

Konfiguracja serwera Minecraft

Dokumentacja konfiguracyjna

Informacja wstępna

Wiadomo czym jest gra Minecraft. Zdecydowanie lepiej jest grać w tą grę ze znajomymi. Aby było to możliwe to potrzebujemy przestrzeni do grania, a żeby była ta przestrzeń potrzebujemy skonfigurować sobie wpierw serwer, który będzie hostował nam serwer. Czyli Jakie pakiety są potrzebne do skonfigurowania serwera Minecraft na systemie openSUSE?

Do obsługi serwera Minecraft będziemy potrzebowali pakietu Java. Dokładnie JDK. Najlepszym rozwiązaniem będzie skorzystanie z pakietów **openjdk-devel**. Aby wyszukać dostępną najnowszą wersję tego pakietu skorzystamy z wyszukiwarki pakietów w **zypper**:

```
# Wyszukiwanie pakietu
sudo zypper search openjdk-devel
```

```
siemv:/home/administrator # zypper search openjdk-devel
Loading repository data...
Reading installed packages...

S | Name | Summary | Type
-----|-----|-----|-----
| java-1_8_0-openjdk-devel | OpenJDK 8 Development Environment | package
| java-9-openjdk-devel | OpenJDK 9 Development Environment | package
| java-10-openjdk-devel | OpenJDK 10 Development Environment | package
| java-11-openjdk-devel | OpenJDK 11 Development Environment | package
| java-17-openjdk-devel | OpenJDK 17 Development Environment | package
```

Jak widać najnowszą dostępną wersją JDK jest Java 17. Więc ten pakiet wykorzystamy.

Instalacja pakietu Java

```
# Aktualizacja wstępna wszystkich pakietów (nie wymagane podczas zajęć)
sudo zypper up
```

```
# Instalacja pakietu odpowiedzialnego za OpenJDK 17
sudo zypper in java-17-openjdk-devel
```

Konfiguracja serwera

Do działania naszego serwera musimy pobrać jeszcze plik serwerowy, który pozwoli na uruchomienie takiego serwera. Tutaj pozostaje kwestia zastanowienia się, jaki serwer chcielibyśmy zrobić. Czy będzie to czysty serwer, serwer z pluginami, czy serwer na modach. W przykładzie wykorzystamy silnik PaperMC. Stwórzmy sobie przestrzeń, w której będzie nasz serwer się znajdował (czyt. Katalog) i pobierzmy sobie plik serwerowy. Możemy pobrać plik za pomocą komendy **wget**. Jest to komenda, która pozwala na pobieranie pliku przy użyciu **HTTP, HTTPS, FTP, SFTP**.

```
siemv:/home/administrator # mkdir Serwer
siemv:/home/administrator # cd Serwer/
siemv:/home/administrator/Serwer # wget -O server.jar https://api.papermc.io/v2/projects/paper/versions/1.20.4/builds/464/downloads/paper-1.20.4-464.jar
--2024-04-02 22:14:49-- https://api.papermc.io/v2/projects/paper/versions/1.20.4/builds/464/downloads/paper-1.20.4-464.jar
Resolving api.papermc.io (api.papermc.io)... 104.26.12.138, 172.67.72.198, 104.26.13.138, ...
Connecting to api.papermc.io (api.papermc.io)|104.26.12.138|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 42769529 (41M) [application/java-archive]
Saving to: 'server.jar'

server.jar          100%[=====>] 40.79M  36.6MB/s   in 1.1s

2024-04-02 22:14:50 (36.6 MB/s) - 'server.jar' saved [42769529/42769529]

siemv:/home/administrator/Serwer #
```

Teraz musimy stworzyć sobie skrypt Bash, który pozwoli na uruchomienie naszego serwera gry Minecraft. Do uruchomienia serwera będziemy potrzebowali teraz "zbudować" komendę do uruchomienia tego serwera.

Składnia komendy, dzięki której uruchomimy serwer z podstawowymi parametrami pamięci:

```
java -Xms2G -Xmx4G -jar server.jar nogui
```

- **Xms** – minimalna pamięć RAM dla procesu
- **Xmx** – maksymalna pamięć RAM dla procesu
- **jar** – plik jaki chcemy uruchomić za pomocą **Java**
- **nogui** – parametr przekazywany do pliku **server.jar** w którym wskazujemy mu, że wszystkie logi mają być w terminalu. Niedodanie tego parametru spowoduje otwieranie dodatkowego okna (w zależności od wybranego pliku serwerowego)
- **server.jar** – nazwa naszego pliku serwerowego. Może się różnić w zależności od tego jak zapisaliśmy nasz plik

Pierwsze uruchomienie serwera pobierze tylko biblioteki oraz inne pliki potrzebne do uruchomienia serwera. Możemy zobaczyć też, że na końcu zamyka nasz serwer i wymaga abyśmy zaakceptowali klauzule EULA.

```
siemv:/home/administrator/Serwer # java -Xms2G -Xmx4G -jar server.jar
Downloading mojang_1.20.4.jar
Applying patches
Starting org.bukkit.craftbukkit.Main
System Info: Java 17 (OpenJDK 64-Bit Server VM 17.0.10+7-suse-150400.3.39.2-x8664) Host: Linux 5.14.21-150500.55
.49-default (amd64)
Loading libraries, please wait...
[22:21:25 WARN]: Failed to load eula.txt
[22:21:25 INFO]: You need to agree to the EULA in order to run the server. Go to eula.txt for more info.
siemv:/home/administrator/Serwer # ls
cache eula.txt libraries logs plugins server.jar server.properties versions
siemv:/home/administrator/Serwer #
```

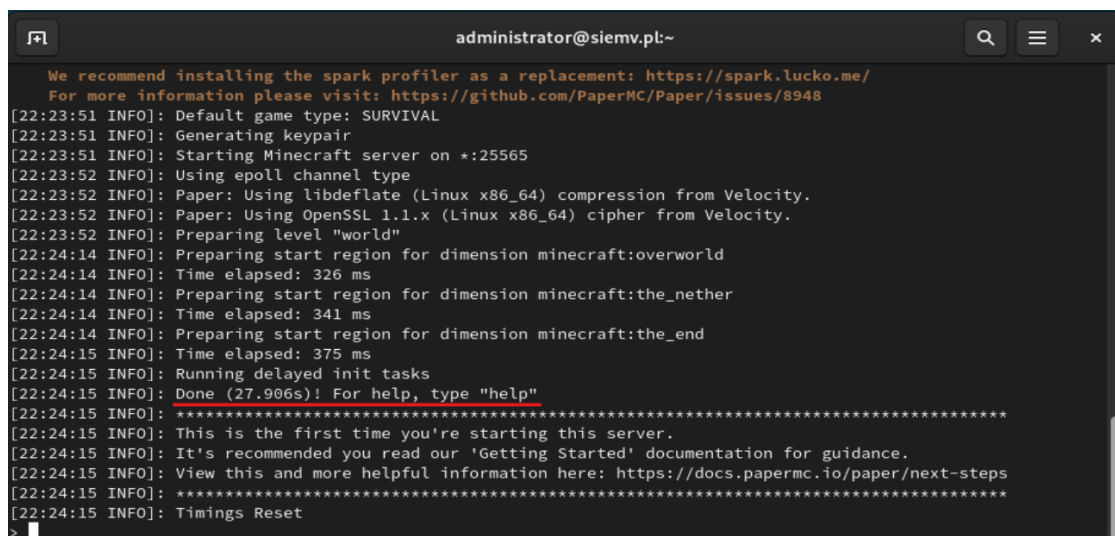
Możemy to zrobić z poziomu terminala nadpisując plik **eula.txt** za pomocą komendy:

```
echo eula=true > eula.txt
```

```
siemv:/home/administrator/Serwer # echo eula=true > eula.txt
siemv:/home/administrator/Serwer # cat eula.txt
eula=true
siemv:/home/administrator/Serwer #
```

Po zmianie na wartość **true** możemy uruchomić już nasz serwer Minecraft. Będziemy wiedzieli, że nasz serwer jest uruchomiony, kiedy ujrzymy komunikat informacyjny:

```
[HH:MM:SS INFO]: Done (SS.MMMs)! For help, type "help".
```



```
administrator@siemv.pl~
We recommend installing the spark profiler as a replacement: https://spark.lucko.me/
For more information please visit: https://github.com/PaperMC/Paper/issues/8948
[22:23:51 INFO]: Default game type: SURVIVAL
[22:23:51 INFO]: Generating keypair
[22:23:51 INFO]: Starting Minecraft server on *:25565
[22:23:52 INFO]: Using epoll channel type
[22:23:52 INFO]: Paper: Using libdeflate (Linux x86_64) compression from Velocity.
[22:23:52 INFO]: Paper: Using OpenSSL 1.1.x (Linux x86_64) cipher from Velocity.
[22:23:52 INFO]: Preparing level "world"
[22:24:14 INFO]: Preparing start region for dimension minecraft:overworld
[22:24:14 INFO]: Time elapsed: 326 ms
[22:24:14 INFO]: Preparing start region for dimension minecraft:the_nether
[22:24:14 INFO]: Time elapsed: 341 ms
[22:24:14 INFO]: Preparing start region for dimension minecraft:the_end
[22:24:15 INFO]: Time elapsed: 375 ms
[22:24:15 INFO]: Running delayed init tasks
[22:24:15 INFO]: Done (27.906s)! For help, type "help"
[22:24:15 INFO]: *****
[22:24:15 INFO]: This is the first time you're starting this server.
[22:24:15 INFO]: It's recommended you read our 'Getting Started' documentation for guidance.
[22:24:15 INFO]: View this and more helpful information here: https://docs.papermc.io/paper/next-steps
[22:24:15 INFO]: *****
[22:24:15 INFO]: Timings Reset
>
```

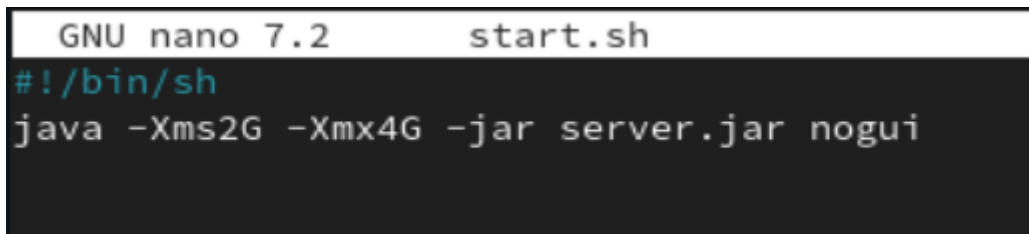
Aby zakończyć działanie naszego serwera wpisujemy komendę **stop** lub wciskamy **Ctrl+C**.

Naszą komendę teraz zapiszmy do pliku Bash, np. **start.sh**. W pierwszej linijce dajmy:

```
#!/bin/sh
```

Aby wspomnieć naszemu systemowi, że jest to plik wykonywalny Bash który ma być wykonany przez **shell**, a w następnej linijce zapiszmy naszą komendę:

```
java -Xms2G -Xmx4G -jar server.jar
```



```
GNU nano 7.2      start.sh
#!/bin/sh
java -Xms2G -Xmx4G -jar server.jar nogui
```

Zapiszmy I na koniec zrobmy nasz plik **start.sh** plikiem wykonywalnym dodając parametr **wykonać** za pomocą komendy **chmod**:

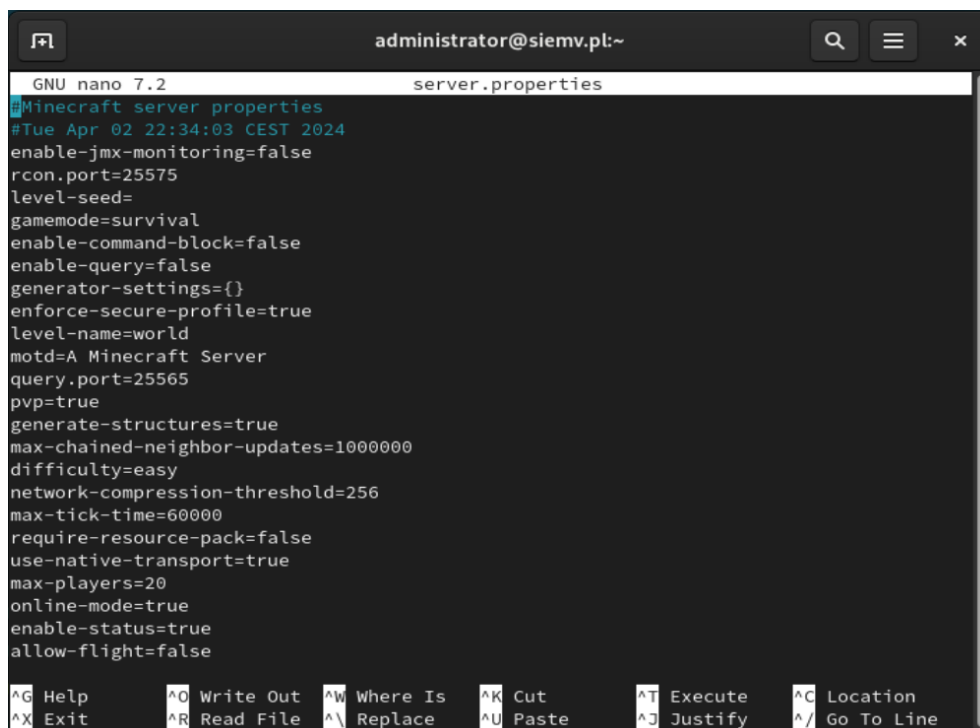
```
chmod +x start.sh
```

Jeżeli chcemy uruchomić nasz serwer z poziomu terminala wpisujemy teraz **./start.sh**. Pamiętajcie, aby być w miejscu, gdzie się znajduje nasz serwer!



```
siemv:/home/administrator/Serwer # ./start.sh
```

Aby jednak nasz serwer działał na zewnątrz musimy pamiętać o konfiguracji Zapory sieciowej naszego Linuxa. Jeżeli chcemy zmodyfikować np. liczbę slotów, poziom trudności, czy chcemy mieć PvP, tryb online, czy chcemy mieć Command Blocki to możemy ustawić je w pliku **server.properties**.



```
administrator@siemv.pl:~
GNU nano 7.2      server.properties
Minecraft server properties
#Tue Apr 02 22:34:03 CEST 2024
enable-jmx-monitoring=false
rcon.port=25575
level-seed=
gamemode=survival
enable-command-block=false
enable-query=false
generator-settings={}
enforce-secure-profile=true
level-name=world
motd=A Minecraft Server
query.port=25565
pvp=true
generate-structures=true
max-chained-neighbor-updates=1000000
difficulty=easy
network-compression-threshold=256
max-tick-time=60000
require-resource-pack=false
use-native-transport=true
max-players=20
online-mode=true
enable-status=true
allow-flight=false
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute   ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify   ^_ Go To Line
```

Aby działał nasz serwer poza siecią lokalną, np. chcemy, aby znajomy zagrał na naszym serwerze ze swojego domu potrzebujemy skonfigurować przekierowanie portów na routerze lub skorzystać z oprogramowania takiego jak Hamachi.