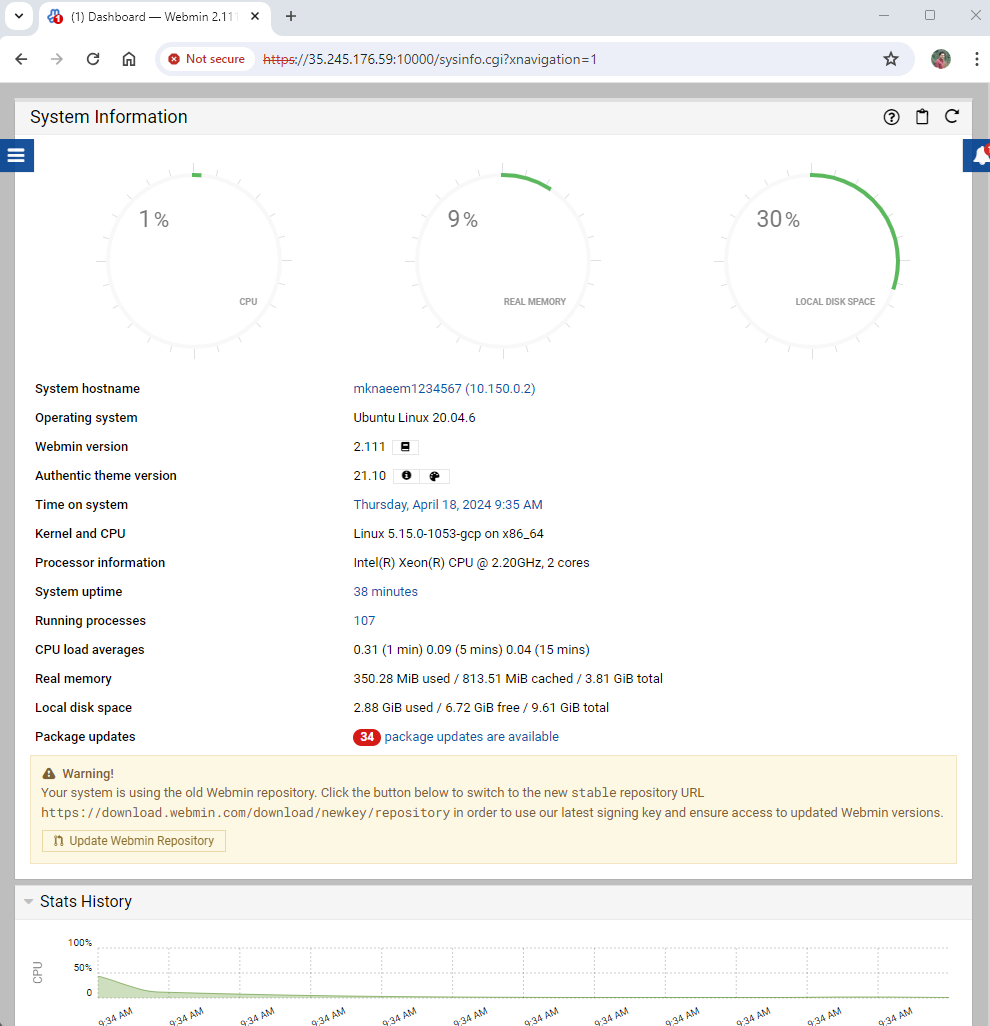
1. Install the Webmin package as sudo via the following command:

**apt install webmin**

1. The system might prompt you for confirmation by providing you with a **y/n** option. Hit **y** to continue the installation process.
2. Next, find out what your Linux Server Public IP address and record it below:  
     
   **[your Linux Public IP address]**
3. We will need to use another machine to act as a Windows "Internet client" to access your Linux server.
4. Start a web browser and then go to the URL: **https://35.245.176.59:10000** [Substitute **35.245.176.59** with your Linux server’s Public IP address.
5. The web browser may complain about the website's security certificate:



1. Click on “Advanced” and then click on" Proceed to 35.245.176.59 (unsafe)."
2. Login to Webmin as root and your root password.
3. You should see a page like the following:

****

**[Paste your screenshot here]**

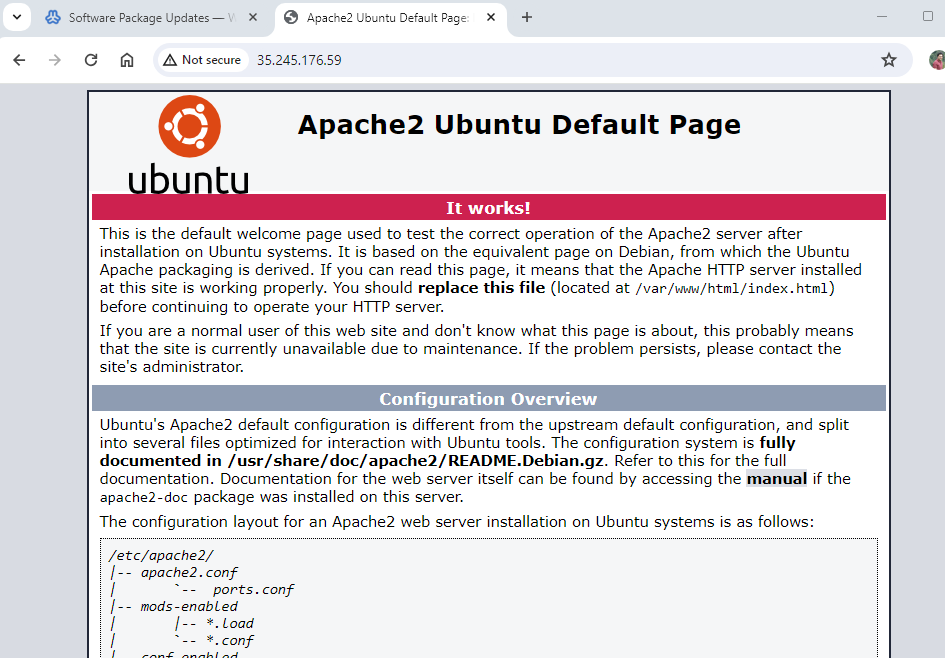
## Task 3 - Install Linux, Apache, MySQL, PHP (LAMP) stack on your server

## Install Apache

The Apache web server is currently the most popular web server in the world, which makes it a great default choice for hosting a website.

**apt install apache2**

1. Verify if apache was installed as planned by visiting your server's IP address in your Windows (Client Machine/VM) web browser and <http://your_server_IP_address/> -Forexample **http://35.245.176.59**
2. You will see the default Apache web page, which is there for informational and testing purposes. It should look something like this:



**Paste your screenshot here:**

## Install MySQL

Now that you have a web server up and running, you need to install the database system to be able to store and manage data for your site. MySQL is a popular database management system used within PHP environments.

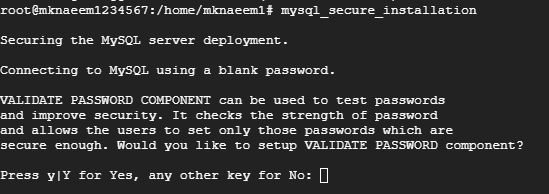
1. Again, use apt to acquire and install this software:

**apt install mysql-server**

1. When prompted, confirm installation by typing Y, and then ENTER.
2. Now that our MySQL database is running, we want to run a simple security script that will remove some dangerous defaults and lock down access to our database system a little bit. Start the interactive script by running:

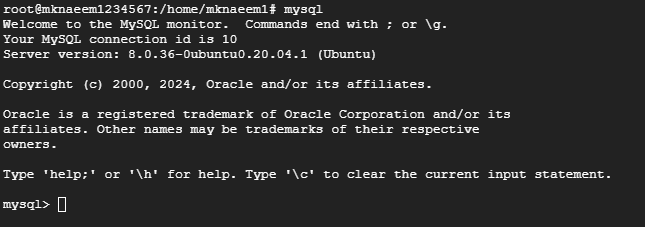
**mysql\_secure\_installation**

The prompt will ask you for your current root password. Since you just installed MySQL, you most likely won’t have one, so leave it blank by pressing enter. Then the prompt will ask you if you want to set a root password. Go ahead and enter Y, and follow the instructions:



1. Choose low password policy and set password as ‘**password**’
2. For the rest of the questions, press yes. When you’re finished, test if you’re able to log in to the MySQL console by typing:

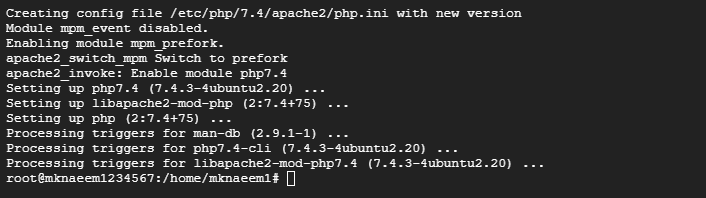
**mysql**

****

## Install PHP

PHP is the component of our setup that will process code to display dynamic content. It can run scripts, connect to our MySQL databases to get information, and hand the processed content over to our web server to display.

**apt install php libapache2-mod-php php-mysql**

****

## Install BIND on your server.

BIND is open-source software that implements the Domain Name System (DNS) protocols for the Internet which provides ability to perform name to Ip conversion. The name BIND stands for “Berkeley Internet Name Domain”, because the software originated in the early 1980s at the University of California at Berkeley. It is a reference implementation of DNS protocols, but it is also production-grade software, suitable for use in high-volume and high-reliability applications.

**apt-get install bind9 bind9utils bind9-doc**

Type the following command to see the current status of BIND server:

**service bind9 status**

**[Capture your screen and paste it here]**

**Note:** pressing "**q**" will return you to the command prompt.

Note: To download and install updates please type this command

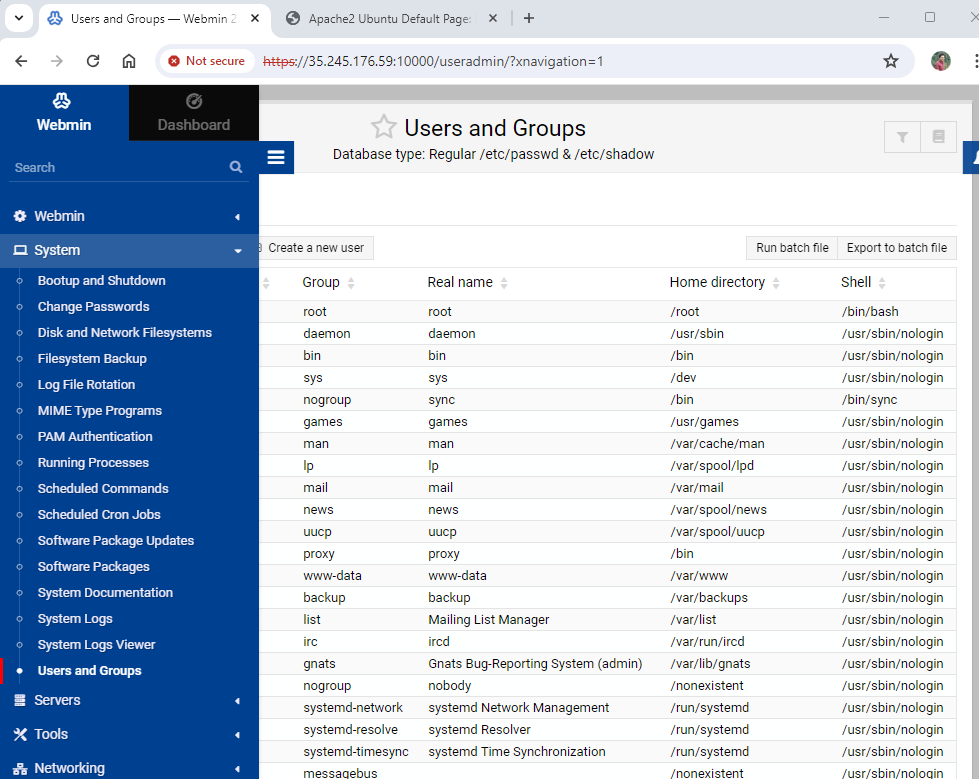
**apt update -y && apt upgrade -y**

## Task 4- Preparing the Windows Client

1. Whilst keeping the Linux server running, on the client Windows machine or VM, start a web browser and then go to the URL: **https://**[**35.204.55.123**](https://35.204.55.123/)**:10000** [Substitute **35.204.55.123** with your Linux server’s Public IP address.

**Remember to check that your server public IP address is still the same. You might be assigned a different IP address every time your server restarts.**

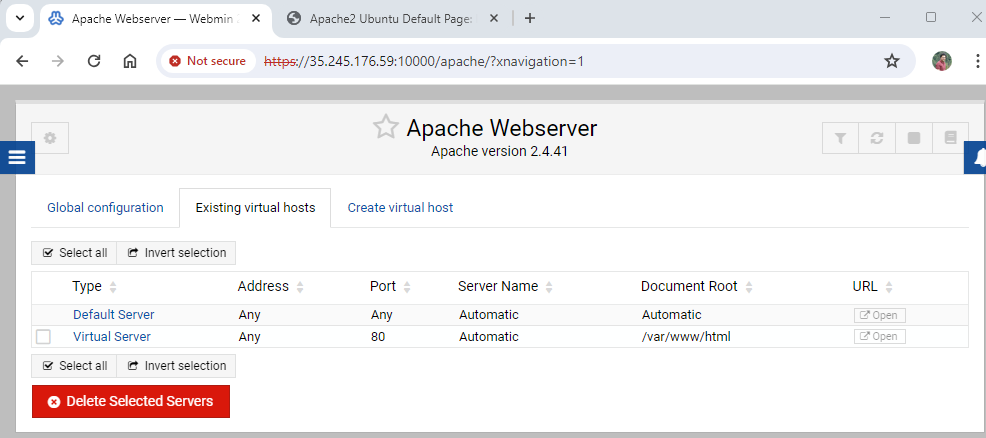
1. The web browser may complain about the website's security certificate. Click on "Continue to this website (not recommended)."
2. Login to Webmin as root and you root password.
3. Once you are logged in Webmin, click **System**, then **Users** **and** **Groups**, and **Create a new user**.



1. Create a new user called "**testuser**" with the password of "**password**".
2. Fill in the Username and Real name and click on the "Normal password" selection and fill in a password.
3. Scroll to the bottom of the page and click "Create".
4. When it is done, Webmin will show you the list of users including the new one that you have just created. Scroll towards the bottom of this page until it shows the new "testuser" details and capture the screen and paste it below:  
     
   **[paste your screenshot here]**

## Task 5 - Setting up the Apache Web Server

1. On the left-hand side of Webmin, click on Servers. Apache, MySQL and DNS servers should figure on the list. If not click on “Refresh Modules” at the bottom of the list to refresh these modules. Then reload webmin. Click on Apache Webserver. You will see the following screen:



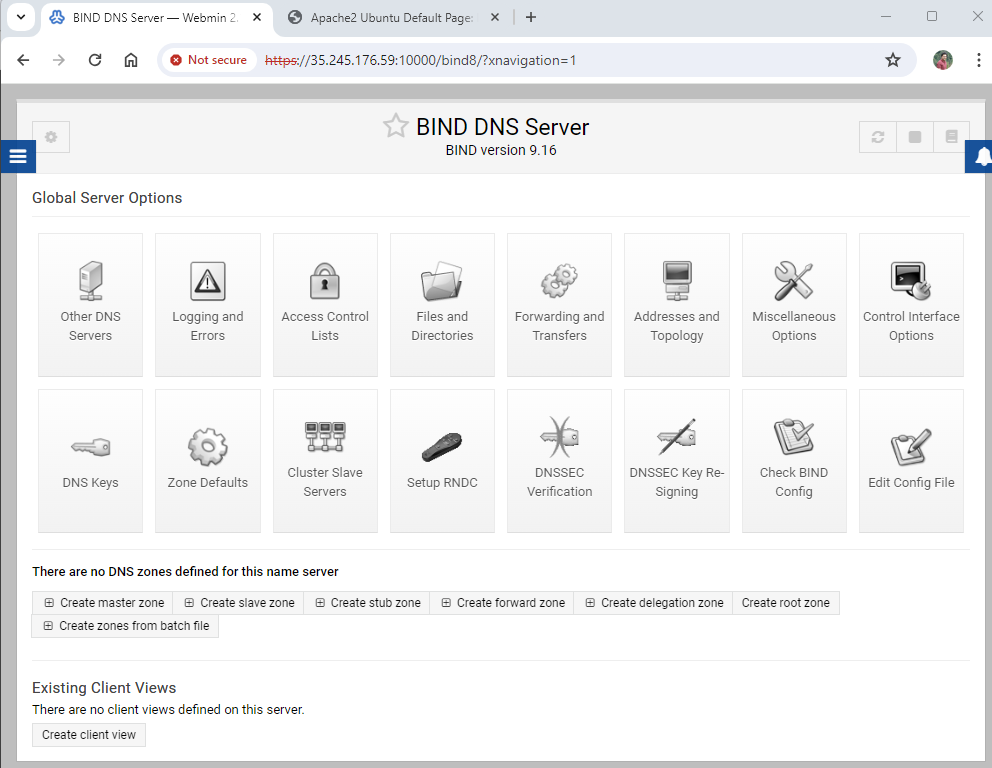
**[paste your screenshot here]**

## Task 6 - Setting up the Domain Name Service on the Linux Server

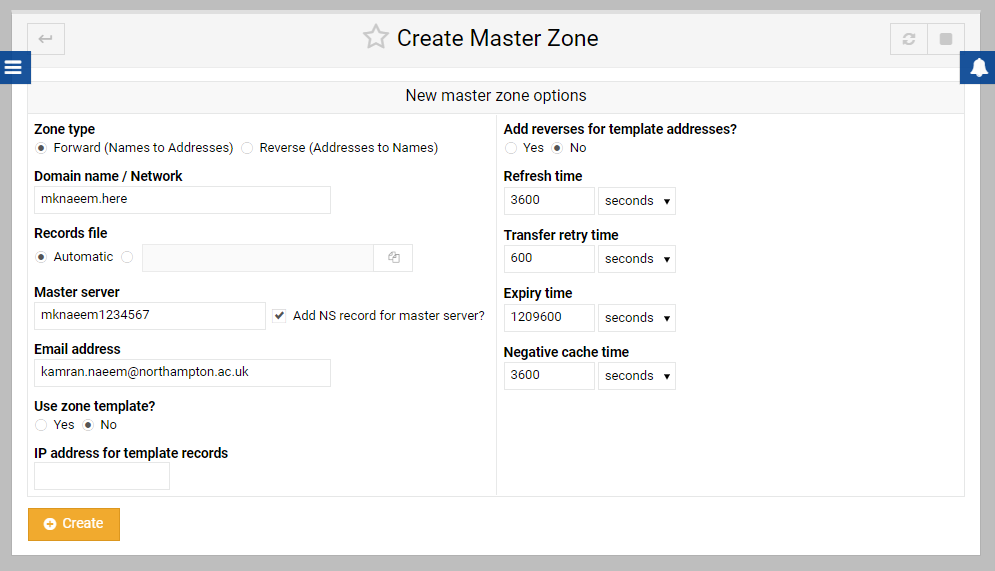
The Domain Name Service is the mechanism that allows computers and servers on the internet to have names. When you enter a name like <http://www.northampton.ac.uk/> into a web browser, the web browser must translate that name into an IP address in order to contact the correct machine. If the web browser PC does not have the IP address for <http://www.northampton.ac.uk/>, it would ask its network's DNS server(s) and it would be the DNS servers' job to translate the name into an IP address.

The DNS software that we will be using is the most popular DNS system on Linux and UNIX servers on the internet. It is called BIND (Berkeley Internet Name Domain).

1. On the left-hand side of Webmin, and click Servers and then BIND DNS Server and you should see the following screen:

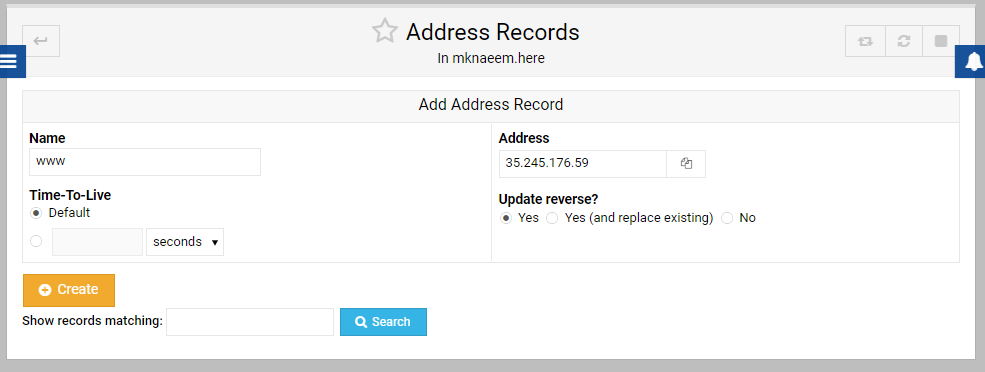


1. Click on “**Setup RNDC”** and then Click on “**Yes, Setup RNDC**”.
2. Go back to the BIND server main page and click on "**Create master zone**".
3. Then create the domain name **yourname.here**. Obviously, replace "yourname" with your own name: For example "mknaeem.here". **Do not use any spaces** or any other symbols in the name. Use just the letters a to z.

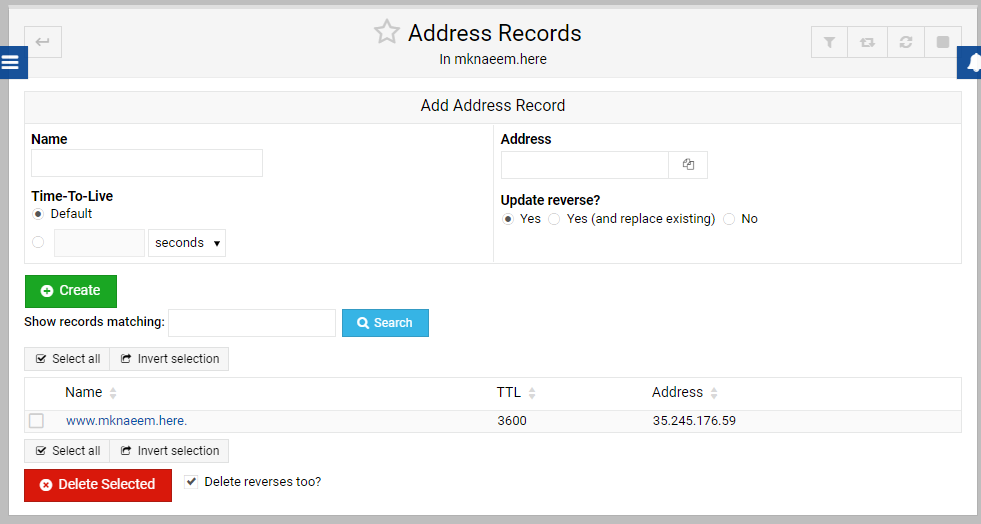
  
  
Select "**No**" for "**Add reverses for template addresses**", and then your email address, and then click "**Create**".

1. On the "Edit Master Zone" screen, click "**Address(0)**"
2. Fill in "**www**" for name and **your Linux server's IP address** for "Address":

**Remember to check that your server public IP address is still the same. You might be assigned a different IP address every time your server restarts.**



1. And then click Create.
2. You should then see a screen similar to the following:



1. Capture your version of the above screen and paste it below:  
     
   **[paste your screenshot here]**
2. Click on **Apply configuration** and **Apply Zone** in the top right-hand corner. Then stop and start bind again (top right-hand corner).
3. Stop apache and restart it again.
4. Now on the client Machine or VM, click the Windows "Start" button and then select "Control Panel"
5. Then select "Network and Sharing Centre" and Click on your network connections. In this case is WiFi:

A screenshot of a computer

Description automatically generated

1. Then right click on the "WiFi (VM1231814) – (Please Note that the name of WiFi will be different in your case)" and select "Properties"  
     
   A screenshot of a computer

   Description automatically generated
2. Then select "Internet Protocol Version 4 (TCP/IPv4)" and click "Properties".

A screenshot of a computer

Description automatically generated

1. Next, configure the network settings so that the Preferred DNS server is set with the IP address of your Linux server and click OK:

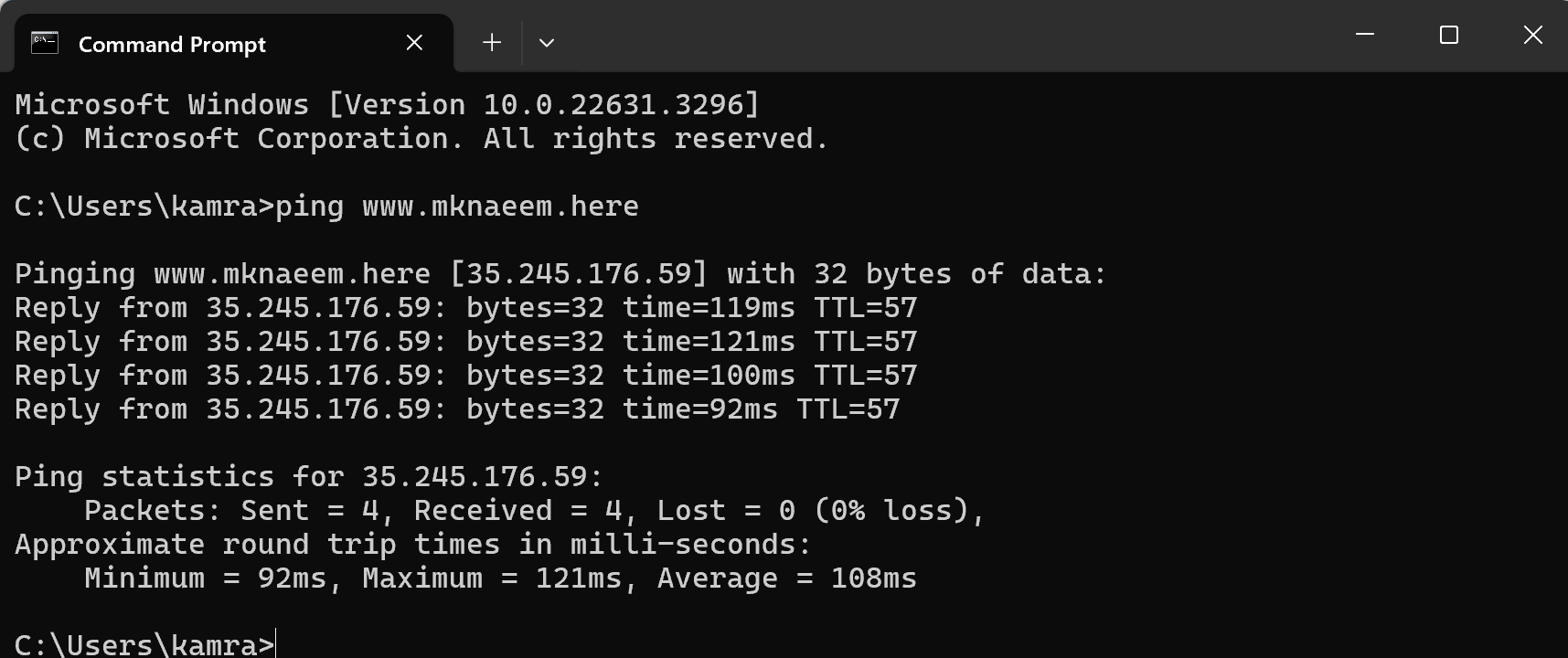
**Remember to check that your server public IP address is still the same. You might be assigned a different IP address every time your server restarts.**

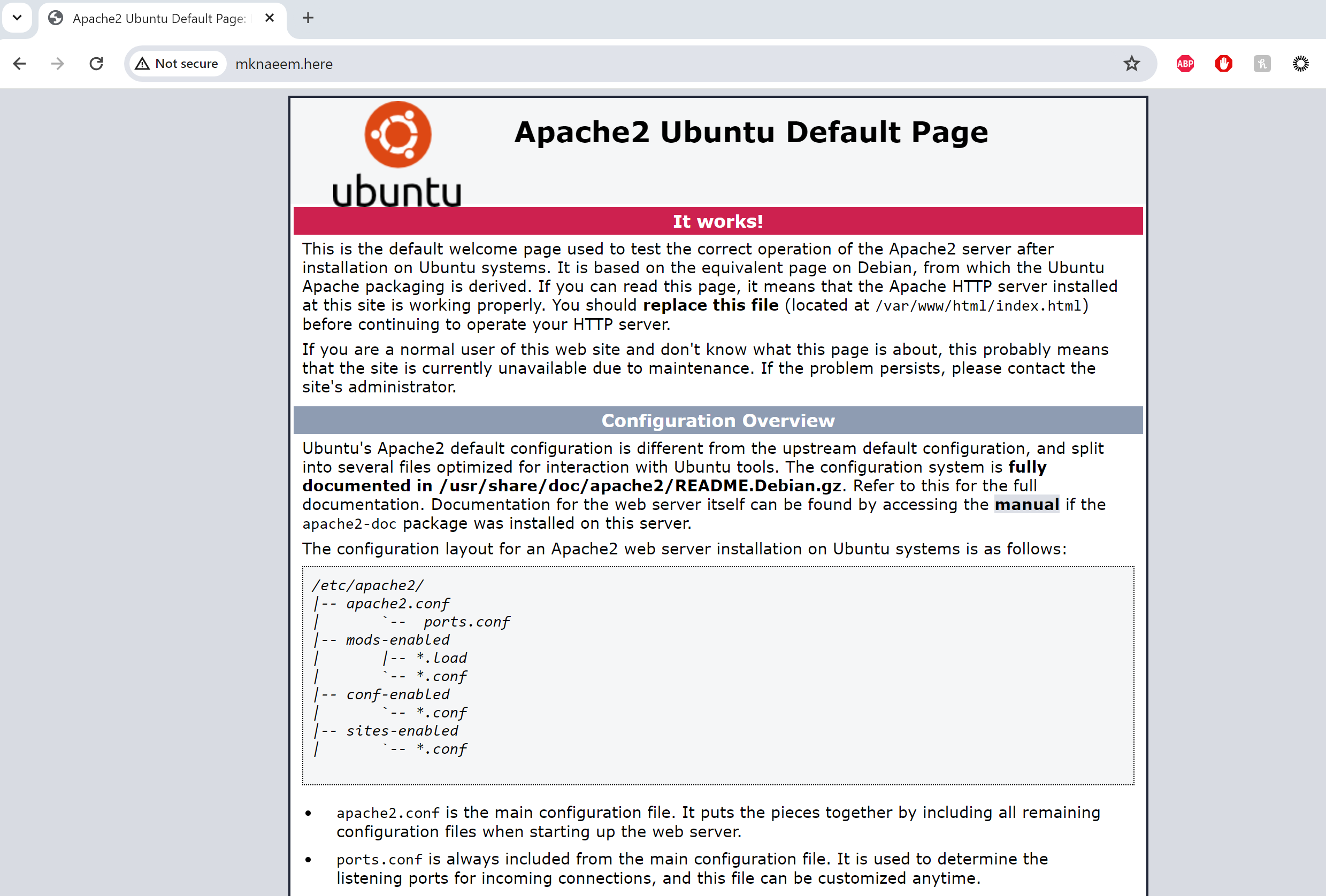
1. A screenshot of a computer

   Description automatically generated
2. Still in the **Client Machine or VM (Windows)** navigate to the following location C:\Windows\System32\drivers\etc. Locate the file called **“hosts”.**
3. **Before editing this file, modify its security settings to give full access to “Users”**
4. Open the file for edit using notepad. At the bottom of the file add the following information about your server to something like

A computer error message

Description automatically generated

1. Make sure you replace the address with your server address and the domain name with your own domain name.
2. Now check that the new DNS settings is working correctly. Start a Command Prompt on the client and ping your new domain name:  
     
   **ping www.mknaeem.here**  
   Obviously substituting your own domain for the above.
3. You should have a screen similar to the following:  
   
4. Capture your version of the screen and paste it below:  
     
   **[paste your screenshot here]**
5. Open a web browser and visit your web server on [www.yourname.here](http://www.yourname.here) and paste the screenshot below:



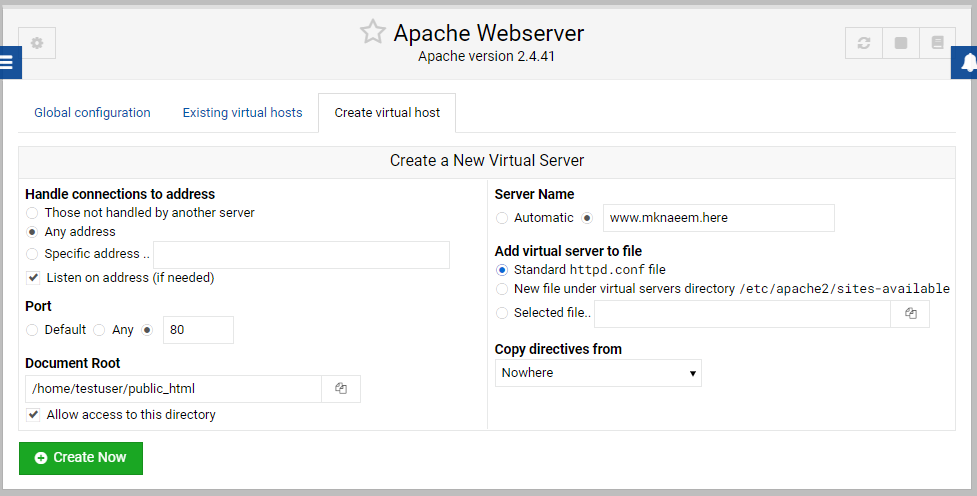
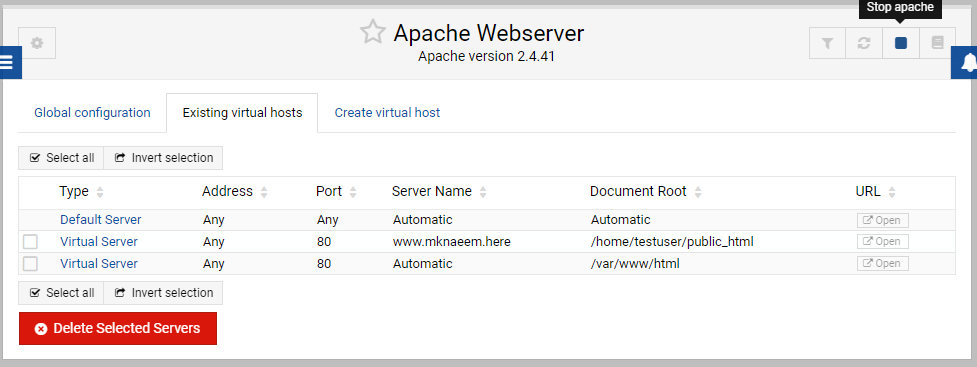
## Task 7 - Setting up a website for the new domain name on the Linux Server

1. First you need to enable the use of User Directory on apache. Open a shell and type the following command.

**a2enmod userdir**

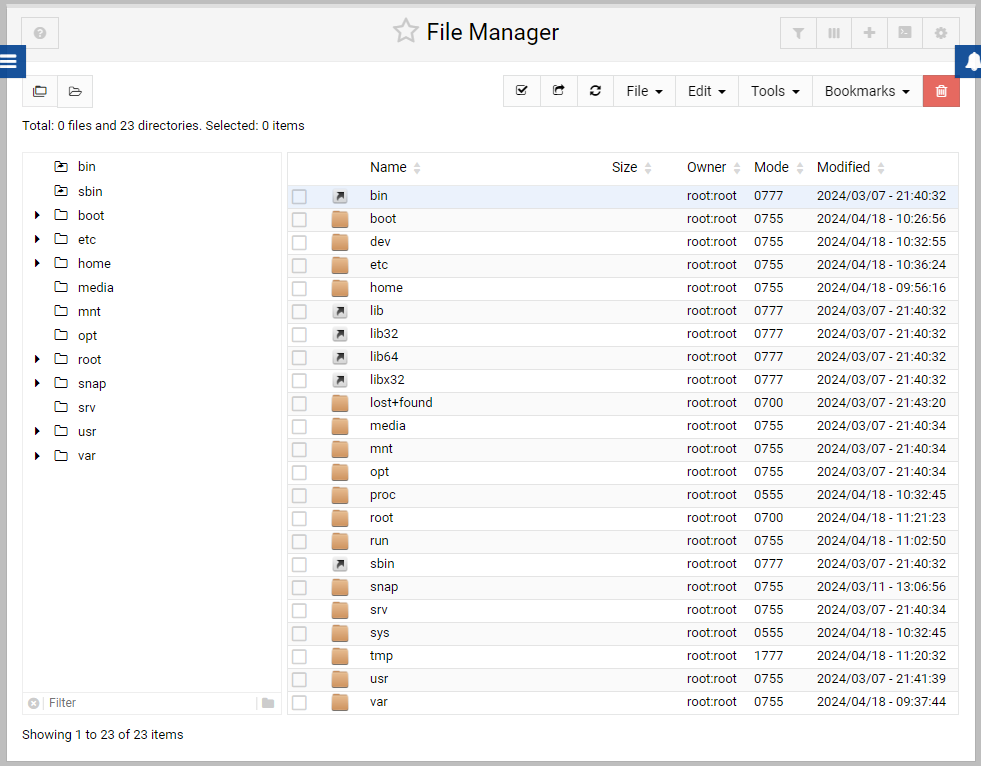
1. It will prompt you to restart apache using the following:

**systemctl restart apache2**

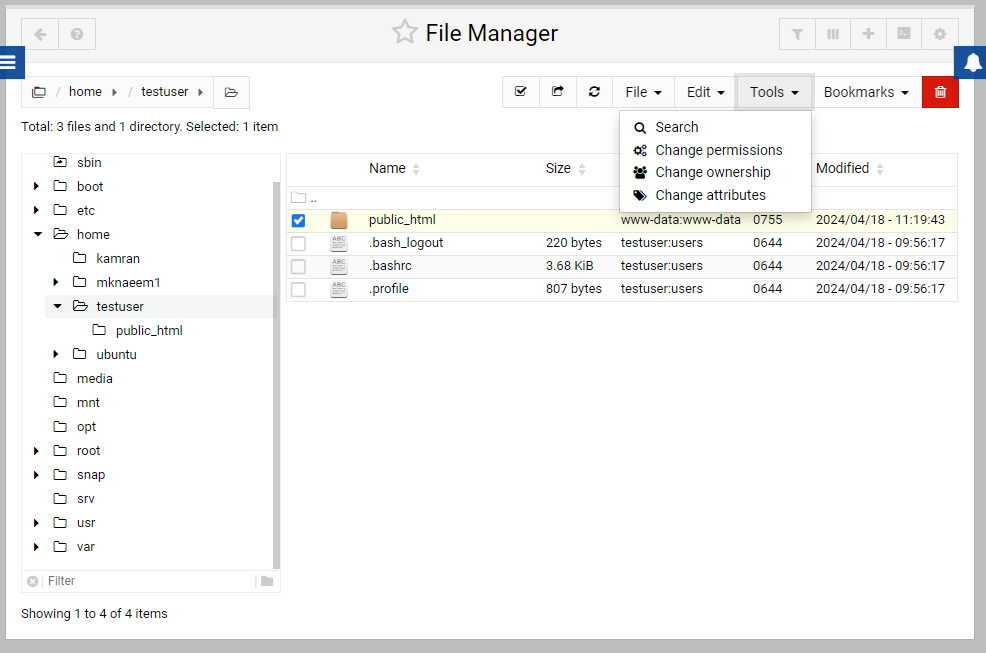
1. In Webmin, and click Servers and then Apache Webserver:
2. Click on Create virtual host:  
     
   
3. Select and fill in Port **80** for Port.
4. Fill in **"/home/testuser/public\_html"** for Document Root.
5. For Server Name, fill in "**www.yourname.here**" (using your own domain name that you have just created).
6. Click "**Create Now**",
7. Stop and Restart Apache  
     
   
8. Now capture your version of the above screen and paste it below:  
     
   [paste your screenshot here]

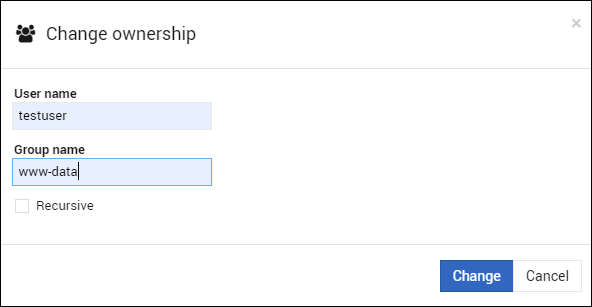
## Task 8 - Manipulating files on the Linux Server via Webmin

1. In Webmin, click Tools and then File Manager.

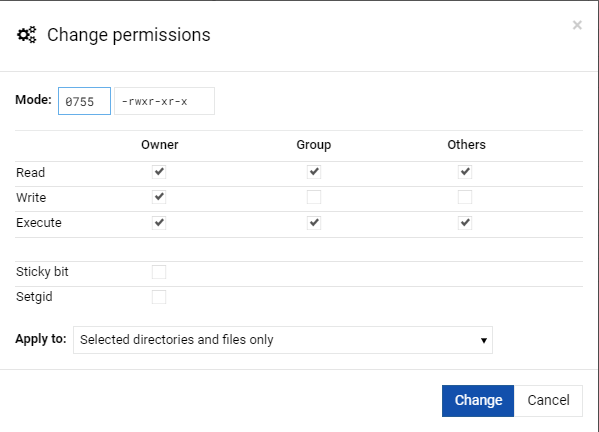


1. Navigate the File Manager by double-clicking the folders, to the directory "/home/testuser".
2. Select the directory "public\_html" click on “Tools then Change ownership”:

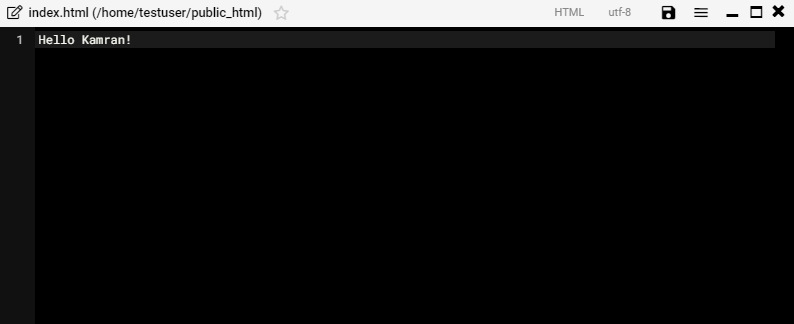


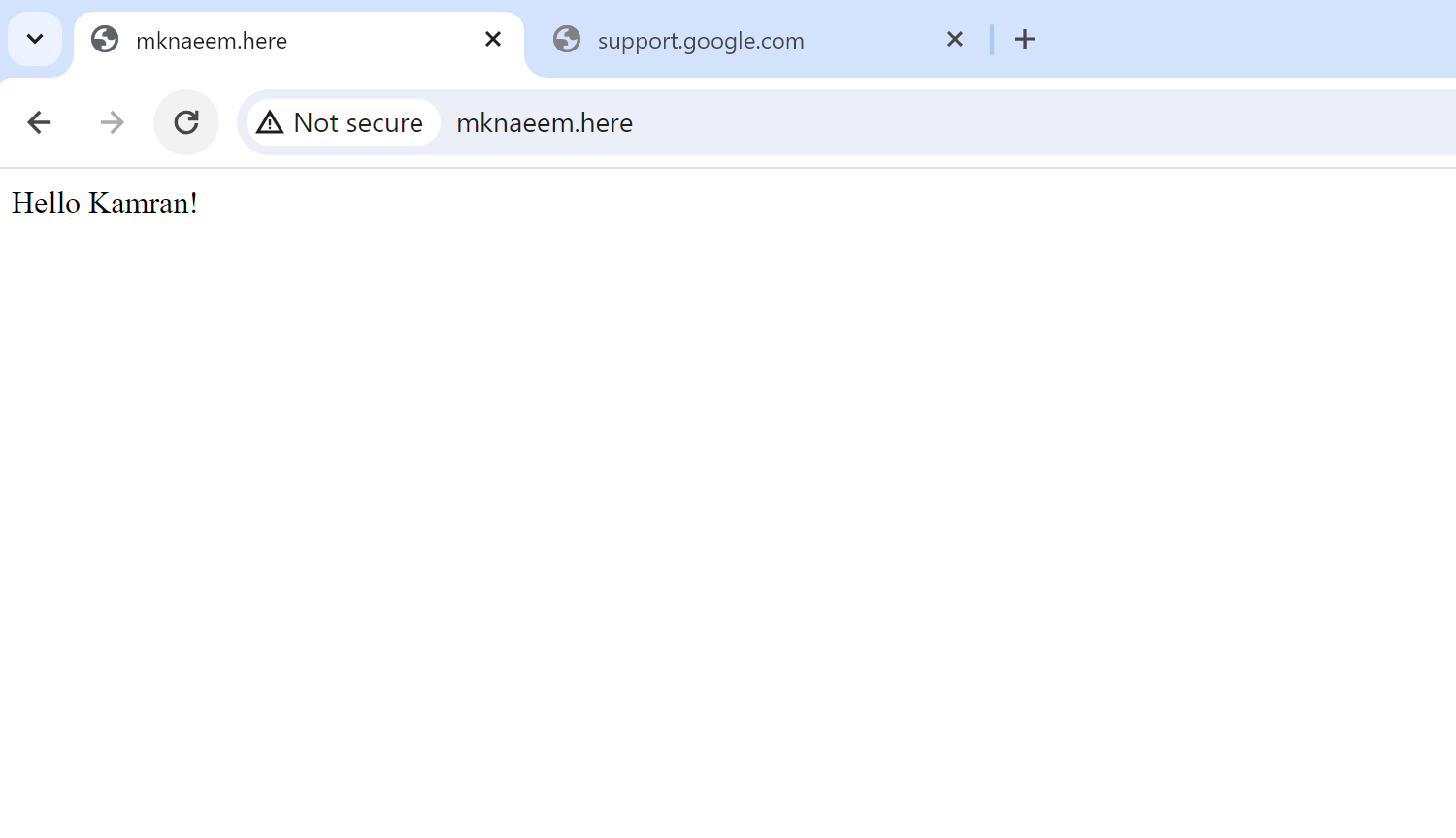
  
  
Make sure the User is set to **testuser** and group to **www-data**

1. Go back and select the folder again. Click “tools” and “Change permissions. Change permissions to 0755



1. Double-click into "public\_html":  
     
   Click on the "File” and create new file. Name the file **index.html**  
     
   Right click the index.html file and select Edit
2. "Hello yourstudentnumber" or similar text into the editor box
3. Click "Save and Close".



1. Select the index.html file that you have just created change the permissions to 0755.  
     
   Now verify that your new website is working. Start a new Web Browser and go to <http://www.mknaeem.here/> (using your own domain name of course!):  
     
   
2. Capture your version of the screen above and paste it below:  
     
   **[paste your screenshot here]**

## Task 9 - Creating another new domain and website

Based on what you have learnt from the above, now create another new domain name by putting the word "new" in front of your own name, for example newyourname.here and new web site, and capture the following specified screenshots below. Make sure that you stop and restart both the BIND DNS server and the Apache server after you have created the new settings just to make sure that the new settings are valid and current. On the Windows (Client Machine/VM), close and restart your web browser before attempting to connect to your new website:

**Screenshot of the BIND DNS Address Record of www.yournewdomainname.here.**

**Screenshot of the pings to the new domain name in the command prompt.**

**Screenshot of the "virtual host" list of websites and names in the Apache Server page to include the new website.**

**Screenshot of the finished and running website.**

## Task 10 - Answer the following questions by researching on the web

1. List 3 web-based Linux administration software below (not including Webmin):  
   1. [your answer here]
   2. [your answer here]
   3. [your answer here]
2. What do the acronyms LAMP and WAMP stand for, in the context of web servers?  
     
   [your answer here]

**Now shut down your server and stop your VM instance to stop incurring charges. This is very important!!!**

This is the end of this workshop. If you have finished the workshop, please save your work.