

# SSH | Secure Shell

- [Secure a new computer | SSH](#)
- [log in to remote machine | SSH](#)

# Secure a new computer | SSH

Secure new computer by creating a new user for the server/computer, then removing rights from root

This assumes that this is a new machine and that you are currently signed in as the root user.

Add your new user. Lets pretend the user name we want to create is "serverman".

```
sudo adduser serverman
```

 enter any user name you want.

Check password file. Make sure the user is valid

```
tail /etc/passwd
```

Make user a Super User

```
usermod -aG sudo serverman
```

Check to make sure that your new user is apart of ground "sudo" (super user).

```
groups serverman
```

The output should show that it is now apart of the sudo group.

Now, log out and then log back in under new user name, serverman (or whatever user name you created).

See if sudo works under new user

```
sudo ls
```

If that worked Good! NEXT...

lock root access: disable root

Go into the ssh config file and change some settings.

```
sudo nano /etc/ssh/sshd_config
```

 I like vim as my editor. If you don't know what nano is, look it up.  
"nano" text editor

Find and Set PermitRootLogin to "no"

add to the file: "AllowUsers serverman"

Save and exit.

Open a second SSH window. Test to make sure the new user name can sign in. If it doesn't YOUR SCREWED!

# log in to remote machine | SSH

Remote meaning, not the same computer you keyboard is directly connected to.

We are going to say that the user name for this machine is "serverman"

From terminal

ssh username @ address of your machine

If it is a part of your local network it might looks something like.

```
ssh serverman@192.168.0.0
```

If it is public and out of your network, it might look something like

```
ssh serverman@172.217.164.100
```

IF the computer is public, AND you have a domain name. You can use the domain name all the same.

```
ssh serverman@www.google.com
```

(I cant really SSH into there. This was just an example...)