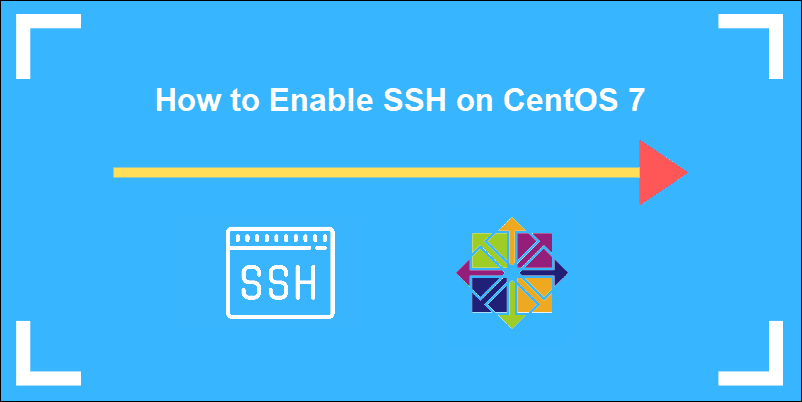
Introduction

**Secure Shell** (SSH) is a cryptographic protocol that allows a client to interact with a remote server in a secure environment.

High-level encryption protects the exchange of sensitive information and allows flie trans or issue commands on remote machines securely.

**Learn how to enable SSH on CentOS 7**by following the instructions in this short tutorial.



**Outlet Servers Starting at Only $30.00/mo**

[CONFIGURE NOW](https://phoenixnap.com/servers/dedicated)

Prerequisites

* CentOS 7 system to act as an SSH server
* A user with necessary permissions
* Access to a command line (Ctrl-Alt-T)
* **yum** utility (included by default)

**Installing and Enabling OpenSSH on CentOS 7**

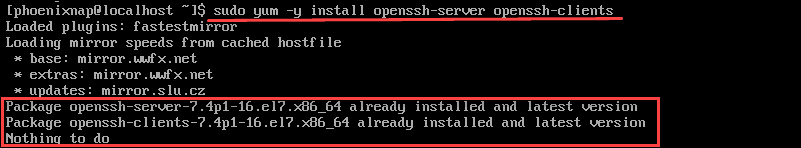
SSH software packages are included on CentOS by default. However, if these packages are not present on your system, easily install them by completing Step 1, outlined below.

**Step 1: Install OpenSSH Server Software Package**

Enter the following command from your terminal to start the installation process:

sudo yum –y install openssh-server openssh-clients

This command installs both the OpenSSH client applications, as well as the OpenSSH server daemon, **sshd**.



In this example, the system informs us that the latest version is already present.

**Step 2: Starting SSH Service**

To start the **SSH daemon** on the OpenSSH server:

sudo systemctl start sshd

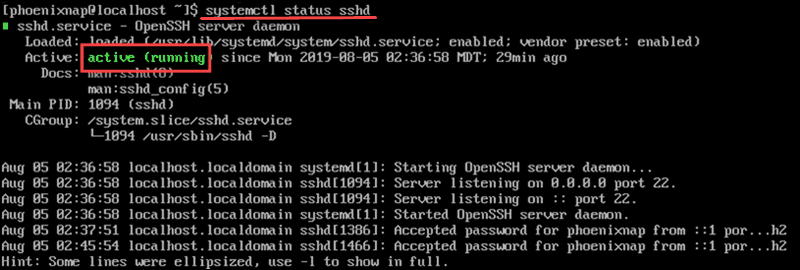
When active, **sshd** continuously listens for client connections from any of the client tools. When a connection request occurs, **sshd** sets up the correct connection.

**Step 3: Check sshd status**

Check the status of the SSH daemon:

sudo systemctl status sshd

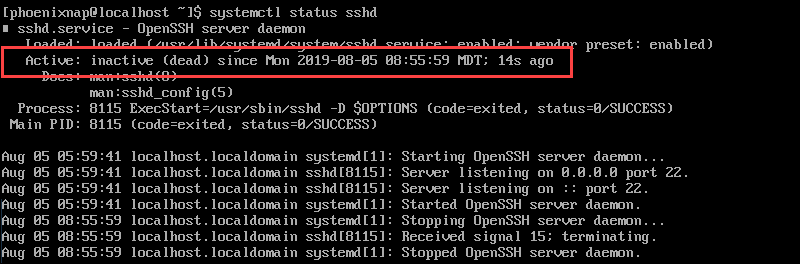
As we have previously started the service, the output confirms that it is active.



To stop the SSH daemon enter:

systemctl stop sshd

We can check if the [service has stopped](https://phoenixnap.com/kb/start-stop-restart-linux-services) by verifying the status. The output shows that the service is inactive and the time and date when the status last changed.



**Step 4: Enable OpenSSH Service**

Enable SSH to start automatically after each system reboot by using the **systemctl** command:

sudo systemctl enable sshd

To disable SSH after reboot enter:

sudo systemctl disable sshd

**Enterprise-Grade Dedicated Servers Starting at Only  $90.00/mo**

[CONFIGURE NOW](https://phoenixnap.com/servers/dedicated)

**OpenSSH Server Configuration**

Properly configuring the **sshd** configuration file [hardens server security](https://phoenixnap.com/kb/server-security-tips). The most common settings to enhance security are changing the port number, disabling root logins, and limiting access to only certain users.

To edit these settings access the /**etc/ssh/sshd\_config** file:

sudo vim /etc/ssh/sshd\_config

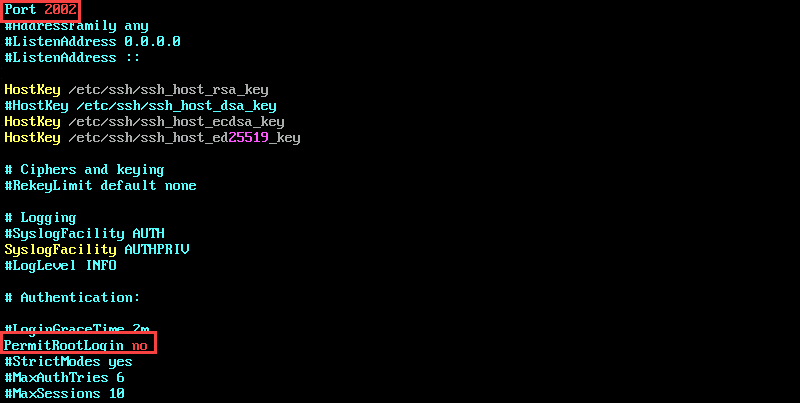
Once you access the file by using a text editor (in this example we used **vim**), you can disable root logins and edit the default port number:

* To disable root login:

***PermitRootLogin no***

* Change the SSH port to run on a non-standard port. For example:

***Port 2002***



Remember to uncomment the lines that you edit by removing the hashtag.

Save and close the file. Restart **sshd**:

service sshd restart

**Note**: We recommend you [generate SSH keys](https://phoenixnap.com/kb/how-to-generate-ssh-key-centos-7) for authentication, as a safer alternative to passwords.

**Firewall Settings**

After successfully enabling SSH and configuring the **sshd** file, adjust the firewall settings to make sure there are no compatibility issues.

It is also possible to restrict IP access to make the connection even more secure.

To restrict IP access, edit the **iptables** file by typing:

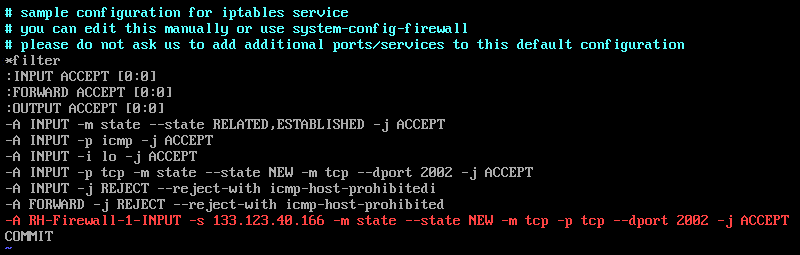
sudo vim /etc/sysconfig/iptables

To allow access using the port defined in the sshd config file, add the following line to the iptables file:

-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 2002 -j ACCEPT

To restrict access to a specific IP, for example 133.123.40.166, edit the line as follows:

-A RH-Firewall-1-INPUT -s 133.123.40.166 -m state --state NEW -p tcp --dport 2002 -j ACCEPT



If your site uses IPv6, and you are editing ip6tables, use the line:

-A RH-Firewall-1-INPUT -m tcp -p tcp --dport 2002 -j ACCEPT

Save and exit the file by pressing Escape (Esc) on your keyboard and typing:

**:X**

Press Enter to confirm.

Restart iptables to apply the changes:

sudo systemctl restart iptables

Conclusion

In this tutorial, we learned how to enable SSH on a CentOS 7 server. Additionally, we configured your firewall and SSH rules to limit access.

Your CentOS 7 server is now able to accept SSH connections.