

Bonito Designs

Design Your Dream Home Now



# OpenSSH Config File Examples

Author: Vivek Gite

Last updated: November 22, 2020

[16 comments](#)

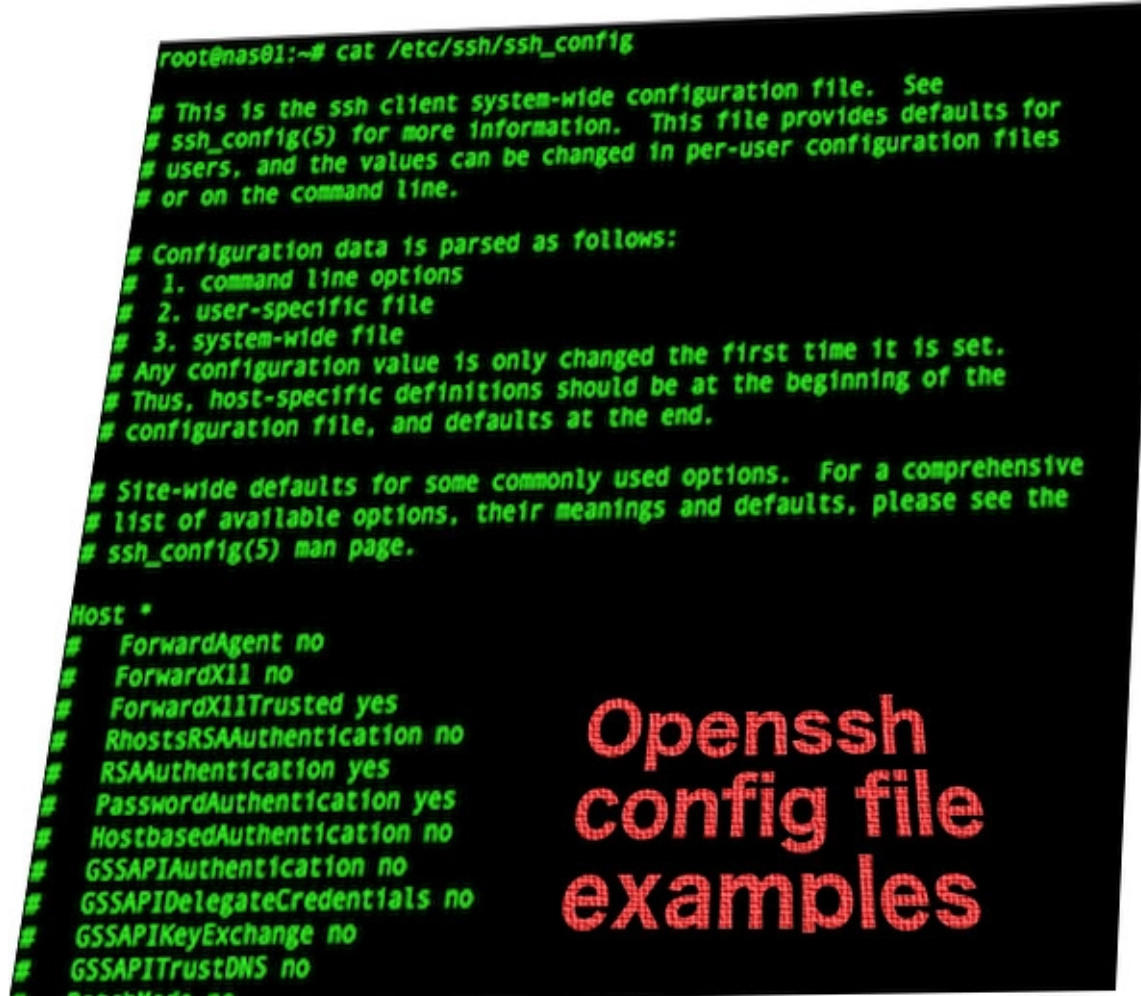
**H**ow do I create and setup an OpenSSH config file to create shortcuts for servers I frequently access under Linux or Unix desktop operating systems?



A global or local configuration file for SSH client can create shortcuts for sshd server including advanced ssh client options.

Tutorial details	
Difficulty level	<a href="#">Intermediate</a>
Root privileges	<a href="#">Yes</a>
Requirements	OpenSSH client
Est. reading time	15m

You can configure your OpenSSH ssh client using various files as follows to save time and typing frequently used ssh client command line options such as port, user, hostname, identity-file and much more:



Let use see some common OpenSSH config file examples.



## System-wide OpenSSH config file client configuration

1. `/etc/ssh/ssh_config` : This files set the default configuration for all users of OpenSSH clients on that desktop/laptop and it must be readable by all users on the system.

# User-specific OpenSSH file client configuration

1. `~/.ssh/config` or `$HOME/.ssh/config` : This is user's own configuration file which, overrides the settings in the global client configuration file, `/etc/ssh/ssh_config`.

## ~/.ssh/config file rules

The rules are as follows to create an ssh config file:

- You need to edit `~/.ssh/config` with a text editor such as vi.
- One config parameter per line is allowed in the configuration file with the parameter name followed by its value or values. The syntax is:

```
config value
config1 value1 value2
```

- You can use an equal sign (=) instead of whitespace between the parameter name and the values.

```
config=value
config1=value1 value2
```

- All empty lines are ignored.
- All lines starting with the hash (#) are ignored.
- All values are case-sensitive, but parameter names are not.

**Tip :** If this is a brand new Linux, Apple OS X/Unix box, or if you have never used ssh before create the `~/.ssh/` directory first using the following syntax:

```
mkdir -p $HOME/.ssh
```

```
chmod 0700 $HOME/.ssh
```

## Examples

For demonstration purpose my sample setup is as follows:

1. Local desktop client – Apple OS X or Ubuntu Linux.
2. Remote Unix server – OpenBSD server running latest OpenSSH server.
3. Remote OpenSSH server ip/host: 75.126.153.206 (server1.cyberciti.biz)
4. Remote OpenSSH server user: nixcraft
5. Remote OpenSSH port: 4242
6. Local ssh private key file path : /nfs/shared/users/nixcraft/keys/server1/id\_rsa

Based upon the above information my ssh command is as follows:

```
$ ssh -i /nfs/shared/users/nixcraft/keys/server1/id_rsa -p 4242  
nixcraft@server1.cyberciti.biz
```

OR

```
$ ssh -i /nfs/shared/users/nixcraft/keys/server1/id_rsa -p 4242 -l nixcraft  
server1.cyberciti.biz
```

You can avoid typing all of the ssh command parameters while logging into a remote machine and/or for executing commands on a remote machine. All you have to do is create an ssh config file. Open the Terminal application and create your config file by typing the following command:

```
## edit file in $HOME dir
```

```
vi ~/.ssh/config
```

OR

```
## edit file in $HOME dir
```

```
vi $HOME/.ssh/config
```

Add/Append the following config option for a shortcut to server1 as per our sample setup:

```
Host server1  
    HostName server1.cyberciti.biz  
    User nixcraft
```

Port 4242

IdentityFile /nfs/shared/users/nixcraft/keys/server1/id\_rsa

[Save and close the file in vi/vim](#) by pressing `Esc` key, type `:w` and hit `Enter` key. To open your new SSH session to server1.cyberciti.biz by typing the following command:

```
$ ssh server1
```

## Adding another host

Append the following to your `~/.ssh/config` file:

```
Host nas01
  HostName 192.168.1.100
  User root
  IdentityFile ~/.ssh/nas01.key
```

You can simply type:

```
$ ssh nas01
```

## Putting it all together

Here is my sample `~/.ssh/config` file that explains and create, design, and evaluate different needs for remote access using ssh client:

```
### default for all ##
Host *
  ForwardAgent no
  ForwardX11 no
  ForwardX11Trusted yes
  User nixcraft
  Port 22
  Protocol 2
  ServerAliveInterval 60
  ServerAliveCountMax 30

## override as per host ##
Host server1
  HostName server1.cyberciti.biz
  User nixcraft
  Port 4242
  IdentityFile /nfs/shared/users/nixcraft/keys/server1/id_rsa

## Home nas server ##
```

```
Host nas01
    HostName 192.168.1.100
    User root
    IdentityFile ~/.ssh/nas01.key

## Login AWS Cloud ##
Host aws.apache
    HostName 1.2.3.4
    User wwwdata
    IdentityFile ~/.ssh/aws.apache.key

## Login to internal lan server at 192.168.0.251 via our public uk office ssh based gateway using ##
## $ ssh uk.gw.lan ##
Host uk.gw.lan uk.lan
    HostName 192.168.0.251
    User nixcraft
    ProxyCommand ssh nixcraft@gateway.uk.cyberciti.biz nc %h %p 2> /dev/null

## Our Us Proxy Server ##
## Forward all local port 3128 traffic to port 3128 on the remote vps1.cyberciti.biz server ##
## $ ssh -f -N proxyus ##
Host proxyus
    HostName vps1.cyberciti.biz
    User breakfree
    IdentityFile ~/.ssh/vps1.cyberciti.biz.key
    LocalForward 3128 127.0.0.1:3128
```

---

## Understanding ~/.ssh/config entries

- **Host** : Defines for which host or hosts the configuration section applies. The section ends with a new Host section or the end of the file. A single `*` as a pattern can be used to provide global defaults for all hosts.
- **HostName** : Specifies the real host name to log into. Numeric IP addresses are also permitted.
- **User** : Defines the username for the SSH connection.
- **IdentityFile** : Specifies a file from which the user's DSA, ECDSA or DSA authentication identity is read. The default is `~/.ssh/identity` for protocol version 1, and `~/.ssh/id_dsa`, `~/.ssh/id_ecdsa` and `~/.ssh/id_rsa` for protocol version 2.
- **ProxyCommand** : Specifies the command to use to connect to the server. The command string extends to the end of the line, and is executed with the user's shell. In the command string, any occurrence of `%h` will be substituted by the host name to connect, `%p` by the port, and `%r` by the remote user name. The command can be basically anything, and should read from its standard input and write to its standard output. This directive is useful in conjunction with `nc(1)` and its proxy support. For example, the following directive would connect via an HTTP proxy at

192.1.0.253:

```
ProxyCommand /usr/bin/nc -X connect -x 192.1.0.253:3128 %h %p
```

- **LocalForward** : Specifies that a TCP port on the local machine be forwarded over the secure channel to the specified host and port from the remote machine. The first argument must be [bind\_address:]port and the second argument must be host:hostport.
- **Port** : Specifies the port number to connect on the remote host.
- **Protocol** : Specifies the protocol versions ssh(1) should support in order of preference. The possible values are 1 and 2.
- **ServerAliveInterval** : Sets a timeout interval in seconds after which if no data has been received from the server, ssh(1) will send a message through the encrypted channel to request a response from the server. See blogpost “[Open SSH Server connection drops out after few or N minutes of inactivity](#)” for more information.
- **ServerAliveCountMax** : Sets the number of server alive messages which may be sent without ssh(1) receiving any messages back from the server. If this threshold is reached while server alive messages are being sent, ssh will disconnect from the server, terminating the session.

## Speed up ssh session

Multiplexing is nothing but send more than one ssh connection over a single connection. OpenSSH can reuse an existing TCP connection for multiple concurrent SSH sessions. This results into reduction of the overhead of creating new TCP connections. Update your ~/.ssh/config:

```
Host server1
    HostName server1.cyberciti.biz
    ControlPath ~/.ssh/controlmasters/%r@%h:%p
    ControlMaster auto
```

See “[Linux / Unix: OpenSSH Multiplexer To Speed Up OpenSSH Connections](#)” for more info. In this example, I go [through one host to reach another server i.e. jump host using ProxyCommand](#):

```
## ~/.ssh/config ##
Host internal
    HostName 192.168.1.100
    User vivek
    ProxyCommand ssh vivek@vpn.nixcraft.net.in -W %h:%p
    ControlPath ~/.ssh/controlmasters/%r@%h:%p
    ControlMaster auto
```

For more info see following tutorials:

- [How To Reuse SSH Connection To Speed Up Remote Login Process Using Multiplexing](#)
- [How To Setup SSH Keys on a Linux / Unix System](#)

## A note about shell aliases (outdated method)



**WARNING!** This bash shell aliased based setup may work out for you. However, I recommend that you use `~/.ssh/config` file for better results in a long run. SSH config file is more advanced and elegant solutions. The alias command only used here for demo purpose and it is here due to historical reasons.

[An alias is nothing but shortcut to commands](#) and you can [create the alias](#) use the following syntax in your [~/.bashrc file](#):

```
## create a new bash shell alias as follow ##
```

```
alias server1="ssh -i /nfs/shared/users/nixcraft/keys/server1/id_rsa -p 4242 nixcraft@server1.cyberciti.biz"
```

Then, to ssh into the server1, instead of typing full `ssh -i`

`/nfs/shared/users/nixcraft/keys/server1/id_rsa -p 4242 nixcraft@server1.cyberciti.biz`

command, you would only have to type the command 'server1' and press the [ENTER] key:

```
$ server1
```

### References

- See `ssh_config` man page for more information on syntax and some of the other available options.
- [Top 20 OpenSSH Server Best Security Practices](#)




Get the latest tutorials on Linux, Open Source & DevOps via [RSS feed](#) or [Weekly email newsletter](#).



16 comments so far... [add one](#) ↓



 To search, type & hit enter...

## Related Tutorials

[Linux / Unix: OpenSSH Multiplexer To Speed Up...](#)

[RHEL 6: Change OpenSSH Port To 1255 \( SELinux Config \)](#)

[Linux / UNIX View Only Configuration File Directives...](#)

[Squid test config file for syntax errors](#)

[Apache: Make Changes In a Config File Take Effect...](#)

[HowTo: Use Auto Config Proxy PAC File For Specific Domain](#)

[Gracefully Restart Nginx Web Server After Changes...](#)

Category	List of Unix and Linux commands
Disk space analyzers	<a href="#">ncdu</a> • <a href="#">pydf</a>
File Management	<a href="#">cat</a>
Firewall	<a href="#">Alpine Awall</a> • <a href="#">CentOS 8</a> • <a href="#">OpenSUSE</a> • <a href="#">RHEL 8</a> • <a href="#">Ubuntu 16.04</a> • <a href="#">Ubuntu 18.04</a> • <a href="#">Ubuntu 20.04</a>
Network Utilities	<a href="#">NetHogs</a> • <a href="#">dig</a> • <a href="#">host</a> • <a href="#">ip</a> • <a href="#">nmap</a>

Category	List of Unix and Linux commands
OpenVPN	<a href="#">CentOS 7</a> • <a href="#">CentOS 8</a> • <a href="#">Debian 10</a> • <a href="#">Debian 8/9</a> • <a href="#">Ubuntu 18.04</a> • <a href="#">Ubuntu 20.04</a>
Package Manager	<a href="#">apk</a> • <a href="#">apt</a>
Processes Management	<a href="#">bg</a> • <a href="#">chroot</a> • <a href="#">cron</a> • <a href="#">disown</a> • <a href="#">fg</a> • <a href="#">jobs</a> • <a href="#">killall</a> • <a href="#">kill</a> • <a href="#">pidof</a> • <a href="#">pstree</a> • <a href="#">pwdx</a> • <a href="#">time</a>
Searching	<a href="#">grep</a> • <a href="#">whereis</a> • <a href="#">which</a>
User Information	<a href="#">groups</a> • <a href="#">id</a> • <a href="#">lastcomm</a> • <a href="#">last</a> • <a href="#">lid/libuser-lid</a> • <a href="#">logname</a> • <a href="#">members</a> • <a href="#">users</a> • <a href="#">whoami</a> • <a href="#">who</a> • <a href="#">w</a>
WireGuard VPN	<a href="#">Alpine</a> • <a href="#">CentOS 8</a> • <a href="#">Debian 10</a> • <a href="#">Firewall</a> • <a href="#">Ubuntu 20.04</a>

16 comments... [add one](#) ↓

**scott carlson**

Oct 13, 2013 @ 20:30

Nice examples.... here is what mine starts with, because I use the control master all the time. Which you can start in the background with “ssh -MNf host”.

I’ve found that attempting GSSApiAuth slows everything down, so I turn that off, and I’ve picked the order of the ciphers to be faster as well.

Also with newer versions of ssh, you don’t need netcat anymore for proxying, you can use this line in the host definition instead: “ProxyCommand ssh -W %h:%p”

Last comment, is that I recommend, using multiple aliases for the host, and include all possible ways you’ll refer to the box. So that if you cut and paste a name, you’ll still get the same settings. As an example, if your DNS search path has company.com, then you might do this: “host web webserver.priv.city webserver.priv.city.company.com ”

```
# ssh -Mnf starbuck sleep 30d
Host *
ControlPath ~/.ssh/%l-master-%r@%h:%p
ControlMaster auto
ServerAliveInterval 60
GSSAPIAuthentication no
```

Ciphers arcfour256,arcfour128,arcfour,blowfish-cbc,aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-cbc,3des-cbc,cast128-cbc,aes192-cbc,aes256-cbc,arcfour  
ForwardAgent yes  
LogLevel quiet

[reply](#) [link](#)

---

## Just a lurker

Jul 9, 2014 @ 15:18

This is really very nice article, however I would discuss the place of the default config, because of the manual:

For each parameter, the first obtained value will be used.

The configuration files contain sections separated by “Host” specifications, and that section is only applied for hosts that match one of the patterns given in the specification.

The matched host name is the one given on the command line.

Since the first obtained value for each parameter is used, more host-specific declarations should be given near the beginning of the file, and general defaults at the end.

...so I have my defaults at the end.

Thanks!

[reply](#) [link](#)

---

## Olivier Mengué (DOLMEN)

Jul 14, 2014 @ 7:07

For Github you may be interested in a tool I wrote that completely automates this setup using the most secure settings: github-keygen

<https://github.com/dolmen/github-keygen/>

[reply](#) [link](#)

---

## SAFDAR

Aug 7, 2015 @ 20:44

Is there a limit on number of IdentityFile we can use in config file under one block?

example:

Host server1

HostName server1.cyberciti.biz

User nixcraft

Port 4242

IdentityFile /nfs/shared/users/nixcraft/keys/server1/id\_rsa

IdentityFile /nfs/shared/users/nixcraft/keys/server1/id\_rsa1

IdentityFile /nfs/shared/users/nixcraft/keys/server1/id\_rsa2

IdentityFile /nfs/shared/users/nixcraft/keys/server1/id\_rsa3

.

.

.

IdentityFile /nfs/shared/users/nixcraft/keys/server1/id\_rsa10

[reply](#) [link](#)

---

**Amit kumar**

Jul 28, 2016 @ 16:56

issue get resolved after saving the cipher values in /etc/ssh/ssh\_config file..

Really thanks for this article.

[reply](#) [link](#)

---

**walee**

Jul 29, 2016 @ 15:28

I have a problem win I enable public key on fedora 23 or centos 7 I receive this wrong “Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

[reply](#) [link](#)

---

**Chris Duong**

Dec 30, 2016 @ 4:43

Is there any use case for using “Include Statement”. I had done this, the file is read but the hostname does not work.

[reply](#) [link](#)

---

**Nurd**

Mar 10, 2017 @ 17:50

Thanks! Useful article.

[reply](#) [link](#)

---

**dedRabbitt**

Apr 22, 2017 @ 15:52

Very nice!

[reply](#) [link](#)

---

**Bill**

Oct 10, 2017 @ 8:57

This is one of the most useful, simple and informative pages I've ever read in the technical context! :)

[reply](#) [link](#)

---

 **Vivek Gite**

Oct 10, 2017 @ 16:04

thank you!

[reply](#) [link](#)

---

**John**

Dec 1, 2017 @ 0:22

Thanks so much, I totally agree with Bill—superb post.

[reply](#) [link](#)

---

**Andrew McGlashan**

Aug 18, 2020 @ 9:03

There is a problem with the “putting it all together” example.

As you can see from the following, if you define something, then it cannot be redefined later. You need to “\*” grouping at the end of the file to catch things that aren’t yet defined for a “Host” entry. Two example config files and attempts to use them shown below demonstrate this fact.

```
$ cat /tmp/configx
```

```
Host *
```

```
    Port 24
```

```
    Protocol 2
```

```
Host sadsack
```

```
    Port 333
```

```
    Hostname aaa
```

```
andrewm@mx-hvk-1:/tmp
```

```
$ ssh -F /tmp/configx sadsack
```

ssh: connect to host aaa port 24: Connection refused

```
$ cat /tmp/configy
```

```
Host sadsack
```

```
    Port 333
```

```
    Hostname aaa
```

```
Host *
```

```
    Port 24
```

```
    Protocol 2
```

```
andrewm@mx-hvk-1:/tmp
```

```
$ ssh -F /tmp/configy sadsack
```

ssh: connect to host aaa port 333: Connection refused

## Dietmar (in Western Germany)

Oct 18, 2020 @ 11:33

Great page! The whole site is of outstanding quality an reliability!!

.

One demand about ssh config file is not covered:

.

What, if I want to “land” in a particulart directory? Can you cover this question?

.

The only solution I found elsewhere:

```
sudo vim .ssh/config  
add
```

```
## needs both!!:  
RequestTTY yes  
RemoteCommand  cd /srv/terra-daten/; exec $SHELL
```

May be you find a better solution?

.

Much appreciate your work!

[reply](#) [link](#)

---

## ömer

Nov 22, 2020 @ 11:08

Thank you in advance.

My problem is that I couldn't “Save and close the file”. After Add/append config, how can I exit that screen?

[reply](#) [link](#)

---



**Vivek Gite**

Nov 22, 2020 @ 12:38

Are you using vim or vi? If so see:

[Vi / Vim Save And Quit The Editor Command](#)

[reply](#) [link](#)

## Leave a Reply

Your email address will not be published.

Comment

Name

Email

Website

Post Comment


Use HTML `<pre>...</pre>` for code samples. Still have questions? [Post it on our forum](#)

---

Next FAQ: [Linux Change Password Using `passwd` Command Over SSH](#)

Previous FAQ: [SSH Into Google Cloud Compute Engine Instance Using Secure Shell Client](#)



 To search, type & hit enter...

## FEATURED ARTICLES

1

[30 Cool Open Source Software I Discovered in 2013](#)

---

2

[30 Handy Bash Shell Aliases For Linux / Unix / Mac OS X](#)

---

3

[Top 32 Nmap Command Examples For Linux Sys/Network Admins](#)

---

4

[25 PHP Security Best Practices For Linux Sys Admins](#)

---

5

[30 Linux System Monitoring Tools Every SysAdmin Should Know](#)

---

6

[40 Linux Server Hardening Security Tips](#)

---

7

[Linux: 25 Iptables Netfilter Firewall Examples For New SysAdmins](#)

---

8

[Top 20 OpenSSH Server Best Security Practices](#)

---

9

[Top 25 Nginx Web Server Best Security Practices](#)

---

10

[My 10 UNIX Command Line Mistakes](#)

---

Sign up for my newsletter

