

Linux Server

- [SSH verbinden unter Windows](#)
- [Grundlegende Linux-Befehle: Ein einfacher Guide](#)
- [Tmux: Terminal Multiplexer](#)
- [Steamcmd Linux](#)
- [Java installieren auf Debian-basierende Linux](#)

SSH verbinden unter Windows

Was ist SSH?

SSH (Secure Shell) ist ein Netzwerkprotokoll, das eine sichere Verbindung zwischen einem Client und einem Server über ein unsicheres Netzwerk ermöglicht. Es wird häufig verwendet, um sich sicher auf entfernte Maschinen einzuloggen und Remote-Befehle auszuführen.

Voraussetzungen

1. Zugangsdaten für den SSH-Server (IP-Adresse oder Hostname, Benutzername, Passwort oder SSH-Schlüssel).
2. [Windows Terminal](#) aus dem Microsoft Store installiert oder andere Programme die SSH unterstützen

Verbindung zu einem SSH-Server herstellen

1.Windows Terminal öffnen

Öffne das Startmenü, suche nach "**Windows Terminal**", und starte die Anwendung

2.SSH-Befehl eingeben

Um dich mit einem SSH-Server zu verbinden, gib folgenden Befehl ein:

```
ssh username@hostname
```

3.Passwort eingeben

Du wirst zur Eingabe deines Passworts aufgefordert. Gib es ein und drücke Enter. (Das Passwort wird nicht sichtbar sein, während du es eingibst, aus Sicherheitsgründen.)

4.SSH-Schlüssel verwenden (optional):

Wenn du SSH-Schlüssel zur Authentifizierung verwendest, stelle sicher, dass dein Schlüssel im Standardverzeichnis (`~/.ssh/id_rsa` für private Schlüssel) vorhanden ist, oder gib den Pfad zum Schlüssel mit dem `-i` Parameter an:

```
ssh -i /path/to/private/key username@hostname
```

Beispiel:

Angenommen, du möchtest dich als Benutzer `john` mit dem Server `example.com` verbinden:

```
ssh john@example.com
```

Wenn du einen SSH-Schlüssel mit dem Pfad `C:\Users\John\.ssh\id_rsa` verwendest:

```
ssh -i C:\Users\John\.ssh\id_rsa john@example.com
```

Verbindungsoptionen und Tipps

Port ändern:

Standardmäßig verwendet SSH Port 22. Um einen anderen Port zu verwenden, füge `-p portnummer` hinzu:

```
ssh john@example.com -p 2222
```

Verbose Mode:

Um mehr Details zur Fehlersuche zu sehen, benutze den `-v` Schalter

```
ssh -v john@example.com
```

Abschluss

Mit diesen Schritten kannst du eine sichere SSH-Verbindung zu einem entfernten Server herstellen und diesen verwalten. Windows Terminal bietet eine flexible und leistungsstarke Umgebung für diese und viele weitere Aufgaben.

Probiere es aus und genieße die einfache und sichere Verbindung zu deinen Remote-Servern!

Grundlegende Linux-Befehle: Ein einfacher Guide

Anmerkung

Diese Befehle können direkt auf einem Linux-System oder über eine SSH-Verbindung genutzt werden. Um zu erfahren, wie du eine SSH-Verbindung unter Windows einrichtest, sieh dir diesen Guide [SSH verbinden unter Windows](#) an.

1. `ls` - Verzeichnisinhalt anzeigen

Zeigt die Dateien und Ordner im aktuellen Verzeichnis an.

```
ls
```

Optionen:

- `ls -l`: Detaillierte Ansicht mit Berechtigungen, Eigentümer, Größe und Änderungsdatum.
- `ls -a`: Zeigt auch versteckte Dateien an.

2. `cd` - Verzeichnis wechseln

Wechselt in ein anderes Verzeichnis.

```
cd /pfad/zu/einem/verzeichnis
```

Optionen:

- `cd ..`: Wechselt in das übergeordnete Verzeichnis.
- `cd ~`: Wechselt in das Heimatverzeichnis des Benutzers.

3. `nano` - Texteditor

Öffnet den Nano-Texteditor, um Textdateien zu bearbeiten.

```
nano dateiname
```

Befehle im Nano-Editor:

- `Ctrl + O`: Datei speichern.
- `Ctrl + X`: Nano beenden.

4. `mkdir` - Verzeichnis erstellen

Erstellt ein neues Verzeichnis.

```
mkdir verzeichnisname
```

5. `rmdir` - Verzeichnis löschen

Löscht ein leeres Verzeichnis.

```
rmdir verzeichnisname
```

6. `rm` - Datei oder Verzeichnis löschen

Löscht Dateien oder Verzeichnisse.

```
rm dateiname
```

Optionen:

- `rm -r verzeichnisname`: Löscht ein Verzeichnis und dessen Inhalt rekursiv.

7. `cp` - Dateien kopieren

Kopiert Dateien oder Verzeichnisse.

```
cp quelle ziel
```

Optionen:

- `cp -r quelle ziel`: Kopiert ein Verzeichnis und dessen Inhalt rekursiv.

8. `mv` - Dateien verschieben oder umbenennen

Verschiebt oder benennt Dateien oder Verzeichnisse um.

```
mv quelle ziel
```

9. `cat` - Dateien anzeigen

Zeigt den Inhalt einer Datei an.

```
cat dateiname
```

10. `touch` - Datei erstellen

Erstellt eine neue, leere Datei oder aktualisiert den Zeitstempel einer existierenden Datei.

```
touch dateiname.dateityp
```

11. `chmod` - Dateiberechtigungen ändern

Ändert die Berechtigungen einer Datei oder eines Verzeichnisses.

```
chmod berechtigungen dateiname
```

Beispiel:

- `chmod +x dateiname`: Macht eine Datei ausführbar.

12. `chown` - Dateibesitzer ändern

Ändert den Besitzer einer Datei oder eines Verzeichnisses.

```
chown benutzer:gruppe dateiname
```

13. `sudo` - Befehle als Superuser ausführen

Führt einen Befehl mit Superuser-Rechten aus.

```
sudo befehl
```

14. `apt` - Paketverwaltung (für Debian-basierte Systeme wie Ubuntu)

Verwaltet Softwarepakete.

```
sudo apt update          # Paketlisten aktualisieren
sudo apt upgrade         # Installierte Pakete aktualisieren
sudo apt install paketname # Paket installieren
sudo apt remove paketname # Paket entfernen
```

Tmux: Terminal Multiplexer

Wofür ist Tmux?

Tmux ist ein Terminal-Multiplexer, der es dir ermöglicht, mehrere Terminal-Sitzungen innerhalb eines einzigen Fensters zu erstellen, zu verwalten und zwischen ihnen zu wechseln. Dies ist besonders nützlich, wenn du Prozesse im Hintergrund laufen lassen möchtest, ohne das Terminal die ganze Zeit offen zu halten.

Installation

Um Tmux zu installieren, führe auf einem Debian-basierten System (wie Ubuntu) folgenden Befehl aus:

```
sudo apt install tmux
```

Grundlegende Tmux-Befehle

1. Neue Tmux-Session starten:

```
tmux
```

2. Neue Tmux-Session mit Namen starten:

```
tmux new -s session_name
```

3. Von einer Tmux-Session trennen:

Drücke `Ctrl + B` und dann `D`.

4. Liste der laufenden Tmux-Sessions anzeigen:

```
tmux ls
```

5. Mit einer bestehenden Tmux-Session verbinden:

```
tmux attach -t session_name
```

Mit diesen Befehlen kannst du Tmux effektiv nutzen, um deine Terminal-Sitzungen zu verwalten und deine Arbeit effizienter zu gestalten.

Steamcmd Linux

Steamcmd installieren unter Ubuntu

1. Füge das Multiverse-Repository hinzu und aktualisiere die Paketliste:

```
sudo add-apt-repository multiverse  
sudo dpkg --add-architecture i386  
sudo apt update
```

2. Installiere SteamCMD:

```
sudo apt install steamcmd
```

Steamcmd Starten

Um SteamCMD zu starten, gib folgenden Befehl im Terminal ein:

```
steamcmd
```

Nun befindest du dich in SteamCMD und kannst Game Server installieren.

Mit diesen Schritten kannst du SteamCMD erfolgreich auf Ubuntu installieren und starten.

Java installieren auf Debian-basierende Linux

Die Installation mit einem Paketmanager erfordert Root-Berechtigungen. Melden Sie sich als Root an oder verwenden Sie sudo beim Ausführen der folgenden Befehle.

1. Importieren Sie den öffentlichen Schlüssel von Azul:

```
sudo apt install gnupg ca-certificates curl
curl -s https://repos.azul.com/azul-repo.key | sudo gpg --dearmor -o
/usr/share/keyrings/azul.gpg
echo "deb [signed-by=/usr/share/keyrings/azul.gpg] https://repos.azul.com/zulu/deb stable
main" | sudo tee /etc/apt/sources.list.d/zulu.list
```

2. Aktualisieren Sie die Informationen über verfügbare Pakete:

```
sudo apt update
```

3. Installieren Sie das erforderliche Azul Zulu-Paket:

```
sudo apt install zulu<jdk-version>-jdk

# For example
sudo apt install zulu17-jdk
```

4. Um Ihre Installation zu überprüfen, führen Sie den Befehl `java -version` aus. Sie sollten eine Ausgabe sehen, die ähnlich der folgenden ist:

```
$ java -version
openjdk version "17.0.6" 2023-01-17 LTS
OpenJDK Runtime Environment Zulu17.40+19-CA (build 17.0.6+10-LTS)
OpenJDK 64-Bit Server VM Zulu17.40+19-CA (build 17.0.6+10-LTS, mixed mode, sharing)
```