

# alpha

## <ed/tech>

SSH  
File Transfer Protocol

<Módulo 07 />

# SFTP (SSH File Transfer Protocol)

## 1. Transferência de Arquivos Segura com SFTP

O **SFTP (SSH File Transfer Protocol)** é um protocolo de rede que permite a transferência, gerenciamento e manipulação de arquivos de forma segura através de um canal de dados criptografado.

Embora o nome lembre a ferramenta FTP, o SFTP é um protocolo completamente diferente, construído do zero como uma extensão do **SSH (Secure Shell)**. Enquanto o FTP tradicional utiliza duas conexões (uma para comandos e outra para dados), o SFTP realiza tudo através de uma única conexão segura na porta padrão **22**.

### Como ele funciona no Linux?

Em um ambiente Linux, o SFTP não exige a instalação de um software servidor dedicado (como o vsftpd ou ProFTPD). Ele geralmente utiliza o subsistema SFTP do **OpenSSH**, que já vem instalado por padrão na maioria das distribuições (Ubuntu, Debian, CentOS).

## 2. Vantagens de usar o SFTP

A escolha pelo SFTP em vez do FTP convencional ou do FTPS (FTP sobre SSL) traz benefícios críticos para a infraestrutura de TI:

### A. Segurança e Criptografia Total

Diferente do FTP, que envia usuários e senhas em **texto puro** (permitindo que qualquer pessoa na rede "intercepte" as credenciais), o SFTP criptografa tanto a autenticação quanto os dados transferidos. Isso protege o servidor contra ataques de *sniffing*.

### B. Simplicidade de Firewall (Single Port)

O FTP clássico é conhecido como "inimigo dos firewalls" por abrir múltiplas portas aleatórias para transferência de dados.

- **FTP:** Precisa das portas 21 e de um intervalo enorme de portas passivas.
- **SFTP:** Utiliza **apenas uma porta (22)** para tudo. Isso torna a configuração de regras de segurança muito mais simples e limpa.

### C. Autenticação por Chaves SSH

Além da senha comum, o SFTP permite o uso de **Chaves Públicas/Privadas**. Isso significa que você pode configurar transferências automatizadas entre servidores sem precisar digitar senhas, usando um método de autenticação muito mais robusto e difícil de quebrar por força bruta.

### D. Integridade dos Dados

O SFTP possui mecanismos nativos para verificar se o arquivo chegou ao destino exatamente como saiu da origem. Ele utiliza verificações de integridade (como MAC - Message Authentication Code) para garantir que os pacotes não foram alterados durante o trajeto.

### E. Manipulação Remota de Arquivos

O SFTP não serve apenas para "enviar e receber". Ele permite que o usuário realize operações complexas no servidor remoto, como:

- Alterar permissões de arquivos (chmod).

- Criar e remover diretórios.
- Retomar transferências interrompidas (resume).

### 3. Resumo Comparativo

Característica	FTP Tradicional	SFTP (SSH)
<b>Criptografia</b>	Nenhuma	Forte (SSH)
<b>Portas no Firewall</b>	Múltiplas (21 + aleatórias)	Única (22)
<b>Credenciais</b>	Enviadas em texto plano	Criptografadas
<b>Facilidade de Uso</b>	Requer servidor extra	Nativo no Linux (OpenSSH)

**Dica de Estudo:** Para testar o SFTP no seu terminal Ubuntu, tente o comando:

```
sftp usuario@ip-do-servidor
```

### Passo 1: Configurar o Servidor SFTP no Ubuntu

O SFTP utiliza o protocolo SSH, então o primeiro passo é garantir que o serviço de SSH esteja instalado e rodando.

#### 1. Instale o OpenSSH Server:

No terminal do Ubuntu, execute:

```
sudo apt update
sudo apt install openssh-server -y
```

#### 2. Verifique se está ativo:

```
sudo systemctl status ssh
```

Se estiver "active (running)", está tudo certo. 3. **(Opcional) Criar um usuário apenas para SFTP:** Se você quiser que o usuário **não** tenha acesso ao terminal (shell), apenas para enviar arquivos:

```
sudo adduser sftpuser --shell /bin/false
```

### Passo 2: Configurar a Rede no VirtualBox

Por padrão, a VM usa o modo **NAT**, o que impede conexões externas. Você tem duas opções:

#### Opção A: Redirecionamento de Portas (Mais fácil se usar NAT)

1. Com a VM desligada, vá em **Configurações > Rede > Avançado**.

2. Clique em **Redirecionamento de Portas**.
3. Adicione uma nova regra:
  - **Nome:** SFTP
  - **Protocolo:** TCP
  - **Porta do Hospedeiro:** 2222
  - **Porta do Convidado:** 22
4. Agora, para conectar, você usará o endereço 127.0.0.1 na porta 2222.

### Opção B: Placa em Modo Bridge (Cria um IP próprio na sua rede)

1. Vá em **Configurações > Rede**.
2. Mude "Conectado a:" para **Placa em modo Bridge**.
3. A VM passará a ter um IP na sua rede real (ex: 192.168.1.15). Você pode descobrir o IP usando o comando `ip a` no Ubuntu.

## Passo 3: Testar a Conexão

Agora, abra um cliente como o **FileZilla** ou o **WinSCP** no seu Windows/Mac (Hospedeiro) e preencha:

- **Protocolo:** SFTP
- **Host:** 127.0.0.1 (se usou Redirecionamento) ou o IP da VM (se usou Bridge).
- **Porta:** 2222 (se usou Redirecionamento) ou 22 (se usou Bridge).
- **Usuário e Senha:** Seus dados do Ubuntu.

### Dica de Segurança: Chroot

Se você quiser que o usuário fique "preso" apenas em uma pasta específica e não veja o resto do sistema, você precisará editar o arquivo

```
/etc/ssh/sshd_config
```

e adicionar as diretivas `Match User` e `ChrootDirectory`.

- Deseja que eu te mostre como configurar esse bloqueio de pasta (Chroot) para o usuário?

[Criando servidor SFTP no Linux em menos de 10 minutos](#)

Este vídeo demonstra de forma prática como realizar a configuração rápida do servidor SFTP no Linux, complementando os passos técnicos acima.

## Túnel SSH

Para explicar como o SSH protege transferências de arquivos, precisamos entender que o comportamento muda completamente dependendo do protocolo escolhido.

No **SFTP**, o túnel já é o próprio protocolo. No **FTP**, o túnel SSH precisa ser construído manualmente para “esconder” o tráfego inseguro dentro dele.

### 1. O Túnel no SFTP (Nativo e Automático)

No SFTP, você não precisa criar um túnel separado; o protocolo **já roda nativamente dentro de um túnel SSH**.

- **Como funciona:** Quando você inicia uma conexão SFTP, o cliente SSH estabelece uma conexão segura (criptografada) com o servidor. Dentro desse canal seguro, o subsistema SFTP é iniciado.
- **Segurança:** Tudo o que passa por ali — seu usuário, sua senha e os arquivos — viaja dentro de uma “armadura” de criptografia.
- **Cenário:** É como se você estivesse enviando uma carta dentro de um carro blindado. O carro é o SSH, e a carta é o seu arquivo.

### 2. O Túnel SSH com FTP (Encapsulamento Manual)

O FTP tradicional é inseguro. Para protegê-lo, usamos uma técnica chamada **SSH Port Forwarding** (Redirecionamento de Portas). Aqui, criamos um túnel SSH para que o tráfego do FTP passe por dentro dele.

#### Como é utilizado na prática:

1. **Criação do Túnel:** Você executa um comando no seu terminal para ligar uma porta do seu computador à porta do servidor através do SSH: `ssh -L 2121:localhost:21 usuario@servidor`
2. **O “Truque”:** O seu computador passa a acreditar que o servidor FTP está rodando dentro dele mesmo (no localhost).
3. **A Conexão:** Você abre seu programa de FTP (como o FileZilla) e manda ele conectar em 127.0.0.1 na porta 2121.
4. **O Trajeto:** Os dados saem do programa de FTP, entram no túnel SSH no seu PC, viajam criptografados pela internet e saem “desembrulhados” direto no servidor FTP.

### Diferenças de Uso do Túnel

Característica	Túnel no SFTP	Túnel no FTP
<b>Configuração</b>	Automática (padrão).	Manual (via comando SSH).

Característica

**Complexidade**

**Eficiência**

**Visibilidade**

Túnel no SFTP

Baixa: Só precisa do login.

Alta: Otimizado para SSH.

O usuário não vê o túnel.

Túnel no FTP

Alta: Exige lidar com portas e firewall.

Média: O FTP ainda tenta abrir outras portas.

O usuário precisa criar o túnel antes.

### Por que o SFTP é preferido hoje?

O FTP tem uma dificuldade técnica com túneis SSH: ele usa **duas portas** (uma para comandos e outra para os dados). Criar um túnel para o canal de dados do FTP é complexo e muitas vezes falha em redes com NAT.

O **SFTP resolve isso usando apenas uma porta (22)**, fazendo com que o túnel SSH seja estável, simples e extremamente seguro por padrão.