How To: Enable SSH

In today’s article, we’ll learn about enabling SSH. SSH is a useful tool for remotely managing your Linux computer. This is a pretty simple, painless, and quick exercise.

Let’s say your computer is in another room, another state, another country, or on another continent entirely. How are you going to manage it? After all, we have servers across the world and it is not even remotely economical to send a person to administer each one of them in person.

On a home-use note, it’s perfectly suitable to manage my own router using SSH. It’s quick, easy, lightweight, effective, doesn’t require an attached monitor, and more. What’s not to like?

Installing SSH on Linux

My homemade router doesn’t have a keyboard attached. It doesn’t have a monitor attached, so it’s not like I can just easily walk over and deal with it.

I just got a new computer, a lovely laptop that I got for a fantastic price. I got it to test Lubuntu. I don’t always want to have to go over to the device and physically use it.

I have a dedicated server in Las Vegas. I live in Maine. It wouldn’t be practical to fly out to Vegas every few days to run updates on the server. It wouldn’t make financial sense to go out there every time the server needed to be rebooted.

These are all great candidates for SSH. SSH stands for “Secure Shell” and it’s a protocol. It’s not an application, and you can use SSH for all those things. A great many applications can communicate over the SSH protocol, including every terminal emulator that I’m familiar with.

The man page defines SSH as:

> *ssh – OpenSSH remote login client*

It has been around since the mid-90s and does nifty things beyond allowing you to control the remote system with commands, it also allows you to transfer files with things like SCP and SFTP. It's right full of nifty features and you might as well become familiar with it. I'll possibly cover both of those in a future article.

It's very, very easy to get going. Simple use your keyboard to open the terminal, by pressing CTRL + ALT + T.

Now, simply type (this one can be easily adjusted to other package managers):

[code]sudo apt install openssh-server[/code]

There… You're done. Well, you are more or less done. You now effectively have SSH running. It starts itself after you install it. That's pretty handy!

Now, make sure you're on the same subnet, and you can connect to the remote computer – the computer where you just installed and enabled SSH. You can do that in a couple of ways. You can do it like this:

[code]ssh remote_user@ip.a.d.d.r[/code]

You can switch the obvious for the obvious, but you will need to know the IP address for the remote computer. That seems a bit tedious, so let's just skip that part. Rather than remembering the IP address (which may change), just remember the name of the device.

So, instead, you'd run:

[code]ssh remote_user@host_name.local[/code]

If that doesn’t make sense, this is how I’d connect to the new laptop:

[code]ssh kgiii@kgiii-msi.local[/code]

Obviously, the hostname is ‘kgiii-msi’ and your hostname will be different. It’ll be the name you gave the computer during the installation process, typically during the same phase where you generated your user account. If you don’t actually know your hostname, you can easily find it. It’s simply:

[code]hostname[/code]

There’s more to this and we’ll likely cover that soon enough, but that’ll get you started. If you have a firewall installed and enabled, you may need to let SSH through if you want to use it. This is such a simple thing that I’d be remiss in my duties to not make folks aware of how simple it is.

And, with that, I thank you my dear reader for taking the time out of your day to humor this old fool. Your feedback is appreciated and keep signing up to that whole newsletter thing. Being old, I tend to forget to submit and share these articles elsewhere. Signing up means you have no excuses for missing an article!

[CentOS-announce] CESA-2021:1072 Important CentOS 7 libldb Security

Update

CentOS Errata and Security Advisory 2021:1072 Important
Upstream details at : https://access.redhat.com/errata/RHSA-2021:1072

The following updated files have been uploaded and are currently

syncing to the mirrors: (sha256sum Filename)

x86_64:

67364ca692de365478eee5a94879717c1fae2b7a4ba46d128ec04f0477c8c2b5 ldb-tools-1.5.4-2.el7.x86_64.rpm

36ad5a43df60889dd9c1134cb0e042317befa64f7293f44bc91271fddbbfc7e6 libldb-1.5.4-2.el7.i686.rpm

cec370a7441899c3ffcd47f783a0437d9d649fd4a1252c6c317561f431e537c4 libldb-1.5.4-2.el7.x86_64.rpm

359852ce38e0555b23e78c945070ef67c0599138eac0c52de77a819e8fdebce9 libldb-devel-1.5.4-2.el7.i686.rpm

4d0e360eff9294623b345353bcf2cb4623c50a3e2bf31ace6ba05141150d85fd libldb-devel-1.5.4-2.el7.x86_64.rpm

29124a79cce7024da4f024131a66f80d78a70d73c5fafe6456617633e5b83560 pyldb-1.5.4-2.el7.i686.rpm

8043266fac97f3c92dfeaa8fad590469ad37ab990d86e9ece87829bdd9e0c8ae pyldb-1.5.4-2.el7.x86_64.rpm

b3dbb953a4dc9b8b5ee95024d51d922488db5a0f0ec2ece9bf47d8d5cbbf24fa pyldb-devel-1.5.4-2.el7.i686.rpm

8f596e23215f48c31a9bb115c327431b21431ac93d7279b39dc998ca0bdfc6b8 pyldb-devel-1.5.4-2.el7.x86_64.rpm

Source:

e678f1a0df3c67bd8f6319dbe32013a311d6d797b51284ff7d5e254c2f7a1ff5 libldb-1.5.4-2.el7.src.rpm

_

Johnny Hughes

CentOS Project { http://www.centos.org/ }

irc: hughesjr, #[hidden email]

Twitter: @JohnnyCentOS

_____

CentOS-announce mailing list

[hidden email]

https://lists.centos.org/mailman/listinfo/centos-announce

[CentOS-announce] CESA-2021:1071 Important CentOS 7 kernel Security

Update

CentOS Errata and Security Advisory 2021:1071 Important
Upstream details at : https://access.redhat.com/errata/RHSA-2021:1071

The following updated files have been uploaded and are currently

syncing to the mirrors: (sha256sum Filename)

x86_64:

bfe191b783a11c70daf05fb86e81e2e36d80b7dec5eb2243fa223700ce330824 bpftool-3.10.0-1160.24.1.el7.x86_64.rpm

996ee55268c9971d07d38c3217e0fb813a202d1b838963b6db16217069d193db kernel-3.10.0-1160.24.1.el7.x86_64.rpm

33b524d6eec3fc82a17df2220b596193025c1faf20c5939c4271809681f95803 kernel-abi-whitelists-3.10.0-1160.24.1.el7.noarch.rpm

8764032443efee4dc7bfb0ee1a11749205880cd86b230ad538346b697120c5e7 kernel-debug-3.10.0-1160.24.1.el7.x86_64.rpm

e2b80fc90e80e10166a785ab1c718ed12380055d955ab295c5363ad6405fe815 kernel-debug-devel-3.10.0-1160.24.1.el7.x86_64.rpm

52fc84afa30b500c79c2116a67a199c3eba6bbed1b7b171fc4fec483dc2c9f4c kernel-devel-3.10.0-1160.24.1.el7.x86_64.rpm

cda402fcb291052201381d37c733af954d30e2e6e3f24e5b636ae67715e8c0d0 kernel-doc-3.10.0-1160.24.1.el7.noarch.rpm

2a69b561a8c58b7ed126929ce0f305827b54da8604e8f662568fc8ec96090f26 kernel-headers-3.10.0-1160.24.1.el7.x86_64.rpm

150b5e83d6acc1e5a6e22bee18216d6c7e0c581dca489071a61aab70eb9b93fb kernel-tools-3.10.0-1160.24.1.el7.x86_64.rpm

540ad2675ab792c4e347811b9c59c9dfa46be5932ed582b6b0748c7a27660973 kernel-tools-libs-3.10.0-1160.24.1.el7.x86_64.rpm

44904175313b13552ac962d42d456dc5d52bface994ef485f56c33f7f6971440 kernel-tools-libs-devel-3.10.0-1160.24.1.el7.x86_64.rpm

9124221e268619f8424d72980a7b256e0e00ba0639fcf0be1387712d145c7416 perf-3.10.0-1160.24.1.el7.x86_64.rpm

2308127baa502197469a17cef0a36622ccd5c528247af648e424284943e73572 python-perf-3.10.0-1160.24.1.el7.x86_64.rpm

Source:

6fc0eaf2486a736d0793f6165e07c183bb0c8db2c858bd0dbefc1a2b23a0528b kernel-3.10.0-1160.24.1.el7.src.rpm

_

Johnny Hughes

CentOS Project { http://www.centos.org/ }

irc: hughesjr, #[hidden email]

Twitter: @JohnnyCentOS

_____

CentOS-announce mailing list

[hidden email]

https://lists.centos.org/mailman/listinfo/centos-announce

Why I Use Linux

Today’s article will be short and simple. I’ve wanted to write it for a while, and it may end up being another one of those articles that gets updates over time.

I tend to use Lubuntu and that’s my preferred desktop system – even after the change to LXQt. It’s visually simple and familiar. It’s light on resources, even though I have plenty.

I don’t use Linux because I hate Microsoft. You’ll never see me call them a derogatory name. I don’t have any major anger towards Microsoft or their products.

I don’t even care if the product I’m using is opensource. Being closed-source doesn’t bother me. I care that the application works and lets me do the things for which I installed it.

I am using a closed-source browser as I type this. Like Linux, it just works for me. It gets out of my way and lets me accomplish my computing goals.

That’s what I really like about Lubuntu. It just gets out of my way. Once it is configured, I don’t have to keep tinkering with it. I don’t have to continually pay attention to the operating system – it just keeps working and I just keep updating it.

I love the simplicity and efficacy of the terminal. When I boot my computer, a terminal emulator is one of the first things I open. I often have two or three of them open at once.

I never have any hardware issues that I can’t resolve. Sure, it may be a bit difficult to find your wireless driver – but

once you do then you needn’t worry about it again. In the mean time, tether your phone and connect that way. Then again, I always have an adapter that works well enough for me to get the drivers for any built-in hardware.

I don’t see the process as any different than Windows. You have to put some effort in to make it work. But, once it’s installed, all of my updates are done at the same time and with but a single command. The concept of a package manager is fantastic and you get a wide variety to choose from.

I like both the sense of community and the community. If I really want to put the effort in, I can find the person what wrote the driver for my wireless adapter. I don’t suppose that’s really true with any other operating system. With Linux, I can find the person(s) who put my OS together – and, in fact, I do. I talk with them at least once a week at the online team meeting. I recognize many of the names I see across the ‘net and I’ve known some of them for many years.

I suppose that I do like having access to the source code. I don’t tend to make (m)any changes, as my programming skills aren’t that great these days. Still, I do sometimes make a quick change, apply my own patch, and compile applications on my own.

I like that I have the freedom to have as much, or as little, operating system as I want. I can have a distro with everything installed or I can have a distro that barely has a terminal installed – and you’ve gotta compile that yourself. There are so many choices. There’s a Linux for everyone and, if you’re willing to learn, there are seemingly infinite combinations. I like being able to pick my desktop environment, favorite terminal emulator, favorite window manager, etc…

I like that it’s always changing. I legitimately like systemd, for example. I like learning Netplan. I like learning the new

features. I like understanding what's going on under the hood – or having access to people that can actually explain it. I also like that no matter how hard I try, I will never truly understand everything. There's always something new to learn. There's always something new to 'geek out' with.

I guess, with the above, you could say that I like the constant innovation. Sure, sometimes Linux is trying to 'keep up' with the other mainstream operating systems – and sometimes it goes out on the edge and the community does things you simply can't find elsewhere.

Linux isn't perfect. There are bugs aplenty and flaws we'd maybe not tolerate in an operating system we paid for. Sure, we overlook the warts and call it our own – but we can call it our own. We can meaningfully contribute to a project, to a distro, to an organization, and to the community. There are so many ways that we can give back, and that is awesome.

Anyhow, I don't want to make this too long. Feel free to write a sentence or two below, letting me know why you like Linux. I don't think there's a wrong answer to this question – unless you try REALLY hard to write a wrong answer. Instead of responding where you'd normally find me, respond here so that folks can see this in the future and see your contributions to the subject.

Like always, feedback is awesome and the newsletter works. I've been wanting to write this for a while and, well, it's my site. So, I get to do things like that! If you want your own site, that can be arranged. If you want to contribute here, that too can be arranged. Until next time…

USN-4896-2: lxml vulnerability

USN-4896-1 fixed a vulnerability in lxml. This update provides the corresponding update for Ubuntu 14.04 ESM.
Original advisory details:

It was discovered that lxml incorrectly handled certain HTML attributes. A
remote attacker could possibly use this issue to perform cross-site
scripting (XSS) attacks.