PRIVATE OPENVPN SERVER ON A VPS: A STEP-BY-STEP GUIDE

By @Frank In Guides [April 12, 2023]

#OpenVPN #VPN #VPS #Private VPN #Installation Guide #VPN on a VPS #Personal VPN #Crypto VPN

In today's interconnected world, privacy and security are more important than eve
Virtual Private Networks (VPNs) provide a safe and private way to access the internet,

through the process of setting up your own VPN server using a fast installation script.

PREREQUISITES

Before we begin, ensure you have the following ready:

1. A virtual private server (VPS) or dedicated server running a Linux distribution, such as Ubuntu or CentOS.
2. Root access to your server.
3. A domain name (optional, but recommended for easier access).

STEP 1: INSTALL THE NECESSARY SOFTWARE

We will use the open-source OpenVPN software to create our VPN server. To simplify the installation process, we will use a script called "openvpn-install," which automates the process. Log in to your server via SSH and run the following commands:

```
wget https://git.io/vpn -O openvpn-install.sh
chmod +x openvpn-install.sh
```

STEP 2: RUN THE INSTALLATION SCRIPT AND CONFIGURE VPN SERVER

Execute the installation script by running the following command:

```
sudo ./openvpn-install.sh
```

The script will prompt you for various configuration options, such as the public IP address or domain name, the protocol (UDP or TCP), and the port number. If you have a domain name, use it for easier access; otherwise, use your server's public IP address.

For the protocol, we recommend using UDP, as it provides better performance. You can use the default port number (1194) or choose a custom one.

After the installation is complete, the script will generate a client configuration file (.ovpn) in the "/root" directory. You will need this file to connect to your VPN server using an OpenVPN client.

Download the .ovpn file to your local computer using an SCP client or any other file transfer method. For example, using the "scp" command:

```
scp root@your_server_ip:/root/client.ovpn /path/to/local/directory
```

Replace "your_server_ip" with your server's IP address and "/path/to/local/directory" with the local directory where you want to save the file.

STEP 4: CONNECT TO YOUR VPN SERVER

To connect to your VPN server, you will need an OpenVPN client on your device. OpenVPN clients are available for various platforms, including Windows, macOS, Linux, Android, and iOS. Install the appropriate client for your device and import the .ovpn file downloaded earlier.

Once the client is set up, establish a connection to your VPN server. If everything is configured correctly, your device will now be connected to your personal VPN server, and your online activities will be encrypted and secure. Creating your own VPN server gives you total control over your data and privacy, as well as the freedom to bypass geo-restrictions. Remember to keep your server updated and patched to maintain security and performance.

BACK TO BLOG

ARTICLE SUMMARY

Recent Posts

Crypto Wallets vs. Crypto Exchanges: How Are They Different?

Docker vs Kubernetes: Understanding the Best Use Cases for Each

Reaching the Summit: Why Alpine Linux Dominates Docker Landscapes

Special Offer: 50% Off Your First Month!

Why and How to Use RAM-Disk for Docker Containers on Ubuntu: A Comprehensive Guide

Popular Posts

Why and How to Use RAM-Disk for Docker Containers on Ubuntu: A Comprehensive Guide

Running Your Own Electrum Server: A Guide for Bitcoin and Litecoin Enthusiasts

CentOS vs. RHEL: Choosing the Best Linux Distro for Your Needs

Ultimate Linux Server Security Checklist: Hardening and Best Practices

How to Set Up a Private WireGuard VPN Server on a VPS

COIN HOST
THE CRYPTO HOSTING PROVIDER

Urgent
Bitcoin
Cryptocurrencies
Blockchain
Ethereum
ICO
Current Specials
CBDC
Crypto
VPN
Guides
DeFi
Affiliate program
Crypto hosting
Masternode
CMS Hosting
Docker
DevOps

Tags

Web Cardano Node Staking Smart Contracts Node Setup Cardano Setup Wallet Protection, XMR Wallet Cold wallet Docker vs Kubernetes Alpine Linux vs Ubuntu Docker Alpine Linux Compatibility Docker Image Optimization Secure Docker Base Image Lightweight Docker Images Docker Containers Alpine Alpine Linux Kubernetes Kubernetes Orchestration Hot wallet Crypto exchange Crypto wallet Docker hub K8s Kubernetes for Enterprise Docker in Development Containerization Best Practices Kubernetes Automation Ubuntu RAM-Disk Docker

SHOW MORE TAGS

COINHOST
THE CRYPTO HOSTING PROVIDER

COINHOST

Privacy-focused web hosting solutions for Bitcoin, Crypto and Blockchain enthusiasts.

BITCOIN HOSTING

Web Hosting Bitcoin

Dedicated Servers Bitcoin

VPS Bitcoin

Cloud Servers

DDoS Protection

Colocation Bitcoin

BITCOIN SERVICES

cPanel Bitcoin

CDN Bitcoin

WordPress Bitcoin

Drupal Bitcoin

Joomla Bitcoin

Object Storage Bitcoin

VPN Bitcoin

Affiliate Program Bitcoin

ABOUT

Blog

COIN.HOST
THE CRYPTO HOSTING PROVIDER

Speed Test

Knowledge Base

Terms of Service

Acceptable Use Policy

Privacy Policy

SLA

ACCOUNT & HELP

Sign In

Create Account

Contact

Proudly operated by Solar Communications GmbH. Twitter