Linux Tutorial Part One

https://youtu.be/CmNNSe60Nvk?si=eKm45PvY-MRM9_hF

In this article we will:

- **Learn how to update and upgrade packages.**
- **Learn how to create and add a user to the sudo/wheel group.**
- **Learn how to set up ssh, including making/placing the key.**
- **Learn how to disable both password authentication and root login to ssh.**
- **Learn to set up and configure UFW (Uncomplicated Firewall).**

Let's get you up and running with linux!

Picking your flavor:

When it comes to Linux based operating systems, we have what we like to refer to as *flavors* of Linux. Some of the bigger names you'll see may include; Debian (great minimal stable release option, community supported), Ubuntu (supported by Canonical, likely to feel less foreign switching from windows), Red Hat Enterprise Linux (RHEL), this is a stable release os primarily used in

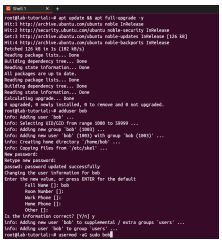
enterprise environments as you'd expect by the name, openSuse Leap (stable) or Tumbleweed (rolling), leap is binary compatible with upstream RHEL, while tumbleweed is more geared towards the latest rolling release updates.

Whatever you choose is up to you, I personally daily drive Ubuntu, but I use several others via virtual machines all the time. Once you have picked your flavor, you're going to want to install via live USB or spin up a virtual machine. Take note of your chosen distribution's package manager, `apt, yum, pacman, dnf, zypper,` **etc.**

** If you want to follow along, I recommend Ubuntu to start with, you can download it here.

** Learn how to install here.

Getting started: Updating, User Creation, Group Modification



So, let's get to it. Open a terminal. At first you will have access to a privileged user on fresh installs, if you chose a username prior to install it will have administrator privileges, let's update and upgrade quick. Use your package manager for your distribution. Run `"apt update && apt full-upgrade -y", "pacman -Syu", "dnf update"` **, etc.**

Now that we are up to date, it's time for users. If you haven't made a user account for yourself (i.e you're a default user like ubuntu, localhost, or even root), we will do that next. If you have already have an account name from installation, you can skip adding the user and group modification.

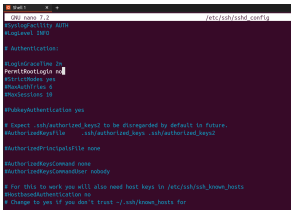
Let's add our own user. Obviously, this will require admin privileges. We will run " `sudo adduser $USER` **" with our preferred username in place of $USER.** `Useradd` **is also valid, but that command is old and lazy, adduser will prompt us for some info, a password, name, number etc, fill out whatever you want.**

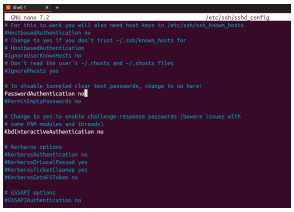
After we created the user, the next thing we want to do if we plan to have admin privileges on the new account, we need to add the user to the sudo group, or in some cases, the wheel group, we can do this by running "`usermod -aG sudo/wheel $USER.`"

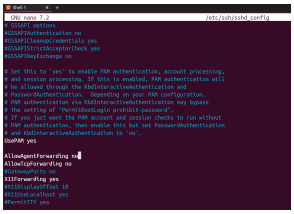
Setting up: Key Creation, Key Copying, SSH Configuration

Now that we have the user added to the sudo or wheel group, we can set up a couple more things. If we want to be able to access the machine via ssh (recommended for secure remote access) we should set it up for our new account and disallow root login (that's why we added our user to the sudo/wheel group).

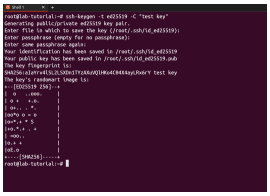
We also do not want to allow password authentication, these are commonly hit with brute force attacks. We will set up a key for this case and configure the settings in /etc/ssh/sshd_config. Simply, `cd /etc/ssh` then edit `sshd_config`, or "`nano /etc/ssh/sshd_config`". Press Ctrl+x, y, then Enter to save.







We're almost done! Now we want to copy any public key that you want to have access to the server into `/home/$USER/.ssh/authorized_keys` on the new machine. Say the host (your main machine, laptop, desktop, etc) has an IP of `10.10.10.10`, and for simplicity the VM sits at `10.10.10.11`, but the address can be different assuming it is routed properly.
If you dont have any keys, we can make them. First, on the host (main machine@ `10.10.10.10`) run " `ssh-keygen -t ed25519` " add a `-C "Name or Message here."` if you want to name or label the key.



You'll then accept the default location to save the key, and I highly recommend creating a passphrase for the key for added security, better safe than sorry! Password managers are FANTASTIC!

We have a key! Let's run `ssh-copy-id -i ~/.ssh/id_ed25519.pub $USER@10.10.10.11` this will copy the key id entered following `-i`, you can do a dry run by adding `-n` before the `-i` flag.

```
Shell 1
root@lab-tutorial:~# ssh-copy-id -n -i /home/kishin/.ssh/id_ed25519.pub kishin@127.0.0.1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/kishin/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
=-=-=-=-=-=-=-=-=
Would have added the following key(s):

ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGFSA6ByirGS0uCyXvGEFLvtmetflGwIatEidGZtgI/Q test key
=-=-=-=-=-=-=-=-=
root@lab-tutorial:~#
```

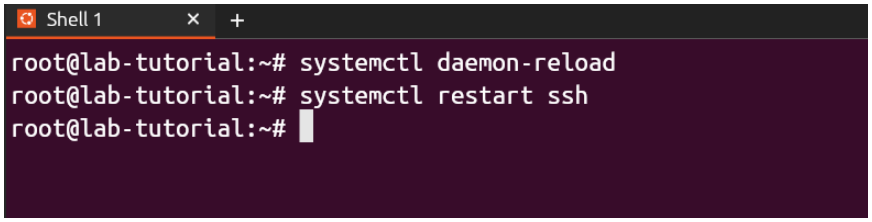
KEEP YOUR SHELL OPEN AND OPEN A NEW TAB OR WINDOW!!

IN ANOTHER SHELL, RUN `ssh $USER@10.10.10.11` AND CONFIRM YOU CAN ACCESS WITHOUT A PASSWORD PROMPT. THIS IS IMPORTANT BECAUSE YOU CAN GET LOCKED OUT.

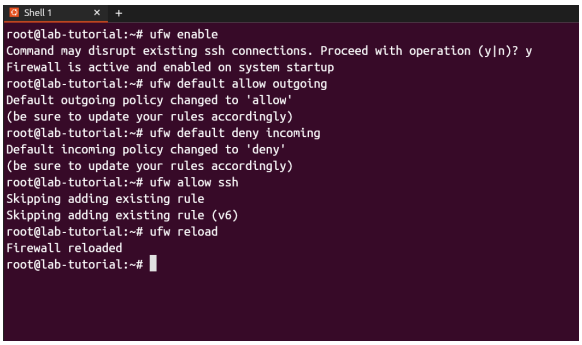
AFTER confirming you have access to the new machine via ssh without password input, we can finish up.

Finishing up

We need to restart ssh to apply all of the changes, do this by running `systemctl daemon-reload` " then run " `systemctl restart ssh` ". We don't want to `sudo` everything, worst case it just prompts for a password, only use `sudo` if necessary.



Last thing, we need to make sure we got a firewall up. `Ufw` is a decent option and should either be installed, or available via your package manager. Configuring UFW is as simple as running " `sudo ufw enable` " and configure with " `sudo ufw default deny incoming` " then either " `sudo ufw default allow outgoing` " or "sudo* ufw allow out to port 80, 443, etc". Now, simply run " `sudo ufw allow ssh` " (if you have a cloud instance touching the internet, restrict IP access, be careful if you have a dynamic public IP).



Moment of truth, run `ufw reload`. If you still have your session open, CONGRATS! You're all set.

Hope this guide can help someone with a quick start or reference.

- ***Alex a.k.a KishinInfosec 2025***