

LAMP STACK SERVER BUILD QUICK GUIDE

Linux Debian 9 (Stretch), **A**pache2, **M**ySQL 8, **P**HP 7

Written by Eran Ben-Shahar

Last update: 17/Jan/2019

Part 1: LAMP Stack Installation

1. Web server setup from blank

The following step by step tutorial will take you through setting up a hosting server and installation of the websites, from scratch.

1.2.1 Install a linux Debian operating system

Debian is the most stable and side spread linux operating system. You can download a debian image disk from the internet (make sure it is a reliable source like <https://www.debian.org/CD/http-ftp/>) and install it on any server. Most hosting companies of dedicated or cloud servers would install the operating system for you and will connect the server to the internet so you could manage it with SSH. Don't worry which version is installed - as long as you got the basic Debian working, we will do the rest through this guide.

1.2.2 Get the server's public IP numbers

If the server has more than one IP#, make sure you know which one is the main one and which one is added to it. You would need that later on.

1.2.3 Confirm the correctness of *apt-get* repositories

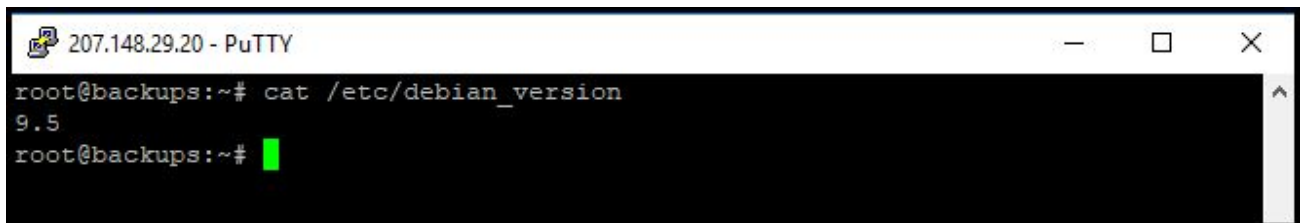
apt-get is an application that downloads installation packages from the internet and installing it on the server.

The file `/etc/apt/sources.list` is defining from which internet repositories to download the installation packs.

Make sure that it includes the following entries which match with the recent stable debian version.

To find your current installed debian version write the command:

```
cat /etc/debian_version
```



```
207.148.29.20 - PuTTY
root@backups:~# cat /etc/debian_version
9.5
root@backups:~#
```

The format of the `/etc/apt/sources.list` is given at <https://wiki.debian.org/SourcesList> :


```
deb http://site.example.com/debian distribution component1 component2 component3
```

```
deb-src http://site.example.com/debian distribution component1 component2 component3
```

Open the `/etc/apt/sources.list` file to update it:

```
nano /etc/apt/sources.list
```

For the recent debian 9 “stretch” version, you should enter the following sources:

```
deb http://deb.debian.org/debian stretch main contrib non-free
```

```
deb-src http://deb.debian.org/debian stretch main contrib non-free
```

```
deb http://deb.debian.org/debian-security/ stretch/updates main contrib non-free
```

```
deb-src http://deb.debian.org/debian-security/ stretch/updates main contrib non-free
```

```
deb http://deb.debian.org/debian stretch-updates main contrib non-free
```

```
deb-src http://deb.debian.org/debian stretch-updates main contrib non-free
```

The real life looks like the following, note that – (1) you need to comment all old sources, like - the CD ROM sources been used to install the base operating system are commented, they are there from the initial installation, (2) if you are upgrading the operating system from previous versions, say from “jessie” to “stretch” - you should comment all the “jessie” repositories. (3) **you should check carefully before you enter any other repositories to here as it must be official debian source, otherwise you may get hacked!**


```
debian@debian: ~
GNU nano 2.2.6      File: /etc/apt/sources.list      Modified
#deb cdrom:[Debian GNU/Linux 8.11.0 _Jessie_ - Official amd64 NETINST Binary-1 20180623-$
#deb cdrom:[Debian GNU/Linux 8.11.0 _Jessie_ - Official amd64 NETINST Binary-1 20180623-$

#deb http://mirror.it.ubc.ca/debian/ jessie main
#deb-src http://mirror.it.ubc.ca/debian/ jessie main

#deb http://security.debian.org/ jessie/updates main
#deb-src http://security.debian.org/ jessie/updates main

# jessie-updates, previously known as 'volatile'
#deb http://mirror.it.ubc.ca/debian/ jessie-updates main
#deb-src http://mirror.it.ubc.ca/debian/ jessie-updates main

deb http://deb.debian.org/debian stretch main contrib non-free
deb-src http://deb.debian.org/debian stretch main contrib non-free

deb http://deb.debian.org/debian-security/ stretch/updates main contrib non-free
deb-src http://deb.debian.org/debian-security/ stretch/updates main contrib non-free

deb http://deb.debian.org/debian stretch-updates main contrib non-free
deb-src http://deb.debian.org/debian stretch-updates main contrib non-free

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Exit the editor and save the file with the new resources list.

1.2.4 Update server repositories

This will download the recent repositories:

```
apt-get update
```



```
207.148.29.20 - PuTTY
1764 B]
Get:12 http://deb.debian.org/debian-security stretch/updates/contrib Translation-en [
1759 B]
Get:13 http://deb.debian.org/debian-security stretch/updates/non-free amd64 Packages
[1600 B]
Get:14 http://deb.debian.org/debian-security stretch/updates/non-free Translation-en
[691 B]
Get:15 http://deb.debian.org/debian stretch-updates/main Sources [3748 B]
Get:16 http://deb.debian.org/debian stretch-updates/main amd64 Packages [5152 B]
Get:17 http://deb.debian.org/debian stretch-updates/main Translation-en [4512 B]
Get:18 http://deb.debian.org/debian stretch/main Sources [6751 kB]
Get:19 http://deb.debian.org/debian stretch/contrib Sources [44.7 kB]
Get:20 http://deb.debian.org/debian stretch/non-free Sources [79.5 kB]
Get:21 http://deb.debian.org/debian stretch/main amd64 Packages [7089 kB]
Get:22 http://deb.debian.org/debian stretch/main Translation-en [5388 kB]
Get:23 http://deb.debian.org/debian stretch/contrib amd64 Packages [50.9 kB]
Get:24 http://deb.debian.org/debian stretch/contrib Translation-en [45.9 kB]
Get:25 http://deb.debian.org/debian stretch/non-free amd64 Packages [78.6 kB]
Get:26 http://deb.debian.org/debian stretch/non-free Translation-en [80.4 kB]
Fetched 20.8 MB in 7s (2803 kB/s)
Reading package lists... Done
root@backups:~#
```

1.2.5 Upgrade the system

This will upgrade the system with the latest releases:

```
apt-get upgrade
```

Click “Y” when prompted with the upgrade question, and wait for the system to perform all updates:

```
207.148.29.20 - PuTTY
Get:22 http://deb.debian.org/debian stretch/main Translation-en [5388 kB]
Get:23 http://deb.debian.org/debian stretch/contrib amd64 Packages [50.9 kB]
Get:24 http://deb.debian.org/debian stretch/contrib Translation-en [45.9 kB]
Get:25 http://deb.debian.org/debian stretch/non-free amd64 Packages [78.6 kB]
Get:26 http://deb.debian.org/debian stretch/non-free Translation-en [80.4 kB]
Fetched 20.8 MB in 7s (2803 kB/s)
Reading package lists... Done
root@backups:~# apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
 base-files curl gnupg gnupg-agent gpgv grub-common grub-pc grub-pc-bin grub2-common
 hdparm libcurl3 libcurl3-gnutls libfuse2 libgnutls30 libpam-systemd libperl5.24
 libseccomp2 libssl1.1 libsystemd0 libudev1 libx11-6 libx11-data libxapian30
 linux-image-4.9.0-8-amd64 openssl perl perl-base perl-modules-5.24 systemd
 systemd-sysv tzdata udev
32 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 63.6 MB of archives.
After this operation, 119 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```


1.2.6 Confirm the system update

To confirm the system update, run **apt-get update** once again and then **apt-get upgrade**

```
207.148.29.20 - PuTTY
root@backups:~# apt-get update
Ign:1 http://deb.debian.org/debian stretch InRelease
Hit:2 http://deb.debian.org/debian-security stretch/updates InRelease
Hit:3 http://deb.debian.org/debian stretch-updates InRelease
Hit:4 http://deb.debian.org/debian stretch Release
Reading package lists... Done
root@backups:~# apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@backups:~#
```

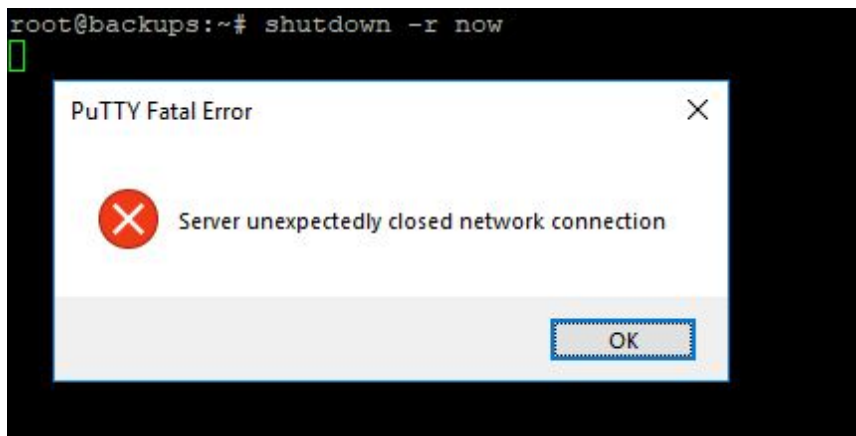
You can also **cat /etc/debian_version** to make sure you are on the new version.

```
debian@debian: ~
root@debian:/home/debian# cat /etc/debian_version
9.6
root@debian:/home/debian#
```

1.2.7 Restart the server

Restart the server (this action is not compulsory but it is not a bad idea) and login to it after the restart is finished:

```
shutdown -r now
```

1.2.8 Install sudo application

sudo = “Super User DO” will allow you to manage the server in a secured way, i.e. without using the root account.

apt-get install sudo

This will install sudo on your machine. Response screen should be similar to:

```
207.148.29.20 - PuTTY

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 17 06:32:36 2018 from 203.219.108.190
root@backups:~# apt-get install sudo
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  sudo
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 1055 kB of archives.
After this operation, 3108 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian stretch/main amd64 sudo amd64 1.8.19p1-2.1 [1
055 kB]
Fetched 1055 kB in 0s (5148 kB/s)
Selecting previously unselected package sudo.
(Reading database ... 33469 files and directories currently installed.)
Preparing to unpack .../sudo_1.8.19p1-2.1_amd64.deb ...
Unpacking sudo (1.8.19p1-2.1) ...
Setting up sudo (1.8.19p1-2.1) ...
Processing triggers for systemd (232-25+deb9u6) ...
Processing triggers for man-db (2.7.6.1-2) ...
root@backups:~#
```


1.2.9 Setup an admin account

Follow a procedure to install all accounts of administrators. This is done by the following command:

```
adduser <username>
```

And enter the password and other details as requested:

```
root@analytics:~# adduser eran
Adding user `eran' ...
Adding new group `eran' (1000) ...
Adding new user `eran' (1000) with group `eran' ...
Creating home directory `/home/eran' ...
Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: it is too short
New password:
BAD PASSWORD: it is WAY too short
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for eran
Enter the new value, or press ENTER for the default
    Full Name []: Eran Ben-Shahar
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] Y
root@analytics:~#
```

1.2.10 Add "youradminuser" to sudo group:

```
adduser youradminuser sudo
```

```
root@WLGWEBPOC1:~# adduser eran sudo
Adding user `eran' to group `sudo' ...
Adding user eran to group sudo
Done.
root@WLGWEBPOC1:~#
```

And to allow you later on write files into the websites folder (this will be explained in the next chapters):


```
adduser youradminuser www-data
```

1.2.11 Now logout and login to the “youradminuser” account.

Best practice is to continue the installation from SUDO and not as ROOT, since it will prevent you from doing mistakes (like deleting all system files by mistake)

```
logout
```

1.2.12 Configure network IP numbers

In this section, you are trying to configure the server IPs to communicate with the outer world. You may need to contact the ISP / server hosting company for some details. The Debian network communication documentation is given at:

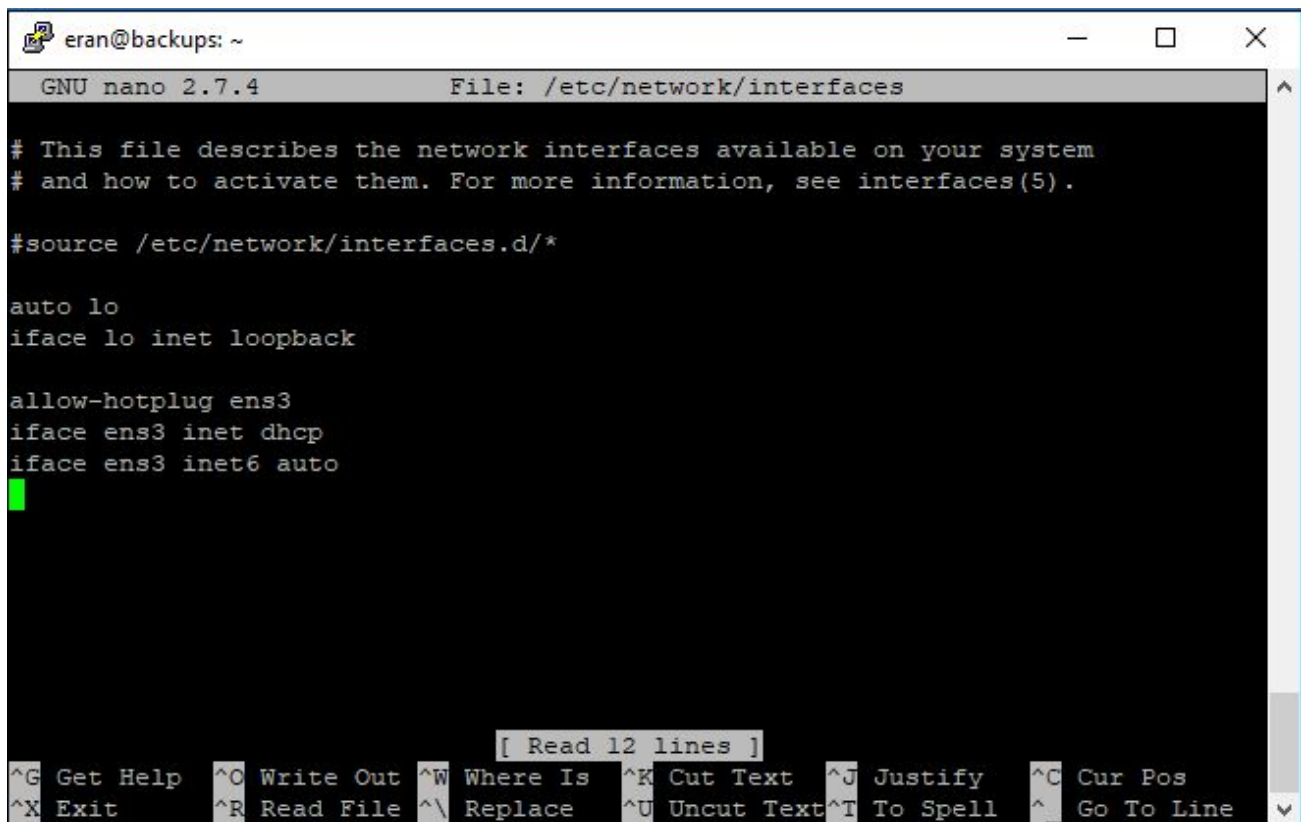
<https://wiki.debian.org/NetworkConfiguration>

There are several situations here, we will describe two of the most common scenarios. To browse the `/etc/network/interfaces` file type:

```
sudo cat /etc/network/interfaces
```

Option number 1: the server host is running DHCP

In that case, the server would ask for the IP allocated by the ISP, so it configures it automatically. The `/etc/network/interfaces` configuration file should be similar to this:



```
eran@backups: ~
GNU nano 2.7.4      File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

#source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

allow-hotplug ens3
iface ens3 inet dhcp
iface ens3 inet6 auto

```

[Read 12 lines]

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^_ Replace	^U Uncut Text	^T To Spell	^_ Go To Line

(on that server, the hotplug name is ens3)

Option number 2: manually configure static IPs

Edit the `/etc/network/interfaces` file so it will reflect the added IP (refer to the appendixes if you require more help):

```
sudo nano /etc/network/interfaces
```



```
GNU nano 2.2.6      File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth1
iface eth1 inet static
    address 71.19.241.189
    netmask 255.255.240.0
    network 71.19.240.0
    broadcast 71.19.255.255
    gateway 71.19.240.1
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 209.139.209.33 8.8.8.8

auto eth0:0
iface eth0:0 inet static
    address 71.19.242.34
    netmask 255.255.240.0
```

For this server, the allocated IPs are 71.19.241.189 and 71.19.242.34 and the hotplug name is eth1. The first lines – of the first IP – were entered by the service provider when they installed the operating system. The last section (with the four lines) was entered in order to add IP# 71.19.242.34 to the server.

For a server with one IP address it may look like this:

```
eran@debian:~$ sudo cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
    address 71.19.249.5
    netmask 255.255.255.0
    network 71.19.249.0
    broadcast 71.19.249.255
    gateway 71.19.249.1
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 8.8.8.8
    dns-search debian
eran@debian:~$
```

Now exit the editor and run the `ifup eth0:0` command:


```
sudo ifup eth0:0
```

And restart the networking with:

```
sudo /etc/init.d/networking restart
```

1.2.13 Installation of PROFTPD

ProFTPD is an open source software which runs a FTP server on the server. It will allow you to connect to the server with any FTP client software (FileZilla is recommended) in order to transfer files from your computer to the server. FTP stands for File Transfer Protocol. ProFTPD is a system server meaning that users that have system accounts (like the one you set previously) will be able to connect to the server. You need to consider then (1) Security (2) File and Directory permissions (see below).

It is important to install the FTP server at this stage in case you wanted to FTP in the websites' files. Another option is to wget the websites' files which means that PROFTPD won't be necessary yet. Still, this installation is fairly fast and easy, so it is a good idea to do it at this stage.

To install it type:

```
sudo apt-get install proftpd
```

It will ask you if you want to install it with inetd or standalone. inetd is for occasional use (it will be loaded on demand). You need to select standalone and click ok button.

After the installation is complete, it is important to re-configure it:

```
sudo nano /etc/proftpd/proftpd.conf
```

We will do three things: change the server's prompt line, jail users to their home directory, and change the server name from "debian" (this is giving information which system we use) to "my-server". Add the following lines to the config file:

```
DefaultRoot ~  
IdentLookups off  
ServerIdent on "FTP Server ready."  
ServerName "My-Server"
```


You could jail users to other directories than their home directories. eran, for example, should be jailed to /srv/www – it has nothing to do outside this directory, on the other hand – this user requires access to all the web server's data:

```
GNU nano 2.7.4      File: /etc/proftpd/proftpd.conf      Modified:
#
# /etc/proftpd/proftpd.conf -- This is a basic ProFTPD configuration file.
# To really apply changes, reload proftpd after modifications, if
# it runs in daemon mode. It is not required in inetd/xinetd mode.
#
# Includes DSO modules
Include /etc/proftpd/modules.conf

# Set off to disable IPv6 support which is annoying on IPv4 only boxes.
UseIPv6                                on
# If set on you can experience a longer connection delay in many cases.
IdentLookups                           off
ServerIdent on [redacted] "FTP server Ready"
ServerName                             "My-Server"
# Set to inetd only if you would run proftpd by inetd/xinetd.
# Read README.Debian for more information on proper configuration.
ServerType                             standalone
DeferWelcome                           off
```

```
# Use this to jail all users in their homes
DefaultRoot /srv/www      eran
DefaultRoot               ~

# Users require a valid shell listed in /etc/shells to login.
# Use this directive to release that constrain.
# RequireValidShell       off

# Port 21 is the standard FTP port.
Port                      21
```

Note that if you jail all users, a specific user jail must be set before the **DefaultRoot ~** command otherwise it gets ignored.

You may want to limit the FTP access to specific IPs. Type "What is my IP" in Google.com to find your IP address, and add the following lines, considering all the IP addresses you work from, in the end of the configuration file `/etc/proftpd/proftpd.conf`:


```
<limit LOGIN>

DenyAll

Allow from 120.0.30.45

</limit>
```

In this example, my IP# is 120.0.30.45

```
GNU nano 2.7.4      File: /etc/proftpd/proftpd.conf      Modified
#  #  Umask          022  022
#  #  <Limit READ WRITE>
#  #  DenyAll
#  #  </Limit>
#  #  <Limit STOR>
#  #  AllowAll
#  #  </Limit>
#  #  </Directory>
#
# </Anonymous>

# Include other custom configuration files
Include /etc/proftpd/conf.d/

<limit LOGIN>
DenyAll
Allow from 120.0.30.45
</limit>
█
```

Save the configuration file and exit. You now need to restart the FTP server:

```
sudo /etc/init.d/proftpd restart
```

```
eran@analytics: ~
eran@analytics:~$ sudo /etc/init.d/proftpd restart
[ ok ] Restarting proftpd (via systemctl): proftpd.service.
eran@analytics:~$ █
```


1.2.14 FTP only shell

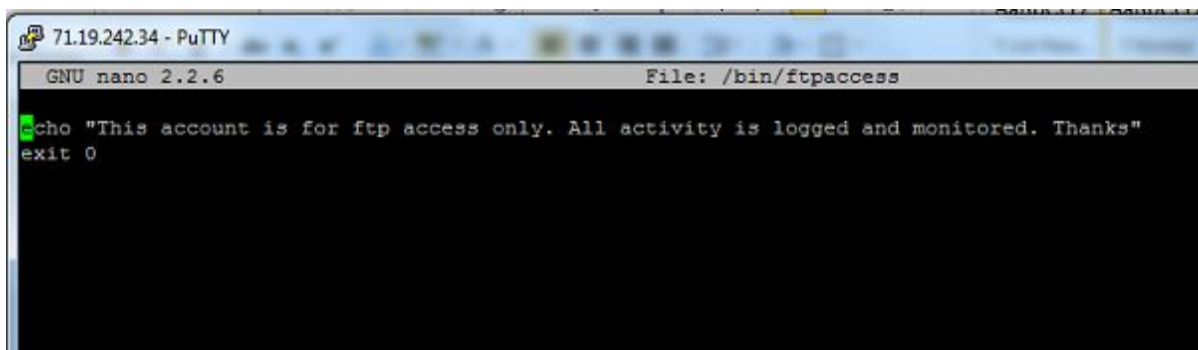
We now create a “ftp only” shell. This will be used in the future for users which we do not want them to login to the system, but we may want them to FTP files (like – outsource developers). Write the following command which will create the new file:

```
sudo nano /bin/ftpassess
```

And add the following lines in it:

```
echo "This account is for ftp access only. All activity is logged and monitored. Thanks"  
exit 0
```

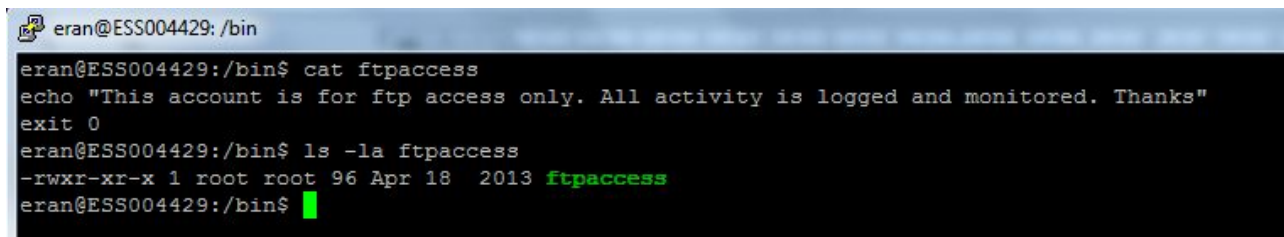
It should look like this:

A screenshot of a terminal window titled "71.19.242.34 - PuTTY". The terminal shows the GNU nano 2.2.6 editor with the file /bin/ftpassess open. The editor contains the following text:

```
echo "This account is for ftp access only. All activity is logged and monitored. Thanks"  
exit 0
```

Save the file and give it Read and Execute permissions to all:

```
sudo chmod go+rx /bin/ftpassess
```

A screenshot of a terminal window showing the verification of the ftpaccess file. The user is logged in as eran@ESS004429. The commands and their outputs are:

```
eran@ESS004429:/bin$ cat ftpaccess  
echo "This account is for ftp access only. All activity is logged and monitored. Thanks"  
exit 0  
eran@ESS004429:/bin$ ls -la ftpaccess  
-rwxr-xr-x 1 root root 96 Apr 18 2013 ftpaccess  
eran@ESS004429:/bin$
```

Now add the “ftpassess” to the list of system shells in `/etc/shells` : edit the file


```
sudo nano /etc/shells
```

And add `/bin/ftpaccess` to it:

```
eran@ESS004429: /etc
eran@ESS004429:/etc$ cat shells
# /etc/shells: valid login shells
/bin/csh
/bin/sh
/usr/bin/es
/usr/bin/ksh
/bin/ksh
/usr/bin/rc
/usr/bin/tcsh
/bin/tcsh
/usr/bin/esh
/bin/dash
/bin/bash
/bin/rbash
/bin/ftpaccess
eran@ESS004429:/etc$
```

Now every user that has a “ftpaccess” shell, will not be able to login to the server with ssh – only with ftp.

Users’ shells are defined in the `/etc/passwd` file. If you edit it, you could disable an ability of a user to login to the server. The following is an example:

```
GNU nano 2.7.4      File: /etc/passwd      Modified

systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
messagebus:x:105:109::/var/run/dbus:/bin/false
ntp:x:106:111::/home/ntp:/bin/false
sshd:x:107:65534::/run/sshd:/usr/sbin/nologin
rdnssd:x:108:65534::/var/run/rdnssd:/bin/false
eran:x:1000:1000:Eran Ben-Shahar,,,:/home/eran:/bin/ftpaccess
proftpd:x:109:65534::/run/proftpd:/bin/false
ftp:x:110:65534::/srv/ftp:/bin/false

^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```


Note that if you add user to group / make changes, changes will take effect only after login again.

1.2.15 Remove shell access from *all users*

Now remove all shell access from all users except the one which are trusted.

1.2.16 MySQL Database server installation

The following command will install MySQL server application:

```
sudo apt-get install mysql-server mysql-client
```

Make the following selections:

[illegible]

During the installation you would be asked to choose a mysql root username and password. If not, you can set it manually:

In case the current password is empty:


```
sudo mysqladmin -u root password 'newpass'
```

In case the current password is not empty i.e. already set, use:

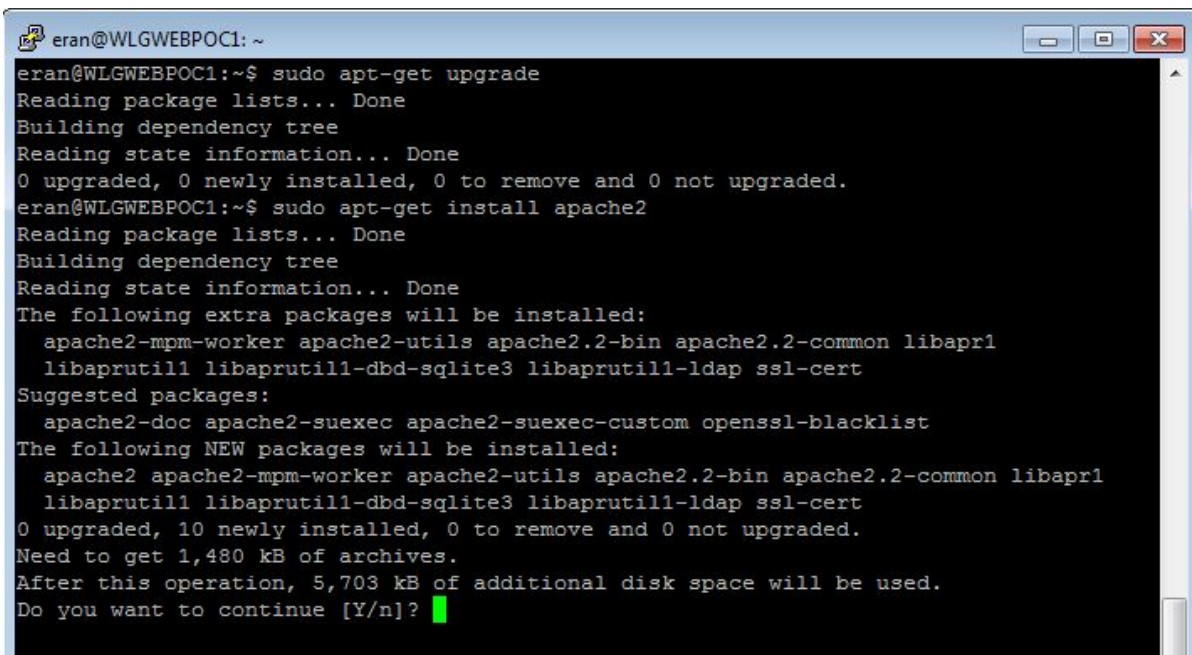
```
mysqladmin -u root -p'oldpass' password  
'newpass'
```

Note that when you enter a MYSQL password in command line, there isn't any space between the `-p` flag and the password itself.

1.2.17 Apache2 Installation

Install apache2 web server with the following command:

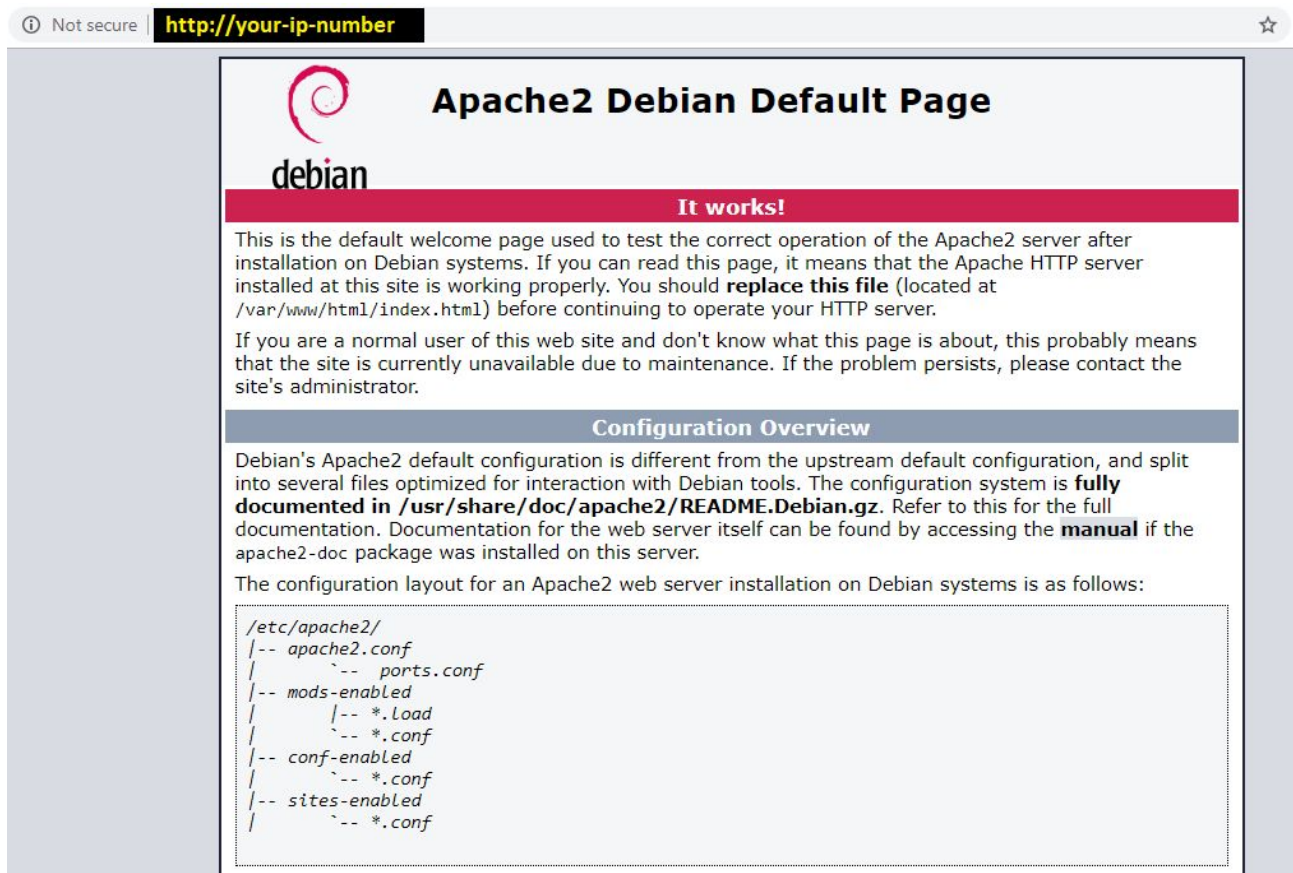
```
sudo apt-get install apache2
```



```
eran@WLGWEBPOC1: ~  
eran@WLGWEBPOC1:~$ sudo apt-get upgrade  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
eran@WLGWEBPOC1:~$ sudo apt-get install apache2  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common libapr1  
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert  
Suggested packages:  
  apache2-doc apache2-suexec apache2-suexec-custom openssl-blacklist  
The following NEW packages will be installed:  
  apache2 apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common libapr1  
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert  
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.  
Need to get 1,480 kB of archives.  
After this operation, 5,703 kB of additional disk space will be used.  
Do you want to continue [Y/n]? █
```

choose Y to complete the installation.

Once the installation is complete, your server is exposed live to the internet, you should try it by writing the IP number in the browser address line, the page you should get is the default apache webserver page:



To make sure that this is your server, update the `/var/www/html/index.html` file:

```
sudo nano /var/www/html/index.html
```


Add for example the IP number to the "it works!" line:


```
GNU nano 2.7.4 File: /var/www/html/index.html

    <a href="#files">Config files</a>
  </div>
</div>
-->
<div class="content_section floating_element">

  <div class="section_header section_header_red">
    <div id="about"></div>
    It works! My server is ok!
  </div>
  <div class="content_section_text">
    <p>
      This is the default welcome page used to test the correct
      operation of the Apache2 server after installation on Debian systems.
      If you can read this page, it means that the Apache HTTP server install$
      this site is working properly. You should <b>replace this file</b> (loc$
      <tt>/var/www/html/index.html</tt>) before continuing to operate your HT$
    </p>
  </div>
</div>
[ Wrote 368 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Now save the file and refresh the browser screen:



Apache2 Debian Default Page

It works! My server is ok!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. 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

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian 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 Debian systems is as follows:

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


1.2.18 Enable basic modules

Enable SSL and REWRITE and some other useful modules with the following commands:

```
sudo a2enmod ssl
```

```
sudo a2enmod rewrite
```

```
sudo a2enmod expires
```

```
sudo a2enmod deflate
```

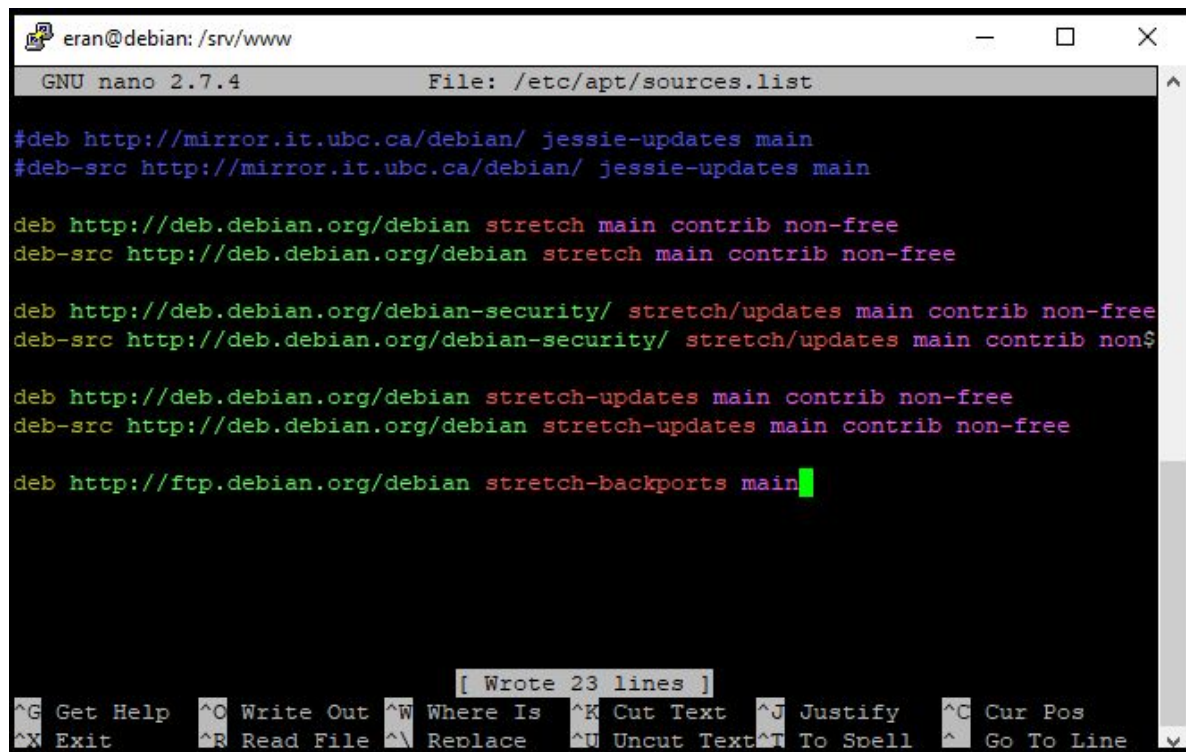
Changes will take effect after the next restart (coming below).

1.2.19 Install Lets Encrypt agent

Lets encrypt is a free public domain SSL certificates issuer, to install the SSL agent it type. Add the following repository-

```
deb http://ftp.debian.org/debian stretch-backports main  
to /etc/apt-get/sources.list:
```

```
sudo nano /etc/apt/sources.list
```

```
eran@debian: /srv/www
GNU nano 2.7.4 File: /etc/apt/sources.list

#deb http://mirror.it.ubc.ca/debian/ jessie-updates main
#deb-src http://mirror.it.ubc.ca/debian/ jessie-updates main

deb http://deb.debian.org/debian stretch main contrib non-free
deb-src http://deb.debian.org/debian stretch main contrib non-free

deb http://deb.debian.org/debian-security/ stretch/updates main contrib non-free
deb-src http://deb.debian.org/debian-security/ stretch/updates main contrib non$

deb http://deb.debian.org/debian stretch-updates main contrib non-free
deb-src http://deb.debian.org/debian stretch-updates main contrib non-free

deb http://ftp.debian.org/debian stretch-backports main
```

And now install certbot by running the following command:

```
sudo apt-get update
```

```
sudo apt-get install python-certbot-apache -t stretch-backports
```

1.2.20 PHP7 Installation

Run the following command:

```
sudo apt-get install php7.0
```

You should be prompted with the following. Click Y to approve the installation:


```
eran@debian: /srv/www/public
eran@debian:/srv/www/public$ sudo apt-get install php7.0
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
 docutils-common docutils-doc libalgorithm-c3-perl libarchive-extract-perl libasprintf0c2 libclass-c3-perl
 libclass-c3-xs-perl libcpan-meta-perl libdata-optlist-perl libdata-section-perl libintl-perl liblcms2-2
 liblog-message-perl liblog-message-simple-perl libmodule-build-perl libmodule-pluggable-perl
 libmodule-signature-perl libmro-compatible-perl libpackage-constants-perl libpaper-utils libpaperl
 libparams-util-perl libperl4-corelibs-perl libpod-latex-perl libpod-readme-perl libregex-common-perl
 libsoftware-license-perl libsub-exporter-perl libsub-install-perl libterm-ui-perl libtext-soundex-perl
 libtext-template-perl libwebp5 libwebpdemux1 libwebpmux1 libxapian22 python-docutils python-pil python-pygment
 python-roman
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
 libapache2-mod-php7.0 php-common php7.0-cli php7.0-common php7.0-json php7.0-opcache php7.0-readline
Suggested packages:
 php-pear
The following NEW packages will be installed:
 libapache2-mod-php7.0 php-common php7.0-cli php7.0-common php7.0-json php7.0-opcache php7.0-readline
0 upgraded, 8 newly installed, 0 to remove and 90 not upgraded.
Need to get 3,559 kB of archives.
After this operation, 14.0 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

The apache server should be automatically restarted after this operation. But we need to make some changes to the php module, open the php module configuration file:

```
sudo nano /etc/apache2/mods-available/php7.0.conf
```

and add the following lines:

```
#allow php to run in .html, .htm files
<FilesMatch "\.(htm|html)$">
    SetHandler application/x-httpd-php
</FilesMatch>
```

It would look like that:


```
eran@debian: /srv/www/public/zombiebutcher.com
GNU nano 2.7.4 File: /etc/apache2/mods-enabled/php7.0.conf

<FilesMatch ".+\.ph(p[3457]?|t|tml)$">
    SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch ".+\.phps$">
    SetHandler application/x-httpd-php-source
    # Deny access to raw php sources by default
    # To re-enable it's recommended to enable access to the files
    # only in specific virtual host or directory
    Require all denied
</FilesMatch>

#allow php to run in .html, .htm files
<FilesMatch "\.(htm|html)$">
    SetHandler application/x-httpd-php
</FilesMatch>

# Deny access to files without filename (e.g. '.php')
<FilesMatch "^\.ph(p[3457]?|t|tml|ps)$">
    Require all denied
</FilesMatch>

# Running PHP scripts in user directories is disabled by default
#
# To re-enable PHP in user directories comment the following lines
# (from <IfModule ...> to </IfModule>.) Do NOT set it to On as it
# prevents .htaccess files from disabling it.
<IfModule mod_userdir.c>
    <Directory /home/*/public_html>
        php_admin_flag engine Off
    </Directory>
</IfModule>

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line ^V Next Page
```

Now restart apache::

```
sudo /etc/init.d/apache2 restart
```

And now create a test.php file in /srv/www/public/about.html that contains:

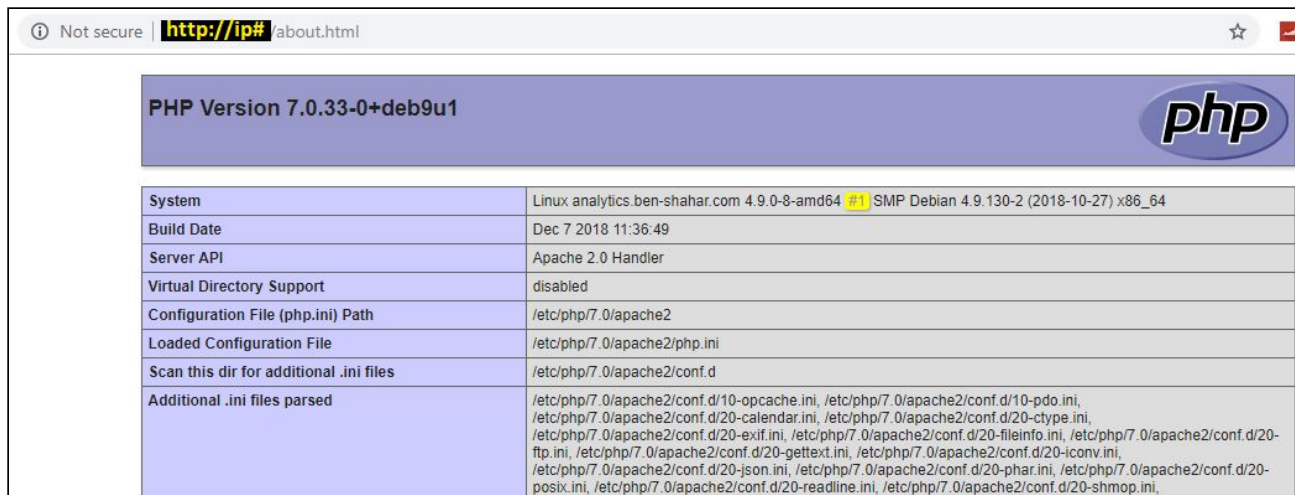
```
<?php phpinfo(); ?>
```


```
sudo nano /var/www/html/about.html
```

```
GNU nano 2.7.4 File: /var/www/html/about.html

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


Access the file through http://your_ip/about.html - and check if it is working properly



PHP Version 7.0.33-0+deb9u1 	
System	Linux analytics.ben-shahar.com 4.9.0-8-amd64 SMP Debian 4.9.130-2 (2018-10-27) x86_64
Build Date	Dec 7 2018 11:36:49
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini,

1.2.21 Additional crucial Installations

The following commands will install crucial libraries which are required to run LAMP (=Linux Apache Mysql PHP) server:

```
sudo apt-get install javascript-common
```

```
sudo apt-get install php7.0-curl
```

```
sudo apt-get install php7.0-mysql
```

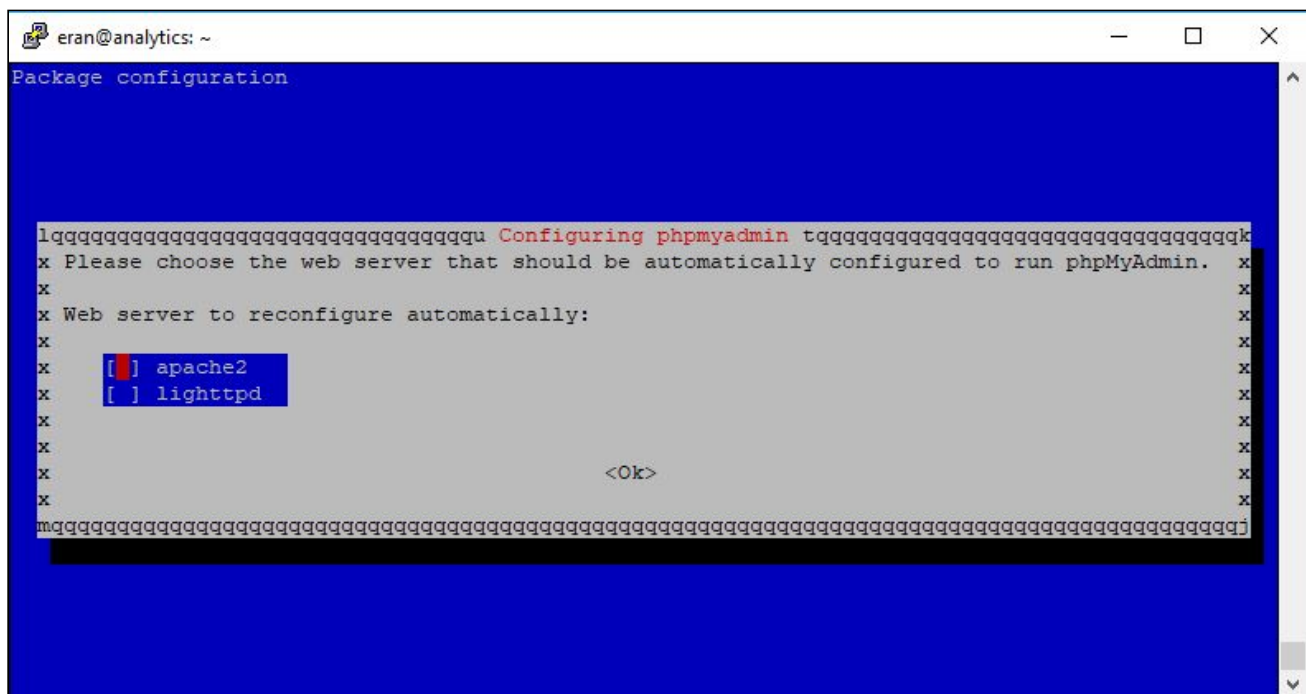
1.2.22 Installation of PHPMYADMIN

PHPMYADMIN is a useful (almost necessary) web application that lets you manage the MySQL server and MySQL databases. To install it:

```
sudo apt-get update
```

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


You will be prompted with the installation configuration screens:



- Select Apache2 for the server
- Choose YES when asked about whether to Configure the database for phpmyadmin with dbconfig-common
- Choose a MySQL password when prompted
- Enter the password that you want to use to log into phpmyadmin – note this is a different password to the MySQL root password. ALWAYS CHOOSE COMPLICATED LONG PASSWORD WITH ODD CHARACTERS.. **note: the phpmyadmin default user is phpmyadmin. The credentials are saved during the installation in /etc/phpmyadmin/donfig-db.php**

Now add the phpmyadmin configuration file to the apache2 configuration:

```
sudo nano /etc/apache2/apache2.conf
```

and add the following line: `Include /etc/phpmyadmin/apache.conf` , also the directory default lines which would secure the default apache2 directories from users to be able to access them by default:


```
eran@debian: /usr/share/phpmyadmin
GNU nano 2.7.4 File: /etc/apache2/apache2.conf

LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

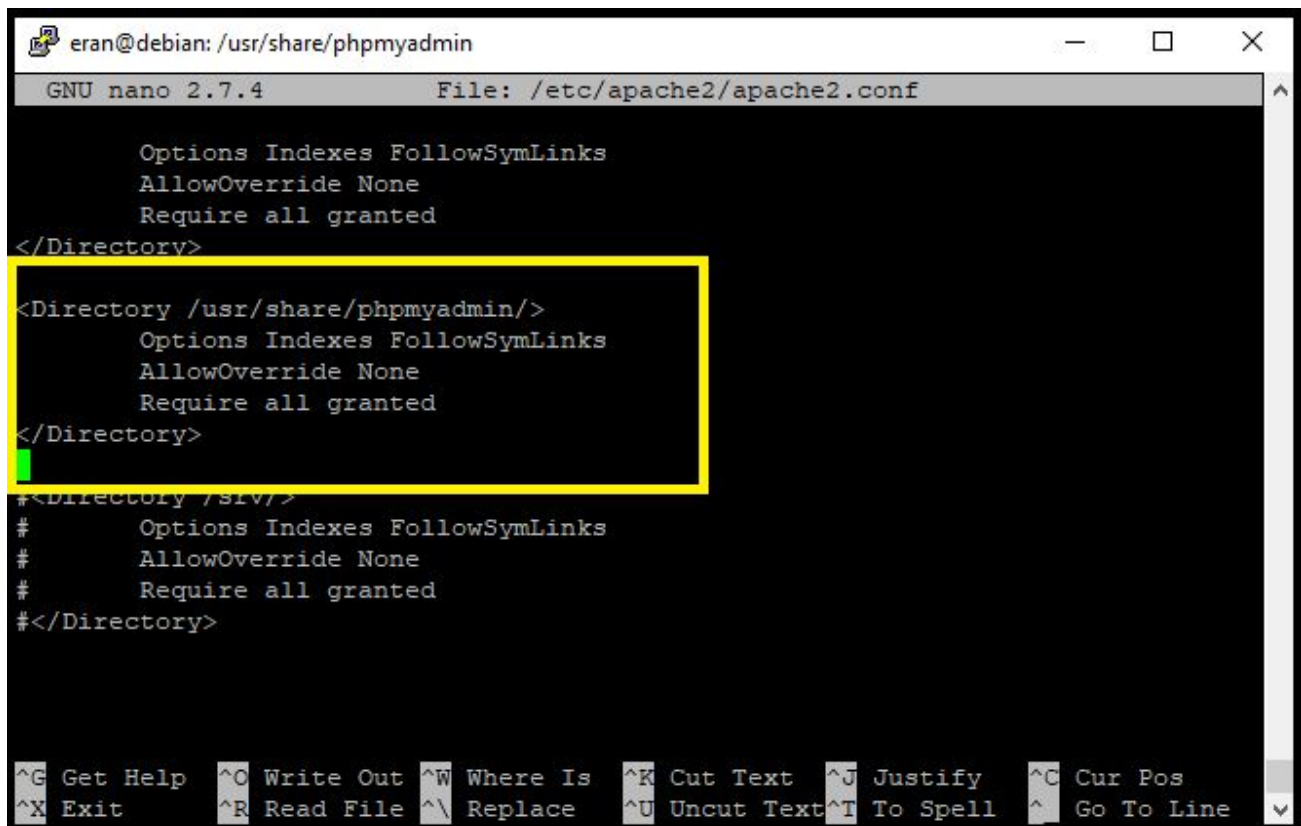
Include /etc/phpmyadmin/apache.conf

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

And allow apache to access to /usr/share/phpmyadmin folder which is the folder where this app is hosted. Add the following lines:

```
<Directory /usr/share/phpmyadmin/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

It looks like that:



```
eran@debian: /usr/share/phpmyadmin
GNU nano 2.7.4 File: /etc/apache2/apache2.conf

Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>
<Directory /usr/share/phpmyadmin/>
Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>
#<Directory /srv/>
# Options Indexes FollowSymLinks
# AllowOverride None
# Require all granted
#</Directory>

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

It is also important to hide some of the server information – to prevent users from knowing which version of Linux, Apache and PHP we are running. Add the following two directives to Apache2.conf:

ServerTokens ProductOnly

ServerSignature Off


```
eran@analytics: ~  
GNU nano 2.7.4 File: /etc/apache2/apache2.conf  
LogFormat "%{Referer}i -> %U" referer  
LogFormat "%{User-agent}i" agent  
  
# Include of directories ignores editors' and dpkg's backup files,  
# see README.Debian for details.  
  
# Include generic snippets of statements  
IncludeOptional conf-enabled/*.conf  
  
# Include the virtual host configurations:  
IncludeOptional sites-enabled/*.conf  
  
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet  
  
Include /etc/phpmyadmin/apache.conf  
ServerTokens ProductOnly  
ServerSignature Off
```

And restart the apache2 server:

```
sudo /etc/init.d/apache2 restart
```

One of the issues with phpmyadmin is security. In order to prevent hackers from trying to enter your database, you should make some changes. The first one is to change the phpmyadmin access path so when bots / hackers trying to access your phpmyadmin folder, they will get a "404 not found" message. To do that:

```
sudo nano /etc/phpmyadmin/apache.conf
```

and add the line `Alias /phpmyadminsABCDEFG /usr/share/phpmyadmin` to it:


```
GNU nano 2.7.4 File: /etc/phpmyadmin/apache.conf Modified
# phpMyAdmin default Apache configuration

Alias /phpmyadminABCDEF /usr/share/phpmyadmin

<Directory /usr/share/phpmyadmin>
    Options SymLinksIfOwnerMatch
    DirectoryIndex index.php

    <IfModule mod_php5.c>
        <IfModule mod_mime.c>
            AddType application/x-httpd-php .php
        </IfModule>
        <FilesMatch ".+\.php$">
            SetHandler application/x-httpd-php
        </FilesMatch>

        php_value include_path .
        php_admin_value upload_tmp_dir /var/lib/phpmyadmin/tmp
        php_admin_value open_basedir /usr/share/phpmyadmin:/etc/phpmyadmin:/var/lib/phpmyadmin

    </IfModule>
</Directory>

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

You can obviously choose any alias you want, and it's not a bad idea to change it from time to time.

And change the file `/etc/php/7.0/apache2/php.ini` so the PHP directive `expose_php` will be Off:

```
sudo nano /etc/php/7.0/apache2/php.ini
```

```
eran@debian: /usr/share/phpmyadmin
GNU nano 2.7.4 File: /etc/php/7.0/apache2/php.ini

; unless "declare(encoding=...)" directive appears at the top of the script.
; Only affects if zend.multibyte is set.
; Default: ""
;zend.script_encoding =

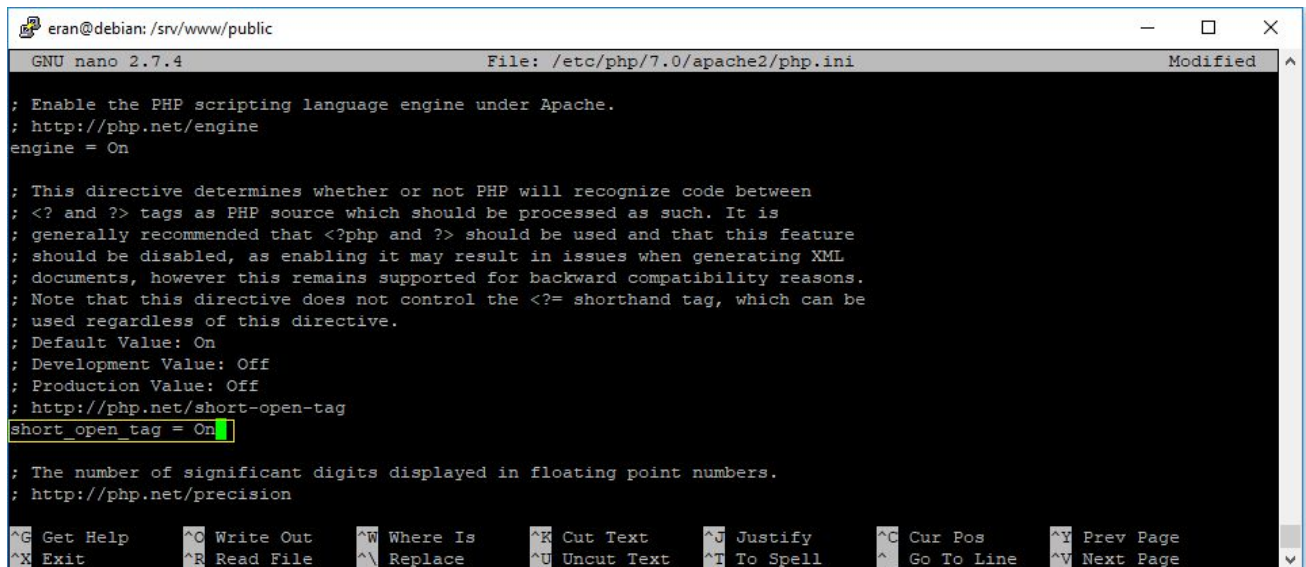
;
; Miscellaneous
;

; Decides whether PHP may expose the fact that it is installed on the server
; (e.g. by adding its signature to the Web server header). It is no security
; threat in any way, but it makes it possible to determine whether you use PHP
; on your server or not.
; http://php.net/expose-php
expose_php = Off

;
; Resource Limits
;

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```


and change the PHP directive `short_open_tag = On` so code using the shorthand `<?>` opening for php code will be recognized:



```
eran@debian: /srv/www/public
GNU nano 2.7.4 File: /etc/php/7.0/apache2/php.ini Modified
; Enable the PHP scripting language engine under Apache.
; http://php.net/engine
engine = On

; This directive determines whether or not PHP will recognize code between
; <? and ?> tags as PHP source which should be processed as such. It is
; generally recommended that <?php and ?> should be used and that this feature
; should be disabled, as enabling it may result in issues when generating XML
; documents, however this remains supported for backward compatibility reasons.
; Note that this directive does not control the <?=> shorthand tag, which can be
; used regardless of this directive.
; Default Value: On
; Development Value: Off
; Production Value: Off
; http://php.net/short-open-tag
short_open_tag = On

; The number of significant digits displayed in floating point numbers.
; http://php.net/precision

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line ^V Next Page
```

and restart apache2:

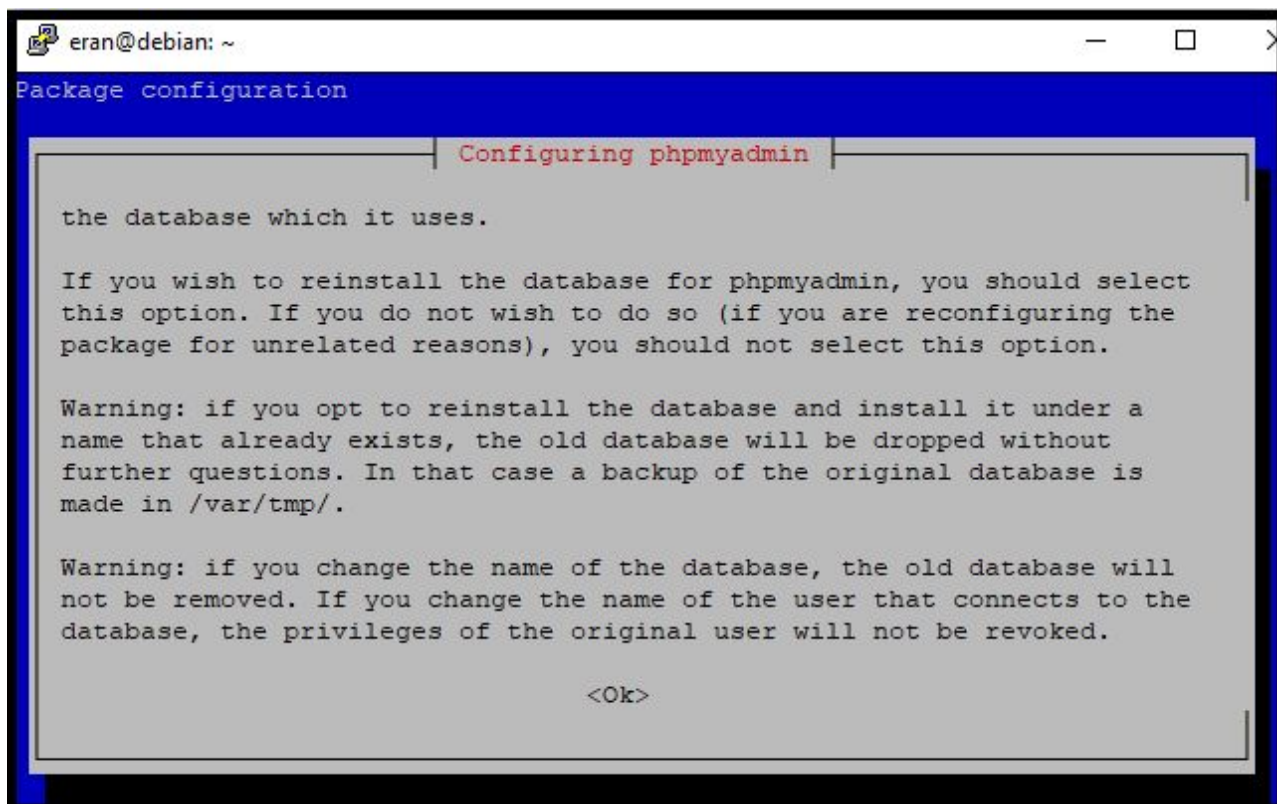
```
sudo /etc/init.d/apache2 restart
```

I STOPPED HERE

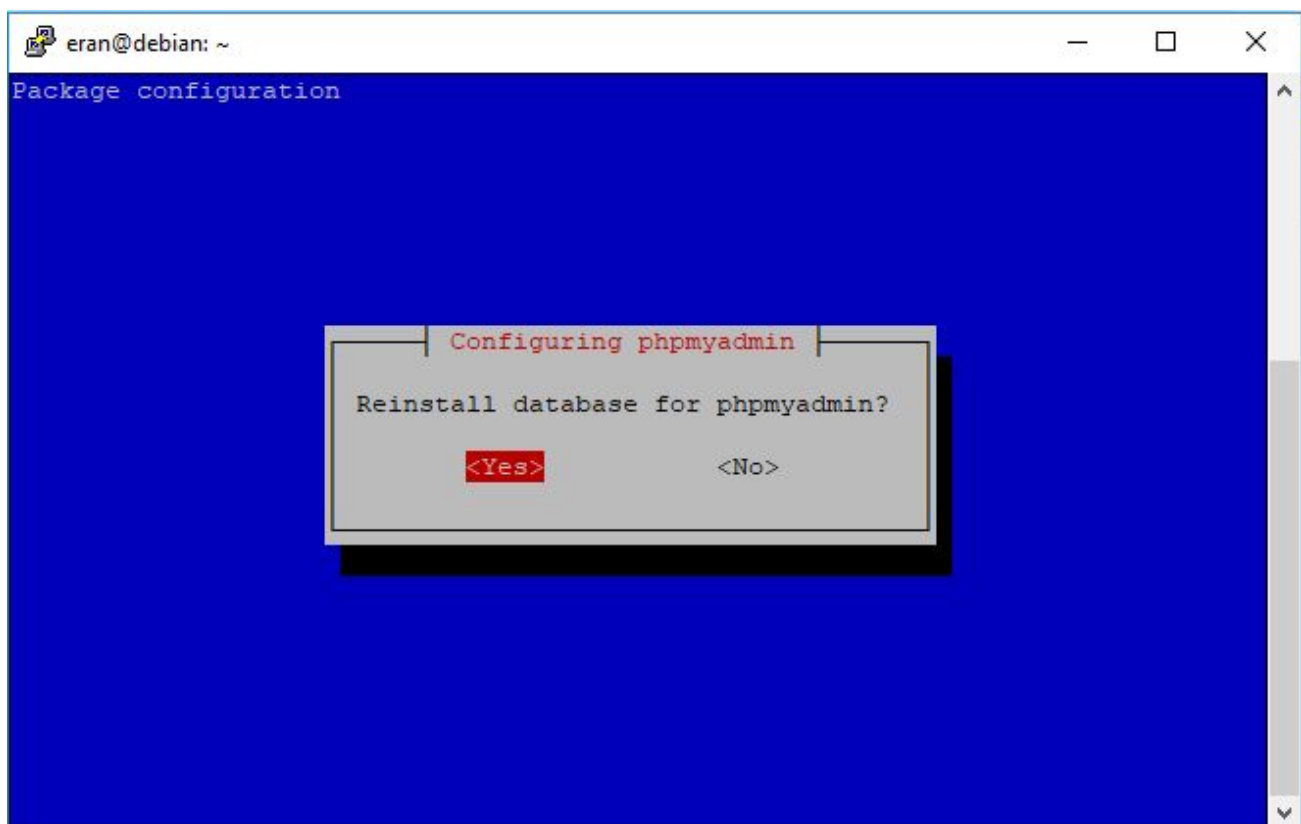
1.2.23 Reconfigure phpmyadmin

The new version of phpmyadmin does not allow to connect to phpmyadmin with a root account. To fix this:

```
sudo dpkg-reconfigure phpmyadmin
```

Choose OK.



Choose Yes

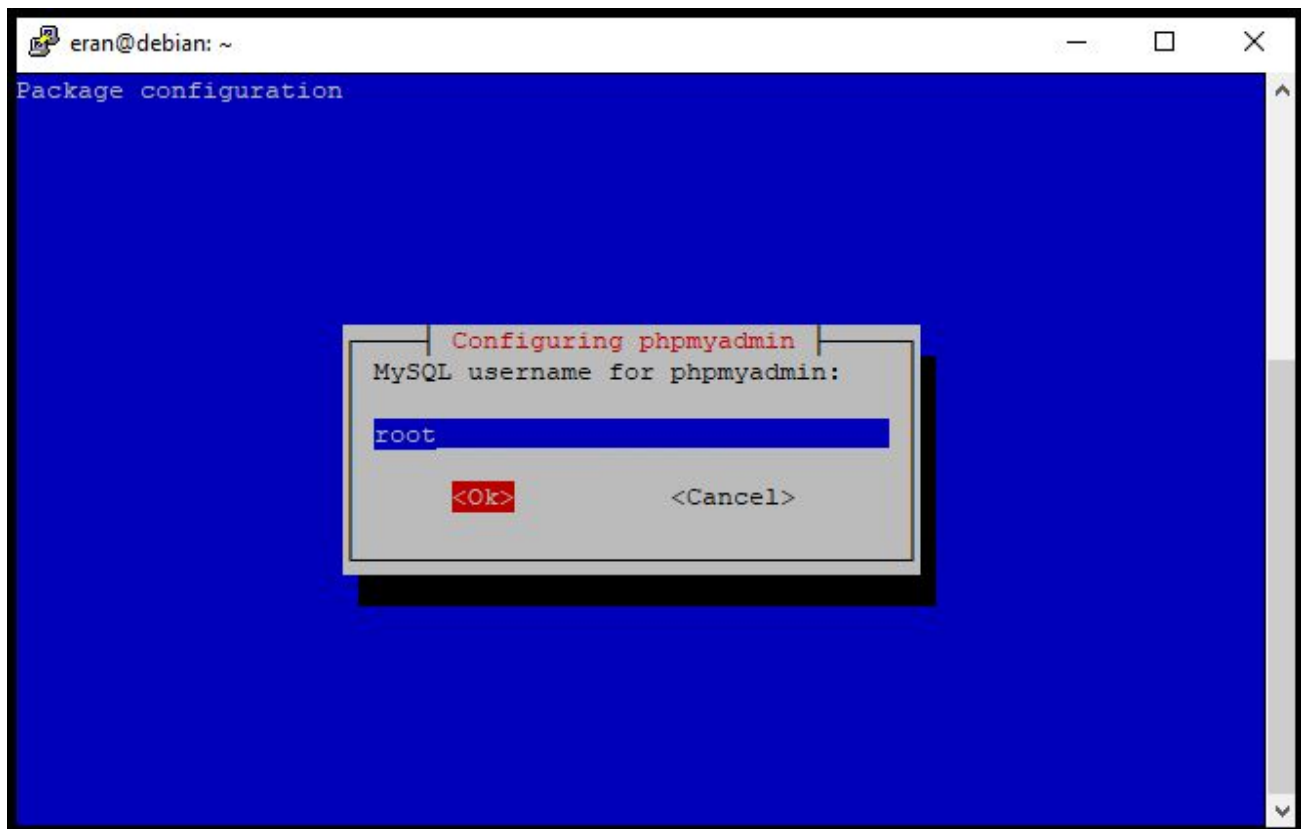
Choose TCP/IP

Choose localhost

Select 3306

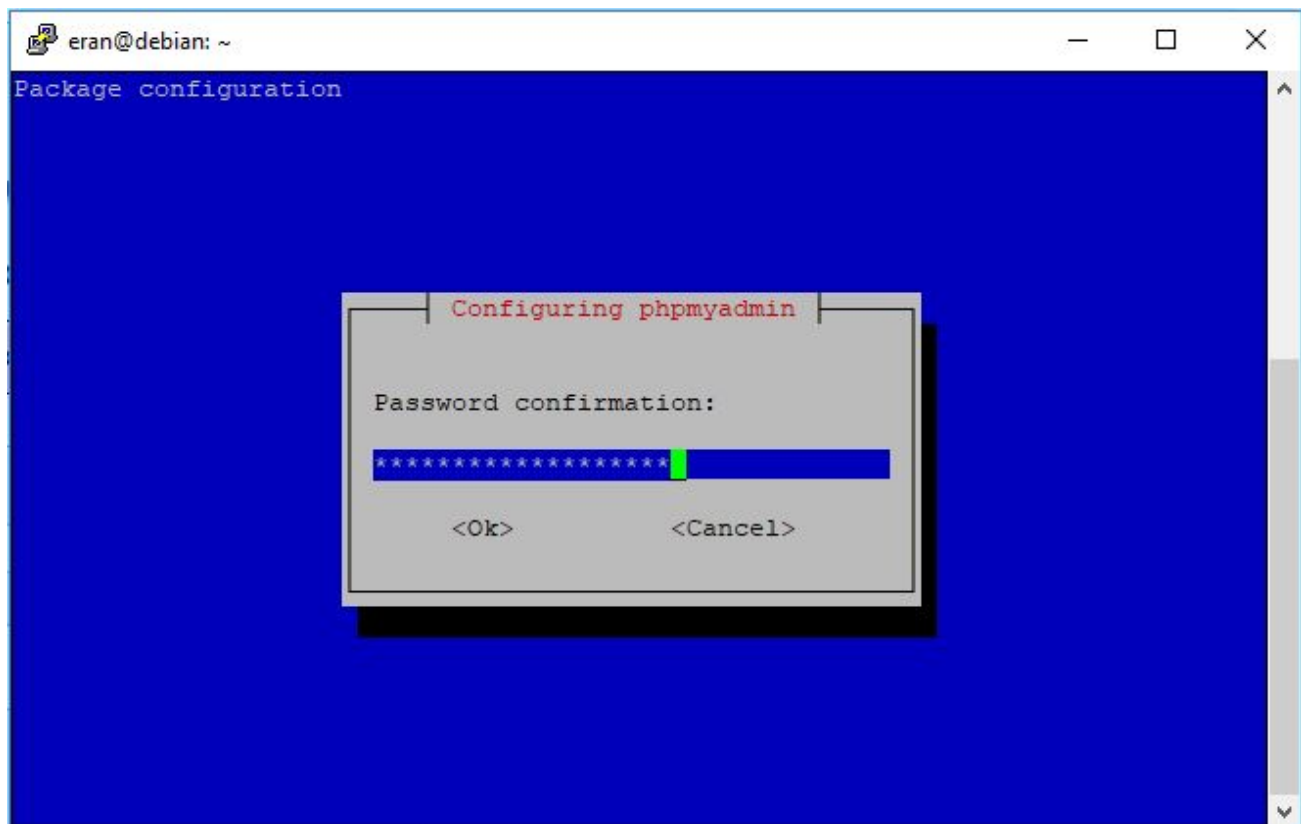
Choose phpmyadmin,

Choose OK.

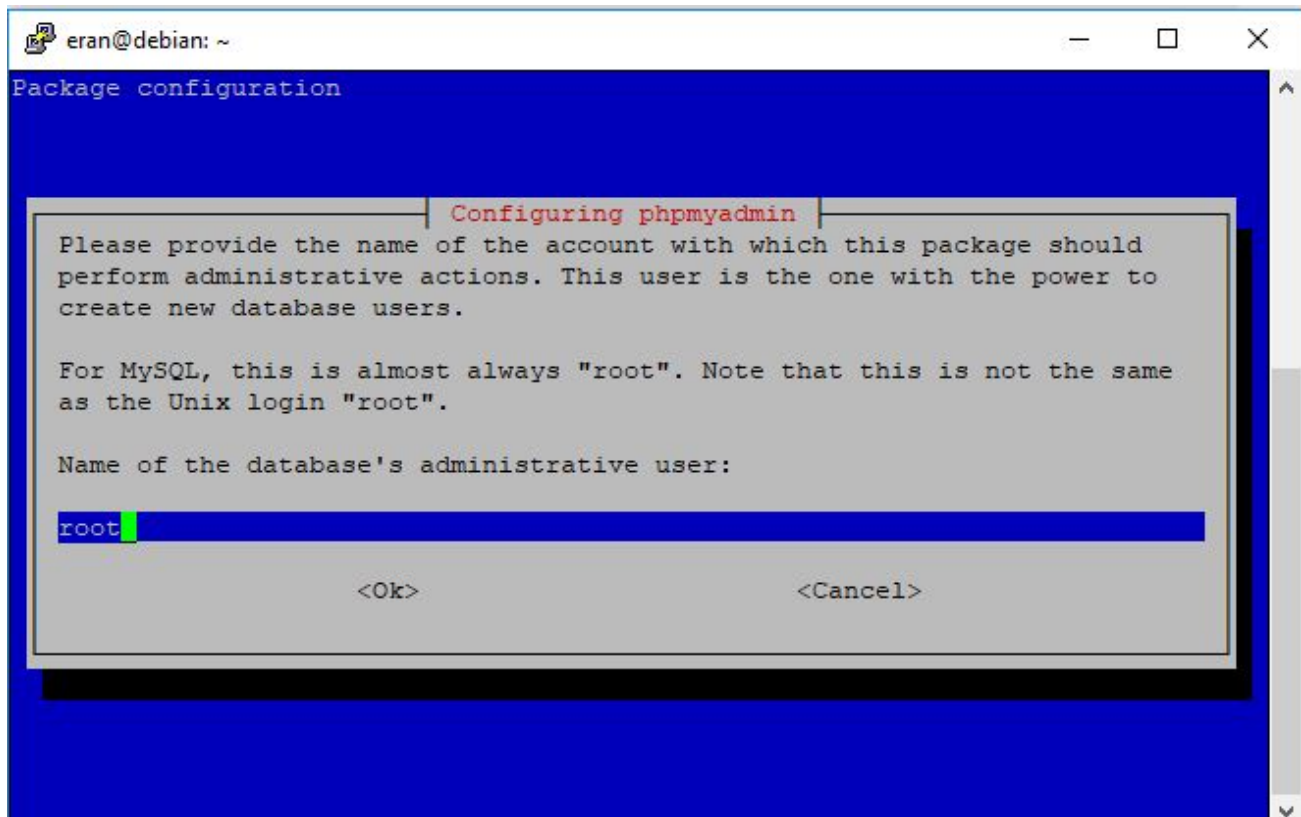


Replace the "phpmyadmin@localhost" with "root" as above screenshot

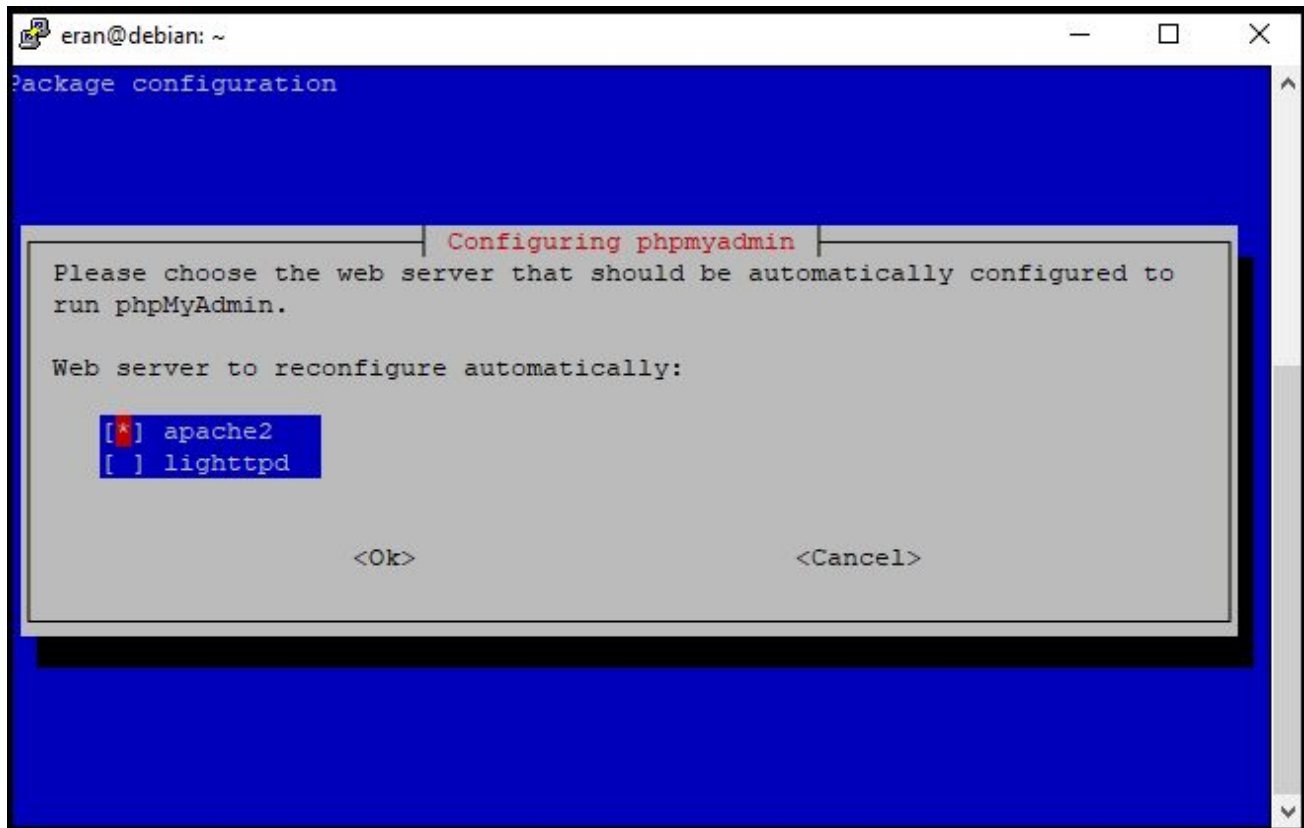
Type in the MYSQL root password



Retype the password for confirmation.

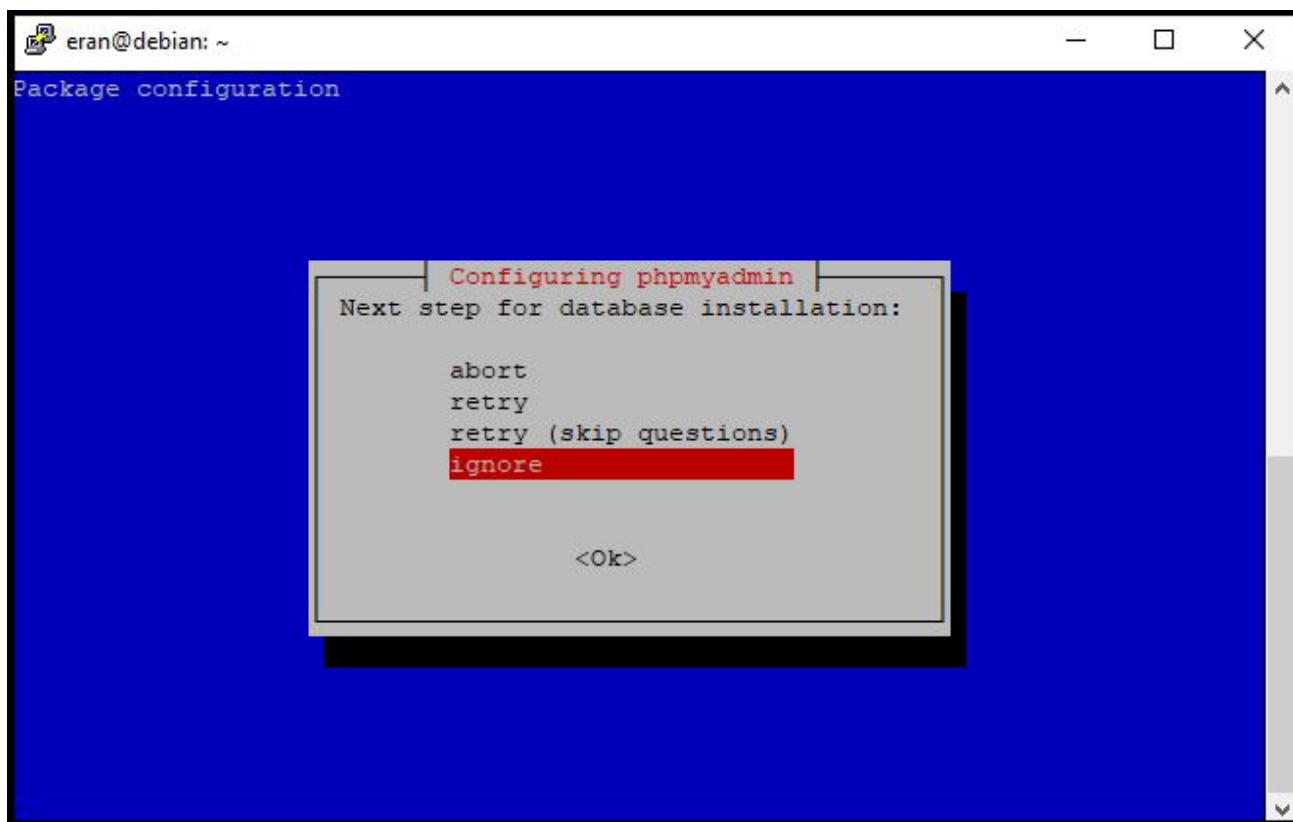


Choose root again.



Select "apache2" (with space) and click enter.

After the error message, click ok:



Choose ignore and click enter. The operation will complete and you will get the shell environment back. You could now login to root account on http://YOUR_IP/phpmyadminsecuredlocation7655438986/

1.2.24 Installation of PHP-MAIL

The following installation and configuration will allow the web server to send Emails via PHP, this is used in many of our sites for reporting, automatic emails etc.

The following will install the core package:

```
sudo apt-get install php-mail
```

Then we install a mailer server application. The easiest to configure is EXIM4:

```
sudo apt-get install exim4
```

Now we will configure exim4 by running:

```
sudo dpkg-reconfigure exim4-config
```

In brief, choose the first option in the first screen, then choose "Debian" then type 127.0.0.1:1

In details, the process looks like that:

In the first screen, choose "internet site":

Package configuration

Mail Server configuration

General type of mail configuration:

- internet site; mail is sent and received directly using SMTP
- mail sent by smarthost; received via SMTP or fetchmail
- mail sent by smarthost; no local mail
- local delivery only; not on a network
- no configuration at this time

<Ok> <Cancel>

Next choose the default domain name should be used – we used “yodfat.com” but you could use “ben-shahar.com” or any other.

Package configuration

Mail Server configuration

The 'mail name' is the domain name used to 'qualify' mail addresses without a domain name.

This name will also be used by other programs. It should be the single, fully qualified domain name (FQDN).

Thus, if a mail address on the local host is foo@example.org, the correct value for this option would be example.org.

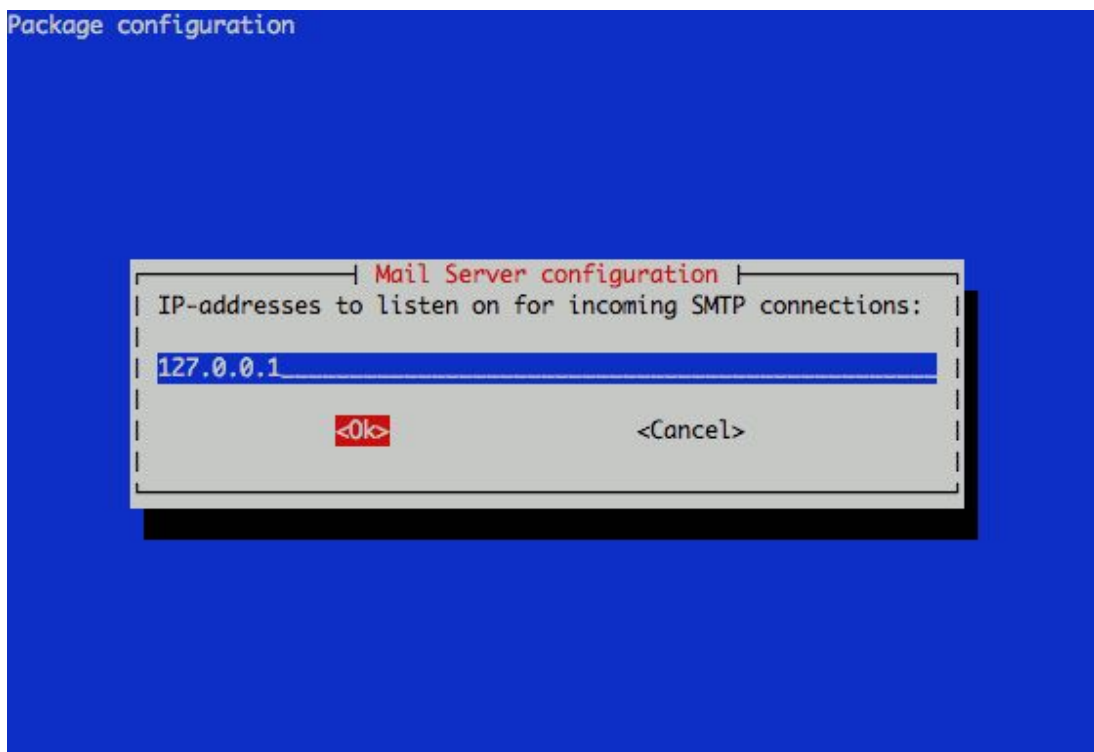
This name won't appear on From: lines of outgoing messages if rewriting is enabled.

System mail name:

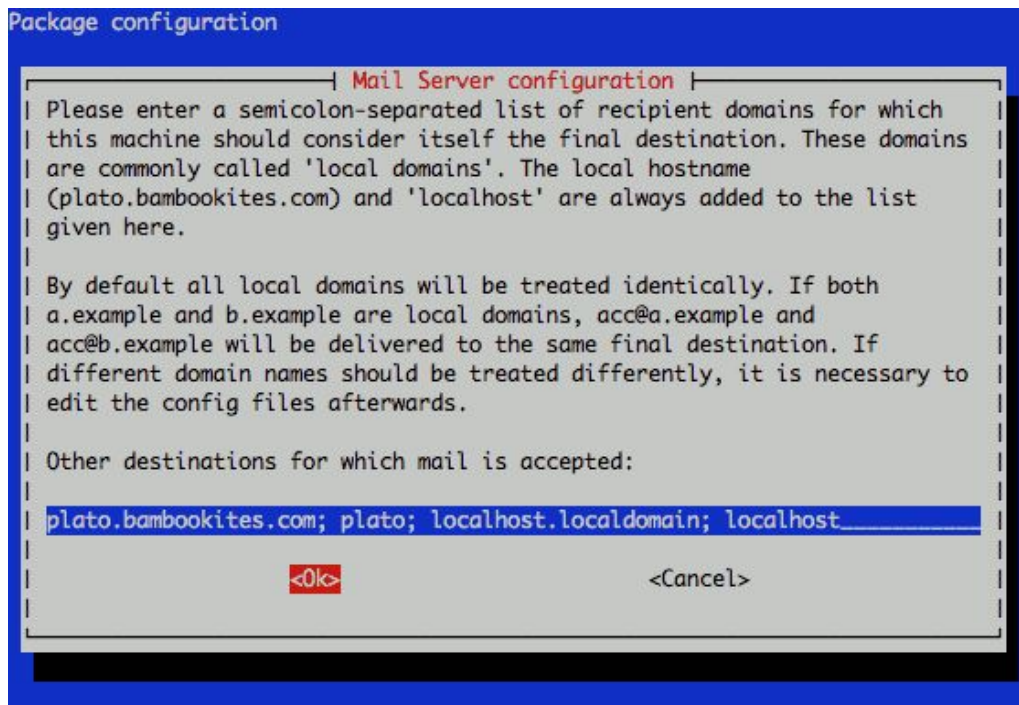
plato.bambookites.com

<Ok> <Cancel>

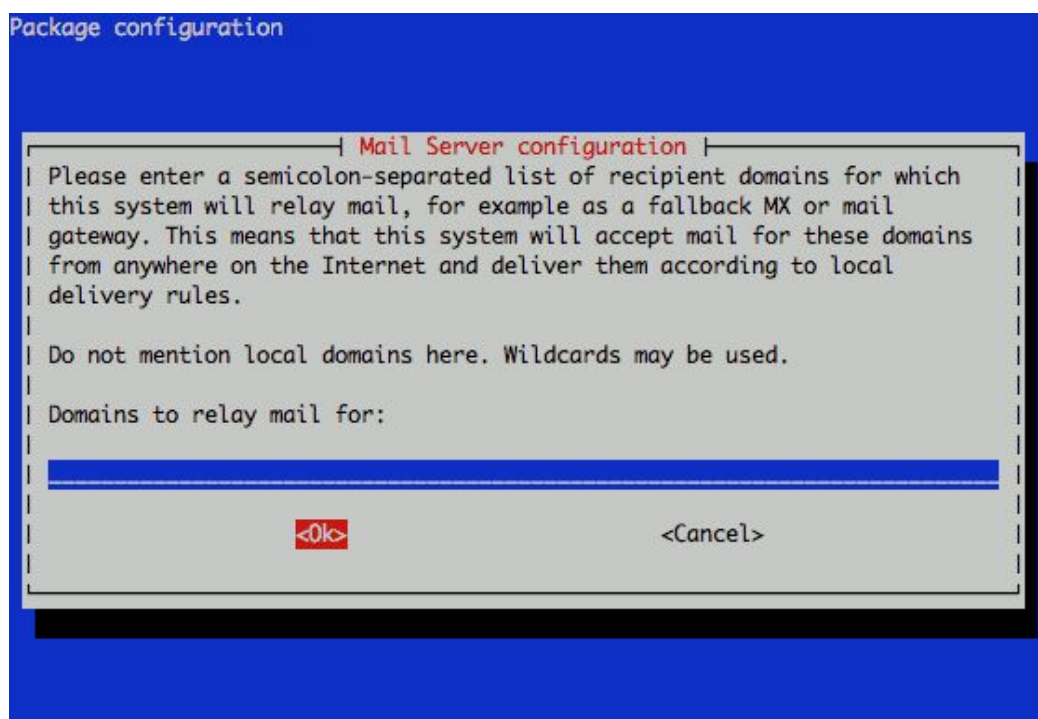
Enter 127.0.0.1 in the following screen:

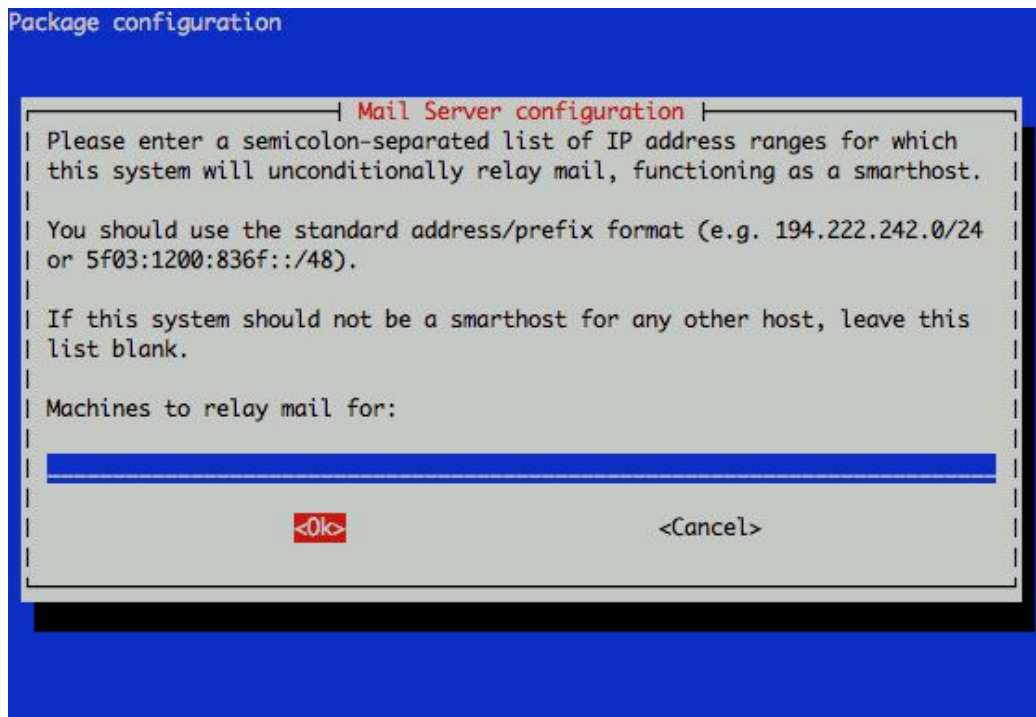


Make sure you list your FQDN, hostname, and localhost entries when you're asked which destinations mail should be accepted for:



Leave the relay domains and relay machines fields blank:

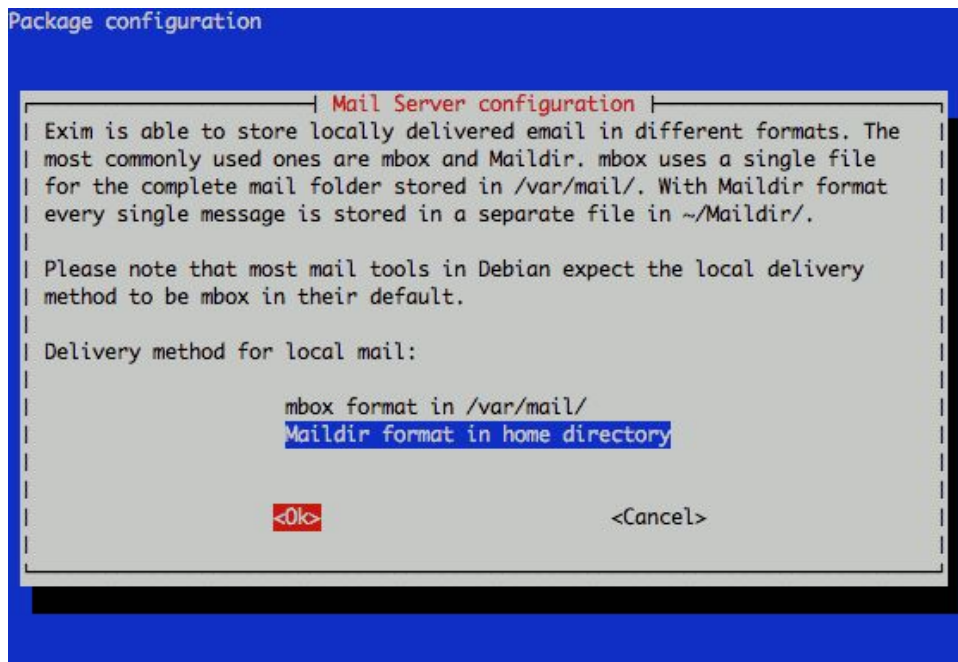




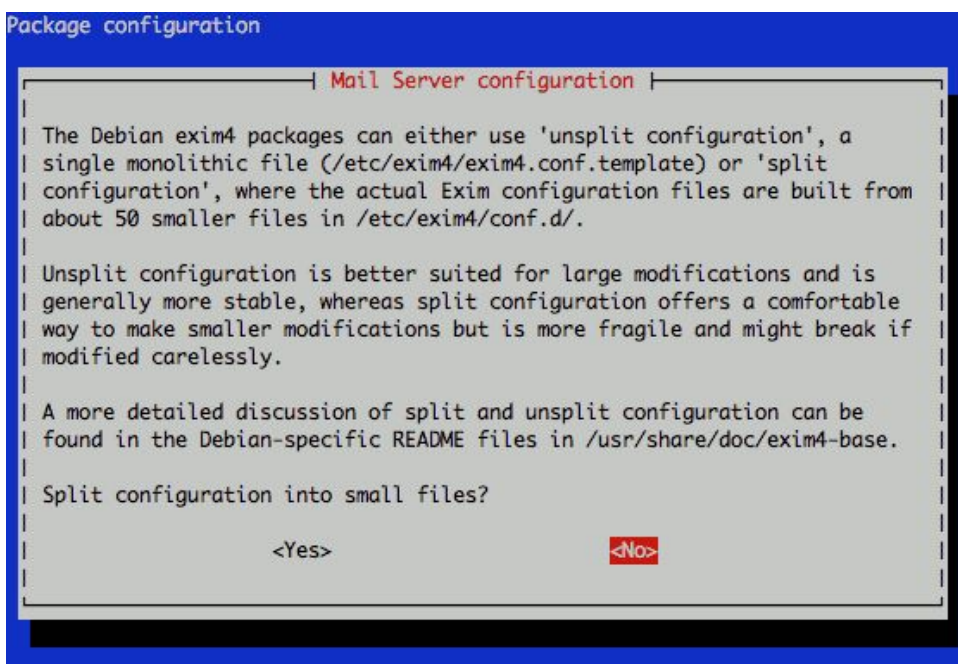
Select "No" when asked whether to keep DNS queries to a minimum:



You may select either "mbox" or "Maildir" when asked about the delivery method used for incoming mail. While many utilities use mbox format, Maildir format can make handling individual locally delivered mail messages easier, and is widely supporting by a range of applications:



Accept the default “non-split” option for your mail configuration file:



In case you still have issues, you could edit the configuration file to match with the following:

```
sudo nano /etc/exim4/update-exim4.conf.conf
```



```
# /etc/exim4/update-exim4.conf.conf
#
# Edit this file and /etc/mailname by hand and execute update-exim4.conf
# yourself or use 'dpkg-reconfigure exim4-config'
#
# Please note that this is _not_ a dpkg-conf file and that automatic changes
# to this file might happen. The code handling this will honor your local
# changes, so this is usually fine, but will break local schemes that mess
# around with multiple versions of the file.
#
# update-exim4.conf uses this file to determine variable values to generate
# exim configuration macros for the configuration file.
#
# Most settings found in here do have corresponding questions in the
# Debconf configuration, but not all of them.
#
# This is a Debian specific file

dc_eximconfig_configtype='internet'
dc_other_hostnames=""
dc_local_interfaces='127.0.0.1'
dc_readhost='mailhost'
dc_relay_domains=""
dc_minimaldns='false'
dc_relay_nets=""
dc_smarthost=""
CFILEMODE='644'
dc_use_split_config='false'
dc_hide_mailname='true'
dc_mailname_in_oh='true'
dc_localdelivery='mail_spool'
```

Now, search for the “sendmail” line in /etc/php5/apache2/php.ini file and update it to the following, if it does not exist then just add it:

```
sudo nano /etc/php/7.0/apache2/php.ini
```

```
sendmail_path = /usr/sbin/sendmail -t -i
```



```
eran@debian: /srv/www/public
GNU nano 2.7.4 File: /etc/php/7.0/apache2/php.ini

[mail function]
; For Win32 only.
; http://php.net/smtp
SMTP = localhost
; http://php.net/smtp-port
smtp_port = 25

; For Win32 only.
; http://php.net/sendmail-from
;sendmail_from = me@example.com

; For Unix only. You may supply arguments as well (default: "sendmail -t $
; http://php.net/sendmail-path
sendmail_path = /usr/sbin/sendmail -t -i

; Force the addition of the specified parameters to be passed as extra par$
[ Wrote 1919 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell
```

you could confirm that by:

```
eran@debian: /srv/www/public
eran@debian:/srv/www/public$ ls -la /usr/sbin/send*
lrwxrwxrwx 1 root root 5 Feb 10 2018 /usr/sbin/sendmail -> exim4
eran@debian:/srv/www/public$
```

Now restart the apache2 server:

```
sudo /etc/init.d/apache2 restart
```

If you make any changes to the exim4 configuration file, you need to restart exim4:

```
sudo /etc/init.d/exim4 restart
```

php mail should be working now. You can test it by sending an email from one of our websites to yourself (in sites like ZapRobot / s4sfree where the site users could send us emails using the site)

1.2.25 Congratulations! Your LAMP server is installed!

Now that the server is up and running, it is time to: (1) start the files transfer / FTP - if you haven't done so yet, and - (2) create and update all databases - otherwise the applications won't work, (3) transfer / set the domain names records to point to the server, and as a final stage (4) issue all the SSL certificates and setup the HTTPS sites. Note that the SSL certificates are done as a last stage since when you issue a certificate, the authority will check that your DNS record is pointing to the right server - otherwise the SSL certificate won't be created.

More details about how to configure your websites - find in my guide "configuration of websites on LAMP"

APPENDIXES

A) how to add additional IP to a debian server

Assuming the new IP address is on the same subnet as the first, add a second virtual interface (sometimes called an "alias") to the primary network interface. This is configured, like all network interface settings, in `/etc/network/interfaces`. The Debian Reference manual has a section on the topic:

<http://www.debian.org/doc/manuals/debian-reference/ch05.en.html#%5Fthe%5Fvirtual%5Finterface>

A simple example, assuming your primary network interface is `eth0` and has an ip of `192.168.1.1` and the new ip is `192.168.1.2`:

```
auto eth0
iface eth0 inet static
    address 192.168.1.1
    netmask 255.255.255.0
    gateway 192.168.1.254

auto eth0:0
iface eth0:0 inet static
    address 192.168.1.2
    netmask 255.255.255.0
```

Once the appropriate settings have been added to `/etc/network/interfaces`, run `ifup eth0:0` to activate the new interface.

If, however, the new ip is on a different subnet, you need to either provision the ip on a physically distinct network interface or create a VLAN interface, depending on how your ISP is prepared to hand it off to you. That's a whole new topic.

B) Basic Linux Administration

Note: in the following guide, we don't use `sudo` prefix, taking into account you have root permissions.

Groups management

To add a new group:


```
groupadd comhype-group
```

To add a user to an existing group:

```
usermod -a -G groupname username
```

To remove a group from user's list of groups:

```
gpasswd -d username groupname
```

Location of all users (to validate the user exists):

```
cat /etc/passwd
```

Location of all groups (to validate the group is listed):

```
cat /etc/group
```

List all groups of a user ((**important** note a user must logout if groups was added in an open session):

```
groups <username>
```

User Management

To list all users you can use:

```
cut -d: -f1 /etc/passwd
```

To add a new user you can use:

```
adduser new_username
```

or:

```
useradd new_username
```

useradd is native binary compiled with the system. But, adduser is a perl script which uses useradd binary in back-end. adduser is more user friendly and interactive than its back-end useradd. There's no difference in features provided

To remove/delete a user, first you can use:

```
userdel username
```

Then you may want to delete the home directory for the deleted user account :


```
rm -r /home/username
```

(Please use with caution the above command!)

To modify the username of a user:

```
usermod -l new_username old_username
```

To change the password for a user:

```
passwd username
```

Owner and Groups

Every unix file can be a directory (d), a file (-), a socket file (s) or symbolic link (l).

When you type “ls -la” you can see the permissions / ownership:

```
drwxr-xr-x 12 eran comhype 4096 Oct 26 03:45 .
drwxr-xr-x 10 root root    4096 Oct 26 03:02 ..
drwxr-xr-x  3 eran comhype 4096 Sep 16 19:11 backup
```

- the first letter is the type (d=directory, -=file, s=socket file, l=symbolic link)
- next three letters are permissions for the “**U**ser” (r=read, w=write, x=execute)
- next three letters are permissions for the “**g**roup” (r=read, w=write, x=execute)
- next three letters are permissions for the “all **O**thers” (r=read, w=write, x=execute).
- next letter is number of links into that file.
- next field is the name of the owner
- next field is the name of the group
- next field is the last date & time that the file had been modified
- last field is the file name. Note that in linux file names are case sensitive

To change the ownership of a file:

```
chown <username> <filename>
```

Recursive change the ownership of a file – will affect all files in all directories:

```
chown -R <username> <directoryname/filename>
```

change the ownership of a group:


```
chgrp -R <groupname> <directoryname/filename>
```

change the ownership of both owner and group in one command:

```
chown -R <username>:<groupname>
```

```
<directoryname/filename>
```

To change the ownership of a symbolic link without affecting the ownership of the linked file: **(important)**

```
chown -h <username> <directoryname/filename>
```

```
chgrp -h <username> <directoryname/filename>
```

list all the permissions of a specific user (if you make changes, they will take affect only if the user logged out and logged in again):

```
id <username>
```

How to change file/folder permissions

Giving / removing user/group permissions is done using the *chmod* command:

```
chmod <options> <mode value>
```

```
<directory/filename>
```

The <mode value> is determined by which permissions you want to give:

400 read by owner

040 read by group

004 read by anybody (other)

200 write by owner

020 write by group

002 write by anybody

100 execute by owner

010 execute by group

001 execute by anybody

While you add the values to get the mode you would like. For example, to give READ and WRITE permissions to the Owner, and Read only to the group, you will write:

```
chmod 640 <directory/filename>
```

It is easier however to run the chmod command by using shortcodes like the following:

Give read, write and execute permissions to the **U**ser:

```
chmod -R u+rwX <directory/filename>
```

Remove execute permissions to the **U**ser:

```
chmod -R u-X <directory/filename>
```

You can replace the “U” with “G” for “Group” and “O” for “Others”. “-“ means “remove” and “+” give. “r” means “read”, “w” means “write” and “x” means “eXecute”.

Count the number of files in a directory (recursively)

```
sudo ls -la folder_path | wc -l
```

(note that the letter before the “wc” is vertical line |, the last letter is a small L)

To check a folder’s size:

```
sudo du -s folder_path
```


C) How to recover MYSQL database from backup

To re-create a database you should follow two steps:

Step a: Create an appropriately named database on the target machine

Step b: Load the file using the mysql command:

```
$ mysql -u [uname] -p[pass] [db_to_restore] < [backupfile.sql]
```

For example, you can restore your tut_backup.sql file to the Tutorials database:

```
$ mysql -u root -p Tutorials < tut_backup.sql
```

D) Reset the MYSQL server root password

The following process (taken from the following online article:

<http://www.debian-administration.org/articles/442>

Resetting the root password of a MySQL database is trivial if you know the current password if you don't it is a little trickier. Thankfully it isn't too difficult to fix, and here we'll show one possible way of doing so.

If you've got access to the root account already, because you know the password, you can change it easily:

```
eran@backups:~$ mysql --user=root --pass mysql
```

Enter password:

```
mysql> update user set Password=PASSWORD('new-password-here') WHERE  
User='root';
```

Query OK, 2 rows affected (0.04 sec)

Rows matched: 2 Changed: 2 Warnings: 0

```
mysql> flush privileges;
```

Query OK, 0 rows affected (0.02 sec)


```
mysql> exit
```

```
Bye
```

However if you don't know the current password this approach will not work - you need to login to run any commands and without the password you'll not be able to login!

Thankfully there is a simple solution to this problem, we just need to start MySQL with a flag to tell it to ignore any username/password restrictions which might be in place. Once that is done you can successfully update the stored details.

First of all you will need to ensure that your database is stopped:

```
root@backups:~# /etc/init.d/mysql stop
```

Now you should start up the database in the background, via the `mysqld_safe` command:

```
root@eran:~# /usr/bin/mysqld_safe --skip-grant-tables &
```

```
[1] 6702
```

```
Starting mysqld daemon with databases from /var/lib/mysql
```

```
mysqld_safe[6763]: started
```

Here you can see the new job (number "1") has started and the server is running with the process ID (PID) of 6702.

Now that the server is running with the `--skip-grant-tables` flag you can connect to it without a password and complete the job:

```
root@eran:~$ mysql --user=root mysql
```

```
Enter password:
```

```
mysql> update user set Password=PASSWORD('new-password-here') WHERE  
User='root';
```

```
Query OK, 2 rows affected (0.04 sec)
```

```
Rows matched: 2  Changed: 2  Warnings: 0
```

```
mysql> flush privileges;
```

```
Query OK, 0 rows affected (0.02 sec)
```



```
mysql> exit
```

```
Bye
```

Now that you've done that you just need to stop the server, so that you can go back to running a secure MySQL server with password restrictions in place. First of all bring the server you started into the foreground by typing "fg", then kill it by pressing "Ctrl+c" afterwards.

This will now allow you to start the server:

```
root@eran:~# /etc/init.d/mysql start
```

```
Starting MySQL database server: mysqld.
```

```
Checking for corrupt, not cleanly closed and upgrade needing tables..
```

Now everything should be done and you should have regained access to your MySQL database(s); you should verify this by connecting with your new password:

```
root@eran:~# mysql --user=root --pass=new-password-here
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 5 to server version: 5.0.24a-Debian_4-log
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> exit
```

```
Bye
```