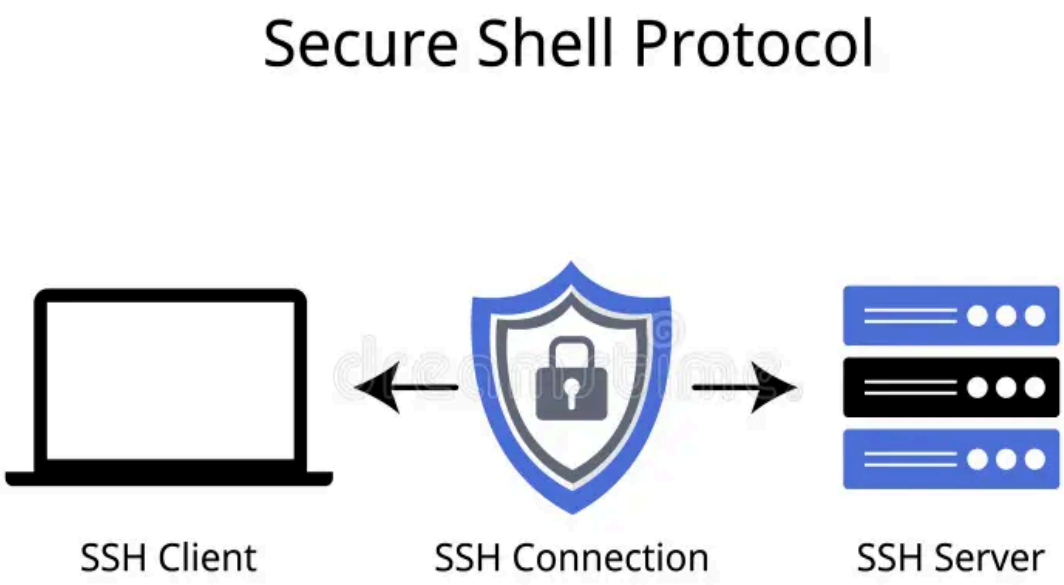
SSH Server Setup (RHEL 9 & opensuse)



★ What is SSH ?

SSH (Secure Shell) is a protocol used to securely connect to remote systems over a network. It encrypts all traffic and ensures secure authentication, making it essential for system administration and remote access.

Kartik Akade

★ How SSH Works Across Linux Systems:

- SSH server runs on the host you want to access remotely (usually on port 22)
- SSH clients (like `ssh`, `scp`, or `sftp`) connect securely to the server
- You can authenticate using a password or a key pair (passwordless login)
- Works seamlessly in the same subnet or over the internet with proper firewall rules

For server side configurations I have RHEL 9

```
[root@localhost ~]# cat /etc/os-release
NAME="Red Hat Enterprise Linux"
VERSION="9.0 (Plow)"
ID="rhel"
ID_LIKE="fedora"
VERSION_ID="9.0"
PLATFORM_ID="platform:el9"
PRETTY_NAME="Red Hat Enterprise Linux 9.0 (Plow)"
ANSI_COLOR="0;31"
LOGO="fedora-logo-icon"
CPE_NAME="cpe:/o:redhat:enterprise_linux:9::baseos"
HOME_URL="https://www.redhat.com/"
DOCUMENTATION_URL="https://access.redhat.com/documentation/red_hat_enterprise_li
nux/9/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"

REDHAT_BUGZILLA_PRODUCT="Red Hat Enterprise Linux 9"
REDHAT_BUGZILLA_PRODUCT_VERSION=9.0
REDHAT_SUPPORT_PRODUCT="Red Hat Enterprise Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="9.0"
[root@localhost ~]#
```

Kartik Akade

And for client side configurations I have opensuse

```
linux-qb9x:~ # cat /etc/os-release
NAME="openSUSE Leap"
VERSION="15.0"
ID="opensuse-leap"
ID_LIKE="suse opensuse"
VERSION_ID="15.0"
PRETTY_NAME="openSUSE Leap 15.0"
ANSI_COLOR="0;32"
CPE_NAME="cpe:/o:opensuse:leap:15.0"
BUG_REPORT_URL="https://bugs.opensuse.org"
HOME_URL="https://www.opensuse.org/"
linux-qb9x:~ #
```

🔐 1. SSH with Password Authentication

Step 1:-Check status server side If it is not active

```
[root@localhost ~]# systemctl status sshd
○ sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: ena>
     Active: inactive (dead) since Sat 2025-06-14 15:59:49 IST; 6s ago
   Duration: 43min 8.009s
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 1099 ExecStart=/usr/sbin/sshd -D $OPTIONS (code=exited, status=0/S>
   Main PID: 1099 (code=exited, status=0/SUCCESS)
        CPU: 230ms
```

then activate it using this command

{ systemctl start sshd }

```
[root@localhost ~]# systemctl start sshd
[root@localhost ~]# systemctl status sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: ena>
     Active: active (running) since Sat 2025-06-14 16:01:22 IST; 2s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 3896 (sshd)
      Tasks: 1 (limit: 10755)
     Memory: 1.6M
        CPU: 35ms
     CGroup: /system.slice/sshd.service
             └─3896 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

Kartik Akade

Step 2:-📂 The main configuration file for the SSH server is

```
[root@localhost ~]# vi /etc/ssh/sshd_config
```

```
#       $OpenBSD: sshd_config,v 1.104 2021/07/02 05:11:21 dtucker Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/
sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

# To modify the system-wide sshd configuration, create a  *.conf  file under
#  /etc/ssh/sshd_config.d/  which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
```

Edit `/etc/ssh/sshd_config` and make sure these lines are set (remove `#` if present):

```
PermitRootLogin yes
```

```
PasswordAuthentication yes
```

Restart sshd using this command { systemctl restart sshd }

```
[root@localhost ~]# systemctl restart sshd
[root@localhost ~]#
```

Kartik Akade

Step 3:- On the server side, add the SSH service to the firewall & reload it.

```
[root@localhost ~]# firewall-cmd --permanent --add-service=ssh
success
[root@localhost ~]# systemctl reload firewalld.service
[root@localhost ~]# firewall-cmd --list-all
drop (active)
  target: DROP
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: https nfs ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@localhost ~]#
```

★ ssh Client:

- ssh username@<ip-address>
- `ssh`: Secure Shell command to start the connection.
- `username`: The user account on the remote server
- `<ip-address>`: The IP address of the remote system

Step 4: SSH Login Using Username and Server IP

```
linux-qb9x:~ # ssh root@192.168.8.57
The authenticity of host '192.168.8.57 (192.168.8.57)' can't be established.
ECDSA key fingerprint is SHA256:CgpwBgI3D+SSs8Y41KGNSNSG4bPlj5gTf9j92BKO/ZM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.8.57' (ECDSA) to the list of known hosts.
root@192.168.8.57's password:
Activate the web console with: systemctl enable --now cockpit.socket

Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Sat Jun 14 20:39:23 2025
[root@localhost ~]# ls
anaconda-ks.cfg  nagios_on_rhel-rockylinux  sample.txt  tree-1.8.0-10.el9.x86_64.rpm
gitesh.txt       nagiosonrocky              shiva.txt   tree-2.2.1-1.fc42.x86_64.rpm
kartik           nagiosonrocky.sh           T-20        tree-2.2.1-1.fc42.x86_64.rpm.1
[root@localhost ~]#
```

NOW :-

1. In server side, some files were created in a directory.

2. Now check in client side is that added files or data is visible in client side or not.

Kartik Akade

- **Server Side in rhel 9 :**

```
[root@localhost ~]# mkdir /jnec
[root@localhost ~]# cd /jnec
[root@localhost jnec]# touch kartik{1..10}.txt
[root@localhost jnec]# ls
kartik10.txt  kartik2.txt  kartik4.txt  kartik6.txt  kartik8.txt
kartik1.txt   kartik3.txt  kartik5.txt  kartik7.txt  kartik9.txt
[root@localhost jnec]#
```

- **Client Side in opensuse :**

```
[root@localhost ~]# ls -ld /jnec
drwxr-xr-x 2 root root 4096 Jun 14 20:52 /jnec
[root@localhost ~]# cd /jnec
[root@localhost jnec]# ls
kartik10.txt  kartik2.txt  kartik4.txt  kartik6.txt  kartik8.txt
kartik1.txt   kartik3.txt  kartik5.txt  kartik7.txt  kartik9.txt
[root@localhost jnec]#
```

🔑 2. SSH Without Password (Key-Based Authentication

Step 1:-📁 The main configuration file for the SSH server is

```
[root@localhost ~]# vi /etc/ssh/sshd_config
```

```
#       $OpenBSD: sshd_config,v 1.104 2021/07/02 05:11:21 dtucker Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/
sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

# To modify the system-wide sshd configuration, create a  *.conf  file under
#  /etc/ssh/sshd_config.d/  which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
```

Edit `/etc/ssh/sshd_config` and make sure these lines are set (remove `#` if present):

```
PermitRootLogin yes
```

```
PasswordAuthentication yes
```

Restart sshd using this command

{ systemctl restart sshd }

```
[root@localhost ~]# systemctl restart sshd
[root@localhost ~]#
```

Kartik Akade

Step 2: Generate SSH Key on Client

```
linux-qb9x:~ # ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:hJZ9QFbswXbgSkTQ8sWID0zswdFUh2Xu+vDIfZ6j2YI root@linux-qb9x
The key's randomart image is:
+---[RSA 2048]----+
|      =+XBB+++    |
|       O*+oO+.    |
|      .+*++oo.    |
|      ..ooo..     |
|         S   .    |
|             .    |
|           o.     |
|          .E*.oo. |
|           o *=+. |
+----[SHA256]-----+
linux-qb9x:~ #
```

Step 3: Copy Public Key to Server Using this command { ssh-copy-id username@server_ip }

```
linux-qb9x:~ # ssh-copy-id root@192.168.8.57
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host '192.168.8.57 (192.168.8.57)' can't be established.
ECDSA key fingerprint is SHA256:CgpwBgI3D+SSs8Y41KGNSNSG4bPlj5gTf9j92BKO/ZM.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any
 that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now
it is to install the new keys
root@192.168.8.57's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@192.168.8.57'"
and check to make sure that only the key(s) you wanted were added.
```

Kartik Akade

Step 4:- You will be able to connect without entering a password, as the SSH key will handle the authentication automatically.

```
linux-qb9x:~ # ls ~/.ssh/
id_rsa  id_rsa.pub  known_hosts
linux-qb9x:~ # ssh root@192.168.8.57
Activate the web console with: systemctl enable --now cockpit.socket

Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Sat Jun 14 21:56:05 2025 from 192.168.8.250
[root@localhost ~]#
```

- **Server Side in rhel 9 :**

```
[root@localhost ~]# ls
anaconda-ks.cfg             nagiosonrocky     T-20
gitesh.txt                  nagiosonrocky.sh  tree-1.8.0-10.el9.x86_64.rpm
kartik                      sample.txt        tree-2.2.1-1.fc42.x86_64.rpm
nagios_on_rhel-rockylinux   shiva.txt         tree-2.2.1-1.fc42.x86_64.rpm.1
[root@localhost ~]#
```

- **Client Side in opensuse :**

```
linux-qb9x:~ # ssh root@192.168.8.57
Activate the web console with: systemctl enable --now cockpit.socket

Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Sat Jun 14 22:00:09 2025 from 192.168.8.250
[root@localhost ~]# ls
anaconda-ks.cfg  nagios_on_rhel-rockylinux  sample.txt  tree-1.8.0-10.el9.x86_64.rpm
gitesh.txt       nagiosonrocky              shiva.txt   tree-2.2.1-1.fc42.x86_64.rpm
kartik           nagiosonrocky.sh           T-20        tree-2.2.1-1.fc42.x86_64.rpm.1
[root@localhost ~]#
```

Kartik Akade

🔐 Important:-

1. SSH works on port 22 by default make sure this port is allowed in the firewall.
2. The SSH service must be running on the server so it can accept connections.
3. The server and client should be connected to the same network, or able to reach each other.

✅ Conclusion:-

SSH (Secure Shell) is a secure and powerful protocol for remote access, command execution, and system administration across networked systems.

For aspiring DevOps or Linux engineers, gaining hands-on experience with SSH setup, key-based authentication, and secure communication practices is crucial. It deepens your understanding of secure remote management, automation, and Linux server administration — all fundamental skills in modern IT environments.

Kartik Akade