Table of Contents

Setting Up an SSH Console

SSH stands for Secure SHell, and is a secure method to connect to a remote computer over a network. There are many advantages to using an SSH console rather than say the shell facility in the FreeNAS GUI.

The SSH console is a window that has a scrolling function which means you can go back and view the output in the console. You can also select large bodies of text and copy and paste them. This can be particularly useful when trying to get help from someone as they need to see what you have done. It is also useful when compiling data (i.e. SMART test data).

An SSH console is also very secure in two ways. Firstly it can be configured to require a Public/Private key and a password before you can log in to the session and the server. Secondly the connection between the server and the client is encrypted. This means any information that goes between the two cannot be read directly.

Configure SSH in FreeNAS

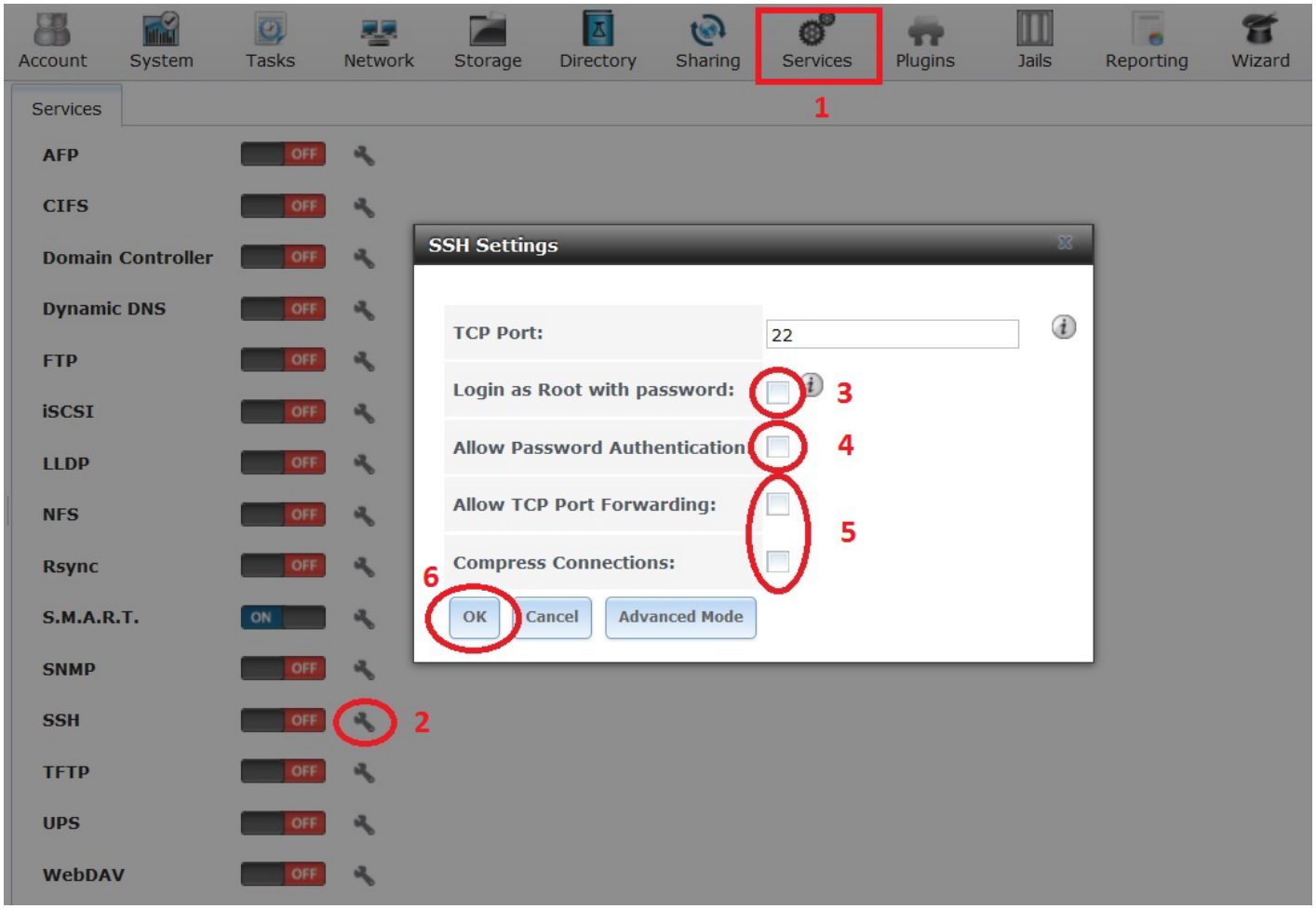
Open your web browser and type in the IP address of the FreeNAS web GUI that you noted down earlier (Fester used 192.168.0.58).

The web GUI will present itself and ask for the login details. Enter the username which is **root** (1) and your password (2) and click the “Log In” button (3).

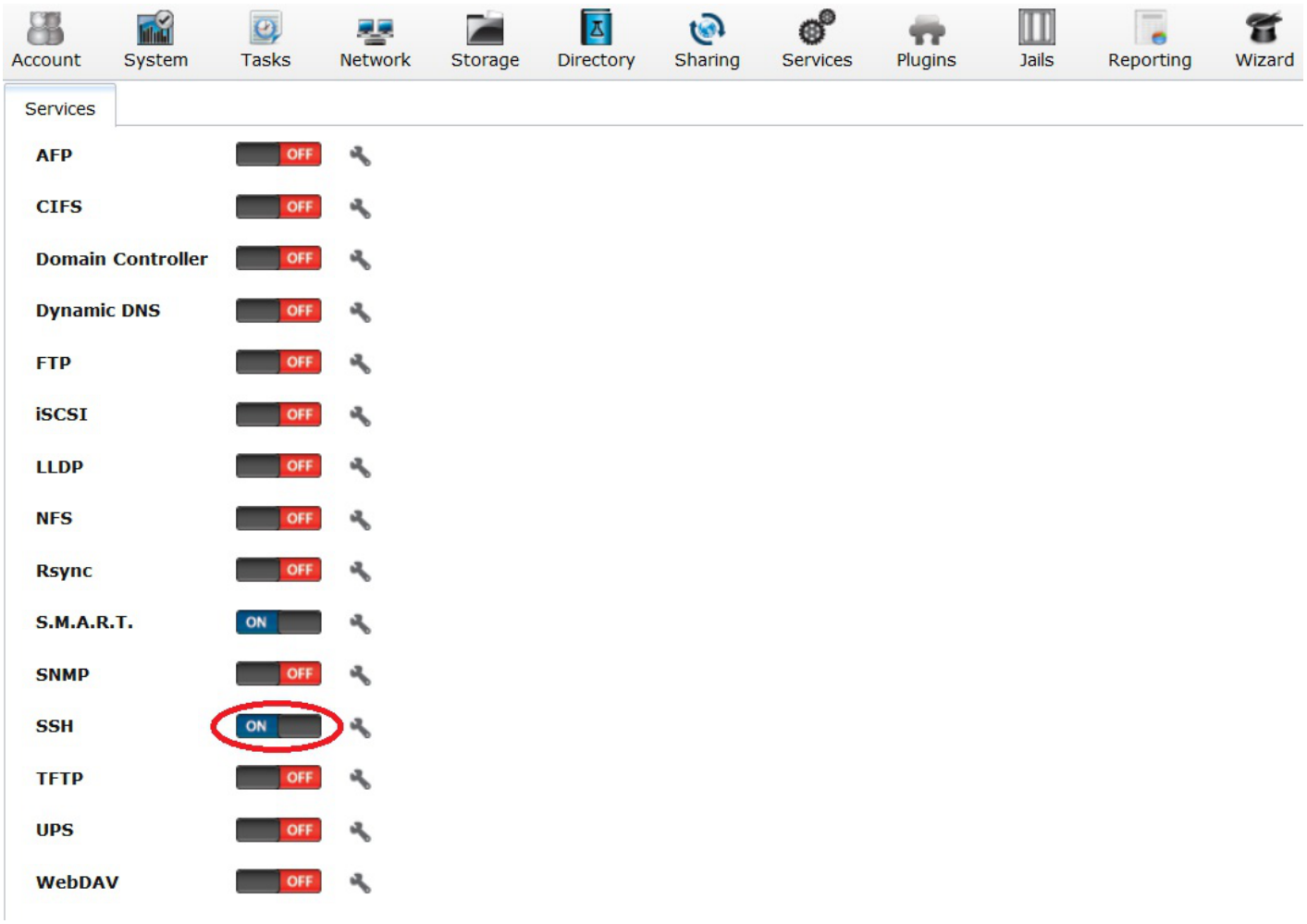
Now you are logged into FreeNAS.

- Now navigate to the “Services” page (1).
- Click on the tiny spanner icon next to “SSH” (2).
- If you do not want the root user to be able to log in using only a password (i.e., if you want to require a public key, or if you don't want the root user to be able to log in remotely at all), uncheck the “Login as Root with password” tick box (3).

- If you want to require public key authentication, uncheck the “Allow password authentication” tick box (4).
- Make sure the remaining tick boxes are unchecked (5).
- Now click the “OK” button (6).

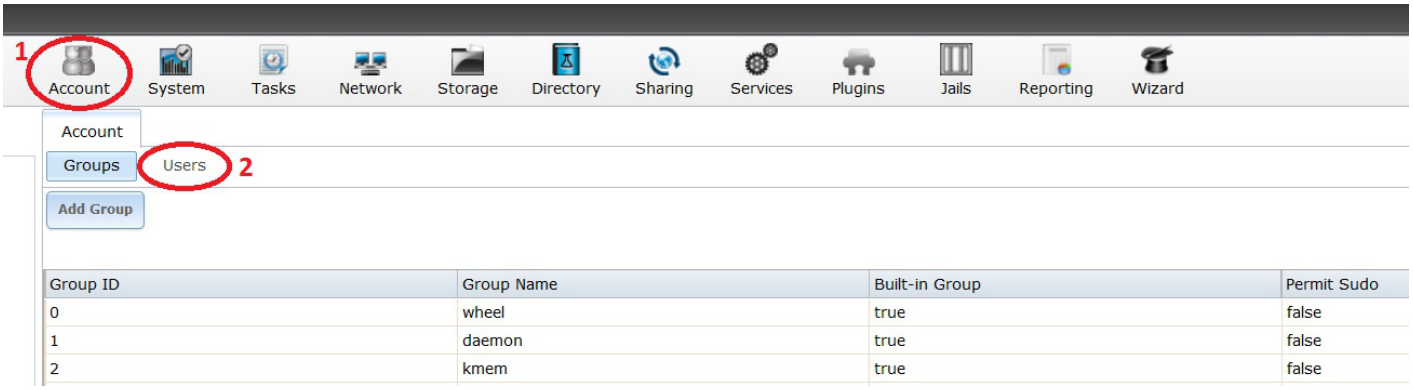


Now turn on the SSH service.

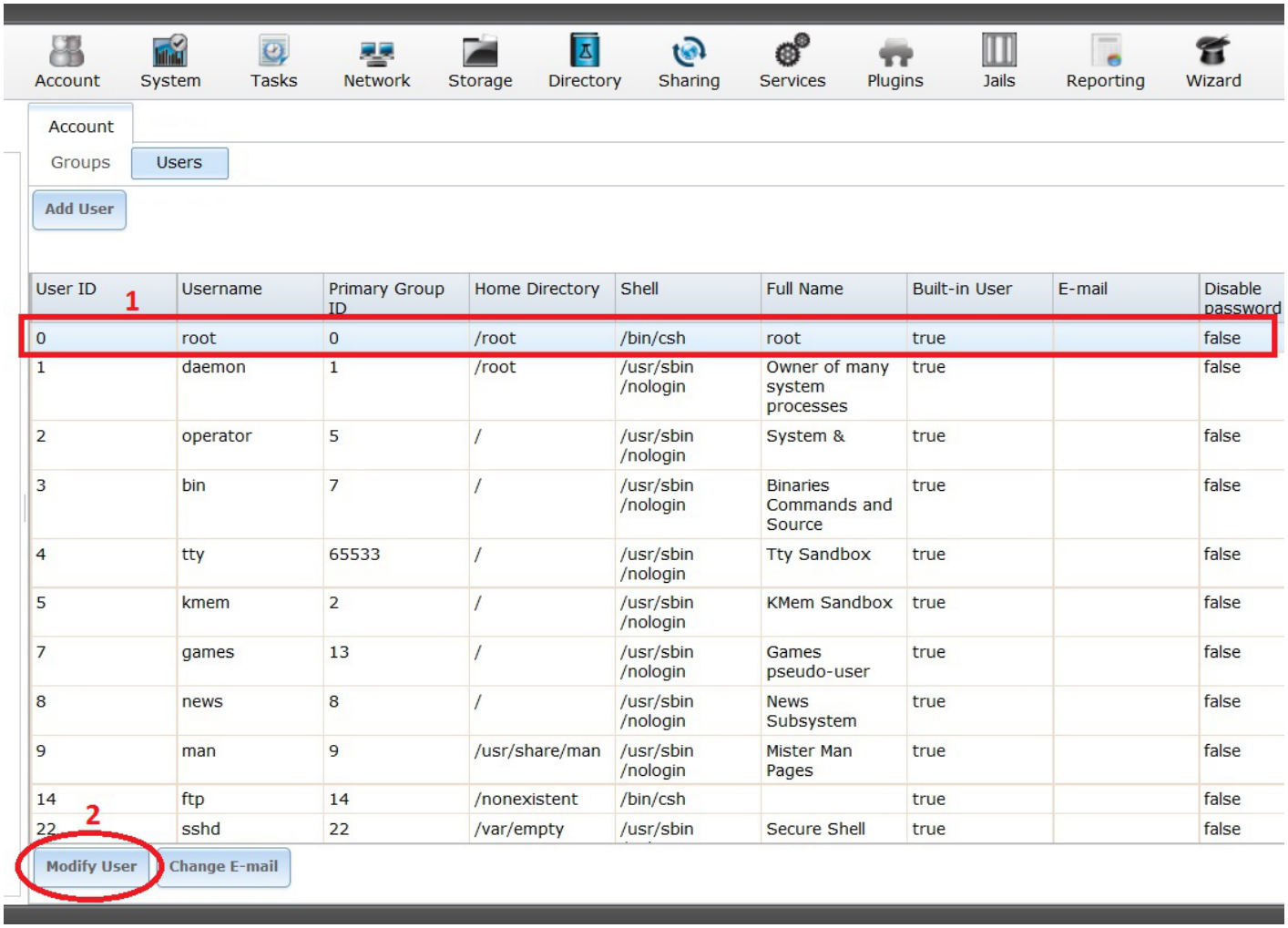


Specify SSH Public Key For a User

If you have required public key authentication in the SSH configuration, you'll need to tell FreeNAS what the public key is for each user who will be connecting via SSH. To do that, Navigate to the “Account” page by clicking on the Account icon (1). Now click on the “Users” button (2).



Now select the “root” user account (1) (it will turn blue when selected) and click on the “Modify User” button (2).

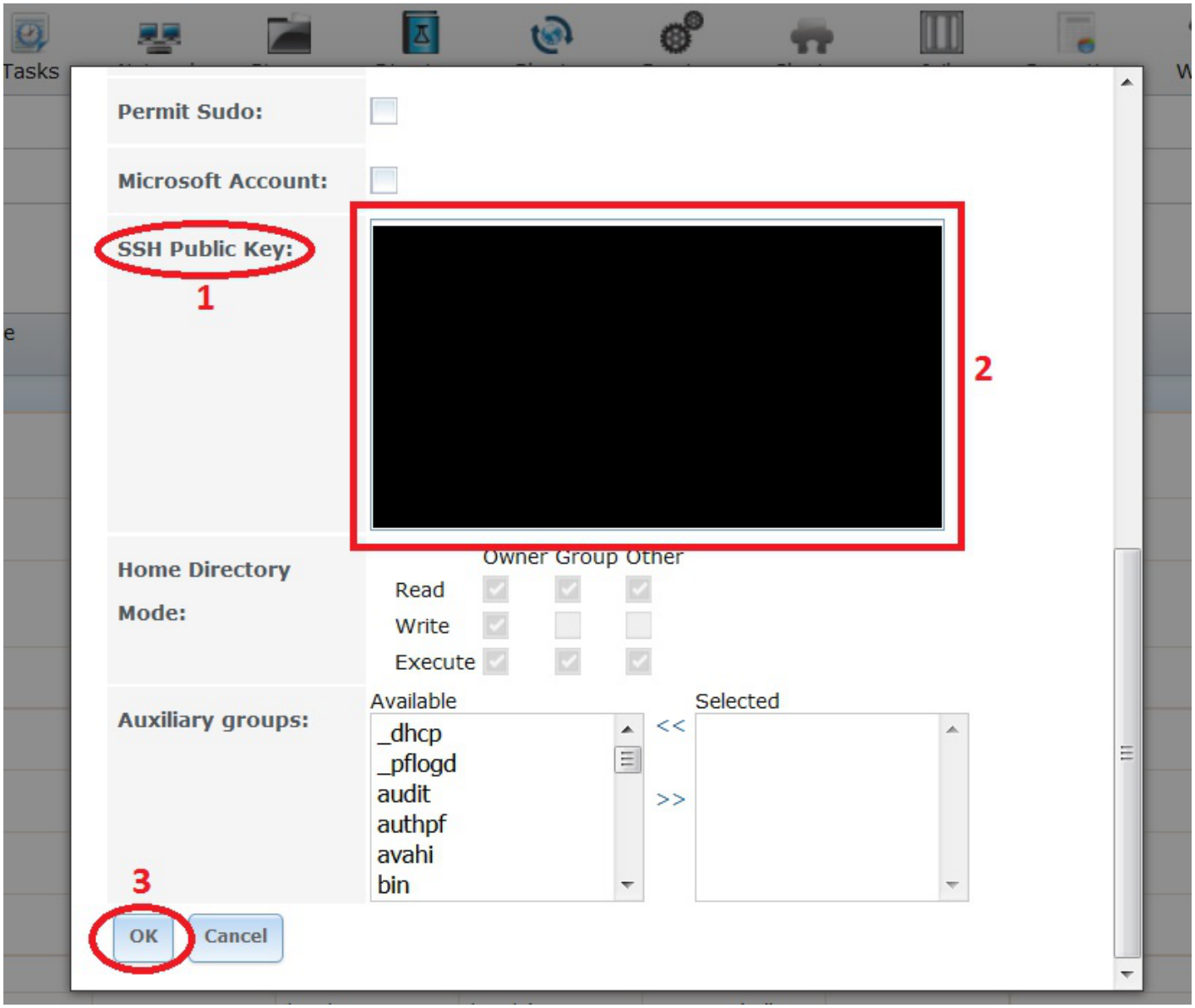


User ID	Username	Primary Group ID	Home Directory	Shell	Full Name	Built-in User	E-mail	Disable password
0	root	0	/root	/bin/csh	root	true		false
1	daemon	1	/root	/usr/sbin /nologin	Owner of many system processes	true		false
2	operator	5	/	/usr/sbin /nologin	System &	true		false
3	bin	7	/	/usr/sbin /nologin	Binaries Commands and Source	true		false
4	tty	65533	/	/usr/sbin /nologin	Tty Sandbox	true		false
5	kmem	2	/	/usr/sbin /nologin	KMem Sandbox	true		false
7	games	13	/	/usr/sbin /nologin	Games pseudo-user	true		false
8	news	8	/	/usr/sbin /nologin	News Subsystem	true		false
9	man	9	/usr/share/man	/usr/sbin /nologin	Mister Man Pages	true		false
14	ftp	14	/nonexistent	/bin/csh		true		false
22	sshd	22	/var/empty	/usr/sbin	Secure Shell	true		false

The modify user window should now pop up. Scroll down till you come across the “SSH Public Key:” entry (1).

Now right click in the blank box next to it and paste in the previously copied public key (2).

Now click the “OK” button (3).



Setting up PuTTY in Windows

Modern operating systems ship with an SSH client installed. Unfortunately, Windows is still not a modern operating system in this regard, so a third-party client will need to be used. Popular clients include Bitvise and PuTTY.

Public Key Authentication in PuTTY

Switch on and boot up a personal computer that is part of your private network (if you use it to connect to the internet then this will probably work).

Download PuTTY and PuTTYgen to your personal computer (not the server).

Install PuTTy and PuTTYgen under an administrator's account or right click on their respective installation programs and run as an administrator.

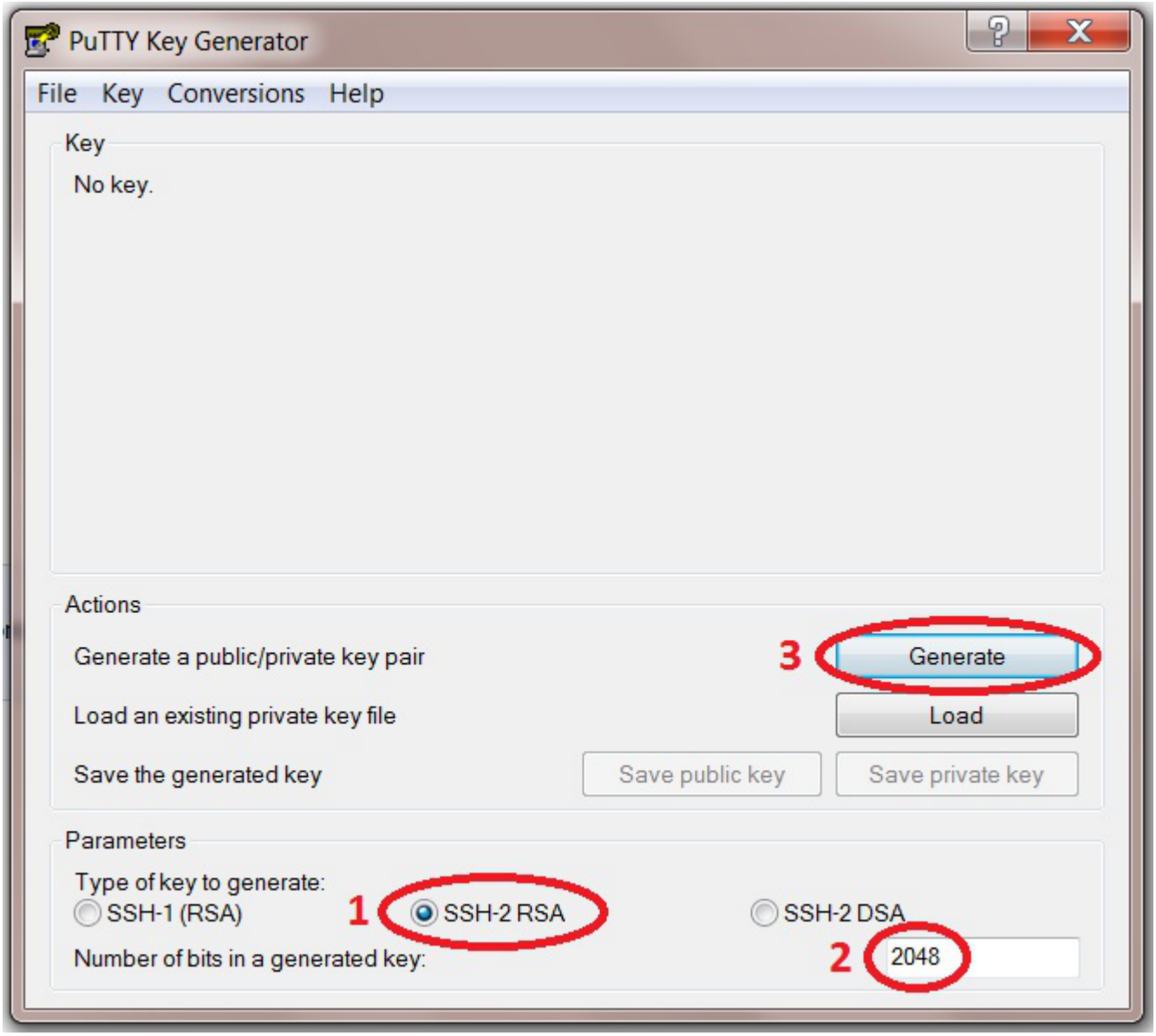
Generating the keypair

When installed run PuTTYgen under an administrator's account or right click on the program and run as an administrator.

When the PuTTYgen window appears check "SSH-2 RSA" is selected (1), if it isn't select it.

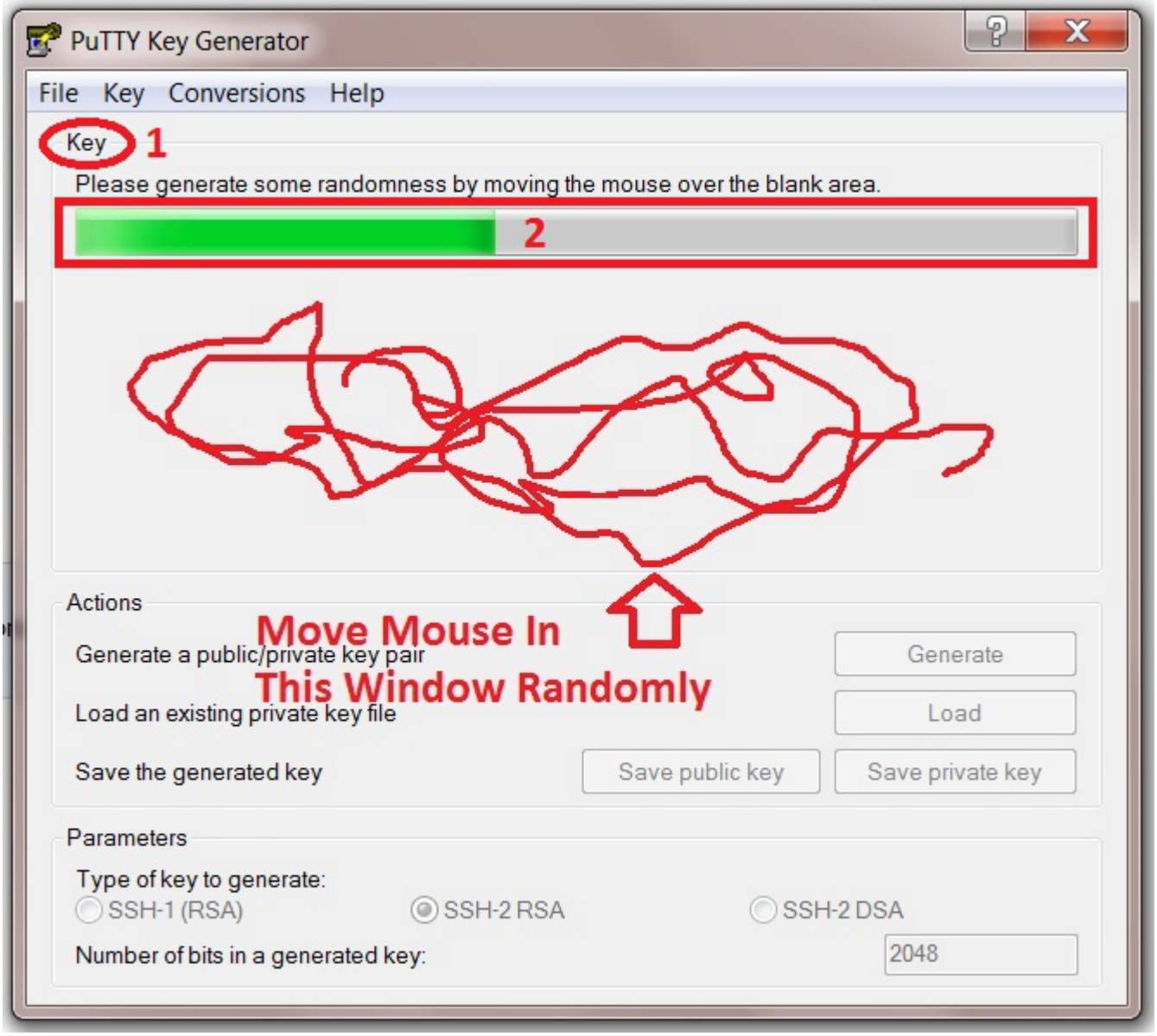
Next check the "Number of bits in a generated key:" is set to 2048 (2).

Now click the "Generate" button (3).



Now move your mouse in a random way within the box labelled "Key" (1) in PuTTYgen until the green bar

fills up (2).



When the green bar is full the key will be generated and a new screen will appear.

In the “Key comment” text box (1) type a comment which will help you identify the key.

Now type in a password for the private key in the “Key passphrase” text box (2), remember it as this will be needed later (Fester just used **test** again).

Retype the password into the “Confirm passphrase” text box (3).

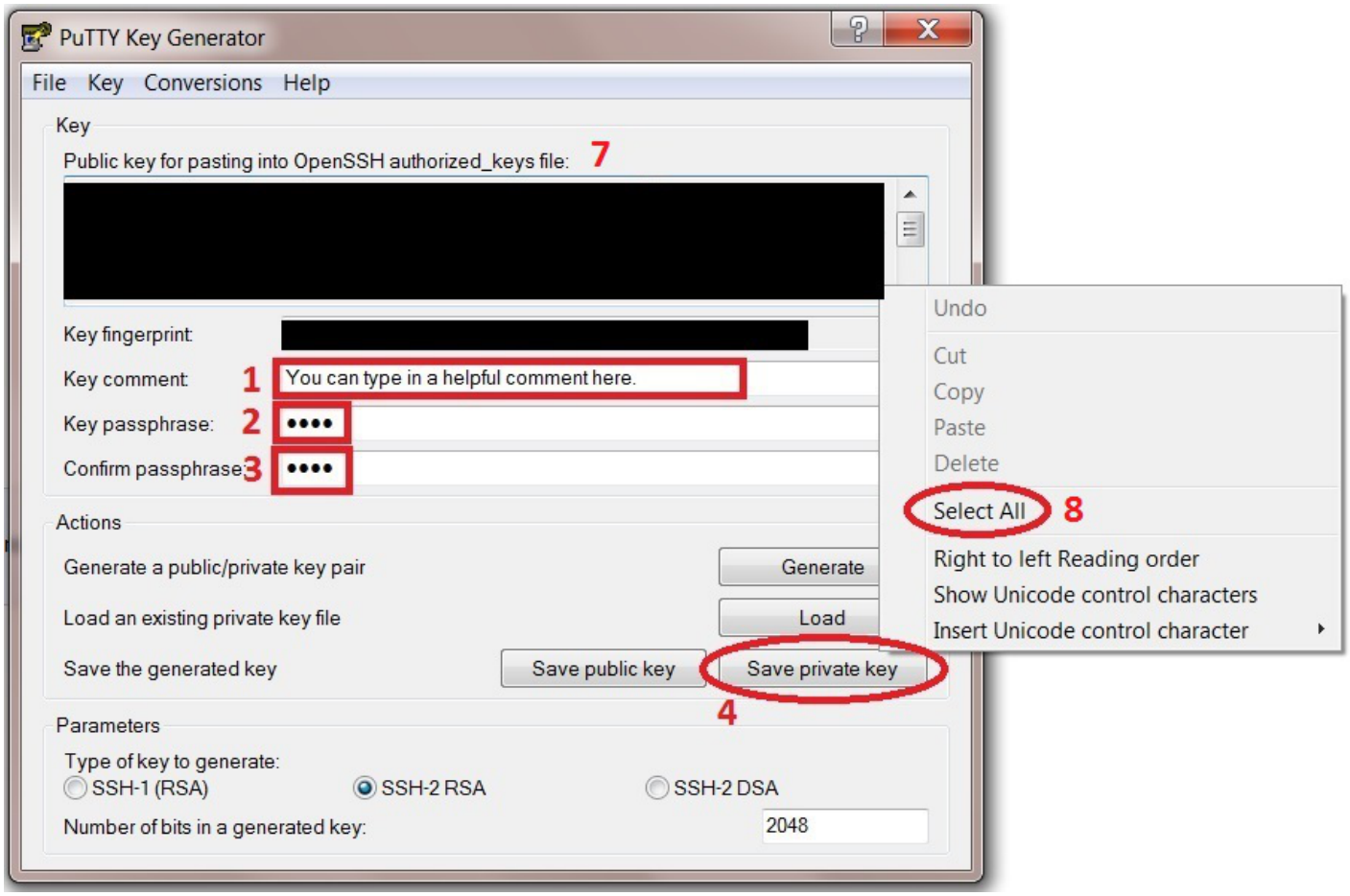
Now save the private key by clicking on the “Save private key” button (4).

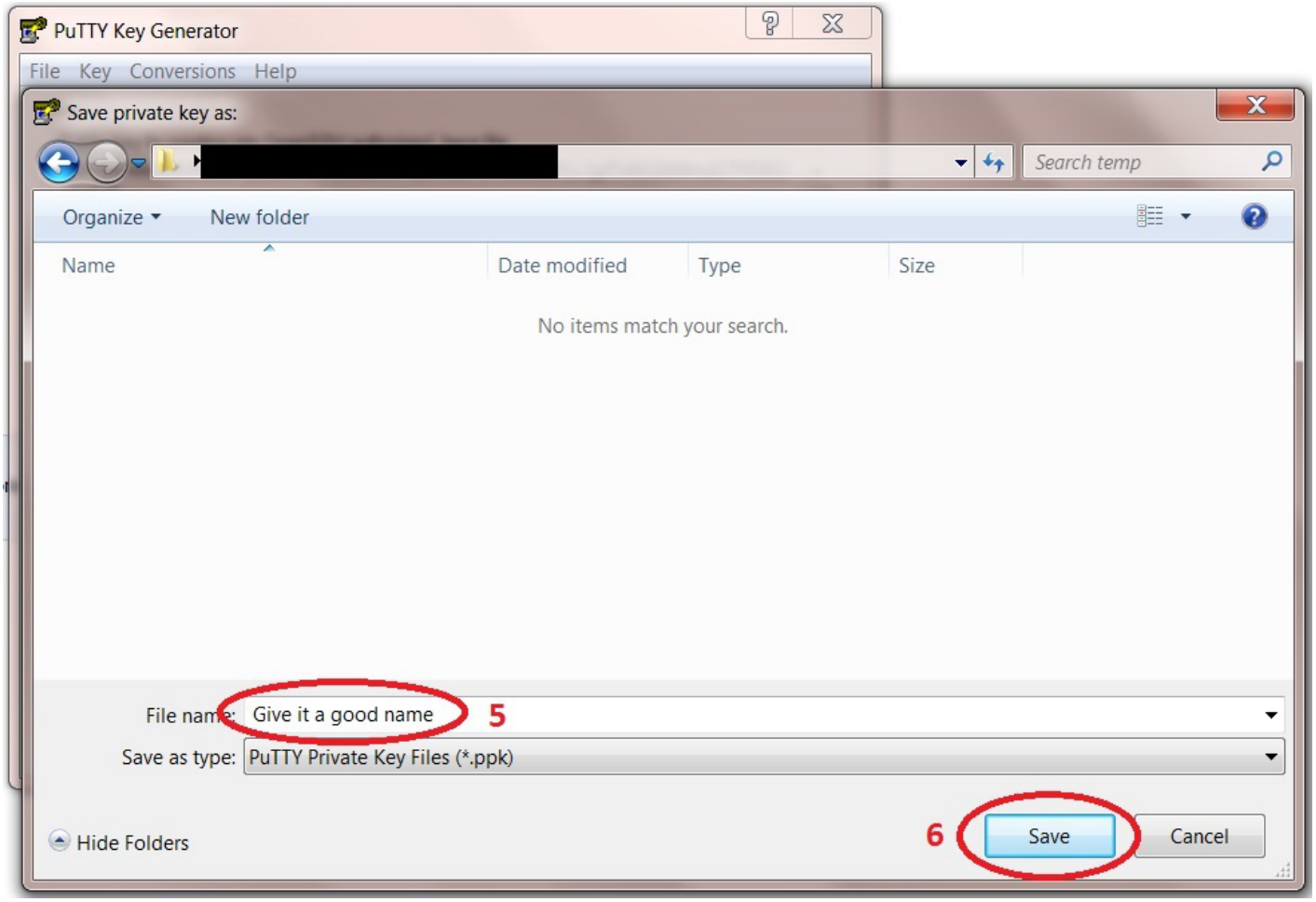
An additional window will pop up, navigate to where you would like to save the key, give it a name (5) and click the “Save” button (6). Save it somewhere convenient as this will be needed soon.

Now right click in the “Public key for pasting into OpenSSH authorized_keys file:” window (7) and from

the pop up submenu chose “Select All” (8).

The text within this window should become highlighted. Now right click again in this window as you did a moment ago and from the pop up submenu this time select “Copy”.





Configuring the connection

Now run PuTTY under an administrator's account or right click on it and run as an administrator.

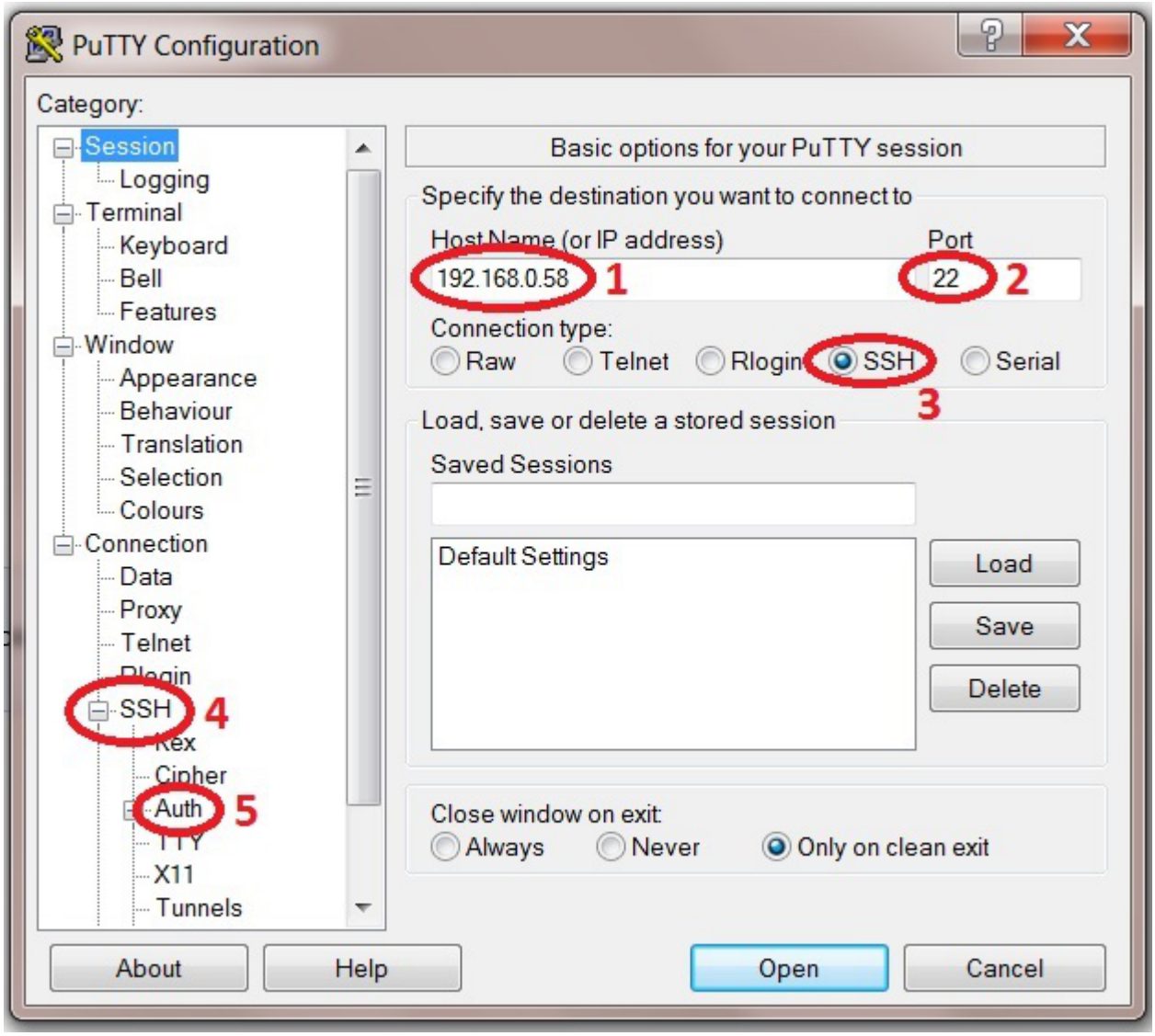
In the "Host Name or (IP address)" box (1) type in the IP address of the FreeNAS web GUI (Fester's was 192.168.0.58).

Check the port number in the "Port" box (2) is set to 22.

The "Connection type:" should be set to SSH (3).

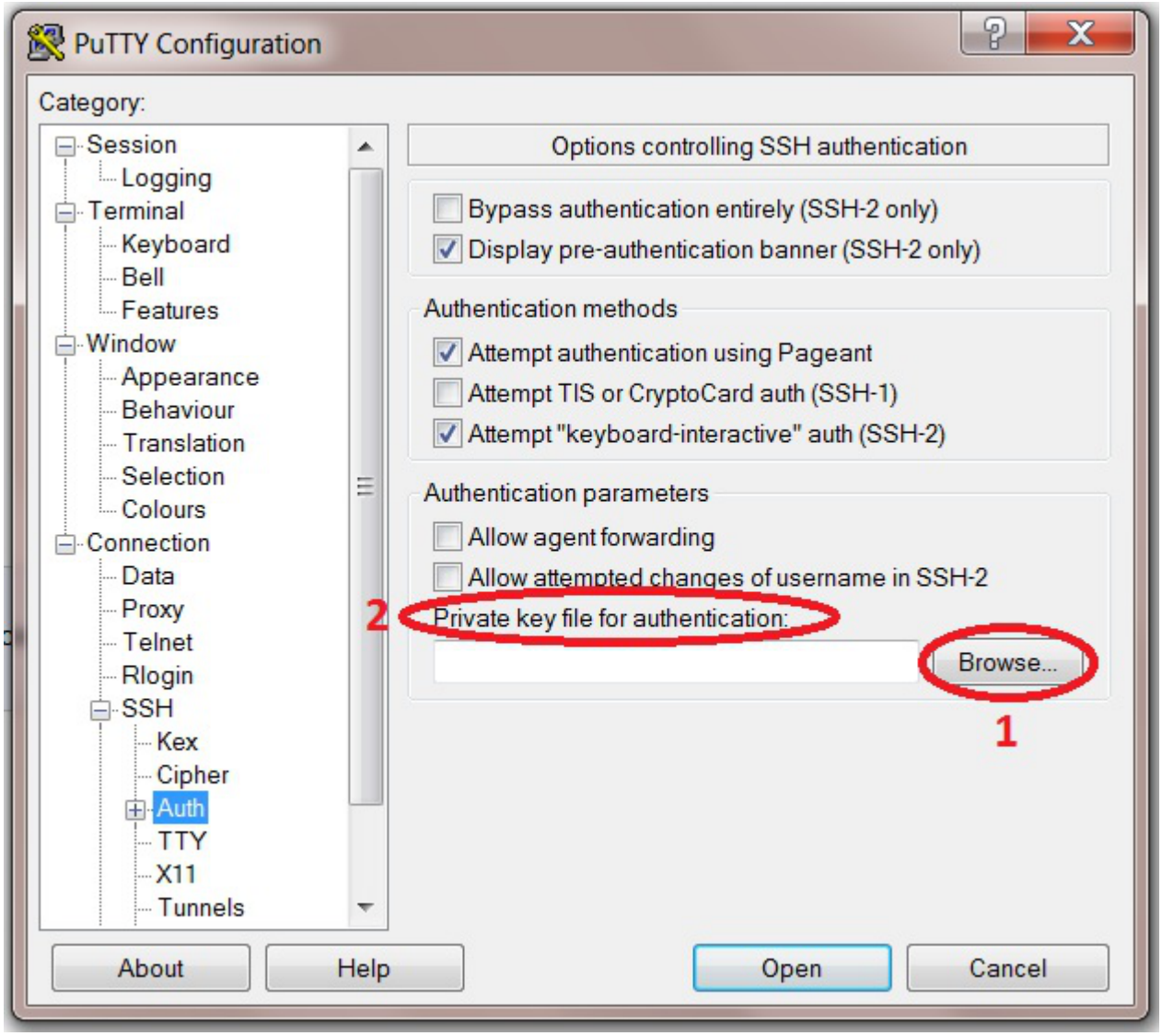
Now in the "Category" window click the small plus symbol "+" next to SSH (4). This should open up this section to reveal subcategories.

Then click on "Auth" (5), not the "+" sign but the actual text itself.

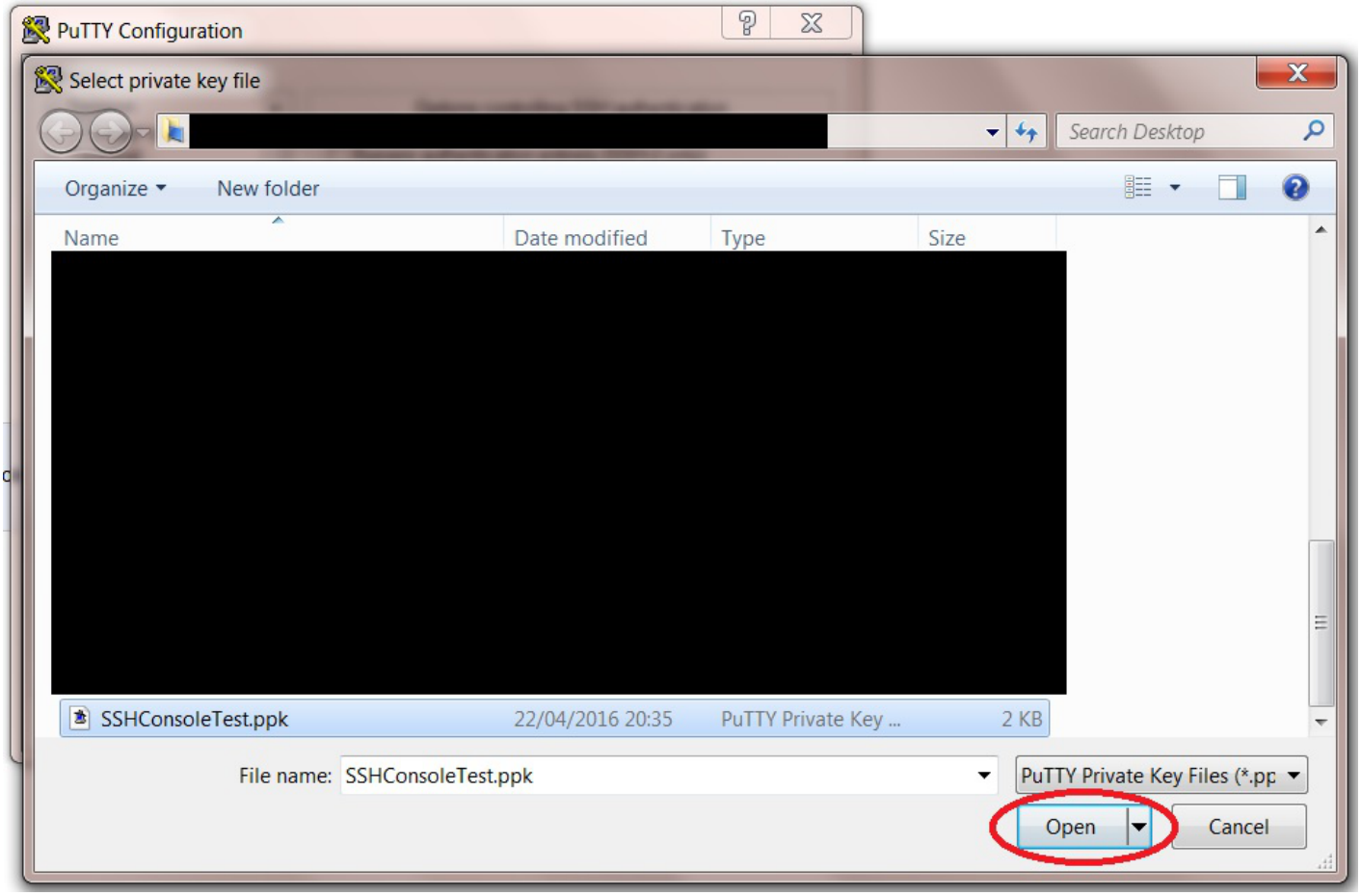


This should take you to a different screen.

On this screen click the “Browse” (1) button next to the “private key file for authentication:” (2).



This will bring up a window in which you can load in the private key into PuTTY. Navigate to where you stored the private key, click on it and then click the “Open” button.



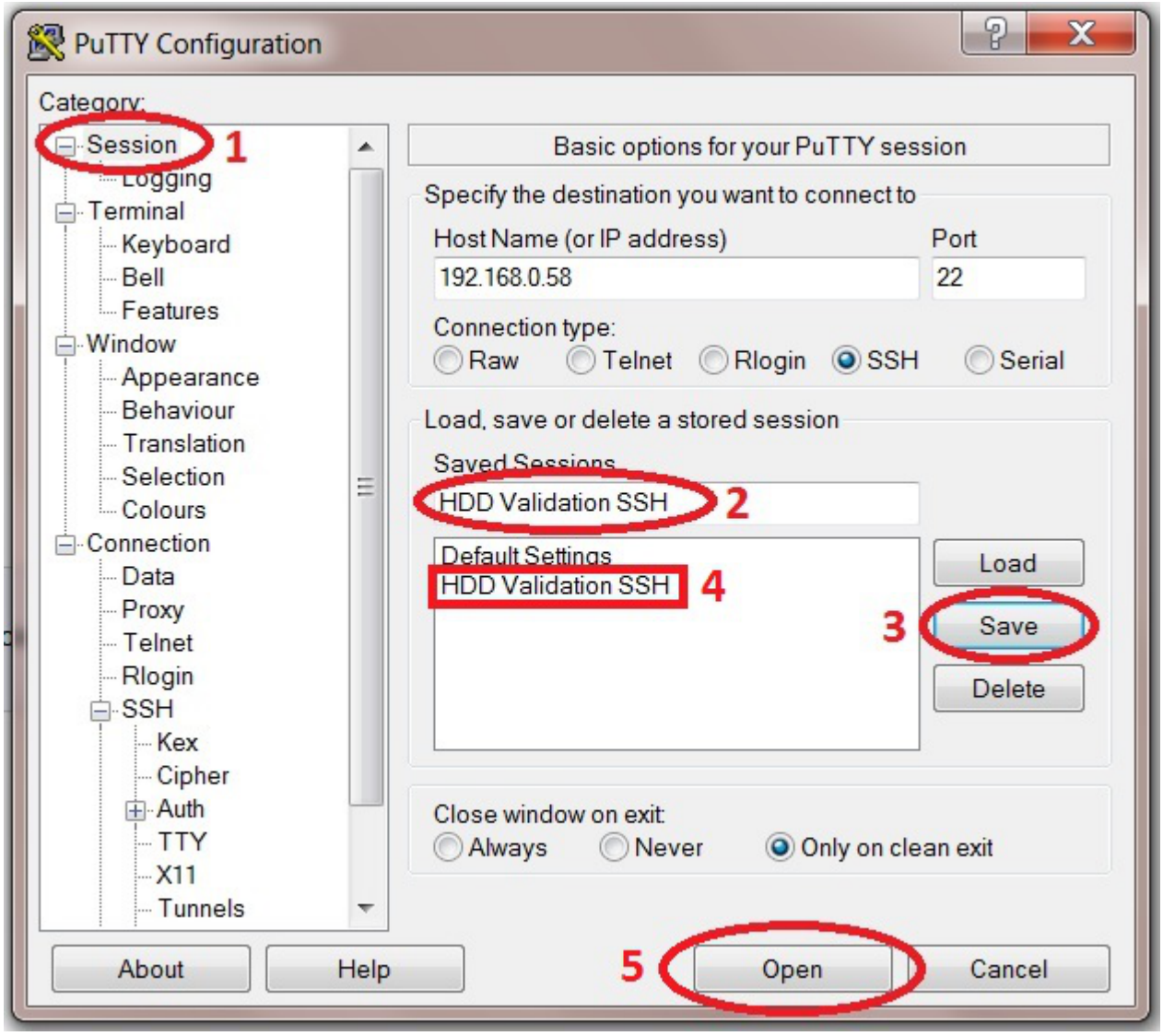
With the key now loaded in, go back to the “Session” category in the “Category” window (1).

It is possible to save the settings of this session. This is a good idea because otherwise we would need to re-enter all the details each time we wanted to start a session in PuTTY.

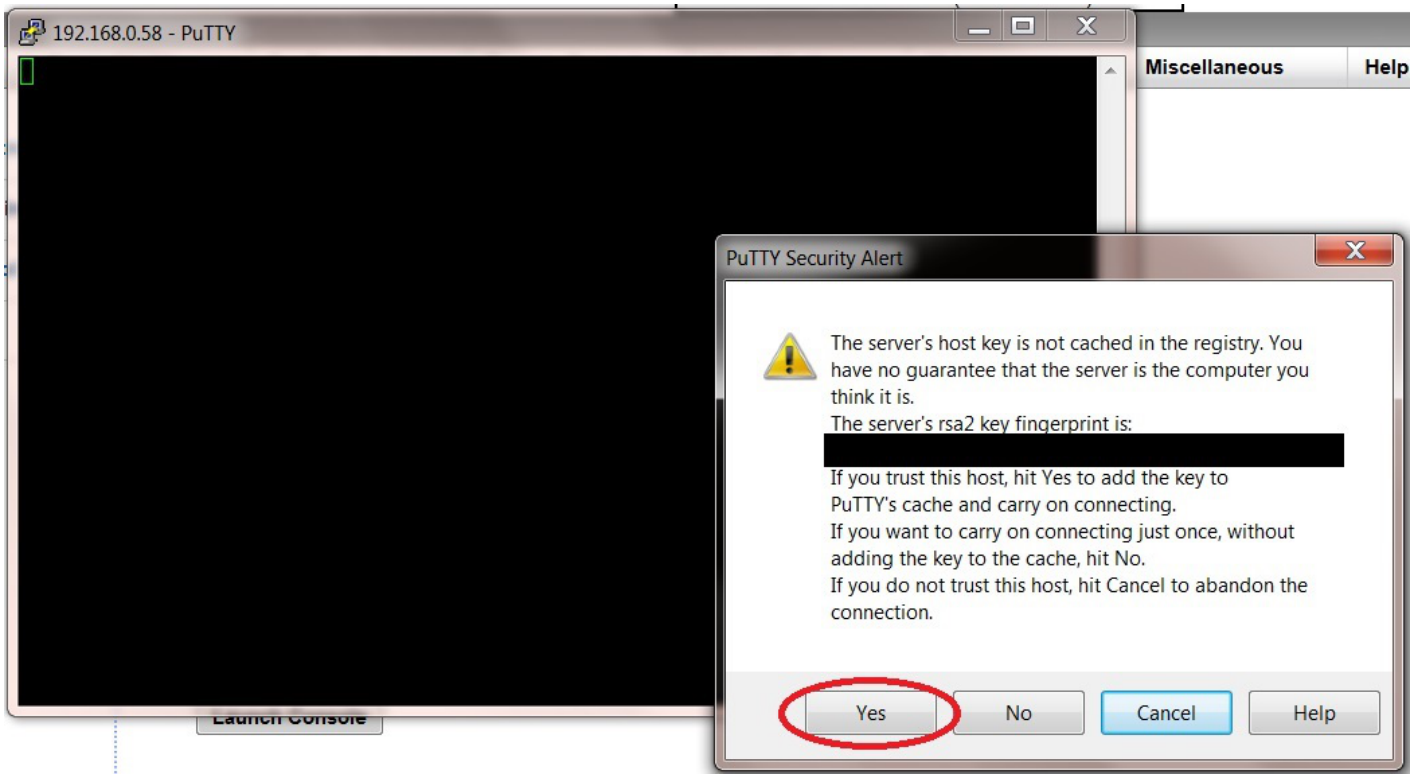
In the “Saved Sessions” box (2) type a good name for the session (Fester called it “HDD Validation SSH”).

Now click on the “Save” button (3). The saved session should now appear in the window to the left of this (4).

Now click on the “Open” button (5) to start the session (have the password you created in PuTTYgen standing by).

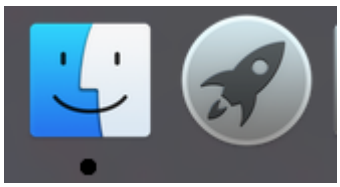


A PuTTY security alert window should now open. It will show the server's RSA2 key fingerprint and will ask if you trust this host before allowing the connection. Click the "Yes" button.

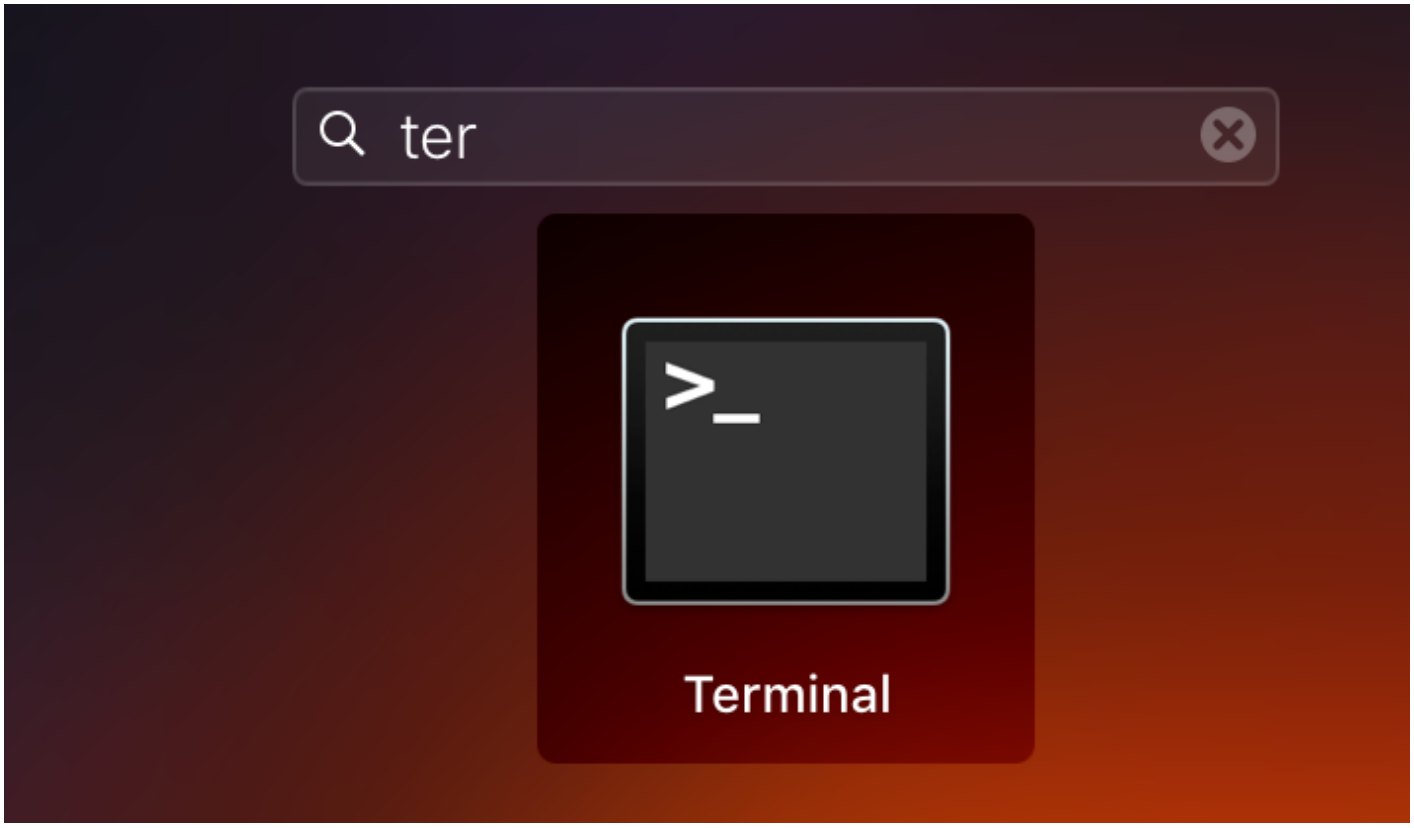


Using SSH on a Mac

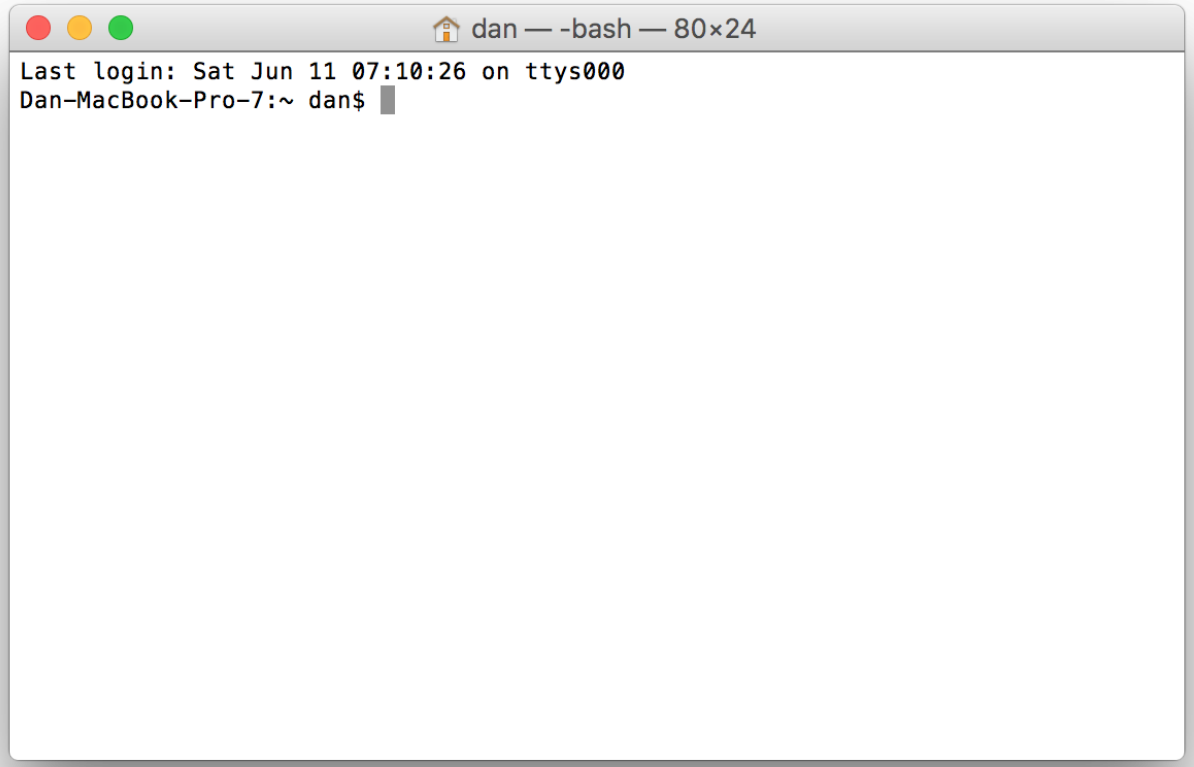
Mac OS X includes an SSH client, but it must be used from the command line. To use it, you'll need to open a terminal window. Start by clicking the launchpad button in your dock (it looks like a rocket):



Begin typing “Terminal” into the search bar at the top, until you see the Terminal icon below:



Then click on the Terminal icon. You'll see a window like this:



To connect to a server using SSH, you can simply type

```
ssh user@host
```

Where “user” and “host” are the username and hostname, respectively, that you want to connect to. For example,

```
ssh root@freenas
```

Or you can use an IP address:

```
ssh root@192.168.0.5
```

If you have required public key authentication on your FreeNAS server, you'll need to generate a keypair. To do this, type

```
ssh-keygen -t rsa
```

...and simply accept the defaults. The result will look like this:

```
Dan-MacBook-Pro-7:~ dan$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/dan/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/dan/.ssh/id_rsa.
Your public key has been saved in /Users/dan/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:ZB0ZFqEWWJuCgEWIyFd/Thk5KOuncH8LT06wOCSdRdk dan@Dan-MacBook-Pro-7.local
The key's randomart image is:
+---[RSA 2048]----+
|+=+ .. +*.B=     |
|=. o o+o E+.     |
|  . . +oB+o      |
|    ..o=+        |
|   ..+ .S.       |
|   .oo..o        |
|    oo+o o       |
|     ...*.       |
|        .+.      |
+----[SHA256]-----+
Dan-MacBook-Pro-7:~ dan$
```

The system will prompt you for a passphrase; this is optional. If you enter a passphrase, you will need to enter it every time you use this keypair (i.e., every time you use ssh). If you leave the passphrase blank, you won't need to enter it when you connect to a remote server, but neither will a thief who manages to steal your computer. You'll now need to view your public key, to enter it in the FreeNAS configuration. To do that, type

```
cat .ssh/id_rsa.pub
```

The result will look like this:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCxoFuJ2Px8sIA0zla1FXjnG+af2kRNhj/FcQ5nhOn6F2LepgX
f/4SQFjx5BWaD88H6/06lTaUAqprxKS4m33SKN7poH6RaeIfbJXwjJ/o0Cx0QbugGAeMKjHOBg4fsHw
vqGLT7o0lcQ0ubmGBZlSx9R9IFNmnDLAru+Z5gjuAwKCXGw2dxVqbq2IwB3jEoA3bbo8gy6Dso5wV75
0EC+dYlB/lQrxW/uscgPjpi1XCFVuWtajyz9jujakR1uHuRRphsp56GXVTovwM3P6h52ADDhr5vkfsk
GKgmETj940x5+MFbmBvC9iIMIeRGLfWIAQY+8NjosQYfieU5U48oDmDb dan@Dan-MacBook-
Pro-7.local
```

Copy this, all on line line, and paste it into your FreeNAS configuration.

Using SSH on Linux

SSH on Linux works just like SSH on a Mac. Follow the instructions above.