

ssh command in Linux with examples

23 August 2021 by Luke Reynolds

The **ssh** command in [Linux](#) is used to manage remote systems. Ordinarily, this would be another Linux system, but it could also be a firewall, router, or even a different operating system entirely. Using the **ssh** command to remotely log into another system will give you a [command line](#) terminal that you can fully access as if you were physically in front of the machine.

As you can imagine, being able to manage a countless number of remote systems without getting up from your chair is a dream for Linux administrators, or even ordinary users that have multiple systems in different locations. SSH can also be used to create [port forwarding tunnels](#), effectively encrypting and securing connections made through any type of application on your Linux system.

The **ssh** command has a few different options that we can specify which allow us to manage our connection with the remote machine. Some of these are definitely worth learning, such as how to specify a username or port number with the **ssh** command.

In this guide, you'll learn how to use the **ssh** command in Linux through examples. Follow along below to learn about the various options that you can use with this command. Before beginning, you may want to check out our guide on [getting the most out of OpenSSH](#), as it explains how to get the protocol installed on any type of Linux distribution.

In this tutorial you will learn:

- How to use the **ssh** command on Linux



ssh command in Linux with examples

Software Requirements and Linux Command Line Conventions

| Category | Requirements, Conventions or Software Version Used |
|-------------|---|
| System | Any Linux distro |
| Software | ssh |
| Other | Privileged access to your Linux system as root or via the sudo command. |
| Conventions | # – requires given linux commands to be executed with root privileges either directly as a root user or by use of sudo command \$ – requires given linux commands to be executed as a regular non-privileged user |

Frequently Used Options

The **ssh** command is essential if you want to manage remote systems. Follow along with our examples below to learn about some of the most common and useful options to use with the command.



```
root@miami:~# hostname
linuxconfig
root@miami:~# ssh root@linuxconfig.org
root@linuxconfig.org's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-140-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sat Aug 21 23:01:32 EDT 2021

System load: 0.31          Processes: 110
Usage of /:  83.7% of 23.40GB   Users logged in: 1
Memory usage: 69%           IP address for ens3: 
Swap usage:  9%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.
   https://ubuntu.com/blog/microk8s-memory-optimisation

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
   https://ubuntu.com/livepatch

0 packages can be updated.
0 of these updates are security updates.

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sat Aug 21 22:54:32 2021 from 
root@linuxconfig.org:~# hostname
miami
root@linuxconfig.org:~#
```

Logging into a remote system by using the ssh command in Linux

ssh command in Linux Basic Examples

Example 1 To login to a remote system with ssh, simply specify the host name or IP address of the remote system in your **ssh** command. As an example, we will show the command to SSH into the server named **linuxconfig.org** that has an IP address of **10.1.1.1**.

```
$ ssh linuxconfig.org
OR
$ ssh 10.1.1.1
```

Example 2 Unless the username that you're currently logged into is the same as the username on the remote system, you will want to specify the username and your

ssh command. There are two different ways to do that, as seen below.

```
$ ssh user@linuxconfig.org  
OR  
$ ssh -l user linuxconfig.org
```

Example 3 The default port for SSH to listen on is 22. If the remote system is running the SSH service on some non default port, you can specify that port with the **-p** option in your command. The following example shows how you would SSH into a remote system that's running the service on port 2210.

```
$ ssh -p 2210 user@linuxconfig.org
```

Example 4 The **-v** (verbose) option will give you details about the connection process of SSH. This is useful when troubleshooting a troublesome connection.

```
$ ssh -v user@linuxconfig.org
```

Example 5 To increase verbosity even further, you can use the **-vvv** option.

```
$ ssh -vvv user@linuxconfig.org
```

NOTE

You can always use the [man command](#) to read more about the **ssh** command and its official documentation. Click the previous link to see how to open the manual pages for any command on a Linux system.

Advanced Usage

The **ssh** command is normally pretty simple. However, it can be used in some pretty advanced ways. Probably the most advanced usage of SSH would be in the case of port forwarding. That's outside the scope of this tutorial, but you can read about it in our guide on [port forwarding with SSH](#).

Another useful and somewhat advanced feature of SSH is logging in without a password. It involves the configuration of RSA keys, which is actually a lot more secure than logging in with a password, and as you can imagine, a lot more convenient. This is an almost essential step for system administrators that continually login to the same remote systems and have to specify a password over and over again. You can read about [configuring passwordless SSH logins](#) in our other guide.

ssh command in Linux Advanced Examples

Example 1 Use the **-4** or **-6** options with the **ssh** command to specify IPv4 or IPv6 connections only, respectively.

```
$ ssh -4 user@linuxconfig.org
OR
$ ssh -6 user@linuxconfig.org
```

Example 2 To enable X forwarding with SSH, use the **-X** option in your command.

```
$ ssh -X user@linuxconfig.org
```

Example 3 The **-C** option can be used to enable compression for your SSH session. Things like standard input and standard error will be compressed with before being sent to your terminal. This option is not very common nowadays

because of widespread fast connections, but it was once pretty common to see on older modems.

```
$ ssh -C user@linuxconfig.org
```


Closing Thoughts

In this guide, we learned all about the **ssh** command on Linux. The **ssh** command is by far the most widespread and commonly supported way to manage remote systems at the command line. We covered all the most basic and advanced options that you will need to know with the command, although even more exist. SSH is a very secure and convenient way to manage remote Linux systems as well as other network devices.

Related Linux Tutorials:

- [How to allow port through firewall on AlmaLinux](#)
- [How to make the most of OpenSSH](#)
- [Enable/Disable Firewall on Manjaro Linux](#)
- [Things to do after installing Ubuntu 20.04 Focal Fossa Linux](#)
- [Ubuntu 20.04 SSH Server](#)
- [Things to install on Ubuntu 20.04](#)
- [How to analyze and interpret Apache Webserver Log](#)
- [How to disable/enable firewall on AlmaLinux](#)
- [How to generate and manage ssh keys on Linux](#)
- [Install Arch Linux in VMware Workstation](#)

Commands

 bash, beginner, commands, terminal

< [ls-\(1\) manual page](#)

> [mkdir-\(1\) manual page](#)

NEWSLETTER

Subscribe to Linux Career Newsletter to receive latest news, jobs, career advice and featured configuration tutorials.

SUBSCRIBE

WRITE FOR US

LinuxConfig is looking for a technical writer(s) geared towards GNU/Linux and FLOSS technologies. Your articles will feature various GNU/Linux configuration tutorials and FLOSS technologies used in combination with GNU/Linux operating system.

When writing your articles you will be expected to be able to keep up with a technological advancement regarding the above mentioned technical area of expertise. You will work independently and be able to produce at minimum 2 technical articles a month.

APPLY NOW

CONTACT US

web (at) linuxconfig (dot) org

TAGS

18.04 administration [apache](#) [applications](#) [backup](#) [bash](#) [beginner](#) [browser](#)

centos centos8 **commands** database debian **desktop development** docker
fedora **filesystem** firewall gaming gnome Hardware **installation** java kali manjaro
multimedia networking nvidia **programming** python redhat rhel8 scripting
security server ssh storage **terminal ubuntu ubuntu 20.04** video virtualization
webapp **webserver**

FEATURED TUTORIALS

[How to install the NVIDIA drivers on Ubuntu 20.04 Focal Fossa Linux](#)

[Bash Scripting Tutorial for Beginners](#)

[How to check CentOS version](#)

[How to find my IP address on Ubuntu 20.04 Focal Fossa Linux](#)

[Ubuntu 20.04 Remote Desktop Access from Windows 10](#)

[Howto mount USB drive in Linux](#)

[How to install missing ifconfig command on Debian Linux](#)

[AMD Radeon Ubuntu 20.04 Driver Installation](#)

[Ubuntu Static IP configuration](#)

[How to use bash array in a shell script](#)

[Linux IP forwarding – How to Disable/Enable](#)

[How to install Tweak Tool on Ubuntu 20.04 LTS Focal Fossa Linux](#)

[How to enable/disable firewall on Ubuntu 18.04 Bionic Beaver Linux](#)

[Netplan static IP on Ubuntu configuration](#)

[How to change from default to alternative Python version on Debian Linux](#)

[Set Kali root password and enable root login](#)

[How to Install Adobe Acrobat Reader on Ubuntu 20.04 Focal Fossa Linux](#)

[How to install the NVIDIA drivers on Ubuntu 18.04 Bionic Beaver Linux](#)

[How to check NVIDIA driver version on your Linux system](#)

[Nvidia RTX 3080 Ethereum Hashrate and Mining Overclock settings on HiveOS Linux](#)

LATEST TUTORIALS

How to build a Tkinter application using an object oriented approach

How to disable SELinux on CentOS 7

Getting started with Tkinter for Python tutorial

Zsh shell installation and configuration on Linux

How to use LUKS with a detached header

How to scrape web pages from the command line using htmlq

How to install and manage fonts on Linux

Kali http server setup

yt-dlp vs youtube-dl

How to manage Bash history

How to install the NVIDIA drivers on Ubuntu 22.04

How to create a flatpak package

How to set, change and delete music tags with Mutagen

Introduction to MySQL storage engines

Introduction to crypttab with examples

How to create temporary files using mktemp on Linux

How to crash Linux

How to build an initramfs using Dracut on Linux

How to uncompress and list an initramfs content on Linux

How to install PipeWire on Ubuntu Linux