

# Tutorial zu Installation eines Minecraft-Servers auf einem Raspberry Pi 3

Das folgende Tutorial richtet sich an Kinder und ihre Eltern, die mit wenig Vorwissen einen heimischen Minecraft-Server aufsetzen möchten und auf diesem sogar Mods<sup>1)</sup> betreiben möchten.

## Vorbereitung

Damit der Server jeder Zeit zur Verfügung steht, aber nicht eine Hochleistungsrechner laufen soll, kann man dies mit einem Raspberry Pi 3 realisieren. Dieser ist verhältnismäßig leistungsfähig, benötigt aber nur minimale Ressourcen.

Folgende Dinge werden benötigt bzw. damit geht es am schnellsten.

- Raspberry Pi 3 <sup>2)</sup>
- USB-Tastatur
- HDMI-Kabel
- SD-Card (mind. 8GB), LAN-Kabel
- raspbian als Linux-Betriebssystem [DOWNLOAD](#)
- Java als Umgebung für minecraft (ist in raspbian bereits enthalten)
- minecraft-Server als eigentliche Server-Software (wird später direkt auf den Raspberry Pi heruntergeladen)
- forge-Server, wenn Mods gewünscht sind (wird ebenfalls später heruntergeladen)

Den Raspberry Pi 3 gibt es bei einschlägigen Händler auch als Starter-Bundels für ca. 70-80 €. Diese Pakete enthalten bis auf das HDMI-Kabel und die USB-Tastatur alles. Beides wird nur für die Erstinbetriebnahme benötigt, so dass man diese auch gut von einem bestehenden PC „ausborgen“ kann. Später erfolgt der Zugriff über die Ferne. Aber dazu später mehr.

## Installation des Raspberry Pi

Zunächst muss der Raspberry selbst in Betrieb genommen werden. Hierzu lädt man sich von den Raspbian-Machern die neueste Version des [Linux für Raspberry Pi](#) herunter. Hier lädt man die Version **RASPBIAN JESSIE LITE** herunter, also ohne Oberfläche. Der Server kann ohnehin nur über die Kommandozeile gestartet werden, da ansonsten nicht genügend Arbeitsspeicher zur Verfügung steht. Aber dazu später mehr.

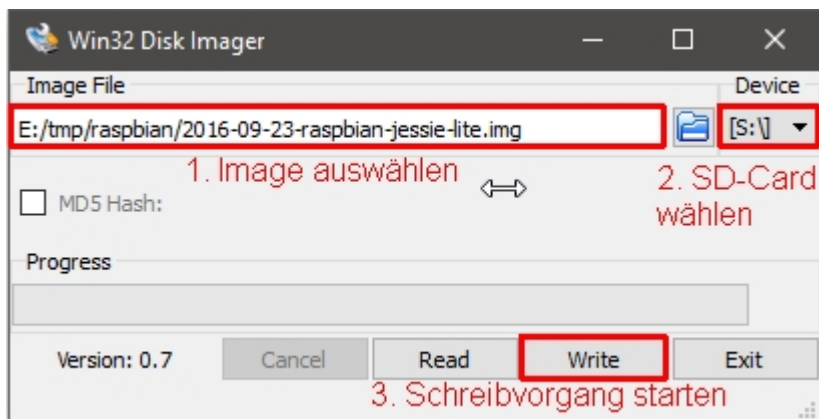
Hinweis: Alternativ kann auch Noobs verwendet werden, allerdings wollen wir uns auf das Minecraft-Server-Projekt konzentrieren.

### Raspbian OS auf SD-Karte bringen

Nun muss das Image auf eine micro-SD-Karte gebracht werden. Dies kann mit der Software [Win32diskimager](#) durchgeführt werden. ACHTUNG: Bei der Wahl des Laufwerks sollte auf keinen Fall

ein weiterer USB-Stick im PC stecken, damit dieser nicht aus Versehen gelöscht wird.

Das folgende Bild zeigt die drei Schritte zum Beschreiben der SD-Card mit Raspbian.



## Raspberry Pi Grundeinrichtung

Nun kann das frisch installierte Betriebssystem gestartet werden. Wir belassen zu nächst die Grundeinstellungen. Sobald die Kommandozeile (**shell** genannt) erscheint, können wir uns mit folgenden Daten anmelden:

- Benutzername: pi
- Passwort: pi

Damit wir den Raspberry nicht dauerhaft mit einer eigenen Tastatur und einem Monitor betreiben müssen. Richten wir uns nun den Fernzugriff mittels ssh.<sup>3)</sup> In diesem Zusammenhang richten wir auch noch weitere Details ein.

Vorher müssen wir uns allerdings noch die IP-Adresse des Raspberry notieren, damit wir ihn später per ssh über das Netzwerk erreichen können.

```
ip -4 addr show eth0
```

Wir erhalten folgende Ausgabe:

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    inet 192.168.0.16/24 brd 192.168.0.255 scope global eth0
        valid_lft forever preferred_lft forever
```

Uns interessiert hier nur der Wert hinter inet. Dies ist die sogenannte IPv4-Adresse. Im Beispiel lautet diese 192.168.0.16. Der Wert /24 dahinter ist die Subnetz-Maske, die für uns hier nicht von Interesse ist.

Die IP-Adresse notieren wir uns für später.

Nun geht es zur eigentlichen Grundkonfiguration.

Hierzu starten wir das Raspberry Pi Konfigurationssystem `raspi-config` mit folgendem Befehl:





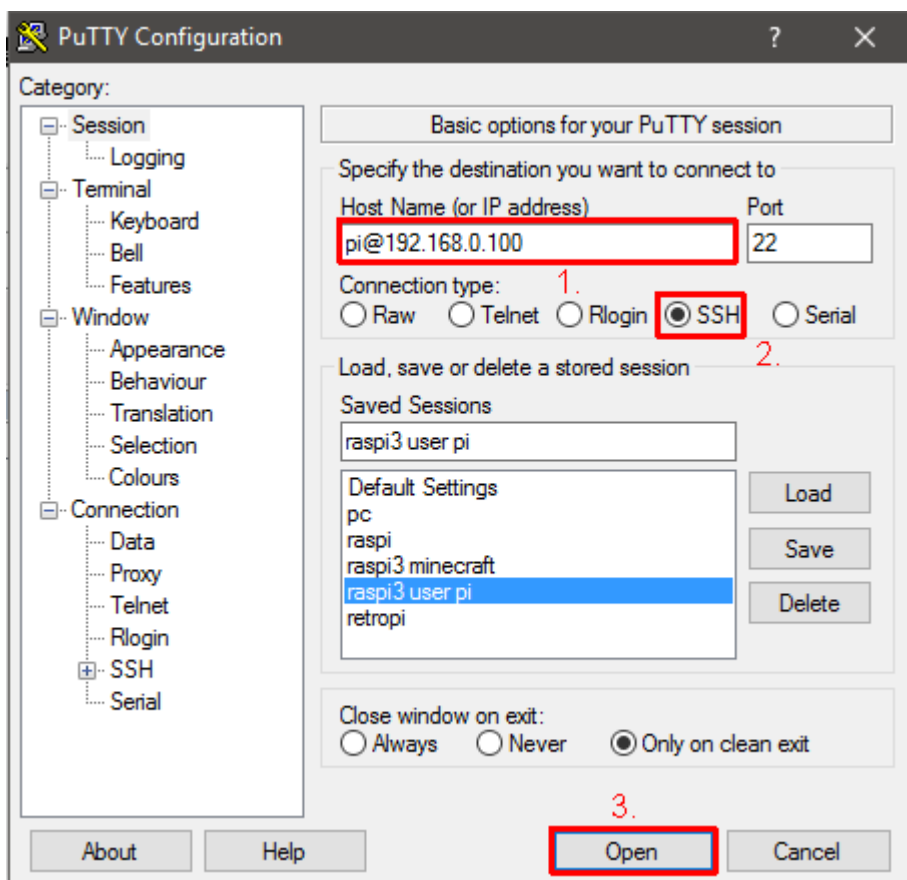
## Fernzugriff auf Raspberry

Durch den oben aktivierten ssh-Server sind wir nun in der Lage von einem anderen Rechner aus auf den Raspberry zu zugreifen. Hierzu werden für die unterschiedlichen Betriebssysteme eigene Tools benötigt:

- ssh-Kommandozeilenbefehl (Linux, MacOS)
- putty-Programm (Windows) [DOWNLOAD](#)

Für Windows würde putty.exe ausreichen. Da wir aber auch Dateien auf den Raspberry mittels pscp.exe kopieren möchten sollte das "komplett Paket" heruntergeladen werden.

Der Dialog von putty sieht wie folgt aus.



Als erstes geben wir den Nutzer und die IP-Adresse ein. Hier: pi@192.168.0.16 Die IP-Adresse haben wir uns weiter oben ja bereits notiert. Nun können wir diesen Werten einen klingenden Namen geben, so dass wir die Konfiguration später wieder finden. z.B. raspi user pi Anschließend speichern wird die Konfiguration mit Save.

Es öffnet sich nach einem Klick auf Open ein schwarzes Fenster, in dem die shell des Raspberry angezeigt wird. Jeder Befehl, der dort eingegeben wird, wird auf dem Raspberry ausgeführt. Also Vorsicht mit sudo-Befehlen!

Wir sollten uns zur Sicherheit einen eigenen Benutzer für unseren minecraft-Server einrichten. Dies geschieht mit folgendem Befehl:

```
sudo adduser mc
```

Man wird nun nach einem Passwort für den neuen Benutzer gefragt. Dieses sollte möglichst ein anderes sein, als das des Hauptnutzers pi. Das Passwort muss nochmals wiederholt werden. Die anschließenden Fragen können einfach mit Enter übersprungen werden. Am Ende wird gefragt, ob alles korrekt ist. Hinweis: In der englischen Version muss mit y geantwortet werden. Ist die Tastatur nicht umgestellt worden, so muss man hier die z-Taste drücken.

Nun melden wir uns mit exit ab und starten eine neue ssh-Verbindung mit putty. Damit wir nicht ständig alle Werte in putty neu eingeben müssen, speichern wir mit Save die IP-Adresse wie folgt:

```
mc@192.168.0.16
```

Die neue Konfiguration nutzen wir, um uns erneut, nun als Benutzer mc, mit dem Raspberry zu verbinden. Wir werden aufgefordert unser Passwort einzugeben. Hier gibt man das Passwort, dass wir bei adduser mc verwendet haben ein.

Wenn alles richtig war können wir nun mittels `pwd`<sup>7)</sup> überprüfen, ob wir im richtigen Pfad des Dateiverzeichnisses ist.

Befehl: `pwd`

Ausgabe: `/home/mc`

Der Server besteht später aus vielen Dateien und Unterordnern. Damit wir unseren Server nicht im ganzen sogenannten Home-Ordner verteilen, erstellen wir zunächst einen eigenen Ordner für den Forge-Server. Den Namen wählen wir entsprechend der Version die wir verwenden wollen. Hier: 1.7.10

```
mkdir forge_1.7.10
```

Mit

```
cd forge_1.7.10
```

wechseln wir in diesen neuen Ordner.

TIPP: Die Tabulator-Taste ist unser bester Freund unter Linux. Sie ergänzt Dateinamen und Ordnernamen automatisch. Im Zweifelsfall einfach zweimal kurz hintereinander tippen und man erhält mögliche Eingaben als Vorschau.

Nun sind wir bereit den eigentlichen Server zu installieren.

## Herunterladen des Forge-Servers

### Weg 1: Kopieren des Forge-Servers von einem Windows-PC

Hierzu benötigen wir `pscp.exe` (s. [Fernzugriff auf Raspberry](#)) und den `forge-installer` ([Download-Seite](#)) **ACHTUNG:** Auch wenn dieser Weg etwas umständlich erscheint, so sollte er immer

zu erst ausprobiert werden. Die Entwickler sind so in der Lage durch Werbung einen Teil ihrer Kosten und Aufwände entschädigt zu bekommen!

Folgendes wird für das Kopierskript vorausgesetzt:

Variable	Bedeutung	Beispiel
mod_file	Pfad zur heruntergeladener forge-installer-Datei	„D:\downloads\forge-1.7.10-10.13.4.1558-1.7.10-installer.jar
mc_raspi	Pfad zu den Mods auf dem Raspberry-Server	“/home/mc/forge_1.7.10/„
raspi_ip	IP-Adresse des Raspberry	192.168.0.16
mc_user		

Die Variablen sind ggf. anzupassen, wenn man andere Pfade verwendet. Das folgende Skript kann per Klick auf `copyforge2raspi.bat` heruntergeladen werden.

[copyforge2raspi.bat](#)

```
@echo off
set mod_file="D:\downloads\forge-1.7.10-10.13.4.1558-1.7.10-
installer.jar"

set mc_raspi="/home/mc/"
set raspi_ip="192.168.0.16"
set mc_user="mc"

d:\apps\putty\pscp.exe %mod_file% %mc_user%@%raspi_ip%:%mc_raspi%
```

Um es auszuführen, wird unter Windows ein Kommandozeile (Eingabeaufforderung) benötigt. Diese lässt sich durch Win+R und anschließende Eingabe von cmd bei Öffnen: starten. Nun muss das Skript starten. Dies geschieht über `D:\Downloads\copyforge2raspi.bat`, wobei `D:\Downloads` der Ordner ist, in dem `copyforge2raspi.bat` heruntergeladen wurde. Ggf. ist dieser erste also anzupassen.

## Weg 2: Herunterladen des Forge-Servers direkt auf den Raspberry

Hierzu müssen wir zunächst eine ssh-Verbindung mit dem Raspberry aufbauen (s. [Fernzugriff auf Raspberry](#) )

Mit dem folgenden Befehl wird der Forge-Server Installer auf den Raspberry in den aktuellen Ordner heruntergeladen:

```
wget
http://files.minecraftforge.net/maven/net/minecraftforge/forge/1.7.10-10.13.4.1558-1.7.10/forge-1.7.10-10.13.4.1558-1.7.10-installer.jar .
```

**ACHTUNG:** Dieser Weg sollte aus Fairness den Entwickler gegenüber NICHT benutzt werden, da

diese auf diesem Weg keine Einnahmen durch Werbung generieren können. In der Entwicklung genauso wie in dieser Seite steckt viel Zeit und Arbeit, die honoriert werden sollte. Also bitte möglichst Weg 1 nutzen.

# Installation des Forge-Servers auf dem Raspberry

Anschließend muss der heruntergeladene `forge-installer` noch ausgeführt werden. Auf einer grafischen Oberfläche (z.B. LXDesktop) kann man ihn einfach starten. Auf der Kommandozeile, also per `ssh`, geht dies leider nicht. Der folgende Befehl führt den Installer daher OHNE grafischen Dialog aus und lädt die benötigten Dateien aus dem Internet direkt in den Ordner nach, in dem der Installer liegt.

```
cd forge_1.7.10
java -jar forge-1.7.10-10.13.4.1558-1.7.10-installer.jar --installServer
```

Nachdem alles heruntergeladen wurde, kann der Installer gefahrlos gelöscht werden.

```
rm forge-1.7.10-10.13.4.1558-1.7.10-installer.jar
```

## Erster Start des Forge-Servers

### Zusammenfassung:

- `java -Xms512M -Xmx1008M -jar /home/mc/forge_1.7.10/forge-1.7.10-10.13.4.1558-1.7.10-universal.jar` → Aufruf des Minecraft-Servers

**Ausführlich:** Der Start erfolgt mit folgendem Befehl:

```
java -Xms512M -Xmx1008M -jar /home/mc/forge_1.7.10/forge-1.7.10-10.13.4.1558-1.7.10-universal.jar nogui
```

TIPP: Da die Befehlszeile sehr lang ist, kann man dies auch in ein kleines `shell`-Skript packen und diesem einen kurzen Namen geben.

Eine solche Datei kann man mit dem folgenden Befehl anlegen:

```
nano start_minecraft.sh
```

In der Datei muss folgendes stehen, damit es funktioniert.

[start\\_minecraft.sh](#)

```
#!/bin/bash
```

```
cd /home/mc/forge_1.7.10
java -jar -Xms512M -Xmx1008M forge-1.7.10-10.13.4.1558-1.7.10-
universal.jar nogui && break
```

Damit die Datei als Starter benutzt werden kann, muss sie ausführbar gemacht werden. Dieser Schritt muss nur einmal gemacht werden.

```
chmod +x start_minecraft.sh
```

Anschließend kann der Server immer wieder mit dem folgenden Befehl gestartet werden.

```
./start_minecraft.sh
```

Der erste Start schlägt auf jeden Fall fehl, da die EULA noch nicht akzeptiert wurde. Um die EULA zu akzeptieren, öffnet man die Datei `eula.txt` und ändert `false` in `true` um.

```
nano eula.txt
```

Mit STRG+O kann man die Datei speichern und mit STRG+X kann der Editor geschlossen werden.

Nun kann der Server erneut gestartet werden. Der folgende Start dauert relativ lange, da der Server nun alle Elemente erstmalig einrichten muss. Weitere Starts gehen etwas schneller von statten.

**ACHTUNG:** Der Server wird bei dieser Methode automatisch beendet, so bald die ssh-Verbindung geschlossen wird. Im nächsten Abschnitt wird daher erläutert, wie man den Minecraft-Server aus ohne ssh-Verbindung betreiben kann.

## Minecraft-Server OHNE aktive ssh-Verbindung betreiben

Nachteil der oben beschriebenen Lösung ist es, dass der Server geschlossen wird, sobald die ssh-Verbindung zum Raspberry getrennt wird. Möchte man den Server dauerhaft starten und am besten automatisch neustarten, falls er mal abgestürzt ist, dann kann man dazu folgendes Script erstellen.

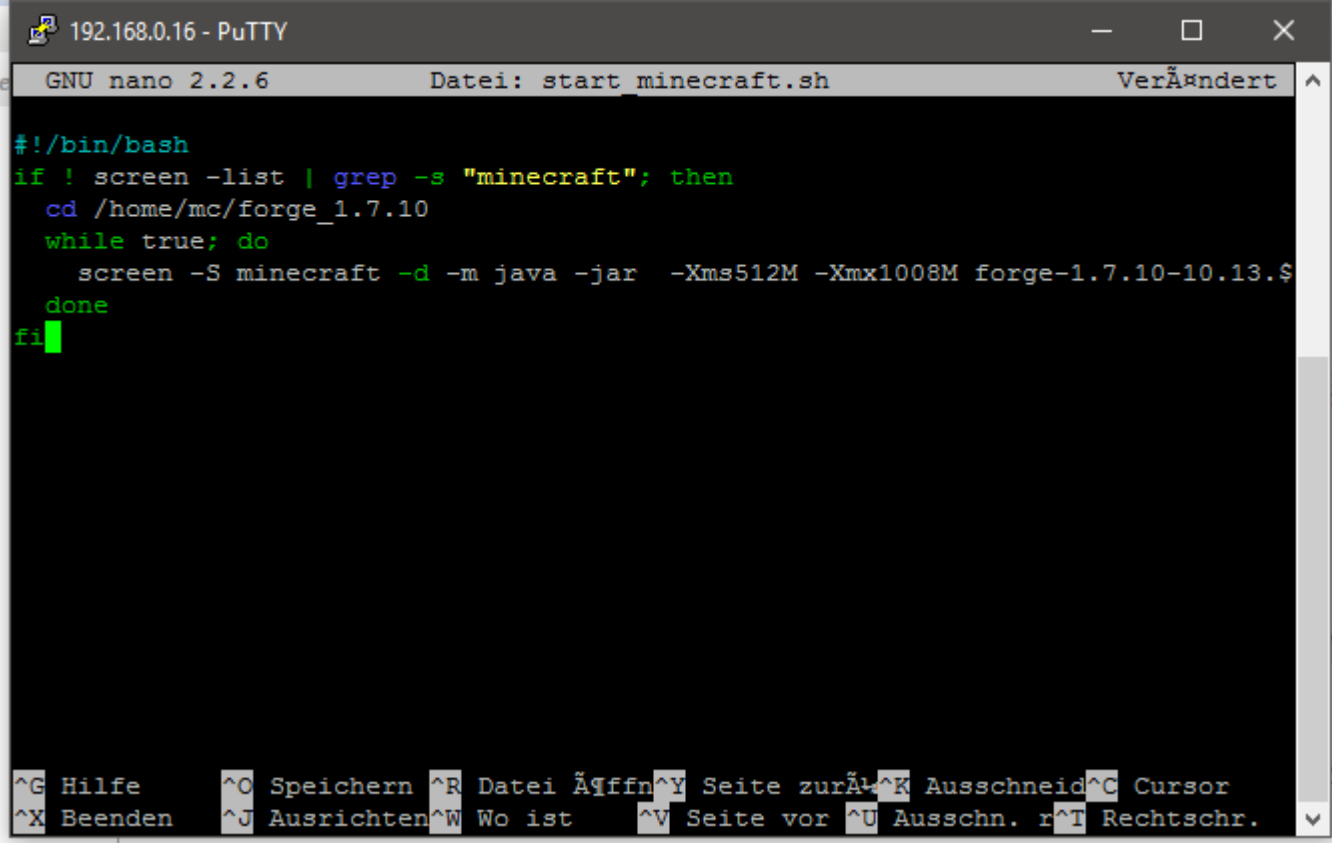
[start\\_minecraft.sh](#)

```
#!/bin/bash
if ! screen -list | grep -s "minecraft"; then
  cd /home/mc/forge_1.7.10
  while true; do
    screen -S minecraft -d -m java -jar -Xms512M -Xmx1008M
forge-1.7.10-10.13.4.1558-1.7.10-universal.jar nogui && break
  done
fi
```

(Das Skript stammt im Original von <http://lemire.me/blog/2016/04/02/setting-up-a-robust-minecraft-server-on-a-raspberry-pi/> und wurde leicht modifiziert)

Um das Skript auf den Raspberry zu bringen kann man Copy&Paste verwenden. Einfach den Inhalt des obigen grauen Kastens markieren und mit STRG+C in die Zwischenablage kopieren. Nun kann man im ssh-Fenster den Editor nano `start_minecraft.sh` öffnen und eine neue Datei `start_minecraft.sh` erzeugen. Im erscheinenden Fenster kann man nun per Klick auf die rechte-Maustaste den Inhalt der Zwischenablage hineinkopieren.

Das Ergebnis sollte so aussehen:



```
GNU nano 2.2.6 Datei: start_minecraft.sh Verändert
#!/bin/bash
if ! screen -list | grep -s "minecraft"; then
cd /home/mc/forged_1.7.10
while true; do
screen -S minecraft -d -m java -jar -Xms512M -Xmx1008M forge-1.7.10-10.13.$
done
fi
```

Nun kann man die Datei mit STRG+O<sup>B)</sup> speichern und mit STRG+X den Editor wieder schließen. Als letzten Schritt muss nun noch folgende Befehle ausgeführt werden, damit das Skript ausführbar wird (`chmod +x start_minecraft.sh`) und gestartet wird (`./start_minecraft &`).

```
chmod +x start_minecraft.sh
./start_minecraft.sh &
```

### Das Skript ausführlich erklärt:

Innerhalb der `if`-Anweisung wird abgefragt, ob bereits ein Minecraft-Server läuft. Ist dies nicht der Fall (s. ! ) dann wird in den Installationsordner (hier: `/home/mc/forged_1.7.10`) gewechselt und in der `screen`-Zeile ein neuer Server gestartet, der unabhängig von der ssh-Verbindung läuft.

Ggf. muss der `screen`-Befehl noch nachinstalliert werden. Dies geschieht mit folgendem Befehl: (Hinweis: Nur der Benutzer `pi` kann den `sudo`-Befehl ausführen.)

```
su pi
```

```
sudo apt-get install -y screen
```

## Status des gestarteten Minecraft-Servers überprüfen

### Zusammenfassung:

- `screen -ls minecraft` → Status überprüfen
- `screen -r minecraft` → Server in den Vordergrund holen (attach)
- `STRG+a d` → Server in den Hintergrund bringen (detach)
- `ps aux | grep forge` → Prozess-Status anzeigen lassen (inkl. CPU/RAM-Last)

**Ausführlich:** Möchte man nun überprüfen, ob der Server tatsächlich läuft, kann man dies über folgenden Befehl tun:

```
screen -ls minecraft
```

Ausgabe:

```
There is a screen on:
    916.minecraft (12.11.2016 22:36:56) (Detached)
1 Socket in /var/run/screen/S-mc.
```

Die Zeile `916.minecraft ... (Detached)` besagt, dass der Prozess mit dem Namen `minecraft` unter der Prozess-Nummer (id) `916` läuft. `Detached` bezieht sich auf den Status bezüglich `screen`. Möchte man den Minecraft-Server wieder über die Kommandozeile steuern, so kann man den Server wieder in den Vordergrund holen.

```
screen -r minecraft
```

Nun sieht man die letzten Ausgaben des Minecraft-Servers und man kann dem Minecraft-Server beliebige Befehle geben. (s. unten Wichtige Befehle ????)

In diesem Zustand kann der Minecraft-Server auch wieder gestopt werden. Dies geschieht per `stop` während man mit dem Minecraft-Server wieder verbunden ist.

Benötigt man den Minecraft-Server nicht mehr, dann kann man diesen über den folgende Befehl wieder in den Hintergrund schicken. Der Server läuft dann im Hintergrund weiter, er wird lediglich `detached` also abgekoppelt von der aktuellen Kommandozeile.

```
STRG+a d
```

Eine weitere Möglichkeit den Status des Minecraft-Servers abzufragen, ist die Prozessliste. Diese lässt sich über den folgenden Befehl abrufen:

```
ps aux | grep forge
```

Ausgabe:

```
mc 1019 0.0 0.2 5336 2244 ? Ss 23:16 0:00 SCREEN -S
```

```
minecraft -d -m java -jar -Xms512M -Xmx1008M  
forge-1.7.10-10.13.4.1558-1.7.10-universal.jar nogui  
mc 1020 23.8 36.7 1184560 362444 pts/0 Ssl+ 23:16 1:06 java -jar -  
Xms512M -Xmx1008M forge-1.7.10-10.13.4.1558-1.7.10-universal.jar nogui
```

Die erste Zeile ist der screen-Prozess mit der Id 1019, der lediglich zur Verwaltung benötigt wird. Die zweite Zeile steht für den eigentlichen Server mit der Id 1020.

Die beiden Zahlen 23.8 und 36.7 sind von Interesse, denn sie stehen für die CPU-Last bzw die RAM-Ressourcen in %, die der Server gerade benötigt. Während der Startphase kann die CPU-Last durchaus höher sein.

## Minecraft-Server automatisch bei Systemstart starten

### Zusammenfassung:

- `/bin/su mc -c "/home/mc/start_minecraft.sh,"` → in `/etc/rc.local` vor `exit 0` eintragen

**Ausführlich:** Damit der Server bei einem Neustart des Raspberry nicht von Hand neugestartet werden muss, kann folgender Befehl verwendet werden. Dieser muss in der Datei `/etc/rc.local` eingetragen werden:

WICHTIG: Die Zeile muss VOR der `exit 0` eingetragen werden.

```
/bin/su mc -c "/home/mc/start_minecraft.sh"
```

Anschließend führt man einen Neustart aus. (s. nächster Abschnitt)

## Neustart des Raspberry

### Zusammenfassung:

- `systemctl reboot -i` → Benutzer auswählen und Neustarten

Einen Neustart des Raspberry kann man über den folgenden Befehl ausführen:

```
systemctl reboot -i
```

Anschließend wird man nach einem `sudo`-Benutzer gefragt. Hier sollte man `pi` wählen und das dazugehörige Passwort eingeben. Danach führt der Raspberry einen Neustart durch.

# Wichtige Parameter für die Konfiguration des Servers

Inhalt der server.properties, über die der Server konfiguriert wird.

```
motd=WELCOME TO MY SERVER
view-distance: 5
```

Variable	Bedeutung	Beispiel
motd	Beschreibung des Servers, die im Client erscheint	WELCOME TO MY SERVER

## Befehle für die Minecraft-Server Konsole

????? Hier sollen noch ein paar Konsolen Kommandos, mit denen man den Server steuern kann hinein.

## Mods auf dem Minecraft-Server installieren

Richtig spanned wird ein Minecraft-Server für viele erst durch sogenannte Mods (Modifications, also Erweiterungen). Bei den folgenden Befehlen wird vorausgesetzt, dass die Mod im Ordner D:\downloads liegt und pscp.exe im Ordner D:\Apps\putty\pscp.exe. Die Mod, die installiert werden soll, wird in Anführungszeichen hinter mod\_file eingetragen. Im Anschluss wird man nach dem Passwort der mc-Benutzers auf dem Raspberry gefragt und es werden die Mod auf dem lokalen Rechner ebenfalls in den Minecraft-Installationsordner kopiert.

**WICHTIG:** Es dürfen keine Leerzeichen vor bzw. hinter den =-Zeichen sein, sonst wird die Variable nicht erkannt.

[copymod2raspi.bat](#)

```
@echo off
set mod_file="D:\downloads\Thaumcraft-1.7.10-4.2.3.5.jar"

set mc_local="%APPDATA%\minecraft\mods"
set mc_raspi="/home/mc/forge_1.7.10/mods/"

d:\apps\putty\pscp.exe %mod_file% minecraft@192.168.0.16:%mc_raspi%
copy %mod_file% %mc_local%
```

Variable	Bedeutung	Beispiel
mod_file	Pfad zur heruntergeladenen Mod-Datei	„D:\downloads\Thaumcraft-1.7.10-4.2.3.5.jar“
mc_local	Pfad zur Windows-Minecraft-Installation	“/home/mc/forge_1.7.10/mods/„

Variable	Bedeutung	Beispiel
mc_raspi	Pfad zu den Mods auf dem Raspberry-Server	

## Offen

D:/apps/putty/pscp MOD.zip mc@192.168.0.16:/home/mc/forge-1.7.10/mods/

Passwort für mc eingeben.

Befehlsaufbau:

Bestandteil	Bedeutung
D:/apps/putty/pscp	Pfad zum pscp-Befehl
MOD.zip	Mod-Datei, die installiert werden soll
mc	Benutzer auf (@) dem Raspberry
192.168.0.16	IP-Adresse des Raspberry (s. ip -4 addr show eth0)
:/home/mc/forge-1.7.10/mods/	Zielordner für die Mods auf dem Raspberry

- erste cli Befehle auf Server bzw Client

## Weitere Befehle mit Powershell

```
powershell Set-ExecutionPolicy Unrestricted
```

Dazwischen kann ein Powershell-Script ausgeführt werden.

```
powershell Set-ExecutionPolicy Restricted
```

1)

Mods: Modifikationen, die fast endlose Möglichkeiten bieten

2)

am besten die dritte Version, da dieser schneller ist und mehr Arbeitsspeicher besitzt

3)

ssh: **secure sh**\*ell; stellt sicheren Zugriff auf entfernte Systeme her

4)

sudo: **S**uper**U**ser **D**O; führe als Superuser/Administrator den folgenden Befehl aus

5)

CPU: Central Processing Unit

6)

graphics processing unit

7)

gibt den aktuellen Pfad aus

8)

wird im nano-Editor als ^O Speichern angezeigt

From:

<https://www.kopfload.de/> - **kopfload - Lad Dein Hirn auf!**

Permanent link:

<https://www.kopfload.de/doku.php?id=allgemein:howto:minecraft&rev=1479057728>

Last update: **2025/11/19 16:13**

