How To: Properly Install Proprietary Drivers in Ubuntu

There is some confusion about installing the proprietary drivers in Ubuntu. This article hopes to clear that up by telling you how to properly install drivers in Ubuntu.

First, this only works for the drivers that Ubuntu has access to. In this case, it's usually things like graphics cards, sound cards, some networking gear, and things like that.

Ubuntu does not have all the possible drivers. If you have to go get them from GitHub and compile them yourself, this obviously isn't the article for you.

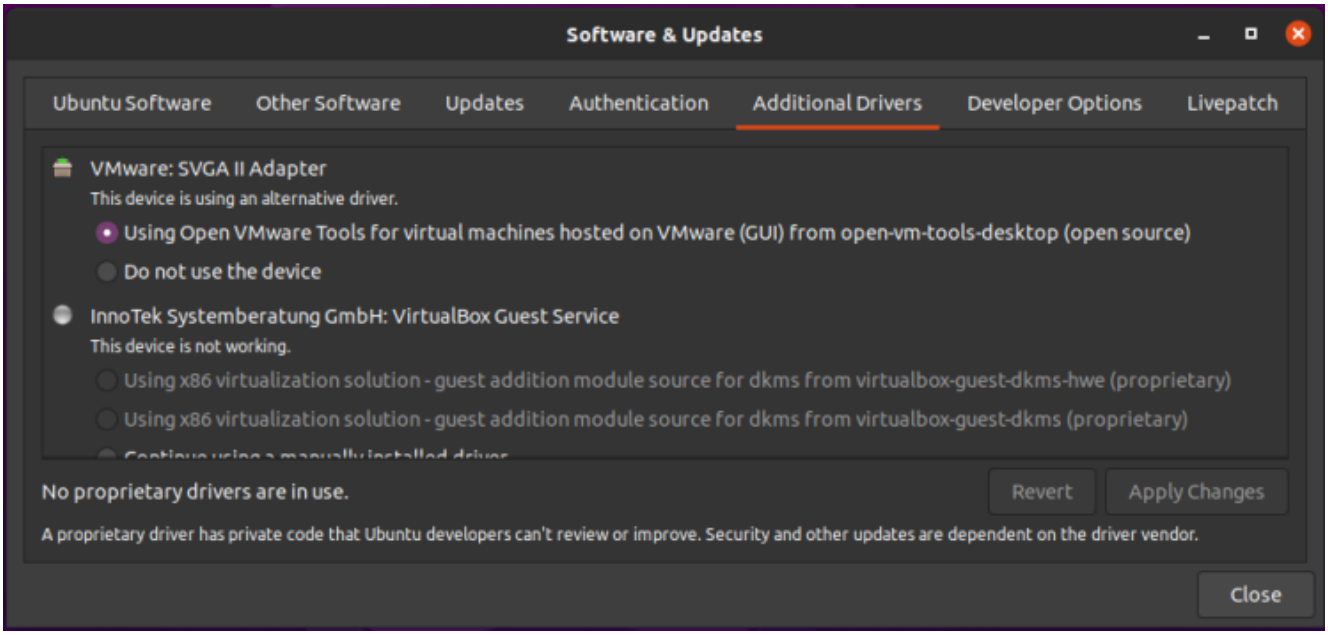
One of the most confusing is the Nvidia video card drivers. If you use Ubuntu, an official flavor of Ubuntu, or a derivative of Ubuntu, **DO NOT** download the .run file from Nvidia's site. While it may work, it will quite possibly not work with dkms and you will have to spend significant time fixing it every time the kernel is updated. It quite likely lead to breakage.

Yes, this means having some patience. But, have some patience because the drivers *will* make it down to the repos and will have then been tested. The drivers you get from the official repos will not only update, they'll update with the rest of the system **AND** they'll work properly with the kernel updates. When the kernel updates, the system itself will insert the appropriate drivers calls.

Doing this any other way will quite likely lead to hardship – and it's a hardship that you don't need to have. It's a hardship that's easily avoided. If you read the forums and question/answer sites, doing this the wrong way results in at least one question *almost every single day*.

The first way is easy. It's even in the GUI. Search your menu

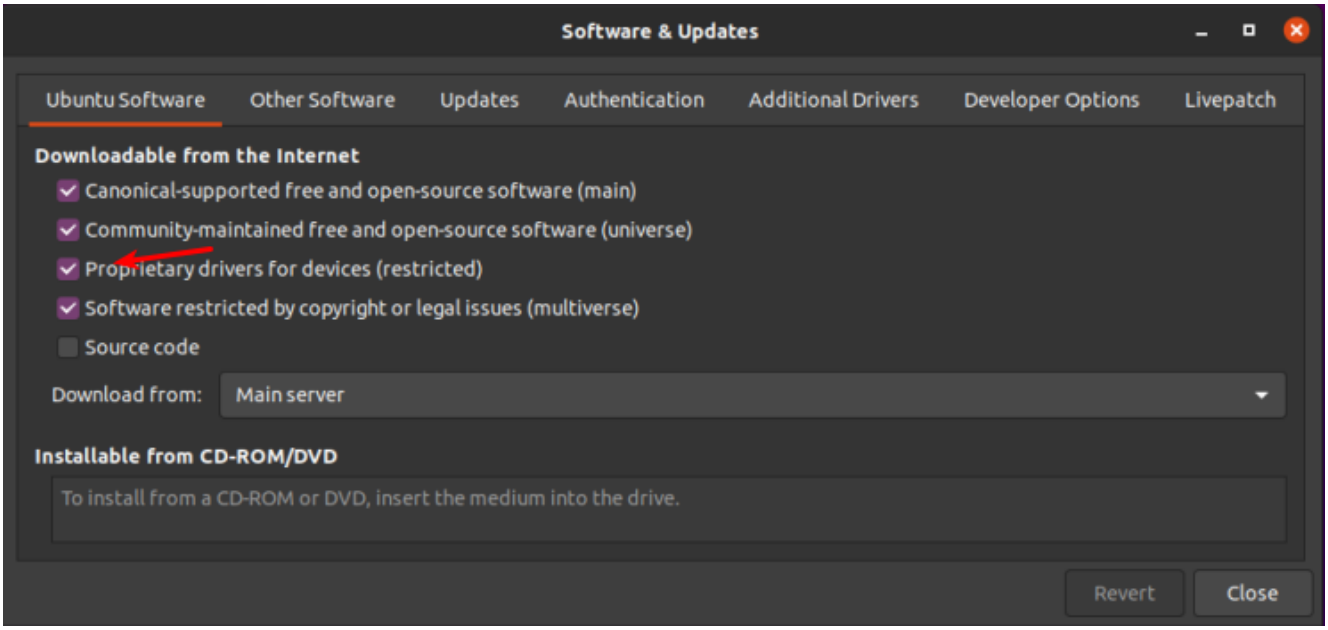
for ‘Additional Drivers’ or similar (it may be only listed as “Software & Updates” depending on your Ubuntu flavor). It looks like this:



This should be self-explanatory, so I will just leave this here.

See? Pretty easy. Just pick what you want, apply the changes, and reboot.

NOTE: You will need to have the ‘restricted’ repository enabled in order to do this. That should be assumed, but some of you may not know this. So, a quick screenshot should make this even easier to figure out – it’s in the same app as the above screenshot, but it’s in the first tab. It looks a little like this:



The arrow should make it clear. That repo needs to be enabled for this.

The second way is to use the terminal. Let's go ahead and get the terminal opened by pressing CTRL + ALT + T on your keyboard.

Now, let's check and see what drivers we can automatically install from the terminal:

[code]ubuntu-drivers devices[/code]

The output should let you know what drivers are available for your devices. Again, this is pretty self-explanatory. You really don't even need to enter that command, you can do it all automatically and get the recommended drivers automatically installed. It's easy. Just run this:

[code]sudo ubuntu-drivers autoinstall[/code]

That will go through and install the recommended drivers for your devices automatically. That will install the proprietary drivers, the ones with binary blobs and decidedly not opensource drivers. If those are the drivers you want, that's the easiest way to install them.

Simply run the command, reboot, and you're done. Not only are

you done, you shouldn’t have to mess with them again – ever again. They will update automatically, they will automatically be applied when you update the kernel, and they will *generally* just work.

For the time being, we’re ignoring the idea of using the opensource drivers. I’ll simply say that I do quite well without needing proprietary drivers. I hardly ever bother installing them – unless I absolutely need a feature that’s not offered with the opensource drivers. I find that it’s still a working operating system and I can still easily meet my goals. You do you and you make the decision, but at least do it the right way after making that decision.

As always, thanks for reading. There’s a newsletter that will email you when a new article is published. That’s all it does. If you’re wanting to keep up with the site, that’s exactly how you do it! Well, there are push notifications available for those that prefer that. So, you do have choices! Either way, I won’t send you any spam. I promise!

How To: GUI Login as Root in Ubuntu

In this article, I’m going to show you how to enable the root account with Ubuntu. This is a terrible idea and you should definitely not do this. Ever.

A while back, I told you how to enable root in Ubuntu. In that article, I also wrote about the people who don’t answer questions when they don’t think you’re doing things the right way. Being the kind of person I am, I’ll happily tell you how to make your OS less secure.

And, trust me, this is a horrible idea from a security perspective – especially if you don't have good physical security. Then again, if you don't have good physical security then your computer is already compromised if someone wants to just boot to a live USB thumb-drive and if you haven't taken the steps to encrypt your private data.

NOTE: This is only good for Ubuntu. It looks like it should work from 18.04 to 20.10, and will probably continue to work until Ubuntu moves on from GDM3. (GDM3 is Gnome Display Manager 3, a drop-in replacement for GDM.) This may work for other Ubuntu flavors, I haven't tested. If you do test or know, please leave a comment below. Thanks!

Anyhow, on with the work. This shouldn't take too long.

The first step is to set up the system so that you can login as root. To do that, you have to enable root login for Ubuntu. You should probably read the warnings on that page and you should think carefully before doing this to your own computer.

The next step is to crack open your default terminal emulator. You can do that by pressing CTRL + ALT + T.

Now let's make you a superuser. You can do that with:

[code]sudo su[/code]

(Press enter and enter your password, of course.)

Our next step is to tell GDM3 to let us use the root login.

[code]nano /etc/gdm3/custom.conf[/code]

You're going to arrow down to just below the automatic login configurations and enter this line:

[code]AllowRoot=true[/code]

Then, you'll press CTRL + X, then Y, and then ENTER. (Congratulations, you've used 'nano' again and edited a file

in the terminal!)

Our next step is to tell PAM (Pluggable Authentication Modules) that logging in as root is okay. That’s pretty easy, and we’ll do it with nano once again.

[code]nano /etc/pam.d/gdm-password[/code]

Now, scroll down and look for this line:

[code]auth required pam_succeed_if.so user != root quiet_success[/code]

What you’re going to do is ‘comment it out’. Basically, you’re adding the **#** symbol which means, in this case, ‘skip this line’. It’s a way to tell the system to ignore a line while leaving the line there in case you change your mind.

So, change that line so that it looks like this:

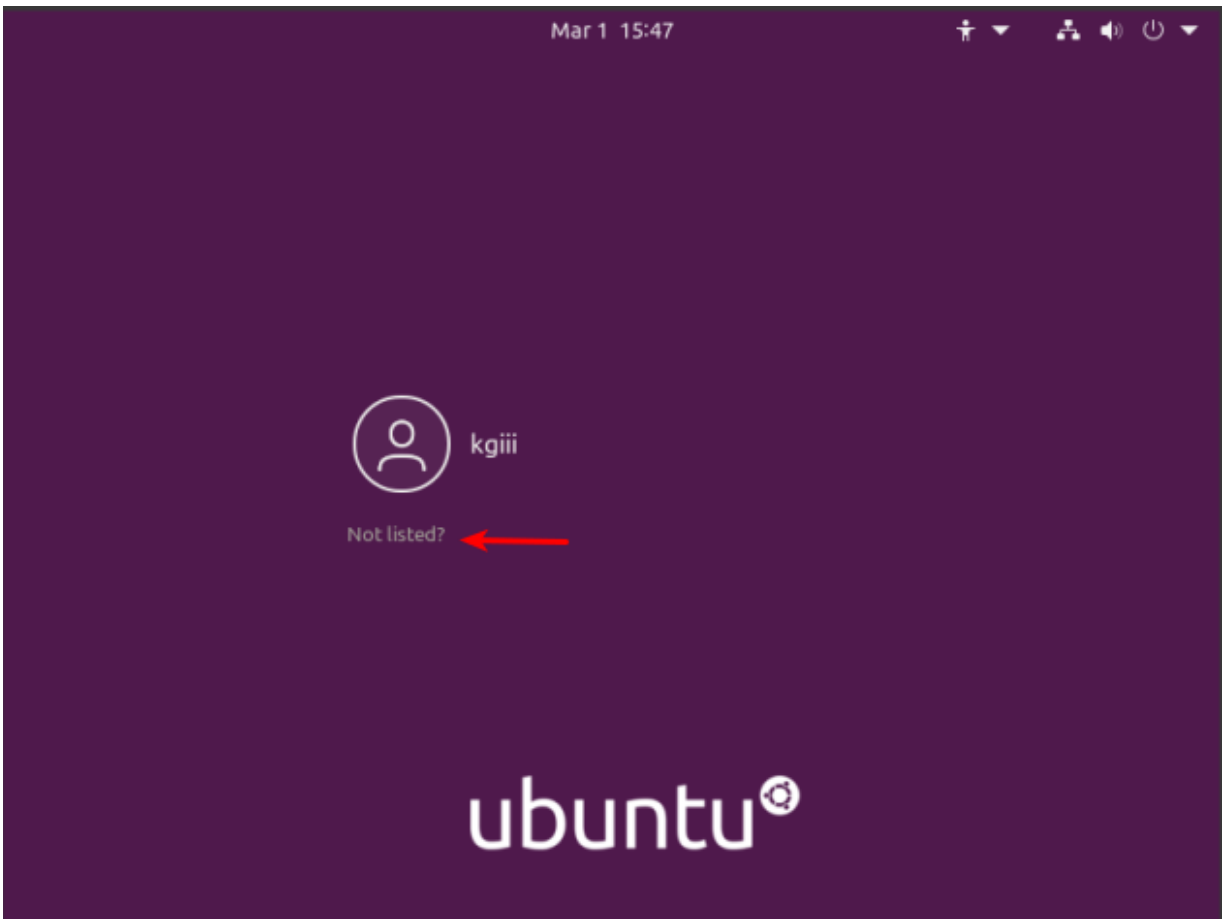
[code]#auth required pam_succeed_if.so user != root quiet_success[/code]

Now, save it just like you did above. (Press CTRL + X, then Y, and then ENTER.)

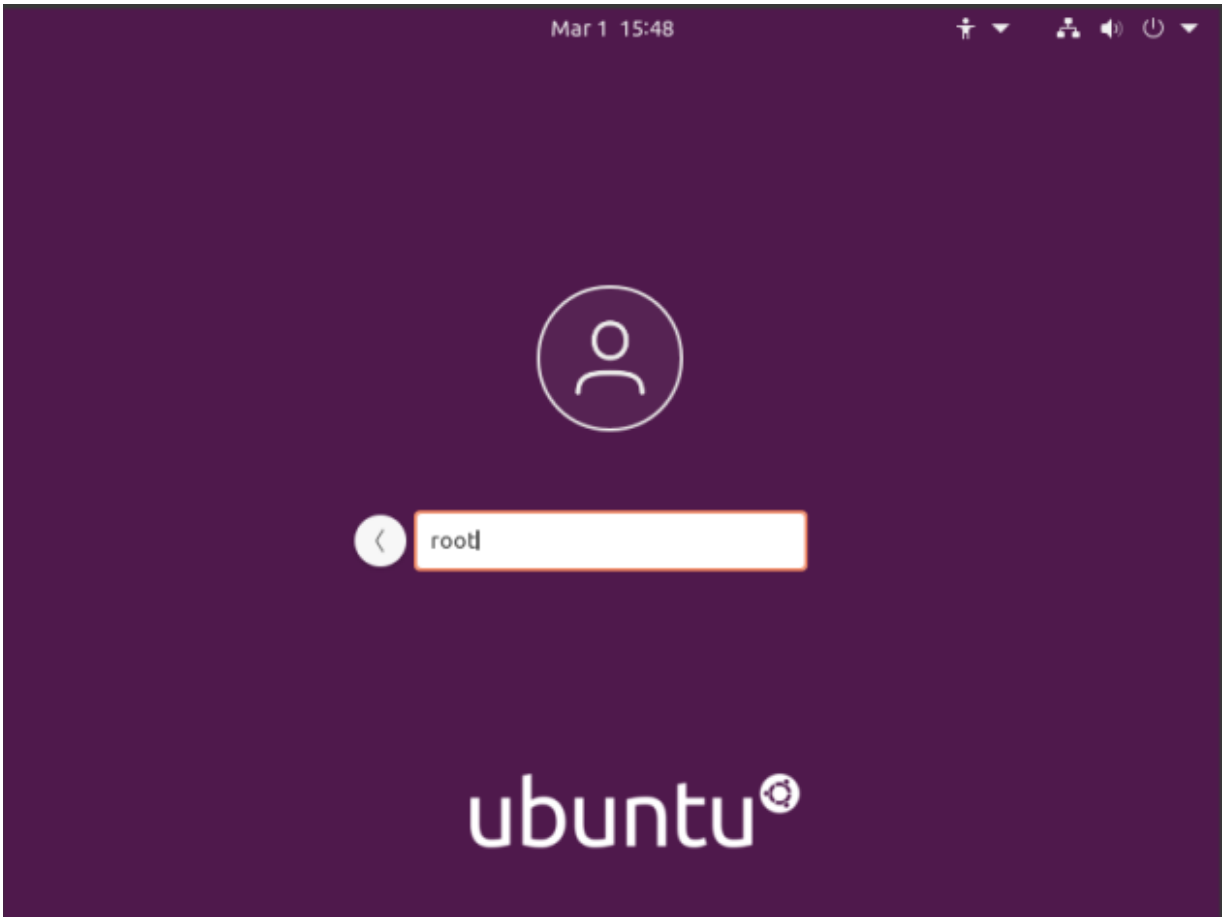
You’re still using ‘sudo su’ and you can get out of that mode with:

[code]exit[/code]

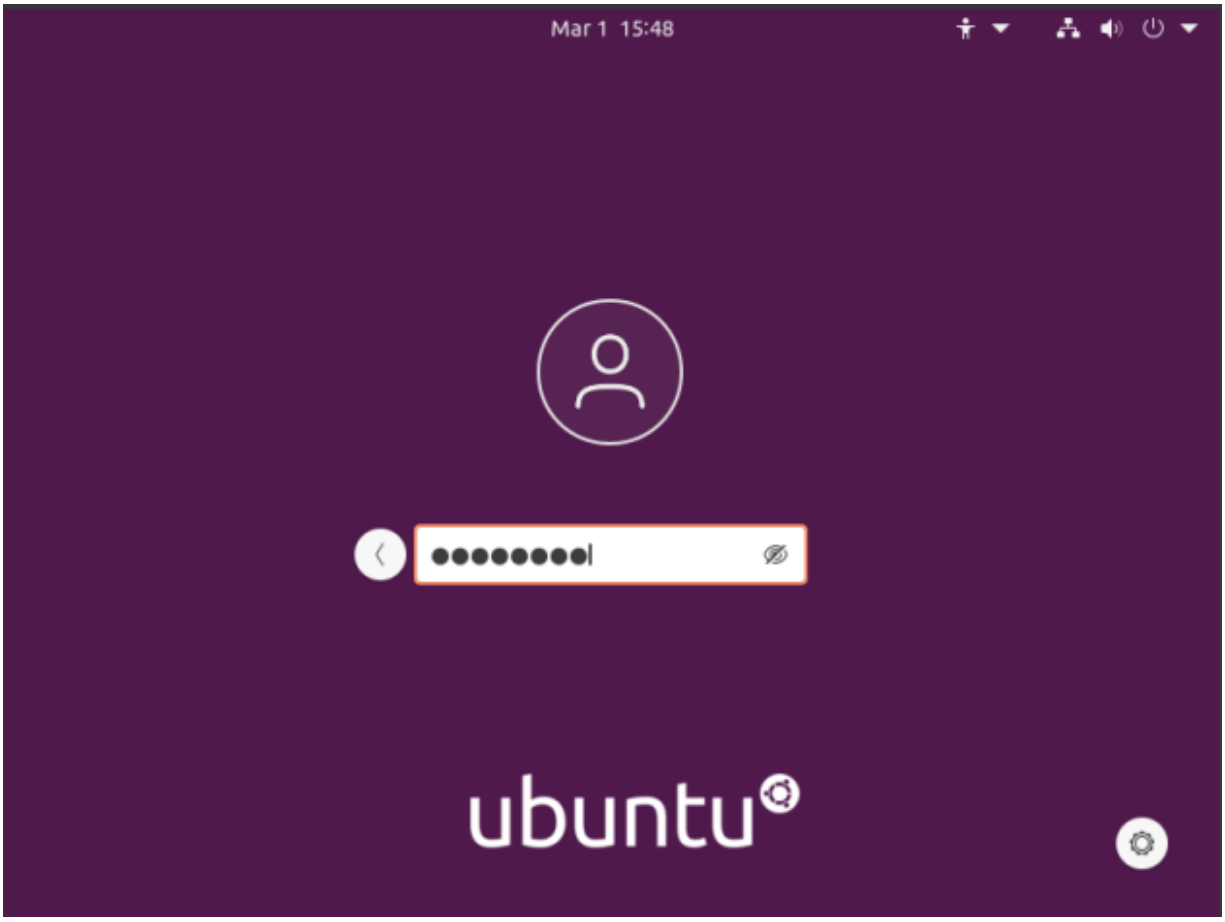
Now, when you next reboot, you can login as root. What you need to do is click on ‘Not Listed’, type in ‘root’, and then enter your password.



Like so…



And like this…



Enter your password and the press the ENTER button.

Tada! You're now logged in as root for no good reason and with almost no benefits! Congratulations! Now, undo it and go back to being a bit more secure. Or not… I don't mind. Just don't let your box get owned and turned into something malicious like a spam bot or a node in bot network used to do things like DDOS sites for money. Seriously, this is a horrible idea and you shouldn't do this.

Anyhow, thanks for reading. I appreciate it and I'm glad to get some of my notes online – finally. Things seem to be going at a good pace right now and I suspect I can keep this up for a while. If you want to be notified of new articles, you can either sign up for the newsletter (which is spam free) or you can use push notifications and your browser will happily tell you when there's something new published. If you sign up for the newsletter, I promise to not send any spam. I'll only ever use it for article notifications or very important site notices.

Install ‘Docky’ on Ubuntu 20.04+

Docky is a Linux application that provides a dock for applications you’d frequently use. It’s no longer in the default repositories. This is how you install Docky on newer versions of Ubuntu.

First, open your terminal.

[code]cd Downloads && mkdir tmp && cd tmp[/code]

Next, you’ll need Docky’s dependencies:

[code]wget http://archive.ubuntu.com/ubuntu/pool/universe/g/gnome-sharp2/libgconf2.0-cil_2.24.2-4_all.deb
wget http://archive.ubuntu.com/ubuntu/pool/main/g/glibc/multiarch-support_2.27-3ubuntu1_amd64.deb
wget http://archive.ubuntu.com/ubuntu/pool/universe/libg/libgnome-keyring/libgnome-keyring-common_3.12.0-1build1_all.deb
wget http://archive.ubuntu.com/ubuntu/pool/universe/libg/libgnome-keyring/libgnome-keyring0_3.12.0-1build1_amd64.deb
wget http://archive.ubuntu.com/ubuntu/pool/universe/g/gnome-keyring-sharp/libgnome-keyring1.0-cil_1.0.0-5_amd64.deb[/code]

Now, let’s install them all at once:

[code]sudo apt-get install ./*.deb[/code]

Alright, you’ve now taken care of the dependencies. Let’s grab

the Docky .deb:

[code]wget http://archive.ubuntu.com/ubuntu/pool/universe/d/docky/docky_2.2.1.1-1_all.deb[/code]

And, of course, you can now install it:

[code]sudo apt-get install ./docky_2.2.1.1-1_all.deb[/code]

There. You now have a running/working Docky and you can customize it and use it just like you did on older versions of Ubuntu. This should, of course, work with all official flavors of Ubuntu and with those distros that base themselves on Ubuntu's 20.04+ version.

Check Your Ubuntu Support Status

Want to know how long your version of Ubuntu is supported? You can find the support status by cracking open your terminal and entering:

ubuntu-support-status

If you're using 20.04 or newer, then the command is slightly different:

ubuntu-security-status

Use Wayland in a Live Instance of Ubuntu

This is from an AU question that I answered. The person wanted to know how to use Wayland while in a live instance of Ubuntu. It is something you can do.

The first thing you need to do, assuming you're already booted into the live instance of Ubuntu, is change the way you login. You can click in the bottom right and 'Show Applications.' Once there, you can enter the word 'users', click on the settings app offered, and disable logging in automatically.

When you're done with that, change the password. It turns out this is mandatory – and perhaps a bug. You'll need to deal with Ubuntu's need for a complex password, so just make sure it's at least 12 characters long, not a dictionary word, and has a mix of numbers and letters. Yes, even in the live instance that's temporary they want a complex password meeting whatever criteria they set.

Next, you want to edit */etc/gdm3/custom.conf* and comment out the line that disables Wayland.

[code]sudo nano /etc/gdm3/custom.conf[/code]

Find the line:

[code]WaylandEnable=false[/code]

Change it to:

[code]#WaylandEnable=false[/code]

Make sure to save it, obviously.

Restart *gdm3* with:

[code] systemctl restart [/code]

Note: That may log you out, that’s fine. If it doesn’t automatically, log out manually.

Log back in, but after you click the user, there’s an icon in the lower right. It’s a gear icon. Click it and choose *Ubuntu on Wayland*. Then enter your password and press **ENTER**.

If everything worked, you’re logged in with Wayland.

Now, to verify this…

Press **CTRL** + **ALT** + **T** and open the terminal and enter:

[code]echo $XDG_SESSION_TYPE[/code]

Which should tell you that you’re using Wayland. If done correctly, it looks like this:

Good luck!