

SandPIM deployment on Fedora 32 in VirtualBox

At the end of this process, you will have a running “LAMP” stack hosting a running instance of SandPIM that you can connect to with a web browser on the host PC or from another machine on the LAN. The procedure should take about 30 minutes.

Assumptions:

- You are using Windows 10 Pro and the PC hardware supports virtualization – you may need to enable “Intel Virtualization” in the PC’s BIOS
- You have Oracle “VirtualBox” 6.1 installed
- Your PC is on a LAN with a DHCP server
- Your PC has at least 5GB of free drive space, and a few gigabytes of RAM to spare.
- Your host PC has Internet access (for downloading Linux and cloning the SandPIM repo)
- You have copies of (or access to) the MySQL versions of the AutoCare reference databases (VCdb, PCdb, PAdb, Qdb)

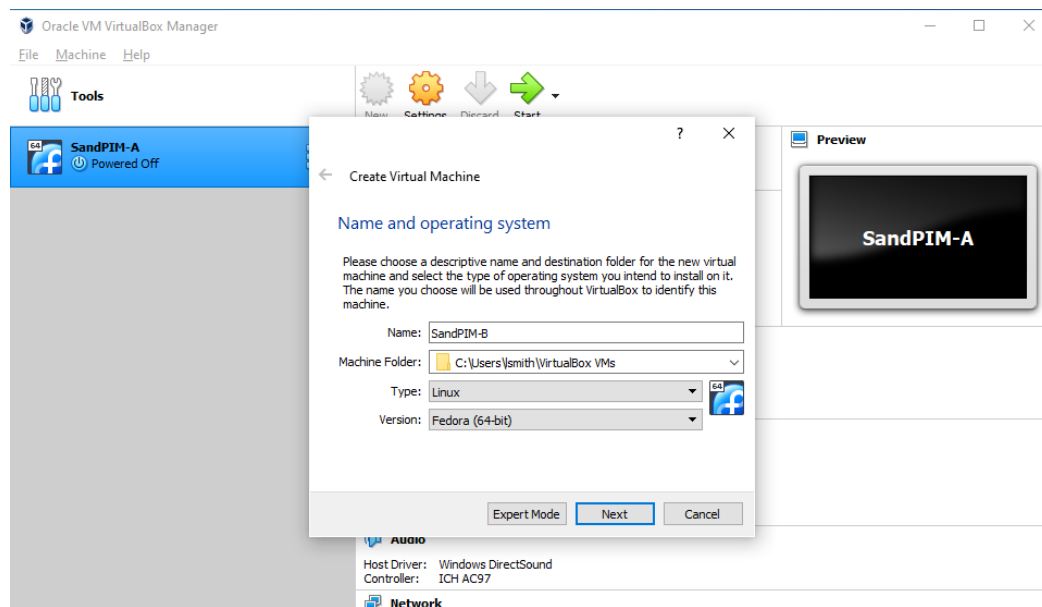
Obtain Linux operating system

Download the Fedora32 “net install” ISO image (about 650MB) from:

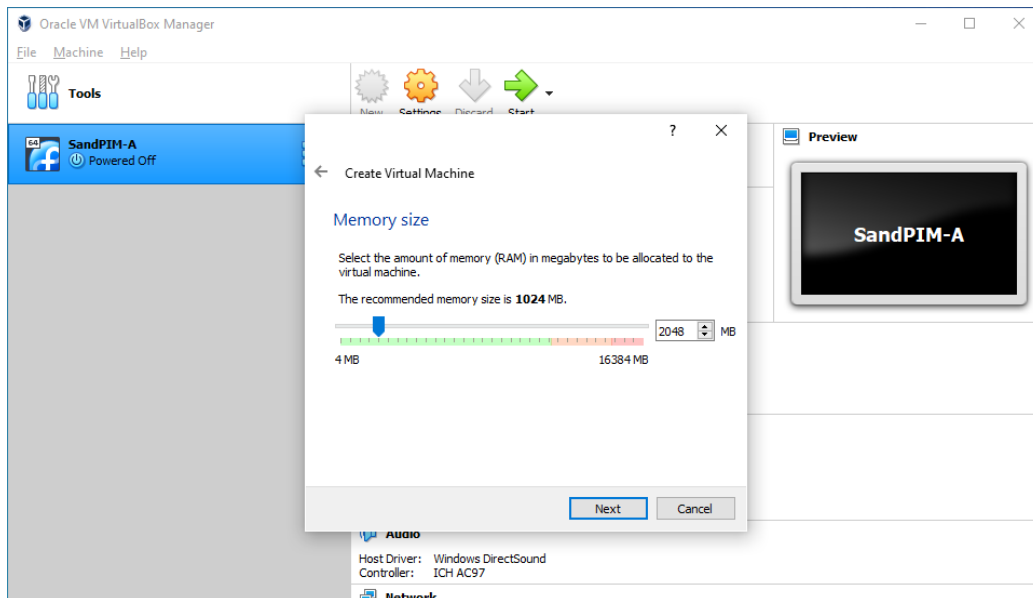
https://download.fedoraproject.org/pub/fedora/linux/releases/32/Server/x86_64/iso/Fedora-Server-netinst-x86_64-32-1.6.iso

Create a Linux Virtual Machine

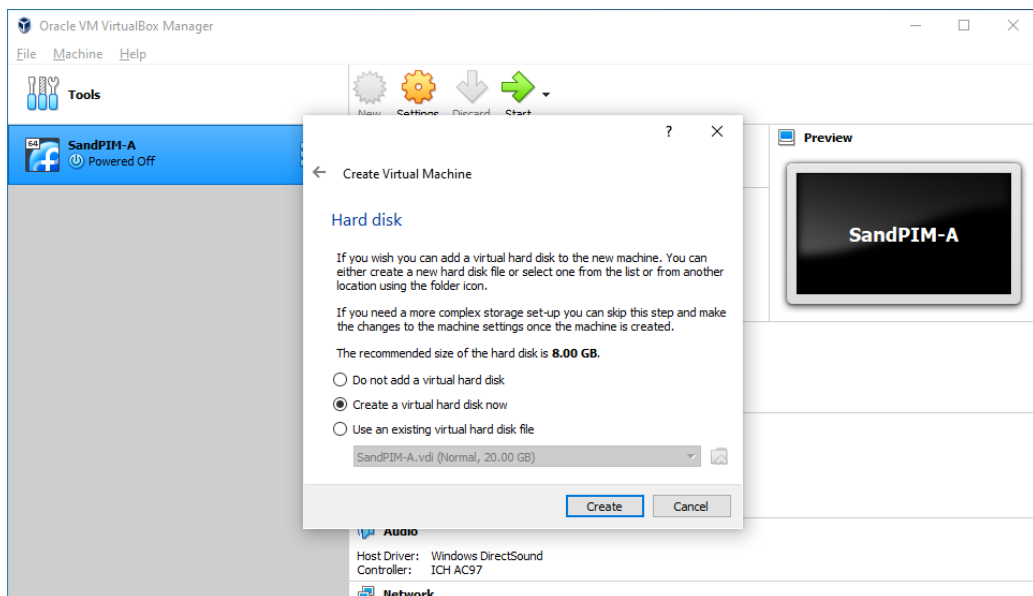
Click the “New” button in the VirtualBox manager



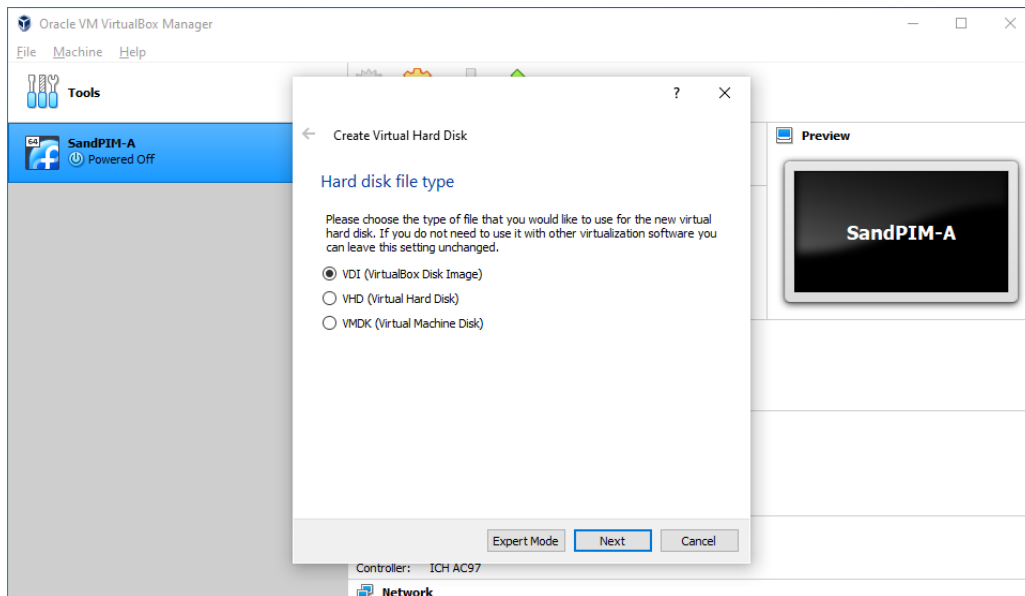
Allocate 2GB (or more) RAM



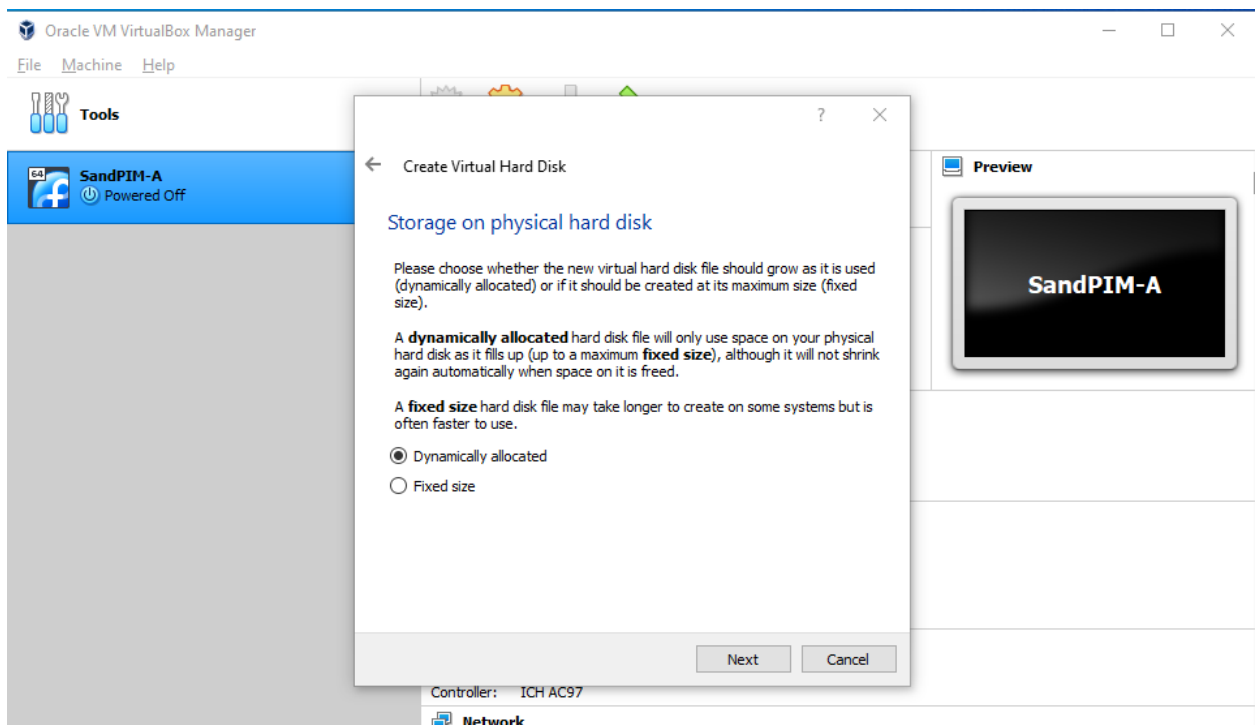
Create the virtual hard disk



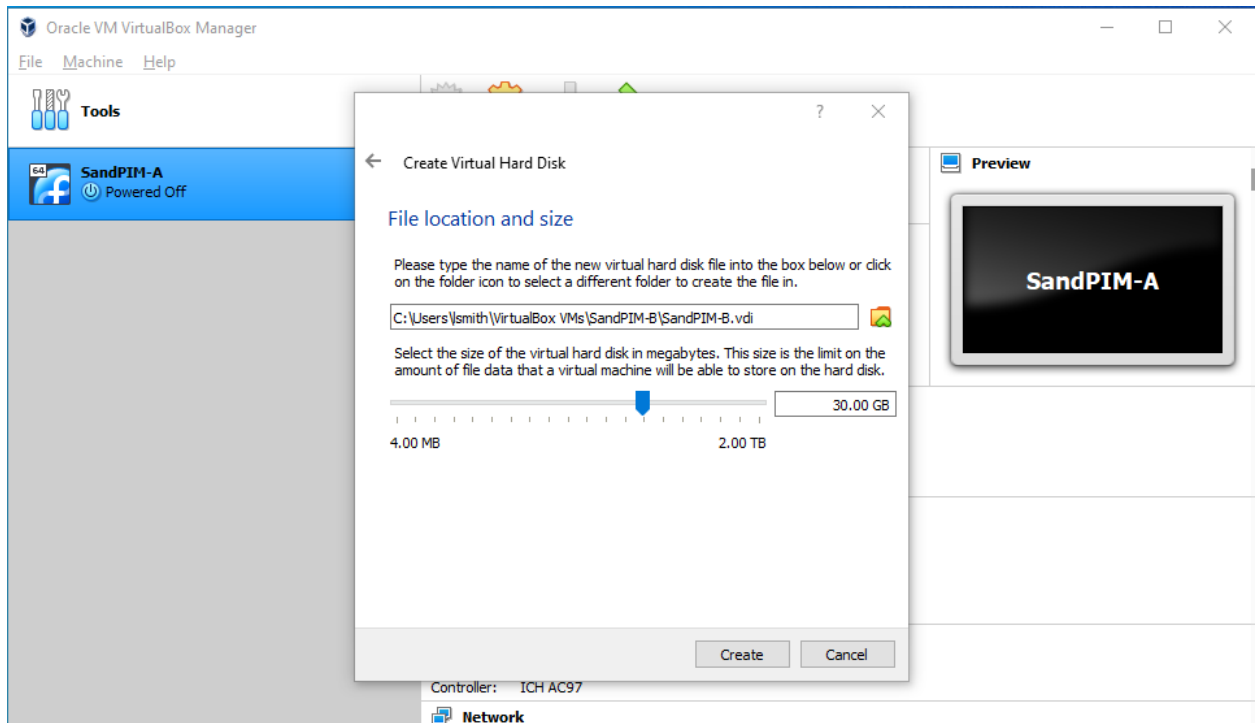
Specify VDI (VirtualBox Disk Image) as the disk type



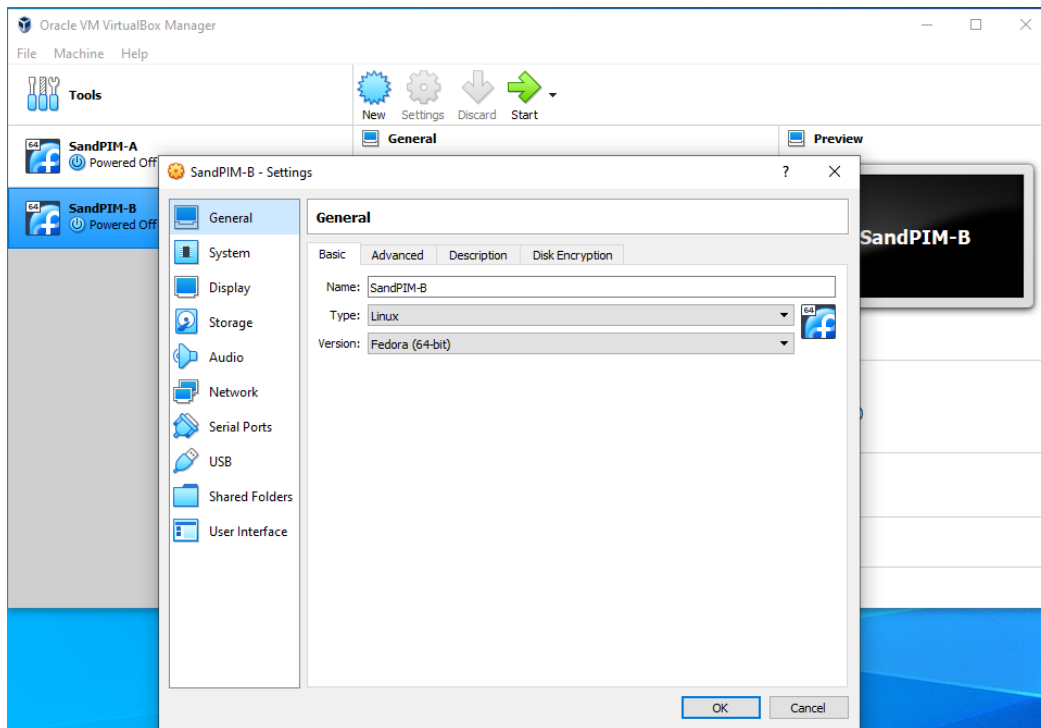
Specify "Dynamically Allocated" storage



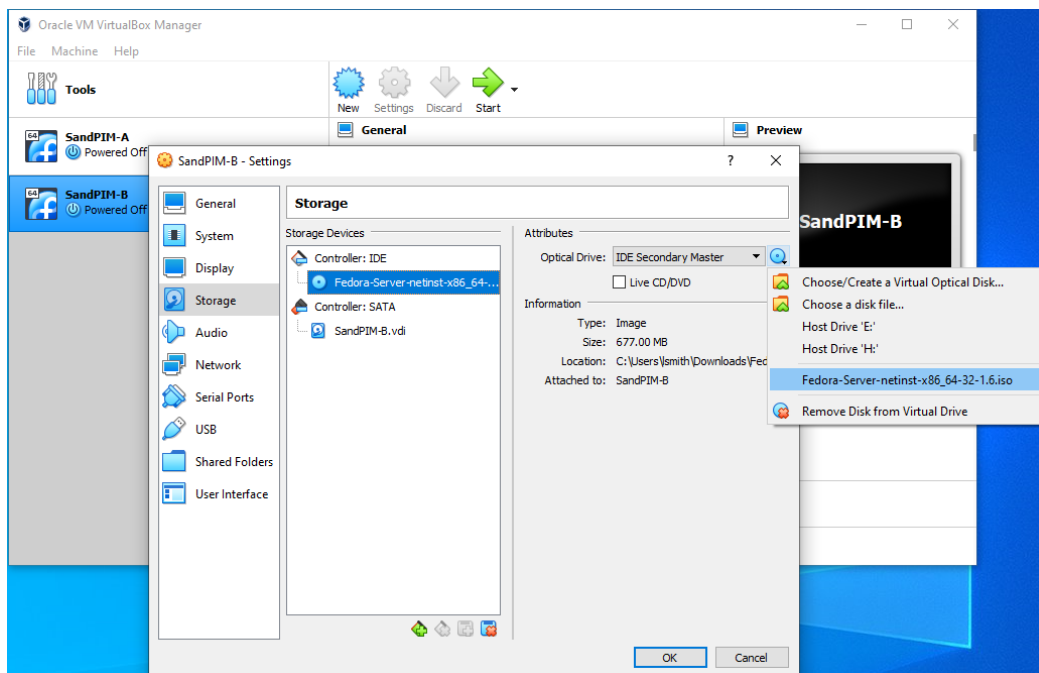
Allocate 30GB (or more) drive space. (this is an allowance – it won't immediately consume 30G of your PC's storage)



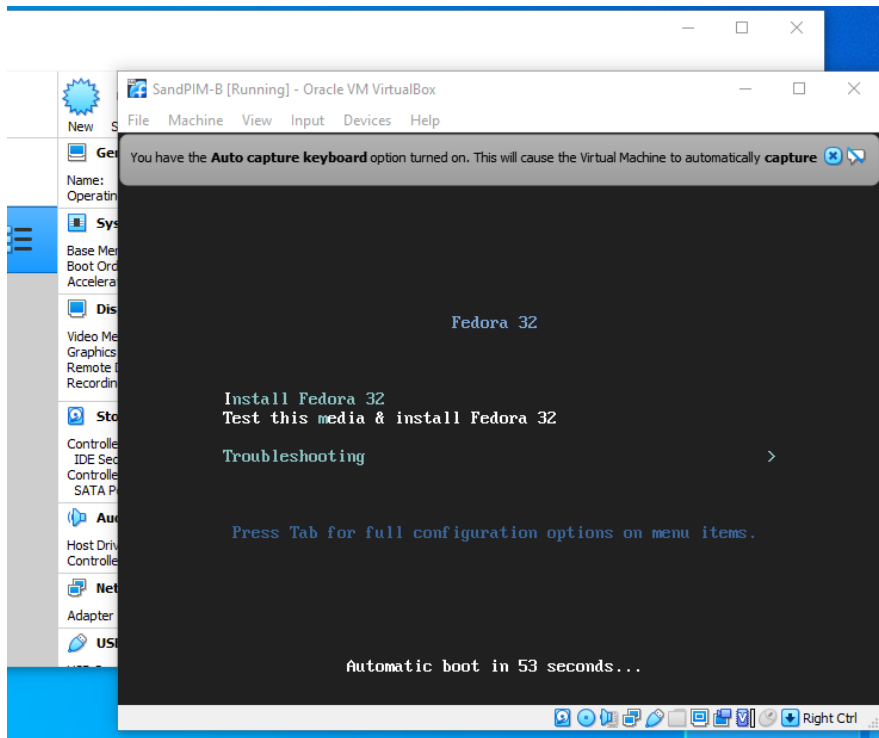
Highlight the new VM, and click the “Settings” icon in the Manager.



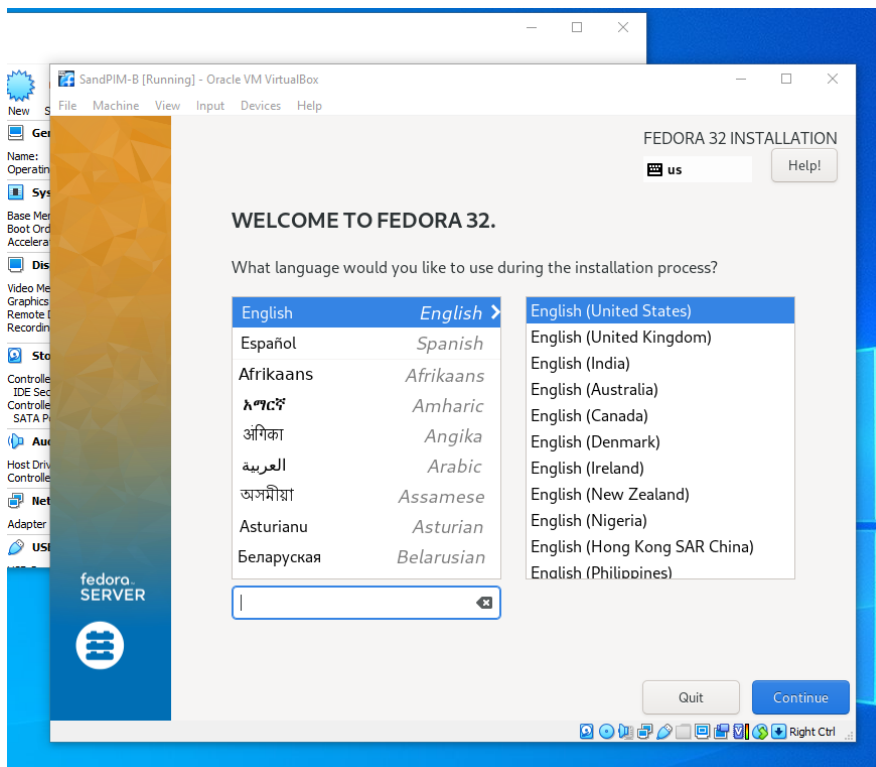
Click the “Storage” icon on the left , and specify the downloaded ISO file as the “disk file” to assign to the primary IDE controller.



Start the VM by selecting it in the manager and clicking the green arrow icon labeled "Start". A console window will appear after a few seconds.

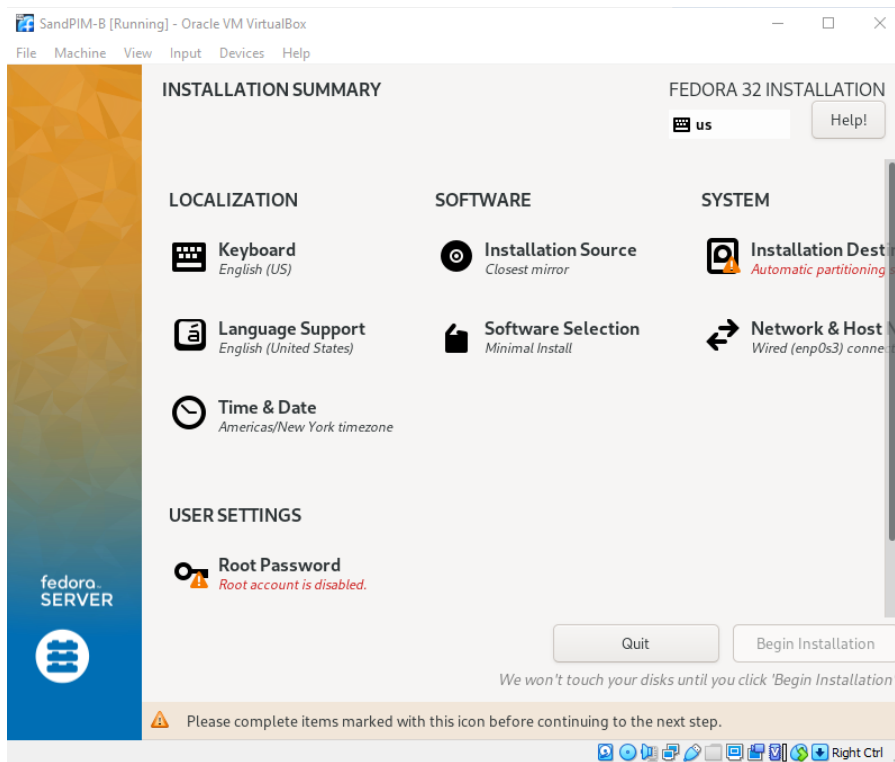


Press enter and after about a minute you will get to the installer GUI screen

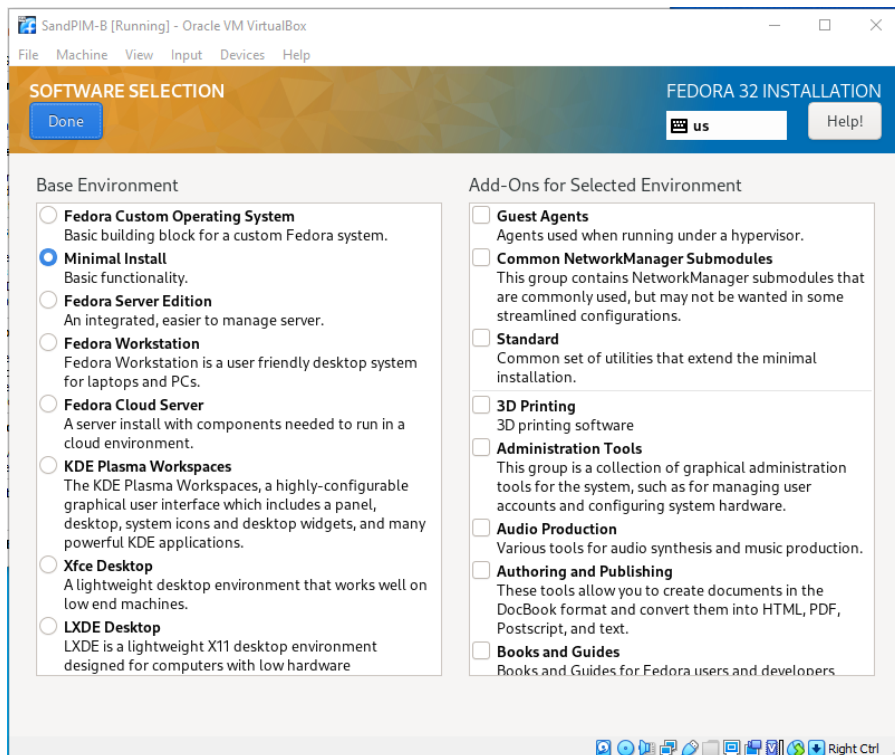


Select language and click “Continue”.

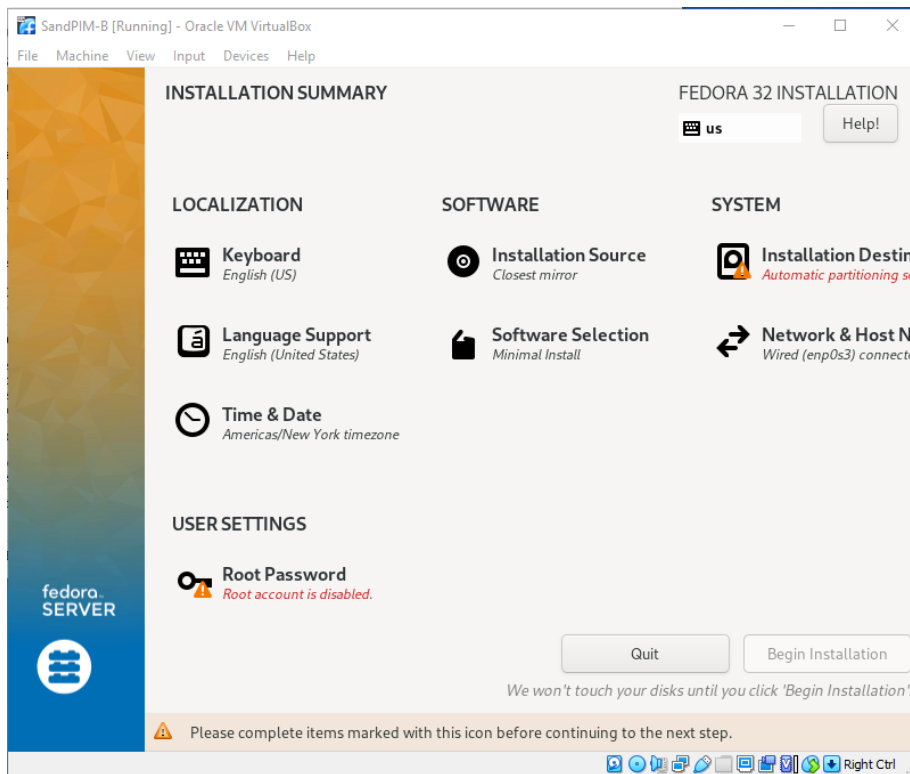
On the “INSTALLATION SUMMARY” screen, click “Software Selection”



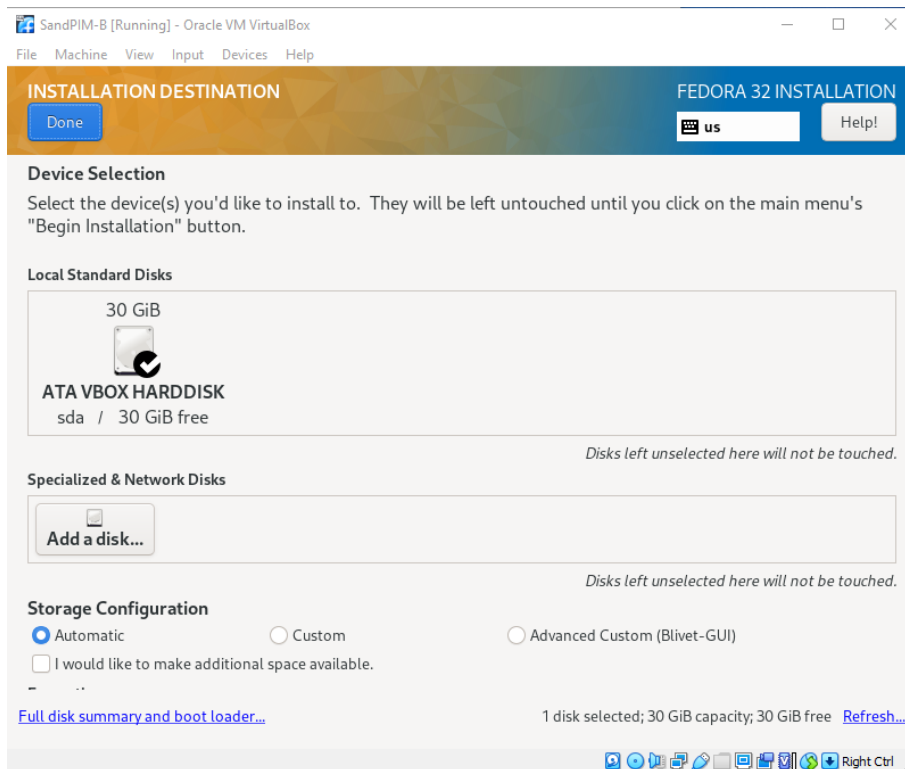
Select “Minimal Install” and click the “Done” button



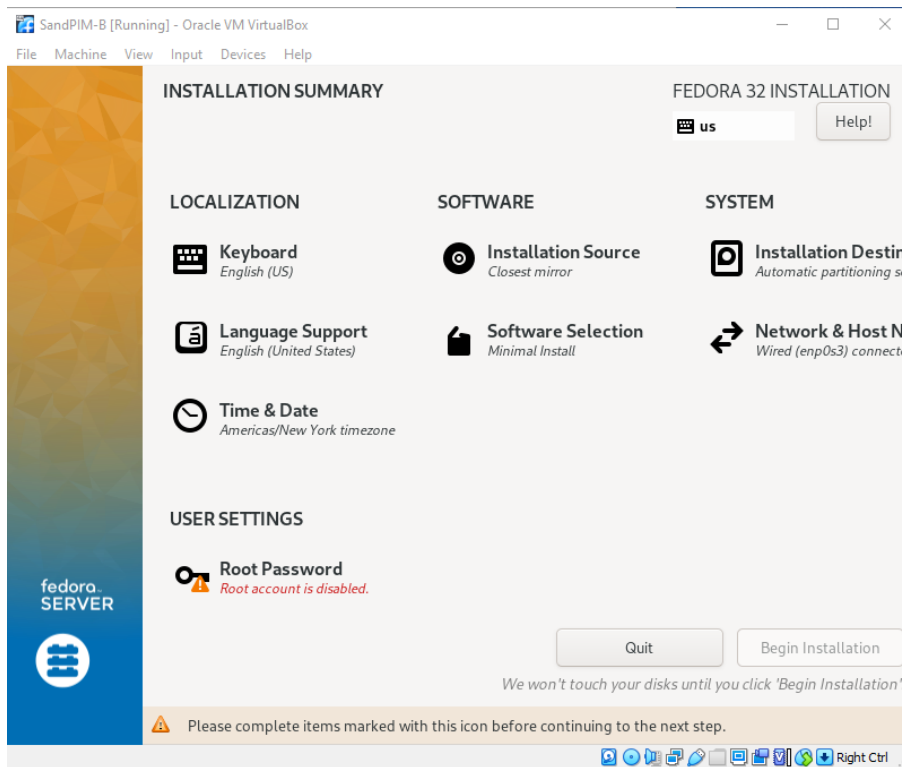
Click “Installation Destination”



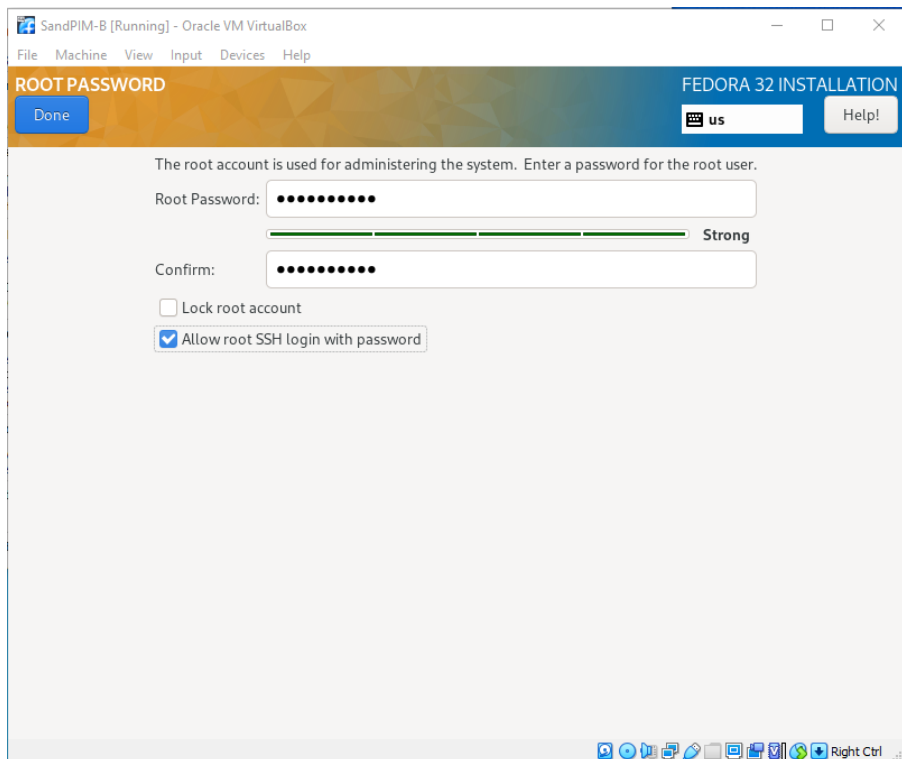
Verify that it sees the virtual disk that we allocated and click “Done”



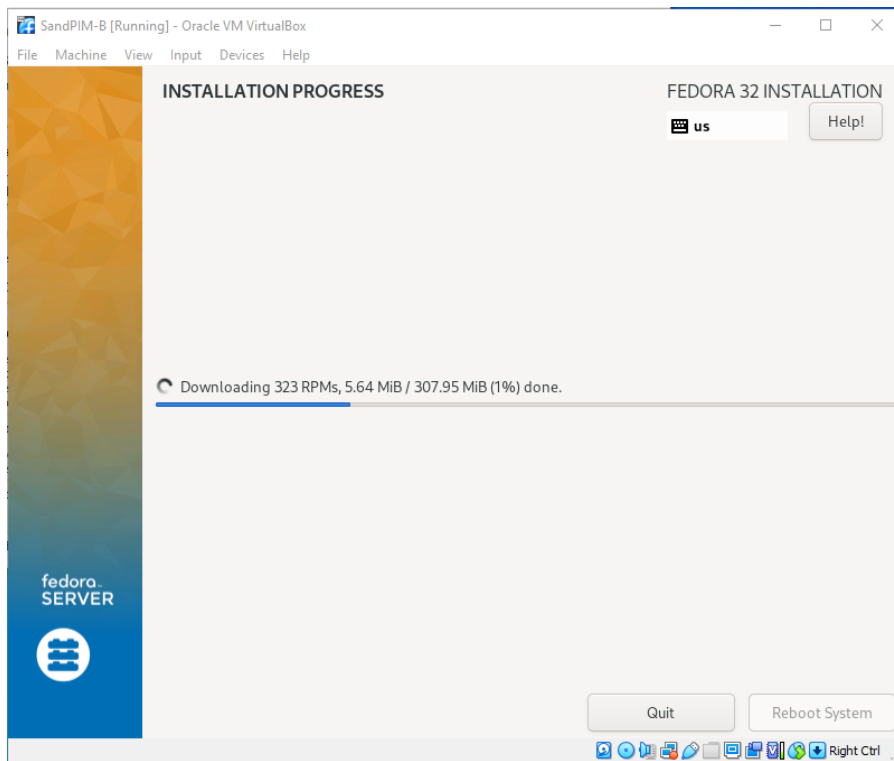
Click on “Root Password”



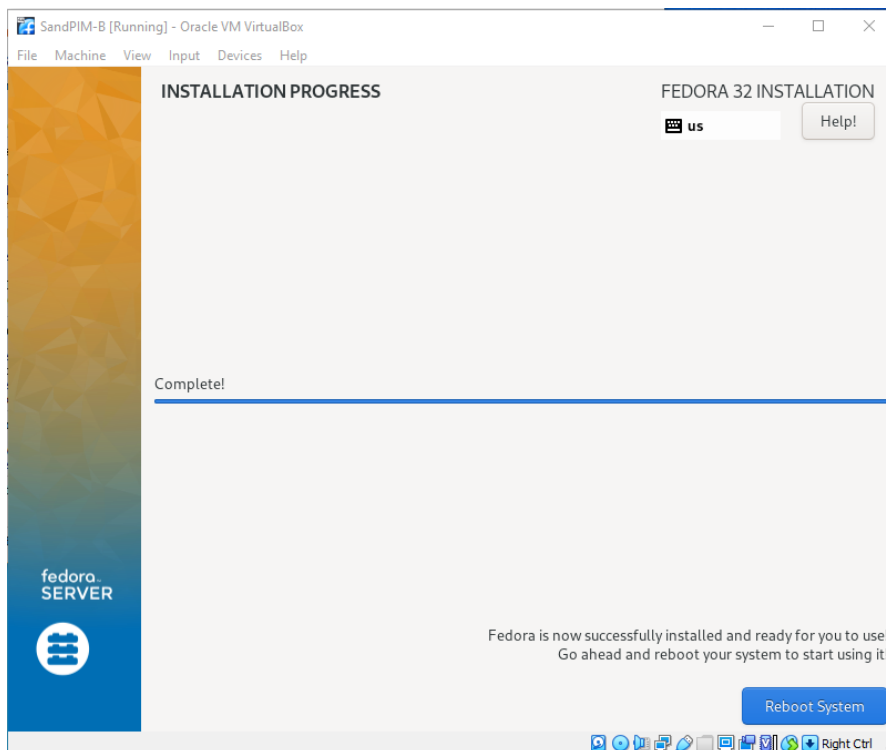
Enter and confirm a root password and check “All root SSH login with password” click “Done”



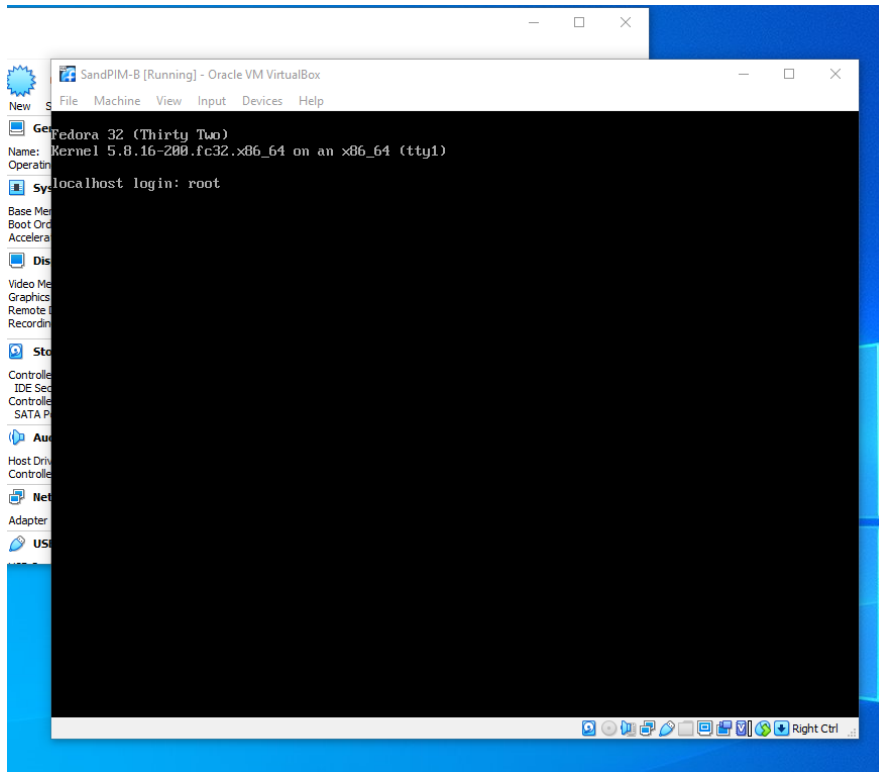
Click “Begin Installation” and you will see a progress bar for the next few minutes



When complete, click “Reboot System”



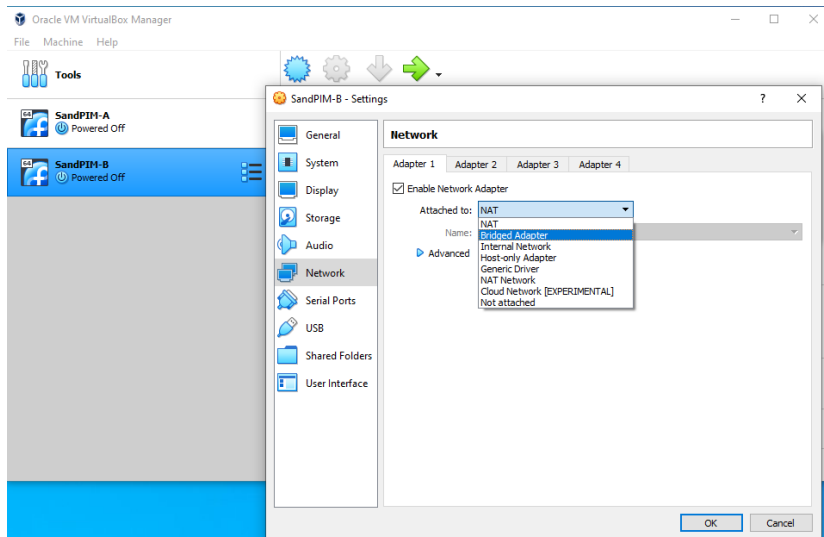
After a few seconds, you will get a prompt. Login as user **root** with the password you created



Shutdown the VM

```
Shutdown -h now
```

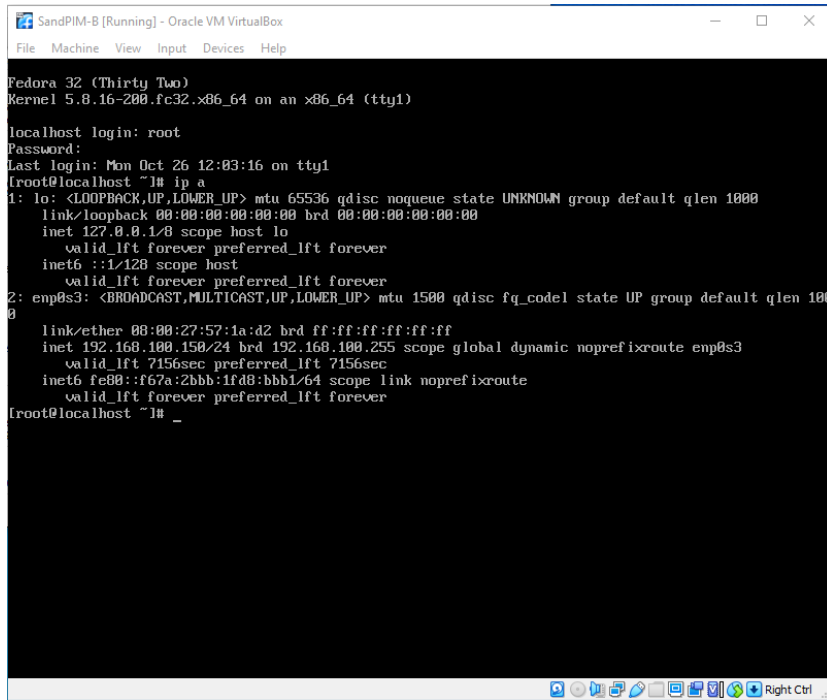
In the VM manager, highlight the new VM and click “Settings”. Then under “Network”, change “Adapter 1” from attached “NAT” to “Bridged Adapter” and click OK.



Start the VM and log in as root at the prompt.

Issue `ip a` the command to see what address was leased to the VM from your local DHCP server

`ip a`



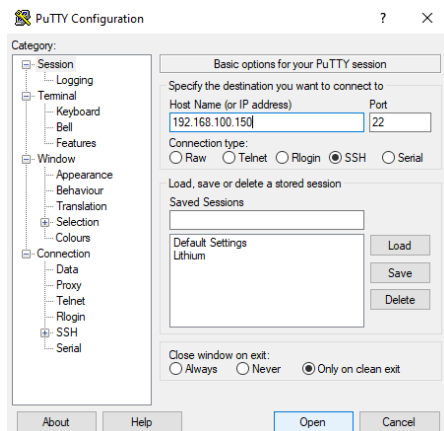
```
SandPIM-B [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Fedora 32 (Thirty Two)
Kernel 5.8.16-200.fc32.x86_64 on an x86_64 (tty1)

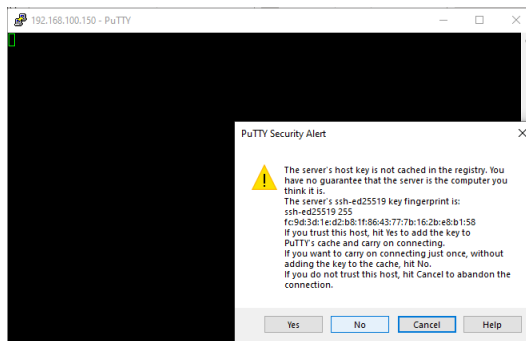
localhost login: root
Password:
Last login: Mon Oct 26 12:03:16 on tty1
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:00:27:57:1a:d2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.150/24 brd 192.168.100.255 scope global dynamic noprefixroute enp0s3
        valid_lft 7156sec preferred_lft 7156sec
    inet6 fe80::f67a:2bbb:1fd8:bbb1/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@localhost ~]# _
```

In my case, it was given 192.168.100.150

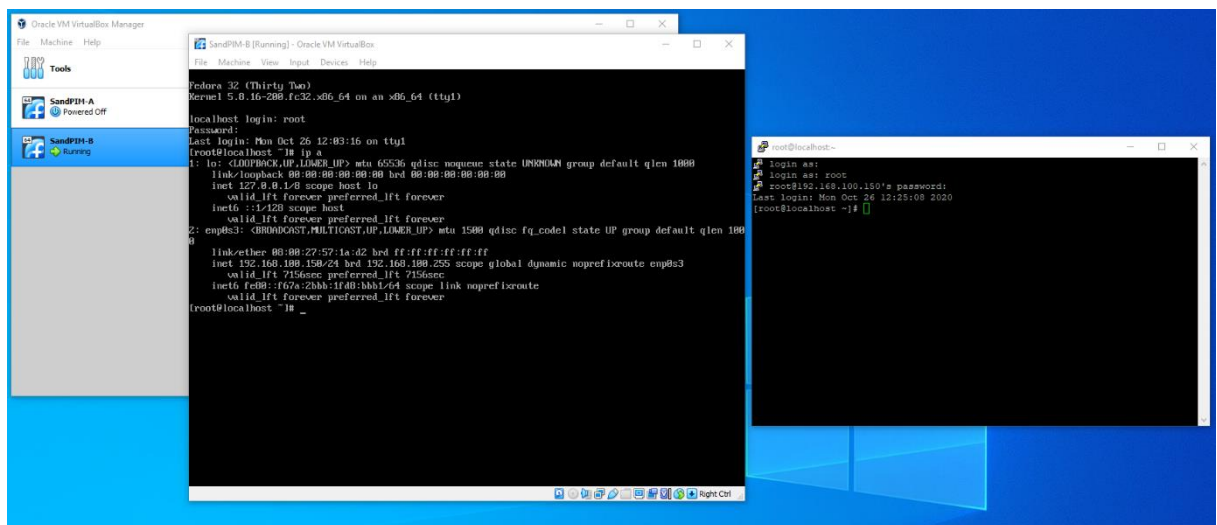
Start Putty and connect to the VM via SSH



You will get a security alert about unknown fingerprint. Click “Yes”



Login as root with the password you created. At this point you will have two console windows side-by-side (the VM's virtual monitor and the Putty SSH session). The remaining steps will be easier to accomplish in the SSH session because it allows easier pasting of command from this document to the command line. If you want to interact with the command line through the virtual monitor (without Putty), you can.



Install server software

Issue these commands one at a time at the prompt (allow each one to complete)

```
dnf update

dnf install lsof htop zip nano wget git cronie cronie-anacron

dnf install php php-mysqli

dnf install mariadb-server

systemctl start mariadb

systemctl enable mariadb

systemctl start httpd

systemctl enable httpd

systemctl start firewalld

systemctl enable firewalld

systemctl start crond.service

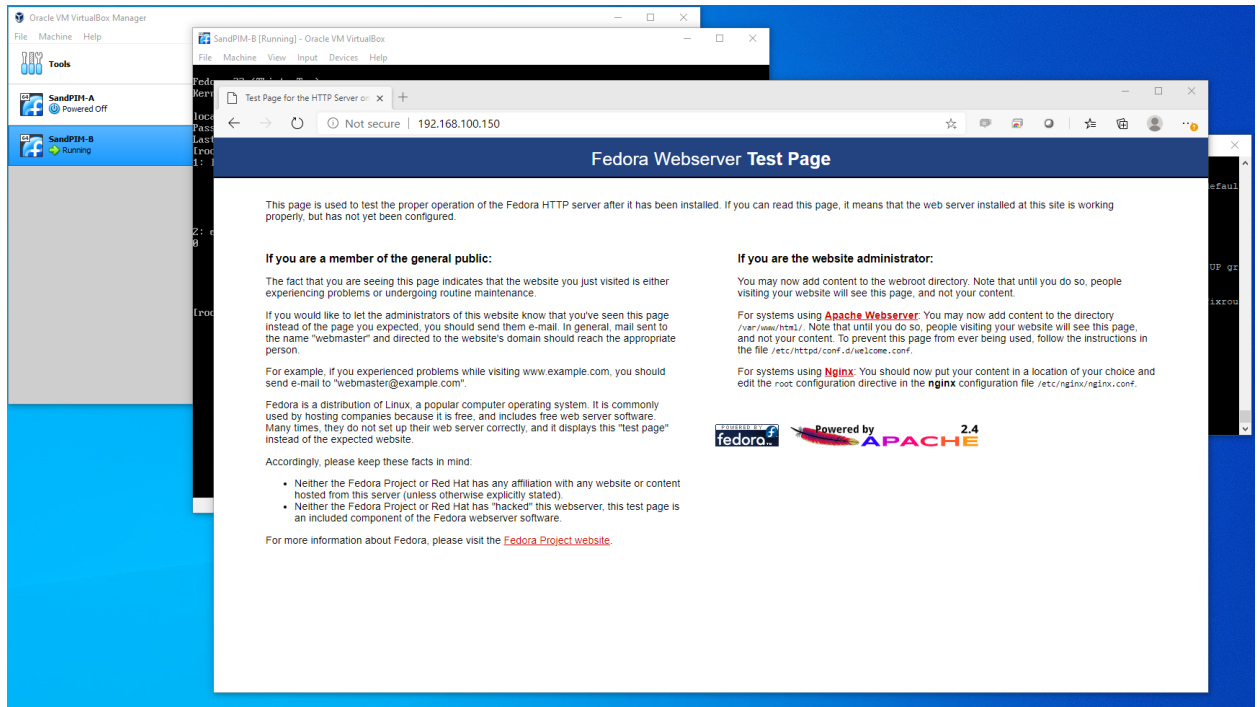
systemctl enable crond.service
```

Allow firewall exception for webservice

```
firewall-cmd --add-service=http

firewall-cmd --runtime-to-permanent
```

Test the Apache server by opening a web browser to the ip address leased to your VM. You should get a default Apache page.



Install the SandPIM php code from the github repository

Enter the document root for the Apache server

```
cd /var/www/html
```

Clone the github repository for the SandPIM project into

```
git clone https://github.com/autopartsource/sandpim.git
```

Set the password for the webservice to access the local database

```
nano /var/www/html/sandpim/class/mysqlClass.php
```

Edit the line in the file containing **`var $passwd = 'OsBBVrgJKw';`** to use the password that you used when creating the *webservice* database user

Ctrl+O will save the file, Ctrl+X will exit back to the Linux command line

Setup the local database

Log into the local database with the command-line mysql client

```
mysql
```

At the mysql prompt (replace *myPassword* with one that you choose):

```
create user 'webservice'@'localhost' identified by 'myPassword';  
  
grant select, insert, update, delete, create, drop, lock tables on *.* to  
'webservice'@'localhost' with grant option;  
  
exit
```

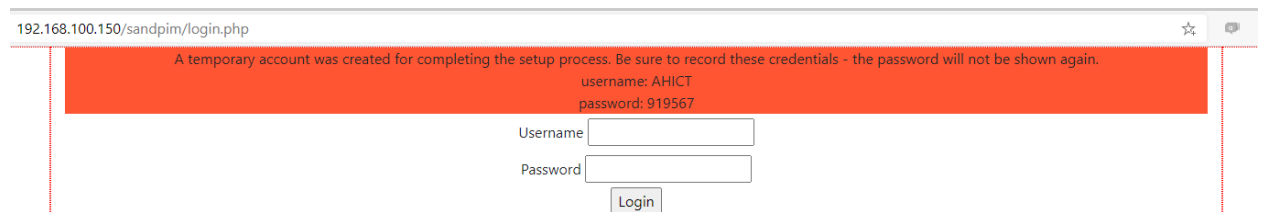
Create the SandPIM database by running the *setup.php* script

```
cd /var/www/html/sandpim  
  
php setup.php
```

Log into the web UI

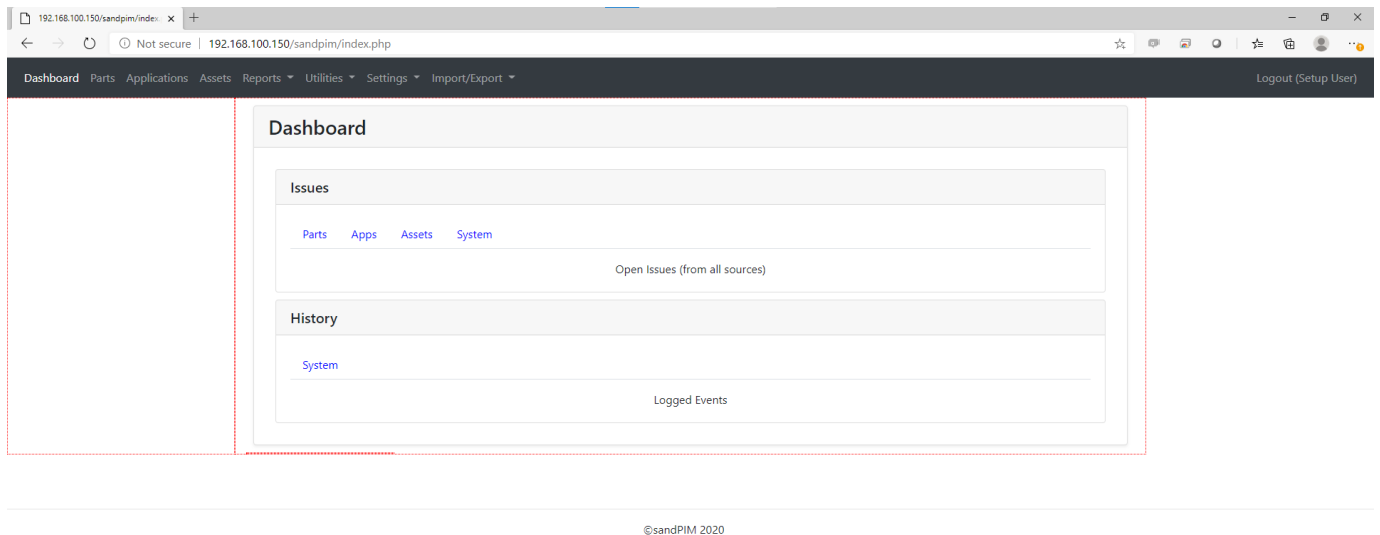
With a web browser, goto http://<VM_ip_address>/sandpim

You will see a login form and the credentials of the “Setup User”. The username and password are randomly generated values for your installation. In other words, the ones show in the screenshot below will not work on your instance.



Write these credentials down – you will need them and they are not shown again.

Login with the credentials shown. You should get to the dashboard page.



Import AutoCare databases (VCdb, PCdb, PAd Qdb)

Download and extract the most recent association files from the VIP server directly into the VP.
Note: you could use Filezilla to push these file into the VM if you don't have an FTP account on the AutoCare server. (the command is one line – word-wrap makes it look like multiples)

```
cd /root

wget --ftp-user=<your_username> --ftp-password=<your_password> --no-check-
certificate
ftp://52.168.10.67/download_vcdb/Complete/MySQL/AAIA%20VCdb2009%20MySQL%20Comple
e%20VCDB%2020200925.zip

wget --ftp-user=<your_username> --ftp-password=<your_password> --no-check-
certificate
ftp://52.168.10.67/download_pcdb/MySQL/AAIA%20PCdb%20MySQL%2020200925.zip

wget --ftp-user=<your_username> --ftp-password=<your_password> --no-check-
certificate
ftp://52.168.10.67/download_qdb/MySQL/AAIA%20Qdb%20MySQL%2020200925.zip

wget --ftp-user= <your_username> --ftp-password=<your_password> --no-check-
certificate
ftp://52.168.10.67/download_padb/MySQL/AAIA%20PAdb%20MySQL%2020200925.zip

unzip AAIA\ VCdb2009\ MySQL\ Complete\ VCDB\ 20200925.zip

unzip AAIA\ PCdb\ MySQL\ 20200925.zip

unzip AAIA\ Qdb\ MySQL\ 20200925.zip

unzip AAIA\ PAdb\ MySQL\ 20200925.zip
```

Using the MySQL CLI, create empty version-named and generic reference databases

```
create database vcdb20200925;

grant select on vcdb20200925.* to 'webservice'@'localhost';

create database vcdb;

grant select on vcdb.* to 'webservice'@'localhost';

create database pcdb20200925;

grant select on pcdb20200925.* to 'webservice'@'localhost';

create database pcdb;

grant select on pcdb.* to 'webservice'@'localhost';

create database qdb20200925;

grant select on qdb20200925.* to 'webservice'@'localhost';

create database qdb;

grant select on qdb.* to 'webservice'@'localhost';

create database padb20200925;

grant select on padb20200925.* to 'webservice'@'localhost';

create database padb;

grant select on padb.* to 'webservice'@'localhost';
```

Exit back to the Linux CLI to import the SQL files from AutoCare into their respective databases - both the generic and specific databases. Note, the VCdb import can take several minutes.

```
mysql vcdb20200925 < 'AAIA VCdb2009 MySQL Complete VCDB 20200925.sql'

mysql vcdb < 'AAIA VCdb2009 MySQL Complete VCDB 20200925.sql'

mysql pcdb20200925 < 'AAIA PCdb MySQL 20200925.sql'

mysql pcdb < 'AAIA PCdb MySQL 20200925.sql'

mysql qdb20200925 < 'AAIA Qdb MySQL 20200925.sql'

mysql qdb < 'AAIA Qdb MySQL 20200925.sql'

mysql padb20200925 < 'AAIA PAdb MySQL 20200925.sql'

mysql padb < 'AAIA PAdb MySQL 20200925.sql'
```

Using the MySQL CLI, grant access and catalog the new versions of the reference databases and fill the “*favorite Makes*” table from the VCdb.

```
use pim;

insert into autocare_databases values('vcdb20200925','vcdb','2020-09-25');

insert into autocare_databases values('pcdb20200925','pcdb','2020-09-25');

insert into autocare_databases values('qdb20200925','qdb','2020-09-25');

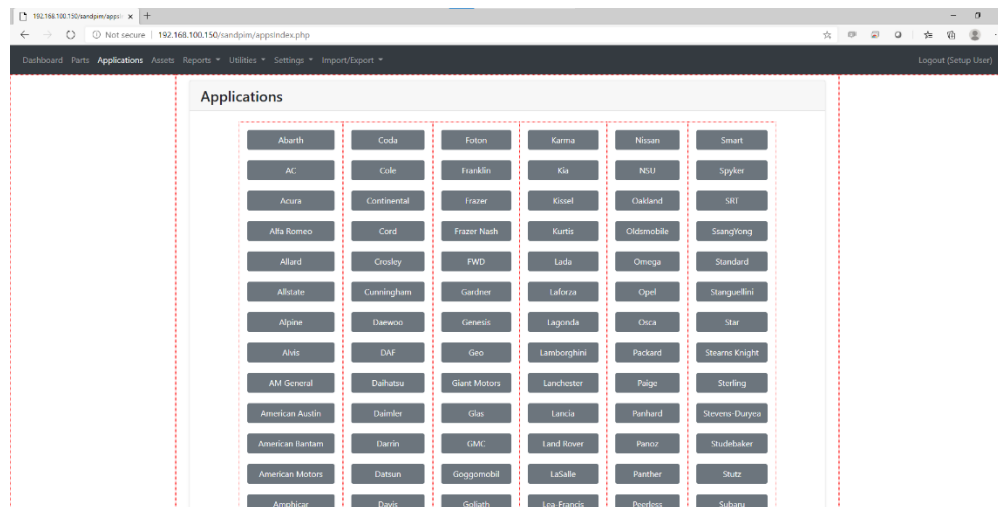
insert into autocare_databases values('padb20200925','padb','2020-09-25');

use vcdb;

INSERT INTO pim.Make (MakeID,MakeName) SELECT distinctrow Make.MakeID,
Make.MakeName FROM BaseVehicle,Make,Model where
BaseVehicle.MakeID=Make.MakeID and BaseVehicle.ModelID=Model.ModelID and
Model.VehicleTypeID in (5,6,7,2187) order by MakeName;
```

*Note: If you only want Passenger Cars in the PIM’s *Favorite Makes* cache, include only “5” in the above SQL statement. VehicleTypeID 6, 7 and 2187 represent *Truck, Van* and *Medium/Heavy Truck* (respectively)

Verify that the VCdb Makes cache was populated by clicking on “Application” in the Web navigation bar at the top of the SandPIM UI.



Lock-down the database

Exit back out to the Linux command line, invoke the lock-down wizard for mysql

```
mysql_secure_installation
```

Unix socket=N

Set a new root password =Y (then follow prompt to enter and re-enter)

Disable remote login by root=Y

Remove test database=Y

Reload privileges=Y

***Optional (depending on your level or paranoia): Invoke the MySQL command line and remove the “create” privilege from the webservice user**

```
use pim
```

```
REVOKE create ON * FROM 'webservice'@'localhost';
```

Setup background (cron) processing of for housekeeping jobs

Edit the schedule jobs list (“crontab”) with the *nano* text editor

```
nano /etc/crontab
```

Add these lines to the file. Then Ctrl+O to save, Ctrl-X to exit. Note: The font is tiny to illustrate that there are no line-breaks in the content.

```
*/5 * * * 0,1,2,3,4,5,6 root /usr/bin/php /var/www/html/sandpim/processACESupload.php &> /dev/null
*/5 * * * 0,1,2,3,4,5,6 root /usr/bin/php /var/www/html/sandpim/processACESexport.php &> /dev/null
*/5 * * * 0,1,2,3,4,5,6 root /usr/bin/php /var/www/html/sandpim/processFlatAppsExport.php &> /dev/null
* * * * 0,1,2,3,4,5,6 root /usr/bin/php /var/www/html/sandpim/auditor.php &> /dev/null
```

Allow Apache to write to the uploads and downloads directories and make connections (for downloading AutoCare resource list)

(under construction)

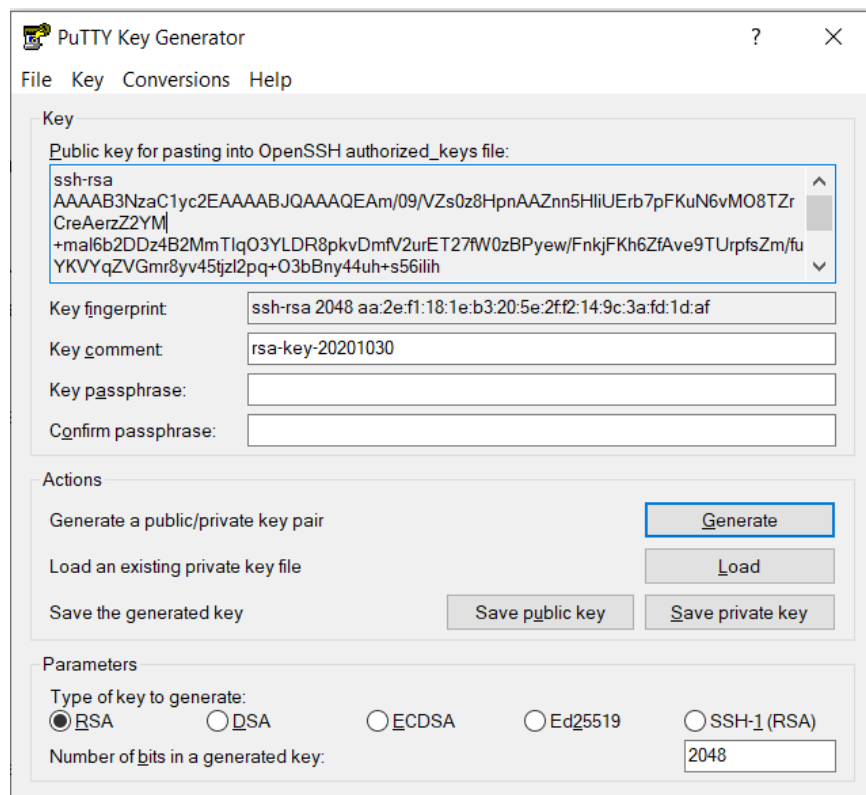
```
setsebool -P httpd_can_network_connect 1
semanage fcontext -a -t httpd_sys_rw_content_t
'/var/www/html/autocaredownloads'

restorecon -v '/var/www/html/autocaredownloads'
```

```
chown apache:apache /var/www/html/autocaredownloads
chmod 744 /var/www/html/autocaredownloads
```

(optional) If you are going to be logging in via SSH on a regular basis, SSH public key authentication will speed up your workflow.

Use PuTTYgen to make an RSA key pair



Save the private key locally in a safe place on your PC (in Documents, etc.) and name something along the lines of "SandPIMroot.ppk". This is the one that you need to keep secret.

Save the public key in a convenient place on your PC and name it something along the lines of "SandPIMroot public key.txt". This file is not sensitive and you will only need it again if you want to setup a new authorization with another server. The server keeps this key to identify/verify your identity at each login. It does this by encrypting a nonce

with this public key and giving it to your client to decrypt with the private key and send back.

Make a hidden directory called `“ssh”` directory in `/root` (if it does not already exist)

```
cd /root
mkdir .ssh
chmod 700 .ssh
cd .ssh/
nano authorized_keys
```

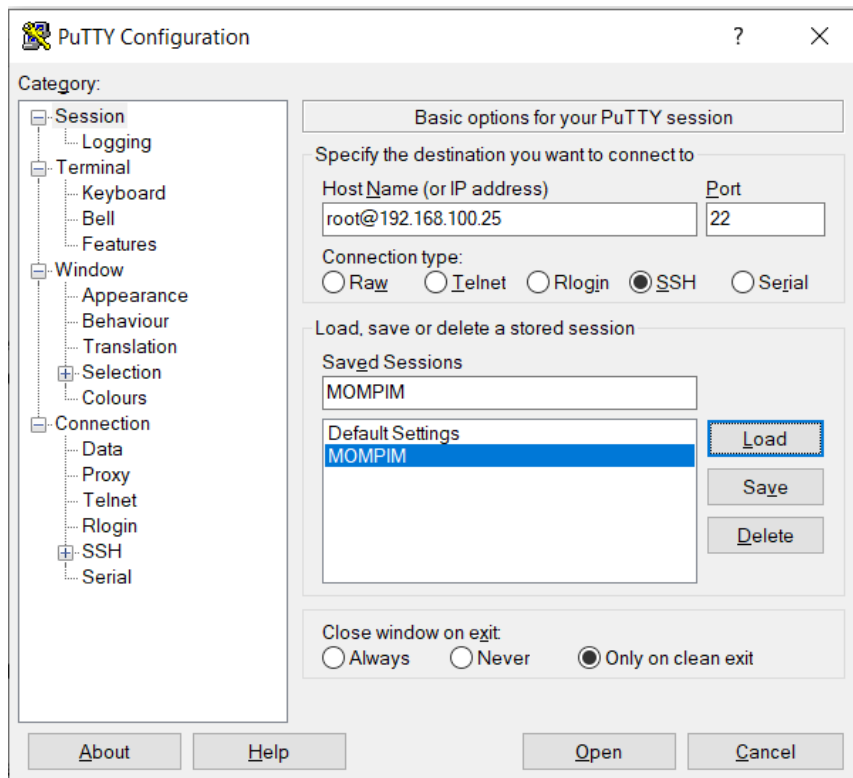
Using nano, paste the public key into the `authorized_keys` file, Ctrl+O to save, Ctrl+X to exit. The key needs to live on a single line and look something like this (without the line-brakes and with a space after `ssh-rsa`)

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAoW83cyWOos+mQItm+f7jhm3uHg+jYE9La4qmlvZPRVGX6
ZmiA3KRgXCA8JjTdxChQ1NWwzrmYCziQYnCiQ3Vz03LoORqlauC5sx5pAOHoYgEmqLDFrBuLK
rxwaj2Qrb7ebT/WsqD5rE72v2LxdtG2nHQnybbaVIdnaeZ/r4tjsSd/zqZSqEdKKAoOEpdgGt
JGSYxdDbIIujp0m9EJj98tPKBDayxVYIBhbsBx+KfvZaUPdESAIVmXZ4KvHUI1DHTGHCR9mFq
NQ9vhZZfA/09D/LFXFZhpmaMv+Vb76HYJvmlLdBdy+PSYpX9ZaYO4RrxZxacapG8/35byS/9Ud
fEQ2w==
```

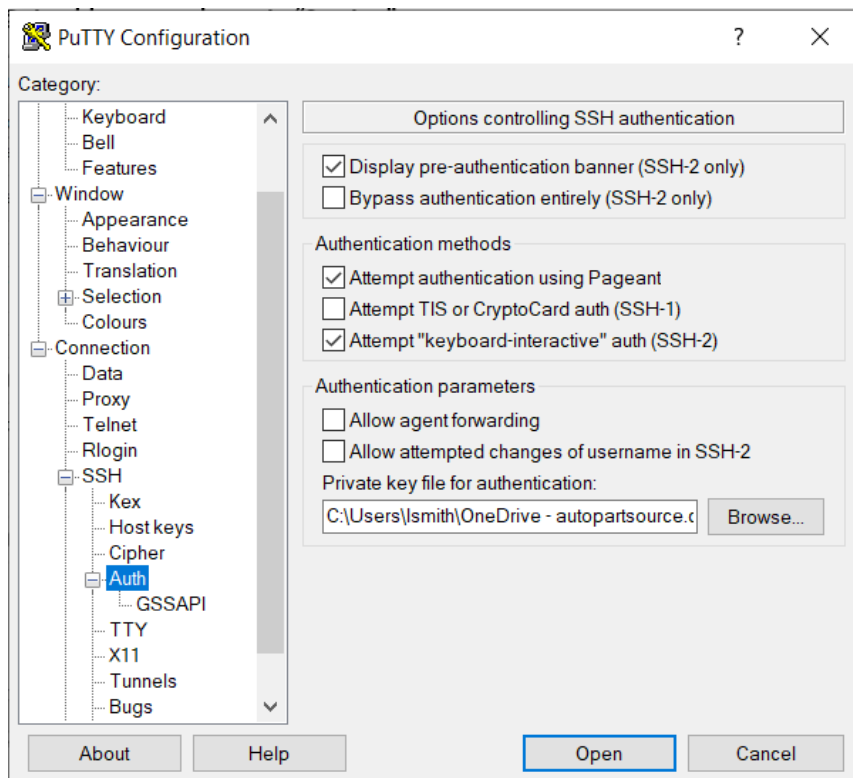
Change the permissions on the file so that only root can read and write it

```
chmod 600 authorized_keys
```

Configure PuTTY to use your private key to log into the server. Specify the Host as `root@<ipaddress>` on the main “Session” screen. Give the session a name (I called mine “MOMPIM”)



Provide PuTTY the path to your private key file in the Connection/SSH/Auth screen



Return to the main “Session” screen and click “Save” to save to configuration.

Disable the ability to login via simple password authentication with SSH
(under construction)