

Minecraft Forge Server auf Debian 12 GNU/Linux

Nikola Mihaylov

September 2, 2024

Contents

1	Überblick	3
2	Voraussetzungen	3
3	Erstellung der VM	3
4	Herunterladen des neuesten Debian 12 Abbilds	4
5	Installationsprozess	4
6	Erste Konfiguration nach der Installation	6
7	Herstellen einer SSH-Verbindung	9
8	Vorbereitung der Minecraft Forge Server-Installation	11
9	Installation von Minecraft Forge	14

1 Überblick

Dieses Dokument beschreibt den Installationsprozess eines Minecraft Forge Servers auf einer Debian 12 Basis, die unter Hyper-V in RemoteLabs läuft.

Bitte beachten Sie, dass dies nur für den privaten Gebrauch bestimmt ist; dieses Dokument ist nicht offiziell mit der GFN GmbH verbunden und die Anweisungen (insbesondere der Java-Abschnitt) sollten in einer Produktionsumgebung mit Vorsicht verwendet werden.

2 Voraussetzungen

Die Voraussetzungen für dieses Projekt sind wie folgt:

- Eine Instanz von Hyper-V
- Debian 12 Basisabbild (netinst)
- JDK-Paket (vorzugsweise OpenJDK, aber in diesem Tutorial wird das Oracle JDK 22 .deb-Paket verwendet)
- Forge .jar Datei

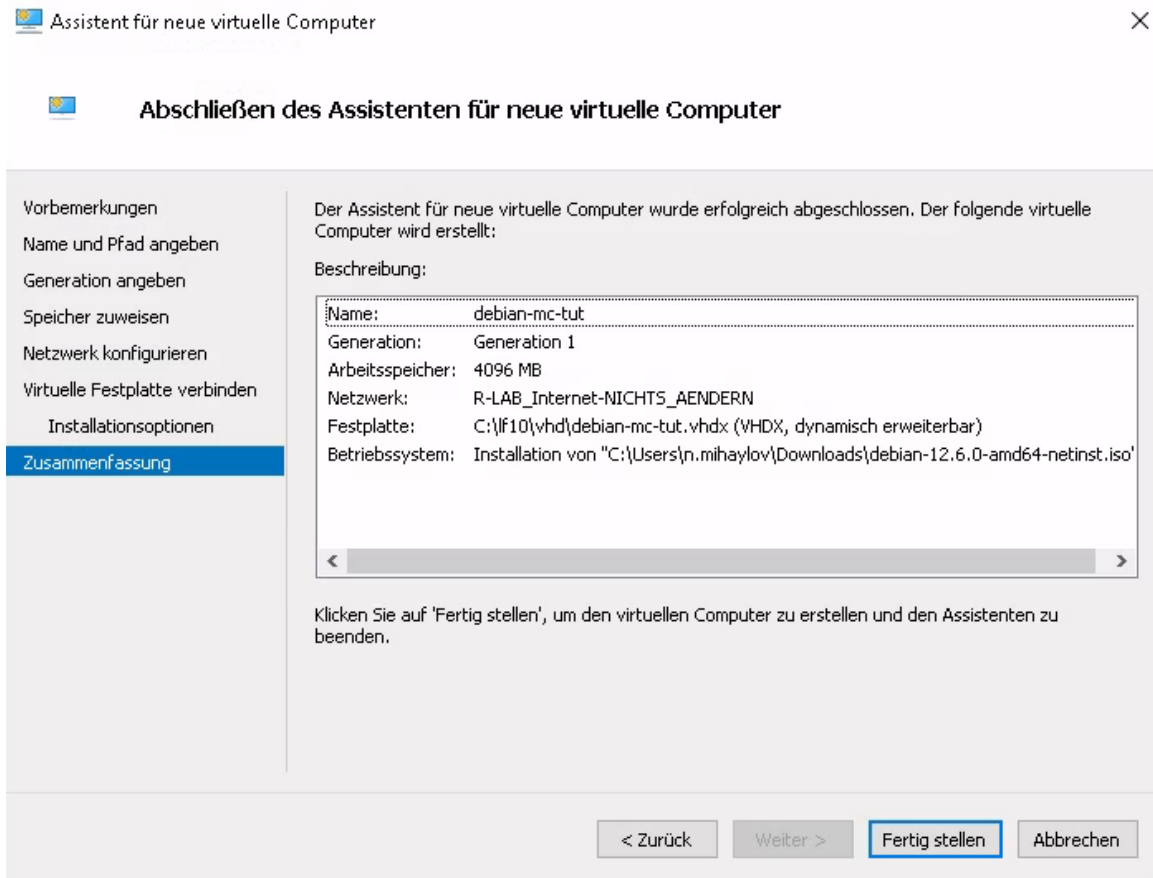
3 Erstellung der VM

Die VM wird mit den folgenden Parametern erstellt:

- Generation 1. (Gen. 2 funktioniert genauso gut, aber wir ersparen uns den Aufwand, uns mit Secure Boot unter GNU/Linux auseinanderzusetzen)
- 4096MB RAM
- 2 virtuelle Prozessoren
- 64GB dynamisch dimensionierte virtuelle Festplatte im VHDX-Format
- Netzwerkschalter "R-LAB-Internet" für die Internetverbindung (erforderlich für den Installationsprozess)

Am Ende sollten Sie Folgendes haben:

Figure 1: Übersichtsseite des Hyper-V-Assistenten



4 Herunterladen des neuesten Debian 12 Abbilds

Die neueste Version des Debian 12 Abbilds kann von der offiziellen Website heruntergeladen werden (hier klicken, um zur Download-Seite zu gelangen). Beachten Sie, dass wir für dieses Tutorial die netinst-Version verwenden, die eine Internetverbindung erfordert, um zusätzliche Pakete herunterzuladen, aber Sie können auch ein reguläres Komplettabbild für die Offline-Installation verwenden.

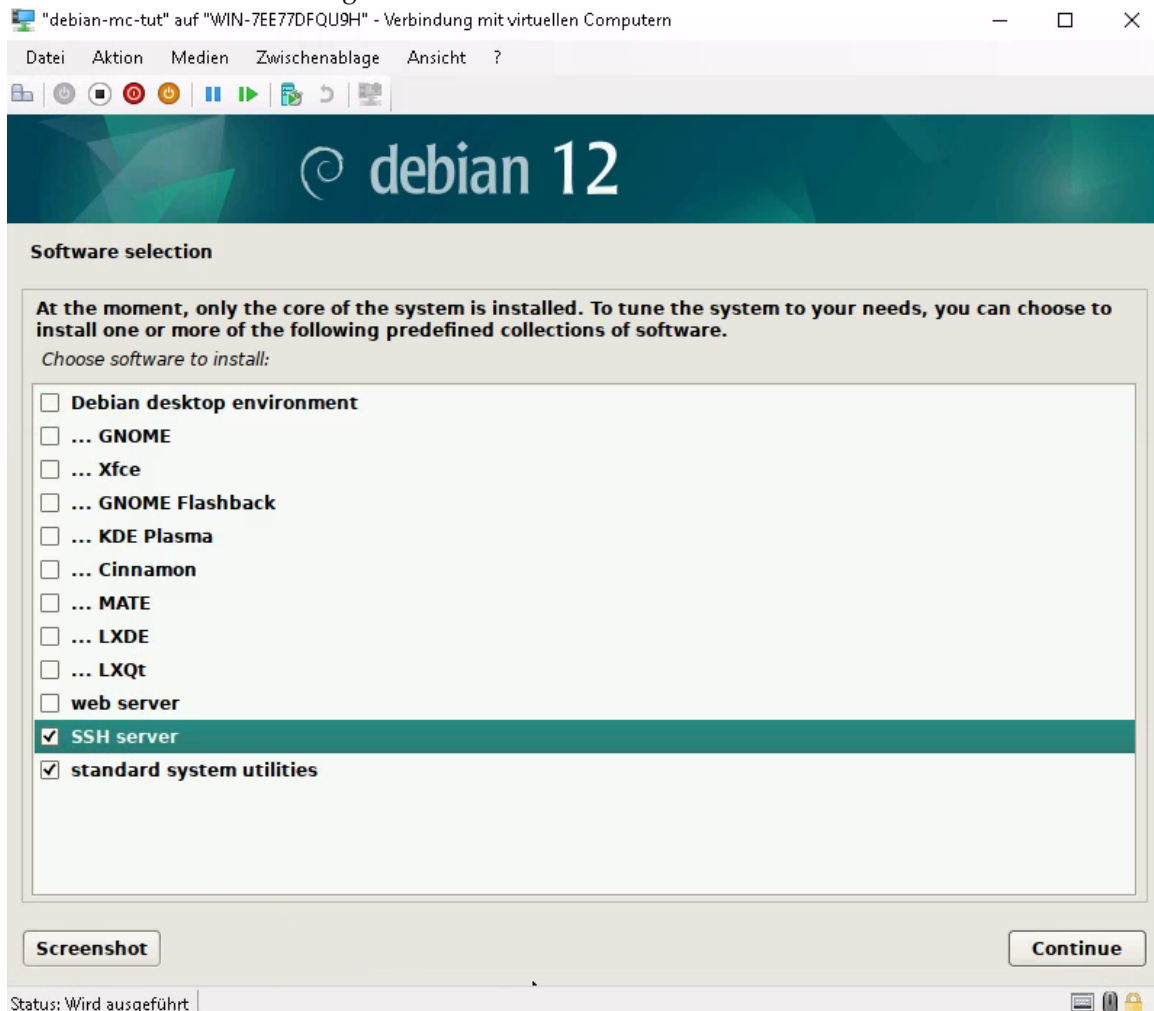
5 Installationsprozess

Die GUI-basierte Installation sollte relativ einfach sein - Sie können die Standardeinstellungen verwenden und den Hostnamen nach Belieben ändern (hier verwenden wir einfach "debian-mc"). Die Standardeinstellungen von Debian übernehmen die Festplattenpartitionierung auf unserer neuen VHDX. Stellen Sie sicher, dass Sie Ihre entsprechende Zeitzone, Sprache und Spiegelregion auswählen!

Einige Dinge, die für den Installationsprozess zu beachten sind: *nicht* eine Desktop-Umgebung installieren; wir benötigen nur eine SSH-Verbindung, um mit unserem Server zu interagieren,

da wir die Belastung durch eine grafische Umgebung vermeiden wollen, daher sollten Sie nur "SSH-Server" und "Standard-Systemwerkzeuge" auswählen (siehe Abbildung 2 für ein visuelles Beispiel).

Figure 2: Unsere Debian 12 Softwareauswahl

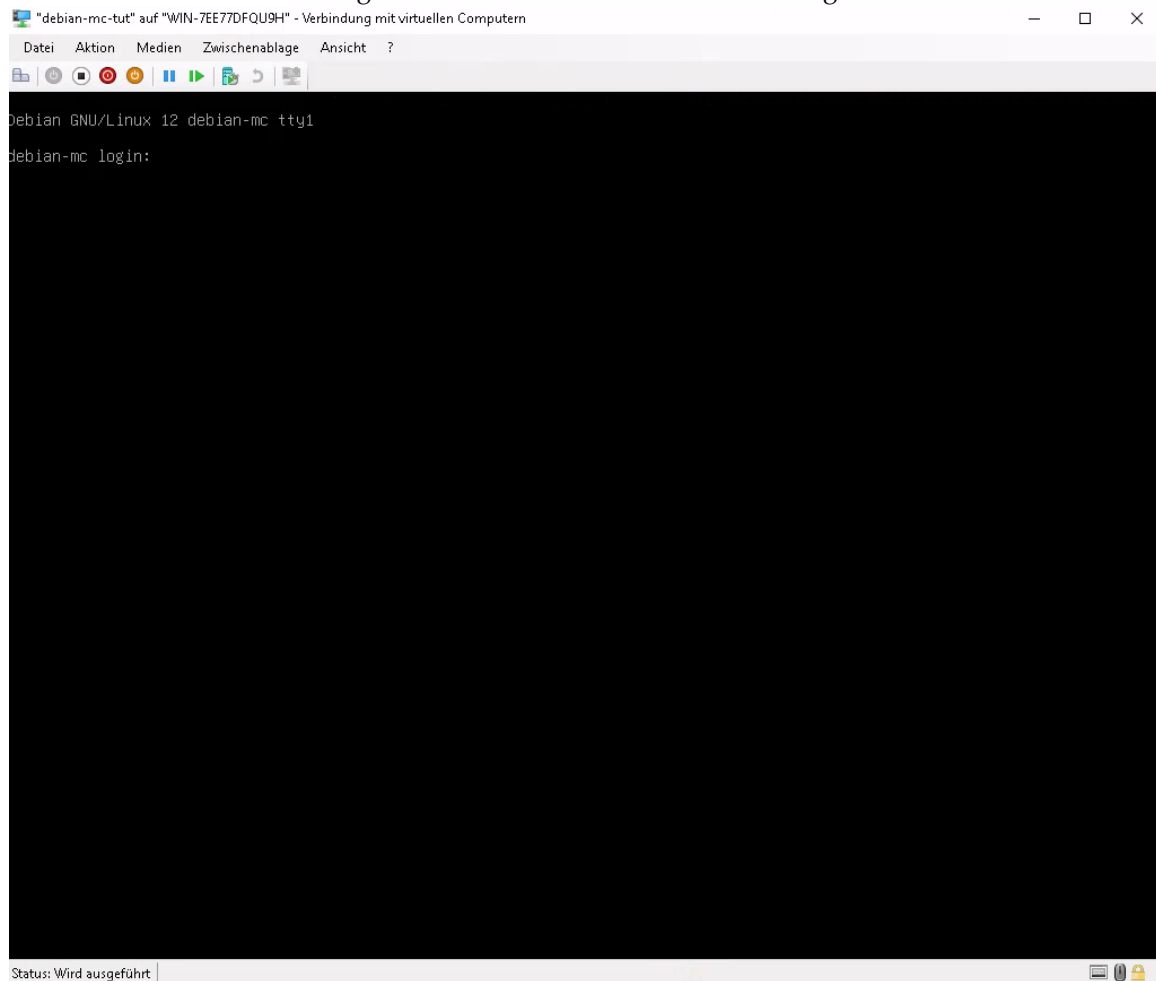


Der Grund, warum wir von Anfang an SSH haben möchten, ist, dass das Kopieren und Einfügen von Text viel einfacher wird, und Sie werden dies wahrscheinlich später während des Tutorials benötigen. Ein Benutzerkonto mit sudo-Rechten wird eine Voraussetzung für den SSH-Zugang sein, und nachdem Sie während der Installation ein Benutzerkonto erstellt haben, müssen Sie diese nur noch nach der Installation zur sudo-Gruppe hinzufügen.

Stellen Sie sicher, dass Sie den GRUB-Bootloader auf Ihrer primären Festplatte (/dev/sda) installieren, sonst können Sie nach der Installation nicht in Ihr System booten.

Jetzt sollten Sie eine funktionierende Debian 12 Installation mit einer TTY-Anmeldeaufforderung haben.

Figure 3: Die Debian 12 TTY-Aufforderung



6 Erste Konfiguration nach der Installation

Bevor wir mit etwas Minecraft-bezogenem beginnen, gibt es ein paar Dinge zu konfigurieren, wobei unser erstes und wichtigstes darin besteht, unserem neu erstellten Benutzer sudo-Rechte zu geben. Dazu melden wir uns als root an:

```
debian-mc login: root
Password: {your-root-password-here}
```

Jetzt sollten Sie als root angemeldet sein und die Eingabeaufforderung sehen.

Das erste, was zu tun ist, ist das sudo-Paket zu installieren, falls es noch nicht installiert ist. Dazu geben wir Folgendes ein:

```
apt install sudo
```

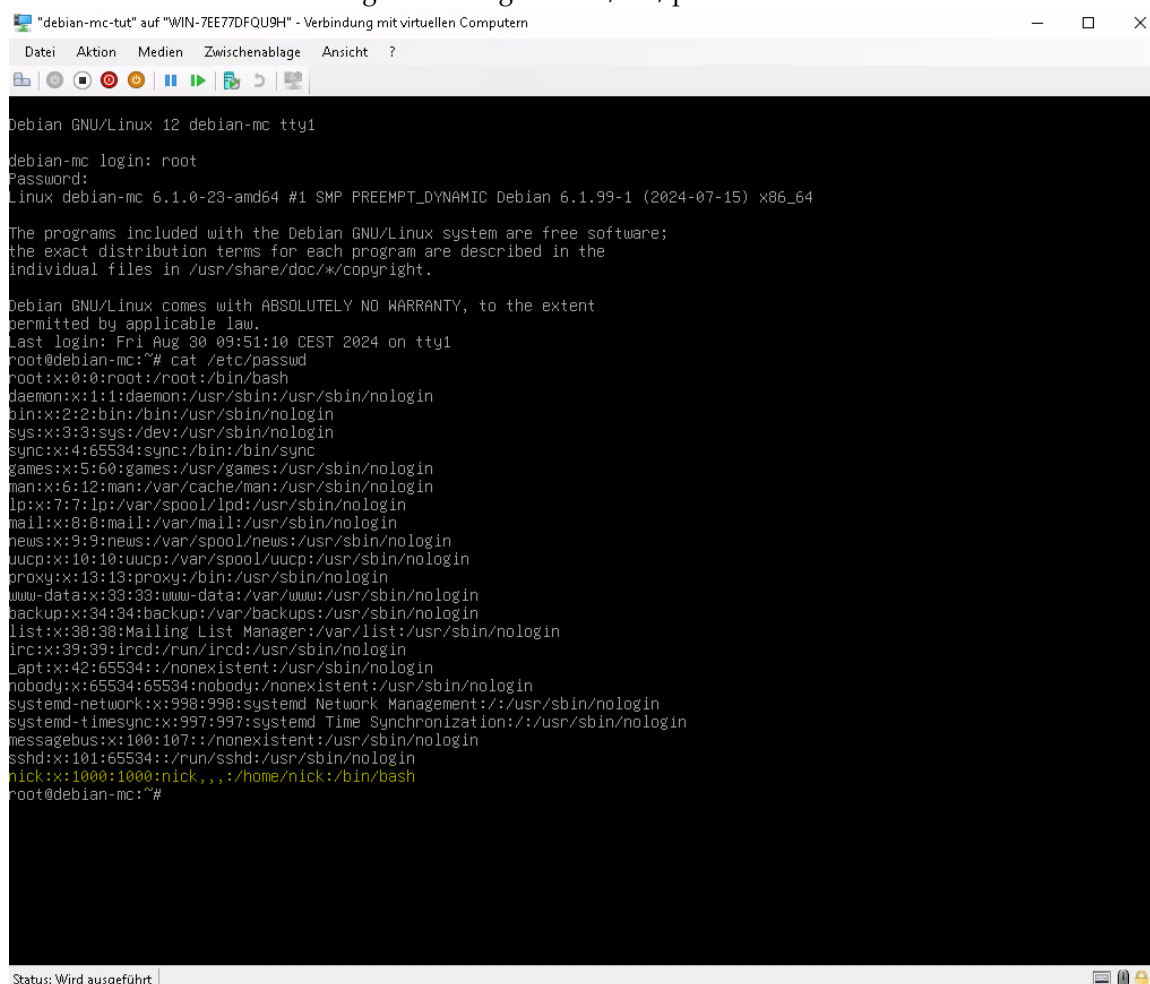
Der apt-Befehl wird verwendet, um Pakete auf dem Debian-System zu verwalten. Mit dem Flag "install" können wir einen Paketnamen angeben, der installiert werden soll, in diesem Fall das sudo-Paket. Wenn sudo auf Ihrem System noch nicht verfügbar war, wird Ihr Paketmanager Sie fragen, ob Sie das Paket installieren möchten. Bestätigen Sie dies mit Enter. Sie können anschließend sudo eingeben, um zu bestätigen, dass das Paket installiert wurde, wobei die Hilfe für den Befehl ausgegeben wird.

Jetzt, da dies erledigt ist, können wir mit unserer grundlegenden Konfiguration fortfahren. Wir können alle Benutzer auf dem System mit dem folgenden Befehl auflisten:

```
cat /etc/passwd
```

Ihr neu erstellter Benutzer aus dem grafischen Installer wird höchstwahrscheinlich ganz unten stehen. Hier können Sie die lokalen Benutzerkonten pro Zeile und ihre zugehörige Benutzer-ID (UID), Gruppen-ID (GID), das Home-Verzeichnis, die Standard-Shell und mehr sehen.

Figure 4: Ausgabe der /etc/passwd Datei



```
Debian GNU/Linux 12 debian-mc tty1
debian-mc login: root
Password:
Linux debian-mc 6.1.0-23-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.99-1 (2024-07-15) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Aug 30 09:51:10 CEST 2024 on tty1
root@debian-mc:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
Lapt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:107::/nonexistent:/usr/sbin/nologin
sshd:x:101:65534:/:run/sshd:/usr/sbin/nologin
nick:x:1000:1000:nick,,:/home/nick:/bin/bash
root@debian-mc:~#
```

Jetzt werden wir unseren Benutzer zur sudo-Gruppe hinzufügen. Die sudo-Gruppe ist eine spezielle Gruppe, die über das sudo-Kommando administrative Rechte hat. Dazu geben wir den folgenden Befehl ein:

```
usermod -aG sudo {your-username-here}
```

Der Befehl `usermod` ermöglicht es uns, die Benutzerkonten zu ändern, die wir zuvor in der Datei `/etc/passwd` gesehen haben. Die Option `-a` steht für "append", also hinzufügen eines Benutzers zu einer Gruppe. Die Option `-G` dient zur Angabe einer Gruppe nach ihrem Namen (in diesem Fall `sudo`). In vielen GNU/Linux-Tools können Sie Optionen zusammenfassen, wie `"-aG"` statt `"-a -G"`. Wenn Sie den Befehl eingegeben haben und keine Ausgabe erhalten haben, sollte alles in Ordnung sein.

Sie können jetzt mit dem Befehl `su` (Benutzer wechseln) zu Ihrem Benutzer zurückwechseln, um Ihre neu erhaltenen sudo-Rechte zu testen:

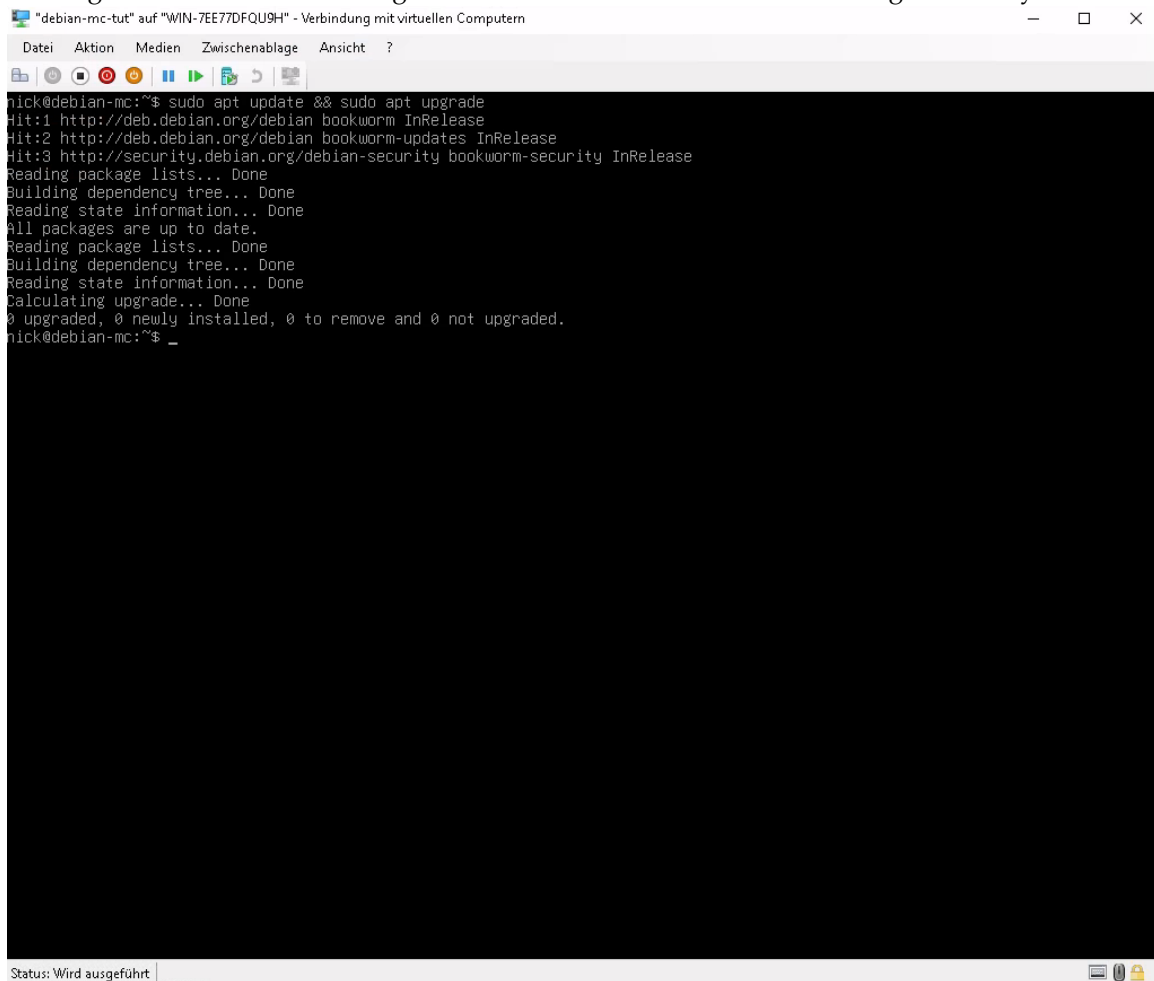
```
su {your-username-here}
```

Versuchen Sie nun, Ihr System mit `sudo` zu aktualisieren:

```
sudo apt update && sudo apt upgrade
```

Wenn alles gut gelaufen ist, sollten Sie Ihr System mit dem `apt`-Repository synchronisieren und alle verfügbaren Updates auf Ihrem Computer installieren können!

Figure 5: Testen der neu vergebenen sudo-Rechte durch Aktualisierung unseres Systems



```
nick@debian-mc:~$ sudo apt update && sudo apt upgrade
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://deb.debian.org/debian bookworm-updates InRelease
Hit:3 http://security.debian.org/debian-security bookworm-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
nick@debian-mc:~$
```

7 Herstellen einer SSH-Verbindung

Wenn Sie den Installationsabschnitt dieses Tutorials befolgt haben, haben Sie höchstwahrscheinlich OpenSSH installiert, das für diesen Abschnitt erforderlich ist. Um dies zu überprüfen, können Sie den folgenden Befehl ausführen:

```
ssh
```

Sie erhalten nun die Hilfe-Ausgabe des Befehls, wenn das OpenSSH-Paket auf Ihrem System installiert ist. Wenn nicht, installieren Sie es mit dem folgenden Befehl:

```
sudo apt install openssh-server
```

Wenn dies abgeschlossen ist, können wir zu unserem Host-System zurückkehren und ein Befehlszeilenfenster öffnen (z.B. PowerShell). Um eine Verbindung herzustellen, müssen wir eine bestimmte Syntax befolgen:

```
ssh {username}@{ipv4-address}
```

Wir verbinden uns als Benutzer auf unserem SSH-Server unter der angegebenen IPv4-Adresse des Servers. Wenn Sie Ihre IPv4-Adresse nicht kennen, können Sie den folgenden Befehl ausführen, um die IPv4-Adresse Ihres Ethernet-Adapters/NIC zu erhalten:

```
ip addr | grep "eth0"
```

Der Befehl "ip addr" gibt Informationen zu den Netzwerkschnittstellen des Systems aus. Da wir mit einer VM arbeiten, ist es sehr wahrscheinlich, dass wir einen virtuellen Ethernet-Adapter verwenden, daher verwenden wir standardmäßig "eth0", um die Informationen zu erhalten. Wenn Sie normalerweise "ip addr" ausführen, werden alle Schnittstellen angezeigt, was bei unserer frischen Installation kein Problem darstellen sollte. Wenn Sie jedoch mit Befehlen arbeiten, die viele Informationen ausgeben, von denen wir nur einen kleinen Teil benötigen, können wir gleichzeitig den Befehl "grep" verwenden. Dieses Hilfsprogramm durchsucht die Ausgabe Ihres vorherigen Befehls (in unserem Fall "ip addr") und gibt nur die Zeile aus, die als Argument für grep angegeben ist (in diesem Fall "eth0").

Die Pipe zwischen diesen beiden Befehlen ist der wichtigste Teil; dies sollte Ihre Einführung in das Konzept der Ein-/Ausgabeumleitung in Linux sein, wobei die Pipe die Ausgabe unseres linken Befehls nimmt und sie als Eingabe an den rechten Befehl weitergibt. Grob gesagt sieht der Ablauf dieses gesamten Befehls nun folgendermaßen aus: "ip addr" -> | -> "grep 'eth0'".

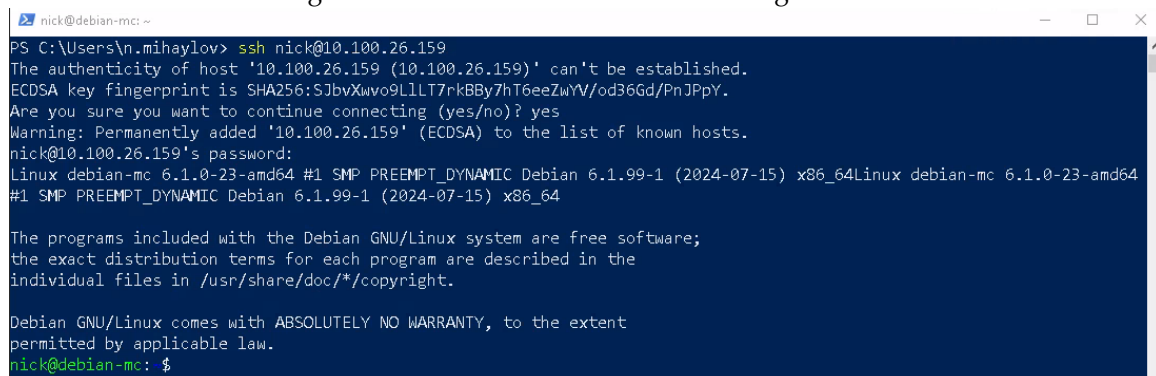
Ihre Ausgabe sollte etwa so aussehen:

Figure 6: Die Ausgabe des ip addr | grep "eth0"-Befehls

```
nick@debian-mc:~$ ip addr | grep "eth0"
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    inet 10.100.26.159/24 brd 10.100.26.255 scope global dynamic eth0
nick@debian-mc:~$
```

Mit unserer IP-Adresse in der Hand können wir nun unseren bevorzugten Terminal-Emulator verwenden und die Syntax von früher verwenden, um uns von unserem Host aus mit unserer Maschine zu verbinden:

Figure 7: Herstellen einer SSH-Verbindung zur VM



```
PS C:\Users\n.mihaylov> ssh nick@10.100.26.159
The authenticity of host '10.100.26.159 (10.100.26.159)' can't be established.
ECDSA key fingerprint is SHA256:SJbvXwvo9LLT7rkBBY7hT6eeZwYV/od36Gd/PnJPpY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.100.26.159' (ECDSA) to the list of known hosts.
nick@10.100.26.159's password:
Linux debian-mc 6.1.0-23-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.99-1 (2024-07-15) x86_64Linux debian-mc 6.1.0-23-amd64
#1 SMP PREEMPT_DYNAMIC Debian 6.1.99-1 (2024-07-15) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
nick@debian-mc: $
```

Erfolg! Sie sollten nun eine SSH-Verbindung zu Ihrer Debian 12-VM haben. Natürlich ist es wichtig zu beachten, dass sich die Maschinen im selben Netzwerk befinden sollten, aber Sie können

so weit gehen, dass Sie anstelle von RemoteLabs einen Terminal-Emulator auf Ihrem von GFN ausgestellten Laptop verwenden, um sich über SSH mit der VM zu verbinden. Damit können Sie jetzt lange Befehle einfach in Ihre VM kopieren und einfügen, ohne sie manuell eingeben zu müssen.

8 Vorbereitung der Minecraft Forge Server-Installation

Um unsere Instanz von Forge für den Minecraft-Server zu hosten, benötigen wir eine Installation des Java Development Kit auf unserem System.

Hier entsteht ein Problem: Die neueste Version von Forge erfordert Version 21 des JDK, aber diese Version von OpenJDK befindet sich zum Zeitpunkt des Schreibens dieses Dokuments noch im Test-Repository, und das Oracle JDK ist unter deren eigener Lizenz lizenziert und ebenfalls nicht in den Repositories verfügbar. An dieser Stelle muss ich klarstellen, dass dies nicht in einer Produktionsumgebung verwendet werden sollte, allein wegen der unübersichtlichen Java-Situation auf Debian.

Das alles vorausgeschickt, wird dieses Dokument mit der Oracle JDK 21 .deb-Datei fortfahren, die (hier) gefunden werden kann. Kopieren Sie den Link zur .deb-Datei von dieser Seite und halten Sie ihn in Ihrer Zwischenablage.

Zurück zur eingerichteten SSH-Verbindung zu unserem Debian 12 müssen wir einen Ordner erstellen, um unsere heruntergeladene Datei zu speichern. Dazu navigieren wir zuerst in das Home-Verzeichnis unseres Benutzers, indem wir einfach "cd" eingeben (Sie können mit dem Befehl "pwd" überprüfen, in welchem Verzeichnis Sie sich gerade befinden), und dann den Befehl "mkdir" (Verzeichnis erstellen) innerhalb unseres Home-Verzeichnisses verwenden, um ein neues wie folgt zu erstellen:

```
mkdir Downloads
```

Wenn wir nun ein "ls" (Liste) ausführen, sollten wir sehen, dass das neue Verzeichnis mit dem Namen "Downloads" erstellt wurde.

Figure 8: Erstellen eines Verzeichnisses im Home-Verzeichnis des Benutzers

```
nick@debian-mc:~$ cd
nick@debian-mc:~$ pwd
/home/nick
nick@debian-mc:~$ mkdir Downloads
nick@debian-mc:~$ cd Downloads/
nick@debian-mc:~/Downloads$ pwd
/home/nick/Downloads
nick@debian-mc:~/Downloads$ ^
```

An diesem Punkt können wir die JDK .deb-Datei, die wir derzeit in unserer Zwischenablage haben, mit dem Tool "wget" herunterladen. Der Befehl sollte ziemlich einfach sein, jetzt wo wir dank SSH eine funktionierende Zwischenablage zur Hand haben:

```
wget https://download.oracle.com/java/22/latest/jdk-22_linux-x64_bin.deb
```

Der obige Link ist für das Oracle JDK 22. Je nachdem, wann Sie dieses Dokument lesen, könnte das OpenJDK 21-Paket bereits im stabilen Repository von Debian verfügbar sein. Im Rest des Tutorials werden wir mit der heruntergeladenen .deb-Datei von Oracle fortfahren.

Wenn Sie ein "ls" in Ihrem Downloads-Verzeichnis ausführen, sollten Sie die Datei sehen, die Sie gerade mit "wget" heruntergeladen haben:

Figure 9: Die heruntergeladene JDK .deb-Datei

```
nick@debian-mc:~/Downloads$ wget https://download.oracle.com/java/22/latest/jdk-22_linux-x64_bin.deb
--2024-08-30 14:10:26-- https://download.oracle.com/java/22/latest/jdk-22_linux-x64_bin.deb
Resolving download.oracle.com (download.oracle.com) ... 23.32.100.101
Connecting to download.oracle.com (download.oracle.com)|23.32.100.101|:443 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 167396788 (160M) [text/plain]
Saving to: 'jdk-22_linux-x64_bin.deb'

jdk-22_linux-x64_bin.deb 100%[=====] 159.64M 2.30MB/s in 69s
  jdk-22_linux-x64_bin.deb  4%[==>] 6.96M 2.50MB/s
  jdk-22_linux-x64_bin.deb  4%[==>] 7.02M 2.50MB/s
  jdk-22_linux-x64_bin.deb  4%[==>] 7.09M 2.49MB/s

2024-08-30 14:11:36 (2.30 MB/s) - 'jdk-22_linux-x64_bin.deb' saved [167396788/167396788]

nick@debian-mc:~/Downloads$ ls
jdk-22_linux-x64_bin.deb
nick@debian-mc:~/Downloads$
```

Dateien mit der Endung .deb sind vergleichbar mit einer Windows .exe-Installationsdatei - sie enthalten die benötigten Binärdateien und Daten für eine Anwendung sowie alle notwendigen Meta-Informationen für das System. Tatsächlich können Sie diese Datei mit einem Archivbetrachter öffnen und sehen, dass sie aus zwei Tarball-Archiven besteht, jeweils für die Anwendung selbst und ihre Metadaten.

Um eine .deb-Datei als Systempaket zu installieren, benötigen wir den Befehl "dpkg". "apt" verwendet tatsächlich "dpkg" im Hintergrund, um die Installation von Paketen durchzuführen, aber jetzt überspringen wir "apt" und gehen direkt zu "dpkg", um eine Datei zu installieren. Wir verwenden den folgenden Befehl:

```
sudo dpkg -i jdk-22_linux-x64_bin.deb
```

Denken Sie daran, dass Sie die Tab-Vervollständigung verwenden können! Beginnen Sie mit den ersten Buchstaben des Dateinamens und verwenden Sie die Tabulatortaste, um den Rest automatisch auszufüllen. Sie werden nach Ihrem sudo-Passwort gefragt und die Installation sollte beginnen.

Wenn die Installation abgeschlossen ist, wird es alle möglichen Ausgaben auf Ihrem Terminal geben. Wenn es ähnlich wie in der folgenden Abbildung aussieht, sollten Sie gut sein.

Figure 10: Ausgabe der JDK .deb-Installation

```
nick@debian-mc:~/Downloads$ sudo dpkg -i jdk-22_linux-x64_bin.deb
[sudo] password for nick:
Selecting previously unselected package jdk-22.
(Reading database ... 33102 files and directories currently installed.)
Preparing to unpack jdk-22_linux-x64_bin.deb ...
Unpacking jdk-22 (22.0.2-ga) ...
Setting up jdk-22 (22.0.2-ga) ...
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jar to provide /usr/bin/jar (jar) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jarsigner to provide /usr/bin/jarsigner (jarsigner) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/java to provide /usr/bin/java (java) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/javac to provide /usr/bin/javac (javac) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/javadoc to provide /usr/bin/javadoc (javadoc) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/javap to provide /usr/bin/javap (javap) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jcmd to provide /usr/bin/jcmd (jcmd) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jconsole to provide /usr/bin/jconsole (jconsole) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jdb to provide /usr/bin/jdb (jdb) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jdeprscan to provide /usr/bin/jdeprscan (jdeprscan) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jdeps to provide /usr/bin/jdeps (jdeps) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jfr to provide /usr/bin/jfr (jfr) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jhsdb to provide /usr/bin/jhsdb (jhsdb) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jimage to provide /usr/bin/jimage (jimage) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jinfo to provide /usr/bin/jinfo (jinfo) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jlink to provide /usr/bin/jlink (jlink) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jmap to provide /usr/bin/jmap (jmap) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jmod to provide /usr/bin/jmod (jmod) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jpackage to provide /usr/bin/jpackage (jpackage) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jps to provide /usr/bin/jps (jps) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jrunscript to provide /usr/bin/jrunscript (jrunscript) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jshell to provide /usr/bin/jshell (jshell) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jstack to provide /usr/bin/jstack (jstack) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jstat to provide /usr/bin/jstat (jstat) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jstatd to provide /usr/bin/jstatd (jstatd) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/jwebserver to provide /usr/bin/jwebserver (jwebserver) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/keytool to provide /usr/bin/keytool (keytool) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/rmiregistry to provide /usr/bin/rmiregistry (rmiregistry) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/bin/serialver to provide /usr/bin/serialver (serialver) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64 to provide /usr/java/jdk-22-oracle-x64 (java_jdk_22_oracle) in auto mode
update-alternatives: using /usr/lib/jvm/jdk-22.0.2-oracle-x64/lib/jexec to provide /usr/bin/jexec (jexec) in auto mode
nick@debian-mc:~/Downloads$
```

Sie können die Installation überprüfen, indem Sie `java --version` ausführen, was hoffentlich die Version des Pakets anzeigt, das Sie gerade installiert haben (in diesem Fall 22.0.2). Sie haben nun erfolgreich das benötigte Java JDK für die Ausführung von Forge installiert!

9 Installation von Minecraft Forge

Wir sollten zuerst ein dediziertes Verzeichnis für unsere Minecraft-Installation erstellen und in dieses wechseln:

```
sudo mkdir /opt/minecraft
cd /opt/minecraft
```

Jetzt können wir mit "wget" die neueste empfohlene Version von Forge herunterladen, die zum Zeitpunkt des Schreibens dieses Dokuments verfügbar ist. Diese können Sie von (hier) beziehen. Wenn Sie "wget" mit diesem Link als Argument ausführen, wird die Datei in unserem Verzeichnis /opt/minecraft heruntergeladen:

```
sudo wget https://maven.minecraftforge.net/.../...installer.jar
```

Überprüfen Sie mit einem "ls", ob die Datei im aktuellen Verzeichnis heruntergeladen wurde. Wir können nun mit der Installation der Server-Version von Minecraft Forge mit der .jar-Datei unter Verwendung von Java fortfahren:

```
sudo java -jar forge-1.20.6-50.1.0-installer.jar --installServer
```

Dieser Vorgang kann eine Weile dauern. Sie wissen, dass er erfolgreich abgeschlossen wurde, wenn die Meldung "The server installed successfully" am Ende erscheint.

Der nächste Schritt besteht darin, eine "eula.txt"-Datei zu erstellen, die vom Minecraft-Server benötigt wird. Wir können entweder die Datei mit "touch" erstellen, um sie ohne Inhalt zu erstellen, oder sie mit einem Texteditor wie "nano" erstellen, da wir den Inhalt ohnehin in die Datei schreiben müssen.

Durch "Touching" einer Datei wird eine leere Datei mit dem Namen erstellt, den Sie dem Touch-Befehl angegeben haben, wie folgt:

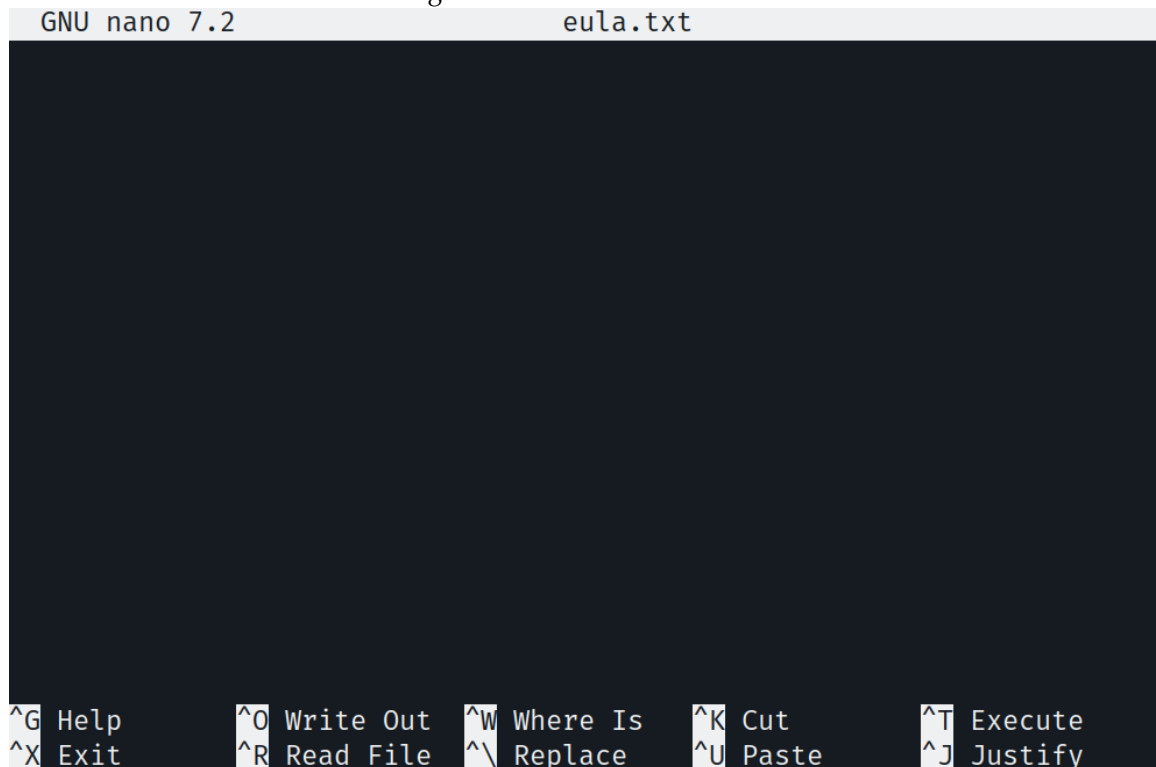
```
sudo touch eula.txt
```

Wenn Sie nun das Verzeichnis mit "ls" auflisten, sollten Sie die Datei sehen. Wenn Sie ihren Inhalt mit "cat" ausgeben, werden Sie nichts sehen, da sie leer ist. Zum Bearbeiten verwenden wir den Texteditor "nano":

```
sudo nano eula.txt
```

Jetzt befinden Sie sich in der Benutzeroberfläche des nano-Texteditors:

Figure 11: Der nano-Texteditor



nano ist ein relativ einfacher Texteditor - Sie können direkt loslegen und Text in den Editor-Puffer schreiben und ihn mit Strg+O speichern, um ihn "auszuschreiben", und mit Enter bestätigen. Hier müssen wir nur die folgende Zeile hinzufügen:

```
eula=true
```

Schreiben Sie den Puffer mit Strg+O in die Datei und drücken Sie Enter, um unter dem Standarddateinamen zu speichern. Wenn beim Speichern ein Fehler auftritt, haben Sie wahrscheinlich vergessen, nano als Superuser mit dem sudo-Befehl zu starten.

Wenn Sie immer noch Lust auf Ausgabeumleitung haben, können wir das Verwenden von "nano" überspringen und unsere Bash-Shell diese Zeile mit diesem Befehl in unsere Datei schreiben lassen:

```
echo "eula=true" | sudo tee -a eula.txt
```

Der Befehl "echo" gibt alles, was danach kommt, in die Shell aus, aber die Pipe nimmt diese Ausgabe und übergibt sie an "tee" mit sudo-Rechten, um sie mit dem -a-Flag an die eula.txt-Datei anzuhängen. Der Befehl "tee" ist ein Dienstprogramm, das die Ausgabe eines Befehls liest, sie auf dem Bildschirm anzeigt und sie auch in eine Ausgabedatei schreibt (in diesem Fall die "eula.txt"). Während es bequemer ist, diese Zeile mit einem Texteditor wie nano zu unserer Datei hinzuzufügen, ist das Konzept der Umleitung wichtig, wenn es darum geht, Skripte zu erstellen, die Ihnen auf Ihrer Linux-Reise nützlich sein könnten.

Wir können den Inhalt der Datei mit "cat" überprüfen, das das, was wir gerade in die Datei geschrieben haben, ausgeben sollte.

Lassen Sie uns nun eine leere Datei "server.properties" mit "touch" erstellen:

```
touch server.properties
```

Lassen Sie uns auch eine vorhandene Datei in diesem Verzeichnis mit dem Namen `user_jvm_args.txt` ändern. Diese Datei sollte die Anweisungen enthalten, wie die Java Virtual Machine gestartet werden soll.

```
sudo nano user_jvm_args.txt
```

Alle Zeilen, die mit einem # beginnen, werden ignoriert, da sie als Kommentare interpretiert werden; alles, was danach folgt und kein # hat, wird als Eintrag in der Konfigurationsdatei gelesen. Wir müssen nur Folgendes hinzufügen:

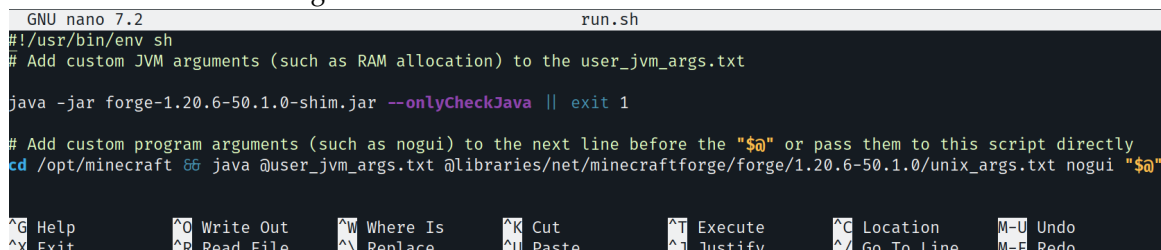
```
-Xms4G
-Xmx4G
```

Damit wird der JVM mindestens 4 GB und maximal 4 GB RAM zugewiesen. Beachten Sie, dass die ersten X's groß geschrieben sind.

Alles, was noch zu tun ist, ist die "run.sh"-Shell-Skriptdatei zu bearbeiten, um die allerletzte Zeile zu ändern, indem wir `cd /opt/minecraft &&` am Anfang der Zeile hinzufügen und ein `nogui` anhängen.

Ihre Datei sollte nun so aussehen:

Figure 12: Der Inhalt der bearbeiteten run.sh-Datei



```
GNU nano 7.2 run.sh
#!/usr/bin/env sh
# Add custom JVM arguments (such as RAM allocation) to the user_jvm_args.txt
java -jar forge-1.20.6-50.1.0-shim.jar --onlyCheckJava || exit 1

# Add custom program arguments (such as nogui) to the next line before the "$@" or pass them to this script directly
cd /opt/minecraft && java @user_jvm_args.txt @libraries/net/minecraftforge/forge/1.20.6-50.1.0/unix_args.txt nogui "$@"

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^N Replace    ^J Paste      ^_ Justify    ^_/ Go To Line M-E Redo
```

Wenn hoffentlich alles abgeschlossen ist, können wir endlich das Shell-Skript ausführen, um unseren Minecraft-Server zu starten:

```
sudo sh run.sh
```

Dieser Vorgang dauert eine Weile, insbesondere während der Weltgenerierungsphase. Sie werden wissen, dass es fertig ist, wenn die Meldung "Done" und das ;-Zeichen der Minecraft-Konsolen-Eingabeaufforderung erscheinen, sobald alles eingerichtet ist. Sie können nun versuchen, sich mit Ihrem neu erstellten Minecraft-Server zu verbinden, wobei Sie beachten sollten, dass Sie möglicherweise eine bestimmte Version des Clients benötigen (in diesem Fall habe ich die Client-Version 1.26.0 verwendet, die vom Prism Launcher verwaltet wird).

Figure 13: Beitritt zum Server mit der IPv4-Adresse

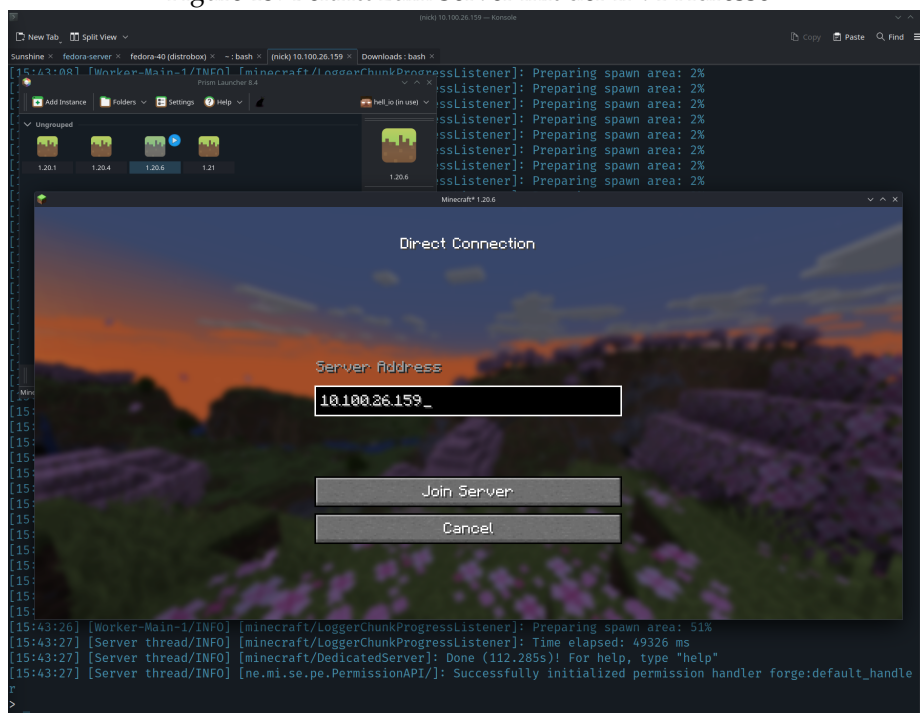
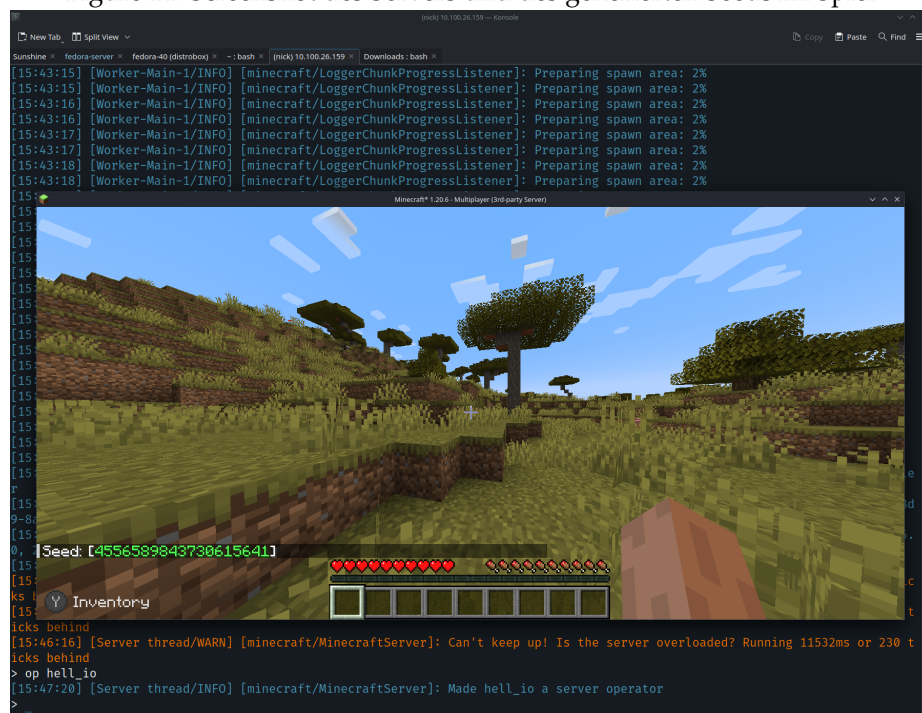


Figure 14: Screenshot des Servers und des generierten Seeds im Spiel



Herzlichen Glückwunsch! Sie sollten nun einen funktionierenden Minecraft Forge-Server auf Ihrer Debian 12-VM haben!