OpenVas Lab Documentation

This is all of the documentation related to the openvas testlab. Before diving into the documentation, refer to the standards for documentation… After that, there’s a table of contents for the rest of the documentation.

**Standards for documentation**

In order to make this process easy to understand for new users, we must provide proper documentation. For each step in the documentation, show screenshots of the output and commands used. Then, provide a clear description underneath each screen shot. Use times new roman, size 14, and 1 inch margins.

FORMATTING OF EACH STEP:

**Step #:**

Explanation text

[Image goes here]

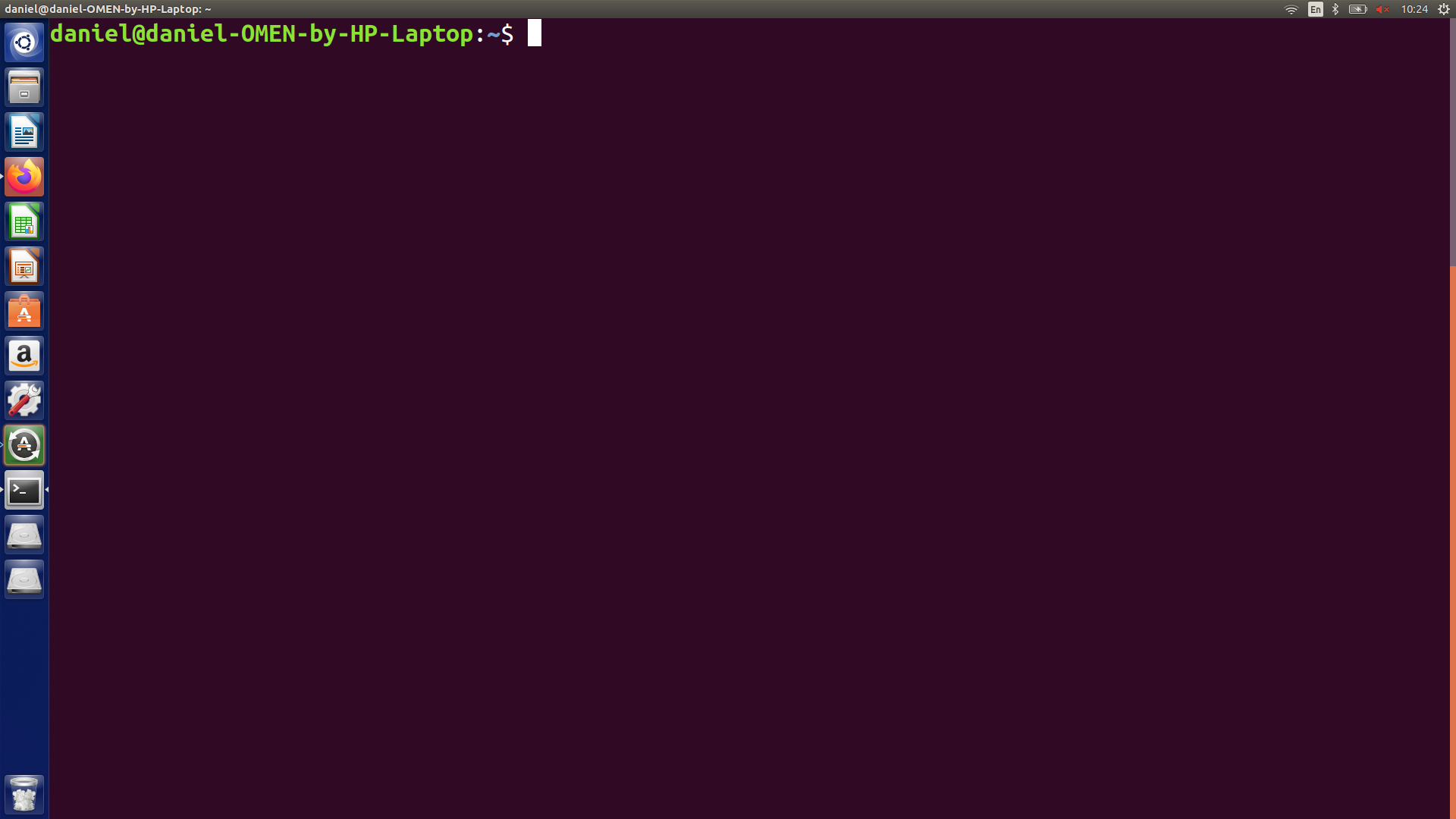
Explanation text

Below is an example of what the documentation should look like.

**How to update all your packages on linux**

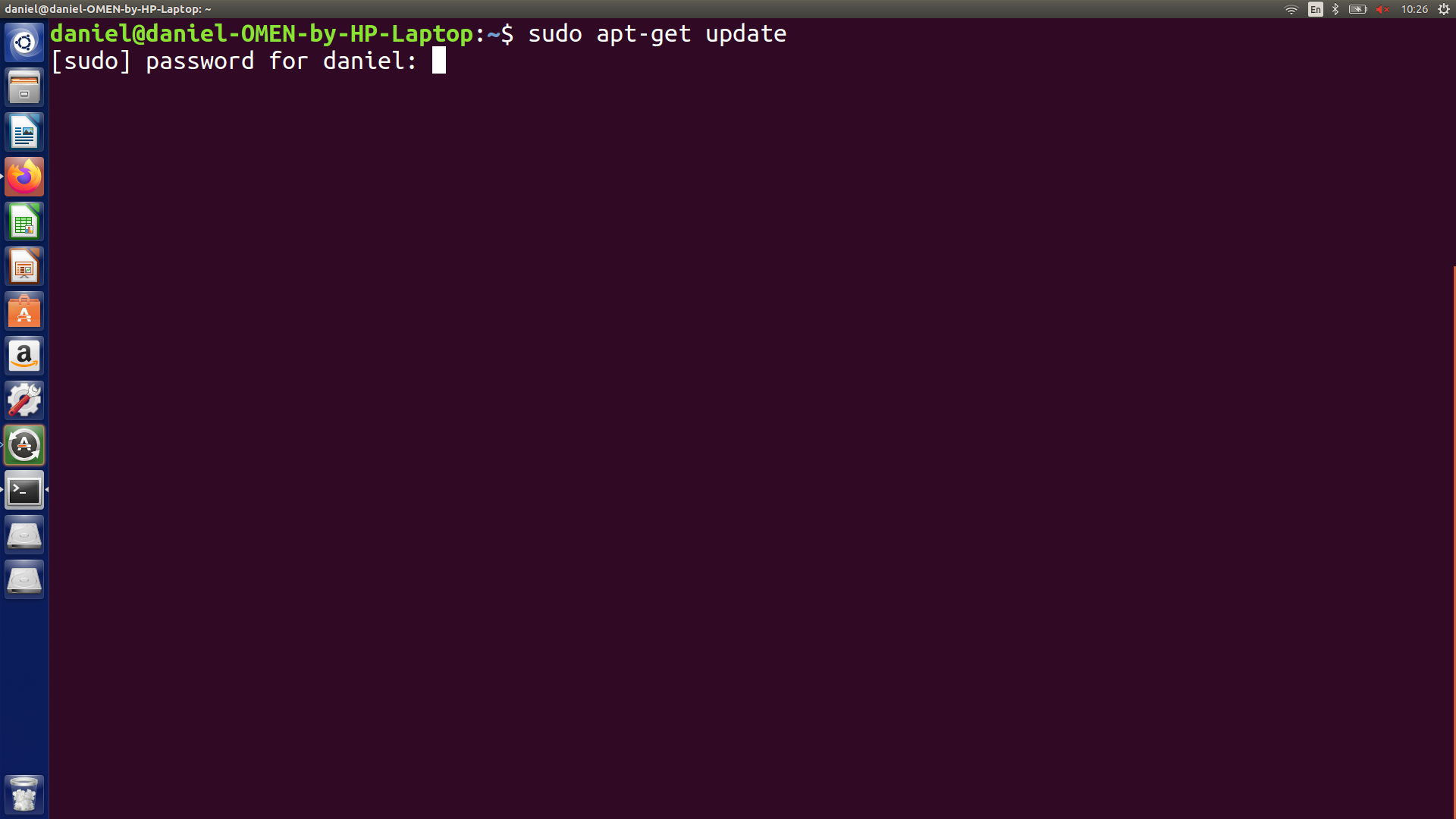
**Step 1:**

Open the terminal



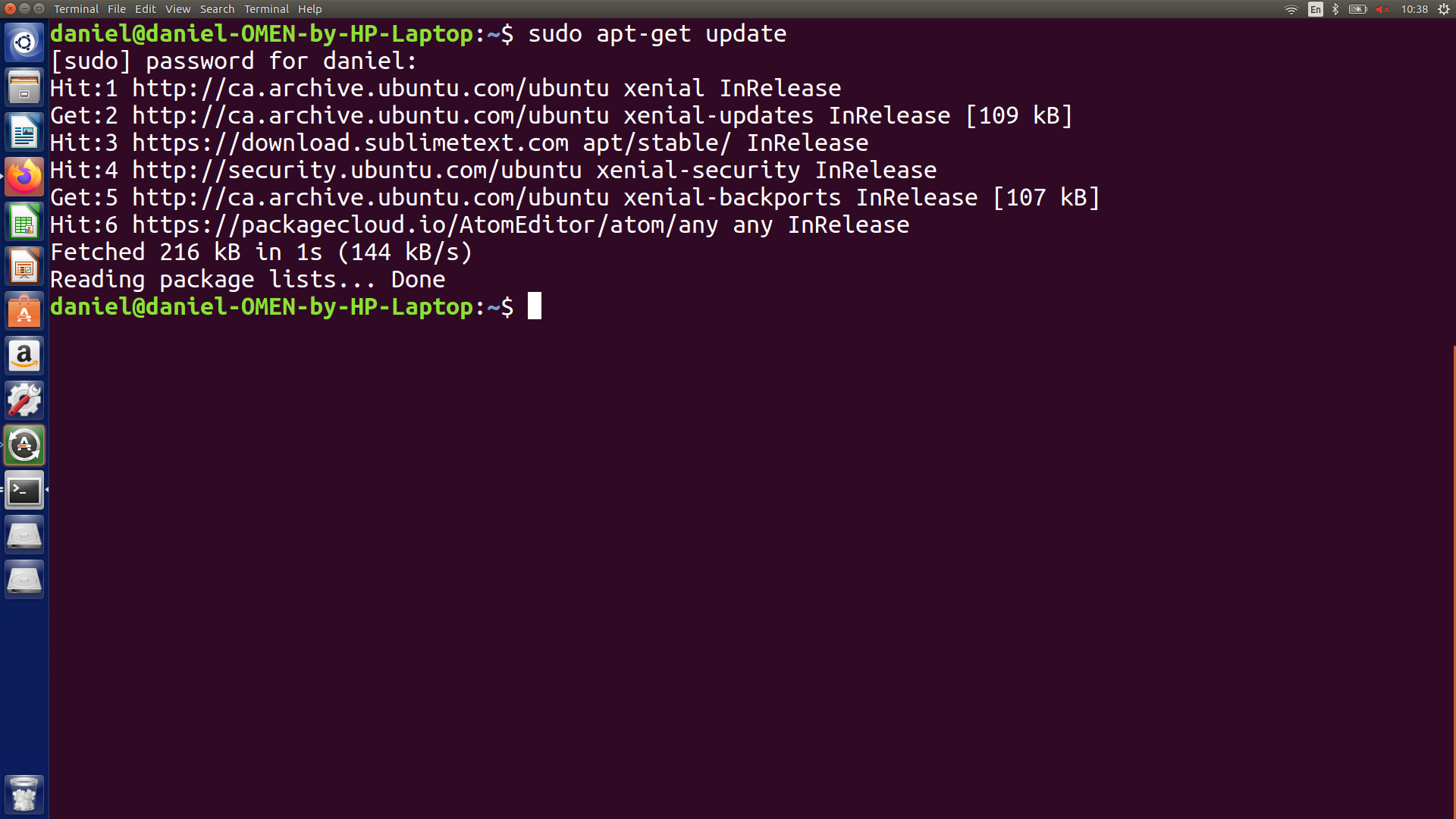
This can be done by pressing CTRL+ALT+T.

**Step 2:**

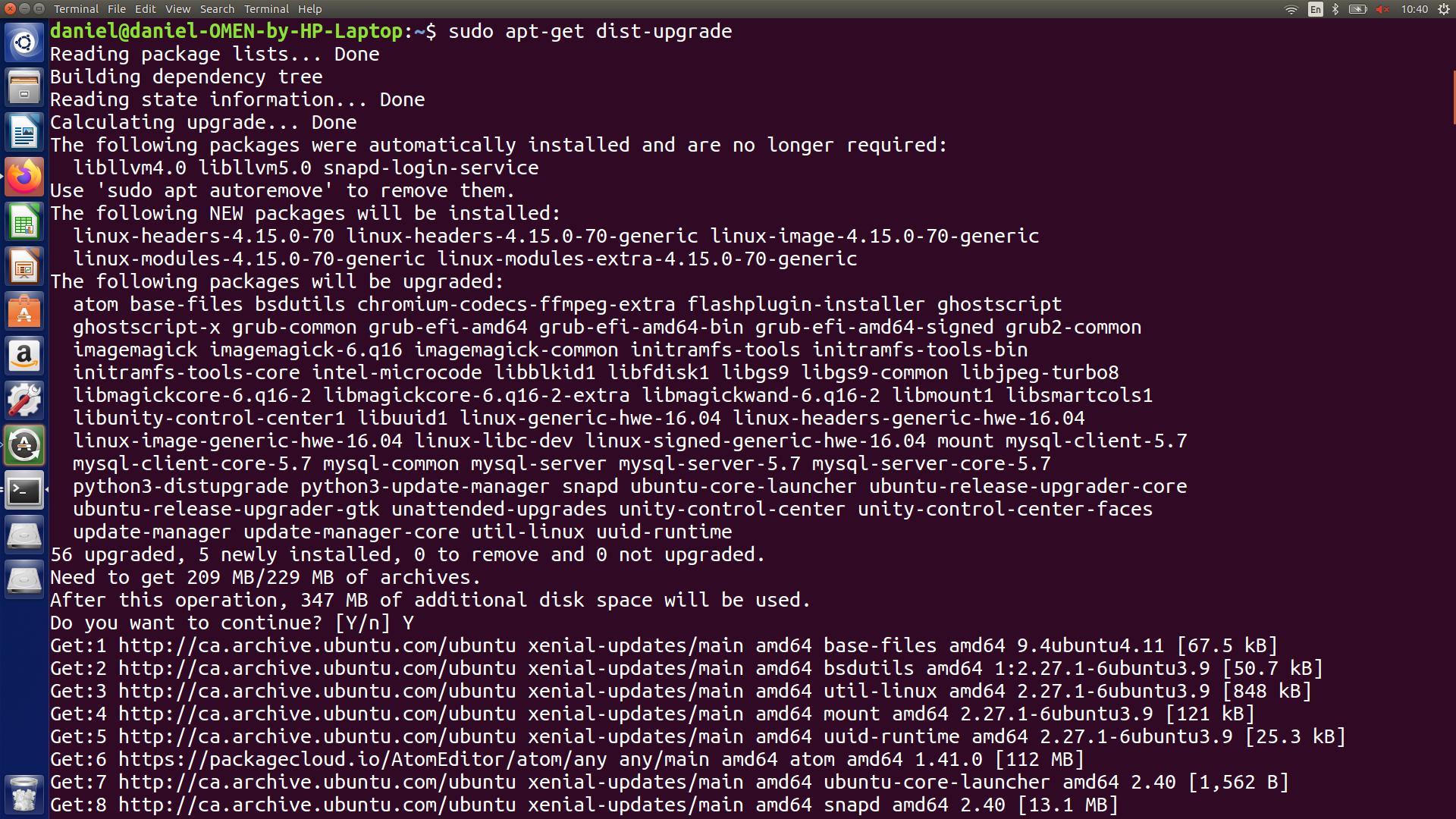


Type in sudo apt-get update

This command will get all the updates for your packages (but it wont update them yet), but you must first enter in your password. Below is the output after putting in your password.



**Step 3:** Installing the updates



Type in sudo-apt upgrade

This will install all the updates to your packages. When prompted, type in your password, and then type Y when it asks if you want to continue.

Table of contents:

1. Installing Ubuntu on your Hardware pg.8
2. Creating a router on ubuntu pgs.9-14
3. Adding a new tool to the environment pgs.15-18
4. Maintenance script standards pgs. 19-21
5. Installing OpenVas pgs. 22-31

**Installing Ubuntu 18.04 onto your hardware**

Supplies Needed:

Cpu

Ram

hdd/ssd

Flash drive

First you want to use a second computer to download ubuntu 18.04 onto a flash drive

Open a web browser and type in

<https://ubuntu.com/download/desktop/thank-you?country=US&version=18.04.3&architecture=amd64>

Download rufus from: https://rufus.ie/

Next run rufus and select the flash drive as your location.

Eject your flash drive

Put your flash drive into the desired device

Power on your new device

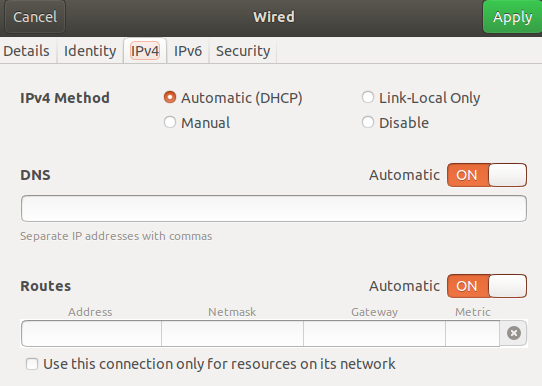
Press f12 until you reach a bootloader option

Select boot from USB

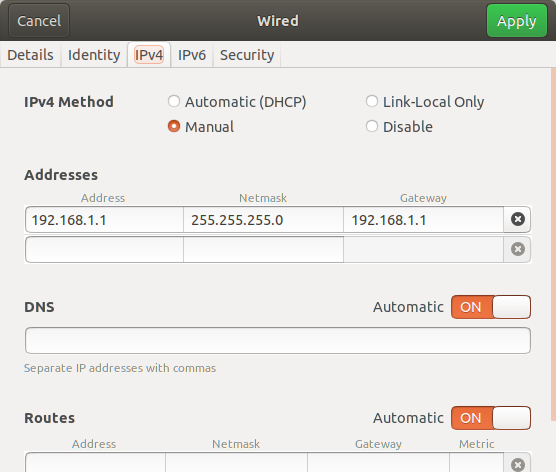
**Creating a router on Ubuntu 18.04**

**The first step** to create a route is to make sure you have two network interface cards (NIC) in your PC. To reach the screen settings for your NICs, press your windows key and search network.

NIC one should be set like this:



NIC two should be set like this:



**The second step**.. Open up the terminal (ctrl+alt+t), type in the command below to install the software to make a DHCP server

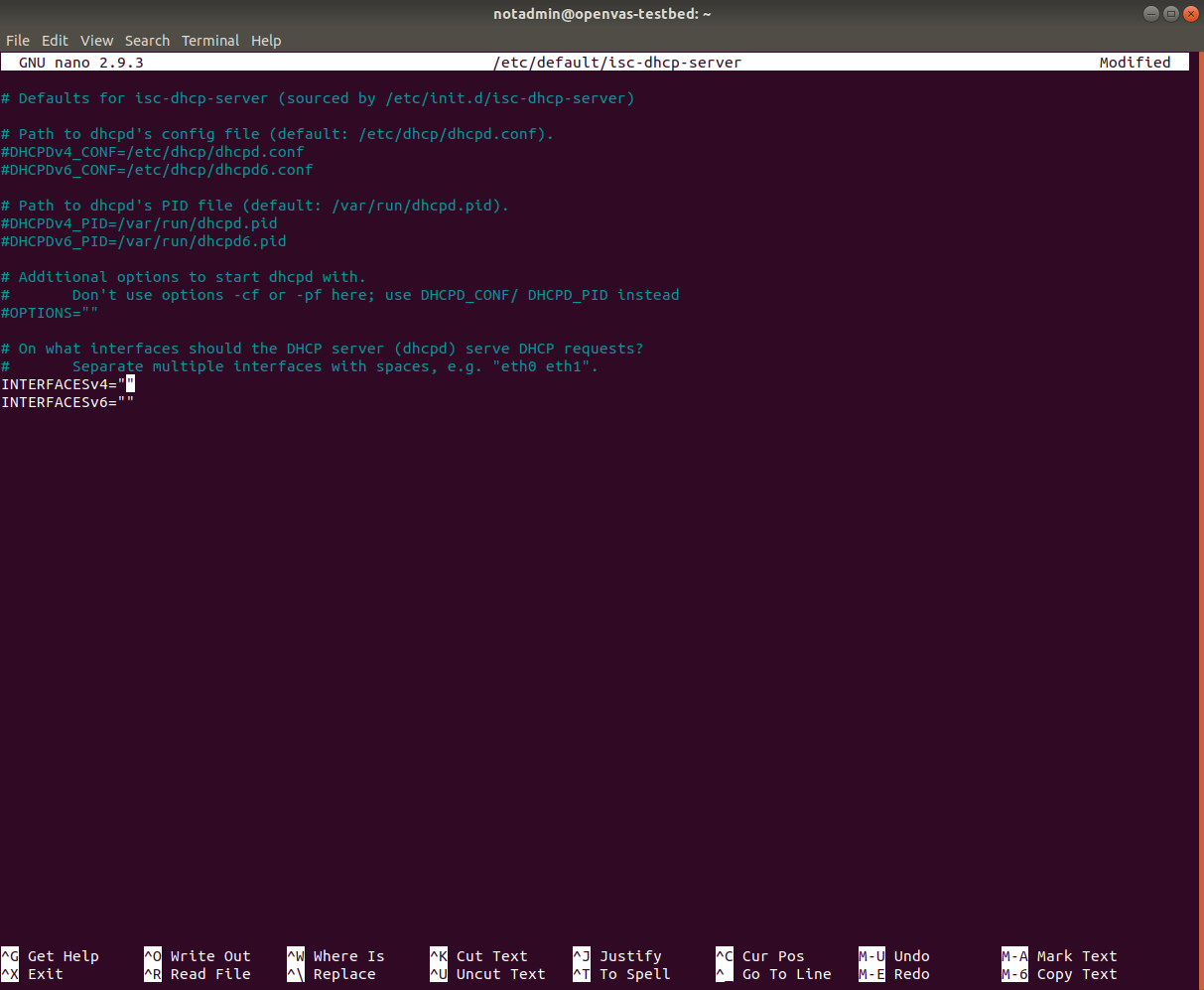


**The third step** is to make sure you select the correct NIC to utilize Dynamic Host Configuration Protocol (dhcp):

Enter nano -w /etc/default/isc-dhcp-server



This will put you on the following screen:



Uncomment the following lines by removing the #:

DHCPDv4\_PID=/var/run/dhcpd.pid

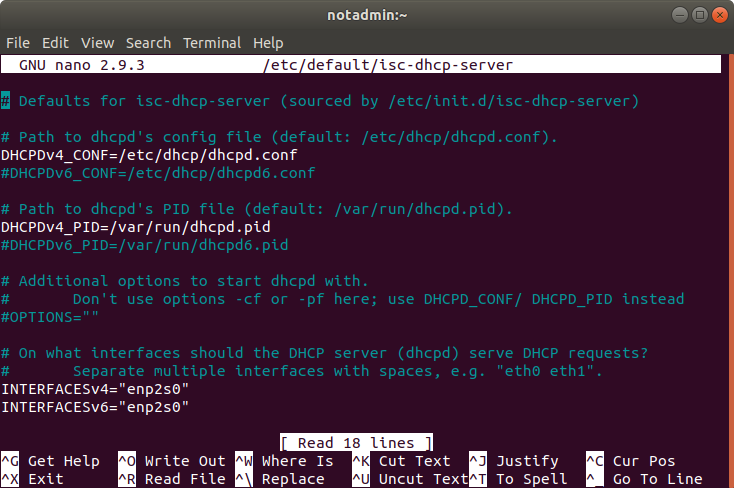
DHCPDv4\_CONF=/etc/dhcp/dhcpd.conf

Next navigate to the Line INTERFACESv4=”” insert the desired NIC name.

For the current router we will be using INTERFACESv4="enp2s0"

Next navigate to the Line INTERFACESv6=”” insert the desired NIC name.

For the current router we will be using INTERFACESv6="enp2s0"



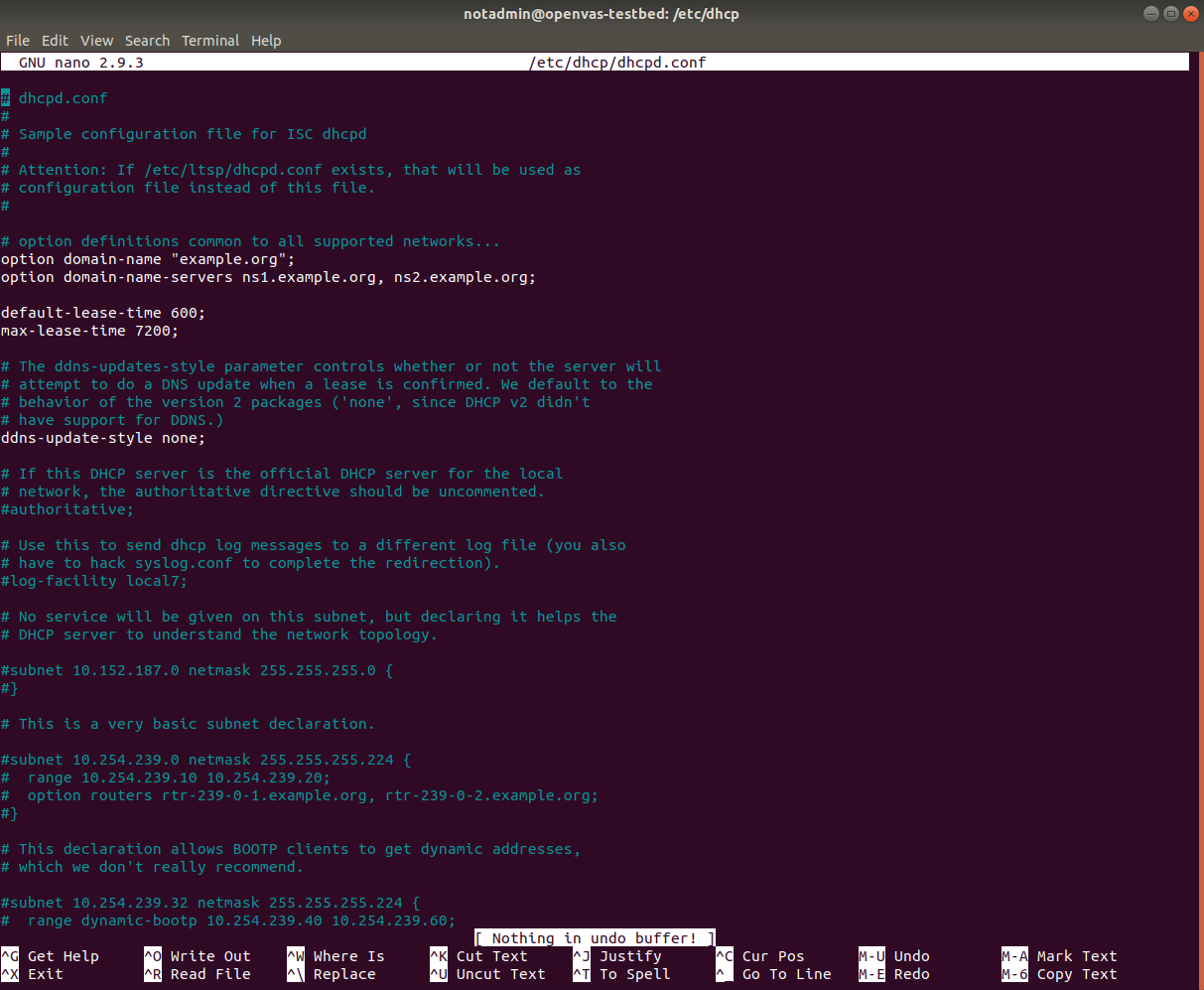
Then save by pressing **ctrl+x, y, enter** (in that exact order).

**The fourth step**.. you must set up a DHCP server on your device.

Start by changing the DHCP config file.



This should get you to a page like below.



Uncomment the following line by removing the #:

authoritative;

Then make the following changes below the authoritative line:

subnet 192.168.1.0 netmask 255.255.255.0 {

range 192.168.1.100 192.168.1.200;

option subnet-mask 255.255.255.0;

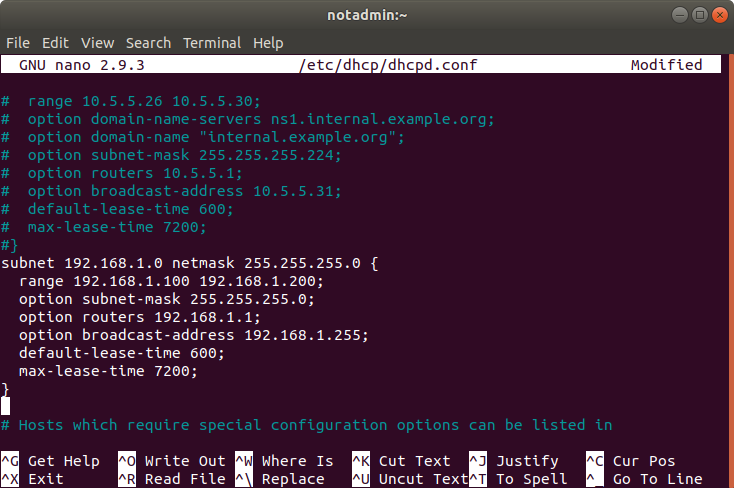
option routers 192.168.1.1;

option broadcast-address 192.168.1.255;

default-lease-time 600;

max-lease-time 7200;

}



Then save by pressing **ctrl+x, y, enter**.

**The fifth step.**. Start your DHCP server:

In the terminal, Type in **sudo systemctl start isc-dhcp-server.service**



This will start the dhcp server

Type in **sudo systemctl enable isc-dhcp-server.service**



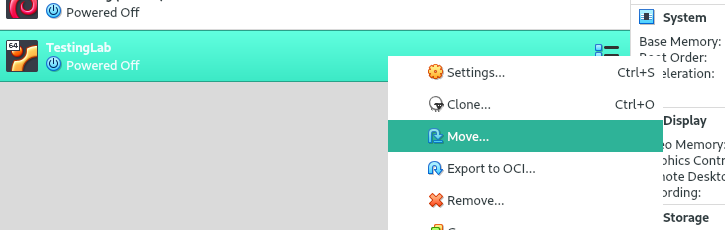
This will start dhcp on restart of the computer

**Adding Tools**

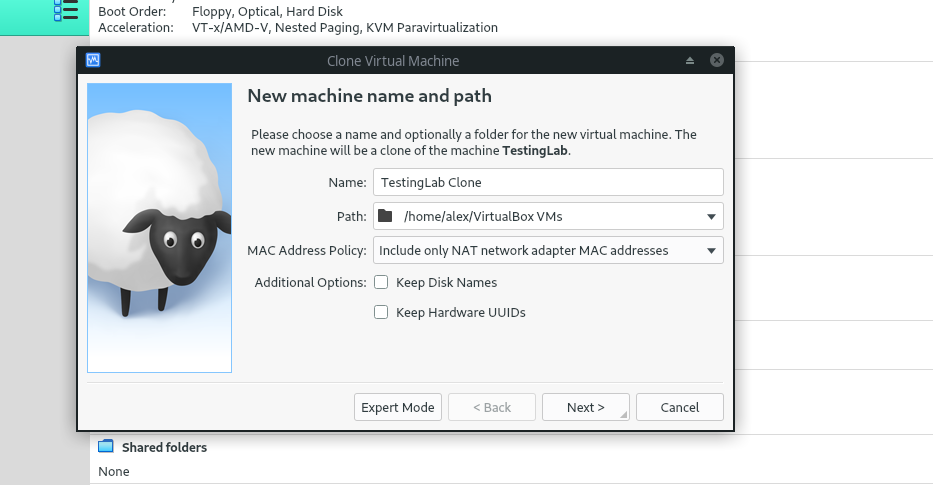
**Step one:** Clone the test Lab environment Following step 1A orinstall from scratch on development computer Following steps from 1B

**1A)Steps for cloning**

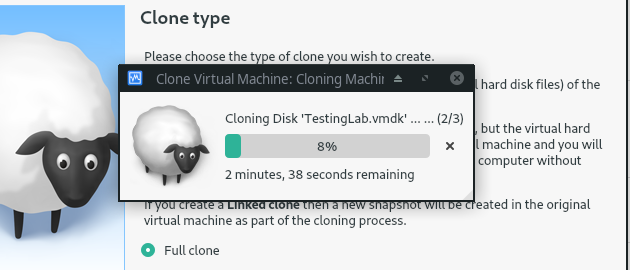
Open virtualbox and right click the test machine and click the sheep that says clone



Next, name the clone machine and leave the options at default and click next



Then select full clone and click clone this may take a while



After the clone is done open the machine and follow the testing standards to make sure that the tools are working.

**1B) From scratch:**

Follow the “Installing ubuntu on virtualbox” documentation to install ubuntu, then follow the installation document of each tool that has already been added to the suite until you have a working environment.

Afterwards run through the testing documentation to check if the installations were successful

**Step two:**

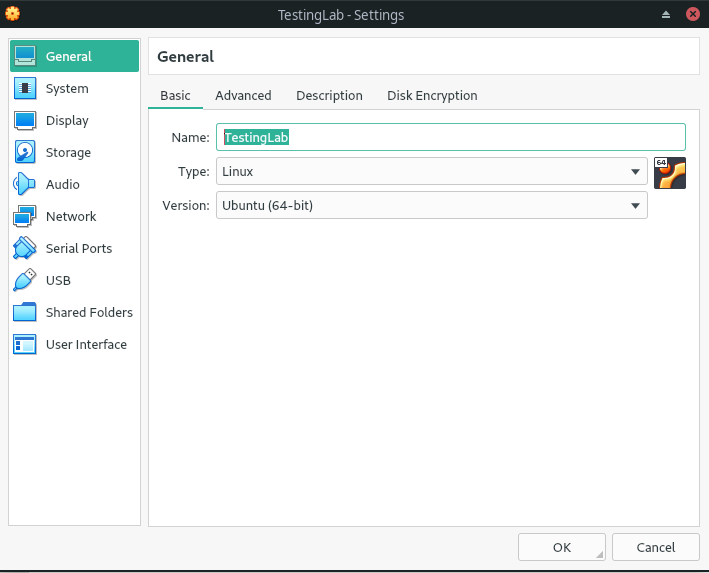
Install the tool on the clone machine and test it. The previously installed tools need to be checked to make sure that the installation did not break them. In the Documentation you will find the test each tool needs to pass to verify functionality.

**Step three:**

Document how to install your tool and testing of your tool following the Documentation standards. An example of these being implemented can be found in the “Installing Openvas on ubuntu 18” and “Testing openvas” documents.

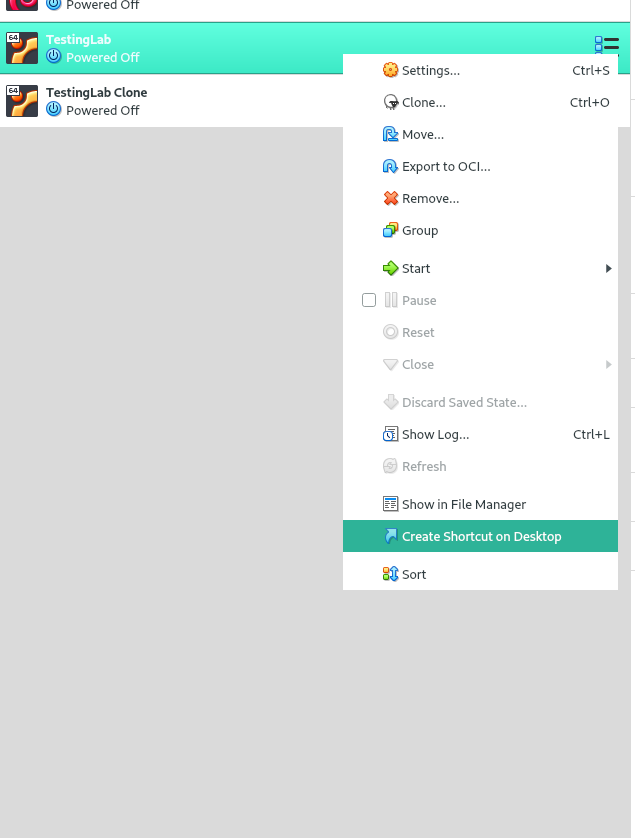
**Step four:**

Rename the Current testing machine to “Backup” and the clone machine to “Testing lab” this can be accomplished in the general settings tab



\*\* if you installed From scratch on another machine you will need to export the machine to a .ova file and import it into the test environment

After this create a link to the virtual machine on the desktop by right clicking on the Testing lab machine and selecting the option. Students will be able to use this environment now



**Step five Finalization:**

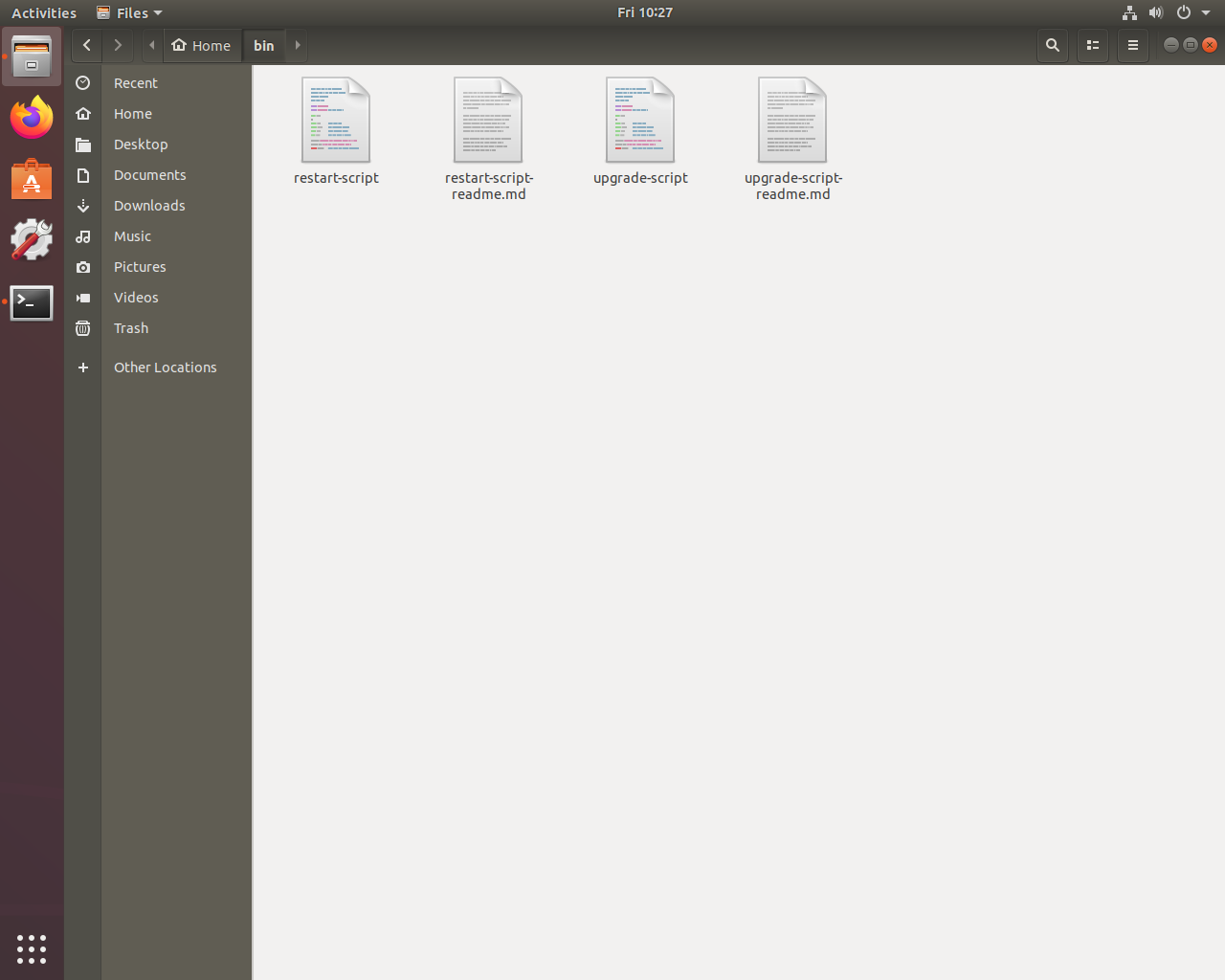
After the environment has been used without issues for at least 3 weeks you can export the machine into a .ova and remove the old backups.

**Script standards**

If you wish to add new scripts to the system, make sure you put them in the folder located at: /home/notadmin/bin.

Each of your scripts **must** have a meaningful name. As you can see below, the restart-script does exactly as it sounds.. It restarts the system. The upgrade script upgrades all of the packages. And, for each script, make sure you provide a detailed readme on what the script does exactly (similar naming conventions).

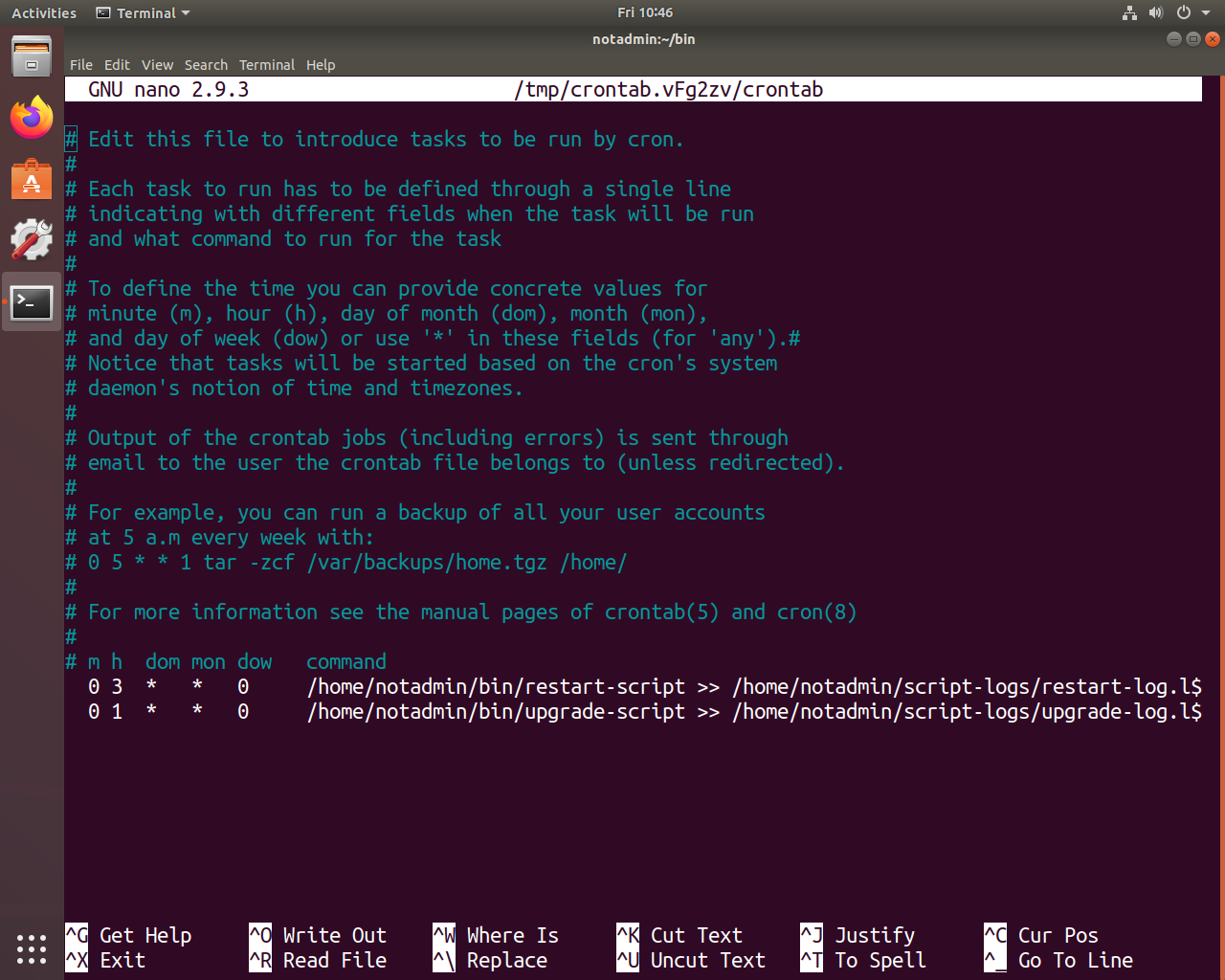
**Make sure you give the proper permissions to your script for it to be executed**



**Once your script is created, put it on the crontab (only if you think it’s necessary)**

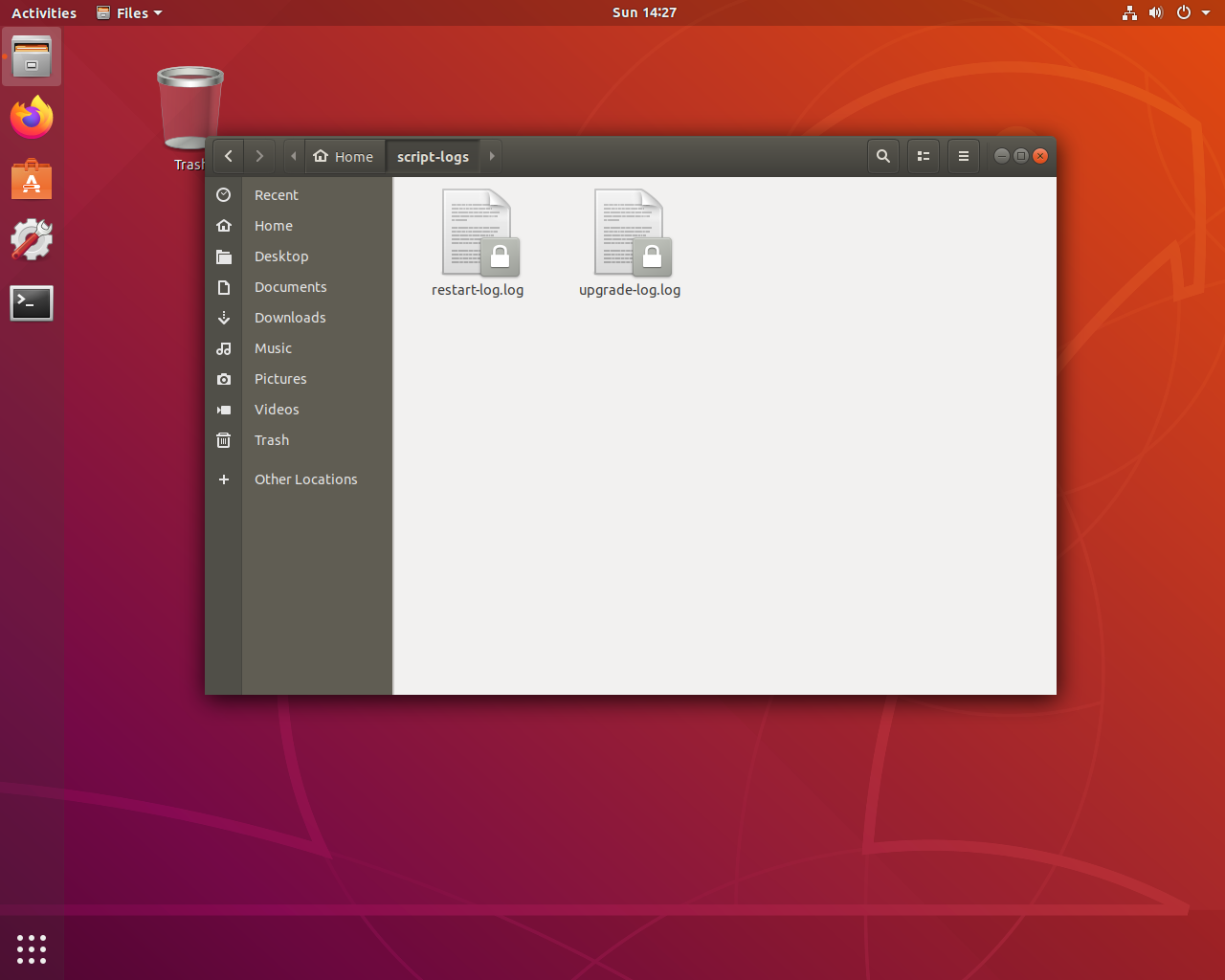
The crontab is a powerful tool. It allows you to schedule tasks to be run automatically at a specific time.

Type this command into the terminal to open crontab: sudo crontab -e



In order from left to right, put in the minute (m, 0-59), the hour (h, 0-23), the day of the month (dom, 1-31), the month (mon, 1-12), the day of the week (dow, 0-6), and the command you want executed. All of the output must be redirected to a log file to the specified path: /home/notadmin/script-logs/your-log-name.log

As shown below, the names of the log files are related to the script names. The restart script has a log named “restart-log.log” and the upgrade script has “upgrade-log.log”



**Installing openvas on ubuntu 18.04**

**Step 1:** update the programs currently on ubuntu with

|  |
| --- |
| sudo apt update  sudo apt upgrade |

**Step 2:** we need to make the folder where we will be installing it

|  |
| --- |
| cd /usr/local/src sudo mkdir gvm10 sudo chown $USER:$USER gvm10 cd gvm10 |

**Step 3:** we need to download the source code (type in the following commands in order):

|  |
| --- |
| wget -O gvm-libs-10.0.0.tar.gz https://github.com/greenbone/gvm-libs/archive/v10.0.0.tar.gz ;\  wget -O openvas-scanner-6.0.0.tar.gz https://github.com/greenbone/openvas-scanner/archive/v6.0.0.tar.gz;\  wget -O gvmd-8.0.0.tar.gz https://github.com/greenbone/gvmd/archive/v8.0.0.tar.gz ;\  wget -O gsa-8.0.0.tar.gz https://github.com/greenbone/gsa/archive/v8.0.0.tar.gz ;\  wget -O openvas-smb-1.0.5.tar.gz https://github.com/greenbone/openvas-smb/archive/v1.0.5.tar.gz ;\  wget -O ospd-1.3.2.tar.gz https://github.com/greenbone/ospd/archive/v1.3.2.tar.gz |

**Step 4:** extract the source code and get rid of the compressed files

|  |
| --- |
| Extract find . -name \\*.gz -exec tar zxvfp {} \; Remove  rm \*.gz |

Type in the following command to gain root access.. this will make it easier to install/run programs

|  |
| --- |
| sudo su |

**Step 5:**

Install the requirements for building the code

|  |
| --- |
| apt install software-properties-common ;\ add-apt-repository universe ;\ apt install -y cmake pkg-config libglib2.0-dev libgpgme11-dev uuid-dev libssh-gcrypt-dev libhiredis-dev \ gcc libgnutls28-dev libpcap-dev libgpgme-dev bison libksba-dev libsnmp-dev libgcrypt20-dev redis-server \ libsqlite3-dev libical-dev gnutls-bin doxygen nmap libmicrohttpd-dev libxml2-dev apt-transport-https curl \ xmltoman xsltproc gcc-mingw-w64 perl-base heimdal-dev libpopt-dev graphviz nodejs rpm nsis wget sshpass \ socat snmp gettext python-polib git curl --silent --show-error https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add - ;\ echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.list ;\ sudo apt-get update ;\ sudo apt-get install yarn  apt-get install texlive-latex-extra --no-install-recommends  apt-get install texlive-fonts-recommended |

**Step 7:** we need to install GVM libraries

|  |
| --- |
| cd gvm-libs-10.0.0 ;\  mkdir build ;\  cd build ;\  cmake .. ;\  make ;\ make doc-full ;\  make install ;\  cd /usr/local/src/gvm10 |

**Step 8:** Install openvas-smb

|  |
| --- |
| cd openvas-smb-1.0.5 ;\  mkdir build ;\  cd build/ ;\  cmake .. ;\  make ;\  make install ;\  cd /usr/local/src/gvm10 |

Install the openvas scanner itself

|  |
| --- |
| cd openvas-6.0.0 ;\  mkdir build ;\  cd build/ ;\  cmake .. ;\  make ;\  make doc-full ;\  make install ;\  cd /usr/local/src/gvm10 |

**Step 9:** Configure redis so it works with openvas

|  |
| --- |
| cp /etc/redis/redis.conf /etc/redis/redis.orig ;\ cp /usr/local/src/gvm10/openvas-6.0.0/build/doc/redis\_config\_examples/redis\_4\_0.conf /etc/redis/redis.conf ;\  sed -i 's|/usr/local/var/run/openvas-redis.pid|/var/run/redis/redis-server.pid|g' /etc/redis/redis.conf ;\ sed -i 's|/tmp/redis.sock|/var/run/redis/redis-server.sock|g' /etc/redis/redis.conf ;\ sed -i 's|dir ./|dir /var/lib/redis|g' /etc/redis/redis.conf sysctl -w net.core.somaxconn=1024 sysctl vm.overcommit\_memory=1 echo "net.core.somaxconn=1024" >> /etc/sysctl.conf echo "vm.overcommit\_memory=1" >> /etc/sysctl.conf  cat << EOF > /etc/systemd/system/disable-thp.service  [Unit]  Description=Disable Transparent Huge Pages (THP)  [Service]  Type=simple  ExecStart=/bin/sh -c "echo 'never' > /sys/kernel/mm/transparent\_hugepage/enabled && echo 'never' > /sys/kernel/mm/transparent\_hugepage/defrag"  [Install]  WantedBy=multi-user.target  EOF  systemctl daemon-reload ;\ systemctl start disable-thp ;\ systemctl enable disable-thp ;\ systemctl restart redis-server  cat << EOF > /usr/local/etc/openvas/openvassd.conf db\_address = /var/run/redis/redis-server.sock EOF |

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:‘/usr/local/lib/

**Step 10:** Next we need to do the initial openvas setup

|  |
| --- |
| greenbone-nvt-sync  ldconfig openvassd # watch "ps -ef | grep openvassd" |

Wait until it says waiting for incoming then quit with ctrl+C

**Step 11:** Now we need to install the manager

|  |
| --- |
| cd gvmd-8.0.0 ;\ mkdir build ;\ cd build/ ;\ cmake .. ;\ make ;\ make doc-full ;\ make install ;\ cd /usr/local/src/gvm10 |

**Step 12:** Now it’s time to install GSA the web browser interface

|  |
| --- |
| cd gsa-8.0.0 ;\  sed -i 's/#ifdef GIT\_REV\_AVAILABLE/#ifdef GIT\_REVISION/g' ./gsad/src/gsad.c ;\  sed -i 's/return root.get\_result.commands\_response.get\_results\_response.result/return root.get\_result.get\_results\_response.result/g' ./gsa/src/gmp/commands/results.js ;\  mkdir build ;\  cd build/ ;\  cmake .. ;\  make ;\  make doc-full ;\  make install ;\ cd /usr/local/src/gvm10 |

**Step 13:**it is time to do the setup for the whole system

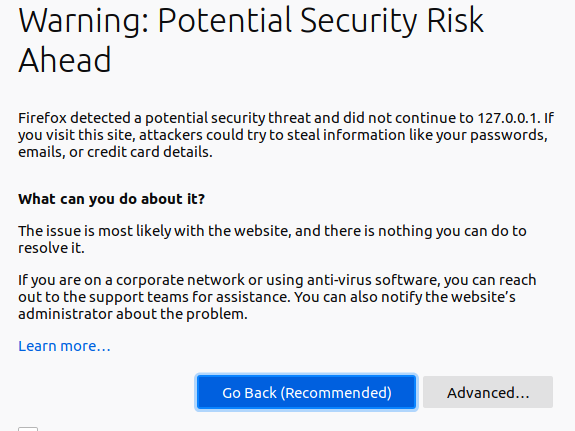
|  |
| --- |
| gvm-manage-certs -a gvmd --create-user=admin \* gvmd ;\ openvassd ;\ gsad |

\*This will echo out your admin password it will look something like this

Go to this web address

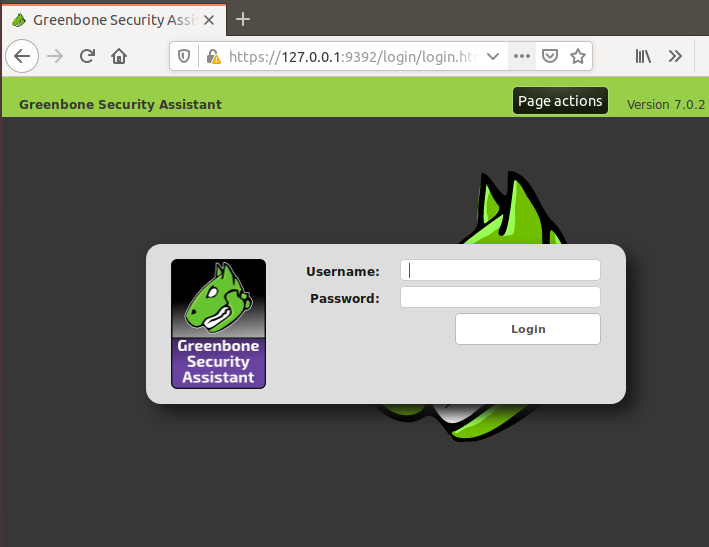
|  |
| --- |
| 127.0.0.1 |

You will probably get a page similar to this



This is ok click advanced then accept risk and continue in firefox

You should see a page like this now



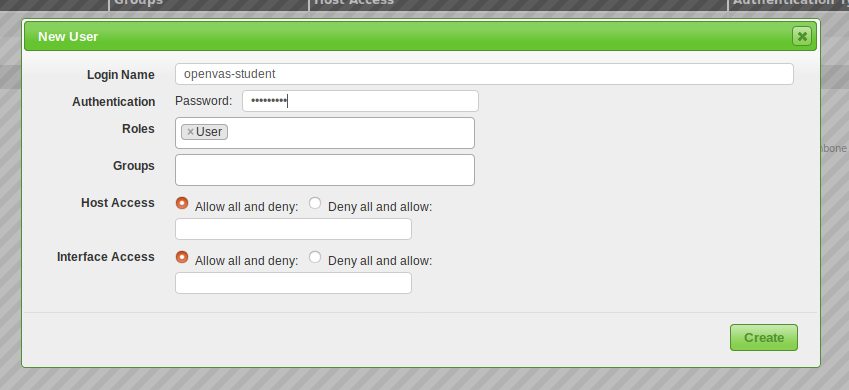
This is the greenbone web application. You can log into it with the username “admin”

And the password is the one you took down earlier

The next few steps will be to set up a user account so the command line interface can run normally

**Step 14:**

Go to the administration tab and click users

Click the star in the top right and it will open a pane like this 

Give the user the name openvas-student and password password1 with the role of user

While we are here, click the wrench and change the admin password to something too

**Step 15:**

Setting up command line interface

First we need to go through and change a whole bunch of ownership

|  |
| --- |
| cd /usr/local/etc sudo chown $USER:$USER -R gvm/ sudo chown $USER:$USER -R openvas/ cd /usr/local/var/ sudo chown $USER:$USER -R run cd /usr/local/var/log sudo chown $USER:$USER -R gvm  cd /usr/local/bin  sudo chown $USER:$USER gvm-manage-certs openvas-nasl openvas-nasl winexe winexe |

**Troubleshooting**

You will need to run these commands

Each time you reset

gvmd

openvassd

gsad

**### STILL need**

**Gvm tools?**

**#source for the info**

[**https://sadsloth.net/post/install-gvm10-src/**](https://sadsloth.net/post/install-gvm10-src/)