HOSTMAN

Log In Sign Up

Hostman › Tutorials › How to Install and Configure SSH on Ubuntu 22.04

How to Install and Configure SSH on Ubuntu 22.04

24.11.2023

Reading time: 5 min

Share

Ubuntu

Hostman Team

Technical writer

SSH is a network protocol that provides a secure connection between a client and a server. All communication is encrypted, preventing theft

The first thing you need to do before you start installing SSH on Ubuntu is to update all `apt` packages to the latest versions. To do this, use the following command:

```
sudo apt update && sudo apt upgrade
```

Step 2: Install SSH on Ubuntu

OpenSSH is not pre-installed on the system, so let's install it manually. To do this, type in the terminal:

```
sudo apt install openssh-server
```

The installation of all the necessary components will begin. Answer "Yes" to all the system prompts.

After the installation is complete, go to the next step to start the

On successful startup, you will see the following system message.

The `--now` key helps you launch the service and simultaneously set it to start when the system boots.

To verify that the service is enabled and running successfully, type:

```
sudo systemctl status ssh
```

The output should contain the `Active: active (running)` line, which indicates that the service is successfully running.

If you want to disable the service, execute:

```
sudo systemctl disable ssh
```

In our case, we have the UFW installed, so we will use the following

```
sudo ufw status
```

In the output, you should see that SSH traffic is allowed. If you don't have it listed, you need to allow incoming SSH connections. This command will help with this:

```
sudo ufw allow ssh
```

Step 5: Connect to the server

Once you complete all the previous steps, you can log into the server using the SSH protocol.

To do this, you will need the server's IP address or domain name and the name of a user created on the server.

In the terminal line, enter the command:

> Important: To successfully connect to a remote server, SSH must be installed and configured on the remote server and the user's computer from which you make the connection.

Step 6: Configure SSH

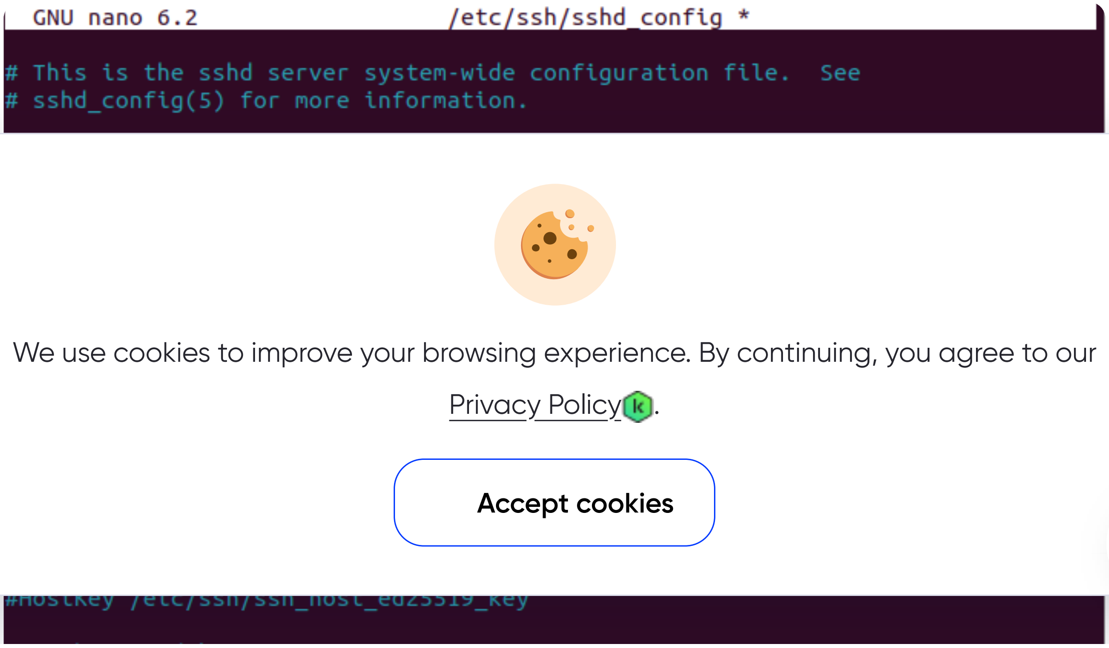
Having completed the previous five steps, you can already connect to the server remotely. However, you can further increase the connection's security by changing the default connection port to another or

If you get any errors after editing the configuration file, you can restore the original file without problems.

After creating the backup, you can proceed to edit the configuration file. To do this, open it using the `nano` editor:

```
sudo nano /etc/ssh/sshd_config
```

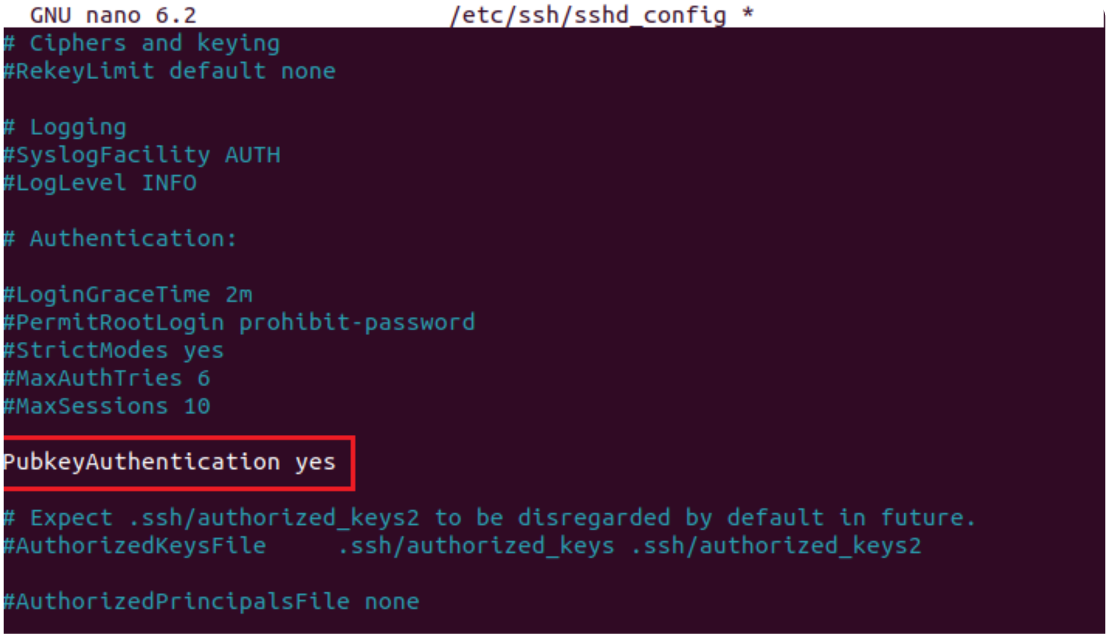
In the file, change the port to a more secure one. It is best to set values from the dynamic range of ports (49152 - 65535) and use different numbers for additional security. For example, let's change the port value to 49532. To do this, we uncomment the corresponding line in the file and change the port as shown in the screenshot below.



```
  GNU nano 6.2                    /etc/ssh/sshd_config *

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

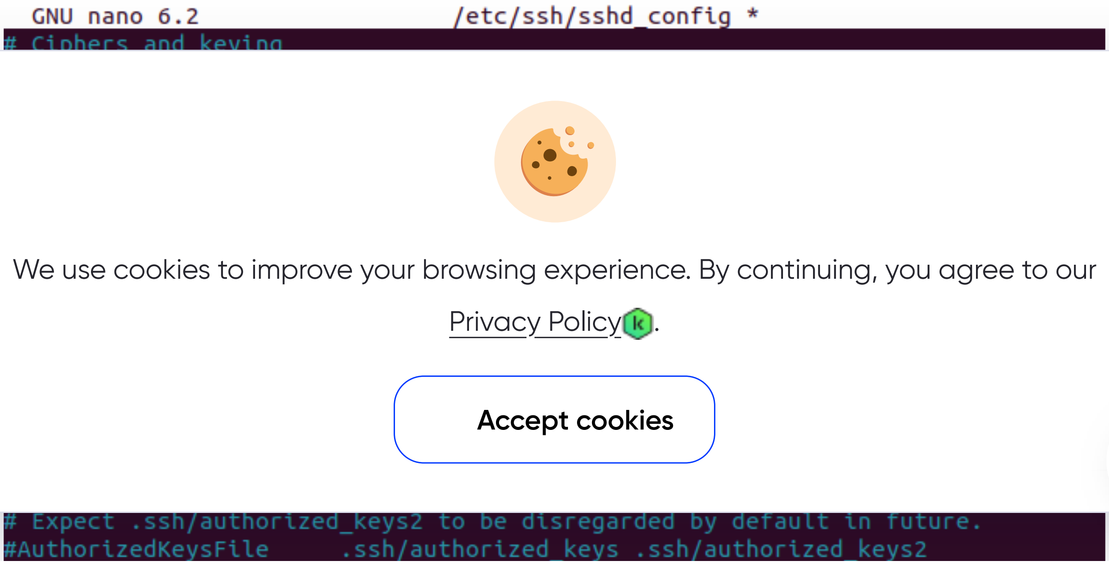
```
#HostKey /etc/ssh/ssh_host_ed25519_key
```

In addition to this setting, we recommend changing the password authentication mode to a more secure key authentication mode. To do this, uncomment the corresponding line and make sure the value is "Yes", as shown in the screenshot.



```
  GNU nano 6.2                    /etc/ssh/sshd_config *
# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none
```

Now, let's prohibit logging on to the server as a superuser by changing the corresponding line as shown in the picture below.



```
  GNU nano 6.2                    /etc/ssh/sshd_config *
# Ciphers and keying
```

```
# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2
```

security:

- `UseDNS` checks if the hostname matches its IP address. The value "Yes" enables this parameter.
- `PermitEmptyPasswords` prohibits using empty passwords for authentication if the value is "No."
- `MaxAuthTries` limits the number of unsuccessful attempts to connect to the server within one communication session.
- `AllowUsers` and `AllowGroups` are responsible for the list of users and groups allowed to access the server:

```
# AllowUsers User1, User2, User3
# AllowGroups Group1, Group2, Group3
```

- `Login GraceTime` sets the time provided for successful authorization. We recommend reducing the value of this parameter by four times.
- `ClientAliveInterval` limits the time of user inactivity. After

If you have changed the port in the configuration file, you should connect using the new port:

```
ssh -p port_number username@IP_address
```

Or:

```
ssh -p port_number_port_username@domain
```

Conclusion

This article presents a step-by-step guide on installing and configuring SSH in Ubuntu 22.04 and describes how to edit the main configuration file to improve security. We hope this guide helps you to set up a

Ubuntu

Popular content

HOSTMAN

Log In Sign Up

Products

Cloud Servers

Databases

App Platform

DBaaS

Cloud Server Hosting

VPS For Rent

Community

Tutorials

Blog

Docs

FAQ

Company

About

Sign Up

HOSTMAN LTD
28 Oktovriou, 367, Mediterranean Court, Floor 1,
Office A5, 3107, Limassol, Cyprus

Privacy Policy Terms of Service
Service Level Agreement