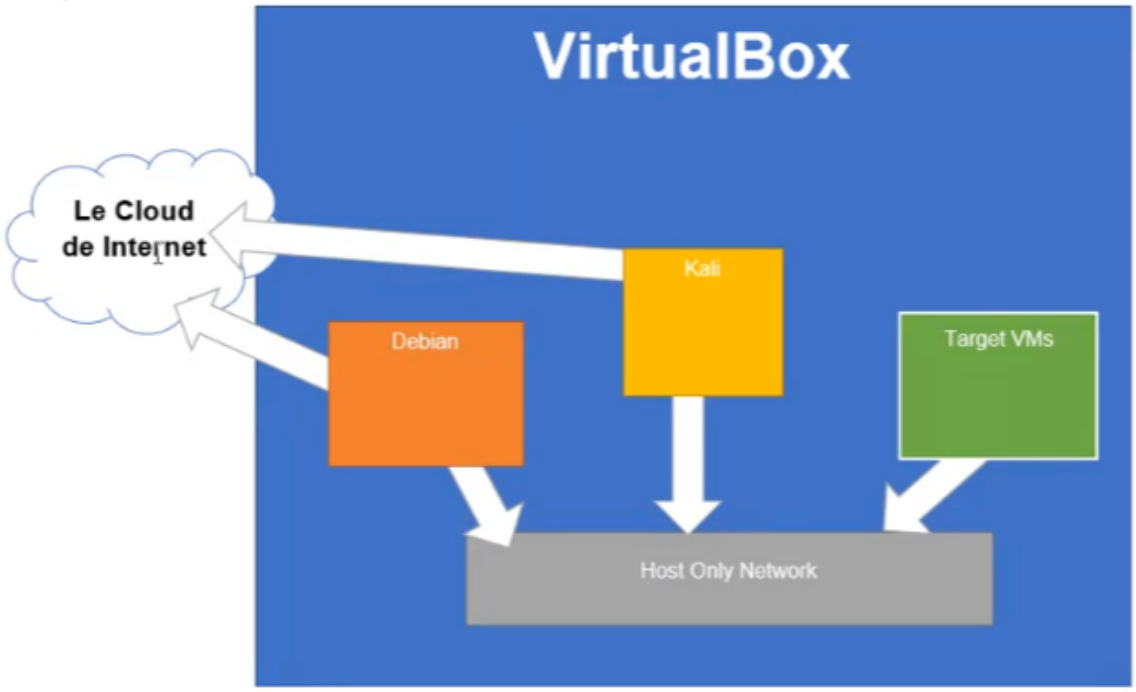
Download the following files:

SQL Manager:
https://www.sqlmanager.net/tools/free
-> For MySQL (https://www.sqlmanager.net/products/mysql/manager/download/128)

VirtualBox VM Client:
https://www.virtualbox.org/wiki/Downloads
-> Windows hosts (https://download.virtualbox.org/virtualbox/6.1.30/VirtualBox-6.1.30-148432-Win.exe)

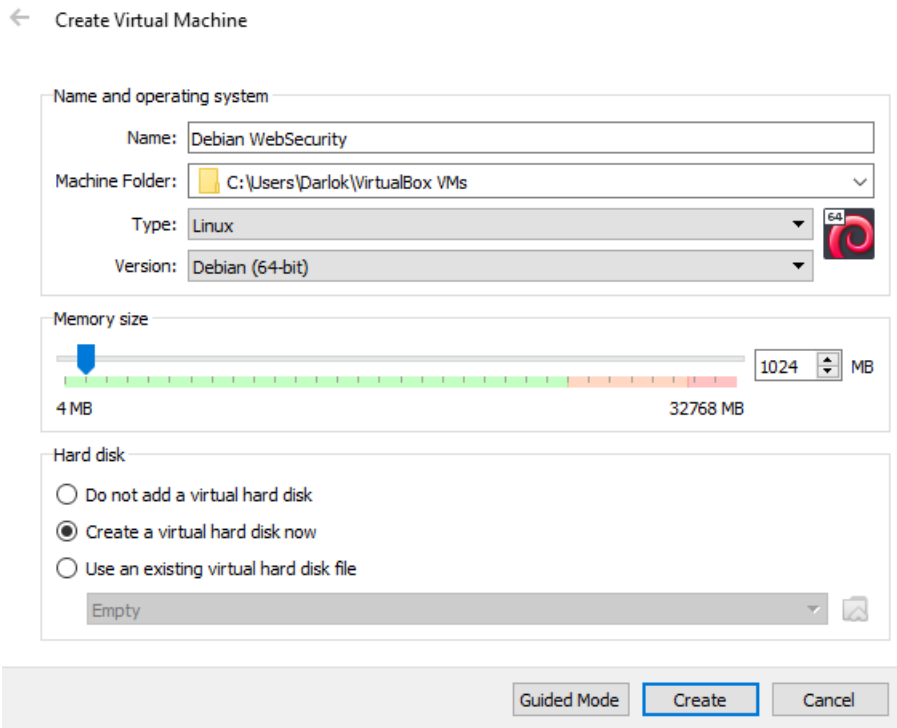
Linux OS – Debian Distribution:
https://www.debian.org/distrib/
-> Smaller net install ISO is fine: https://www.debian.org/distrib/netinst
(Choose your processor architecture link from there)

Kali (Linux) OS:
https://cdimage.kali.org/kali-2021.4a/kali-linux-2021.4a-installer-amd64.iso



Install VirtualBox and load it up!

Click “New” or use: Ctrl-N



Enter a “Name”, I used “Debian WebSecurity”.

Choose “Linux” as the Type and “Debian (64-bit)” as your version.

Troubleshooting issues: Not seeing a64 bit version??

https://superuser.com/questions/866962/why-does-virtualbox-only-have-32-bit-option-no-64-bit-option-on-windows-7

Hardware virtualization is enabled in the BIOS. (Your CPU must support it.)

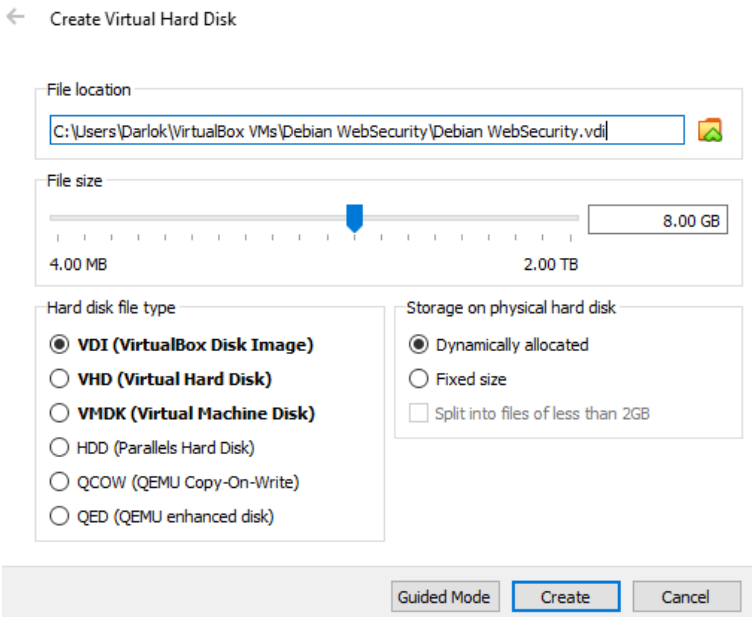
For Intel x64: VT-x (Intel Virtualization Technology) and VT-d are both enabled

For AMD x64: AMD SVM (Secure Virtual Machine) is enabled

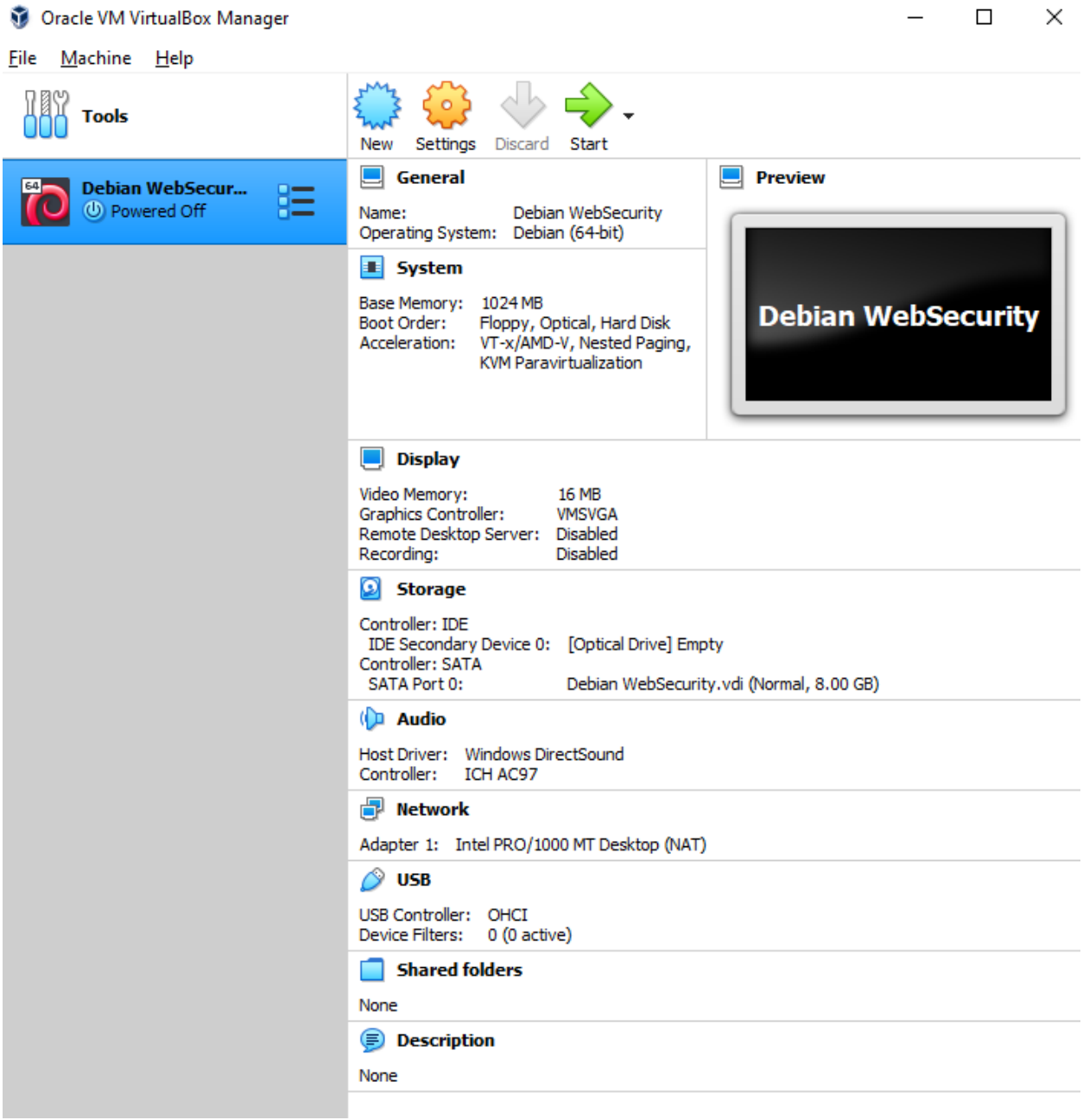
We can keep the default values of memory to 1024MB and Create a virtual hard disk now.

Click “Create”

On the following screen leave the default options as is and click “Create” again.



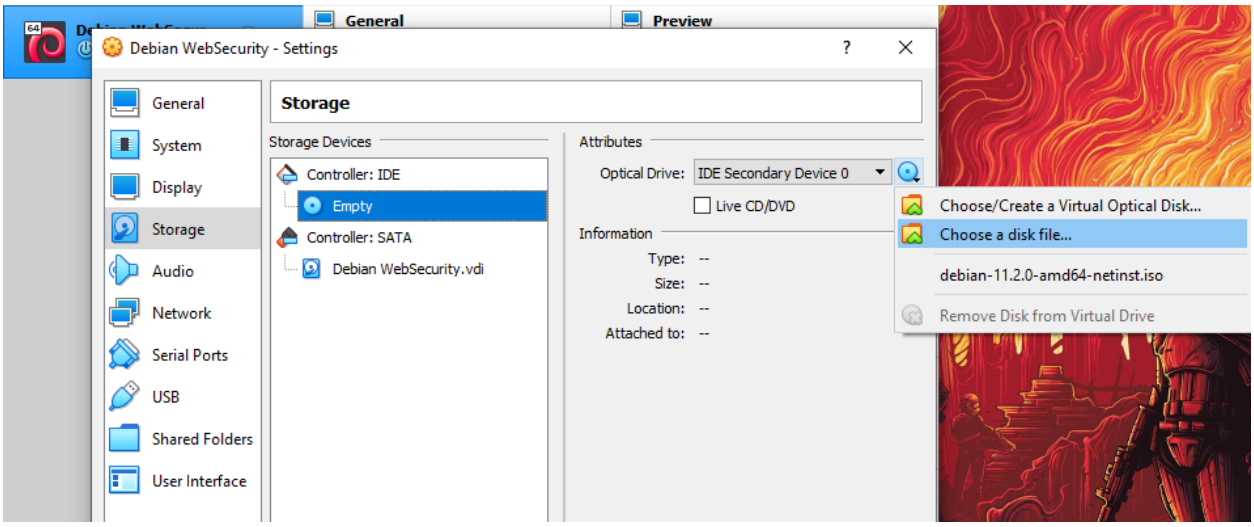
You should now have this:



We need to edit the settings to add the Debian ISO.

Right-click on Debian and go to: **Settings**…

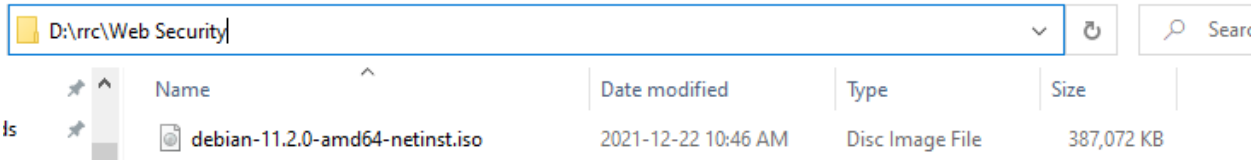
(Alternatively: click on the **Storage** title on the right hand side)



Click on the “Empty” Controller: IDE

Use the icon on the right of “Optical Drive” to select “Choose a disk file…”

Find and select the Debian ISO file you downloaded earlier

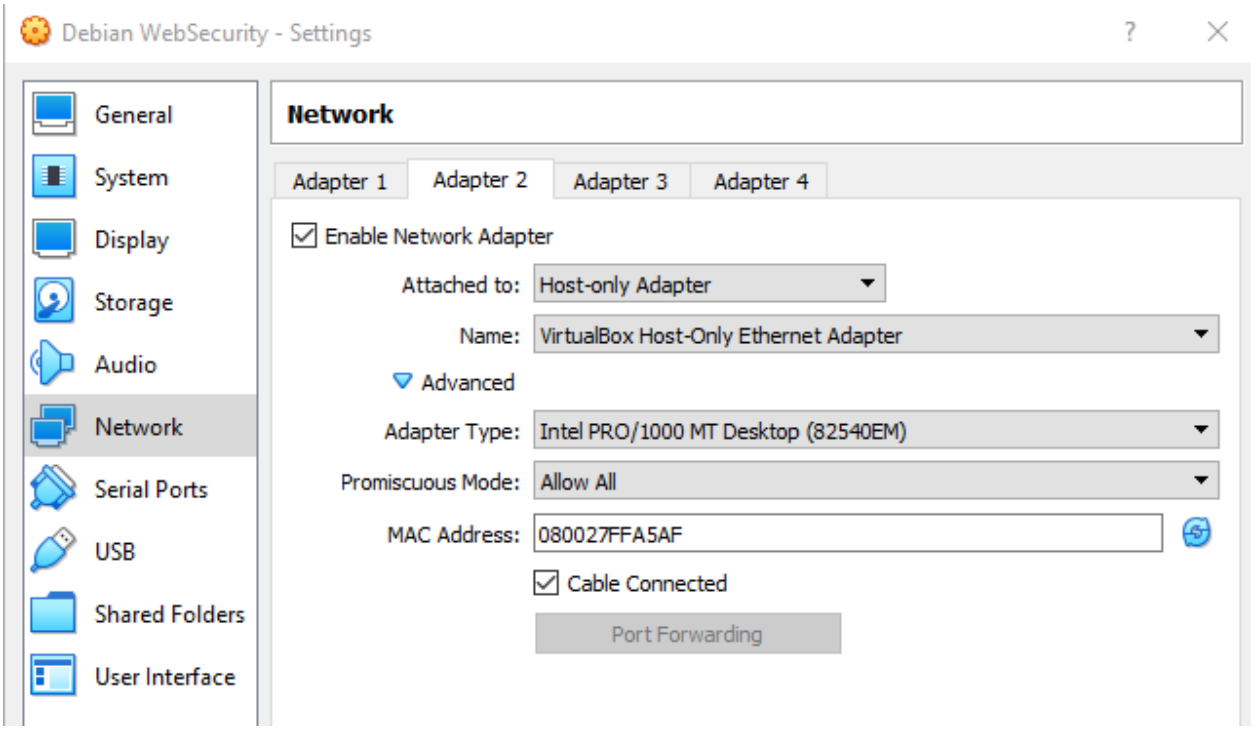


Lastly, switch to “Network” in the settings and click on “Adapter 2”

Click “Enable Network Adapter”, attach to: “Host-only Adapter”

Choose the VirtualBox Host-Only Ethernet Adapter from the list (should only be 1)

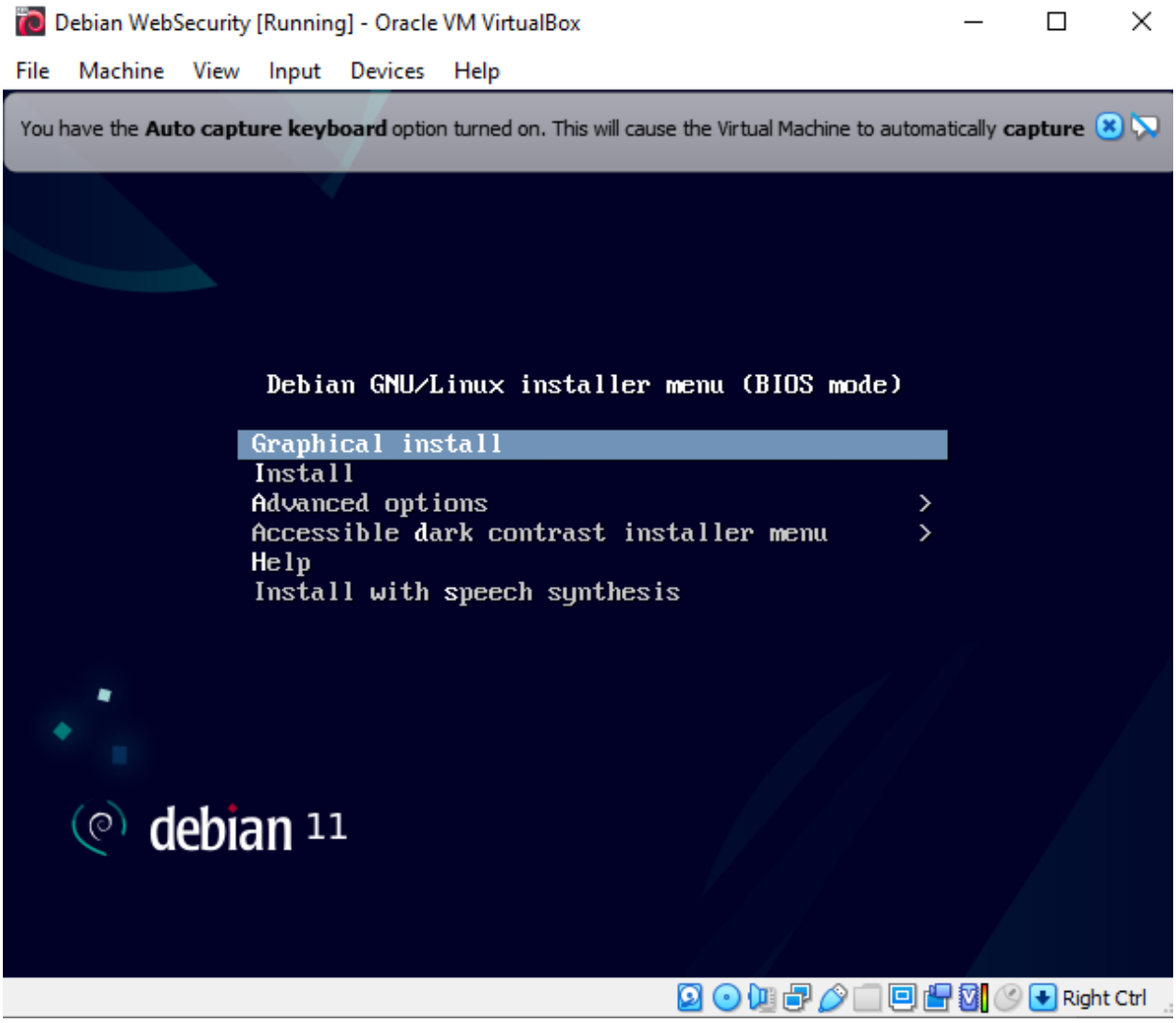
Change Promiscuous Mode to “Allow All”



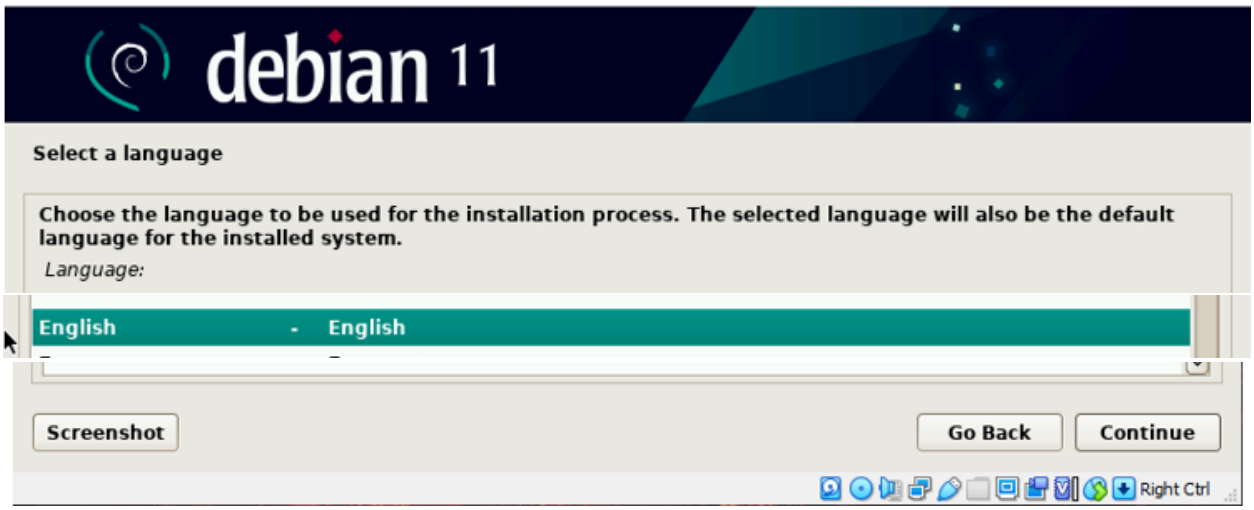
Click “OK” and we are done with the settings! Let’s launch the VM now.

Double clicking the Debian VM on the left will launch it. If it asks you, you may have to find the ISO again from a list by adding it again. If it doesn't ignore this comment.

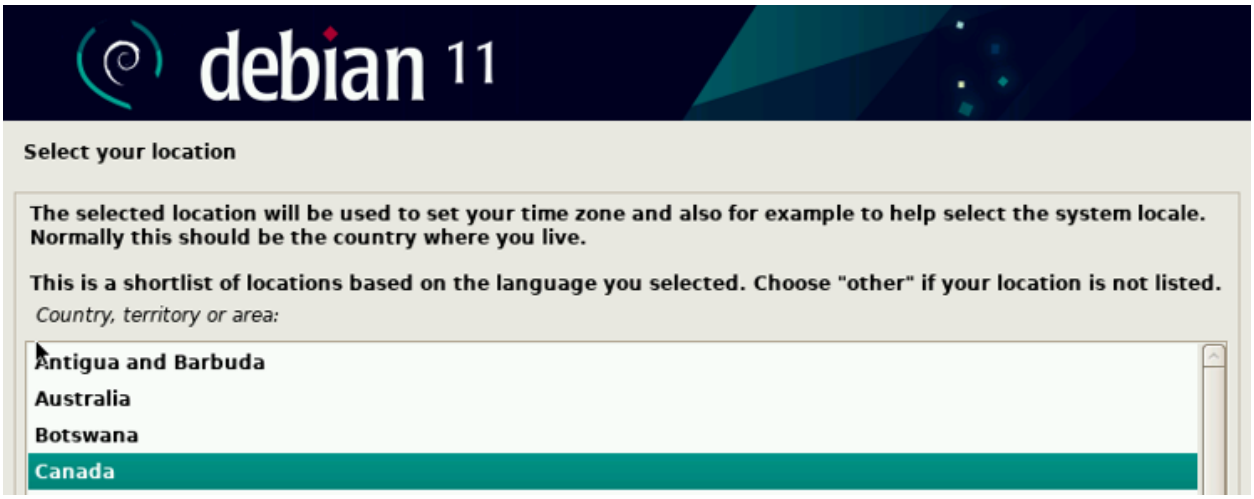
It's good if it looks like this:



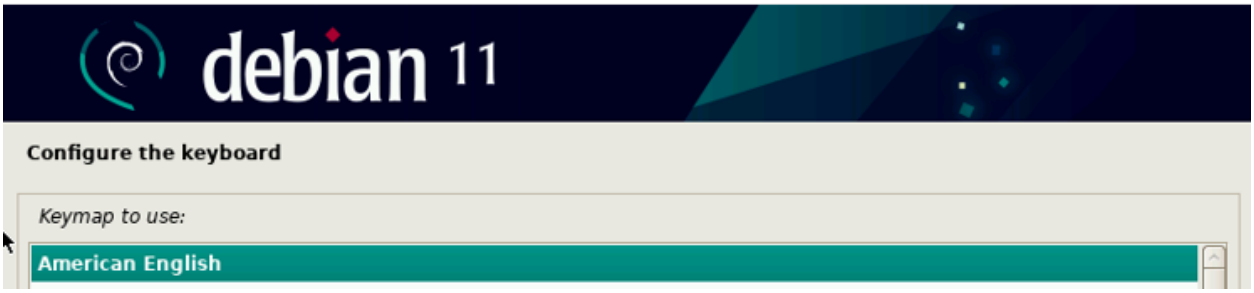
Use the keyboard up and down arrows to navigate to "Graphical Install" and hit enter.



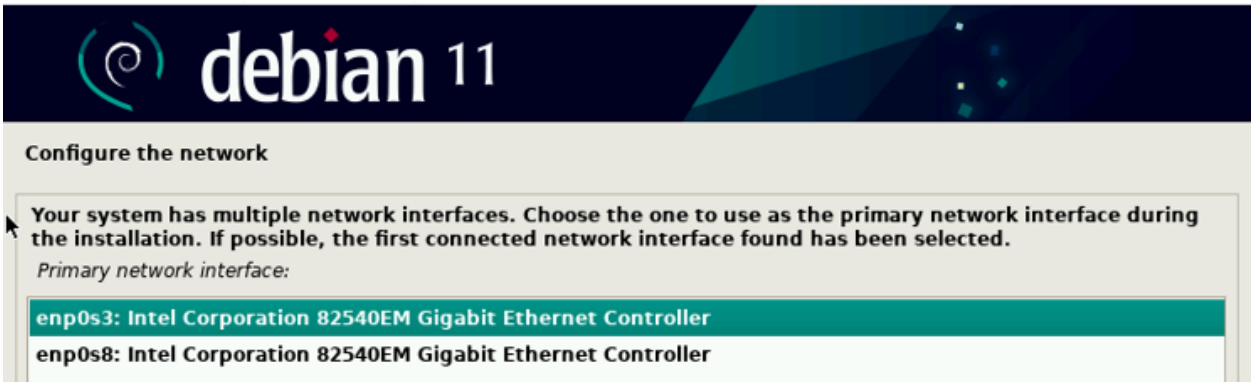
Choose English and click “Continue”



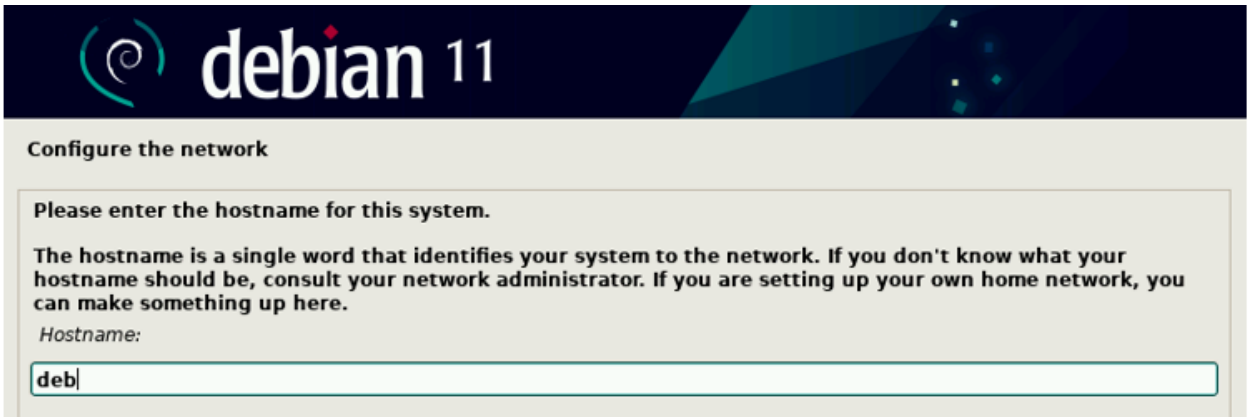
Choose “Canada” and click “Continue”



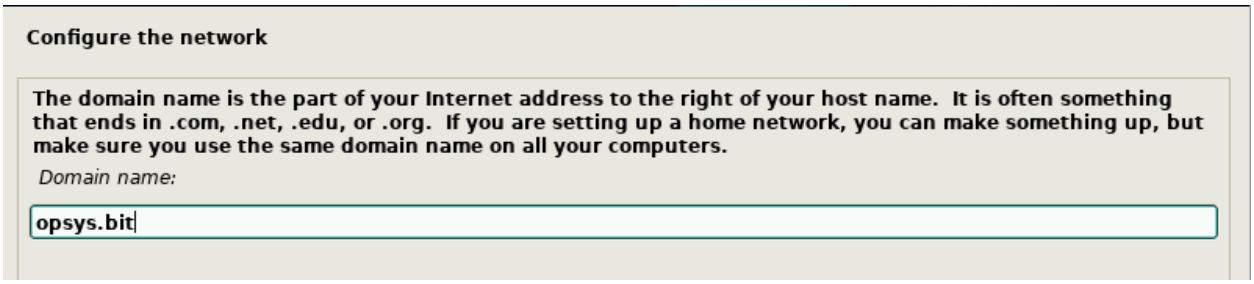
Choose “American English” and click “Continue”



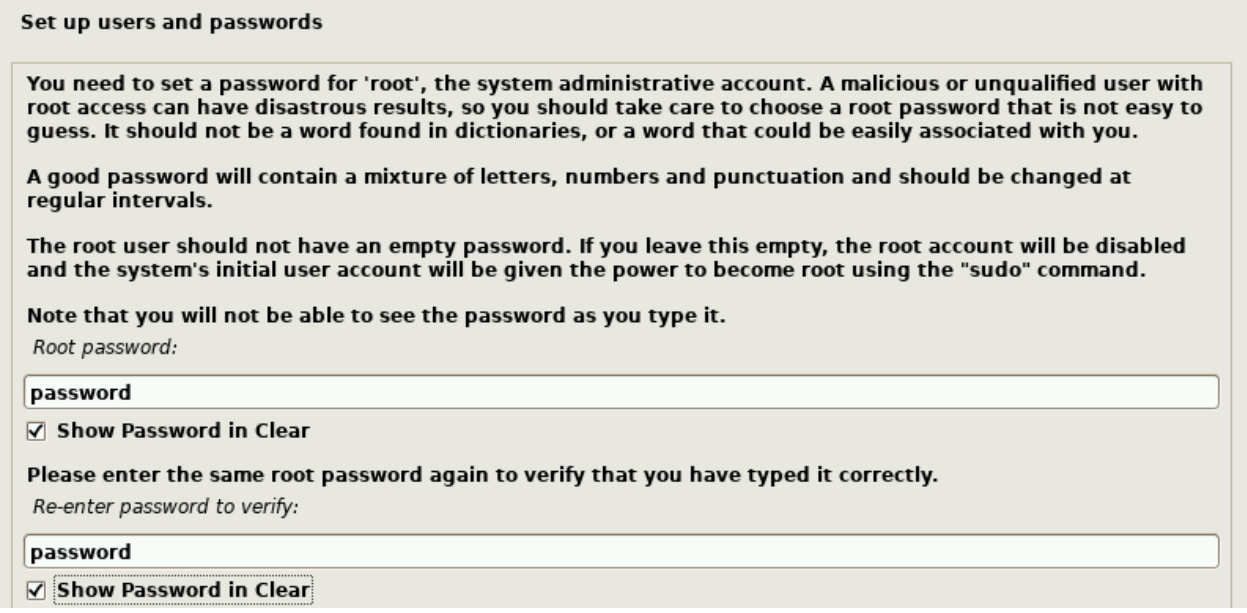
“Choose the network that has the smaller number first (in the screenshot: **enp0s3**)



Hostname can be anything, but smaller is nice for later so: **deb**

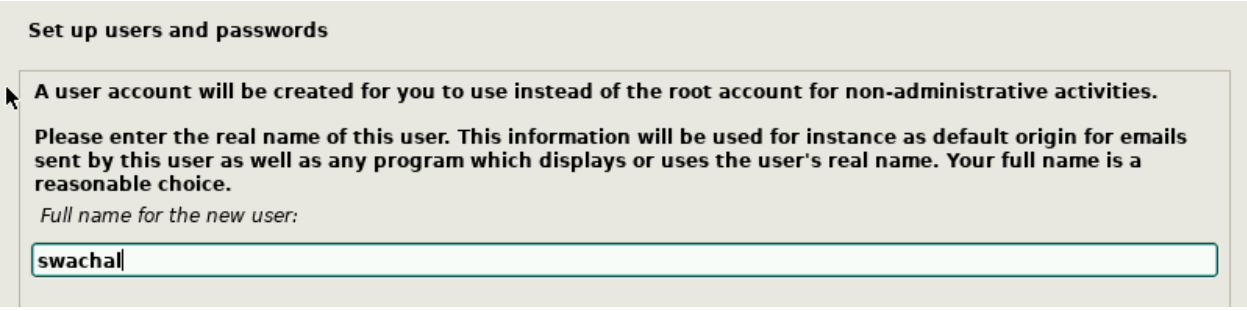


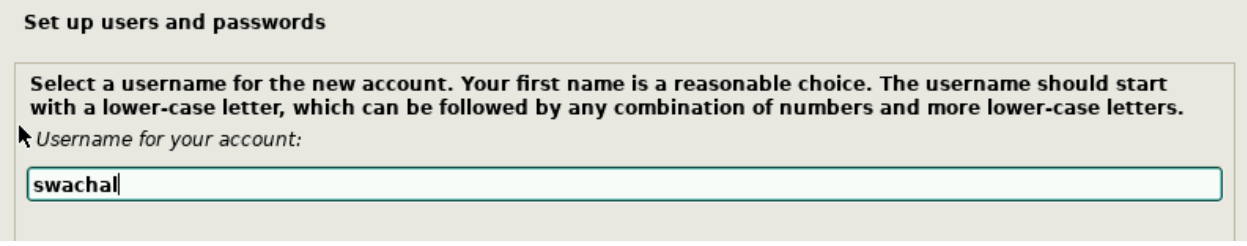
Domain name can also be anything, but we’ll use: **opsys.bit**



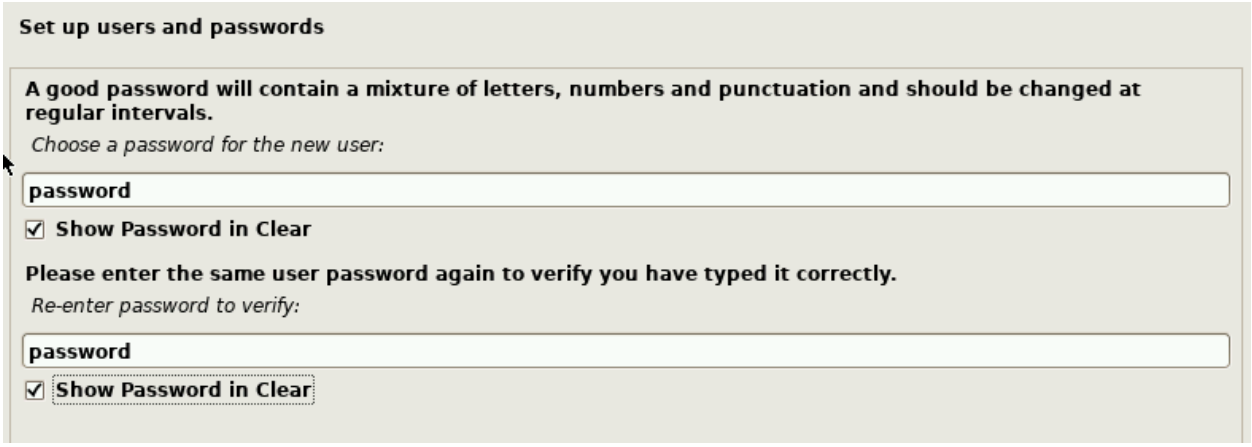
Set the password to: **password**

Yes, all of the irony of using this password for this class. 😊 It's ok. It's for school and we don't need the issue of forgetting a password here.

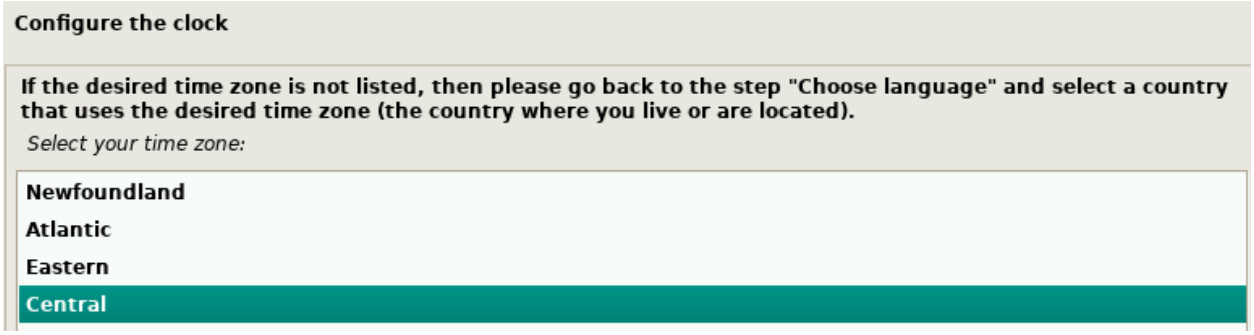




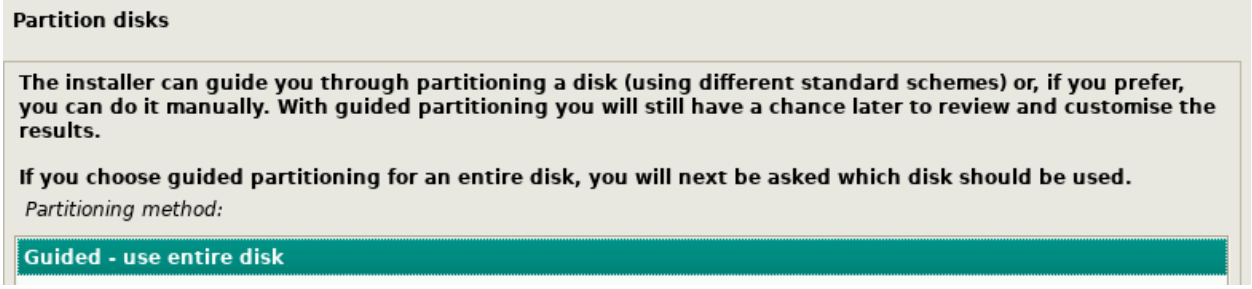
I am using the same name/username for the next two prompts (my first initial and last name). You can use anything you want but keep it small and memorable!



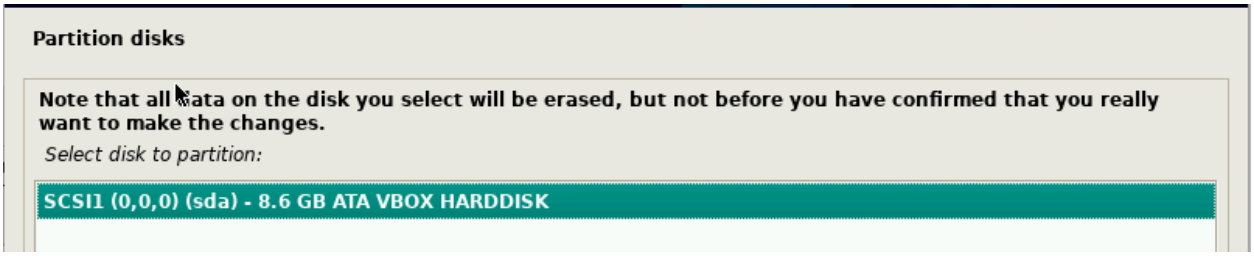
Again for the password on your account, use something you will never forget for school purposes.



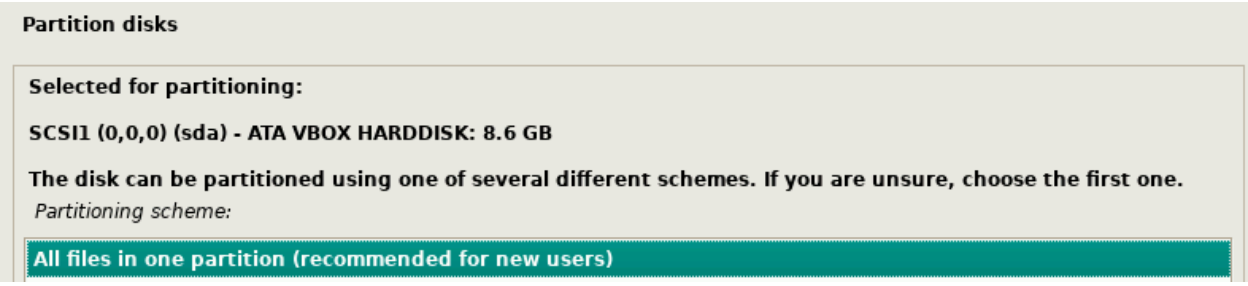
Be sure to pick Central timezone if you're in Winnipeg!



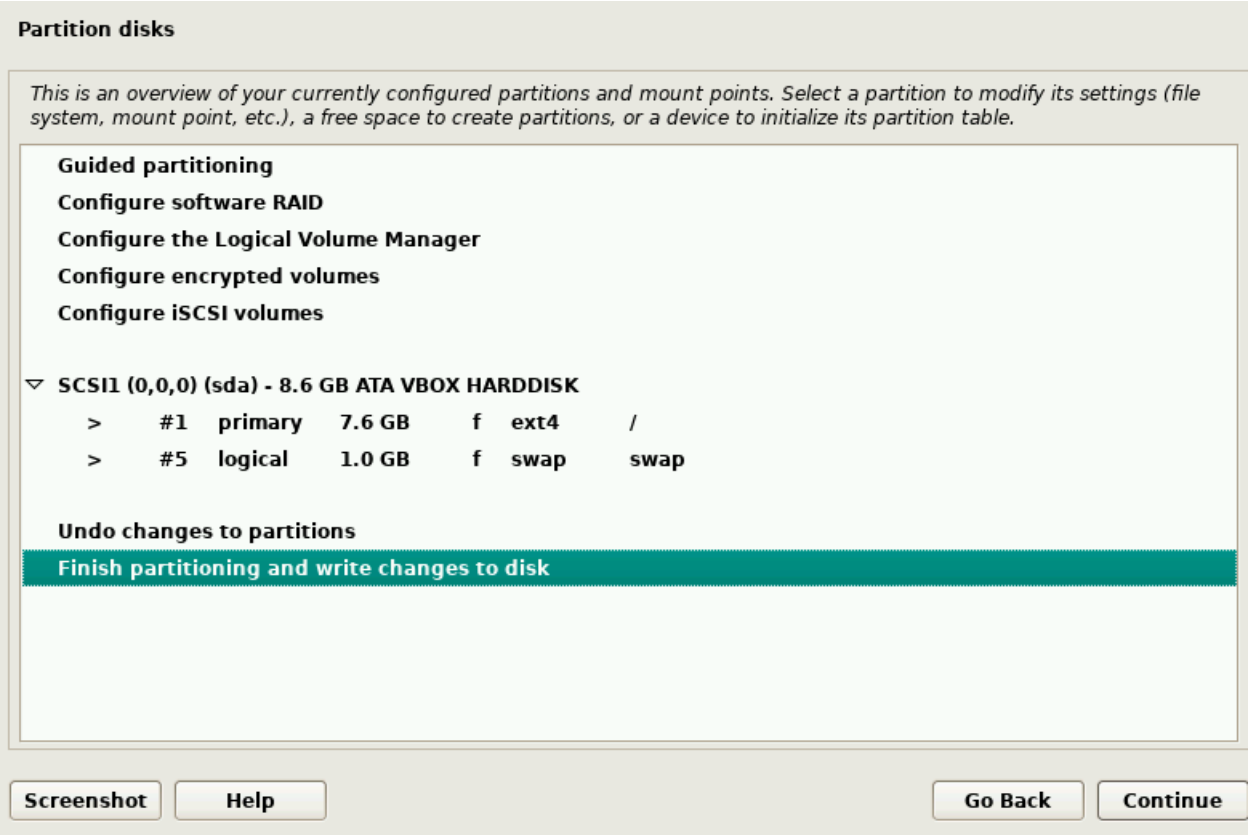
Select "Guided – use entire disk"



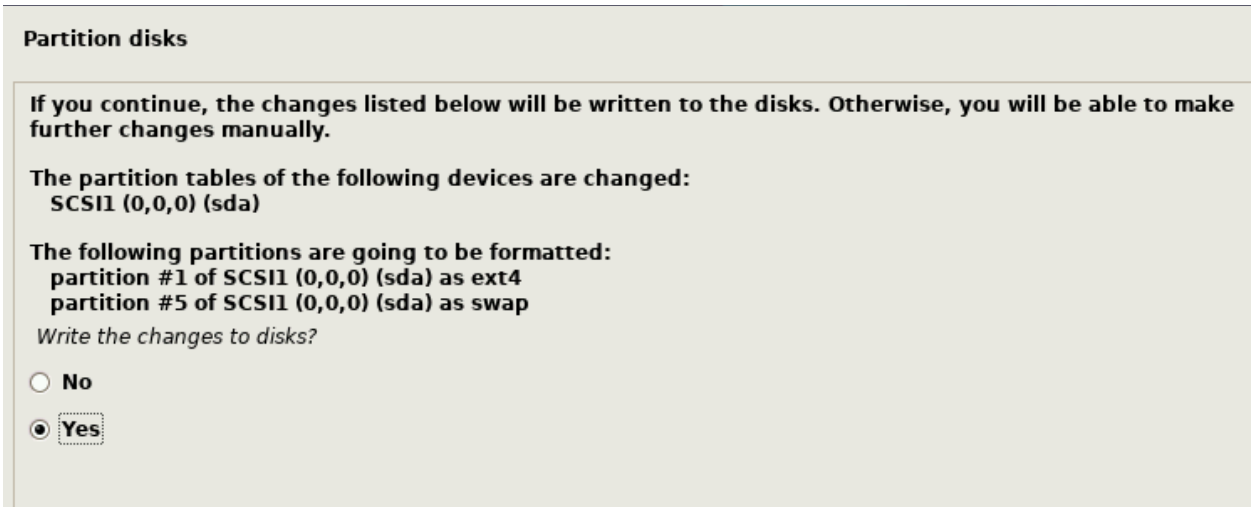
Choose the drive that shows up



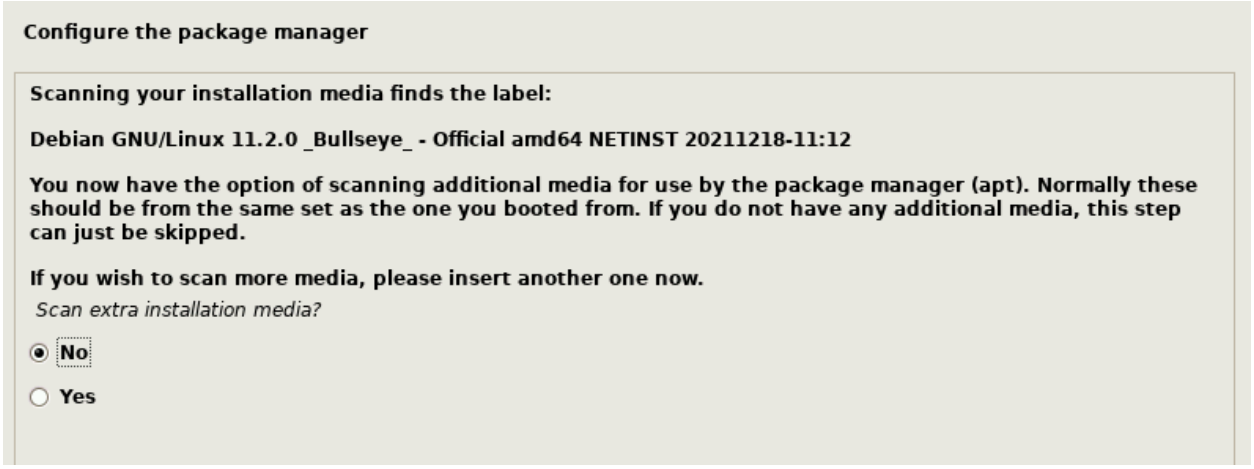
Choose all files in one partition



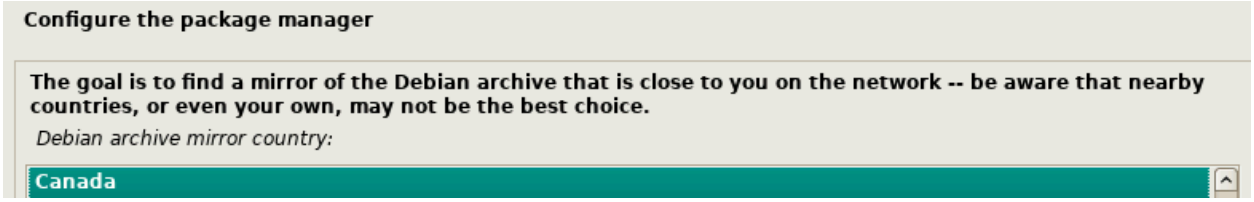
Click continue!



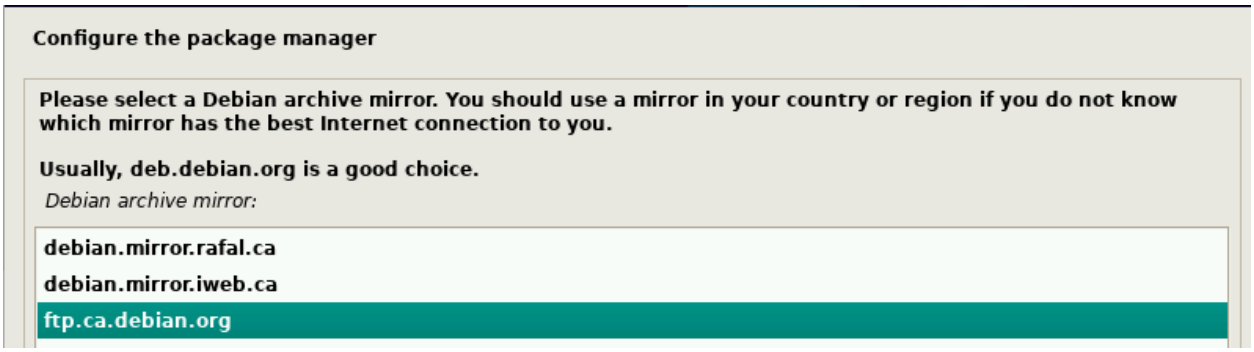
Click yes and then continue.



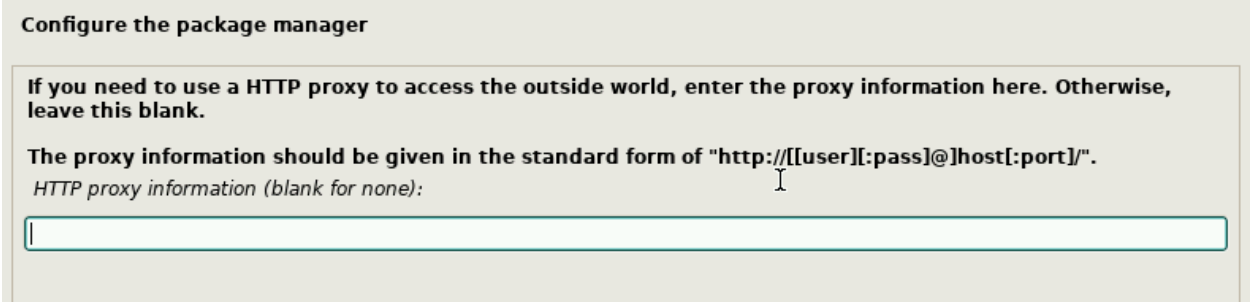
Keep the default of “No” and click continue



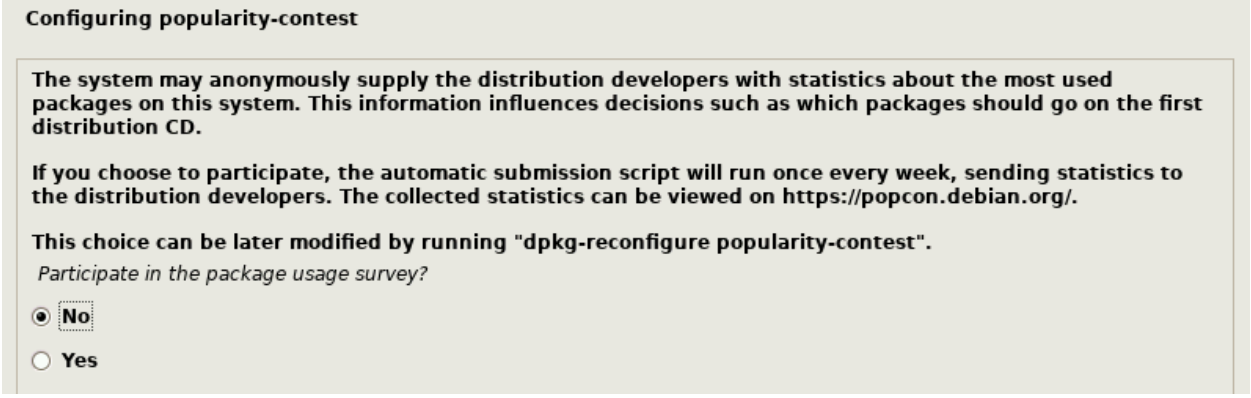
Choose Canada and click continue



Select the ftp.ca.debian.org for your package manager

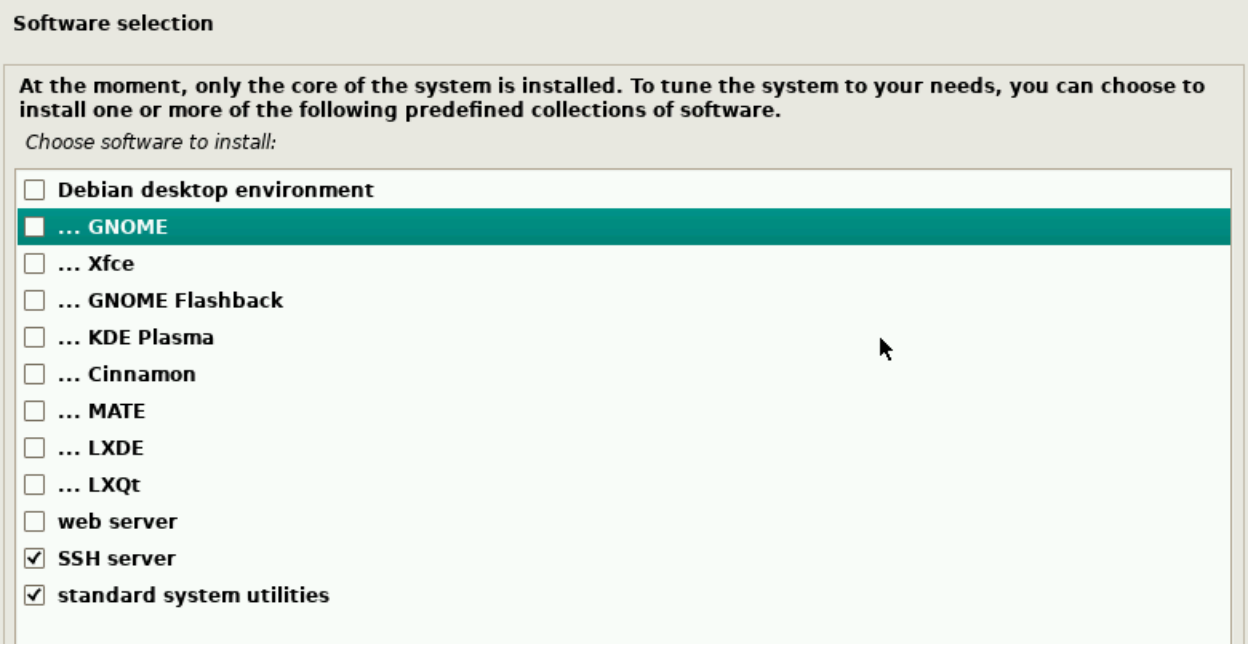


You should probably keep this blank. (If you really do know what you're doing here and have a lot of unique network settings, we'll trust you to adjust this setting).



You can leave it on "No" here. We aren't using Debian in a typical way, so providing them feedback may skew their results.

One must not add unnecessary server services. As we will see in pen testing, one out of date or compromised service can lead to a compromised server. As such we only install the necessary components:

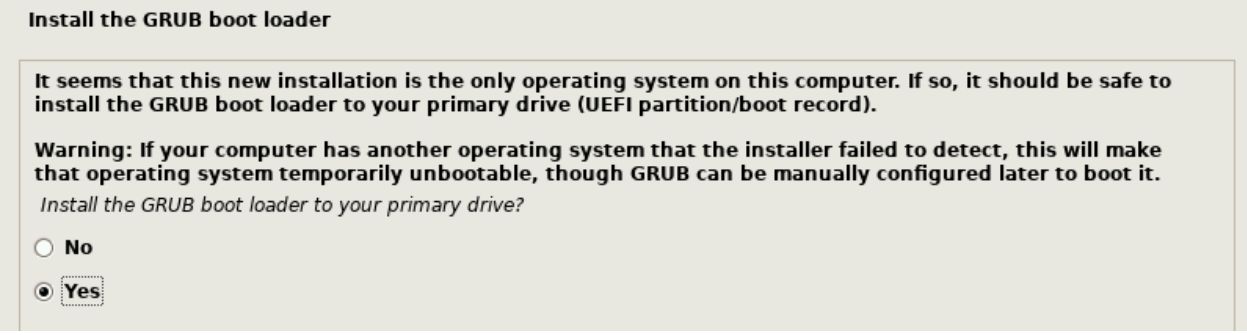


Only check the last two options: (We will install the web server on our own terms)

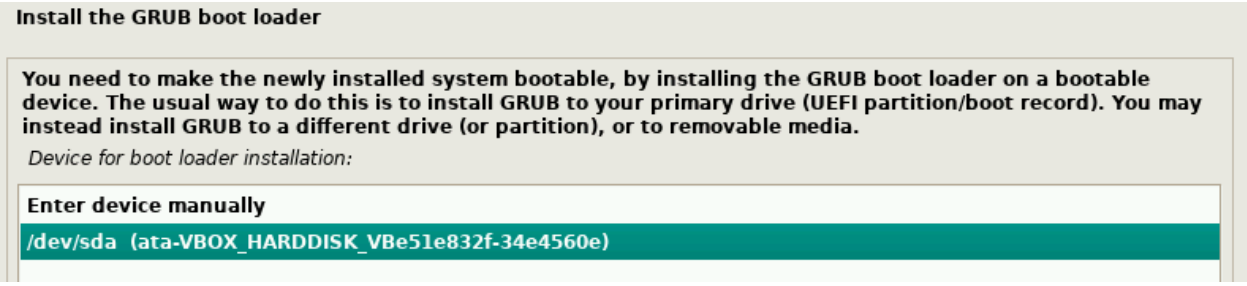
- SSH Server
- Standard system utilities

Three components for loading the OS:
BIOS (UEFI) on your motherboard, Kernel (OS; Debian) and Boot Loader software

GRUB is the boot loader:

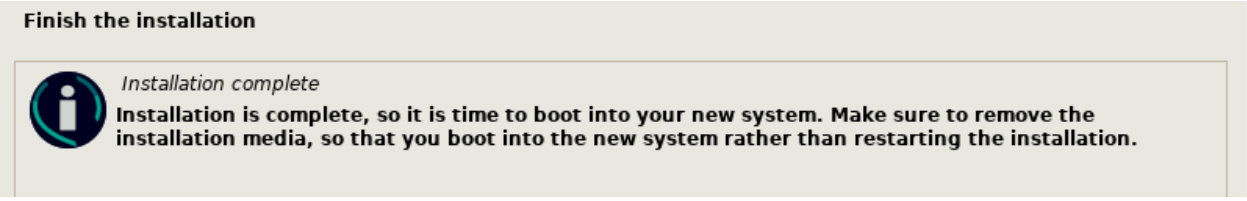


Click Yes.

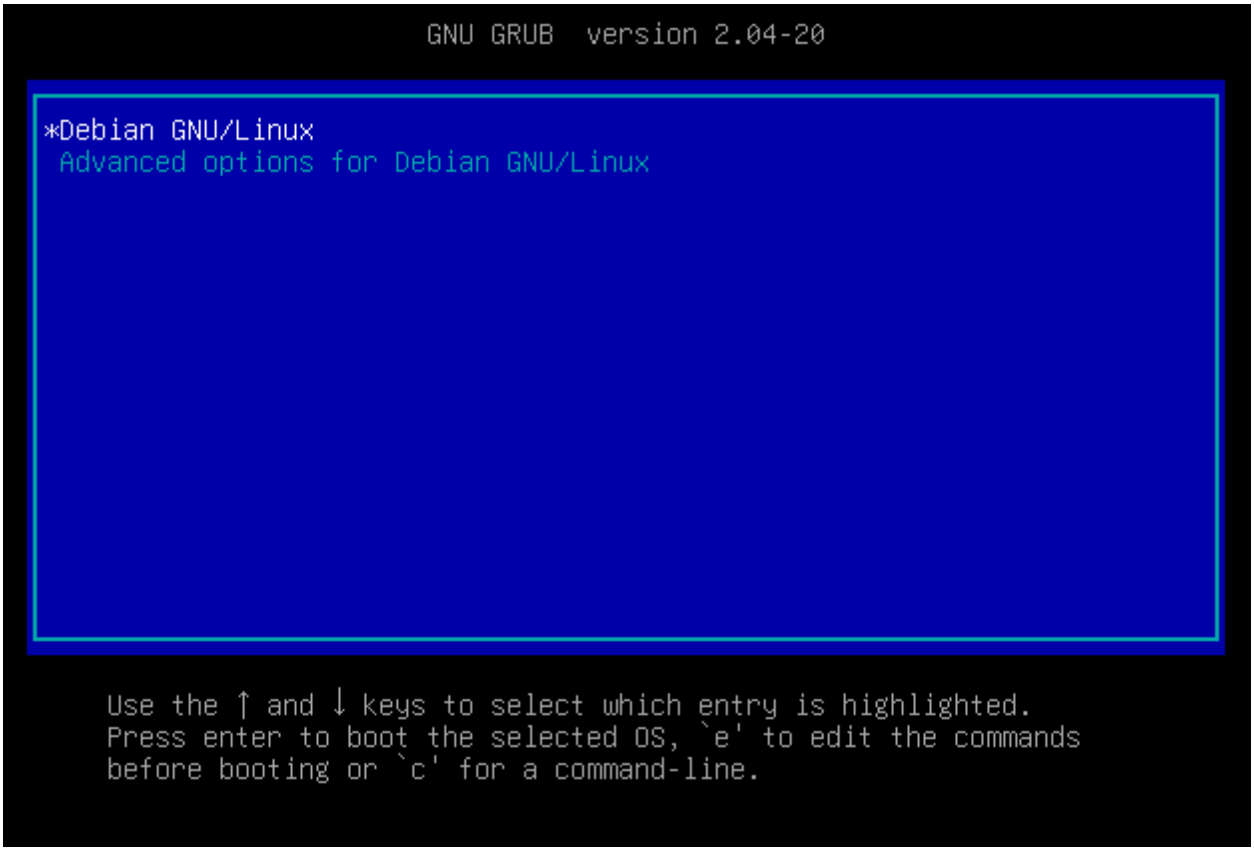


Pick the device (should be similar to the above screenshot).

If everything went well, you should see this (click ok after).



Note: the VM will have ejected our digital optical drive disc (the Debian iso file).



Hit enter on Debian GNU/Linux

```
Debian GNU/Linux 11 deb tty1

deb login: swachal
Password:
Linux deb 5.10.0-10-amd64 #1 SMP Debian 5.10.84-1 (2021-12-08) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
swachal@deb:~$ su -
Password:
root@deb:~# _
```

Login with your username/password setup from before. (Mine was: **swachal** / **password**)

Note: you could also choose to login as **root** directly

To switch to root from your normal account, type out the command:
su –

It will prompt you for the root password, I set this as:
password

Type out:
ip addr

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo
000
    link/ether 08:00:27:88:f3:2c brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
```

This adapter gets an IP address. It's pretty much always this 10.x.x.x ip here.

```
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:ff:a5:af brd ff:ff:ff:ff:ff:ff
```

This 2nd adapter does not have an IP, but it has a MAC address. We need it to have one!

cat /etc/network/interfaces

The term **cat** is short for concatenate. It joins files together. If we do not provide a 2nd file as a parameter, it will output the first parameter file name to the screen.

```
root@deb:~# 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 enp0s3
iface enp0s3 inet dhcp
```

Aside:

Often you will want to make sure you're logged in as the root before performing certain operations in linux. To do this quickly, type out:
whoami

It should say: **root**

As the **root user**, type out the following to load a free text editor called "nano". We're going to add the 2nd network adaptor to the **/etc/network/interfaces** file.

nano /etc/network/interfaces

```
  GNU nano 5.4                    /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 enp0s3
iface enp0s3 inet dhcp
```

```
  GNU nano 5.4                          /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 enp0s3
iface enp0s3 inet dhcp

# add host only network interface - sfw
allow-hotplug enp0s8
iface enp0s8 inet dhcp
File Name to Write: /etc/network/interfaces
```

Using the down arrow, navigate to the bottom of the file and type out:

```
# add host only network interface -YOURINITIALS
allow-hotplug enp0s8
iface enp0s8 inet dhcp
```

Use **ctrl-O** to save, then **hit enter** to confirm the file name, then use **ctrl-X** to exit out of the editor.

Now let's get the network adapter to get an IP address, type out:

```
ifup enp0s8
```

it should assign an IP in the output, but you can also just type out: **`ip addr`** to check at any time…

```
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen
000
    link/ether 08:00:27:ff:a5:af brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic enp0s8
       valid_lft 582sec preferred_lft 582sec
    inet6 fe80::a00:27ff:feff:a5af/64 scope link
       valid_lft forever preferred_lft forever
```

You can see above I got: 192.168.56.101. Yours will be similar but likely different on the last numbers.

Setting up Debian for a Secure Web Server and a Database

Now that you have Debian up and running, you want to set it up as a **LAMP** stack, specifically using **Apache2**, **PHP7**, and **MariaDB Server**. The following will walk you through that.

Update your Debian VM. It is unlikely to be necessary after a first install but it is always a good idea when installing software. First, ensure you execute these commands as **root:**

```
apt update
apt upgrade
```

Web servers are applications that run on machines that allow webpages to be shared/presented. One of the most common web servers is the Apache Web Server (www.apache.org)

Execute the following commands:

```
apt install apache2
apt install lynx
```

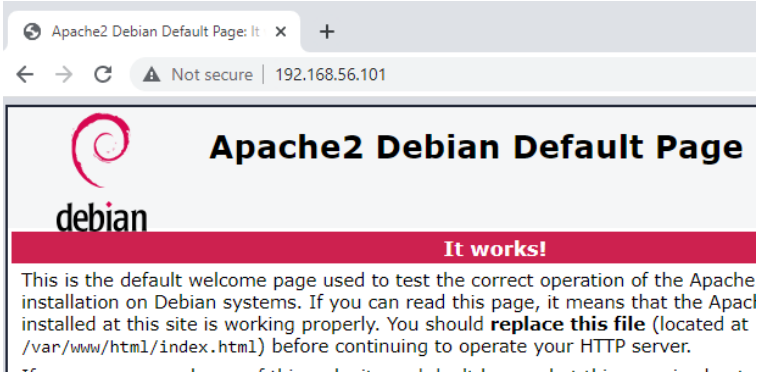
Test if apache is running by running:

```
service apache2 status
```

```
root@deb:~# service apache2 status
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2022-01-09 22:52:25 CST; 15min ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 1143 (apache2)
      Tasks: 55 (limit: 1133)
     Memory: 9.0M
        CPU: 36ms
     CGroup: /system.slice/apache2.service
             ├─1143 /usr/sbin/apache2 -k start
             ├─1145 /usr/sbin/apache2 -k start
             └─1146 /usr/sbin/apache2 -k start

Jan 09 22:52:25 deb systemd[1]: Starting The Apache HTTP Server...
Jan 09 22:52:25 deb systemd[1]: Started The Apache HTTP Server.
```

You can also go to your Windows and browse to your Debian server.

This will require you to determine the IP address of your Debian server, which you should now know how to do (**`ip addr`**).



You may also launch Lynx command line browser in your Debian VM and test your Apache install by typing:

```
lynx localhost
```

Lynx will show that Apache is working, and remove the network from the equation as a potential source of problems.

You may now install the remainder of utilities.

```
apt install php
apt install mariadb-server
apt install php-mysqli
systemctl restart apache2
```

Database Access

Rather than deploying a package like **PHPMyAdmin** to your web server for database management, we are going to create a remote database connection and database users that allows remote management. First thing, we need to modify our database server to allow this connection.

We need to log in as **root** to make the following changes. Modify the server config as below:

```
nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

You need to change the address you support to something that makes sense.

```
bind-address                                    = 0.0.0.0
```

This can introduce a vulnerability that allows anyone to connect, something a secure server doesn't need. We will address this with network connections in the real world, blocking traffic at the router level, and is out of scope to this course. We will help with this by allowing specific access to databases via the built in database management tools.

This allows both localhost access and our host only network. Should look like the following:

```
  GNU nano 3.2                              50-server.cnf

#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
#
# See the examples of server my.cnf files in /usr/share/mysql

# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mysqld standalone daemon
[mysqld]

#
# * Basic Settings
#
user                    = mysql
pid-file                = /run/mysqld/mysqld.pid
socket                  = /run/mysqld/mysqld.sock
#port                   = 3306
basedir                 = /usr
datadir                 = /var/lib/mysql
tmpdir                  = /tmp
lc-messages-dir         = /usr/share/mysql
#skip-external-locking

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0_

#
# * Fine Tuning
#
                                        [ Wrote 133 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line M-E Redo
```

Ctrl + O to save (write out) and Ctrl + X to exit. Left control key.

Now restart MariaDB server service with the following:

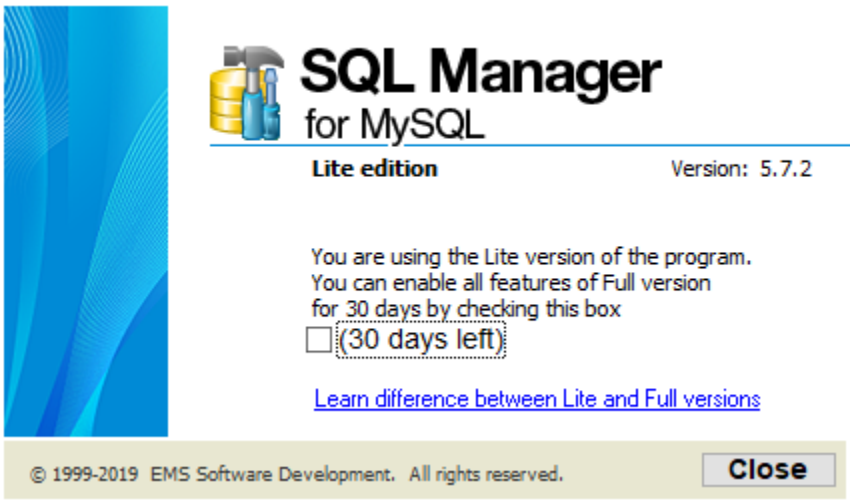
```
systemctl restart mysql.service
systemctl restart mariadb.service
```

We now need to install a database management tool to replace web-based tools like PHPMyAdmin. There are many, however, consider the free version of tools from SQLManager for MySQL, which will work for MariaDB. Link can be found in Learn, but is here as well:

https://www.sqlmanager.net/en/tools/free

Defaults for installation and initial launch should be fine, but for the first month of use, every time you will be prompted to try the full version. **I strongly recommend you always say No, as you cannot go back to the free one, and will have to pay after the trial is over, or try something else.**

Just click **Close** to the dialog box below



Now we need to create a database to connect to with SQLManager. Go back to Debian and type the following (again, logged in as **root**):

```
mysql -u root
```

At this point, you can create a database for the DVWA application, as we have seen before with the following:

```
create database dvwa;
```

Next create two different users to access the database, one with network access (dvwadmin) and one with local access that will be configured in the config files for DVWA. You may substitute your own username and passwords as you see fit, but you should remember what you did.

```
grant all privileges on dvwa.* to 'dvwadmin'@'%' identified by
'password' with grant option;
```

```
grant all privileges on dvwa.* to 'dvwa'@'localhost' identified by
'password' with grant option;
```

Finally, flush privileges to apply the permissions:

```
flush privileges;
```

(this forces the privileges to be active immediately)
Your display should look like this:

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0

#
# * Fine Tuning
#


root@debian:/etc/mysql/mariadb.conf.d# systemctl restart mysql.service
root@debian:/etc/mysql/mariadb.conf.d# systemctl restart mariadb.service
root@debian:/etc/mysql/mariadb.conf.d# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.3.15-MariaDB-1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> create database dvwa;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> grant all privileges on dvwa.* to 'dvwadmin'@'%' identified by 'password' with gra
nt option;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> grant all privileges on dvwa.* to 'dvwa'@'localhost' identified by 'password' with
 grant option;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> exit
Bye
root@debian:/etc/mysql/mariadb.conf.d#
```

Next, review and verify your network is configured properly for the host only network. For v10 of Debian, you should ensure both enp0s3 and enp0s8 are configured for DHCP as below.

You will need to type in the following:

```
ifup enp0s8
```

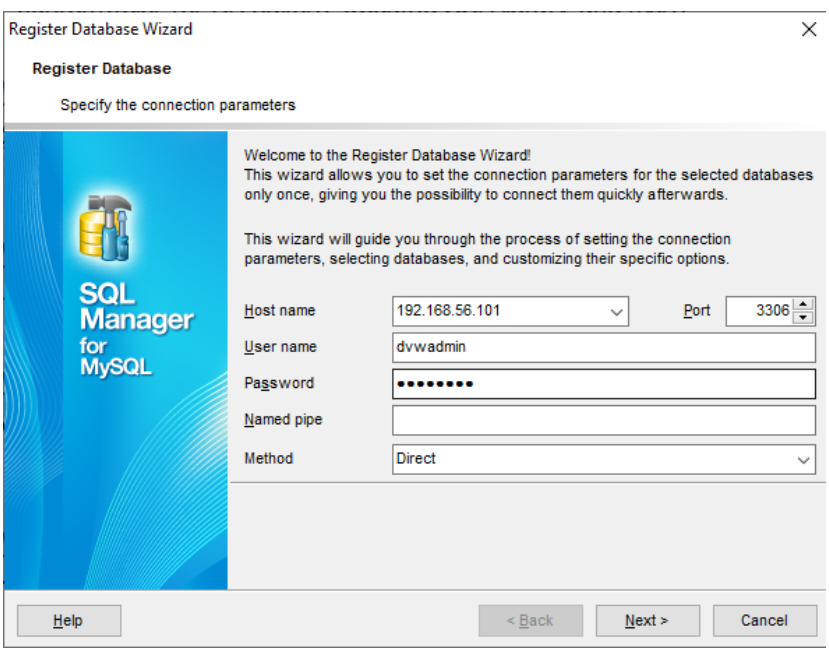
You can then grab the IP with:

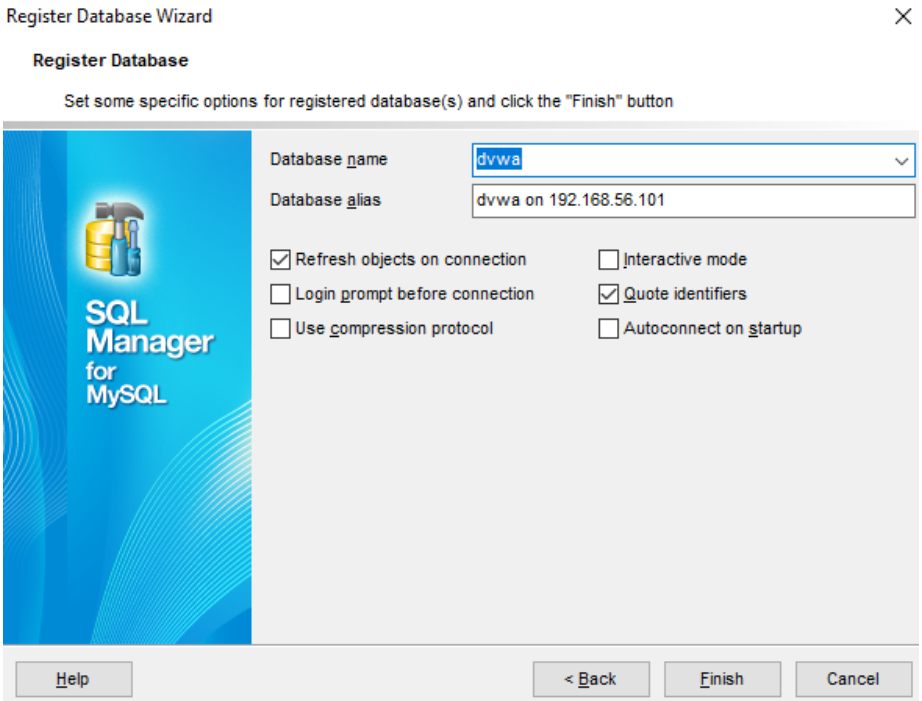
```
ip addr
```

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

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet dhcp

# The primary network interface
allow-hotplug enp0s8
iface enp0s8 inet dhcp
root@debian:/etc/mysql/mariadb.conf.d# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:e4:71:55 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 83574sec preferred_lft 83574sec
    inet6 fe80::a00:27ff:fee4:7155/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:a4:05:09 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.112/24 brd 192.168.56.255 scope global dynamic enp0s8
       valid_lft 1133sec preferred_lft 1133sec
    inet6 fe80::a00:27ff:fea4:509/64 scope link
       valid_lft forever preferred_lft forever
root@debian:/etc/mysql/mariadb.conf.d# _
```

We use this info to configure our database connection settings in SQLManager as below. First, we right click in the Database window, and select Register Database.

This brings up the register database dialog box, which we can complete as below:





If you do the above properly, on the second dialog box it should find the database.

This indicates you have configured the database and the database account to properly allow access. Just keep in mind that when you config a PHP configuration file, **you need to use the local access database account**, not the network based admin account we used above, as seen in the code below:

```
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ]     = 'dvwa';
$_DVWA[ 'db_password' ] = 'password';
```

Again, adjust the above if you didn't use that user name.

Test Deployment

Now that we have Debian set up, let's deploy a test application. There is no perfect way to balance setting up permissions to work/deploy a website vs protecting the website from attacks. As it is a balance, you can consider the following, but we will implement adding our login ID to the www-data group to upload to the appropriate directory.

Possible solutions that don't apply to what we are going to do include:

- Set an environmental variable called umask to 020, which would add write permissions to the members of a group to a folder, often /var/www/html. This involves creating a separate group, adding the web team members who need group permissions to that group, and ensuring that they have the umask permissions set for their connection, often through their ssh connection settings
- Uploading to a directory and copying/moving from that directory to the destination directory, again /var/www/html. This works for one-off deployments, but anything requiring regular updates requires repeating the copy/move steps
- Changing the owner of the /var/www/html directory (chown command) to the single person who deploys. While OK, can be limiting in a multi user environment.
- Logging in as root – never desirable, and requires changing connection settings again, which opens up the ssh connection to remote root login vulnerabilities, a serious threat
- Changing the group permissions to SGID, a special group permission that allows anyone to temporarily gain group permissions to create and/or execute scripts. Opens up too many permissions and is overkill for what we want to do.
- Adding users who need to upload content to the www-data group and set the permissions to allow them to upload. It is a bit of a balance of the umask permissions and permanently setting the permission through the shell. We will be looking at this.

First, download the DVWA zip file from Learn. You may unzip/unarchive the zip you download to a directory. Somewhere easily accessible, like your Desktop or Documents folders.

You will need to do the following. I advise launching WinSCP before asked to, as you are changing your access to the Document Root directory of Apache, and you will need to restart your WinSCP client if it is already running, so please hold off starting WinSCP until asked to.

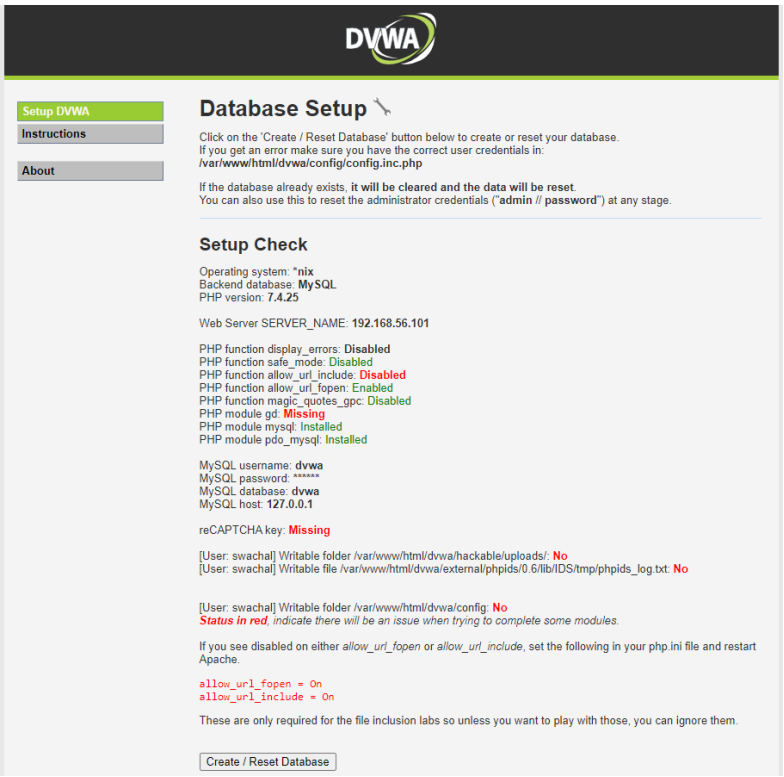
- Open VirtualBox, start your Debian VM. Login as usual, and become **root**
- Change to the **/var/www directory**
- Change the group ownership of the html directory (**`chown root.www-data html`**)
- Change the directory permissions of the html directory to rwxrwxr-x (**`chmod 775 html`**)
- Add your user account to the www-data group
    - **`adduser swachal www-data`**
- At this point, **launch WinSCP**
- You should be able to connect to your Debian server using WinSCP and your regular account, change to the /var/www/html directory, and through WinSCP as your regular account, create a directory called dvwa.

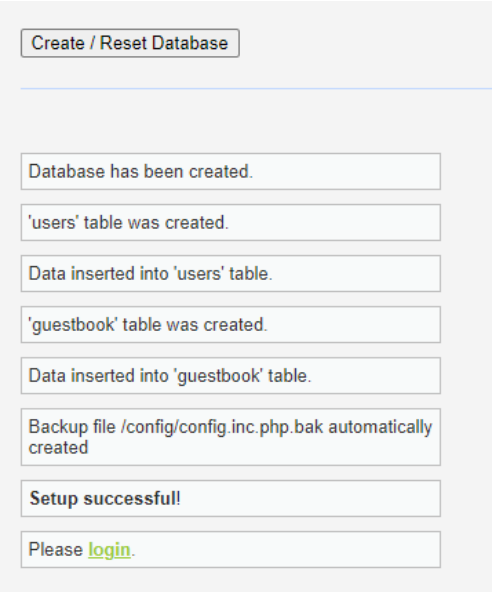
Unarchive the dvwa zip locally, modify the config file (in the config directory) based on the permissions:

```
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ]     = 'dvwa';
$_DVWA[ 'db_password' ] = 'password';
```

- As we are deploying this to a more harden, imitation production environment, do not leave sample files or backup files.
- Do not copy the config.inc.php.dist, rename it to config.inc.php, and modify the file.
- If your text editor makes backups, delete those files before you upload.
- Upload the unarchived contents of the DVWA zip to the dvwa directory.

That should be it. You should be able to go to your preferred browser, navigate to the IP address of your Debian machine, and the /dvwa directory (likely 192.168.56.21/dvwa).





You should also be able to go back to your database management tool and verify that the database has been populated with the two tables: