# **How to Install and Configure SSH on Ubuntu 22.04**

[How to install and configure SSH on Ubuntu 22.04: A Step-by-Step Guide | Hostman](https://hostman.com/tutorials/how-to-install-and-configure-ssh-on-ubuntu-22-04/)

## **Introduction**

SSH is a network protocol that provides a secure connection between a client and a server. All communication is encrypted, preventing theft of data transmitted over the network and other remote network attacks.

Let’s say you have ordered a cloud server from Hostman. You will need SSH installed and configured to connect to and administer the server.

This guide describes how to install SSH on Ubuntu 22.04 and configure it.

## **Step 1: Prepare Ubuntu**

Before installing SSH on Ubuntu, update all apt packages to their latest versions. 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 you need to install it manually. Run the following command in the terminal:

sudo apt install openssh-server

Answer "Yes" to all system prompts during installation.

Once the installation is complete, proceed to the next step to start the service.

## **Step 3: Start SSH**

Enable the SSH service you just installed using the following command:

sudo systemctl enable --now ssh

The --now flag starts the service and sets it to launch on system boot.

To verify that the service is running, use the command:

sudo systemctl status ssh

The output should contain the line Active: active (running), indicating the service is running successfully.

To disable the service, execute:

sudo systemctl disable ssh

This disables the service and prevents it from starting at boot.

## **Step 4: Configure the Firewall**

Before connecting to the server via SSH, ensure the firewall is configured correctly.

If you are using UFW, check its status with:

sudo ufw status

If SSH traffic is not allowed, enable it with the command:

sudo ufw allow ssh

## **Step 5: Connect to the Server**

Once the previous steps are completed, you can log into the server using the SSH protocol.

You will need the server's IP address or domain name and the username created on the server. Use the command:

ssh username@IP\_address

Or:

ssh username@domain

**Important**: Ensure that SSH is installed and configured on both the remote server and the client machine.

## **Step 6: Configure SSH**

To enhance security, you can modify the default SSH configuration settings, such as changing the port number or using key-based authentication instead of passwords.

The main OpenSSH server settings are stored in the sshd\_config file located at /etc/ssh.

### **Create a Backup**

Before editing, back up the configuration file:

sudo cp /etc/ssh/sshd\_config /etc/ssh/sshd\_config.initial

If any errors occur after editing, you can restore the original file.

### **Edit the Configuration File**

Open the configuration file using the nano editor:

sudo nano /etc/ssh/sshd\_config

#### **Change the Port**

Change the default port (22) to a more secure one, preferably within the dynamic range (49152–65535). For example:

Port 49532

Uncomment the corresponding line and set the desired port number.

#### **Enable Key Authentication**

Enable key-based authentication by uncommenting the relevant line and setting the value to Yes:

PasswordAuthentication no

#### **Disable Root Login**

For added security, disable root login by setting the following value:

PermitRootLogin no

## **Additional Security Settings**

Here are other recommended configurations to improve server security:

* **UseDNS**: Ensures the hostname matches its IP address.
* **PermitEmptyPasswords**: Prohibits empty passwords if set to No.
* **MaxAuthTries**: Limits the number of failed login attempts.
* **AllowUsers/AllowGroups**: Specifies allowed users/groups for SSH access.
* **LoginGraceTime**: Reduces the allowed time for successful login attempts.
* **ClientAliveInterval**: Disconnects inactive users after the specified time.

Save and close the editor after making the necessary changes.

### **Restart the SSH Service**

Apply the changes by restarting the SSH service:

sudo systemctl restart ssh

If you changed the port number, connect using the new port:

ssh -p port\_number username@IP\_address

Or:

ssh -p port\_number username@domain

## **Conclusion**

This article provides a step-by-step guide on installing and configuring SSH on Ubuntu 22.04. It also outlines how to edit the SSH configuration file to enhance security.

Follow these instructions to set up a secure remote connection to your Ubuntu server.